CMPSC 464
Intro to the Theory of Computation

About

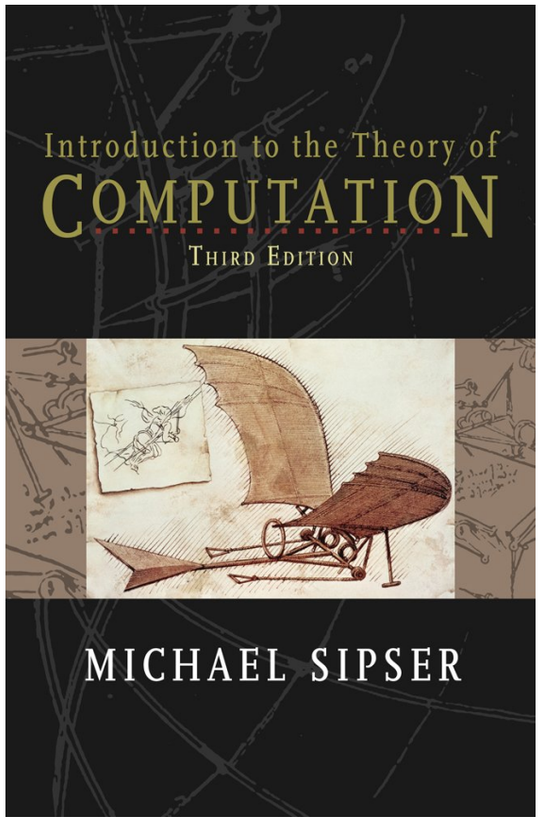
Lecturer: Pei Wu Email: pei.wu@psu.edu

Time: TR 16:35-17:50pm Location: Sparks 121.

Office hours: T 15:30-16:30pm @Westgate W315

TAs: Medha Kumar, Danrong Li, Haowei Li, Zecheng Li

Textbook: *Introduction to the Theory of Computation*, 3rd Edition, Michael Sipser



Motivating question: What can be computed efficiently?

Min cut

Shortest path

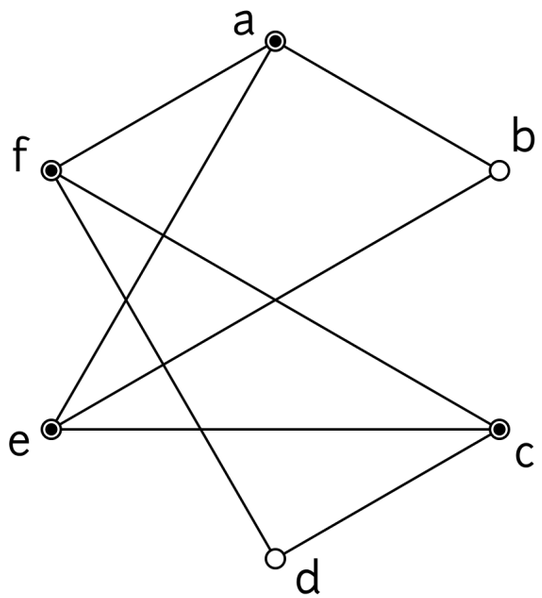
Eulerian path

Find a perfect matching

Determinant of a matrix

2-SAT

2-COLOR



Max cut

Longest simple path

Hamiltonian path

Count perfect matchings

Permanent of a matrix

3-SAT

3-COLOR

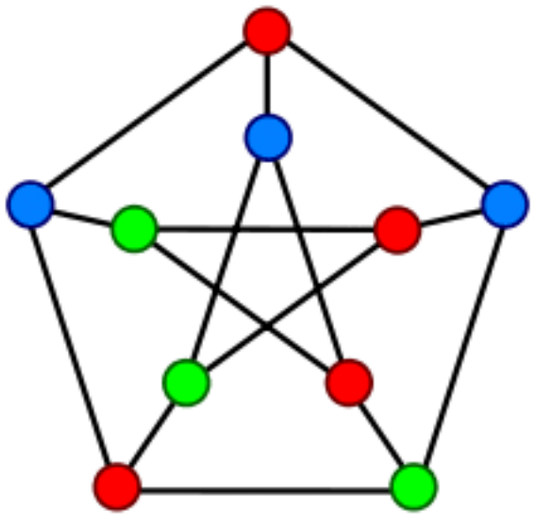
Complexity Theory

A million dollar question:

