CSCI 2200

Foundations of Computer Science

Lecture 0x01: Discrete Structures & Basic Logic

CSCI 2200

Foundations of Computer Science

Lecture 0x01: Discrete Structures & Basic Logic

"Time's Scar" from *Chrono Cross*, Yasunori Mitsuda

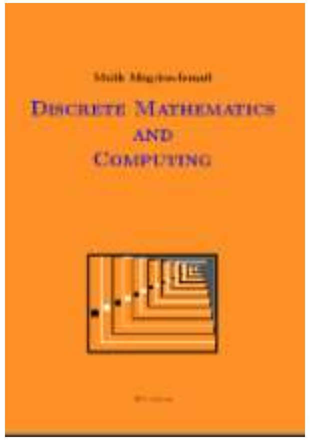
Quick administrative stuff

Office hours:

Boning's OH are now Fridays 14:00-18:00.
Dr. DiTursi's Thursday OH are now 8:30-9:45 and 13:00-14:45. (Tuesdays are unchanged at 9:00-12:00.)

Textbook:

Discrete Mathematics & Computing
Malik Magdon-Ismail
ISBN 978-0-578-56787-7



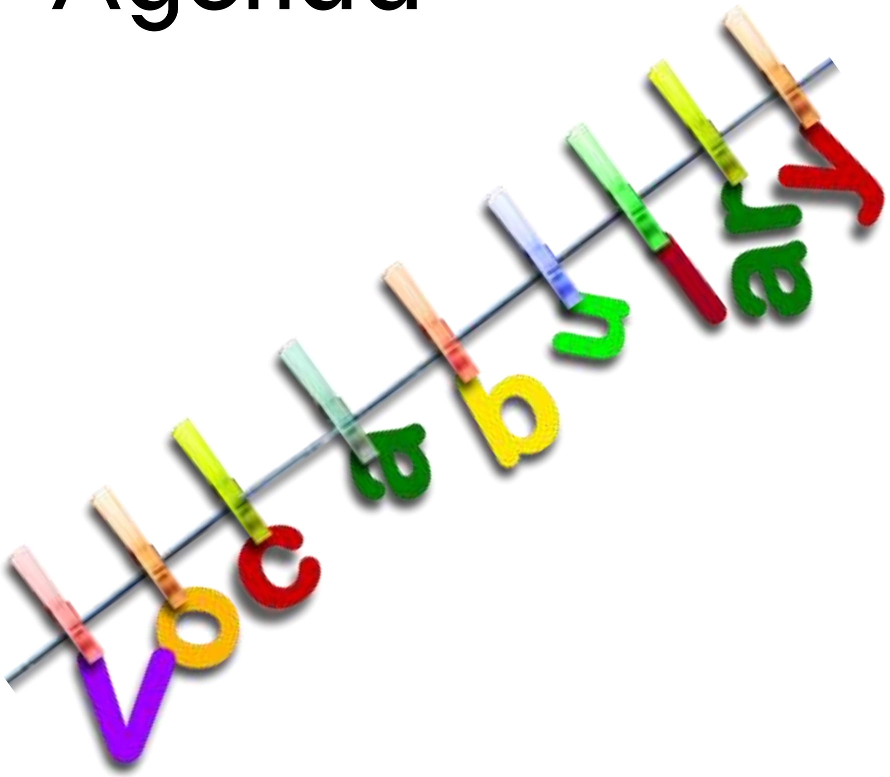
Homework submissions

- File size has been an issue.
- I’ve asked tech support if we can get the cap raised.
- In the meantime, I understand that there are PDF compressors that will do the trick.
- Due 8:59pm tonight.

Recitations and lab activities

- First recitation is tomorrow; attendance is (indirectly) graded.
- TA will introduce themselves and present solutions (or at least solution sketches) for the homework problems. (~20 minutes)
- Then lab activity will be distributed. You may work in groups of up to five.
- When complete, enter the Office Hours queue in Submitty. Someone will come check your work and mark it as complete.

