Notes 9

Office hours monday.

Today we give an overview of set theory, including relations on sets.
Probably much of the first part of this will be review from high school math.

We will define an interesting kind of relation called an *equivalence relation*.

An important example of equivalence relation is number equivalence mod m, where m is a positive integer. We will define this relation and give some of its properties.
We will conclude with an application of modular equivalence ideas to get fast RSA encryption and decryption.

9 Sets

9.1 Definitions

A *set* is a collection of objects.
(Collection, object are undefined terms.)

The order of elements is not important. I.e., the sets $\{1,2\}$ and $\{2,1\}$ are considered the same set.
Each element is included only once. So, if we write $\{a,b\}$ or $\{a,a,b\}$ they both denote the same set and the element a is considered to be present in the set only once. (repeating an element in the description of a set may be useful sometimes).

Examples of sets:

- $\{1,2,3\}$.

  A simple finite set, obtained by enumerating all the elements.
- $\emptyset$, the *empty set*.

  A special case of a finite set, with *no* elements.
- $\{\emptyset\}$.

  This funny set is different from $\emptyset$.
  It has one element, which is a set.

- $\{1, \{1, 2\}\}$.

  This one contains two elements, one of which is a set itself.
  The set happens to contain the other element 1, but that does not contradict the condition that an element is included only once.

- $\{n \in \mathbb{N} \mid n \text{ is even }\}$.

  An infinite set, specified by giving a universal set (domain of discourse) and a property.
  In general, given a domain of discourse D and a predicate P over D, $\{x \in D \mid P(x)\}$ denotes the set of all x in D that satisfy predicate P.

We write that $a \in A$ ("a is in A" or "a is an element of A") if a is an object in the set A.

The empty set satisfies the proposition $(\forall x) x \notin \emptyset$.

Sets can be defined by axioms and deduction rules, giving a very formal *axiomatic set theory*.
But usually, people reason about sets less formally, using *naive set theory*.
Either way, set theory can be used as the basis for defining almost everything else in mathematics and is therefore regarded as the most fundamental part of math.

E.g., natural numbers can be defined as certain kinds of sets (!). For example we can represent the number 0 by the empty set $\emptyset$, number 1 by the set $\{\emptyset\}$, number 2 by $\{\emptyset, \{\emptyset\}\}$, and so on. In general, if S_n is the set representing the number n, the successor of n can be represented by the set $S_{n+1} = \{\emptyset\} \cup \{\{x\} \mid x \in S_n\}$. Then the Peano axioms about natural numbers can be *proved as theorems* of set theory, rather than being assumed.

Some typical axioms of axiomatic set theory:
(Seem ridiculously obvious but aren't)

- The *axiom of extensionality* says that two sets are equal iff they have the same elements.

  E.g., the two sets $\{n \in \mathbb{N} \mid 2 \text{ divides } n\}$ and $\{2n \mid n \in \mathbb{N}\}$
  are defined in different ways.
  But they are "equal" as sets because they have exactly the same elements.

- The *axiom of comprehension* says that we can form a new set by including all objects of a given set that share a common property.

  For example, $\{x \in \mathbb{R} \mid P(x)\}$ where $P(x) =$ "x is rational" denotes the set of all rational numbers.
  The axioms says that this is a set.

The set building notation that we used above has given the universe of discourse explicitly.
Sometimes, when this seems obvious, we will leave this out.
Just like what we did for predicate quantifiers.
E.g., we might write: $\{n \mid n \text{ is even }\}$ if the universe $\mathbb{N}$ seems clear.
(Could be confusing – the universe could alternatively be the integers, positive and negative.)

Definition 9.1 We say that $A \subseteq B$ ("A is a subset of B" or "A is contained in B") if every element of A is also an element of B (that is, $(\forall x)(x \in A \implies x \in B)$).

Definition 9.2 We say that A is a *proper* subset of B ($A \subset B$) if $A \subseteq B$ and $A \neq B$.

Note $(\forall X)\emptyset \subseteq X$, i.e. the empty set is contained in any other set.

Q: Can we have both $A \in B$ and $A \subseteq B$?

A: Yes, consider $\{1, \{1\}\}$.

Theorem 9.3 $A = B$ if and only if $A \subseteq B$ and $B \subseteq A$.

Proof by axiom of extensionality. Ignore here.

This observation is very important as it provides a simple way to prove that 2 sets are equal.

Example 1 Prove that the two sets $\{n \in \mathbb{N} \mid 2 \text{ divides } n\}$ and $\{2n \mid n \in \mathbb{N}\}$ are equal.

Must show containment in both ways.
$\subseteq$:
We assume $x \in \mathbb{N}$ and 2 divides n, and show that there is some $y \in \mathbb{N}$ such that $x = 2y$.
This is true by definition of divisibility.
$\supseteq$:
Assume $x = 2y$. Then x is divisible by 2, again by definition of divisibility.

