CSCI 2200
Foundations of Computer Science

Lecture 2:
Discrete Structures & Elements of Proofs

CSCI 2200

Foundations of Computer Science

Lecture 2: Discrete Structures & Elements of Proofs

“Time’s Scar” from *Chrono Cross*, Yasunori Mitsuda

Quick administrative stuff

Links to important stuff:

- Main website: https://www.cs.rpi.edu/~gittea/old-site/teaching/spring2025/focs.html
- List of assignments: https://www.cs.rpi.edu/~gittea/old-site/teaching/spring2025/assignments.html
- Piazza: https://piazza.com/class/m5gy7aew1i72ka/
- My slides: https://www.cs.rpi.edu/~ditursi/FOCS/slides/

These links are also in Course Materials in Submitty.

Office hours

Prof. DiTursi – AE 123A – Tu 9:30-12:00, Th 14:00-16:30

TAs (all in AE 127)

Eric (Section 1): Tu 10:00-12:00
Mei (Section 2): M 12:00-14:00
Xingjian (Section 3): M 14:00-16:00
Jun (Section 5): Th 14:00-16:00 *(nb: change from Monday)*
Lilian (Head TA): F 14:00-16:00

Sunday office hours 14:00-18:00 in AE 118

Problem sets

- Assignments published Thursdays after class
  - Two groups of problems: Recitation problems and submission problems.
- Submission problems are due the following Thursday <u>before 9pm</u>
  - No late submissions accepted!
- Other than diagrams, work must be typeset and submitted as a single PDF file smaller than 10MB

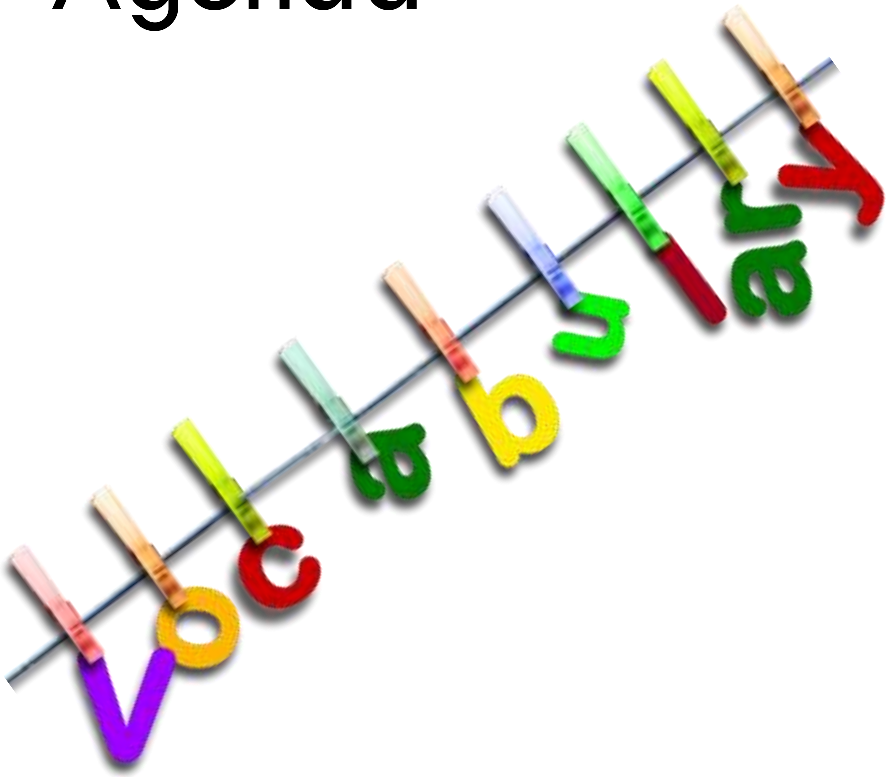
Recitations

- All recitations are in Troy 2018.
- During recitations, your TA will present solutions to the recitation problems.
- After that, your TA and an undergrad mentor will be available to answer student questions.
- There will be no recitations on exam days (1/29, 2/26, and 4/9).

A note on slides

- My slides are (intentionally) not designed as stand-alone reading material – they're meant to be *presented*.
- They are posted for you to look back on after class, in the hopes that they will help spark your memory; however, they are not a substitute for:
    - Attending class
    - Taking your own notes
    - Reading the textbook

Agenda



Building blocks of discrete math:
 * Sets
 * Sequences
 * Graphs

Mathematical / logical statements

Intro to propositional logic

The Building Blocks of Discrete Math

Sets – the fundamental unit of mathematics

Simply a collection of things – any objects you like: numbers, people, shapes, variables, other sets, etc.