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:

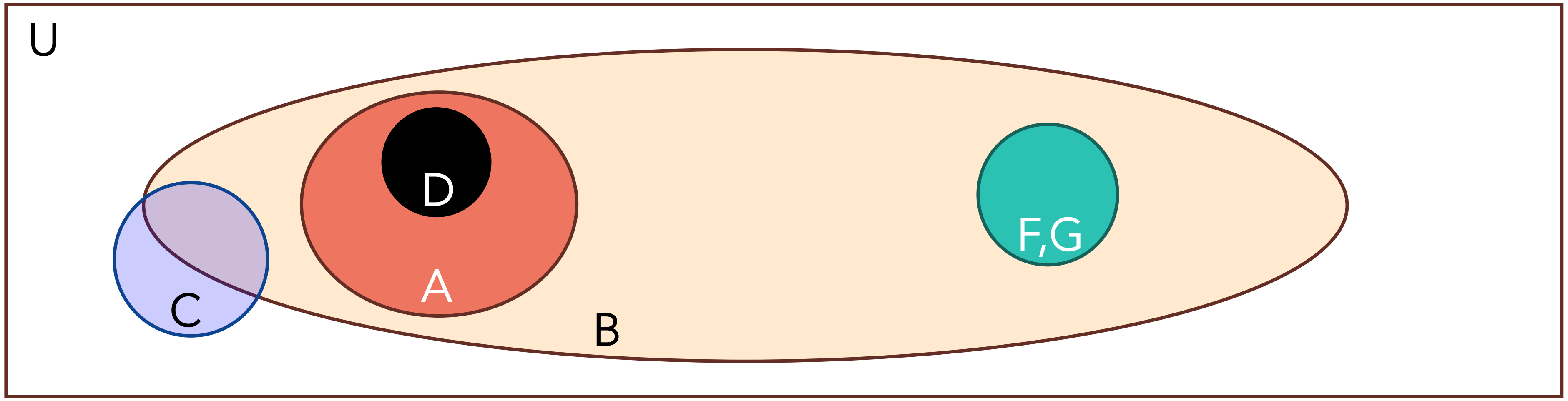
$$E = \{\text{all even positive integers}\}$$

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?
$A \subseteq B$, $D \subseteq A$, $D \subseteq B$, $C \not\subseteq B$

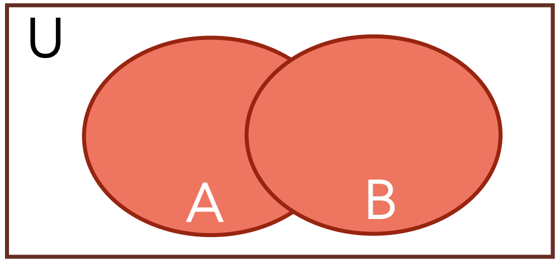
Superset: Does the first set fully contain the second set?
$B \supseteq A$, $A \supseteq D$, $B \supseteq D$, $B \not\supseteq C$

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

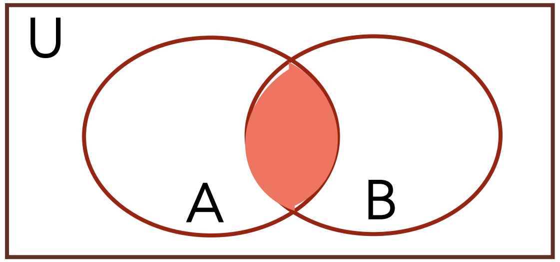
Set Equivalence: Do the sets contain precisely the same members? $F = G$

Set operations (new sets from known sets)

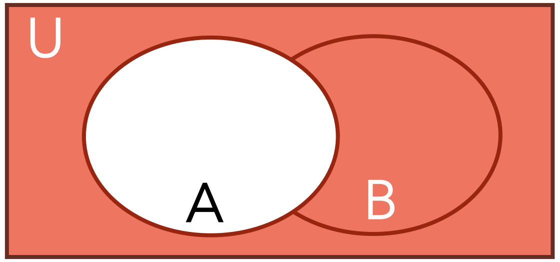
- *Union*: All elements in **either** set – $A \cup B$