Example 2 Slightly more interestingly:

The set of evens ≥ 2, i.e., $\{2n \mid n \in \mathbb{N}, n \geq 1\}$,
is equal to $\{x \in \mathbb{N} \mid \exists y, z \text{ odd }, x = y + z\}$.
$\subseteq$:
If $x = 2n$, then write $x = 1 + (x - 1)$, sum of two odds.

$\supseteq$:
If $x = y + z$, where both y and z are odd, then write $y = 2a + 1$, $z = 2b + 1$.
Then $x = 2(a + b + 1)$, which is even.

The size of a set A (the number of elements it contains) is written $|A|$.
For example, the set $\{a, b, c\}$ has size $|\{a, b, c\}| = 3$. The empty set has size $|\emptyset| = 0$.
The size of a finite set is a natural number.
If A is an infinite set, we write $|A| = \infty$.

Notice that for any two finite sets A, B, we have $|A \cup B| \leq |A| + |B|$ and equality holds if and only if $A \cap B = \emptyset$. If we allow A and B to be infinite sets, we still have $|A \cup B| \leq |A| + |B|$ if we define $x + \infty = \infty + x = \infty$ and $x \leq \infty$ for all $x \in \mathbb{N}$ and $x = \infty$.

9.2 Russell's paradox

We might think that set theory is obvious, but this is not the case.
Consider the set $S = \{x \mid x \text{ is a set}\}$.
This seems like a perfectly reasonable set (well, maybe a bit strange—note $S \in S$).
If our set theory allows for such an entity, we get into trouble as follows.

Consider the set $Q = \{x \in S \mid x \text{ is a set and } x \notin x\}$.

If S is a set, then so is Q by standard axioms like comprehension.
Note $Q \subseteq S$.

Then some sets are in Q and some aren't.

Example of something in Q:
The set $A = \{1, 2, 3\}$ is not an element of itself, so $A \in Q$.

Example of something that is not in Q:
The set $P = \{$ all infinite sets$\}$ is a set (by comprehension from S).
Therefore, $P \in P$ as the set P itself is infinite.
So $P \notin Q$.

The troublesome question is: is $Q \in Q$?

1. If $Q \in Q$ then $Q \notin Q$, a contradiction.
2. If $Q \notin Q$ then $Q \in Q$ also a contradiction.

This is clearly a paradox.
In fact, this paradox, at the time it was discovered by Russell ($\sim$ 1900), was a blow to the

then current attempts to formalize all of mathematics axiomatically.

Mathematicians had to work very hard to find a consistent set of axioms (note that a consistent set of axioms does not allow this paradox to arise).

The problem with naive set theory is that you can form a set by describing its members, as we did for Q above.
Axiomatic set theory has a specific set of rules for how you can form sets—rules that don't allow the set Q above to be specified, and therefore avoid the paradox.

Fortunately, we'll be working naively...

9.3 Basic set operations

There are several standard operations on sets.
In axiomatic set theory, the claims that these operations yield new sets are axioms.

1. Union
   $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$.
2. Intersection
   $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$.
3. Difference
   $A - B = \{x \mid x \in A \text{ and } x \notin B\}$.
   Also written as $A \setminus B$.
4. Complement
   $A^c = \{x \mid x \notin A\}$.
   Also written as $\bar{A}$.
5. Symmetric difference
   $A \oplus B = A \cup B - A \cap B$.

Sets and operations on sets can be represented using Venn diagrams (see textbook, sections 1.4, 1.5).

Union and intersection can be extended to apply to more than two sets.
Given a finite collection of sets A_i, $i = 1, \ldots, n$,

$$\bigcup_{i=1}^{n} A_i$$

is the set of elements in at least one of the A_i, and

$$\bigcap_{i=1}^{n} A_i$$

is the set of elements in all of the A_i. Both of these make sense if $i = 1$ – reduce to just A_1. In fact, for $i = 0$, default says union of no sets is $\emptyset$ and intersection of no sets is universal set.
This definition of union and intersection of no sets might seem arbitrary, but in fact it can be motivated in several ways. For example, if we want to define the union of a finite family of sets $A_1, A_2, \ldots, A_n$, we can define $\bigcup_{i=1}^{1} A_1 = A_1$ and $\bigcup_{i=1}^{n+1} A_i = (\bigcup_{i=1}^{n} A_i) \cup A_{n+1}$. If we want this recursive definition to work also for base case $n = 0$ we must set $\bigcup \emptyset = \bigcup_{i=1}^{0} A_i = \emptyset$. Analogously, for intersection. The recursive definition of $\bigcap_{i=1}^{n+1} A_i = (\bigcap_{i=1}^{n} A_i) \cap A_{n+1}$ must have base case $\bigcap \emptyset = \bigcap_{i=1}^{0} A_i = U$ if we want $\bigcap\{A_1\} = (\bigcap \emptyset) \cap A_1 = A_1$.