The things in the set are *elements* or *members*. The number of elements may be zero, finite, or infinite.

The order within a set does not matter, and there are no duplicates – objects are either in the set or not.

Set notation

Sets => capital letters: A, B, WF, Q_3, …

Certain special sets have other notation: $\mathbb{Z}$

Generic elements => lowercase letters: a, p, x_1, …

If we have specific elements (names, numbers), we can just use those directly.

Curly braces are used as set containers:

S = {a, s, m, r}
N = {Alice, Bob, Charlie, Doug, Eve}
R = {2, 4, 8, …}

$\in$ is used for membership: $m \in S$, Eve $\in$ N

Special sets – everything and nothing

We generally reserve U (often $\mathcal{U}$ or $\mathbb{U}$ in texts) to indicate the *universe of discourse* – the set of every object we could be talking about in this context.

The set containing **no** elements is called the *empty set*, and is often of great importance. We represent the empty set with either { } or ∅

NOT {∅} – that would be the set containing the empty set, which is not empty!

Special sets – types of numbers

$\mathbb{N}$ - natural numbers: {1, 2, 3…}

$\mathbb{N}_0$ - whole numbers: {0, 1, 2, 3, …}

$\mathbb{Z}$ - integers: {…, -3, -2, -1, 0, 1, 2, 3, …}

$\mathbb{Q}$ - rational numbers: anything that can be written as a fraction of two integers

$\mathbb{R}$ - real numbers: the full (continuous) number line, rationals + irrationals (e, π, sqrt(2), etc.)

Infinite sets

We (obviously) can't list every element in an infinite set. If we are being quick (sloppy), we can use an ellipsis and **hope** everyone gets the idea:

$$E = \{2, 4, 6, \ldots\}$$

If this is the set of even positive integers, we're probably fine. If not… 😬

Set descriptions and set builder notation

It would be more precise to just go ahead and describe the set we intend:

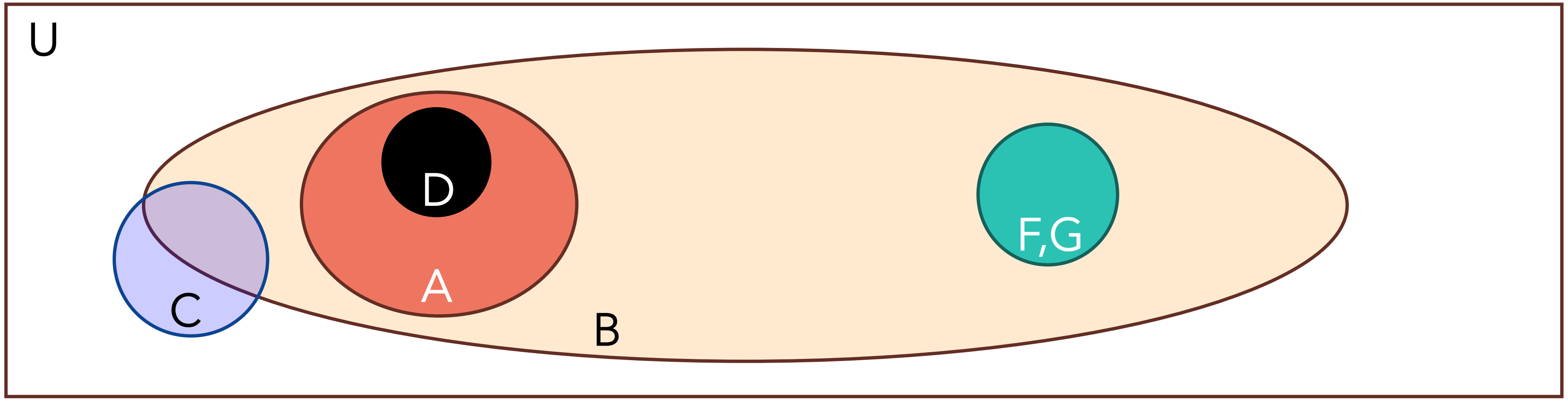
$$E = \{\text{all even positive integers}\}$$

We can also use variables in a form called *set builder notation* – this often still includes some English description as well:

$$E = \{\, n \mid n = 2k, \text{ where } k \in \mathbb{N} \,\}$$

We would read this as "the set of n such that n = 2k, where k is a positive integer."

