The Last Lecture

A Brief Encounter with the Uncomputable

Lecture 26

Question

Question

- Which of the following are countably infinite?
  1. Set of all prime numbers
  2. Set of all bit strings of length 32
  3. Set of all bit strings of finite length
  4. Set of all infinitely long bit strings

  A. 1, 2, 3 and 4
  B. 1, 2 and 3
  C. 1 and 4
  D. 1 and 3
  E. None of the above choices

The Uncountable

The Uncountable

- S uncountable if no one-to-one $f:S \to \mathbb{N}$ (equiv'ly, no onto $f:\mathbb{N} \to S$)

The Uncountable

- S uncountable if no one-to-one $f:S \to \mathbb{N}$ (equiv'ly, no onto $f:\mathbb{N} \to S$)
- <u>Claim</u>: $\mathbb{R}$ is uncountable

The Uncountable

- S uncountable if no one-to-one $f:S \to \mathbb{N}$ (equiv'ly, no onto $f:\mathbb{N} \to S$)
- <u>Claim</u>: $\mathbb{R}$ is uncountable
- Related claims:

The Uncountable

- S uncountable if no one-to-one $f:S \to \mathbb{N}$ (equiv'ly, no onto $f:\mathbb{N} \to S$)
- Claim: $\mathbb{R}$ is uncountable
- Related claims:
  - Set $\mathbb{T}$ of all infinitely long binary strings is uncountable

The Uncountable

- S uncountable if no one-to-one $f:S \to \mathbb{N}$ (equiv'ly, no onto $f:\mathbb{N} \to S$)
- Claim: $\mathbb{R}$ is uncountable
- Related claims:
  - Set $\mathbb{T}$ of all infinitely long binary strings is uncountable
    - Contrast with set of all finitely long binary strings, which is a countably infinite set

The Uncountable

- S uncountable if no one-to-one $f:S \to \mathbb{N}$ (equiv'ly, no onto $f:\mathbb{N} \to S$)
- Claim: $\mathbb{R}$ is uncountable
- Related claims:
  - Set $\mathbb{T}$ of all infinitely long binary strings is uncountable
    - Contrast with set of all finitely long binary strings, which is a countably infinite set
  - The power-set of $\mathbb{N}$, $\mathbb{P}(\mathbb{N})$ is uncountable

The Uncountable

- S uncountable if no one-to-one $f:S \to \mathbb{N}$ (equiv'ly, no onto $f:\mathbb{N} \to S$)
- Claim: $\mathbb{R}$ is uncountable
- Related claims:
  - Set $\mathbb{T}$ of all infinitely long binary strings is uncountable
    - Contrast with set of all finitely long binary strings, which is a countably infinite set
  - The power-set of $\mathbb{N}$, $\mathbb{P}(\mathbb{N})$ is uncountable
    - There is a bijection $f: \mathbb{T} \to \mathbb{P}(\mathbb{N})$ defined as $f(s) = \{ i \mid s_i = 1 \}$

The Uncountable

- S uncountable if no one-to-one $f:S \to \mathbb{N}$ (equiv'ly, no onto $f:\mathbb{N} \to S$)
- Claim: $\mathbb{R}$ is uncountable
- Related claims:
  - Set $\mathbb{T}$ of all infinitely long binary strings is uncountable
    - Contrast with set of all finitely long binary strings, which is a countably infinite set
  - The power-set of $\mathbb{N}$, $\mathbb{P}(\mathbb{N})$ is uncountable
    - There is a bijection $f: \mathbb{T} \to \mathbb{P}(\mathbb{N})$ defined as $f(s) = \{ i \mid s_i = 1 \}$

e.g., set of even numbers in $\mathbb{N}$ corresponds to the string 101010...

The Uncountable

- S uncountable if no one-to-one $f: S \to \mathbb{N}$ (equiv'ly, no onto $f: \mathbb{N} \to S$)
- Claim: $\mathbb{R}$ is uncountable
- Related claims:
  - Set $\mathbb{T}$ of all infinitely long binary strings is uncountable
    - Contrast with set of all finitely long binary strings, which is a countably infinite set
  - The power-set of $\mathbb{N}$, $\mathbb{P}(\mathbb{N})$ is uncountable
    - There is a bijection $f: \mathbb{T} \to \mathbb{P}(\mathbb{N})$ defined as $f(s) = \{ i \mid s_i = 1 \}$
- How do we show something is not countable?!

e.g., set of even numbers in $\mathbb{N}$ corresponds to the string 101010...

The Uncountable

- S uncountable if no one-to-one $f:S \to \mathbb{N}$ (equiv'ly, no onto $f:\mathbb{N} \to S$)
- Claim: $\mathbb{R}$ is uncountable
- Related claims:
  - Set $\mathbb{T}$ of all infinitely long binary strings is uncountable
    - Contrast with set of all finitely long binary strings, which is a countably infinite set
  - The power-set of $\mathbb{N}$, $\mathbb{P}(\mathbb{N})$ is uncountable
    - There is a bijection $f: \mathbb{T} \to \mathbb{P}(\mathbb{N})$ defined as $f(s) = \{ i \mid s_i = 1 \}$
- How do we show something is not countable?!
- Cantor's "diagonal slash"

e.g., set of even numbers in $\mathbb{N}$ corresponds to the string 101010...

Cantor's Diagonal Slash

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$

f(0) =	1	0	0	1	0	0	0
f(1) =	0	0	1	0	1	0	0
f(2) =	1	1	1	1	1	1	1
f(3) =	1	1	0	1	0	1	0
f(4) =	1	1	0	0	0	0	1
f(5) =	0	1	0	1	1	0	1
f(6) =	0	1	0	1	0	1	0

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$
- Consider the set $X \subseteq \mathbb{N}$ corresponding to the "flipped diagonal"

f(0) =	1	0	0	1	0	0	0
f(1) =	0	0	1	0	1	0	0
f(2) =	1	1	1	1	1	1	1
f(3) =	1	1	0	1	0	1	0
f(4) =	1	1	0	0	0	0	1
f(5) =	0	1	0	1	1	0	1
f(6) =	0	1	0	1	0	1	0

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$
- Consider the set $X \subseteq \mathbb{N}$ corresponding to the "flipped diagonal"

f(0) =	1	0	0	1	0	0	0
f(1) =	0	0	1	0	1	0	0
f(2) =	1	1	1	1	1	1	1
f(3) =	1	1	0	1	0	1	0
f(4) =	1	1	0	0	0	0	1
f(5) =	0	1	0	1	1	0	1
f(6) =	0	1	0	1	0	1	0

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$
- Consider the set $X \subseteq \mathbb{N}$ corresponding to the "flipped diagonal"
  - $X = \{ j \mid T_{jj} = 0 \}$
    $= \{ j \mid j \notin f(j) \}$

f(0) =	1	0	0	1	0	0	0
f(1) =	0	0	1	0	1	0	0
f(2) =	1	1	1	1	1	1	1
f(3) =	1	1	0	1	0	1	0
f(4) =	1	1	0	0	0	0	1
f(5) =	0	1	0	1	1	0	1
f(6) =	0	1	0	1	0	1	0

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$
- Consider the set $X \subseteq \mathbb{N}$ corresponding to the "flipped diagonal"
  - $X = \{ j \mid T_{jj} = 0 \}$
    $= \{ j \mid j \notin f(j) \}$
- X doesn't appear as a row in this table (why?)

f(0) =	1	0	0	1	0	0	0
f(1) =	0	0	1	0	1	0	0
f(2) =	1	1	1	1	1	1	1
f(3) =	1	1	0	1	0	1	0
f(4) =	1	1	0	0	0	0	1
f(5) =	0	1	0	1	1	0	1
f(6) =	0	1	0	1	0	1	0

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$
- Consider the set $X \subseteq \mathbb{N}$ corresponding to the "flipped diagonal"
  - $X = \{ j \mid T_{jj} = 0 \}$
    $= \{ j \mid j \notin f(j) \}$
- X doesn't appear as a row in this table (why?)
  - So f not onto

f(0) =	1	0	0	1	0	0	0
f(1) =	0	0	1	0	1	0	0
f(2) =	1	1	1	1	1	1	1
f(3) =	1	1	0	1	0	1	0
f(4) =	1	1	0	0	0	0	1
f(5) =	0	1	0	1	1	0	1
f(6) =	0	1	0	1	0	1	0

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$
- Consider the set $X \subseteq \mathbb{N}$ corresponding to the "flipped diagonal"
  - $X = \{ j \mid T_{jj} = 0 \}$
    $= \{ j \mid j \notin f(j) \}$
- X doesn't appear as a row in this table (why?)
  - So f not onto

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$
- Consider the set $X \subseteq \mathbb{N}$ corresponding to the "flipped diagonal"
  - $X = \{ j \mid T_{jj} = 0 \}$
    $= \{ j \mid j \notin f(j) \}$
- X doesn't appear as a row in this table (why?)
  - So f not onto

Generalizes:
No onto function $f: A \to \mathbb{P}(A)$
for any set A

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$
- Consider the set $X \subseteq \mathbb{N}$ corresponding to the "flipped diagonal"
  - $X = \{ j \mid T_{jj} = 0 \}$
    $= \{ j \mid j \notin f(j) \}$
- X doesn't appear as a row in this table (why?)
  - So f not onto

Generalizes:
No onto function $f: A \to \mathbb{P}(A)$
for any set A

May not have a table enumerating f

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$
- Consider the set $X \subseteq \mathbb{N}$ corresponding to the "flipped diagonal"
  - $X = \{ j \mid T_{jj} = 0 \}$
    $= \{ j \mid j \notin f(j) \}$
- X doesn't appear as a row in this table (why?)
  - So f not onto

Generalizes:
No onto function $f: A \to \mathbb{P}(A)$
for any set A

May not have a table enumerating f

Let $X = \{ j \mid j \notin f(j) \}$

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$
- Consider the set $X \subseteq \mathbb{N}$ corresponding to the "flipped diagonal"
  - $X = \{ j \mid T_{jj} = 0 \}$
    $= \{ j \mid j \notin f(j) \}$
- X doesn't appear as a row in this table (why?)
  - So f not onto

Generalizes:
No onto function $f: A \to \mathbb{P}(A)$
for any set A

May not have a table enumerating f

Let $X = \{ j \mid j \notin f(j) \}$

Claim: $\nexists\, i \in A$ s.t. $X = f(i)$

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$
- Consider the set $X \subseteq \mathbb{N}$ corresponding to the "flipped diagonal"
  - $X = \{ j \mid T_{jj} = 0 \}$
    $= \{ j \mid j \notin f(j) \}$
- X doesn't appear as a row in this table (why?)
  - So f not onto

Generalizes:
No onto function $f: A \to \mathbb{P}(A)$ for any set A

May not have a table enumerating f

Let $X = \{ j \mid j \notin f(j) \}$

Claim: $\nexists\, i \in A$ s.t. $X = f(i)$

Suppose not: i.e., $\exists i$, $X = f(i)$.

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$
- Consider the set $X \subseteq \mathbb{N}$ corresponding to the "flipped diagonal"
  - $X = \{ j \mid T_{jj} = 0 \}$
    $= \{ j \mid j \notin f(j) \}$
- X doesn't appear as a row in this table (why?)
  - So f not onto

Generalizes:
No onto function $f: A \to \mathbb{P}(A)$
for any set A

May not have a table enumerating f

Let $X = \{ j \mid j \notin f(j) \}$

Claim: $\nexists\, i \in A$ s.t. $X = f(i)$

Suppose not: i.e., $\exists i$, $X = f(i)$.
$i \in X \leftrightarrow i \in f(i) \leftrightarrow i \notin X$

Cantor's Diagonal Slash

- Take any function $f: \mathbb{N} \to \mathbb{P}(\mathbb{N})$
- Make a binary table with $T_{ij} = 1$ iff $j \in f(i)$
- Consider the set $X \subseteq \mathbb{N}$ corresponding to the "flipped diagonal"
  - $X = \{ j \mid T_{jj} = 0 \}$
    $= \{ j \mid j \notin f(j) \}$
- X doesn't appear as a row in this table (why?)
  - So f not onto

Generalizes:
No onto function $f: A \to \mathbb{P}(A)$
for any set A

May not have a table enumerating f

Let $X = \{ j \mid j \notin f(j) \}$

Claim: $\nexists\, i \in A$ s.t. $X = f(i)$

Suppose not: i.e., $\exists i$, $X = f(i)$.
$i \in X \leftrightarrow i \in f(i) \leftrightarrow i \notin X$

Contradiction!

