How to have more things by forgetting where you put them

Mike Oliver

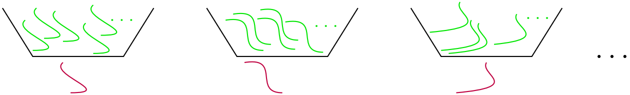
March 17, 2010

Two questions, not obviously related

- Given a string of letters, reaching off to infinity in both directions, is there any rule for finding its “center” — that is, the origin?

Two questions, not obviously related

- Given a string of letters, reaching off to infinity in both directions, is there any rule for finding its "center" — that is, the origin?
- In a picture like this:



can there ever be, in any sense, more purple squiggles than green ones?

Doubly infinite strings

The “things” I want to consider are doubly infinite strings on a finite alphabet, which might as well be the two letters 0 and 1:

$$\ldots 10111101000100010100001 \ldots$$

Formally, these can be represented as maps from the integers to the set $\{0,1\}$; each string is a function $f : \mathbb{Z} \to \{0,1\}$.

Doubly infinite strings

The "things" I want to consider are doubly infinite strings on a finite alphabet, which might as well be the two letters 0 and 1:

$$\ldots 1011110100010001010000 1\ldots$$

Formally, these can be represented as maps from the integers to the set $\{0, 1\}$; each string is a function $f : \mathbb{Z} \to \{0, 1\}$.
Or. . . can they?

Doubly infinite strings

The “things” I want to consider are doubly infinite strings on a finite alphabet, which might as well be the two letters 0 and 1:

$$\ldots 1011110100010001010 0001\ldots$$

Formally, these can be represented as maps from the integers to the set $\{0, 1\}$; each string is a function $f : \mathbb{Z} \to \{0, 1\}$.

Or... can they?

Actually, given such an f, there's a particular value for $f(0)$. So really one of these maps f is more like:

$$\ldots 10111101{\color{red}0}00100010100001\ldots$$

where $f(0)$ is marked in red, $f(1)$ is the letter to the right of the red one, $f(-1)$ is the one to the left of the red one, and so on.

Forgetting the origin

So, if we really want to model doubly infinite strings, we have to forget where the origin is. That is, we want

$$\ldots 111111111\underline{001110\textcolor{red}{0}}1111111111111\ldots$$

with the 1s repeating infinitely in both directions, to be the same string as

$$\ldots 1111\underline{0011100}1111\textcolor{red}{1}1111111111111\ldots$$

again with 1s stretching out to infinity.

The underlining is not meant to have any significance for the string — it's just to make it visually clear why the two strings are the same. Or rather, would be the same, if we could forget which letter is red.

Forgetting the origin

So, if we really want to model doubly infinite strings, we have to forget where the origin is. That is, we want

$$\ldots 111111111\underline{0011100}1111111111111\ldots$$

with the 1s repeating infinitely in both directions, to be the same string as

$$\ldots 1111\underline{0011100}111111111111111111\ldots$$

again with 1s stretching out to infinity.
The underlining is not meant to have any significance for the string — it's just to make it visually clear why the two strings are the same. Or rather, would be the same, if we could forget which letter is red. Like so.

Forgetting is hard

Well, not really *hard*, but the formalities might not be obvious if you haven't done them before.

- We're pretty much stuck with functions $f : \mathbb{Z} \to \{0, 1\}$.
- But here's the trick: We'll say that two such functions are *equivalent* for our (momentary) purposes, if they are constant shifts of each other.

Forgetting is hard

Well, not really *hard*, but the formalities might not be obvious if you haven't done them before.

- We're pretty much stuck with functions $f : \mathbb{Z} \to \{0, 1\}$.
- But here's the trick: We'll say that two such functions are *equivalent* for our (momentary) purposes, if they are constant shifts of each other.
- Given f and g, if, say, you can shift f by three spaces and get g; that is, $f(n+3) = g(n)$ for *every* integer n, then f and g are equivalent, $f \sim g$. Or, any other integer in place of the 3.
- Then, formally, our objects are the quotient of the functions from $\mathbb{Z}$ to $\{0, 1\}$, modulo $\sim$.

Forgetting is hard

Well, not really *hard*, but the formalities might not be obvious if you haven't done them before.

- We're pretty much stuck with functions $f : \mathbb{Z} \to \{0, 1\}$.
- But here's the trick: We'll say that two such functions are *equivalent* for our (momentary) purposes, if they are constant shifts of each other.
- Given f and g, if, say, you can shift f by three spaces and get g; that is, $f(n+3) = g(n)$ for *every* integer n, then f and g are equivalent, $f \sim g$. Or, any other integer in place of the 3.
- Then, formally, our objects are the quotient of the functions from $\mathbb{Z}$ to $\{0, 1\}$, modulo $\sim$.
- But mostly we won't get that formal. We'll just require that all "well-defined" questions about doubly infinite strings, *must give the same answer* for f as for g, if f and g are shifts of one another.

What does "more" mean?

After all, we're talking about infinite sets. Aren't all infinities the same?

What does “more” mean?

After all, we’re talking about infinite sets. Aren’t all infinities the same?

- No.

What does “more” mean?

After all, we’re talking about infinite sets. Aren’t all infinities the same?

- No.
- Although a lot of infinite sets are of equal size, when you might not think so.

What does “more” mean?

After all, we’re talking about infinite sets. Aren’t all infinities the same?

- No.
- Although a lot of infinite sets are of equal size, when you might not think so.
- For example, there are exactly as many rational numbers as natural numbers ($\aleph_0$). Both sets are *countably infinite*.

What does "more" mean?

After all, we're talking about infinite sets. Aren't all infinities the same?

- No.
- Although a lot of infinite sets are of equal size, when you might not think so.
- For example, there are exactly as many rational numbers as natural numbers ($\aleph_0$). Both sets are *countably infinite*.
- But there are strictly more real numbers ($2^{\aleph_0}$) than natural numbers. The set of real numbers is *uncountable*.

What does "more" mean?

After all, we're talking about infinite sets. Aren't all infinities the same?

- No.
- Although a lot of infinite sets are of equal size, when you might not think so.
- For example, there are exactly as many rational numbers as natural numbers ($\aleph_0$). Both sets are *countably infinite*.
- But there are strictly more real numbers ($2^{\aleph_0}$) than natural numbers. The set of real numbers is *uncountable*.