Instead of using explicit index, can just consider a set of sets: If $\mathcal{S}$ is a *set of sets*, then

$$\bigcup \mathcal{S}$$

denotes the union of all the sets in $\mathcal{S}$, in any order.
(Can prove that the order doesn't matter, will do below.)
Likewise,

$$\bigcap \mathcal{S}$$

denotes the intersection of all the sets in $\mathcal{S}$.

Complement is different from all the other operations, because it assumes some universe U from which we extract the things not in A.
All sets in question are supposed to be subsets of U.
Given a universal set U, the complement operation can be expressed as $A^c = U - A$.
The universe U is often implicit, where there seems to be little chance of confusion.

There is a close correspondence between set operations and logical operations: consider sets A, B and C defined by $A = \{x \mid P(x)\}$, $B = \{x \mid Q(x)\}$, $C = \{x \mid R(x)\}$, where P, Q, R are predicates.

- set union vs. predicate or: $A \cup B = C$ if and only if $P(x) \vee Q(x) \equiv R(x)$

- set intersection vs. predicate and: $A \cap B = C$ if and only if $P(x) \wedge Q(x) \equiv R(x)$

- set difference vs. predicate "non implication": $A - B = C$ if and only if $\neg(P(x) \to Q(x)) \equiv R(x)$

- set complement vs. predicate not: $A^c = B$ if and only if $\neg P(x) \equiv Q(x)$

- set symmetric difference vs. predicate exclusive or: $A \oplus B = C$ if and only if $P(x) \oplus Q(x) \equiv R(x)$.

Predicates and sets often form alternative ways of looking at the same notions:
Given predicates p and q, define $P = \{x \mid p(x)\}$,
$Q = \{x \mid q(x)\}$.
Then $P \cup Q = \{x \mid p(x) \vee q(x)\}$.
Etc.

9.4 Power set (set of all subsets)

Another very important operation on sets is that of the *power set* of a given set A.
It is denoted by $\mathcal{P}(A)$ and consists of the set of all subsets of A.
In other words, $\mathcal{P}(A) = \{x \mid x \subseteq A\}$.

In particular, for any set A, $\emptyset \in \mathcal{P}(A)$ and $A \in \mathcal{P}(A)$.
Ex:
If $A = \{1, 2\}$ then $\mathcal{P}(A) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$.

Theorem 9.4 If A has n elements, then $P(A)$ has 2^n elements.

Proof. Ordinary Induction on n. BASE: $n = 0$. Then $P(A)$ has one element, $\emptyset$.
INDUCTIVE: Assume for n, prove for $n + 1$.
Pick one element, a.
The subsets can be divided into two groups – those containing a and those not containing a.
Moreover, the set of subsets of A not containing a is $\mathcal{P}(A - \{a\})$, and the set of subsets of A containing a is the set $\{\{a\} \cup B \mid B \in \mathcal{P}(A - \{a\})$ which has the same number of elements as $\mathcal{P}(A - \{a\})$. By induction hypothesis $\mathcal{P}(A - \{a\})$ has 2^{n-1} elements, so the total is $2^{n-1} + 2^{n-1} = 2^n$. ∎

Q: I did a base case of 0 here. Does the inductive step from 0 to 1 work correctly?

A: Yes, it corresponds to the sets $\{\emptyset\}$ and $\{a\}$.

Because of this cardinality relationship, $\mathcal{P}(A)$ is sometimes denoted by 2^A.

9.5 Set identities

There are a large number of set identities involving relationships among sets built using various operations.
See p. 50 of book.

Here we just list some sample properties, and show how they can be proved.

Theorem 9.5 (Distributivity: Union distributes over intersection)
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Proof. We show the two sets are equal by proving that each is contained in the other.

1. $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$.

   Suppose $x \in A \cup (B \cap C)$.
   Then by definition, $x \in A$ or $x \in (B \cap C)$.
   We argue by cases.

   If $x \in A$, then by definition $x \in A \cup B$ and similarly $x \in A \cup C$.
   It follows by definition of intersection that x is in the intersection of these two sets, which is the RHS.

   On the other hand, if $x \in (B \cap C)$, then $x \in B$ and $x \in C$.
   Then by definition of union, $x \in A \cup B$ and $x \in A \cup C$.
   It again follows that x is in the intersection.

2. $A \cup (B \cap C) \supseteq (A \cup B) \cap (A \cup C)$.

   Suppose $x \in (A \cup B) \cap (A \cup C)$.
   It follows that $x \in A \cup B$ and $x \in A \cup C$.
   We argue by cases.

   If $x \in A$, then certainly $x \in A \cup (B \cap C)$.

   If $x \notin A$, then since $x \in A \cup B$, we know $x \in B$.
   Similarly, $x \in C$.

Therefore, $x \in B \cap C$.
It follows that $x \in A \cup (B \cap C)$.

■

Theorem 9.6 De Morgan's rule: $(A \cup B)^c = A^c \cap B^c$.