Set relations (i.e. T/F statements)



Subset: Is the first set fully contained in the second set?
$A \subseteq B$, $D \subseteq A$, $D \subseteq B$, $C \not\subseteq B$

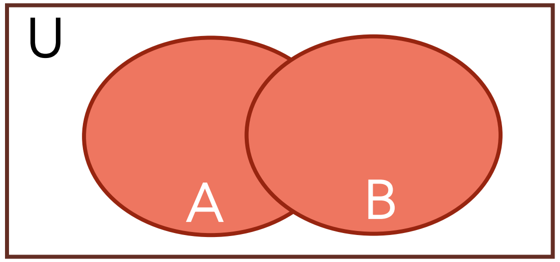
Superset: Does the first set fully contain the second set?
$B \supseteq A$, $A \supseteq D$, $B \supseteq D$, $B \not\supseteq C$

Proper Subset / Superset: Does the larger set have any elements outside the smaller set?
$F \subseteq G$, $F \not\subset G$, $F \subset B$

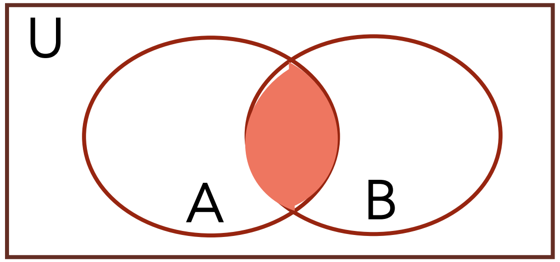
Set Equivalence: Do the sets contain precisely the same members? $F = G$

Set operations (new sets from known sets)

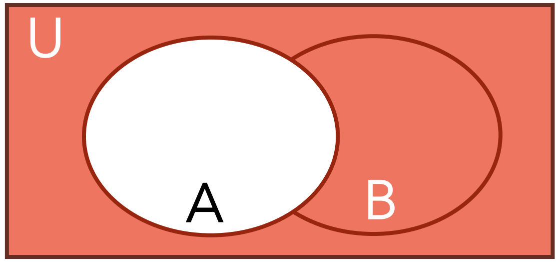
- *Union*: All elements in **either** set – $A \cup B$



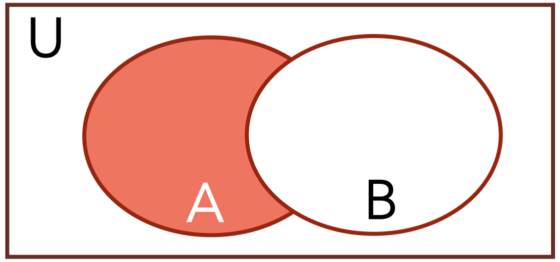
- *Intersection*: Only elements that are in **both** sets – $A \cap B$



- *Negation*: All elements from U not in a set – $\overline{A}$

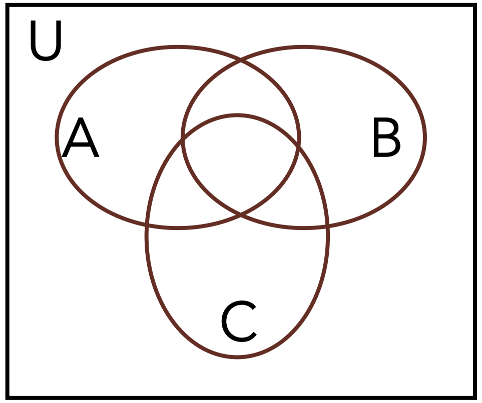


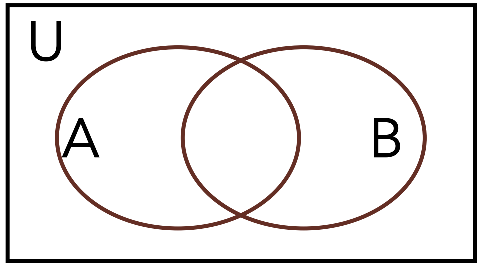
- *Set Difference*: All elements in one set and not in the other – $A - B$



Quick practice

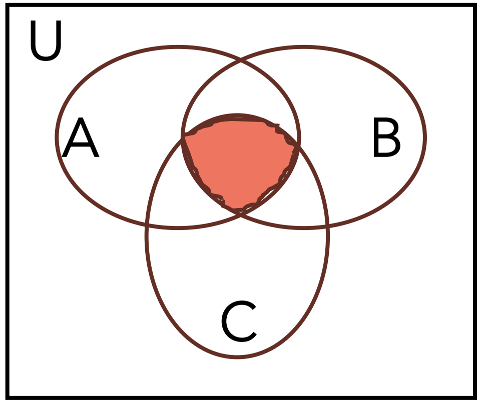
- Draw diagrams for the following:
  - $A \cap B \cap C$
  - $A \cup B \cap C$
  - $\overline{A \cup B}$
  - $\overline{A} \cap \overline{B}$