Question

Question

- Pick the correct statement. A is a non-empty set.

A. There is no one-to-one function from A to $\mathbb{P}(A)$
B. There is no onto function from $\mathbb{P}(A)$ to A
C. There is no one-to-one function from $\mathbb{P}(A)$ to A
D. There is a bijection between A and $\mathbb{P}(A)$ iff A is finite
E. None of the above

Question

- Pick the correct statement. A is a non-empty set.

A. There is no one-to-one function from A to $\mathbb{P}(A)$
B. There is no onto function from $\mathbb{P}(A)$ to A
C. There is no one-to-one function from $\mathbb{P}(A)$ to A
D. There is a bijection between A and $\mathbb{P}(A)$ iff A is finite
E. None of the above

There is an onto function from A to B iff there is a one-to-one function from B to A

Paradoxes and Relatives

Paradoxes and Relatives

- Russell's Paradox: In the universe of all sets, let $S = \{ s \mid s \notin s \}$. Then $S \in S \leftrightarrow S \notin S$!!

Paradoxes and Relatives

- Russell's Paradox: In the universe of all sets, let $S = \{ s \mid s \notin s \}$. Then $S \in S \leftrightarrow S \notin S$!!
  - "Naïve Set Theory" is inconsistent. Consistent theories developed which do not let one define such sets.

Paradoxes and Relatives

- Russell's Paradox: In the universe of all sets, let $S = \{ s \mid s \notin s \}$. Then $S \in S \leftrightarrow S \notin S$!!
  - "Naïve Set Theory" is inconsistent. Consistent theories developed which do not let one define such sets.
- In a library of catalogs, can you have a catalog of all catalogs in the library that don't list themselves? (answer: No!)

Paradoxes and Relatives

- Russell's Paradox: In the universe of all sets, let $S = \{ s \mid s \notin s \}$. Then $S \in S \leftrightarrow S \notin S$!!
  - "Naïve Set Theory" is inconsistent. Consistent theories developed which do not let one define such sets.
- In a library of catalogs, can you have a catalog of all catalogs in the library that don't list themselves? (answer: No!)
- Liar's paradox: "This statement is false." (The statement is true iff it is false! Requires logic with "unknown/undefined" as truth value.)

Paradoxes and Relatives

- Russell's Paradox: In the universe of all sets, let $S = \{ s \mid s \notin s \}$. Then $S \in S \leftrightarrow S \notin S$!!
  - "Naïve Set Theory" is inconsistent. Consistent theories developed which do not let one define such sets.
- In a library of catalogs, can you have a catalog of all catalogs in the library that don't list themselves? (answer: No!)
- Liar's paradox: "This statement is false." (The statement is true iff it is false! Requires logic with "unknown/undefined" as truth value.)
- Gödel numbered statements in a theory and showed that in any "rich" theory there must be a statement with number g which says "statement with Gödel number g is <u>not provable</u>"

Paradoxes and Relatives

- Russell's Paradox: In the universe of all sets, let $S = \{ s \mid s \notin s \}$. Then $S \in S \leftrightarrow S \notin S$!!
  - "Naïve Set Theory" is inconsistent. Consistent theories developed which do not let one define such sets.
- In a library of catalogs, can you have a catalog of all catalogs in the library that don't list themselves? (answer: No!)
- Liar's paradox: "This statement is false." (The statement is true iff it is false! Requires logic with "unknown/undefined" as truth value.)
- Gödel numbered statements in a theory and showed that in any "rich" theory there must be a statement with number g which says "statement with Gödel number g is <u>not provable</u>"
  - This statement must be true if theory <u>consistent</u> (else a false statement is provable). Then the theory would be <u>incomplete</u>.

Reals are Uncountable

Reals are Uncountable

- Enough to show a <u>one-to-one</u> mapping from $\mathbb{I}$, the set of infinite binary strings to the set of real numbers (why?)

Reals are Uncountable

- Enough to show a <u>one-to-one</u> mapping from $\mathbb{T}$, the set of infinite binary strings to the set of real numbers (why?)
- Idea: treat a binary string $s_1 s_2 s_3 \ldots$ as the real number $0.s_1 s_2 s_3 \ldots$ <u>in decimal</u>

Reals are Uncountable

- Enough to show a <u>one-to-one</u> mapping from $\mathbb{T}$, the set of infinite binary strings to the set of real numbers (why?)
- Idea: treat a binary string $s_1 s_2 s_3 \ldots$ as the real number $0.s_1 s_2 s_3 \ldots$ <u>in decimal</u>
  - This is a one-to-one mapping: a finite difference between the real numbers that two different strings map to

Reals are Uncountable

- Enough to show a <u>one-to-one</u> mapping from $\mathbb{T}$, the set of infinite binary strings to the set of real numbers (why?)
- Idea: treat a binary string $s_1s_2s_3\ldots$ as the real number $0.s_1s_2s_3\ldots$ <u>in decimal</u>
  - This is a one-to-one mapping: a finite difference between the real numbers that two different strings map to
  - Note: if used binary representation instead of decimal representation, we'll have strings 011111.. and 10000... map to the same real number (though that can be handled)

Reals are Uncountable

- Enough to show a <u>one-to-one</u> mapping from $\mathbb{T}$, the set of infinite binary strings to the set of real numbers (why?)
- Idea: treat a binary string $s_1s_2s_3\ldots$ as the real number $0.s_1s_2s_3\ldots$ <u>in decimal</u>
  - This is a one-to-one mapping: a finite difference between the real numbers that two different strings map to
  - Note: if used binary representation instead of decimal representation, we'll have strings 011111.. and 10000... map to the same real number (though that can be handled)
- On the other hand $|\mathbb{R}^2| = |\mathbb{R}|$.

Reals are Uncountable

- Enough to show a <u>one-to-one</u> mapping from $\mathbb{T}$, the set of infinite binary strings to the set of real numbers (why?)
- Idea: treat a binary string $s_1s_2s_3\ldots$ as the real number $0.s_1s_2s_3\ldots$ <u>in decimal</u>
  - This is a one-to-one mapping: a finite difference between the real numbers that two different strings map to
  - Note: if used binary representation instead of decimal representation, we'll have strings 011111.. and 10000... map to the same real number (though that can be handled)
- On the other hand $|\mathbb{R}^2| = |\mathbb{R}|$.
  - Because $|\mathbb{T}^2|=|\mathbb{T}|$ (bijection by interleaving), and we saw $|\mathbb{R}|=|\mathbb{T}|$ (and hence $|\mathbb{R}^2|=|\mathbb{T}^2|$ too)

The Uncomputable

The Uncomputable

- We saw:

The Uncomputable

- We saw:
  - The set of functions from $\mathbb{N}$ to $\mathbb{N}$ (or even from $\mathbb{N}$ to $\{0,1\}$) is uncountable

The Uncomputable

- We saw:
  - The set of functions from $\mathbb{N}$ to $\mathbb{N}$ (or even from $\mathbb{N}$ to $\{0,1\}$) is uncountable
  - The set of finite length strings is countably infinite

The Uncomputable

- We saw:
  - The set of functions from $\mathbb{N}$ to $\mathbb{N}$ (or even from $\mathbb{N}$ to $\{0,1\}$) is uncountable
  - The set of finite length strings is countably infinite
- So we cannot assign a different string for each function

The Uncomputable

- We saw:
  - The set of functions from $\mathbb{N}$ to $\mathbb{N}$ (or even from $\mathbb{N}$ to $\{0,1\}$) is uncountable
  - The set of finite length strings is countably infinite
- So we cannot assign a different string for each function
- We cannot have a different <u>program</u> to compute each function

The Uncomputable

- We saw:
  - The set of functions from $\mathbb{N}$ to $\mathbb{N}$ (or even from $\mathbb{N}$ to $\{0,1\}$) is uncountable
  - The set of finite length strings is countably infinite
- So we cannot assign a different string for each function
- We cannot have a different <u>program</u> to compute each function
- "Almost all" functions are <u>uncomputable</u>!

The Uncomputable

The Uncomputable

- "Almost all" functions are <u>uncomputable</u>!

The Uncomputable

- "Almost all" functions are <u>uncomputable</u>!
  - Later (CS 373): distinction between computing a predicate ("deciding") and "recognizing" when the predicate is true

The Uncomputable

- "Almost all" functions are uncomputable!
  - Later (CS 373): distinction between computing a predicate ("deciding") and "recognizing" when the predicate is true
  - Almost all predicates "unrecognizable" too

The Uncomputable

- "Almost all" functions are uncomputable!
  - Later (CS 373): distinction between computing a predicate ("deciding") and "recognizing" when the predicate is true
  - Almost all predicates "unrecognizable" too
- But do we care about all these functions? Often yes!

The Uncomputable

- "Almost all" functions are uncomputable!
  - Later (CS 373): distinction between computing a predicate ("deciding") and "recognizing" when the predicate is true
  - Almost all predicates "unrecognizable" too
- But do we care about all these functions? Often yes!
  - Hilbert's 10th problem: find an algorithm to check if a "Diophantine equation" has a solution (i.e., check if there is an integer solution to all the variables in a polynomial. e.g., the ones in Fermat's last theorem, $x^3+y^3=z^3$, $x^4+y^4=z^4$, ...)

The Uncomputable

- "Almost all" functions are uncomputable!
  - Later (CS 373): distinction between computing a predicate ("deciding") and "recognizing" when the predicate is true
  - Almost all predicates "unrecognizable" too
- But do we care about all these functions? Often yes!
  - Hilbert's 10th problem: find an algorithm to check if a "Diophantine equation" has a solution (i.e., check if there is an integer solution to all the variables in a polynomial. e.g., the ones in Fermat's last theorem, $x^3+y^3=z^3$, $x^4+y^4=z^4$, ...)

shown uncomputable in 1970

The Uncomputable

- "Almost all" functions are uncomputable!
  - Later (CS 373): distinction between computing a predicate ("deciding") and "recognizing" when the predicate is true
  - Almost all predicates "unrecognizable" too
- But do we care about all these functions? Often yes!
  - Hilbert's 10th problem: find an algorithm to check if a "Diophantine equation" has a solution (i.e., check if there is an integer solution to all the variables in a polynomial. e.g., the ones in Fermat's last theorem, $x^3+y^3=z^3$, $x^4+y^4=z^4$, ...)
  - Hilbert's Entscheidungsproblem: given a statement in first order logic, check if it is true/provable

shown uncomputable in 1970

The Uncomputable

- "Almost all" functions are uncomputable!
  - Later (CS 373): distinction between computing a predicate ("deciding") and "recognizing" when the predicate is true
  - Almost all predicates "unrecognizable" too
- But do we care about all these functions? Often yes!
  - Hilbert's 10th problem: find an algorithm to check if a "Diophantine equation" has a solution (i.e., check if there is an integer solution to all the variables in a polynomial. e.g., the ones in Fermat's last theorem, $x^3+y^3=z^3$, $x^4+y^4=z^4$, ...)

    shown uncomputable in 1970
  - Hilbert's Entscheidungsproblem: given a statement in first order logic, check if it is true/provable

    shown uncomputabe by Church and Turing [1936]

The Uncomputable

- "Almost all" functions are uncomputable!
  - Later (CS 373): distinction between computing a predicate ("deciding") and "recognizing" when the predicate is true
  - Almost all predicates "unrecognizable" too
- But do we care about all these functions? Often yes!
  - Hilbert's 10th problem: find an algorithm to check if a "Diophantine equation" has a solution (i.e., check if there is an integer solution to all the variables in a polynomial. e.g., the ones in Fermat's last theorem, $x^3+y^3=z^3$, $x^4+y^4=z^4$, ...)
  - Hilbert's Entscheidungsproblem: given a statement in first order logic, check if it is true/provable
  - The Halting Problem: Given a (program,input) pair decide if the program halts or not

shown uncomputable in 1970

shown uncomputabe by Church and Turing [1936]