Proof. We have to show that $(A \cup B)^c \subseteq A^c \cap B^c$ and that $A^c \cap B^c \subseteq (A \cup B)^c$.

1. $(A \cup B)^c \subseteq A^c \cap B^c$.

   Let $x \in (A \cup B)^c$. Then by definition of complement, $x \notin A \cup B$. It follows (proof by contradiction) that $x \notin A$ and $x \notin B$. Therefore, again by definition of complement, $x \in A^c$ and $x \in B^c$.
   Then by definition of intersection, $x \in A^c \cap B^c$.

2. $A^c \cap B^c \subseteq (A \cup B)^c$.

   Let $x \in A^c \cap B^c$.
   Then x is in A^c and in B^c.
   Therefore, x is in neither A nor B and therefore not in $A \cup B$.
   We conclude that $x \in (A \cup B)^c$.

■

The above two theorems have *dual theorems* produced by swapping $\cap$ and $\cup$.
This is a general phenomenon – look at p. 50, arranged in pairs.
Duality can be applied to any theorem that is true for *all* sets. Each theorem in a pair can be obtained from the other by swapping the operations $\cap$ and $\cup$, and the sets $\emptyset$ and U. Notice that complement operations are not changed.

Theorem 9.7 $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

Theorem 9.8 $(A \cap B)^c = A^c \cup B^c$.

9.6 Cartesian products

Another important set operation is the *Cartesian product.*
For sets A and B define $A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$.
Here, (a, b) is considered an *ordered* pair so that $(a, b) \neq (b, a)$.

Example 3 If $A = \{0, 1\}$ and $B = \{x, y\}$ then $A \times B = \{(0, x), (0, y), (1, x), (1, y)\}$.

Order is important!
In the above example, $(a, 0) \notin A \times B$, though $(a, 0) \in B \times A$.

We can generalize Cartesian products to apply to more than 2 sets and define $A_1 \times \cdots \times A_n = \prod_{i=1}^{n} A_i = \{(a_1, \cdots, a_n) \mid a_i \in A_i \text{ for } 1 \leq i \leq n\}$. Also, we write A^n for $\prod_{i=1}^{n} A$.

Examples:

- $\{0, 1\}^n$ is the set of all n-tuples of 0's and 1's. Or, length n strings of 0's and 1's.
- $\Re^2 = \Re \times \Re$ is the real plane.

10 Relations

A "relation" is a fundamental mathematical notion expressing a relationship between elements of sets.
It's an abstract notion useful in practice for modelling many different sorts of relationships.
It's the basis of the *relational data base* model, the standard data model for practical data processing systems.

10.1 Definitions

A *binary relation from* a set A *to* a set B is a subset $R \subseteq A \times B$.
A binary relation *on* a set A is a subset $R \subseteq A^2$ (I.e., a relation from A to A).
We often write $a \sim_R b$ or aRb instead of $(a, b) \in R$.

We can also define a *ternary* relation on A as a subset $R \subseteq A^3$ or, in general, an n-ary relation as a subset $R \subseteq A^n$, or $R \subseteq A_1 \times A_2 \times \cdots \times A_n$ if the sets A_i are different.

Relations are used to model many different things:

1. The relation "is taking class" as a subset of {students at MIT} $\times$ {classes at MIT}. A relation from students to classes.
2. The relation "is living in the same room" as a subset of {students at MIT} $\times$ {students at MIT}. A relation on students.
3. The relation "can drive from first to second city". (Not necessarily directly – just some way, on some roads.)

4. Relation on computers, "are connected (directly) by a wire"
5. "meet one another on a given day"
6. "likes"
7. Let $A = \mathbb{N}$ and define $a \sim_R b$ iff $a \leq b$.
8. Let $A = \mathcal{P}(\mathbb{N})$ and define $a \sim_R b$ iff $|a \cap b|$ is finite.
9. Let $A = \Re^2$ and define $a \sim_R b$ iff $d(a, b) = 1$.
10. Let $A = \mathcal{P}(\{1, \cdots, n\})$ and let $a \sim_R b$ if $a \subseteq b$.

10.2 Properties of Relations

There are several standard properties that are satisfied by (some) relations.
These occur commonly enough that they are worth identifying.
Sometimes interesting things can be deduced just from the abstract properties.

A binary relation R on A is:

1. *reflexive* if for every $a \in A$, $a \sim_R a$.
2. *symmetric* if for every $a, b \in A$, $a \sim_R b$ implies $b \sim_R a$.
3. *antisymmetric* if for every $a, b \in A$, $a \sim_R b$ and $b \sim_R a$ implies $a = b$.
4. *transitive* if for every $a, b, c \in A$, $a \sim_R b$ and $b \sim_R c$ implies $a \sim_R c$.
5. *asymmetric* if for every $a, b \in A$, $a \sim_R b$ implies $\neg(b \sim_R a)$.