Quick practice

- Draw diagrams for the following:
  - $A \cap B \cap C$

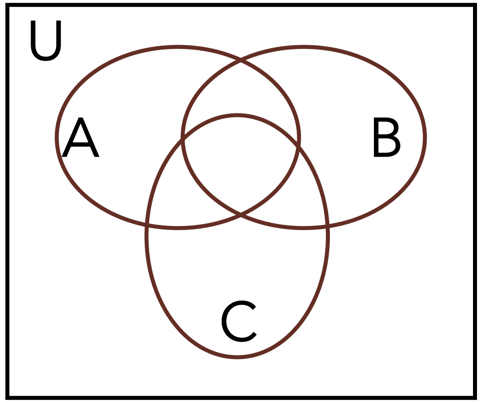


  - $A \cup B \cap C$
  - $\overline{A \cup B}$

  - $\overline{A} \cap \overline{B}$

Quick practice

- Draw diagrams for the following:
  - $A \cap B \cap C$
  - $A \cup B \cap C$

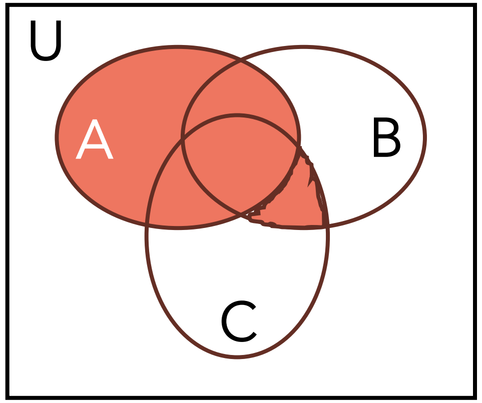


  - $\overline{A \cup B}$

  - $\overline{A} \cap \overline{B}$

Quick practice

- Draw diagrams for the following:
  - $A \cap B \cap C$



  - $A \cup B \cap C$

    $A \cup (B \cap C)$

    order of operations – negation*, intersection, union

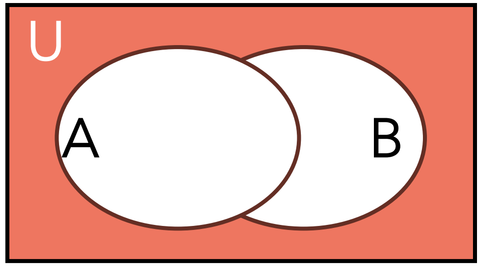
  - $\overline{A \cup B}$

  - $\overline{A} \cap \overline{B}$

Quick practice

- Draw diagrams for the following:
  - $A \cap B \cap C$
  - $A \cup B \cap C$

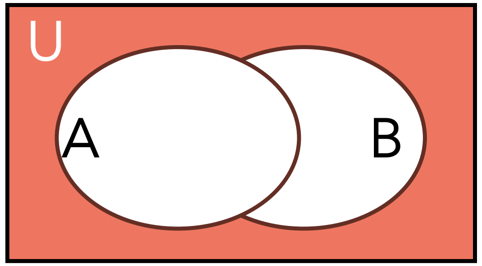
  - $\overline{A \cup B}$



  - $\overline{A} \cap \overline{B}$

Quick practice

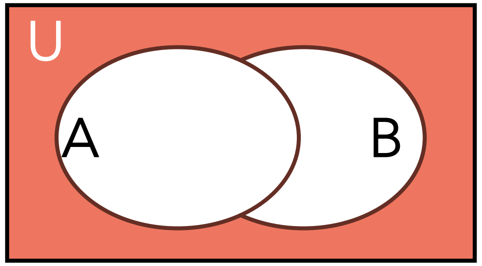
- Draw diagrams for the following:
  - $A \cap B \cap C$
  - $A \cup B \cap C$
  - $\overline{A \cup B}$
  - $\bar{A} \cap \bar{B}$



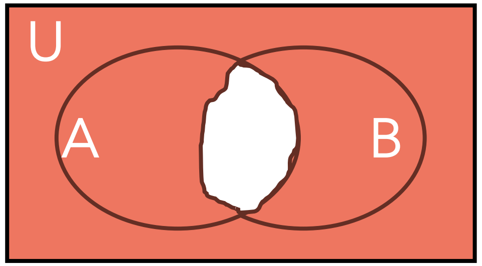
DeMorgan's Laws

- To "distribute" a negation into an intersection or union, negate all of the individual pieces, then swap intersection for union and vice versa.

- $\overline{A \cup B} = \overline{A} \cap \overline{B}$



- $\overline{A \cap B} = \overline{A} \cup \overline{B}$



Power set

- The *power set* operation (written 2^S) creates the set of all subsets.