Nondeterministic polynomial-time (NP) vs polynomial-time (P)

"Efficiently verification ?= efficiently computable"

[Gödel '56, Cook '71, Levin '73]



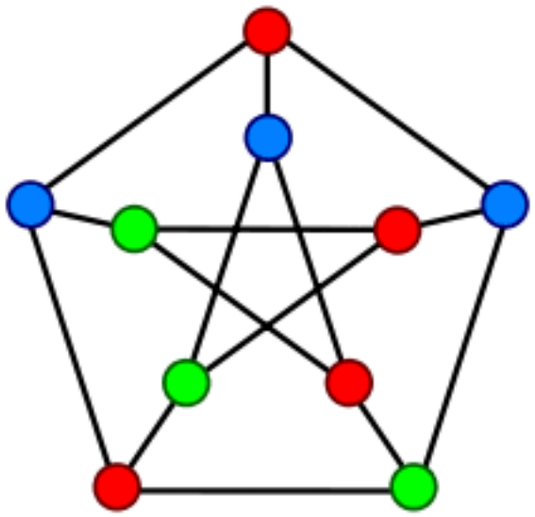
Complexity Theory

A million dollar question:

Nondeterministic polynomial-time (NP) vs polynomial-time (P)

"Efficiently verification ?= efficiently computable"

[Gödel '56, Cook '71, Levin '73]

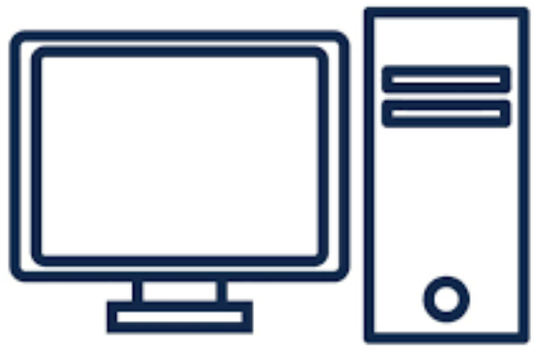


Completeness

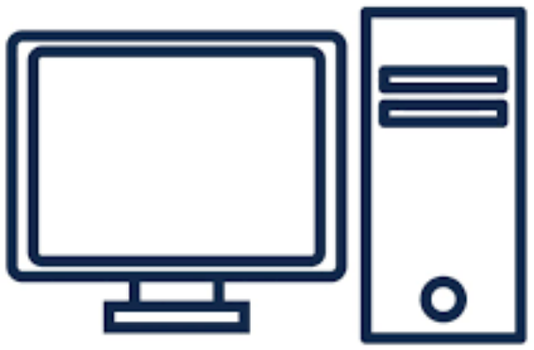
Cook-Levin Theorem

What is computation?
What can be computed?

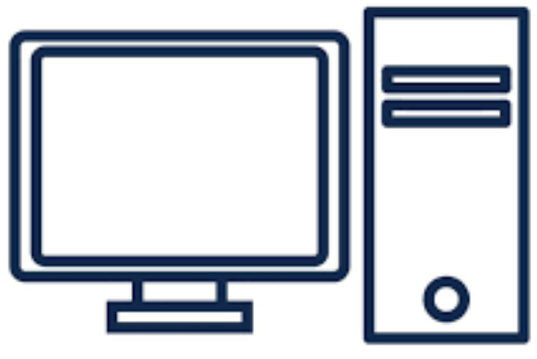
Motivating question: What is computation, what can be computed?