The difference between antisymmetric and asymmetric relations, is that antisymmetric relations may contain pairs (a, a), i.e., elements can be in relations with themselves, while in an asymmetric relation this is not allowed. Clearly, any asymmetric relation is also antisymmetric, but not vice versa.

Among our examples:

- Relation 2 is reflexive, symmetric, transitive.
- Relation 3 is reflexive, transitive.
  Not necessarily symmetric, since roads could be one-way, but in actuality... But definitely not antisymmetric.
- Relation 4 is symmetric but not transitive.
  Whether it is reflexive is open to interpretation.
- Relation 5 likewise.

- Relation 6 is (unfortunately) not symmetric. Not antisymmetric. Not transitive. Not even reflexive!
- Relation 7 is reflexive, antisymmetric, transitive.
- Relation 8 is not reflexive. It is symmetric. Not transitive.
- Relation 9 is only symmetric.
- Relation 10 is reflexive, antisymmetric and transitive.

10.3 Representation

There are different ways of representing relations, e.g., for handling in a computer program.

We can describe by properties, as we did above.
That's about all we can do, for infinite sets.
But for finite sets, usually we use some method that explicitly enumerates all the elements of the relation.

Some alternatives: Lists, Matrices, Graphs.

10.4 Lists

To represent a finite relation from set A to set B, just list all the pairs.

Ex: Consider for example the relation from $A = \{0, 1, 2, 3\}$ to $\{a, b, c\}$ defined by the list $\{(0, a), (0, c), (1, c), (2, b), (1, a)\}$.

Ex: The divisibility relation on natural numbers $\{1, \cdots, 12\}$ is represented by the list:
$\{(1, 1), (1, 2), \ldots, (1, 12), (2, 2), (2, 4), \ldots, (2, 12),$
$(3, 3), (3, 6), (3, 9), (3, 12), (4, 4), (4, 8), (4, 12), (5, 5), (6, 6),$
$(6, 12), (7, 7), (8, 8), (9, 9), (10, 10), (11, 11), (12, 12)\}$.

Reflexivity in this representation: Contains all pairs (a, a).
Symmetry: If contains (a, b) then contains (b, a).
Transitivity: If contains (a, b) and (b, c) then contains (a, c).

10.5 Boolean matrices

Used for finite sets A, B
Rows for elements of A, columns for B.
1 in position if the pair is in the relation, 0 otherwise.

Ex: The relation from $\{0, 1, 2, 3\}$ to $\{a, b, c\}$ of our first example is represented by the matrix

	a	b	c
0	1	0	1
1	1	0	1
2	0	1	0
3	0	0	0

Ex: The divisibility relation over $\{1, 2, \ldots, 12\}$ is represented by the matrix

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	1	0	1	0	1	0	1	0	1	0	1
3	0	0	1	0	0	1	0	0	1	0	0	1
4	0	0	0	1	0	0	0	1	0	0	0	1
5	0	0	0	0	1	0	0	0	0	1	0	0
6	0	0	0	0	0	1	0	0	0	0	0	1
7	0	0	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0
12	0	0	0	0	0	0	0	0	0	0	0	1

Reflexivity in this representation: major diagonal is all 1
Symmetry: the matrix is symmetric around major diagonal
Transitivity: ...

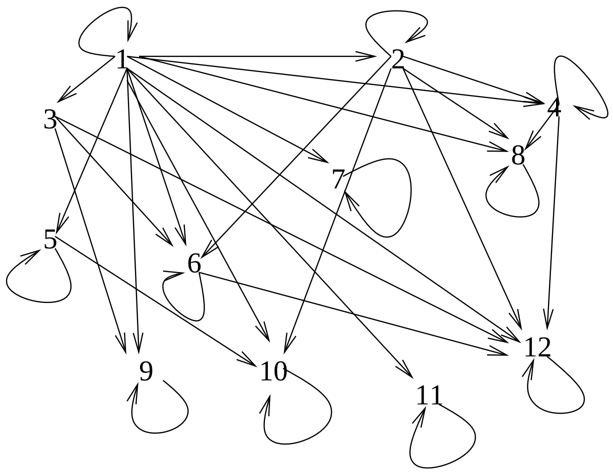
10.6 Digraphs

Digraphs are used for relations where $A = B$, i.e., for relations on a finite set A. To represent a relation as a digraph we draw a dot (vertex) for each element of A, and draw an arrow

from first element to second element of each pair in the relation.
The digraph may contain self-loops, i.e. arrows from a dot to itself, associated to the elements a such that (a, a) is in the relation.

The relation from $\{0, 1, 2, 3\}$ to $\{a, b, c\}$ of the previous examples cannot be represented as a digraph because $A \neq B$.

The divisibility relation over $\{1, 2, \ldots, 12\}$ is represented by the digraph



Reflexivity: All nodes have self-loops.
Symmetry: all edges are bidirectional.
Transitivity: Short-circuits, for any sequence of consecutive arrows, there is a single arrow from the first to the last node.