- A = {x, y, z}

- $2^A = \{ \{\}, \{x\}, \{y\}, \{z\}, \{x,y\}, \{x,z\}, \{y,z\}, \{x,y,z\} \}$

Sequences

- A structure where you care about the order.
- This also implies that repetition matters.
- If we are talking about a sequence of symbols, we will also call that a *string*
  - Strings of letters: focs, abbabaabaaa
  - Binary (or bit) strings: 010, 1111010
  - The length-zero string is the *empty string* or the *null string* and is written ε or λ. (We'll use ε.)

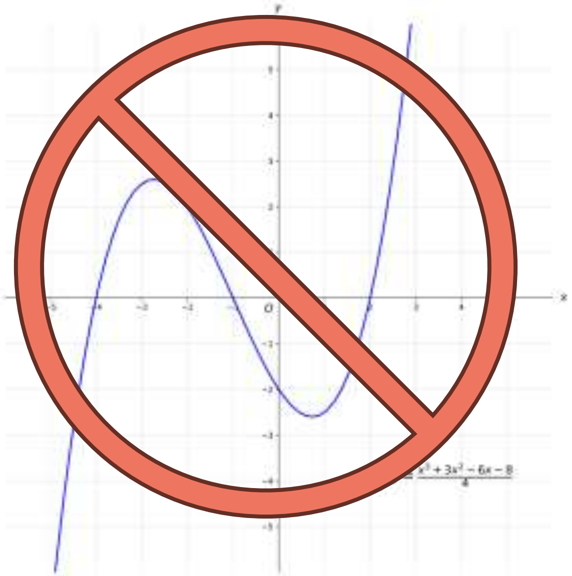
Well-known sequences

- P = {2, 3, 5, 7, 11, 13, 17, …}
- F = {0, 1, 1, 2, 3, 5, 8, 13, …}
  - What's F_{10}? (Note: $F_0 = 0$)

- H = {7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, 4, …}

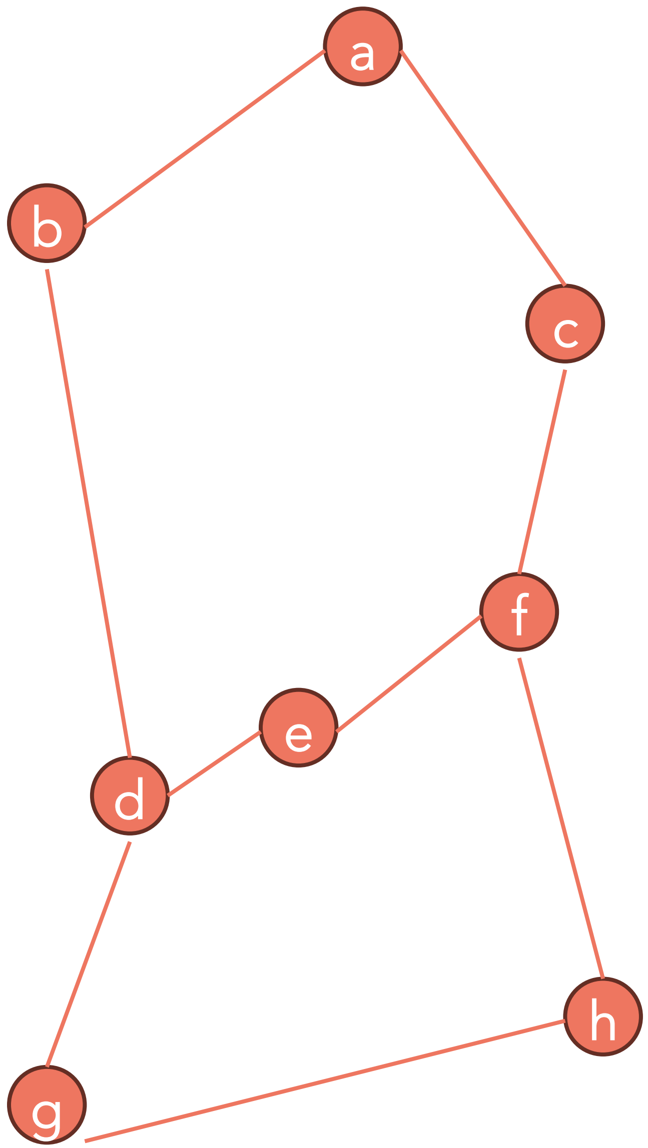
Graphs – modeling *connections*

- Sets do not model relationships between the elements, and sequences only have one type of relationship: precedes / follows
- If we want to be able to model relationships between arbitrary elements, we need a richer structure: a *graph*
- Note this is not a graph as the term is usually used in say, algebra or calculus. . .