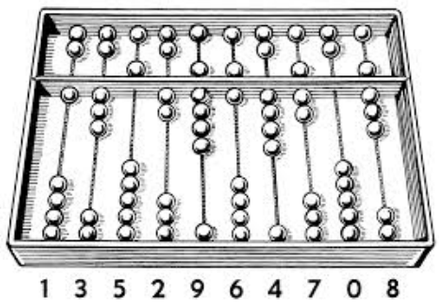


Motivating question: What is computation, what can be computed?

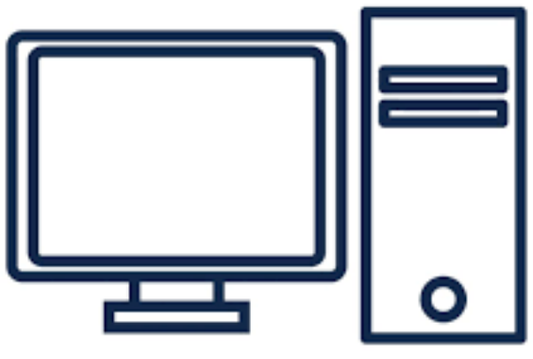


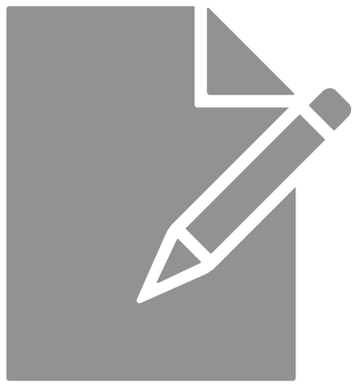
Motivating question: What is computation, what can be computed?

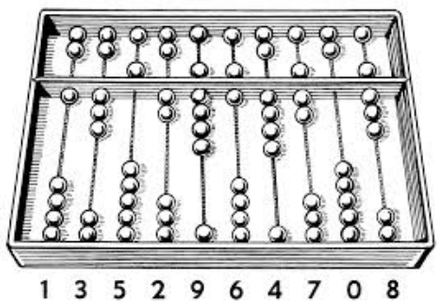


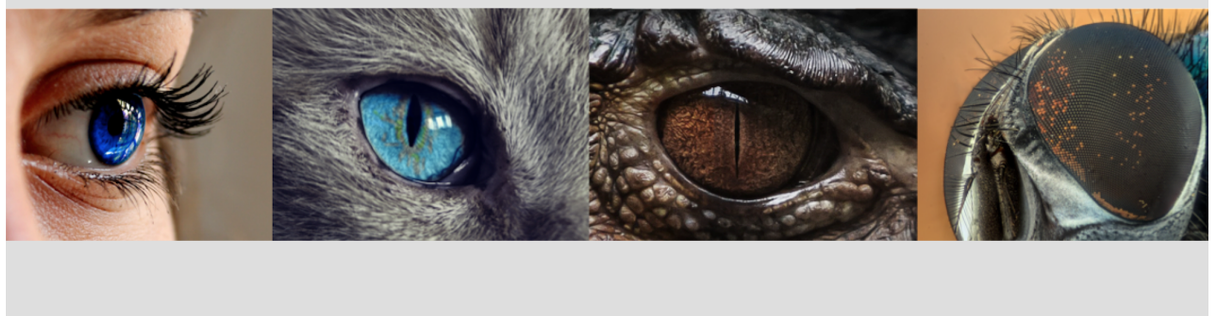


Motivating question: What is computation, what can be computed?