10.7 Composition

In recitation we defined an important operation to combine relations: composition. We then expressed the transitive property of relation in term of the composition operation. We defined an important class of relations, equivalence relations, and gave the example of equivalence modulo an integer.

Notice the difference between properties of relations and operations on relations. A property, such as symmetry or transitivity, is something a relation may or may not satisfy. An operation on relation, such as relation composition we are going to define, is a way to combine one or more relations to build some other relation.

The *composition* of relations $R_1 \subseteq A \times B$ and $R_2 \subseteq B \times C$ is the relation $R_2 \circ R_1 = \{(a, c) \mid (\exists b)((a, b) \in R_1) \wedge ((b, c) \in R_2)\}$.

Notice that $R_1 \circ R_2$ and $R_2 \circ R_1$ are different. This is the notation used in the text book. Notice that some people prefer to write $R_1 \circ R_2$ for the composition of R_1 and R_2, which may seem more natural, (this is the notation that was used in recitation), but from now on we will stick with the book.
Examples:
Composition of parent-of relation with itself gives grandparent-of.
Composition of child-of relation with parent-of relation gives sibling-of relation.

Composition of relations is equivalent to matrix multiplication, with $+$ replaced by $\vee$ (Boolean or) and $*$ replaced by $\wedge$.

Ex:
Let R_1 be relation from first example above:

	a	b	c
0	1	0	1
1	1	0	1
2	0	1	0
3	0	0	0

Let R_2 be relation from $\{a, b, c\}$ to $\{d, e, f\}$ given by:

	d	e	f
a	1	1	1
b	0	1	0
c	0	0	1

Then $R_2 \circ R_1 = \{(0, d), ...\}$, that is,

	d	e	f
0	1	1	1
1	1	1	1
2	0	1	0
3	0	0	0

Same as matrix multiply, using Boolean ops and and or.

Relational composition is associative.
Just like matrix multiplication.

Lemma 10.1 $R_1 \circ (R_2 \circ R_3) = (R_1 \circ R_2) \circ R_3$

The composition $R \circ R$ of R with itself is written R^2.
Similarly R^n denotes R composed with itself n times.
Recursive definition:
$R_1 = R$, $R^n = R \circ R^{n-1}$.
Actually, can define $R_0 = \{(a, a) \mid a \in R\}$ as base case.
We'll just use 1 as base case, as book does.

Could just as well have defined as $R^{n-1} \circ R$:

Theorem 10.2 For $n \geq 2$, $R^n = R^{n-1} \circ R$

Proof. Prove by induction on n.
Base case $n = 2$ trivial.
Inductive step assumes $n \geq 3$.
$R^n = R \circ R^{n-1}$ by definition.
By inductive hypothesis, this is equal to $R \circ (R^{n-2} \circ R)$.
By associativity, Lemma 10.1, this is equal to $(R \circ R^{n-2}) \circ R$.
By definition of R^{n-1}, this is equal to $R^{n-1} \circ R$.
As needed. ■

The following gives simple conditions for transitivity.

Theorem 10.3 R is transitive if and only if $R^2 \subseteq R$ if and only if $R^n \subseteq R$ for all positive integers n.

Proof. Obviously, the third implies the second.

Show second implies the first:
To show transitivity, assume $(a, b) \in R$ and $(b, c) \in R$.
Then $(a, c) \in R^2$.
Second says $(a, c) \in R$, which implies transitivity.

Show first implies third.
Use induction on n.
BASE: $n = 1$, $R \subseteq R$, obvious.
INDUCTIVE STEP:
Suppose $R^n \subseteq R$, show $R^{n+1} \subseteq R$.
Say $(a, b) \in R^{n+1}$.
Then for some c, $(a, c) \in R^n$ and $(c, b) \in R$ for some c.
(Uses theorem above.)

Then $(a, c) \in R$ by induction.
Then by transitivity, $(a, b) \in R$. ■

In the above theorem we wanted to prove the equivalence of three propositions $A \leftrightarrow B \leftrightarrow C$. We proved it by showing that the three implications $A \to B$, $B \to C$ and $C \to A$ holds true. Why is this enough to prove equivalence of A, B and C? Notice that from $B \to C \to A$ we can deduce by transitivity $B \to A$, which together with $A \to B$ proves that $A \leftrightarrow B$. Equivalences $B \leftrightarrow C$ and $C \leftrightarrow A$ can be derived similarly.

11 Equivalence relations and partitions

Now we consider an important special type of relation, called an *equivalence relation.*
We give an important special case – modular arithmetic.
And an application of this to RSA.

11.1 Definitions

An *equivalence relation* is a relation that is reflexive, symmetric and transitive.
For example, the "roommates" relation is an equivalence relation.
Trivial example: "=" relation on natural numbers.
Hallmark of equivalence relations is the word "same".