The Uncomputable

- "Almost all" functions are uncomputable!
  - Later (CS 373): distinction between computing a predicate ("deciding") and "recognizing" when the predicate is true
  - Almost all predicates "unrecognizable" too
- But do we care about all these functions? Often yes!
  - Hilbert's 10th problem: find an algorithm to check if a "Diophantine equation" has a solution (i.e., check if there is an integer solution to all the variables in a polynomial. e.g., the ones in Fermat's last theorem, $x^3+y^3=z^3$, $x^4+y^4=z^4$, ...)

    shown uncomputable in 1970
  - Hilbert's Entscheidungsproblem: given a statement in first order logic, check if it is true/provable

    shown uncomputabe by Church and Turing [1936]
  - The Halting Problem: Given a (program,input) pair decide if the program halts or not

    Turing [1936]

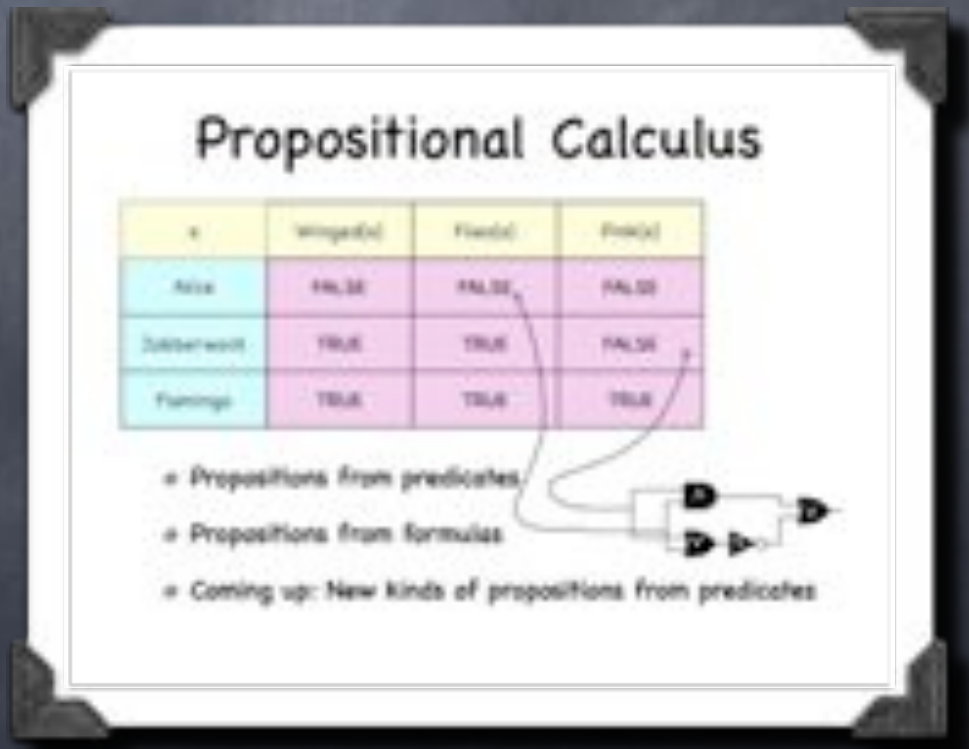
Discrete Structures

Wrap Up!

Discrete Structures (Intro to Mind Bending)

Wrap Up!

We Learned a Lot! :-)

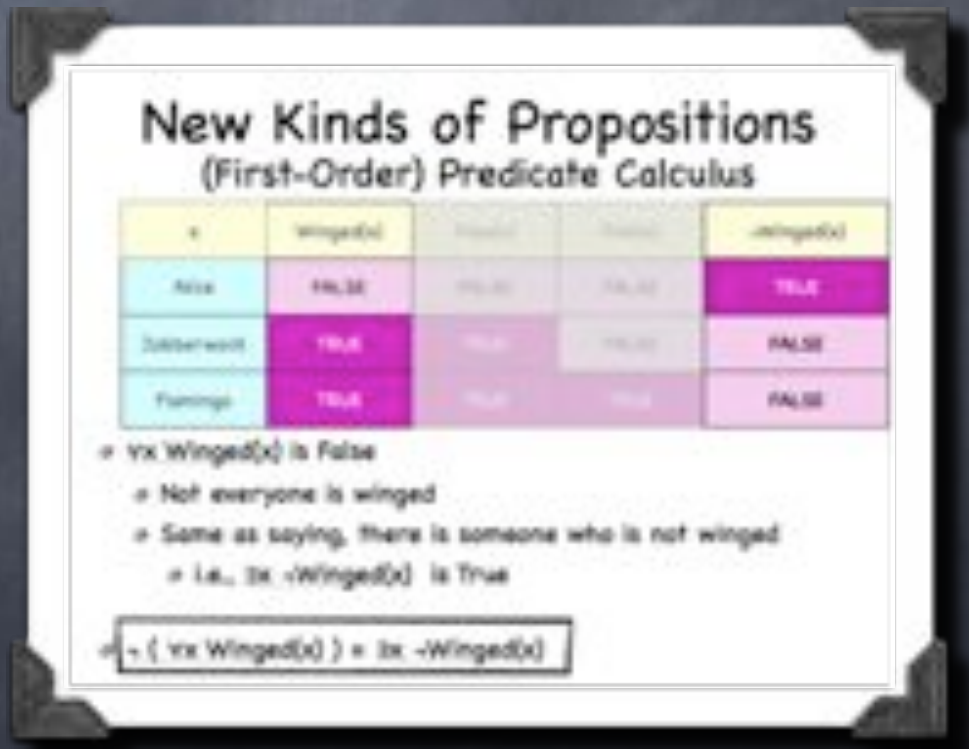


<u>Basic tools for expressing ideas</u>

Logic, Proofs,

Sets, Relations, Functions

We Learned a Lot! :-)

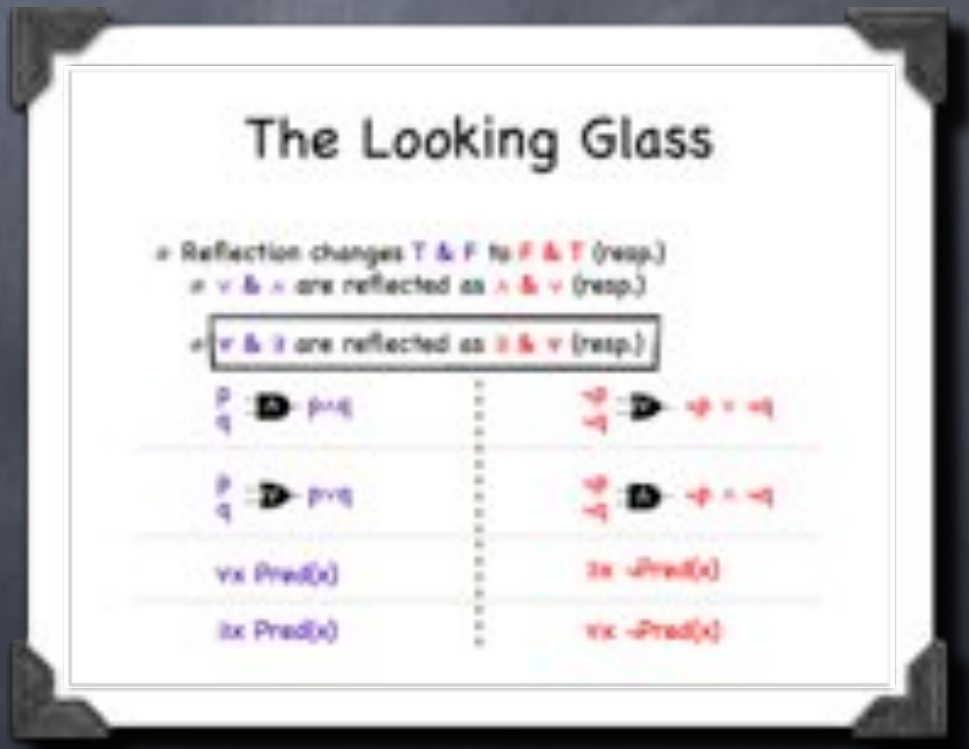


<u>Basic tools for expressing ideas</u>

Logic, Proofs,

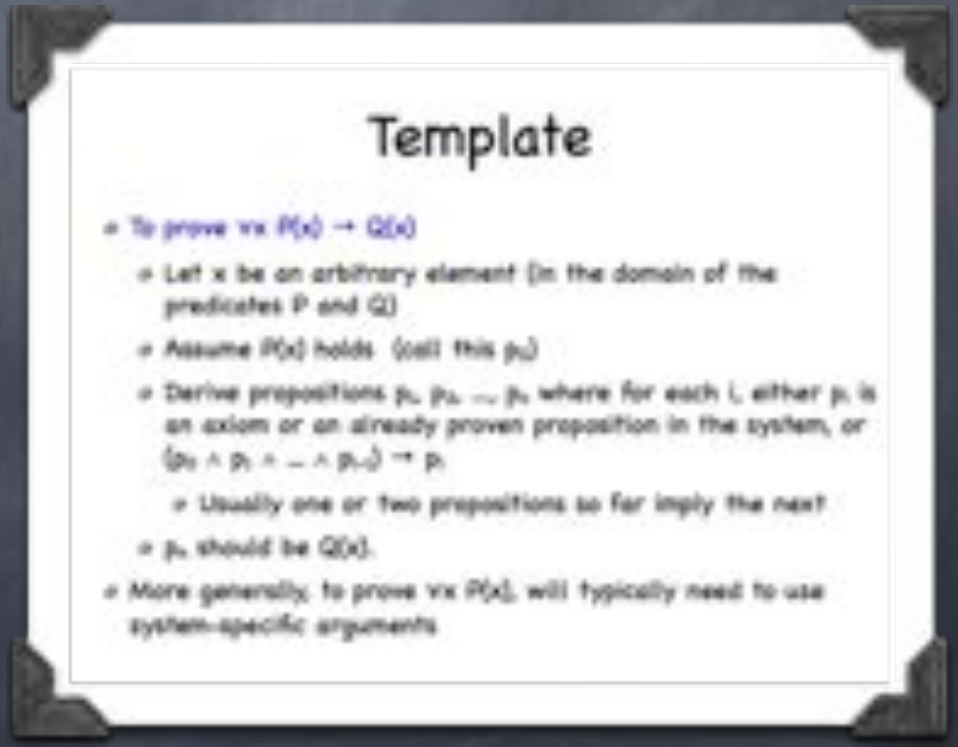
Sets, Relations, Functions

We Learned a Lot! :-)



<u>Basic tools for expressing ideas</u>
Logic, Proofs,
Sets, Relations, Functions

We Learned a Lot! :-)



Template

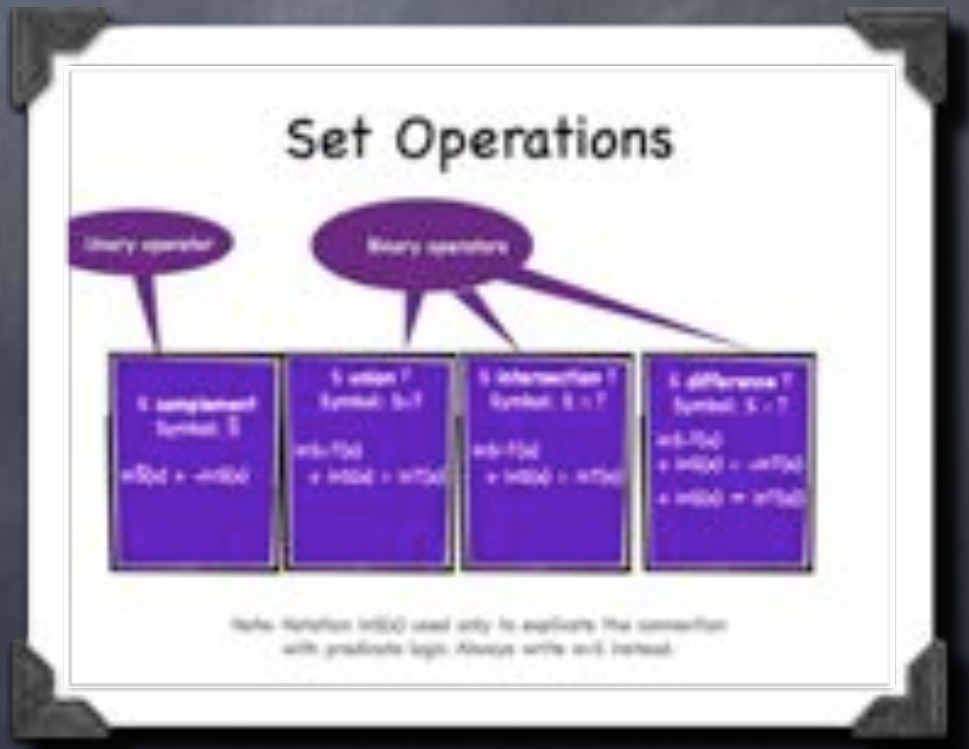
- To prove $\forall x\ P(x) \rightarrow Q(x)$
  - Let x be an arbitrary element (in the domain of the predicates P and Q)
  - Assume P(x) holds (call this p_0)
  - Derive propositions $p_0, p_1, \ldots, p_n$ where for each i, either p_i is an axiom or an already proven proposition in the system, or $(p_0 \wedge p_1 \wedge \ldots \wedge p_{i-1}) \rightarrow p_i$
    - Usually one or two propositions so far imply the next
  - p_n should be Q(x).
- More generally, to prove $\forall x\ P(x)$, will typically need to use system-specific arguments

