

Recap of the previous lecture

- What is this course about?
 - Automata
Some simple models of computation
 - Turing Machines
An accurate model of a general purpose computer
 - Computability
What kinds of problem are savable by computers, given unlimited time and space?
 - Complexity
What kinds of problem can be solved by computers, when taking practical limitations (limited time and space) into account?

Recap of the previous lecture

- Some proof techniques
 - Coloring
 - Constructive proofs
 - Non-constructive proofs
 - Counting (pigeonhole principle)
 - Contradiction
 - Induction

This Lecture?

- A little bit of *Logic*
- A quick overview of set theory

An overview of Logic

- Propositional Logic
 - Consider the values True & False, or simply T & F.
 - We define the following operators which take one (in case of unary operators) or two (in case of binary operators) T/F value(s) as input and generate a T/F value as output:
 - Unary operator: \sim
 - Binary operators: $\vee, \wedge, \oplus, \Rightarrow, \Leftrightarrow, \equiv$

An overview of Logic

- Truth table: List all possible inputs and specify the output for each case.

P	$\sim P$
F	T
T	F

Logical operators

P	Q	$P \vee Q$	$P \wedge Q$	$P \oplus Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$	$P \equiv Q$
F	F	F	F	F	T	T	T
F	T	T	F	T	T	F	F
T	F	T	F	T	F	F	F
T	T	T	T	F	T	T	T

Propositional Logic

- Assume that P & Q are variables which can take T or F as their value (called Boolean variables). Using truth-tables, we get:
 - Double negation: $\sim(\sim P) \equiv P$
 - Commutativity of \vee and \wedge : $P \vee Q \equiv Q \vee P$
 - Associativity of \vee and \wedge :
$$P \vee (Q \vee R) \equiv (P \vee Q) \vee R$$
 - Distributivity of \vee over \wedge and vice versa:
$$P \vee (Q \wedge R) \equiv (P \wedge Q) \vee (P \wedge R)$$

Propositional Logic

– $P \vee P \equiv P \wedge P \equiv P \vee F \equiv P \wedge T \equiv P$

– $P \wedge F \equiv \sim(P \vee T) \equiv F$

– $P \Rightarrow Q \equiv (\sim P \vee Q)$

– $P \Leftrightarrow Q \equiv (P \Rightarrow Q) \wedge (Q \Rightarrow P)$

– Demorgan's law:

$$\sim(P \vee Q) \equiv \sim P \wedge \sim Q, \sim(P \wedge Q) \equiv \sim P \vee \sim Q$$

– Example: $\sim(P \Rightarrow Q) \equiv \sim(\sim P \vee Q) \equiv P \wedge \sim Q$

Propositional Logic

- Using the above rules, prove

a) $(\sim P) \Rightarrow F \equiv P$

b) $P \Rightarrow Q \equiv \sim Q \Rightarrow \sim P$

Propositional Logic

- Using the above rules, prove

a) $(\sim P) \Rightarrow F \equiv P$

b) $P \Rightarrow Q \equiv \sim Q \Rightarrow \sim P$

- Proof:

a) $(\sim P) \Rightarrow F \equiv P \vee F \equiv P$

b) $P \Rightarrow Q \equiv \sim P \vee Q \equiv Q \vee \sim P \equiv \sim(\sim Q) \vee (\sim P) \equiv \sim Q \Rightarrow \sim P$

Propositional Logic

- Using the above rules, prove

a) $(\sim P) \Rightarrow F \equiv P$ b) $P \Rightarrow Q \equiv \sim Q \Rightarrow \sim P$

- a) Now we can formally explain why proof by contradiction works:

- Version 1: To prove P , we prove $(\sim P) \Rightarrow F$.
- Version 2: To prove $P \Rightarrow Q$, we show $(\sim Q) \Rightarrow (\sim P)$.

First order logic

- Now we reify predicates with variables ranged over some set(s) using two kind of quantifiers:

- The universal quantifier \forall :

$$\forall x \in \mathbb{R}, x^2 \geq 0$$

- The existential quantifier \exists

$$\exists x \in \mathbb{N}, (x > 1) \wedge (x | 5)$$

($x|y$ means: x is a divisor of y)

First order logic

- A formula with a universal quantifier is assumed to be true, iff replacing the variable with every element of its range results in a true proposition.

$$\forall x \in \mathbb{R}, x^2 \geq 0$$

- A formula with an existential quantifier is true, iff replacing the variable by at least one element in the range results in a true proposition.

$$\exists x \in \mathbb{R}, x^2 = -1$$

First order logic

- Note! The “no” quantifier, which we use in natural language, can be formulated by a universal quantifier as follows:

$$\nexists x \in S P(x) \equiv \forall x \in S \sim P(x)$$

Negation in FOL

- Negation in FOL:

$$\sim (\forall x \in S, P(x)) \equiv \exists x \in S, \sim P(x)$$

$$\sim (\exists x \in S, P(x)) \equiv \forall x \in S, \sim P(x)$$

- Example: Prove that

$$\sim (\forall x \in S, P(x) \Rightarrow Q(x)) \equiv \exists x \in S, P(x) \wedge \sim Q(x)$$

Why counter example works?

Prove or disprove: *Given an arbitrary tree T , T has at least two vertices of degree 1.*

- Rephrasing the above statement:

\forall graph G , (G is a tree $\Rightarrow G$ has at least two vertices of degree 1)

- To prove that this statement is false, it is enough to show that its negation is true.

- What is the negation of the above statement?