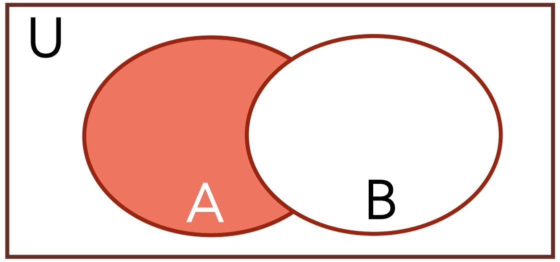
- *Intersection*: Only elements that are in **both** sets – $A \cap B$



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

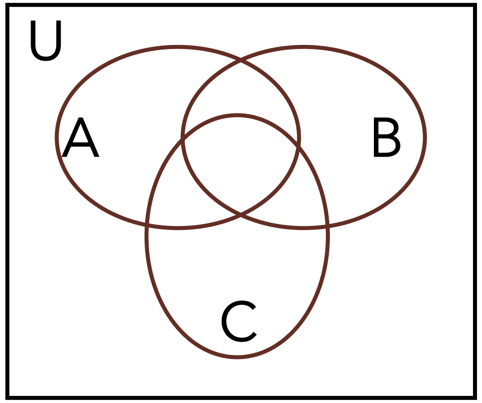


- *Set Difference*: All elements in one set and not in the other – $A - B$

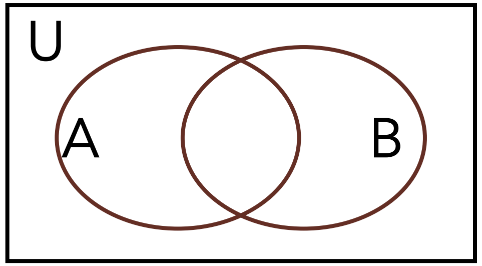


Quick practice

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

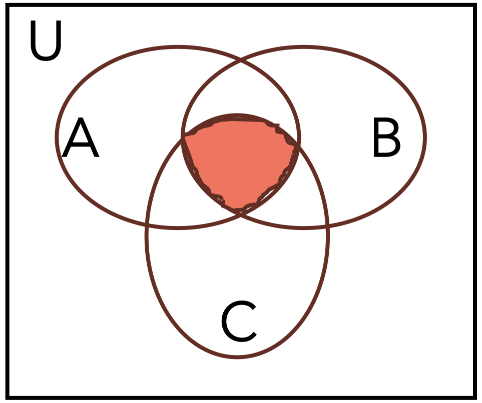


  - $\overline{A \cup B}$
  - $\overline{A} \cap \overline{B}$



Quick practice

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

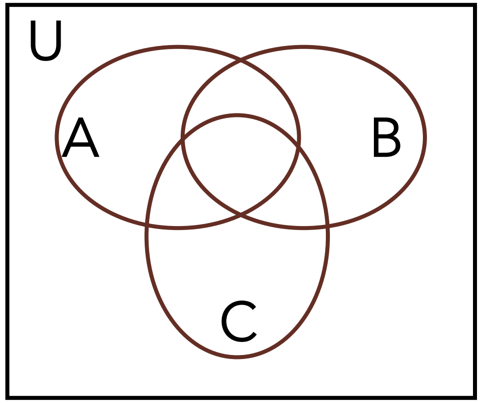


  - $A \cup B \cap C$
  - $\overline{A \cup B}$

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

Quick practice

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

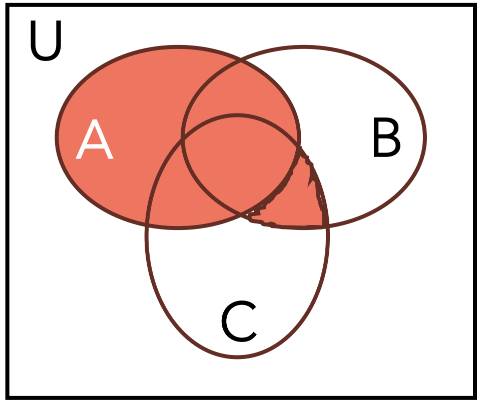


  - $\overline{A \cup B}$

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

Quick practice

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



  - $A \cup B \cap C$

    $A \cup (B \cap C)$

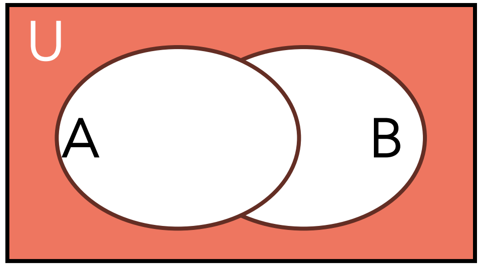
    order of operations – negation*, intersection, union
  - $\overline{A \cup B}$

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

Quick practice