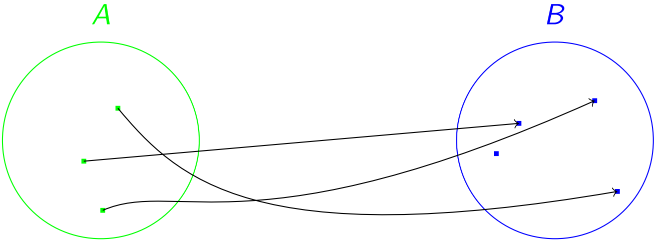
These facts can be seen via very simple arguments due to Georg Cantor, which you have all likely seen.

OK, so what does it mean?

- To claim that set A has fewer elements than (or the same number as) set B, in symbols

$$|A| \leq |B|$$

OK, so what does it mean?

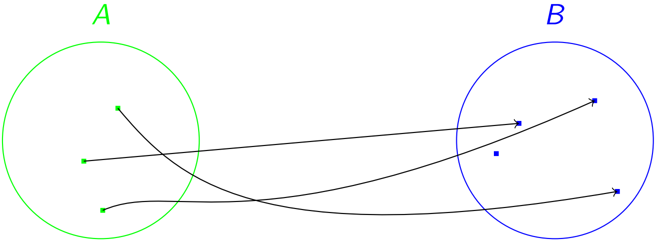


- To claim that set A has fewer elements than (or the same number as) set B, in symbols

$$|A| \leq |B|$$

we need a map that sends every element of A to a *unique* element of B.

OK, so what does it mean?

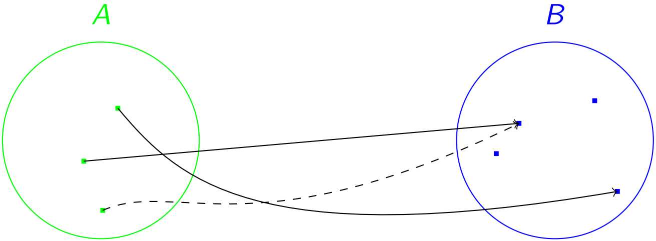


- To claim that set A has fewer elements than (or the same number as) set B, in symbols

$$|A| \leq |B|$$

  we need a map that sends every element of A to a *unique* element of B.
- "Unique" meaning that if you have two different things in A, their arrows don't collide on the B side.

OK, so what does it mean?

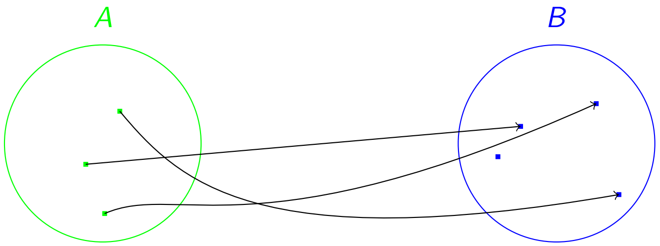


- To claim that set A has fewer elements than (or the same number as) set B, in symbols

$$|A| \leq |B|$$

  we need a map that sends every element of A to a *unique* element of B.
- "Unique" meaning that if you have two different things in A, their arrows don't collide on the B side.
  - That is, you can't do this.

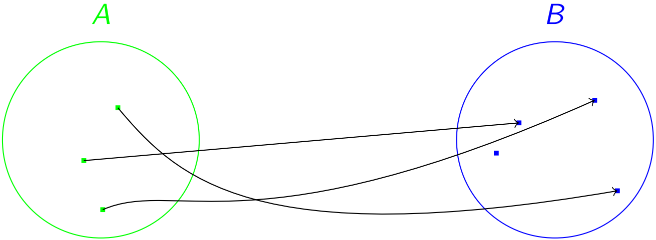
OK, so what does it mean?
(cont.)



- The map doesn't have to be a *rule* of any sort. It can just be an arbitrary bunch of arrows.

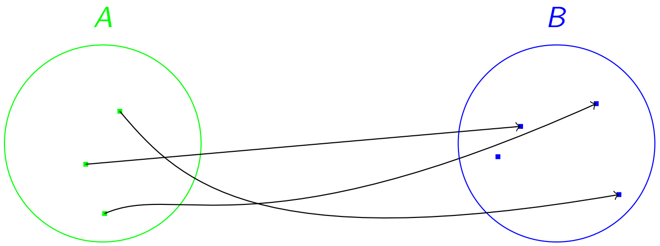
OK, so what does it mean?

(cont.)



- The map doesn't have to be a *rule* of any sort. It can just be an arbitrary bunch of arrows.
- However, we *are* sometimes interested in whether there's a "reasonably definable" rule saying where the arrows go.

OK, so what does it mean?
(cont.)



- The map doesn't have to be a *rule* of any sort. It can just be an arbitrary bunch of arrows.
- However, we *are* sometimes interested in whether there's a "reasonably definable" rule saying where the arrows go. (If we weren't, this talk would be very short.)

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$\ldots \quad \underline{1} \quad \ldots$ HEADS

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$\ldots \quad 1\underline{1} \quad \ldots$ HEADS

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

. . . $\underline{0}11$ . . . TAILS

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$\ldots$ 011$\underline{1}$ $\ldots$ HEADS

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$\ldots$ $\underline{1}0{\color{red}1}11$ $\ldots$ HEADS

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$$\ldots \qquad \mathtt{10{\color{red}1}11\underline{0}} \qquad \ldots \qquad\qquad\qquad \mathtt{TAILS}$$

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

`...     0101110     ...          TAILS`

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$\ldots \quad \texttt{0101110}\underline{\texttt{0}} \quad \ldots \qquad\qquad \texttt{TAILS}$

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$$\ldots \quad \underline{1}01011100 \quad \ldots \qquad\qquad \texttt{HEADS}$$

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$$\ldots \quad 1010\textcolor{red}{1}1100\underline{0} \quad \ldots \qquad\qquad \texttt{TAILS}$$

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$$\ldots \quad 1010\textcolor{red}{1}11000 \quad \ldots$$

What's the probability that:

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$$\ldots \quad 1010\textcolor{red}{1}11000 \quad \ldots$$

What's the probability that:

- The red digit — that is, $f(0)$ — is 0?

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$$\ldots \quad 1010\textcolor{red}{1}11000 \quad \ldots$$

What's the probability that:

- The red digit — that is, $f(0)$ — is 0? $\frac{1}{2}$

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$$\ldots \quad 1010\textcolor{red}{1}11000 \quad \ldots$$

What's the probability that:

- The red digit — that is, $f(0)$ — is 0? $\frac{1}{2}$
- The red digit and the next two are 111?