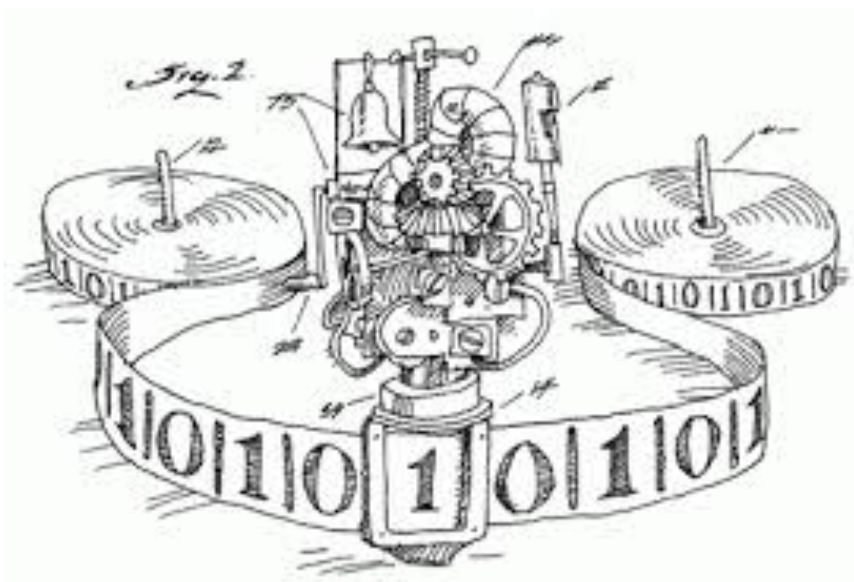






Motivating question: What is computation?

Turing machine [Turing '36]



Church-Turing Thesis

"a function on the natural numbers can be calculated by an effective method if and only if it is computable by a Turing machine...

Computability Theory

Exist uncomputable functions.

Halting problem: Give you a python code, and the input. Decide if the code halts in a finite amount of time.

Computability Theory

Exist uncomputable functions.
Halting problem: Give you a python code, and the input. Decide if the code halts in a finite amount of time.

Gödel's incompleteness theorem.
Roughly, there is a true statement about natural numbers that can not be proved!

Computability Theory

Exist uncomputable functions.
Halting problem: Give you a python code, and the input. Decide if the code halts in a finite amount of time.

Gödel's incompleteness theorem.
Roughly, there is a true statement about natural numbers that can not be proved!

Russell's paradox
"Suppose a barber shaves all men who do not shave themselves and only men who do not shave themselves. Then does the barber shave himself?

Computability Theory

Exist uncomputable functions.
Halting problem: Give you a python code, and the input. Decide if the code halts in a finite amount of time.

Gödel's incompleteness theorem.
Roughly, there is a true statement about natural numbers that can not be proved!

Russell's paradox
"Suppose a barber shaves all men who do not shave themselves and only men who do not shave themselves. Then does the barber shave himself?

Think about it!

Computability Theory

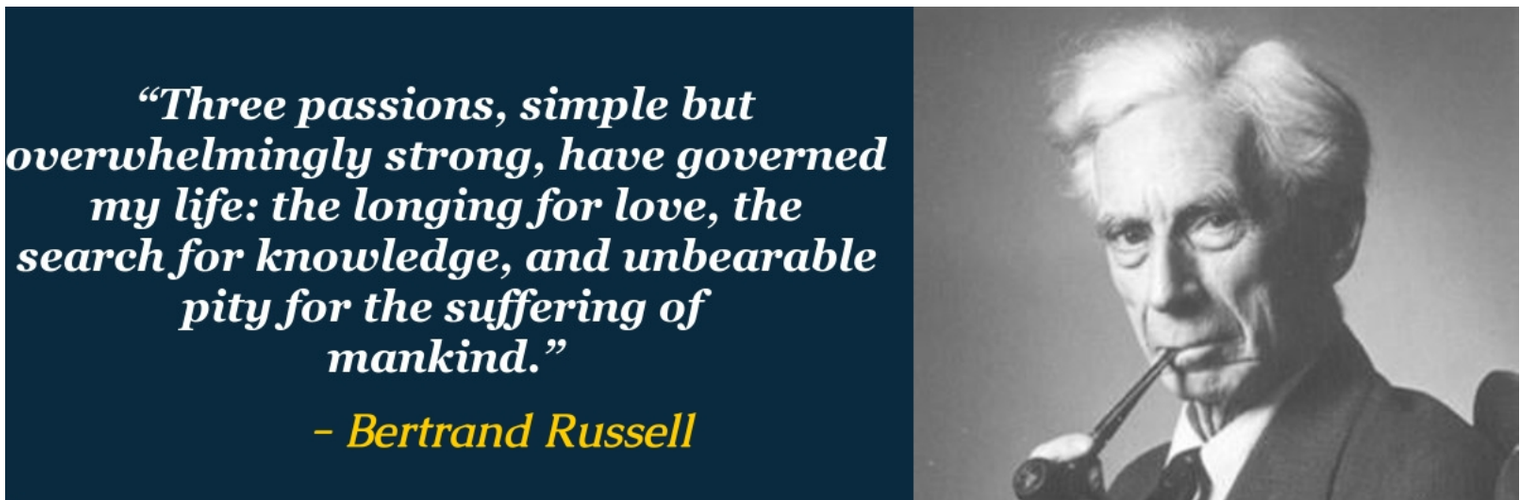
Exist uncomputable functions.
Halting problem: Give you a python code, and the input. Decide if the code halts in a finite amount of time.