<u>Basic tools for expressing ideas</u>

Logic, Proofs,

Sets, Relations, Functions

We Learned a Lot! :-)

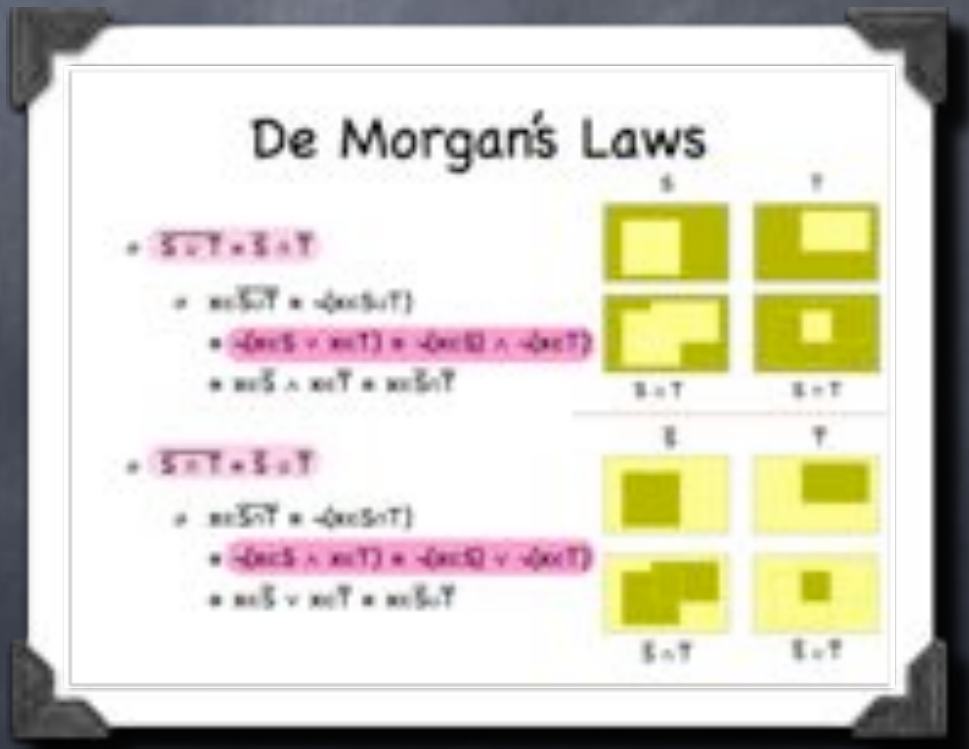


<u>Basic tools for expressing ideas</u>

Logic, Proofs,

Sets, Relations, Functions

We Learned a Lot! :-)

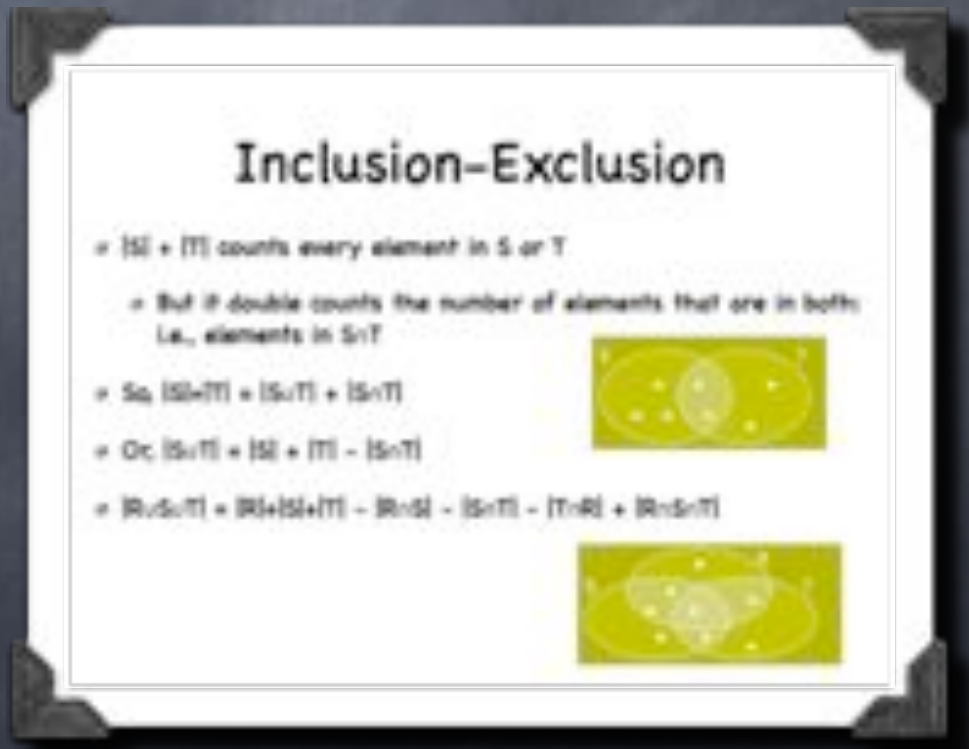


<u>Basic tools for expressing ideas</u>

Logic, Proofs,

Sets, Relations, Functions

We Learned a Lot! :-)

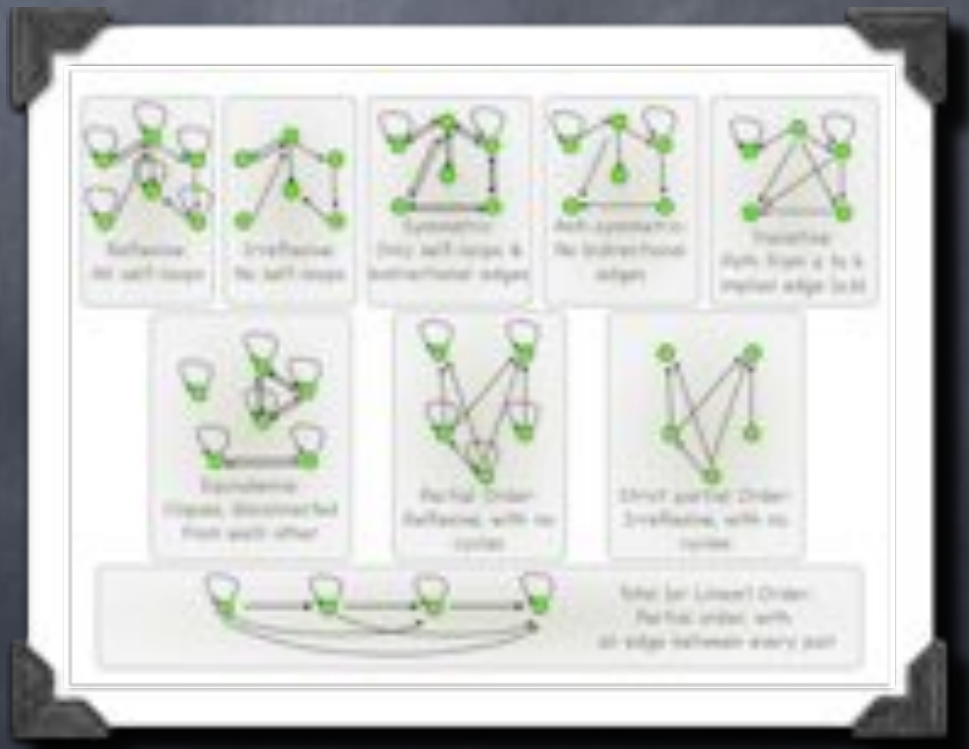


<u>Basic tools for expressing ideas</u>

Logic, Proofs,

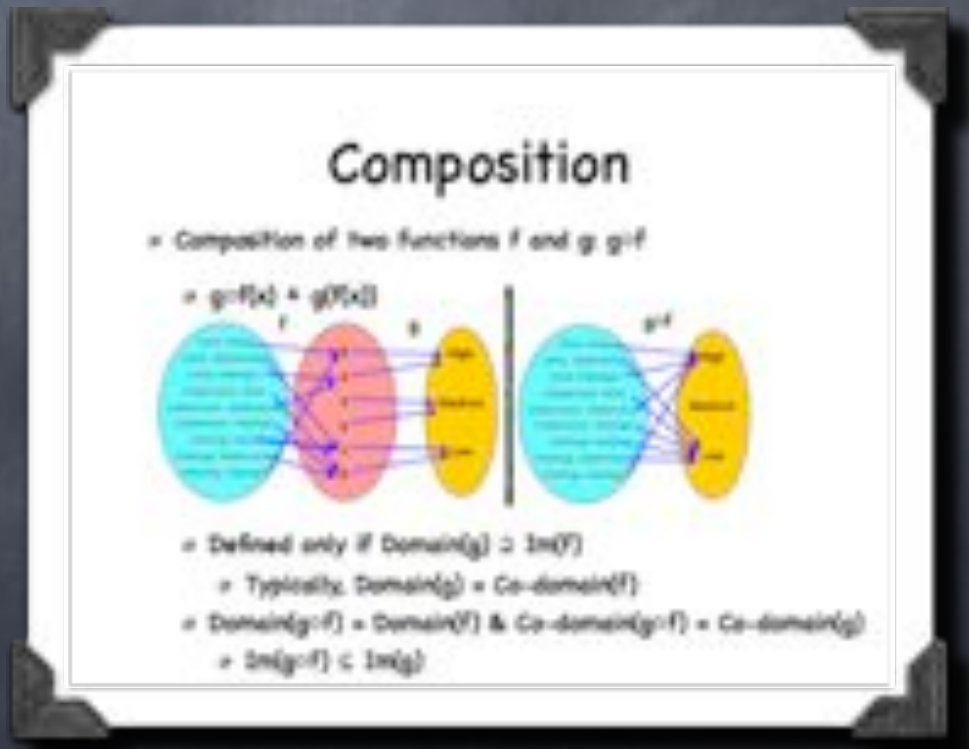
Sets, Relations, Functions

We Learned a Lot! :-)



<u>Basic tools for expressing ideas</u>
Logic, Proofs,
Sets, Relations, Functions

We Learned a Lot! :-)