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$$\ldots \quad 1010\textcolor{red}{1}11000 \quad \ldots$$

What's the probability that:

- The red digit — that is, $f(0)$ — is 0? $\frac{1}{2}$
- The red digit and the next two are 111? $\frac{1}{8}$

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$$\ldots \quad 1010\textcolor{red}{1}11000 \quad \ldots$$

What's the probability that:

- The red digit — that is, $f(0)$ — is 0? $\frac{1}{2}$
- The red digit and the next two are 111? $\frac{1}{8}$
- At least one of the two digits *immediately after* the red one, is 0?

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$$\ldots \quad 1010\textcolor{red}{1}11000 \quad \ldots$$

What's the probability that:

- The red digit — that is, $f(0)$ — is 0? $\frac{1}{2}$
- The red digit and the next two are 111? $\frac{1}{8}$
- At least one of the two digits *immediately after* the red one, is 0? $\frac{3}{4}$

Probabilities, with origin

Suppose we make a doubly infinite string by flipping a coin. Heads, we set $f(0)$ to 1; tails, to 0. Flip again to get $f(1)$, then again for $f(-1)$, then $f(2)$, $f(-2)$, and so on:

$$\ldots \quad 1010\textcolor{red}{1}11000 \quad \ldots$$

What's the probability that:

- The red digit — that is, $f(0)$ — is 0? $\frac{1}{2}$
- The red digit and the next two are 111? $\frac{1}{8}$
- At least one of the two digits *immediately after* the red one, is 0? $\frac{3}{4}$

. . . and so on.

Probabilities, forgetting origin

Suppose we have the same coin-flipping setup, but we forget which digit is red. Now we can't ask *any* of the probability questions on the previous slide. Are there any probability questions we can ask?

Probabilities, forgetting origin

Suppose we have the same coin-flipping setup, but we forget which digit is red. Now we can't ask *any* of the probability questions on the previous slide. Are there any probability questions we can ask? Certainly. We can ask any question whose answer will *always be the same* for f and g such that $f \sim g$; questions that give the same answer if you shift the string.

Probabilities, forgetting origin

Suppose we have the same coin-flipping setup, but we forget which digit is red. Now we can't ask *any* of the probability questions on the previous slide. Are there any probability questions we can ask? Certainly. We can ask any question whose answer will *always be the same* for f and g such that $f \sim g$; questions that give the same answer if you shift the string.

What's the probability that:

Probabilities, forgetting origin

Suppose we have the same coin-flipping setup, but we forget which digit is red. Now we can't ask *any* of the probability questions on the previous slide. Are there any probability questions we can ask? Certainly. We can ask any question whose answer will *always be the same* for f and g such that $f \sim g$; questions that give the same answer if you shift the string.

What's the probability that:

- There's at least one 0 in the string?

Probabilities, forgetting origin

Suppose we have the same coin-flipping setup, but we forget which digit is red. Now we can't ask *any* of the probability questions on the previous slide. Are there any probability questions we can ask? Certainly. We can ask any question whose answer will *always be the same* for f and g such that $f \sim g$; questions that give the same answer if you shift the string.

What's the probability that:

- There's at least one 0 in the string? 1

Probabilities, forgetting origin

Suppose we have the same coin-flipping setup, but we forget which digit is red. Now we can't ask *any* of the probability questions on the previous slide. Are there any probability questions we can ask? Certainly. We can ask any question whose answer will *always be the same* for f and g such that $f \sim g$; questions that give the same answer if you shift the string.

What's the probability that:

- There's at least one 0 in the string? 1
- The average density of 0s in the string is greater than $\frac{2}{3}$?

Probabilities, forgetting origin

Suppose we have the same coin-flipping setup, but we forget which digit is red. Now we can't ask *any* of the probability questions on the previous slide. Are there any probability questions we can ask? Certainly. We can ask any question whose answer will *always be the same* for f and g such that $f \sim g$; questions that give the same answer if you shift the string.

What's the probability that:

- There's at least one 0 in the string? 1
- The average density of 0s in the string is greater than $\frac{2}{3}$? 0

Probabilities, forgetting origin

Suppose we have the same coin-flipping setup, but we forget which digit is red. Now we can't ask *any* of the probability questions on the previous slide. Are there any probability questions we can ask? Certainly. We can ask any question whose answer will *always be the same* for f and g such that $f \sim g$; questions that give the same answer if you shift the string.

What's the probability that:

- There's at least one 0 in the string? 1
- The average density of 0s in the string is greater than $\frac{2}{3}$? 0
- The entire text of *Hamlet* appears in the string?

Probabilities, forgetting origin

Suppose we have the same coin-flipping setup, but we forget which digit is red. Now we can't ask *any* of the probability questions on the previous slide. Are there any probability questions we can ask? Certainly. We can ask any question whose answer will *always be the same* for f and g such that $f \sim g$; questions that give the same answer if you shift the string.

What's the probability that:

- There's at least one 0 in the string? 1
- The average density of 0s in the string is greater than $\frac{2}{3}$? 0
- The entire text of *Hamlet* appears in the string? 1

Probabilities, forgetting origin

Suppose we have the same coin-flipping setup, but we forget which digit is red. Now we can't ask *any* of the probability questions on the previous slide. Are there any probability questions we can ask? Certainly. We can ask any question whose answer will *always be the same* for f and g such that $f \sim g$; questions that give the same answer if you shift the string.

What's the probability that:

- There's at least one 0 in the string? 1
- The average density of 0s in the string is greater than $\frac{2}{3}$? 0
- The entire text of *Hamlet* appears in the string? 1

Probabilities, forgetting origin

Suppose we have the same coin-flipping setup, but we forget which digit is red. Now we can't ask *any* of the probability questions on the previous slide. Are there any probability questions we can ask? Certainly. We can ask any question whose answer will *always be the same* for f and g such that $f \sim g$; questions that give the same answer if you shift the string.

What's the probability that:

- There's at least one 0 in the string? 1
- The average density of 0s in the string is greater than $\frac{2}{3}$? 0
- The entire text of *Hamlet* appears in the string? 1

As it turns out, for any "reasonably definable" probability question we can ask about the forgotten-origin string, the answer is always exactly 0 or exactly 1. (This is called *ergodicity*.)

How many doubly-infinite strings?

It's easy to count the number of doubly-infinite strings. There are exactly as many as there are *singly*-infinite strings; namely, $2^{\aleph_0}$. Remember that to show that the number of doubly-infinite strings is $\leq$ the number of singly-infinite strings, we just have to find a map that converts a doubly-infinite string to a *unique* singly-infinite string.