Gödel's incompleteness theorem.
Roughly, there is a true statement about natural numbers that can not be proved!

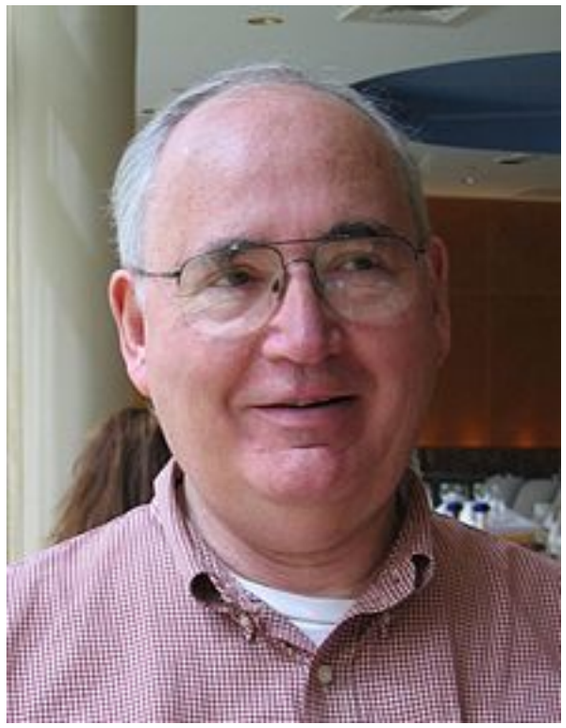
Russell's paradox

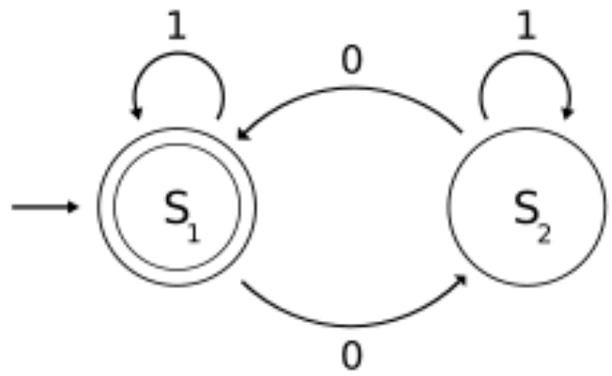
"Suppose a barber shaves all men who do not shave themselves and only men who do not shave themselves. Then does the barber shave himself?

Think about it!