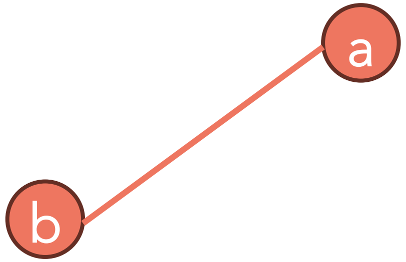
Components of a graph

- Graphs are defined as two sets:
  - *Vertices* (or *nodes*) are the objects we want to model. Often we'll use lowercase letters for these.
  - *Edges* (or *links*) are the connections between the nodes. An edge is written like this: (a,b)
- When drawn, the positions of the vertices DOES NOT MATTER!

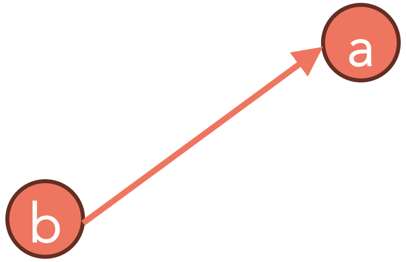


Types of edges

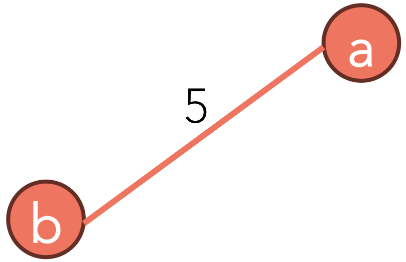
- Often, connection between vertices is transitive: that is, if *a* is connected to *b*, then *b* is automatically connected to *a*. This produces an *<u>undirected graph</u>*.



- If connections are **<u>not</u>** transitive, then the edge *(a,b)* is not the same is *(b,a)*, and we have a *<u>directed graph</u>*.