How many doubly-infinite strings?

It's easy to count the number of doubly-infinite strings. There are exactly as many as there are *singly*-infinite strings; namely, $2^{\aleph_0}$. Remember that to show that the number of doubly-infinite strings is $\leq$ the number of singly-infinite strings, we just have to find a map that converts a doubly-infinite string to a *unique* singly-infinite string.

$$\ldots \qquad \underline{1} \qquad \ldots$$

$$\underline{1} \qquad\qquad\qquad \ldots$$

How many doubly-infinite strings?

It's easy to count the number of doubly-infinite strings. There are exactly as many as there are *singly*-infinite strings; namely, $2^{\aleph_0}$. Remember that to show that the number of doubly-infinite strings is $\leq$ the number of singly-infinite strings, we just have to find a map that converts a doubly-infinite string to a *unique* singly-infinite string.

$$\begin{array}{ccc} \ldots & 1\underline{1} & \ldots \\ 1\underline{1} & & \ldots \end{array}$$

How many doubly-infinite strings?

It's easy to count the number of doubly-infinite strings. There are exactly as many as there are *singly*-infinite strings; namely, $2^{\aleph_0}$. Remember that to show that the number of doubly-infinite strings is $\leq$ the number of singly-infinite strings, we just have to find a map that converts a doubly-infinite string to a *unique* singly-infinite string.

$$\ldots \qquad \underline{0}11 \qquad \ldots$$

$$11\underline{0} \qquad\qquad \ldots$$

How many doubly-infinite strings?

It's easy to count the number of doubly-infinite strings. There are exactly as many as there are *singly*-infinite strings; namely, $2^{\aleph_0}$. Remember that to show that the number of doubly-infinite strings is $\leq$ the number of singly-infinite strings, we just have to find a map that converts a doubly-infinite string to a *unique* singly-infinite string.

$$\begin{array}{ccc} \ldots & 011\underline{1} & \ldots \\ 110\underline{1} & & \ldots \end{array}$$

How many doubly-infinite strings?

It's easy to count the number of doubly-infinite strings. There are exactly as many as there are *singly*-infinite strings; namely, $2^{\aleph_0}$. Remember that to show that the number of doubly-infinite strings is $\leq$ the number of singly-infinite strings, we just have to find a map that converts a doubly-infinite string to a *unique* singly-infinite string.

$$\ldots \quad \underline{1}0111 \quad \ldots$$

$$11011\underline{1} \quad \ldots$$

How many doubly-infinite strings?

It's easy to count the number of doubly-infinite strings. There are exactly as many as there are *singly*-infinite strings; namely, $2^{\aleph_0}$. Remember that to show that the number of doubly-infinite strings is $\leq$ the number of singly-infinite strings, we just have to find a map that converts a doubly-infinite string to a *unique* singly-infinite string.

$$\ldots \quad 10111\underline{0} \quad \ldots$$
$$11011\underline{0} \quad \ldots$$

How many doubly-infinite strings?

It's easy to count the number of doubly-infinite strings. There are exactly as many as there are *singly*-infinite strings; namely, $2^{\aleph_0}$. Remember that to show that the number of doubly-infinite strings is $\leq$ the number of singly-infinite strings, we just have to find a map that converts a doubly-infinite string to a *unique* singly-infinite string.

$$\ldots \quad \underline{0}1011 10 \quad \ldots$$
$$1101100\underline{0} \quad \ldots$$

How many doubly-infinite strings?

It's easy to count the number of doubly-infinite strings. There are exactly as many as there are *singly*-infinite strings; namely, $2^{\aleph_0}$. Remember that to show that the number of doubly-infinite strings is $\leq$ the number of singly-infinite strings, we just have to find a map that converts a doubly-infinite string to a *unique* singly-infinite string.

$$\ldots \quad 01011 10\underline{0} \quad \ldots$$
$$11011000\underline{0} \qquad \ldots$$

How many doubly-infinite strings?

It's easy to count the number of doubly-infinite strings. There are exactly as many as there are *singly*-infinite strings; namely, $2^{\aleph_0}$. Remember that to show that the number of doubly-infinite strings is $\leq$ the number of singly-infinite strings, we just have to find a map that converts a doubly-infinite string to a *unique* singly-infinite string.

$$\ldots \quad \underline{1}010\textcolor{red}{1}1100 \quad \ldots$$

$$\textcolor{red}{1}1011000\underline{1} \qquad\qquad \ldots$$

How many doubly-infinite strings?

It's easy to count the number of doubly-infinite strings. There are exactly as many as there are *singly*-infinite strings; namely, $2^{\aleph_0}$. Remember that to show that the number of doubly-infinite strings is $\leq$ the number of singly-infinite strings, we just have to find a map that converts a doubly-infinite string to a *unique* singly-infinite string.

$$\ldots \quad 1010111000\underline{0} \quad \ldots$$
$$1101100010\underline{0} \qquad\qquad \ldots$$

How many doubly-infinite strings?

It's easy to count the number of doubly-infinite strings. There are exactly as many as there are *singly*-infinite strings; namely, $2^{\aleph_0}$. Remember that to show that the number of doubly-infinite strings is $\leq$ the number of singly-infinite strings, we just have to find a map that converts a doubly-infinite string to a *unique* singly-infinite string.

$$\begin{array}{lll} \ldots & 1010\textcolor{red}{1}11000 & \ldots \\ \textcolor{red}{1}101100010 & & \ldots \end{array}$$

...but of course this requires knowing which letter of the doubly-infinite string is red.

So what if we don't know the origin?

- If we don't have origins for the doubly-infinite strings, then we can't use the trick of the previous slide to assign a unique singly-infinite string to each doubly-infinite string, because we don't know where to start. But maybe it can still be done, just in some more complicated way.

So what if we don't know the origin?

- If we don't have origins for the doubly-infinite strings, then we can't use the trick of the previous slide to assign a unique singly-infinite string to each doubly-infinite string, because we don't know where to start. But maybe it can still be done, just in some more complicated way.
- But first we need to figure out just what it means to make such an assignment (remember we're being kind of informal about just what a doubly-infinite-string-with-forgotten-origin exactly *is*).

So what if we don't know the origin?