Automata theory



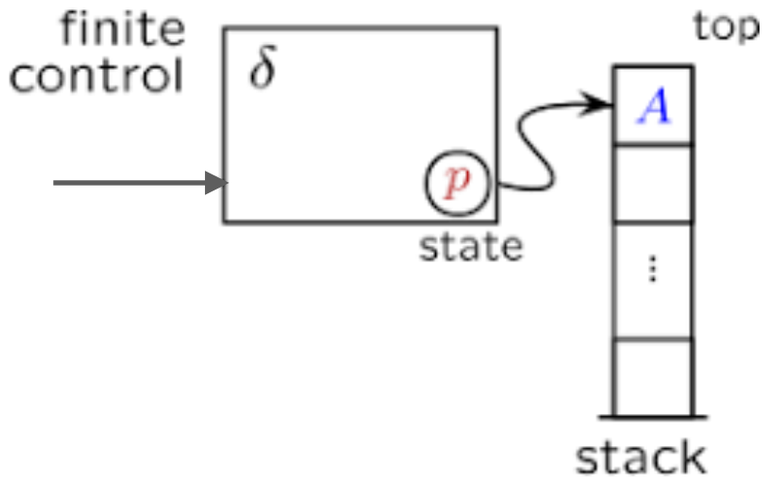


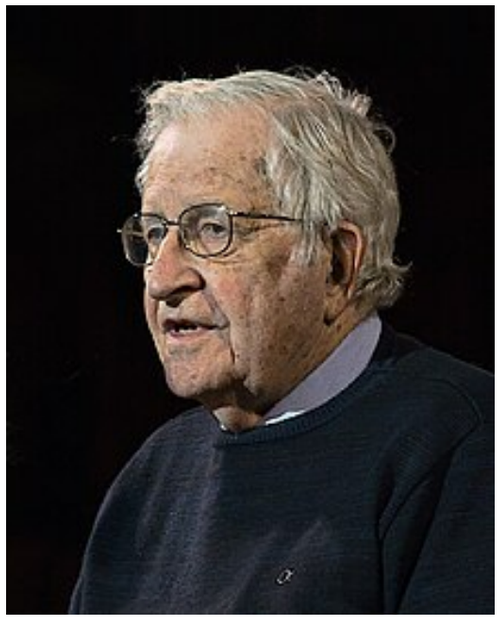
Finite-state automata
(Regular language)
[Rabin-Scott ’59]

Turing Award ’76

“for their joint paper "Finite Automata and Their Decision Problem," which introduced the idea of nondeterministic machines, which has proved to be an enormously valuable concept. Their (Scott & Rabin) classic paper has been a continuous source of inspiration for subsequent work in this field.

Automata theory





Pushdown automata
(Context-free language)
[Chomsky ’62, Evey ’63]

Part 1. Automata theory

Part 2. Computability theory

Part 3. Complexity theory

Logistics:

https://psu.instructure.com/courses/2377902

Sets, Languages, Proofs

Sets

A collection of <u>unordered</u> *elements*. Natural numbers $\mathbb{N} = \{1,2,3,\dots\}$,
odd number $= \{1,3,5,\dots\}$