Equivalence relations partition the universe into subsets in a natural way:

A *partition* of a set A is a collection of subsets $\{A_1, \cdots, A_k\}$ such that any two of them are disjoint (for any $i \neq j$, $A_i \cap A_j = \emptyset$) and such that their union is A.
That is, every element of A is in exactly one subset.

Fix an equivalence relation R on A.
For an element $a \in A$, let $[a]$ denote the set $\{b \in A \mid a \sim_R b\}$.
We call this set the *equivalence class of a under R.*
We claim that the sets $[a]$ for $a \in A$ constitute a partition of A.

Lemma 11.1 For every $a, b \in A$, either $[a] = [b]$ or $[a] \cap [b] = \emptyset$.

Proof. Fix $a, b \in A$.
If $[a] = [b]$ then we are done, so suppose not.
Let c be any element that is in one of the sets but not the other.
Without loss of generality we can assume that $c \in [b] - [a]$ (we know that either $c \in [b] - [a]$

or $c \in [a] - [b]$. In the latter case we can simply swap a and b and reduce to the first case.).
Similarly, if $[a] \cap [b] = \emptyset$ then we are done, so suppose not.
Let d be any element in $d \in [a] \cap [b]$.

We will get a contradiction by showing that $a \sim_R c$ and therefore that $c \in [a]$.

First, $a \sim_R d$ because $d \in [a]$ (note that $d = a$ is a possibility but this is ok because R is reflexive).
Second, $d \sim_R b$ and $b \sim_R c$ because both $c, d \in [b]$ and R is symmetric.
This implies, by transitivity, that $d \sim_R c$.
Finally, by transitivity, $a \sim_R c$ because $a \sim_R d$ and $d \sim_R c$. ■

Conversely, any partition $\{A_1, \cdots, A_k\}$ of A defines an equivalence relation by letting $a \sim_R b$ iff a and b are in the same A_i.
(Note "same".)
This is an equivalence relation:

- Reflexivity: for all a we know $a \in A_i$ for some i, by definition of partition. Clearly a and a are in the same A_i, and $a \sim_R b$.
- Symmetry: Assume $a \sim_R b$, that is a and b are in the same A_i. Also b and a are in the same A_i and therefore $b \sim_R a$.
- Transitivity: Assume $a \sim_R b$ and $b \sim_R c$. By definition of $\sim_R$, a and b are in the same A_i for some i, and b and c are in the same A_j for some j. But by definition of partition b cannot be in two different A_i's. So, it must be $A_i = A_j$ and a and c are in the same A_i, proving $a \sim_R c$.

Therefore, we can look at partitions and at equivalence relations as the same thing.

11.2 Integers modulo m

In Lecture 4, we considered the modulo operation, for a particular positive integer m, as a function.
Another way to think of it is as a definition of an equivalence relation on integers (positive, negative and 0):

Definition 11.2 If a and b are integers, then we say $a = b \pmod{m}$ if $m \mid (a - b)$.

An equivalent formulation says that $a = b + km$ for some integer k.

Theorem 11.3 Equality modulo m is an equivalence relation.

Proof. We need to show the relation is reflexive, symmetric, and transitive.

- Reflexive: Clearly $m \mid a - a = 0$, so $a = a \pmod{m}$.
- Symmetric: If $a = b + km$ then $b = a + (-k)m$.
- Transitive: If $a = b \pmod{m}$ and $b = c \pmod{m}$, this means that $m \mid (a - b)$ and $m \mid (b - c)$. It follows from a theorem in the previous lecture that $m \mid (a - b) + (b - c) = a - c$.

■

The equivalence class of a is simply the set $[a] = \{b \in \mathbb{Z} \mid a = b \pmod{m}\}$, or $\{km + a \mid k \in \mathbb{Z}\}$.

It turns out that we can extend a lot of standard arithmetic to work modulo m.
In fact, we can define notions of sum and product for the equivalence classes mod m.
For example, we define $[a] + [b]$, given two equivalence classes mod m, to be the equivalence class $[a + b]$.
This is not as obvious as it seems: notice that the result is given in terms of a and b, two selected representative from the equivalence classes, but that it is supposed to apply to the equivalence classes themselves.
To prove that this works, we have to show that it doesn't matter which representatives of the equivalence classes we choose:

Lemma 11.4 If $a = x \pmod{m}$ and $b = y \pmod{m}$ then $(a + b) = (x + y) \pmod{m}$.

Proof. $m \mid (a - x)$ and $m \mid (b - y)$ so $m \mid ((a - x) + (b - y)) = (a + b) - (x + y)$, ■

It follows that if we are interested only the result of the addition modulo m, which is the case, for example, in the RSA cryptosystem, that we can at any time replace a given number with a different number equivalent to it, without changing the value (equivalence class) of the final answer.

The same fact can be proven for multiplication.
This is the basis of the fast exponentiation algorithm presented in the next subsection.