- If we don't have origins for the doubly-infinite strings, then we can't use the trick of the previous slide to assign a unique singly-infinite string to each doubly-infinite string, because we don't know where to start. But maybe it can still be done, just in some more complicated way.
- But first we need to figure out just what it means to make such an assignment (remember we're being kind of informal about just what a doubly-infinite-string-with-forgotten-origin exactly *is*).
- The key idea is that any question you ask about such strings, should give the same answer when you shift the string.

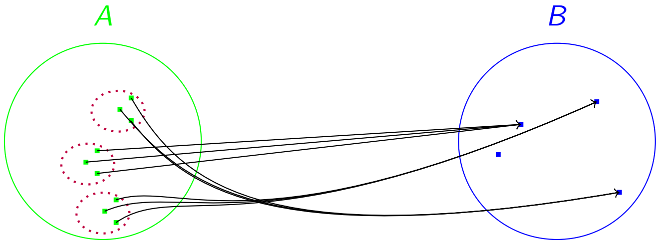
So what if we don't know the origin?

- If we don't have origins for the doubly-infinite strings, then we can't use the trick of the previous slide to assign a unique singly-infinite string to each doubly-infinite string, because we don't know where to start. But maybe it can still be done, just in some more complicated way.
- But first we need to figure out just what it means to make such an assignment (remember we're being kind of informal about just what a doubly-infinite-string-with-forgotten-origin exactly *is*).
- The key idea is that any question you ask about such strings, should give the same answer when you shift the string.
- So we want a map F that takes doubly-infinite strings f and g, and returns singly-infinite strings $F(f)$ and $F(g)$, and if f is a shift of g, then $F(f) = F(g)$.

So what if we don't know the origin?

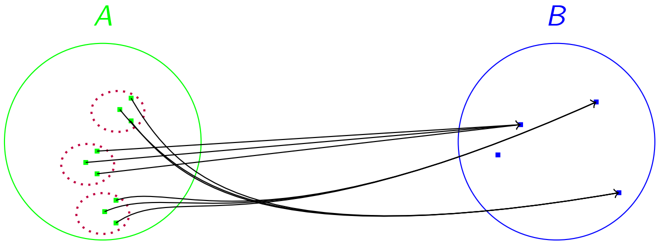
- If we don't have origins for the doubly-infinite strings, then we can't use the trick of the previous slide to assign a unique singly-infinite string to each doubly-infinite string, because we don't know where to start. But maybe it can still be done, just in some more complicated way.
- But first we need to figure out just what it means to make such an assignment (remember we're being kind of informal about just what a doubly-infinite-string-with-forgotten-origin exactly *is*).
- The key idea is that any question you ask about such strings, should give the same answer when you shift the string.
- So we want a map F that takes doubly-infinite strings f and g, and returns singly-infinite strings $F(f)$ and $F(g)$, and if f is a shift of g, then $F(f) = F(g)$.
- But if f is not a shift of g, then $F(f) \neq F(g)$ (that's the uniqueness).

Revisiting cardinality, for sets of equivalence classes



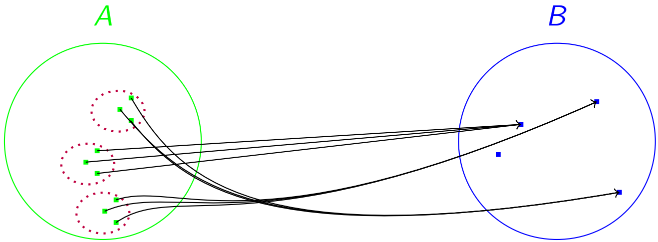
- In this picture, the number of equivalence classes (dotted ovals) in A is $\leq$ the number of points in B

Revisiting cardinality, for sets of equivalence classes



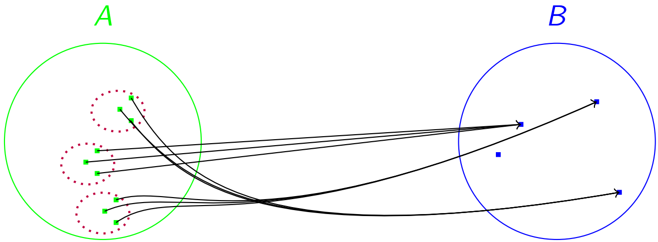
- In this picture, the number of equivalence classes (dotted ovals) in A is $\leq$ the number of points in B
- Think of the ovals as representing strings-without-origin; each green dot is a string-with-origin

Revisiting cardinality, for sets of equivalence classes



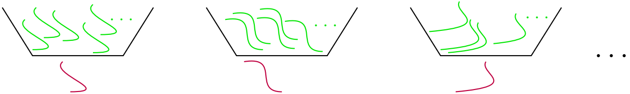
- In this picture, the number of equivalence classes (dotted ovals) in A is $\leq$ the number of points in B
- Think of the ovals as representing strings-without-origin; each green dot is a string-with-origin
- Two green dots in the *same* oval must match to the *same* blue dot

Revisiting cardinality, for sets of equivalence classes



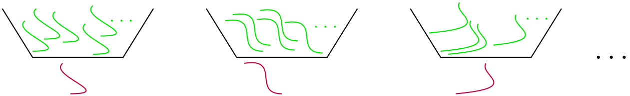
- In this picture, the number of equivalence classes (dotted ovals) in A is $\leq$ the number of points in B
- Think of the ovals as representing strings-without-origin; each green dot is a string-with-origin
- Two green dots in the *same* oval must match to the *same* blue dot
- But green dots in *different* ovals must match to different blue dots

Is there such a map?



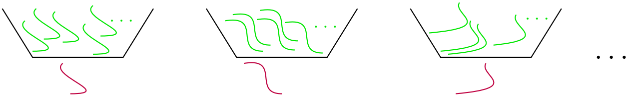
- Yes, there is.

Is there such a map?



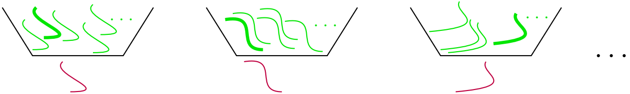
- Yes, there is.
  - Divide all the strings-with-origin into boxes — in one box, any two strings are identical except for origin

Is there such a map?



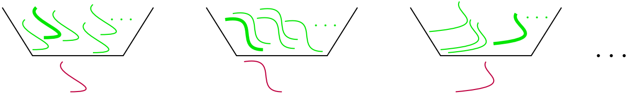
- Yes, there is.
  - Divide all the strings-with-origin into boxes — in one box, any two strings are identical except for origin
  - The *axiom of choice* says that, from each box of strings,

Is there such a map?