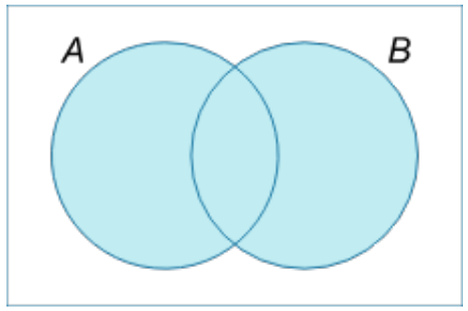
Set operation

Union $A \cup B$

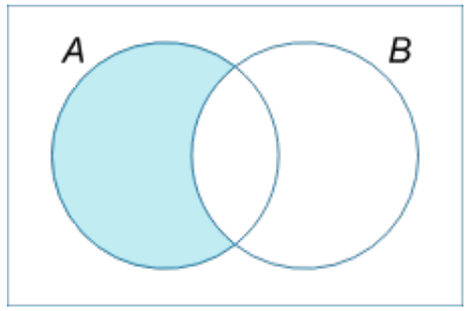
Intersection $A \cap B$

Set different $A \backslash B$

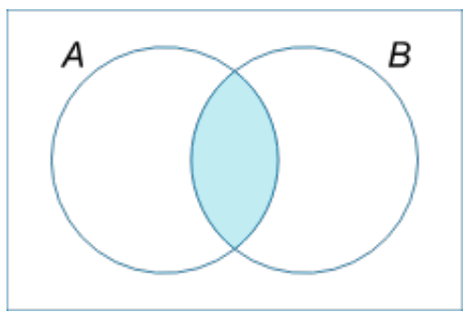
Complement $\bar{A}$



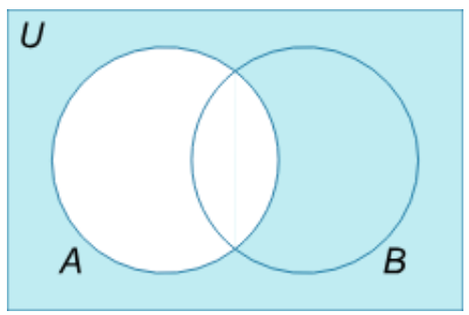
$A \cup B$



$A \backslash B$



$A \cap B$



$\bar{A}$

Sets

Fix two sets A, B,

Set Relation

- Inclusion. $A \subseteq B$, subset; $A \subsetneq B$ proper subset.
- Non-inclusion. $A \not\subseteq B$.
- Containment. $x \in A$.

Construction of new sets

- *Cartesian product* $A \times B := \{(a, b) : a \in A, b \in B\}$ (tuple)
  e.g. $\{0,1\} \times \{a, b\} = \{(0,a), (1,a), (0,b), (1,b)\}$.
- Power set $\mathscr{P}(A) := \{S \subseteq A\}$
  e.g. $\mathscr{P}(\{1,2\}) = \{\varnothing, \{1\}, \{2\}, \{1,2\}\}$

Sets

Cardinality of A, denoted $|A|$, is the number of elements in A

e.g. $A = \{1,2\}$, $|A| = 2$.

Def. $|A| = |B|$ iff there is a bijection map f between A and B.

e.g. $|\{1,2,3\}| = |\{a,b,c\}|$

$|\{1,2,3,\ldots\}| = |\{2,4,6,\ldots\}|$, $f : x \mapsto 2x$ is a bijection

Countable sets: finite, or of the same cardinality as $|\mathbb{N}|$, primes

Uncountable sets: real numbers, ……

Theorem (Cantor).

Real numbers are uncountable.

Strings

Fix an *alphabet* Σ,
a *string* $s \in \Sigma^* := \{\varepsilon\} \cup \Sigma \cup \Sigma^2 \cup \Sigma^3 \cdots$

e.g. Binary alphabet $\Sigma = \{0,1\}$, strings $111, 101001, \varepsilon, \ldots$
Digits $\Sigma = \{0,1,2,\ldots,9\}$, strings $20250114, 01010, \ldots$

String relations and operations

- *Concatenation*. e.g. $000 \circ 111 := 000111, x \circ y, xy$
- *Prefix*, proper prefix.
- *Suffix*, proper suffix.
- *Substring*. Any string obtained by deleting prefix/suffix from the given string
  e.g. 101 is a substring of 010101
- *Subsequence*. Any string obtained by deleting symbols from the given string
  e.g. 000 is a subsequence of 010101

Languages

Fix an *alphabet* Σ, a language is a subset of Σ^*, i.e., a collection of strings.

e.g. $\Sigma = \{0,1\}$,
$\mathscr{L} =$ {strings containing no substring 11}
$\mathscr{L} =$ {binary encoding a connected graph}
$\mathscr{L} =$ {binary encoding of a compilable python algorithm}