11.3 Fast Exponentiation

Let's say that we want to compute $a^n \bmod b$.
Here's the crude, obvious way to do it:

1. Compute a^n.
2. Divide by b and take the remainder.

There are actually two big problems with this.
First of all, a^n could be a gigantic number.
And secondly, we're using n multiplications, which could take a long time if n is large.
Let's correct the first problem first.

1. Initialize $p = 1$.
2. Repeat n times:
   (i) Set $p \leftarrow ap$.
   (ii) Set $p \leftarrow p \bmod b$.
3. Return p.

That is, we are reducing mod b after every multiplication.
To see why this works, we have to use the fact that $(x \cdot y) \bmod b = ((x \bmod b) \cdot y) \bmod b$.
Let's prove the correctness of the algorithm more formally.

Notice that we are using the expression $x \bmod b$ to denote the result of computing the remainder of x when divided by b. In particular $x \bmod b$ is a number between 0 and $b - 1$.

Clearly the algorithm always terminates as it executes the loop only a finite number of times, namely n. We prove by induction on the number of iterations that at the beginning of the i^{th} iteration we have $p = a^i \bmod b$.
BASE CASE: At iteration 0 we have $p = 1 = a^0 \bmod b$.
INDUCTIVE STEP: Assume that at iteration i we have $p = a^i \bmod b$. Executing the loop one more time we get $p' \leftarrow (ap) \bmod b$. Using the induction hypothesis this is equal to $p' = (a \cdot (a^i \bmod b)) \bmod b = (a^{i+1}) \bmod b$.
After executing the loop n times we will have $p = a^n \bmod b$ and the algorithm terminates with the right answer.

Now for the second problem:
To reduce the number of multiplications, we try to use squaring whenever possible.

Algorithm `fast-exp(a,n,b)`

1. If $n = 1$ then return $a \bmod b$.
2. Else if n is odd return $a*$ `fast-exp(a,n-1,b)` mod$b$.
3. Else (n is even), set $p \leftarrow$ `fast-exp(a,n/2,b)` and return $p^2 \bmod b$.

We will prove three properties about the `fast-exp` algorithm.

1. Termination: for any a, b, n algorithm `fast-exp(a,n,b)` terminates.
2. Correctness: when `fast-exp(a,n,b)` terminates it returns the right answer, that is `fast-exp(a,n,b)`$= a^n \bmod b$.
3. Efficiency: for any a, b, n algorithm `fast-exp(a,n,b)` terminates after at most $2\log_2 n$ recursive calls.

We will prove all three properties by strong induction on the value of n passed to the algorithm.
BASE CASE: If $n = 1$ the algorithm terminates immediately with result $a = a^1 \bmod b$. The number of recursive calls is $0 = 2\log_2 1$.
INDUCTIVE STEP: Assume that properties 1,2 and 3 are satisfied whenever the algorithm is called with values of n smaller then some $m > 1$. We will prove that `fast-exp(a,m,b)` terminates with result $a^m \bmod b$ after at most $2\log_2 m$ recursive calls. Consider the execution of `fast-exp(a,m,b)`. The test at step 1 fails because $m > 1$ by assumption. We distinguish two cases:

- If m is even, the test at step 2 fails, the test at step 3 succeeds and `fast-exp(a,m/2,b)` is called recursively. Since $m/2 < m$, we know this procedure call terminates with result $p \leftarrow a^{m/2} \bmod b$. Therefore, `fast-exp(a,m,b)` terminates with result $p^2 \bmod b = (a^{m/2} \bmod b)^2 \bmod b = a^m \bmod b$. The number of recursive calls is 1 plus the number of recursive calls made by `fast-exp(a,m/2,b)`. This is by induction at most $2\log_2(m/2) = 2\log_2 m - 2$, so the total number of recursive calls is at most $1 + 2\log_2 m - 2 < 2\log_2 m$.
- If m is odd, the test at step 2 succeeds and `fast-exp(a,m-1,b)` is called recursively. Since $m - 1 < m$, we know this procedure call terminates with result $a^{m-1} \bmod b$. Therefore, also `fast-exp(a,m,b)` terminates with result $a \cdot (a^{m-1} \bmod b) \bmod b = a^m \bmod b$. To bound the number of recursive calls we need to examine the execution of the algorithm more in depth. The number of recursive calls is 1 plus the number of recursive calls made by `fast-exp(a,m-1,b)`. Now, if m is odd then $m-1$ is even and `fast-exp(a,m-1,b)` will call recursively `fast-exp(a,(m-1)/2,b)` which by inductive hypothesis will make at most $2\log_2((m-1)/2)$ recursive calls. So, the total number of recursive calls made by `fast-exp(a,m-1,b)` is at most $2 + 2\log_2((m-1)/2) = 2\log_2(m-1) < 2\log_2 m$.

This concludes the proof of correctness of the algorithm.

Since the algorithm executes only one multiplication for each recursive call, it uses at most $2\log n$ multiplications to compute $a^n \bmod b$ (a huge savings over n multiplications even for moderate values of n).