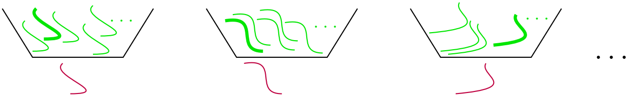
- Yes, there is.
    - Divide all the strings-with-origin into boxes — in one box, any two strings are identical except for origin
    - The *axiom of choice* says that, from each box of strings, you can single out one string

Is there such a map?



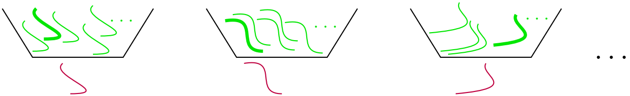
- Yes, there is.
  - Divide all the strings-with-origin into boxes — in one box, any two strings are identical except for origin
  - The *axiom of choice* says that, from each box of strings, you can single out one string
  - So the desired map will say: Given a string with the origin known, go to the distinguished string that's a shift of the one you started with

Is there such a map?



- Yes, there is.
    - Divide all the strings-with-origin into boxes — in one box, any two strings are identical except for origin
    - The *axiom of choice* says that, from each box of strings, you can single out one string
    - So the desired map will say: Given a string with the origin known, go to the distinguished string that's a shift of the one you started with
    - Then unfold that string into a singly-infinite string

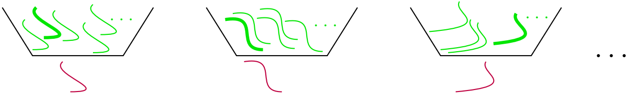
Is there such a map?



- Yes, there is.
  - Divide all the strings-with-origin into boxes — in one box, any two strings are identical except for origin
  - The *axiom of choice* says that, from each box of strings, you can single out one string
  - So the desired map will say: Given a string with the origin known, go to the distinguished string that's a shift of the one you started with
  - Then unfold that string into a singly-infinite string

However this doesn't appear to help us if we want a "reasonably definable" such map. The axiom just tells us that there *is* a way of distinguishing one string in each box; it doesn't tell us how to do it.

Is there such a map?



- Yes, there is.
  - Divide all the strings-with-origin into boxes — in one box, any two strings are identical except for origin
  - The *axiom of choice* says that, from each box of strings, you can single out one string
  - So the desired map will say: Given a string with the origin known, go to the distinguished string that's a shift of the one you started with
  - Then unfold that string into a singly-infinite string

However this doesn't appear to help us if we want a "reasonably definable" such map. The axiom just tells us that there *is* a way of distinguishing one string in each box; it doesn't tell us how to do it. Of course, it also doesn't tell us that there *isn't* a definable way to do it.

Tying things together (with strings)

So suppose we *do* have a "reasonably definable" map F that shows that there are only as many forgotten-origin doubly-infinite strings as there are singly-infinite strings. What can we find out about it? For example, we might want to know, if you put a doubly-infinite string s into F, giving a singly-infinite string $F(s)$, does $F(s)$ start with a 1? Of course, we expect the answer to depend on s....

Tying things together (with strings)

So suppose we *do* have a “reasonably definable” map F that shows that there are only as many forgotten-origin doubly-infinite strings as there are singly-infinite strings. What can we find out about it? For example, we might want to know, if you put a doubly-infinite string s into F, giving a singly-infinite string $F(s)$, does $F(s)$ start with a 1? Of course, we expect the answer to depend on s. . . .

But! That’s a “reasonably definable” question.

Tying things together (with strings)

So suppose we *do* have a "reasonably definable" map F that shows that there are only as many forgotten-origin doubly-infinite strings as there are singly-infinite strings. What can we find out about it? For example, we might want to know, if you put a doubly-infinite string s into F, giving a singly-infinite string $F(s)$, does $F(s)$ start with a 1? Of course, we expect the answer to depend on s....

But! That's a "reasonably definable" question.

So the probability that $F(s)$ starts with 1, for a *random* string s, is either exactly 0 or exactly 1.

Building a special string, based on F

So now let's look at the probability that the nth letter of $F(s)$ is 1 (for a random doubly-infinite string s)

Building a special string, based on F

So now let's look at the probability that the nth letter of $F(s)$ is 1 (for a random doubly-infinite string s)

```
Probability that              0
bit number 0 is 1:
Special string:               0       ...
```

Building a special string, based on F

So now let's look at the probability that the nth letter of $F(s)$ is 1 (for a random doubly-infinite string s)

`Probability that bit number 1 is 1:`	0	
`Special string:`	0<u>0</u>	...

Building a special string, based on F

So now let's look at the probability that the nth letter of $F(s)$ is 1 (for a random doubly-infinite string s)

`Probability that bit number 2 is 1:`	1
`Special string:`	00<u>1</u> ...

Building a special string, based on F

So now let's look at the probability that the nth letter of $F(s)$ is 1 (for a random doubly-infinite string s)

```
Probability that              0
bit number 3 is 1:
Special string:               0010    ...
```

Building a special string, based on F

So now let's look at the probability that the nth letter of $F(s)$ is 1 (for a random doubly-infinite string s)

`Probability that bit number 4 is 1:`	`1`
`Special string:`	`00101 ...`

Building a special string, based on F

So now let's look at the probability that the nth letter of $F(s)$ is 1 (for a random doubly-infinite string s)

```
Probability that              1
bit number 5 is 1:
Special string:               001011 ...
```

Building a special string, based on F

So now let's look at the probability that the nth letter of $F(s)$ is 1 (for a random doubly-infinite string s)

```
Special string:                    001011 ...
```

Building a special string, based on F

So now let's look at the probability that the nth letter of $F(s)$ is 1 (for a random doubly-infinite string s)

`Special string:                001011 ...`

What's the probability that

- F of a random string, equals the special string?