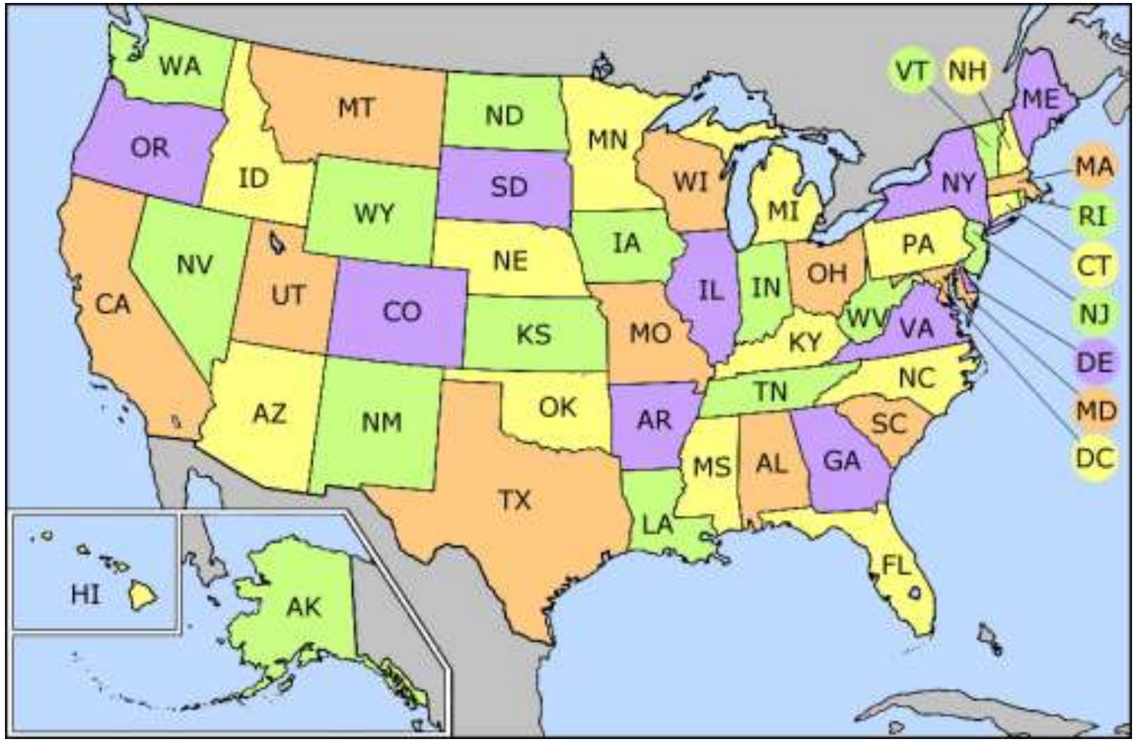


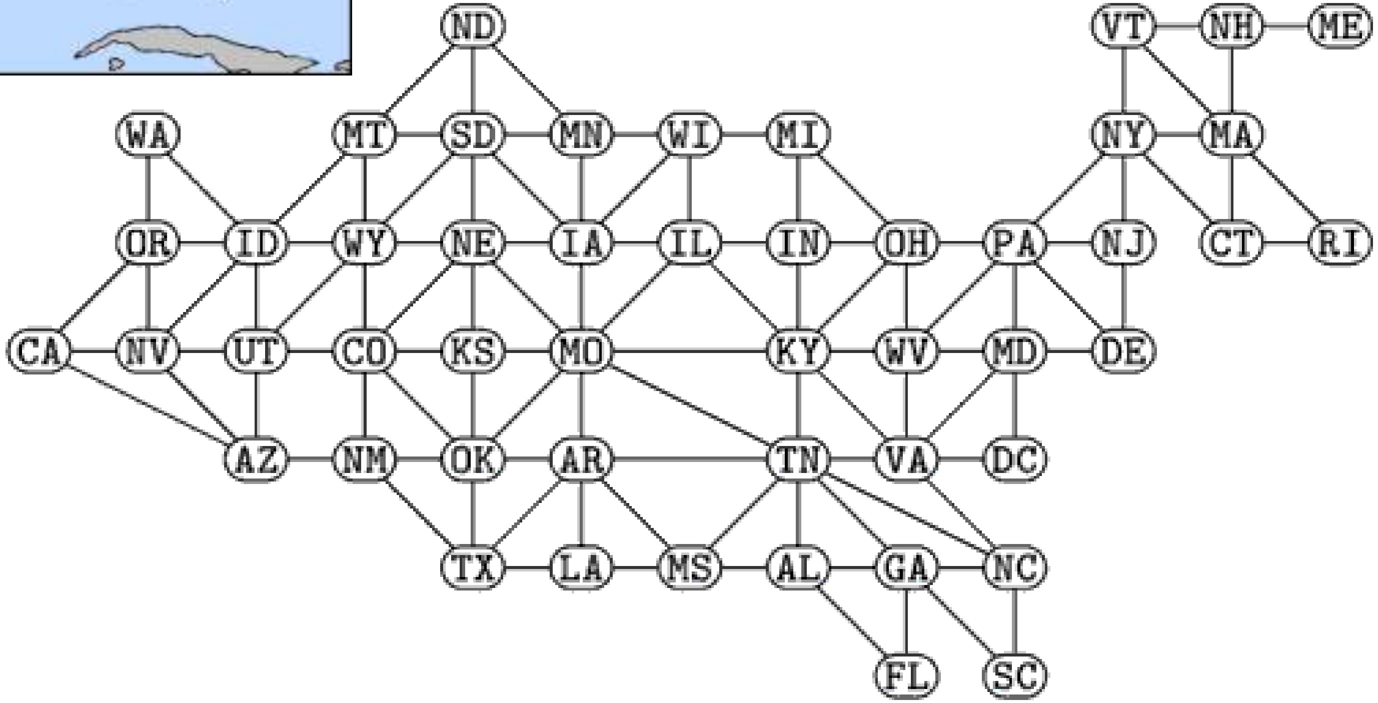
- Edges are sometimes labeled. Often these labels are numbers (indicating capacity, distance, what have you); these numbers are called *<u>weights</u>*.



Some types of graph models

- Social networks (note: social media not required!)
- Affiliation graphs (e.g. students & courses)
- Conflict graphs (edges = problem!)
- Similarity graphs (DNA / RNA analysis)
- Literal maps!


WA
OR
CA
ID
NV
MT
WY
UT
AZ
ND
SD
NE
CO
NM
KS
OK
TX
MN
IA
MO
AR
LA
WI
IL
MS
MI
IN
KY
TN
AL
OH
GA
FL
WV
VA
NC
SC
PA
NY
VT
NH
ME
MA
RI
CT
NJ
DE
MD
DC
HI
AK


ND
VT
NH
ME
WA
MT
SD
MN
WI
MI
NY
MA
OR
ID
WY
NE
IA
IL
IN
OH
PA
NJ
CT
RI
CA
NV
UT
CO
KS
MO
KY
WV
MD
DE
AZ
NM
OK
AR
TN
VA
DC
TX
LA
MS
AL
GA
NC
FL
SC

The Elements of Proof

Statements in English

- Consider the sentence:
  "She said she didn't take his money."
- How you say it matters a lot!