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

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

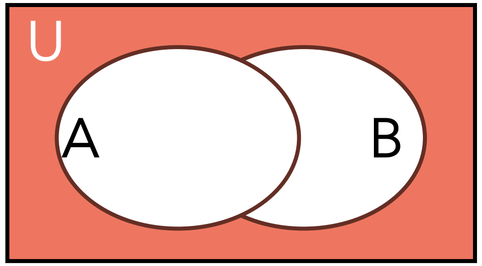


  - $\overline{A} \cap \overline{B}$

Quick practice

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

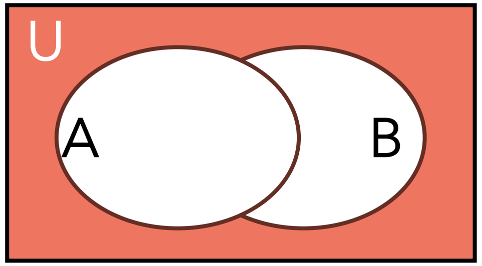
  - $A \cup B \cap C$
  - $\overline{A \cup B}$
  - $\bar{A} \cap \bar{B}$



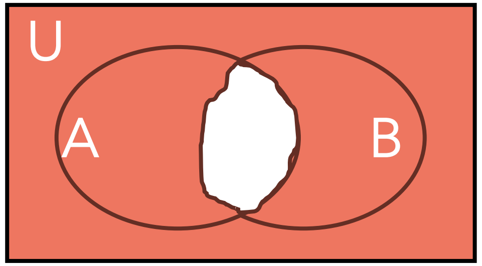
DeMorgan's Laws

- 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.

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

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

Sequences

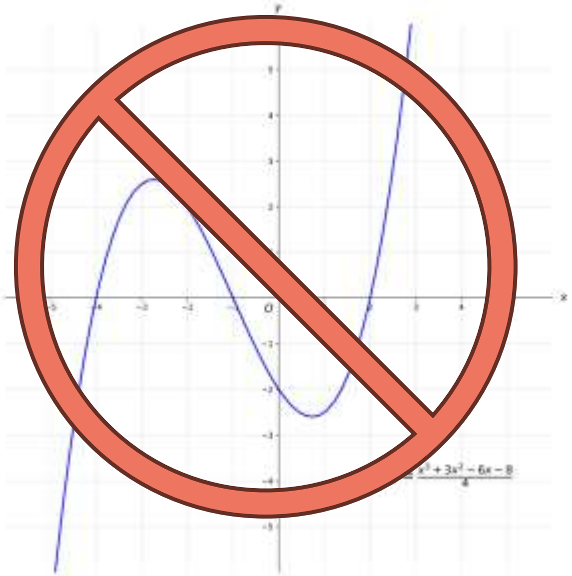
- A structure where you care about the order.
- This also implies that repetition matters.
- If we are talking about a sequence of symbols, we will also call that a *string*
  - Strings of letters: focs, abbabaabaaa
  - Binary (or bit) strings: 010, 1111010
  - The length-zero string is the *empty string*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

Well-known sequences

- $P = \{2, 3, 5, 7, 11, 13, 17, \ldots\}$
- $F = \{0, 1, 1, 2, 3, 5, 8, 13, \ldots\}$
  - What's F_{10}? (Note: $F_0 = 0$)
- $H = \{7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, 4, \ldots\}$