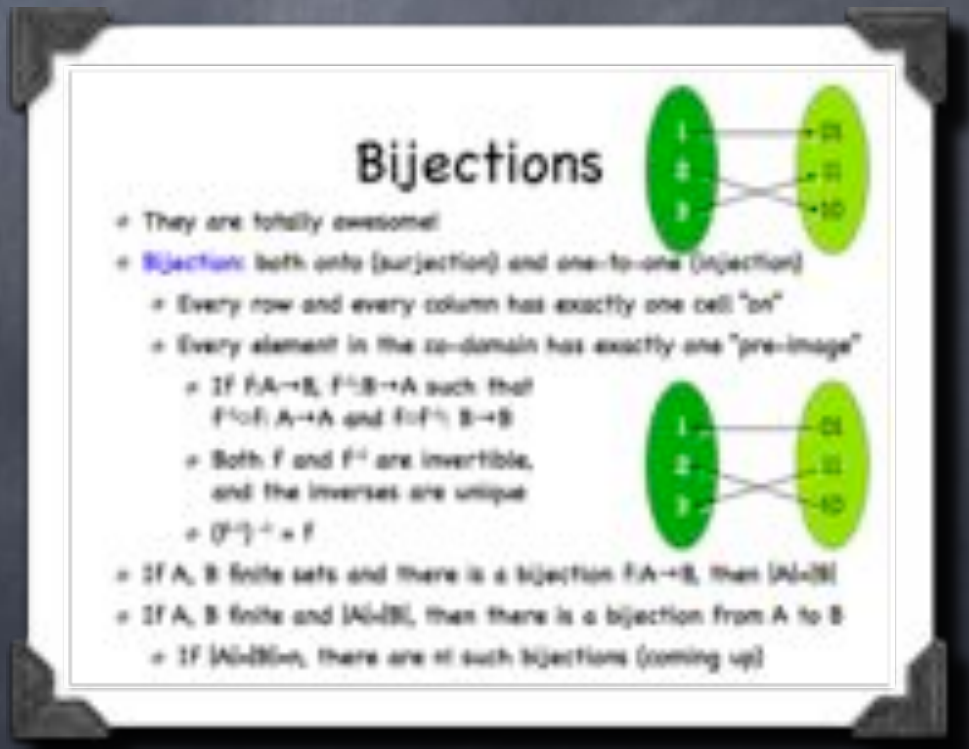


<u>Basic tools for expressing ideas</u>

Logic, Proofs,

Sets, Relations, Functions

We Learned a Lot! :-)



Basic tools for expressing ideas

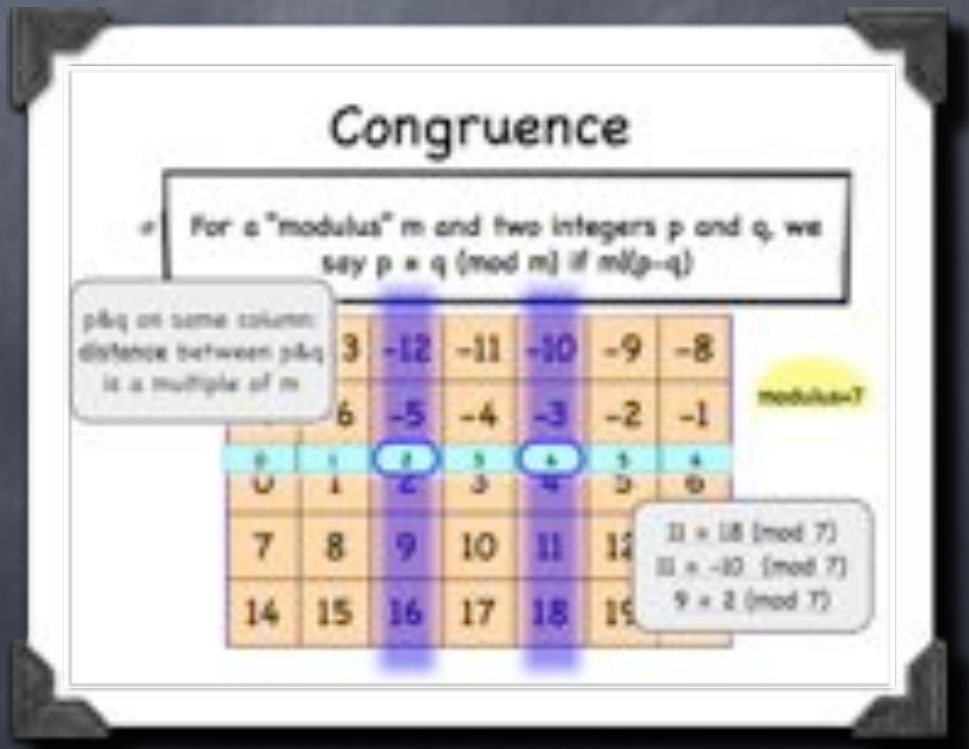
Logic, Proofs,

Sets, Relations, Functions

We Learned a Lot! :-)

Numbers

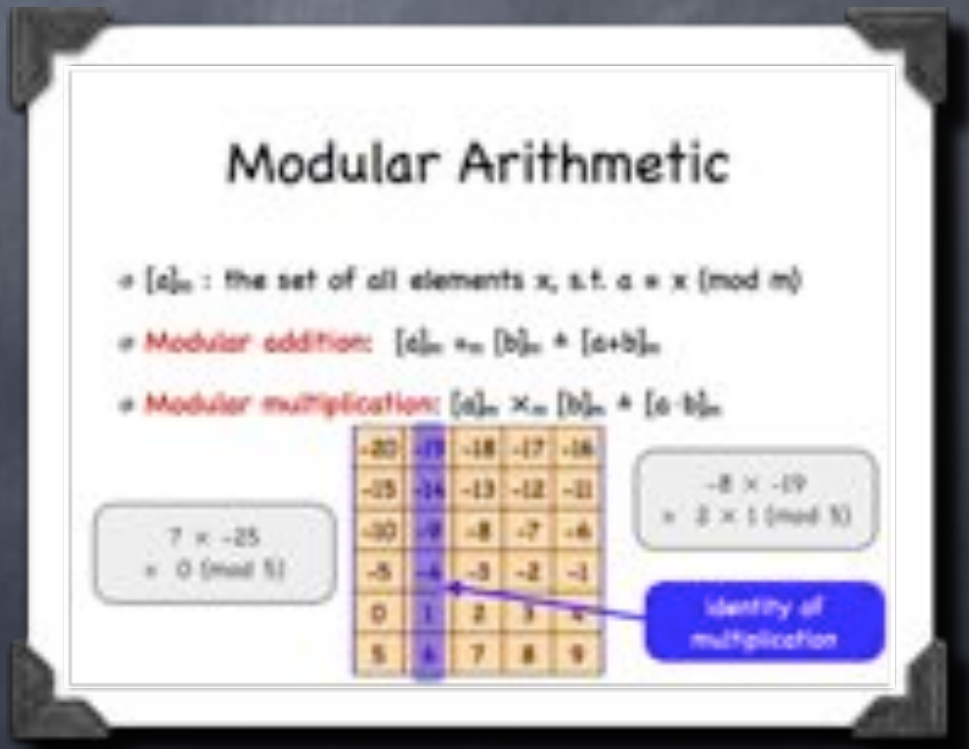
<u>Basic tools for expressing ideas</u>
Logic, Proofs,
Sets, Relations, Functions



We Learned a Lot! :-)

Numbers

<u>Basic tools for expressing ideas</u>
Logic, Proofs,
Sets, Relations, Functions

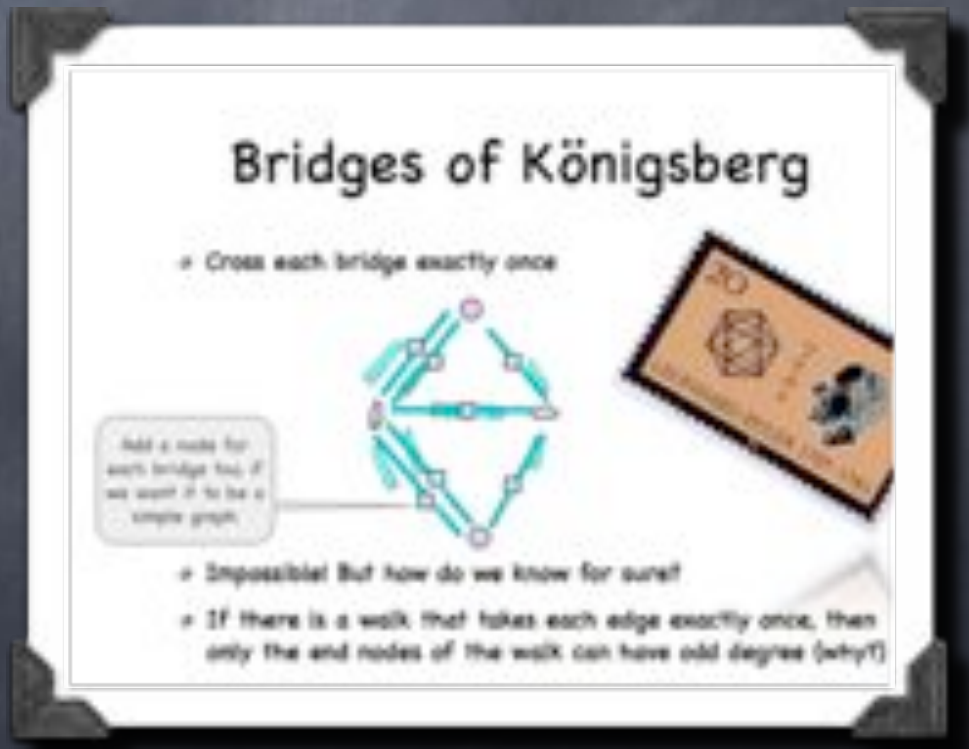


We Learned a Lot! :-)

Numbers

Graphs

<u>Basic tools for expressing ideas</u>
Logic, Proofs,
Sets, Relations, Functions

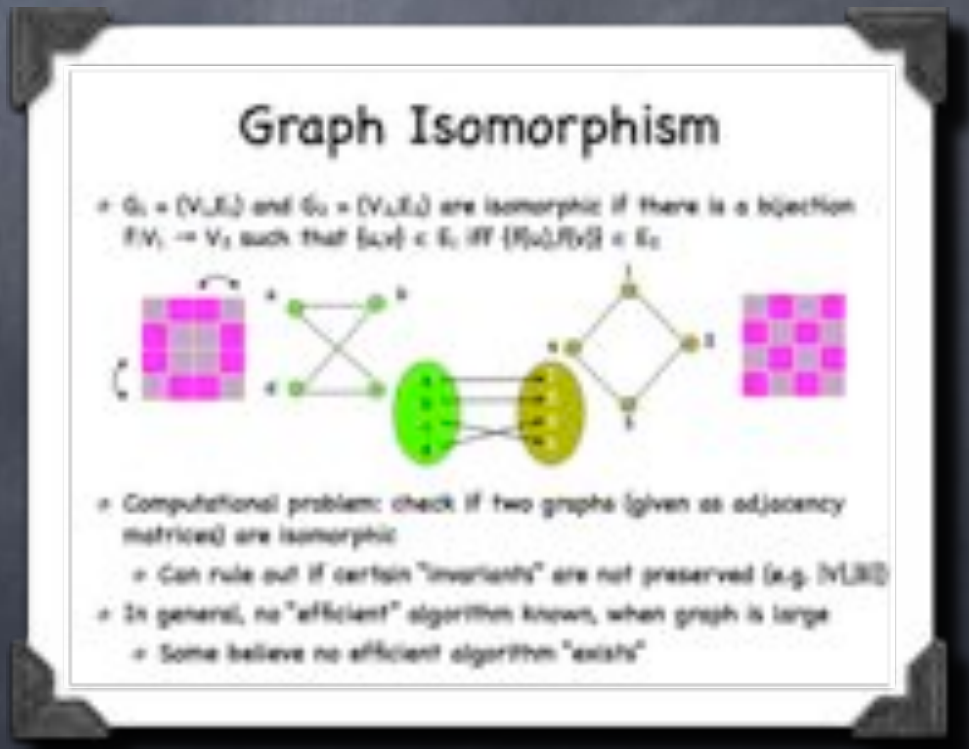


We Learned a Lot! :-)

Numbers

Graphs

<u>Basic tools for expressing ideas</u>
Logic, Proofs,
Sets, Relations, Functions