Building a special string, based on F

So now let's look at the probability that the nth letter of $F(s)$ is 1 (for a random doubly-infinite string s)

`Special string:` `001011 ...`

What's the probability that

- F of a random string, equals the special string? 1

Building a special string, based on F

So now let's look at the probability that the nth letter of $F(s)$ is 1 (for a random doubly-infinite string s)

`Special string:                001011 ...`

What's the probability that

- F of a random string, equals the special string? 1
  - Because for each n, the probability that the n^{th} bit of $F(s)$, is the n^{th} bit of the special string, is 1, by construction

Building a special string, based on F

So now let's look at the probability that the nth letter of $F(s)$ is 1 (for a random doubly-infinite string s)

`Special string:` `001011 ...`

What's the probability that

- F of a random string, equals the special string? 1
  - Because for each n, the probability that the n^{th} bit of $F(s)$, is the n^{th} bit of the special string, is 1, by construction
  - ... and there are only countably many n

Building a special string, based on F

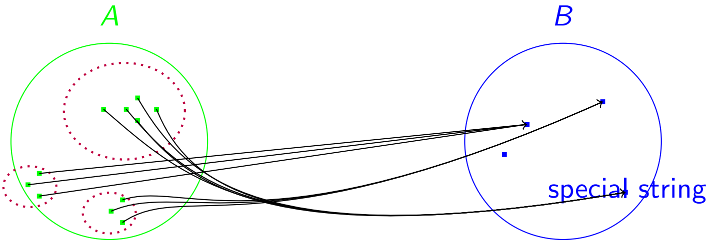
So now let's look at the probability that the nth letter of $F(s)$ is 1 (for a random doubly-infinite string s)

`Special string:` `001011 ...`

What's the probability that

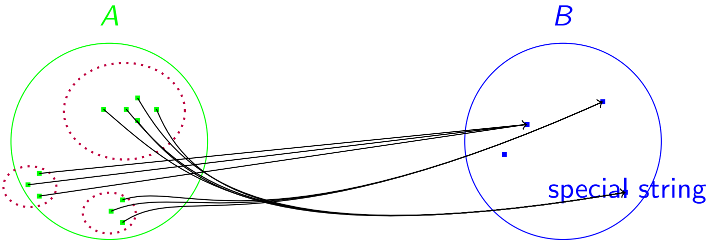
- F of a random string, equals the special string? 1
  - Because for each n, the probability that the n^{th} bit of $F(s)$, is the n^{th} bit of the special string, is 1, by construction
  - ... and there are only countably many n
  - ... and probability is countably additive

Picture of F



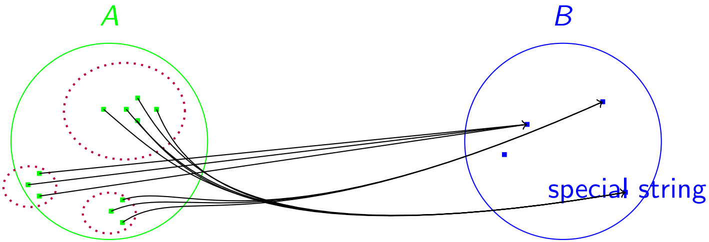
- Again, the green dots are strings-with-origin; the ovals around them gather them into strings-without-origin

Picture of F



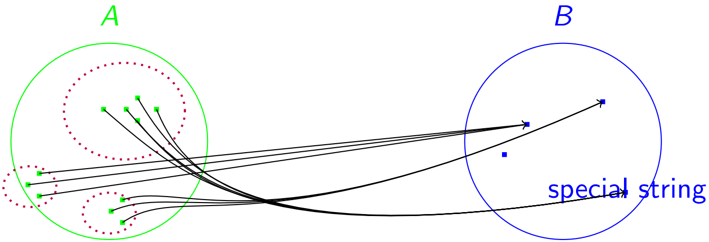
- Again, the green dots are strings-with-origin; the ovals around them gather them into strings-without-origin
- The blue dots are singly-infinite strings — one of them is the "special string"

Picture of F



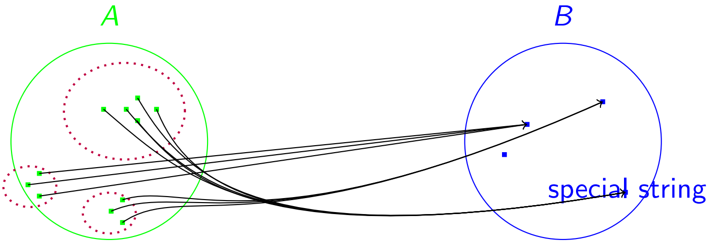
- Again, the green dots are strings-with-origin; the ovals around them gather them into strings-without-origin
- The blue dots are singly-infinite strings — one of them is the "special string"
- *Almost everything* on the left, gets mapped to the special string

Picture of F

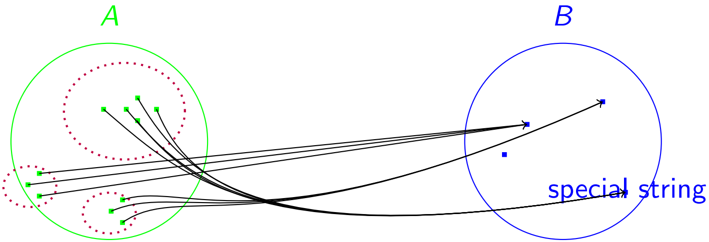


- Again, the green dots are strings-with-origin; the ovals around them gather them into strings-without-origin
- The blue dots are singly-infinite strings — one of them is the "special string"
- *Almost everything* on the left, gets mapped to the special string
- So a random green dot has probability one of being in the big oval

Is this possible?

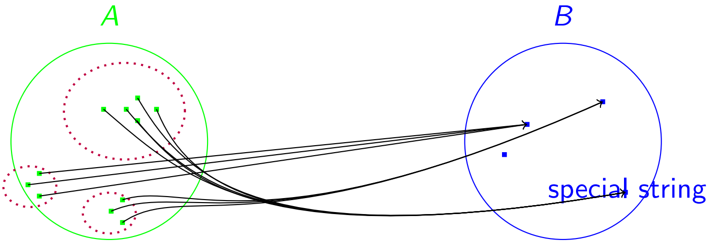


Is this possible?



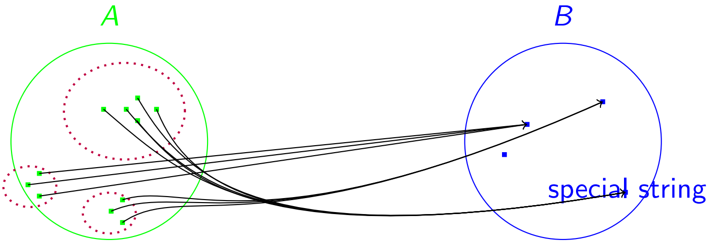
- No

Is this possible?