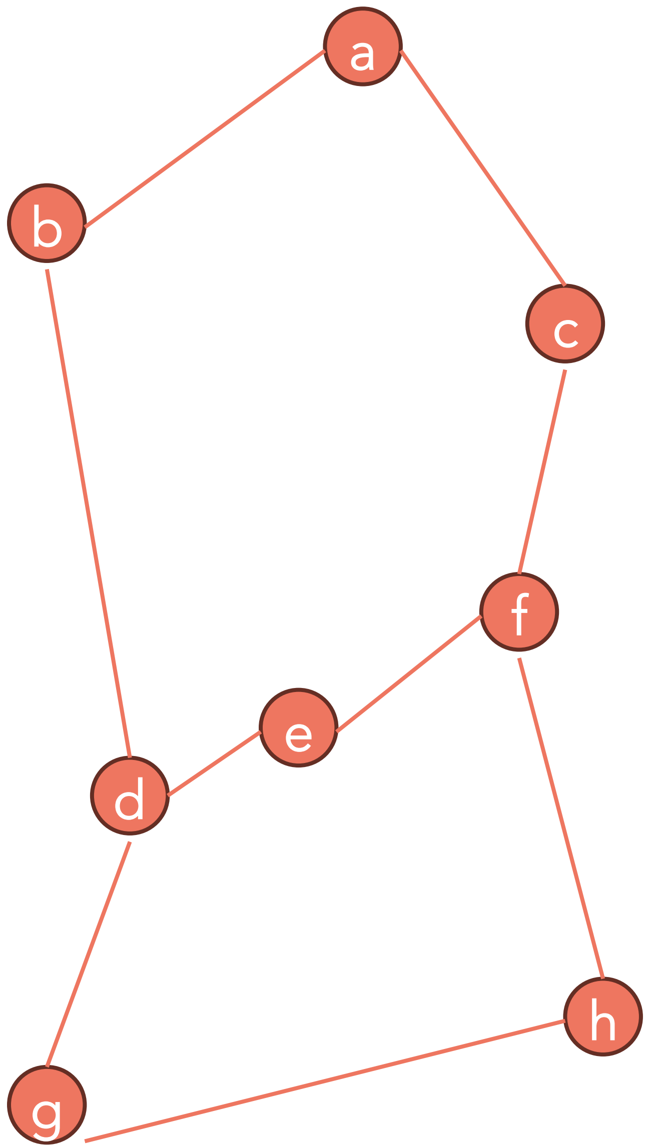
Graphs – modeling *connections*

- 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. . .



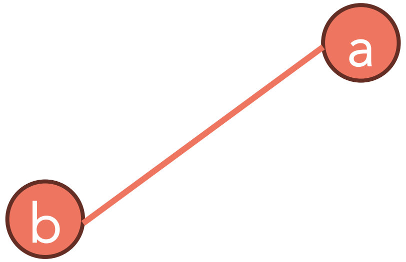
Components of a graph

- 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!

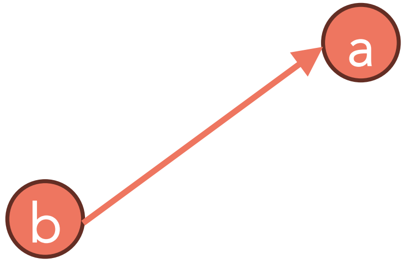


Types of edges

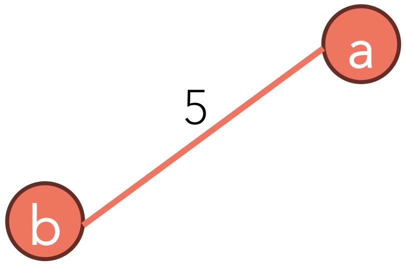
- Often, connection between vertices is transitive: that is, if *a* is connected to *b*, then *b* is automatically connected to *a*. This produces an *undirected graph*.



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



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



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!

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.

Propositions

- The most basic type of logical statement is a *proposition*. This is a sentence that can be assigned a *truth value*: True or False.
  - Note: We might not *know* whether or not it's true. But it must be one or the other; there is no third option. This is known as the Law of the Excluded Middle.

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.