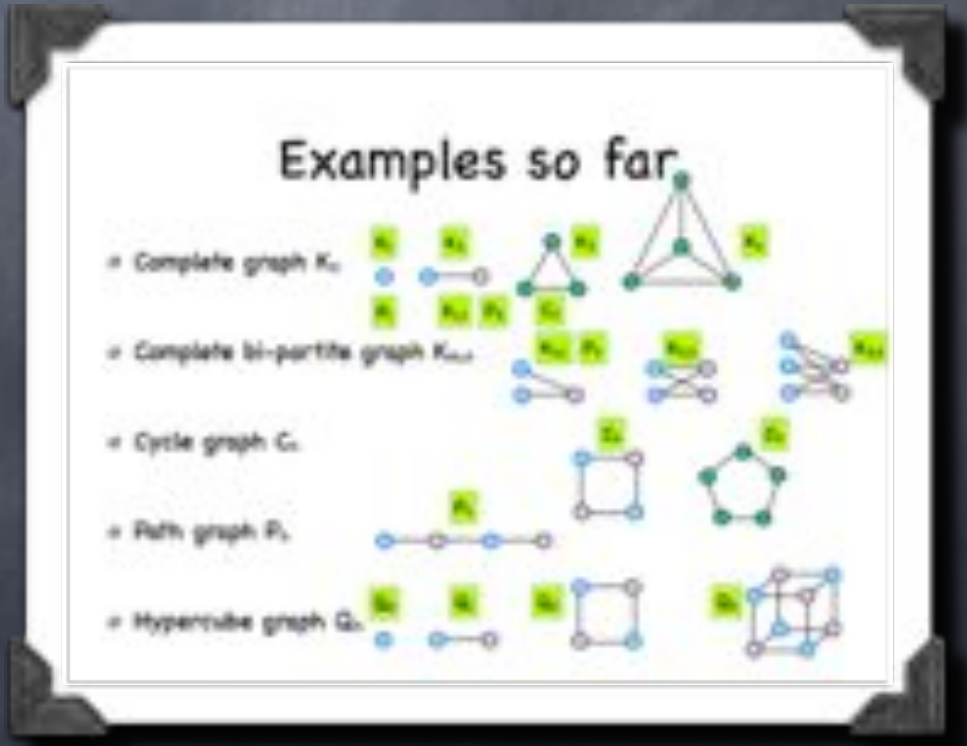


We Learned a Lot! :-)

Numbers

Graphs

<u>Basic tools for expressing ideas</u>
Logic, Proofs,
Sets, Relations, Functions



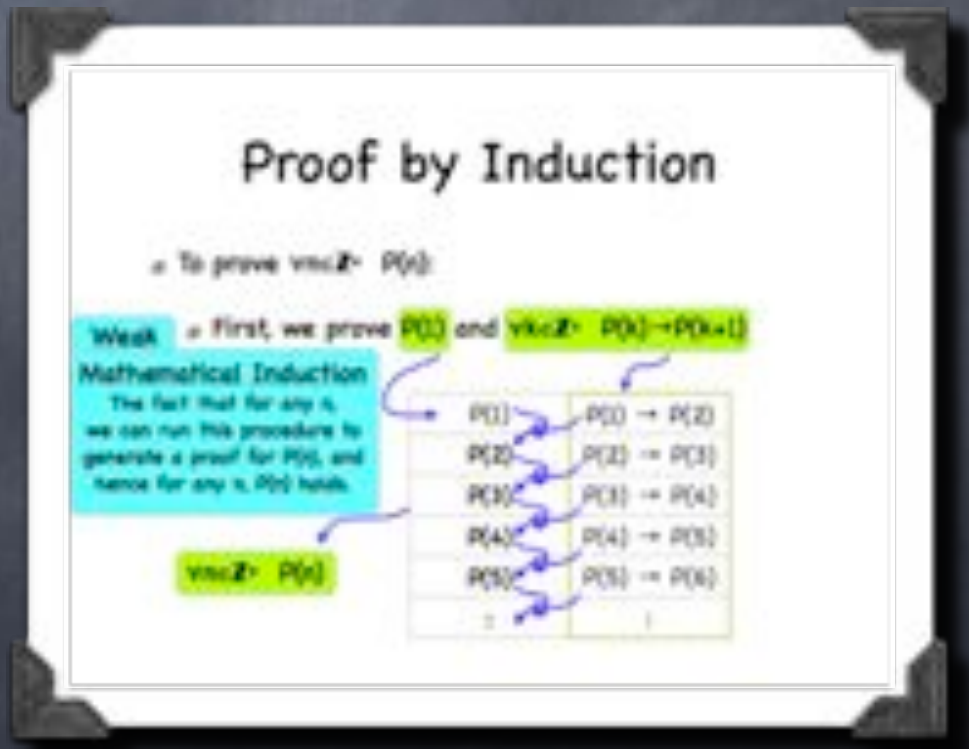
We Learned a Lot! :-)

Induction

Numbers

Graphs

<u>Basic tools for expressing ideas</u>
Logic, Proofs,
Sets, Relations, Functions



We Learned a Lot! :-)

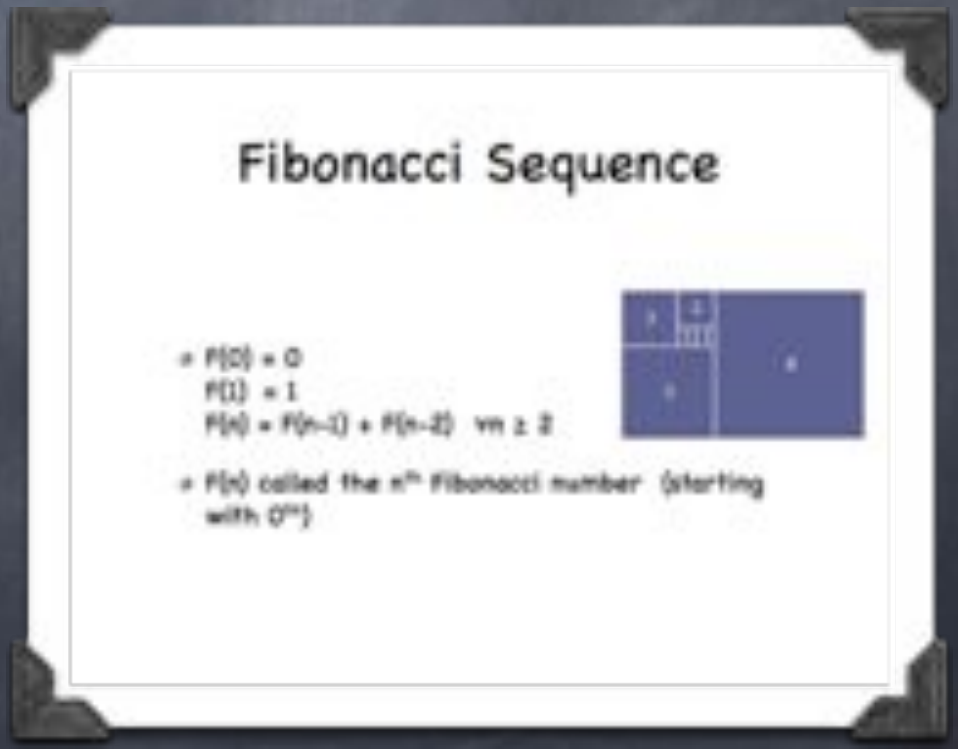
Recursion

Induction

Numbers

Graphs

<u>Basic tools for expressing ideas</u>
Logic, Proofs,
Sets, Relations, Functions



We Learned a Lot! :-)

Recursion

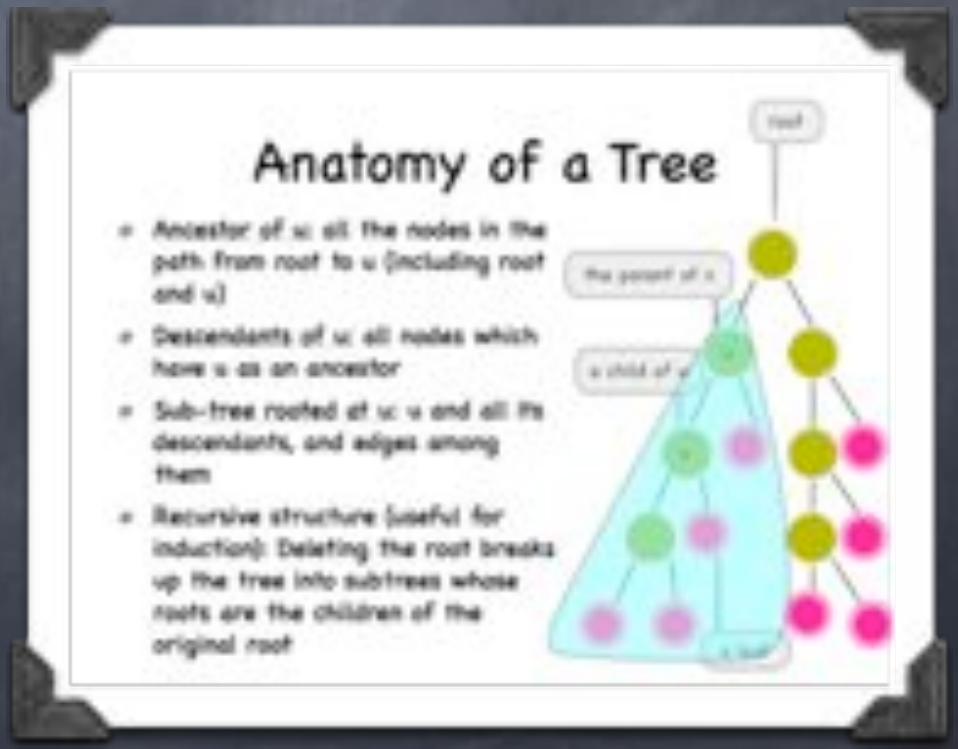
Induction

Trees

Numbers

Graphs

<u>Basic tools for expressing ideas</u>
Logic, Proofs,
Sets, Relations, Functions



We Learned a Lot! :-)

Recursion

Trees

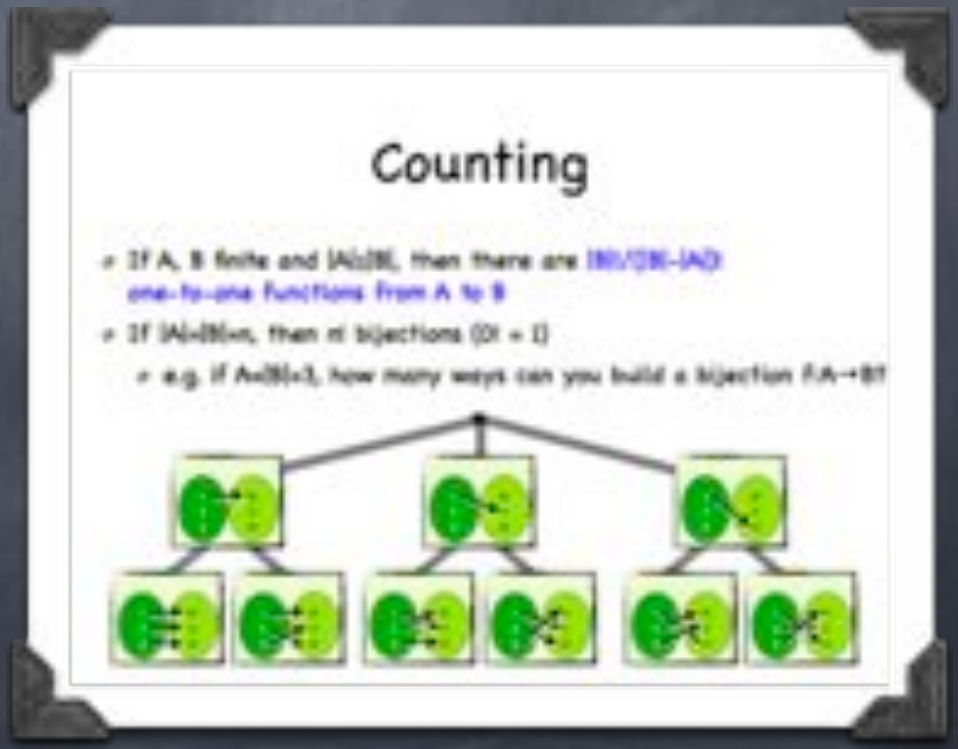
Induction

Counting

Numbers

Graphs

<u>Basic tools for expressing ideas</u>
Logic, Proofs,
Sets, Relations, Functions



We Learned a Lot! :-)