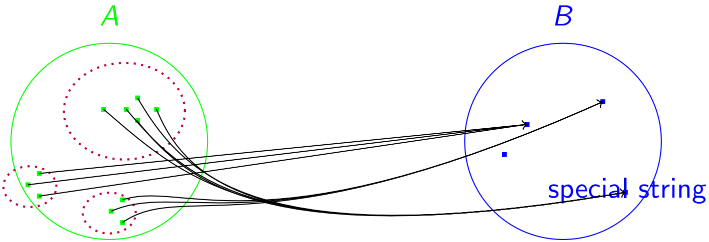
- No
- Each individual green dot has probability zero

Is this possible?



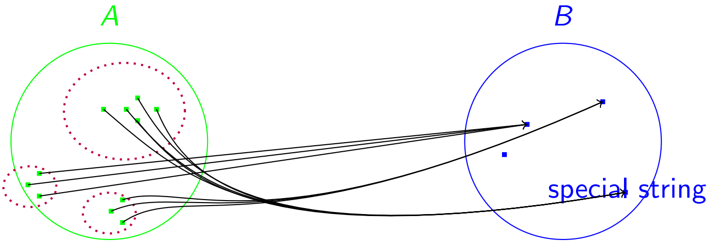
- No
- Each individual green dot has probability zero
- But there are only countably many green dots in an oval (because there are only countably many amounts by which you can shift a string)

Is this possible?



- No
- Each individual green dot has probability zero
- But there are only countably many green dots in an oval (because there are only countably many amounts by which you can shift a string)
- And probability is countably additive

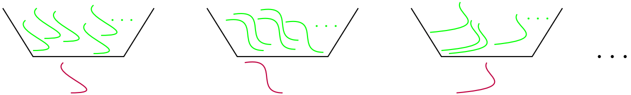
Is this possible?



- No
- Each individual green dot has probability zero
- But there are only countably many green dots in an oval (because there are only countably many amounts by which you can shift a string)
- And probability is countably additive

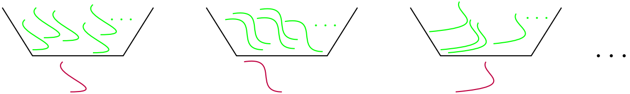
So we have a contradiction on our hands. The only way to resolve it is that no such function F exists.

I hope you appreciate just how very strange this is



This says that there are more boxes, than there are *total* things in all the boxes put together! Even though each box has infinitely many things in it.
In the picture, there are strictly more purple guys than green guys!

I hope you appreciate just how very strange this is



This says that there are more boxes, than there are *total* things in all the boxes put together! Even though each box has infinitely many things in it.

In the picture, there are strictly more purple guys than green guys! (Well, at least in this special sense, where we require definable maps.)

Wait a minute, did we exactly show *more*?

Well, no, not quite. We showed that the number of forgotten-origin strings was, in this special sense, *not less than or equal to* the number of strings with origin:

$$|\{\text{no-origin strings}\}| \not\leq_{\text{definable}} |\{\text{strings with origin}\}|$$

Wait a minute, did we exactly show *more*?

Well, no, not quite. We showed that the number of forgotten-origin strings was, in this special sense, *not less than or equal to* the number of strings with origin:

$$|\{\text{no-origin strings}\}| \not\leq_{\text{definable}} |\{\text{strings with origin}\}|$$

To see that the number is actually *greater*, we need the other direction:

$$|\{\text{strings with origin}\}| \leq_{\text{definable}} |\{\text{no-origin strings}\}|$$

Wait a minute, did we exactly show *more*?

Well, no, not quite. We showed that the number of forgotten-origin strings was, in this special sense, *not less than or equal to* the number of strings with origin:

$$|\{\text{no-origin strings}\}| \not\leq_{\text{definable}} |\{\text{strings with origin}\}|$$

To see that the number is actually *greater*, we need the other direction:

$$|\{\text{strings with origin}\}| \leq_{\text{definable}} |\{\text{no-origin strings}\}|$$

- That is, we need a definable one-to-one map F from strings-with-origin to strings-without-origin

Wait a minute, did we exactly show *more*?

Well, no, not quite. We showed that the number of forgotten-origin strings was, in this special sense, *not less than or equal to* the number of strings with origin:

$$|\{\text{no-origin strings}\}| \not\leq_{\text{definable}} |\{\text{strings with origin}\}|$$

To see that the number is actually *greater*, we need the other direction:

$$|\{\text{strings with origin}\}| \leq_{\text{definable}} |\{\text{no-origin strings}\}|$$

- That is, we need a definable one-to-one map F from strings-with-origin to strings-without-origin
- In this context, one-to-one means that if s_1 is different from s_2, then $F(s_1)$ is not only different from $F(s_2)$; it's not even a shift of $F(s_2)$

How do we get that map?

I'm actually going to let you think about that one. It's a really good problem to check your understanding. But here's a couple of hints:

How do we get that map?

I'm actually going to let you think about that one. It's a really good problem to check your understanding. But here's a couple of hints:

- What you need to do is encode a singly-infinite string into a doubly-infinite one, in such a way that if someone shifts the doubly-infinite string (and doesn't tell you by how much), you can still decode it

How do we get that map?

I'm actually going to let you think about that one. It's a really good problem to check your understanding. But here's a couple of hints:

- What you need to do is encode a singly-infinite string into a doubly-infinite one, in such a way that if someone shifts the doubly-infinite string (and doesn't tell you by how much), you can still decode it
- So just to get started, can you think of a way to code the *first* bit of the singly-infinite string? Maybe, if it's zero, you make a doubly-infinite string that has lots of zeroes, and if it's a one, you make a doubly-infinite string that has lots of ones? Can you generalize?

How do we get that map?

I'm actually going to let you think about that one. It's a really good problem to check your understanding. But here's a couple of hints:

- What you need to do is encode a singly-infinite string into a doubly-infinite one, in such a way that if someone shifts the doubly-infinite string (and doesn't tell you by how much), you can still decode it
- So just to get started, can you think of a way to code the *first* bit of the singly-infinite string? Maybe, if it's zero, you make a doubly-infinite string that has lots of zeroes, and if it's a one, you make a doubly-infinite string that has lots of ones? Can you generalize?
- Alternatively, you might first code the singly-infinite string as a real number, then try to encode the real number into the doubly-infinite string, in such a way that you can decode it after any shift

Opportunities for research

- The example I have shown today is a small sample of a very active research field, usually called the study of *Borel equivalence relations*.

Opportunities for research

- The example I have shown today is a small sample of a very active research field, usually called the study of *Borel equivalence relations*.
- If you are interested in mathematical logic, this is a newer subfield of set theory than many of the traditional ones, and may have more accessible open problems.

Partial chart of equivalence relations, ordered by definable cardinality of quotient