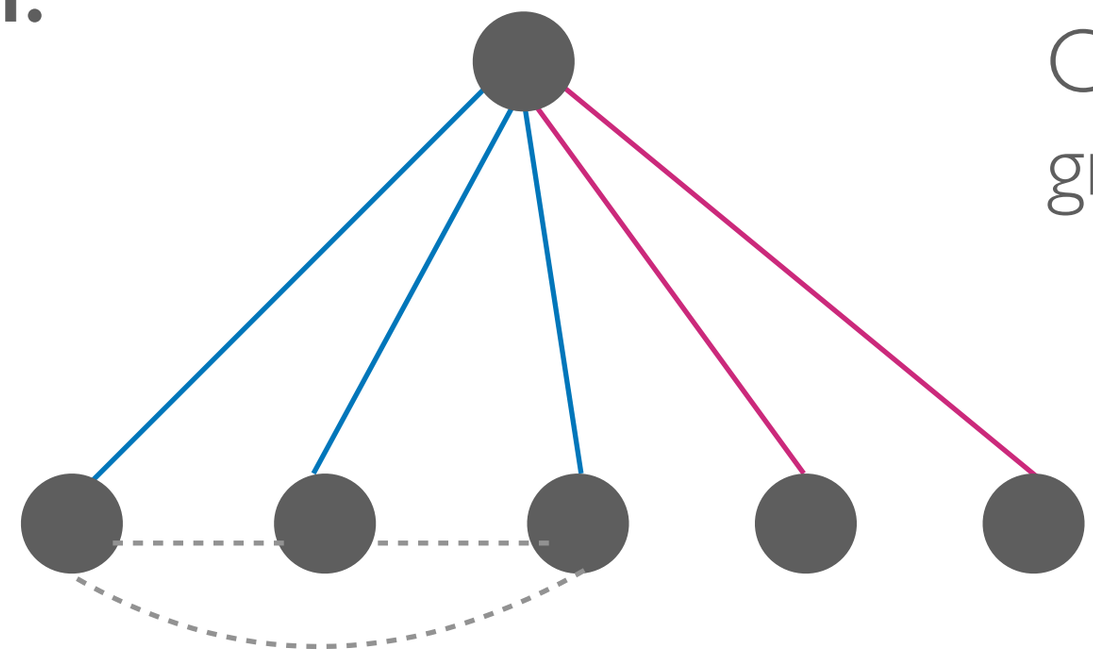
$\Sigma = \{0,1,2,\ldots,9\}$
$\mathscr{L} =$ {primes}
$\mathscr{L} = \{\}$

Proofs

- Logic reasoning

Fact. Among 6 people, there are 3 people who are either friends of each other,
or unknown to each other.

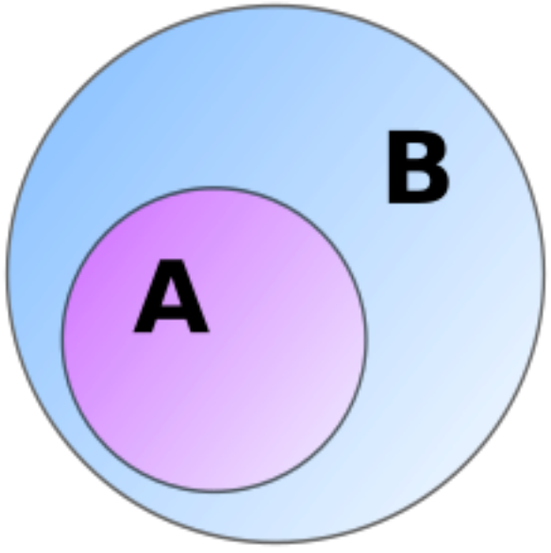
Pf.



Consider the possibility of the grey edges. ∎

Proofs

- By contraposition
  Logic equivalence. "$A \implies B$" is equiv. "$\neg B \implies \neg A$"
  e.g. "living things reproduce",
  "if sth. does not reproduce, not a living thing"



Fact. For $x \in \mathbb{N}$, if x^2 is even, then x is even.

Pf. If x is odd, then x^2 is odd. ∎

Proofs

- By contradiction

Fact. $\sqrt{2}$ is irrational.

Pf. Assume not, $\sqrt{2} = \frac{n}{m}$, for integer n, m.
Then $n^2 = 2m^2$. Wlog, n, m are not both even, so n is even.
But then m must be even.
A contradiction! ∎

Proofs

- By induction

Fact. Every tree graph has $n - 1$ edges.

Pf. By induction.
Base case: $n = 1$. Trivial.
Inductive hypothesis: Suppose $n \leq k$, the statement is true.
Inductive step: $n = k + 1$, take a leaf v, v is adjacent to one edge. Remove v and the edge. ∎