Statements in English

- Consider the sentence:
  "She said she didn't take his money."
- How you say it matters a lot!
  "**She** said she didn't take his money."
  "She **said** she didn't take his money."
  "She said **she** didn't take his money."
  "She said she **didn't** take his money."
  "She said she didn't **take** his money."
  "She said she didn't take **his** money."
  "She said she didn't take his **money**."

Ambiguity

- Even without emphasis, common language can leave meaning unclear. What does "another" mean here?
  - "What drink are you having? I'll get another." "What shirt are you wearing? I'll wear another."
- "Everything that glitters is not gold."
  - Do we mean that nothing that glitters is gold? Or that there are some things that glitter that are not gold?

Avoiding ambiguity

- When talking to a computer, it is necessary to be very precise about what you mean. There is no such thing as ambiguity to a computer – it'll just decide what an instruction means based upon its own rules.
- When working in mathematics, we insist upon the same precision – even if it makes our statements much more complicated, it must be absolutely clear what we mean.

What, precisely, is a proof?

- A proof generally starts with a *claim* – the logical statement you wish to demonstrate.
- Every proof relies on one or more *axioms* – statements that we accept as true without proof. Sometimes these are given in the problem statement; sometimes these are just fundamental ideas in math. (1 + 1 = 2)
- We then give a sequence of **true** statements that are designed to **convince** the reader that our claim is true. Ultimately, this is the point: To ensure that "everyone" (in whatever context) agrees on the truth of our claim. It then becomes a *theorem* we can use in other proofs.

An example proof: Parity of perfect squares

- Consider the following claim: The square of every odd integer is also odd.
- One can't simply say "well, 1^2 is 1, 3^2 is 9, 5^2 is 25… looks like it works for everything!"
- Most proofs rely heavily on the definition of the terms involved. In this case: what does "odd" mean?
- A useful definition is this: A number is odd precisely when it is equal to 2k+1 for some integer k.

An example proof: Parity of perfect squares

- Consider the following claim: The square of every odd integer is also odd.
- Using our definition, we can write:
$$(2k+1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1$$
- Next, we can note that whenever k is an integer, then so is $2k^2 + 2k$. (Why?)
  - The *closure properties* of integers are something that we take as axioms.
- Finally, we can state that the square of any odd integer can be written as two times an integer plus one, and therefore the square is also odd. *Quod erat demonstratum*.
- <u>Are you convinced?</u>

An example proof: Irrationality of $\sqrt{2}$

- Consider the following claim: The square root of 2 cannot be written as $\frac{a}{b}$, where a and b are integers.
- One often hears the phrase "you can't prove a negative." This is false, but what *is* true is that it's often more difficult to prove that something cannot exist than that it can exist.
- Often such proofs begin with the question "Well, what if it did exist?" Let's try that here…

An example proof: Irrationality of $\sqrt{2}$

- Imagine that $\sqrt{2} = \frac{a}{b}$ for some integers a and b, and that $\frac{a}{b}$ is in lowest terms. (If it's not, reduce it until it is.)
- We can square both sides and get $2 = \frac{a^2}{b^2}$ or $a^2 = 2b^2$.
- a^2 is therefore even, by the definition we just used.
- If a^2 is even, then a must also be even.
  - We actually just proved that in a sneaky way in our first example! More on that specific topic (the contrapositive) next class. For now, if you're not convinced, please take it on faith.

An example proof: Irrationality of $\sqrt{2}$

- If a is even, then $a = 2k$ for some integer k, and $a^2 = 4k^2$.
- Plugging into our previous equation, $4k^2 = 2b^2$, or $2k^2 = b^2$.
- This means b is also even, and $\mathrm{b} = 2m$ for some integer m.
- But if a and b are both even, then $\frac{a}{b} = \frac{2k}{2m}$ … which means that $\frac{a}{b}$ is not in lowest terms.

- But we specifically put it in lowest terms!
- We've got a big problem here!

Actually… the problem is solved!

- Creating an impossible situation is in fact the <u>goal</u> of proofs like this – it means that some hypothetical situation you proposed can't actually happen.
- Remember that we started this proof "Imagine that $\sqrt{2} = \frac{a}{b}$ for some integers a and b"
- Well, we did, and we got a *<u>paradox</u>*! So $\sqrt{2}$ must not be able to be written as a fraction!

Today's class survey

At the end of every class, I will ask you to complete a very brief survey about that day's lecture. The QR code at left contains the link to the survey – please complete it now.

Problem Set 1 will be posted around 10am; it is due on Thursday 16 January by 8:59pm.