Recursion

Trees

Bounding big-O

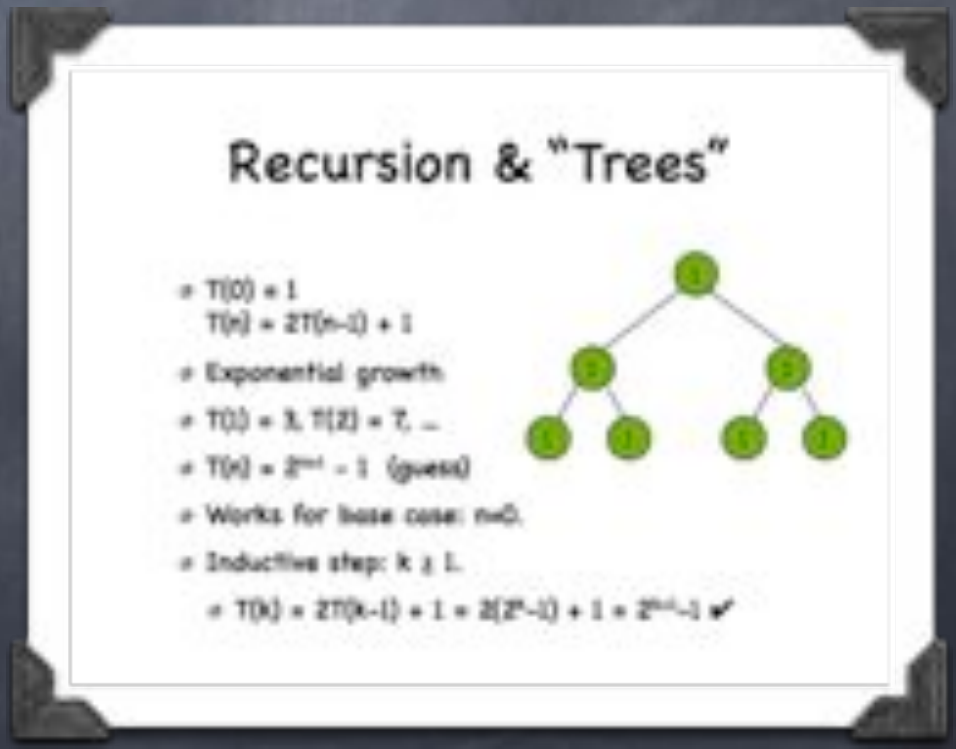
Induction

Counting

Numbers

Graphs

Basic tools for expressing ideas
Logic, Proofs,
Sets, Relations, Functions



We Learned a Lot! :-)

Recursion

Trees

Bounding
big-O

Induction

Counting

Computation
Algorithms, Circuits, Grammars,
Finite State Machines

Numbers

Graphs

Basic tools for expressing ideas
Logic, Proofs,
Sets, Relations, Functions

We Learned a Lot! :-)

Recursion

Trees

Bounding big-O

Induction

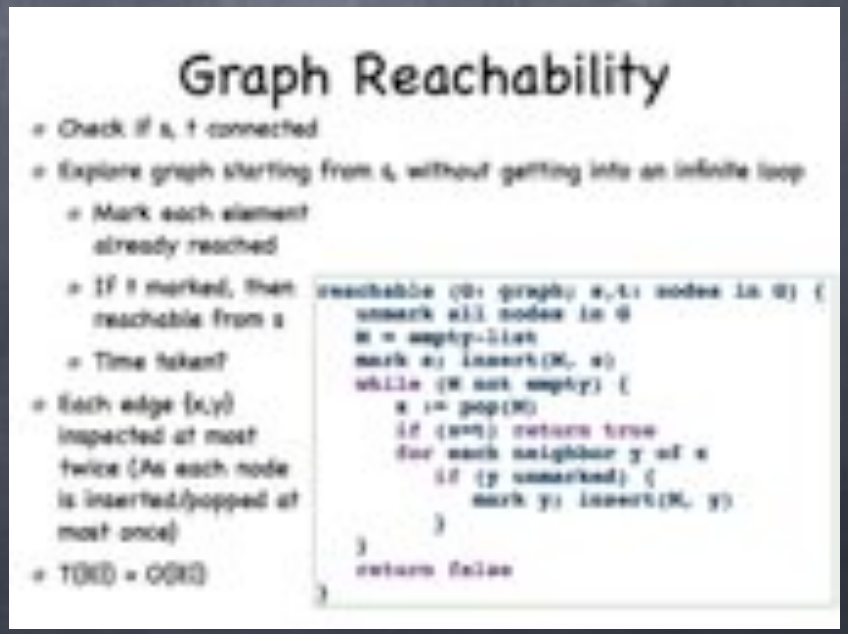
Counting

Computation
Algorithms, Circuits, Grammars, Finite State Machines

Numbers

Graphs

Basic tools for expressing ideas
Logic, Proofs,
Sets, Relations, Functions



We Learned a Lot! :-)

Recursion

Trees

Bounding
big-O

Induction

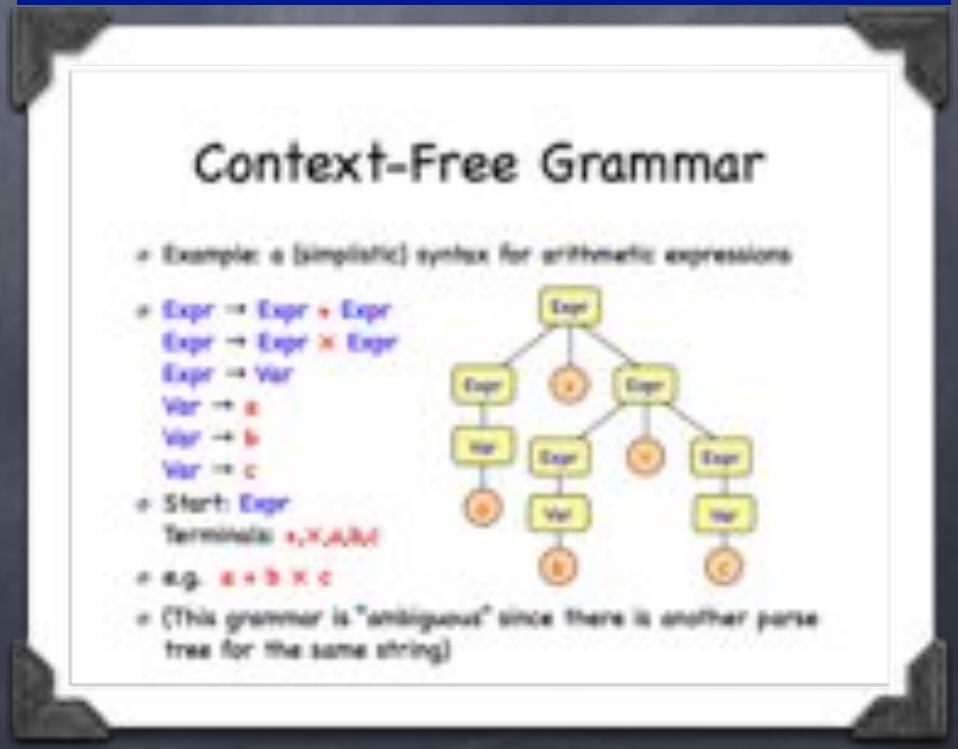
Counting

Computation
Algorithms, Circuits, Grammars,
Finite State Machines

Numbers

Graphs

Basic tools for expressing ideas
Logic, Proofs,
Sets, Relations, Functions



We Learned a Lot! :-)

Recursion

Trees

Bounding
big-O

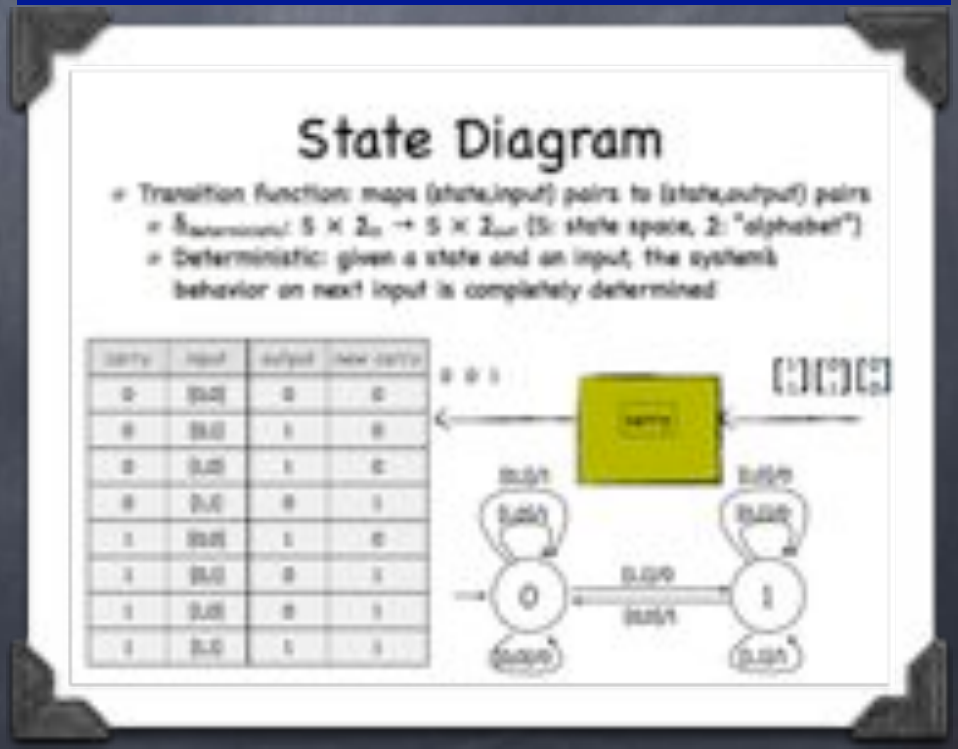
Induction

Counting

Numbers

Graphs

Computation
Algorithms, Circuits, Grammars,
Finite State Machines



Basic tools for expressing ideas
Logic, Proofs,
Sets, Relations, Functions

We Learned a Lot! :-)

Recursion

Trees

Bounding
big-O

Induction

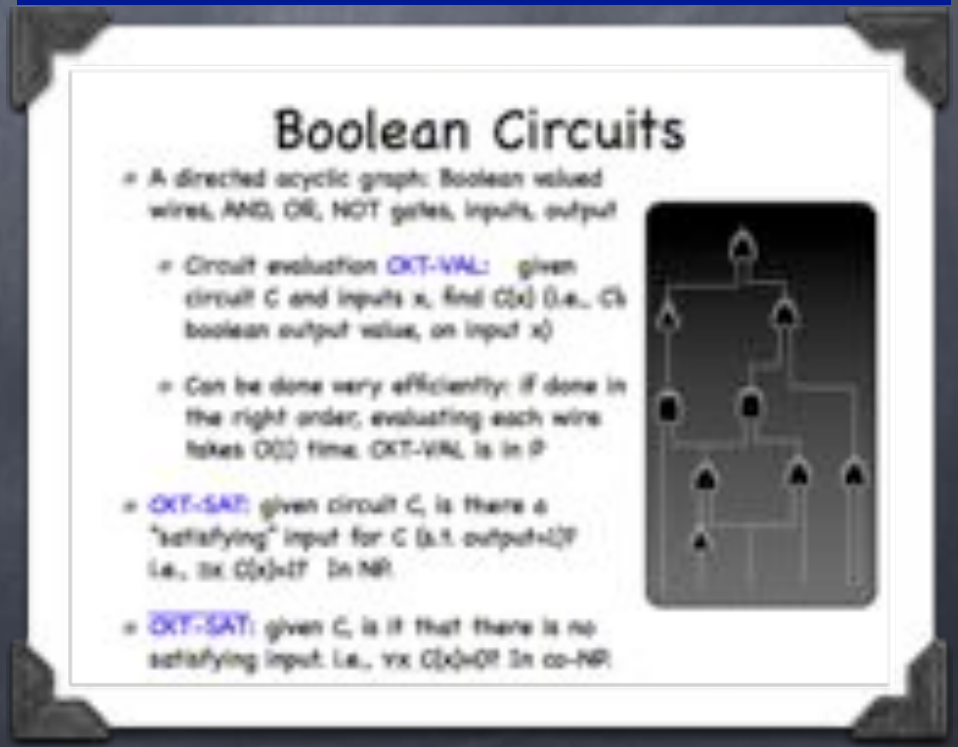
Counting

Computation
Algorithms, Circuits, Grammars,
Finite State Machines

Numbers

Graphs

Basic tools for expressing ideas
Logic, Proofs,
Sets, Relations, Functions



We Learned a Lot! :-)

Recursion

Induction

Trees

Bounding
big-O

Counting

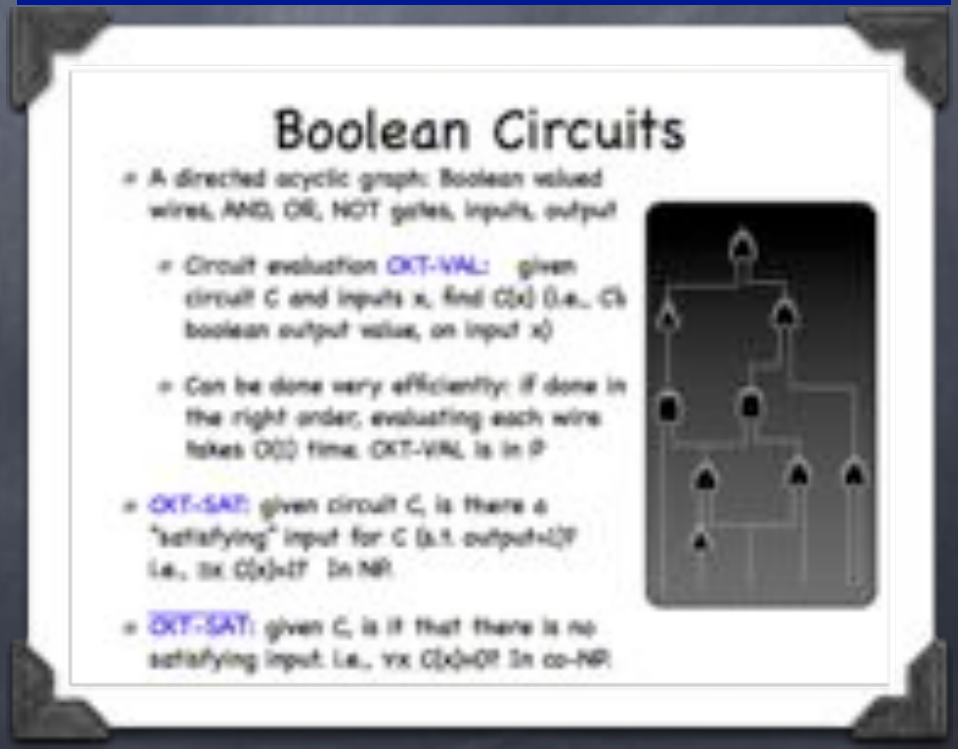
Numbers

Graphs

Basic tools for expressing ideas
Logic, Proofs,
Sets, Relations, Functions

P vs. NP

Computation
Algorithms, Circuits, Grammars,
Finite State Machines



We Learned a Lot! :-)

Recursion

Trees

Bounding
big-O

Induction

Counting

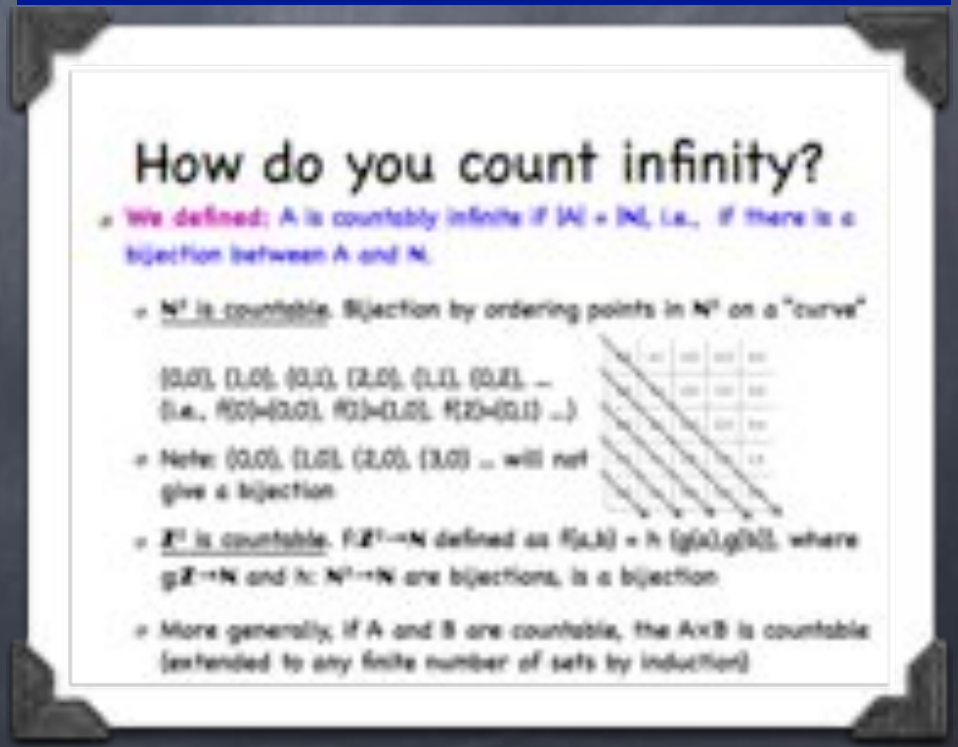
P vs. NP

Computation
Algorithms, Circuits, Grammars,
Finite State Machines

Numbers

Graphs

Basic tools for expressing ideas
Logic, Proofs,
Sets, Relations, Functions



How do you count infinity?

- We defined: A is countably infinite if $|A| = |\mathbb{N}|$, i.e., if there is a bijection between A and $\mathbb{N}$.
  - $\mathbb{N}^2$ is countable. Bijection by ordering points in $\mathbb{N}^2$ on a "curve"

    (0,0), (1,0), (0,1), (2,0), (1,1), (0,2), ...
    (i.e., f(0)=(0,0), f(1)=(1,0), f(2)=(0,1) ...)
  - Note: (0,0), (1,0), (2,0), (3,0) ... will not give a bijection
  - $\mathbb{Z}^2$ is countable. $f: \mathbb{Z}^2 \to \mathbb{N}$ defined as $f(a,b) = h(g(a),g(b))$, where $g: \mathbb{Z} \to \mathbb{N}$ and $h: \mathbb{N}^2 \to \mathbb{N}$ are bijections, is a bijection
  - More generally, if A and B are countable, the $A \times B$ is countable (extended to any finite number of sets by induction)

We Learned a Lot! :-)

Recursion

Trees

Bounding
big-O

Uncomputability

P vs. NP

Induction

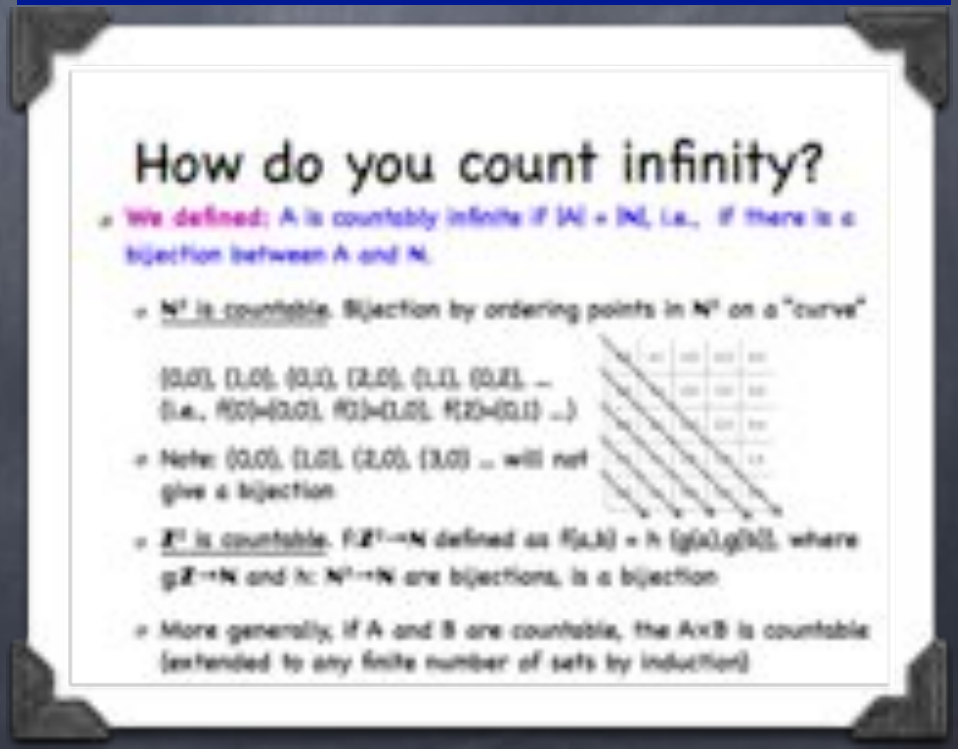
Counting

Computation
Algorithms, Circuits, Grammars,
Finite State Machines

Numbers

Graphs

Basic tools for expressing ideas
Logic, Proofs,
Sets, Relations, Functions



How do you count infinity?

- We defined: A is countably infinite if $|A| = |\mathbb{N}|$, i.e., if there is a bijection between A and $\mathbb{N}$.
  - $\mathbb{N}^2$ is countable. Bijection by ordering points in $\mathbb{N}^2$ on a "curve"

    (0,0), (1,0), (0,1), (2,0), (1,1), (0,2), ...
    (i.e., f(0)=(0,0), f(1)=(1,0), f(2)=(0,1) ...)
  - Note: (0,0), (1,0), (2,0), (3,0) ... will not give a bijection
  - $\mathbb{Z}^2$ is countable. $f: \mathbb{Z}^2 \to \mathbb{N}$ defined as $f(a,b) = h(g(a),g(b))$, where $g: \mathbb{Z} \to \mathbb{N}$ and $h: \mathbb{N}^2 \to \mathbb{N}$ are bijections, is a bijection
  - More generally, if A and B are countable, the $A \times B$ is countable (extended to any finite number of sets by induction)

We Learned a Lot! :-)

Recursion

Trees

Bounding
big-O

Uncomputability

P vs. NP

Induction

Counting

Computation
Algorithms, Circuits, Grammars,
Finite State Machines

Numbers

Graphs

Basic tools for expressing ideas
Logic, Proofs,
Sets, Relations, Functions

Logistics Wrap-Up

Logistics Wrap-Up

- Final Exam on 18th 8:00 AM

Logistics Wrap-Up

- Final Exam on 18th 8:00 AM
  - Locations: Main Library 66 (here) & Psychology 23

Logistics Wrap-Up

- Final Exam on 18th 8:00 AM
  - Locations: Main Library 66 (here) & Psychology 23
    - Seating details TBA

Logistics Wrap-Up

- Final Exam on 18th 8:00 AM
  - Locations: Main Library 66 (here) & Psychology 23
    - Seating details TBA
- Office hours: changes posted on Piazza

Logistics Wrap-Up

- Final Exam on 18th 8:00 AM
  - Locations: Main Library 66 (here) & Psychology 23
    - Seating details TBA
- Office hours: changes posted on Piazza
- Tentative grades post midterm 2 to be posted today

Logistics Wrap-Up

- Final Exam on 18th 8:00 AM
  - Locations: Main Library 66 (here) & Psychology 23
    - Seating details TBA
- Office hours: changes posted on Piazza
- Tentative grades post midterm 2 to be posted today
- HW solutions to be posted tomorrow

Logistics Wrap-Up

- Final Exam on 18th 8:00 AM
  - Locations: Main Library 66 (here) & Psychology 23
    - Seating details TBA
- Office hours: changes posted on Piazza
- Tentative grades post midterm 2 to be posted today
- HW solutions to be posted tomorrow
- A short PHQ9 to be released tomorrow, due Monday morning. HW11 due Friday.

Logistics Wrap-Up

- Final Exam on 18th 8:00 AM
  - Locations: Main Library 66 (here) & Psychology 23
    - Seating details TBA
- Office hours: changes posted on Piazza
- Tentative grades post midterm 2 to be posted today
- HW solutions to be posted tomorrow
- A short PHQ9 to be released tomorrow, due Monday morning. HW11 due Friday.
- Now: ICES forms