\exists graph G , (G is a tree) \wedge (G does not have at least two vertices of degree 1)

- Can you give a counter example?

Quantifiers are tricky!

- Watch for the order of the quantifiers.

True or False?

① $\forall x \in R, \exists y \in R x+y=0$

② $\exists x \in R, \forall y \in R x+y=0$

③ $\exists x \in R, \forall y \in R xy=0$

④ $\forall x \in R, \exists y \in R xy=0$

Which one is stronger 3 or 4?

Quantifiers are tricky!

- Pulling out quantifiers from a logical formula:

$$[(\forall x \in S_1, P(x)) \wedge (\exists y \in S_2, \sim Q(y))] \equiv$$
$$[\forall x \in S_1, \exists y \in S_2, P(x) \wedge \sim Q(y)]$$

$$[(\forall x \in S_1, P(x)) \wedge \sim (\exists y \in S_2, Q(y))] \equiv$$
$$[\forall x \in S_1, \forall y \in S_2, P(x) \wedge \sim Q(y)]$$

Quantifiers are tricky!

- Be careful when pulling out a quantifier from the antecedent of a conditional.

$$[(\forall x \in S_1, P(x)) \Rightarrow (\exists y \in S_2, \sim Q(y))] \equiv \\ \exists x \in S_1, \exists y \in S_2, \sim P(x) \vee \sim Q(y)$$

Quantifiers are tricky!

- State the following statements in FOL.
 - *Every sufficiently large number in S is a perfect square. (S is some infinite subset of natural numbers)*
$$\exists c \in \mathbb{N} \forall n \in S \exists k \in \mathbb{N}, (n \geq c \Rightarrow n = k^2)$$
 - *The only positive even prime number is 2.*
(You are only allowed to use \leq , $=$ and $|$ as predicates)

Elementary set theory

- A *set* is a group of *distinct* objects. Each of these objects are called a *member* or an *element* of the set.
- We often represent finite sets by listing their members between the symbols { and } e.g.
 $\{0, 1\}$, $\{a, b, c\}$, $\{table, seat, carpet\}$, etc.

Elementary set theory

- Infinite sets sometimes informally are shown as $A = \{1, 3, 5, \dots\}$, $B = \{0, 1, 4, 9, 16, \dots\}$, etc.
 - The above sets can be formally defined as
 - $A = \{2k-1 \mid k \in \mathbb{N}\}$
 - $B = \{k^2 \mid k \in \mathbb{Z}^{\geq 0}\}$

Where \mathbb{N} is the set of all natural numbers and \mathbb{Z} is the set of all integers ($\mathbb{Z}^{\geq 0}$ means the set of all non-negative integers)

Elementary set theory

- Membership is shown using the symbol \in , for example, $a \in \{a, b, c\}$.
- Given two sets A & B :
 - A is a *subset* of B (shown as $A \subseteq B$) iff every member of A is also a member of B .
 - A *equals* B (shown as $A=B$) iff $(A \subseteq B)$ and $(B \subseteq A)$.
 - A is a *proper* subset of B iff $(A \subseteq B)$ and $(A \neq B)$.

Elementary set theory

- If A is a *subset* of B , B is called a *superset* of A .
- We define the *powerset* of A , shown as $P(A)$, or 2^A as the set of all subsets of A .
- Given a finite set A , $|A|$ is defined as the number of elements in A .
- It is easy to show that $|2^A| = 2^{|A|}$.

Basic set operations

- *Union:* $A \cup B = \{x \mid x \in A \vee x \in B\}$
- *Intersection:* $A \cap B = \{x \mid x \in A \wedge x \in B\}$
- *Subtraction:* $A \setminus B = \{x \mid x \in A \wedge x \notin B\}$
- *Symmetric difference:* $A \Delta B = (A \setminus B) \cup (B \setminus A)$

Cartesian Product

- A *Cartesian product* of A & B is defines as:

$$A \times B =_{def} \{ (a, b) \mid a \in A \wedge b \in B \}$$

in which (a, b) is called an ordered pair.

- The Cartesian product of n sets is defined as:

$$A_1 \times A_2 \times \dots \times A_n = \{ (a_1, a_2, \dots, a_n) \mid a_1 \in A_1, a_2 \in A_2, \dots, a_n \in A_n \}$$

in which (a_1, a_2, \dots, a_n) is called an n -tuple.

if $A_1 = A_2 = \dots = A_n = A$ then the product is shown as A^n

Relations

- Any subset of $A \times B$ is called a *(binary) relation* from A to B .
- Any subset of $A \times A$ is called a relation *over* A
- $(a, b) \in R$ sometimes is shown as aRb .

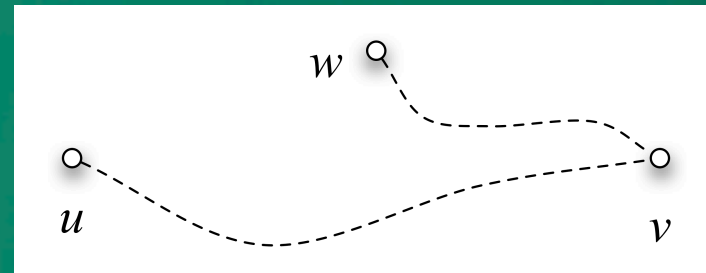
- *Example: $A = \{1, 2, 3\}$, $B = \{a, b\}$*
- $A \times B = \{(1, a), (2, a), (3, a), (1, b), (2, b), (3, b)\}$
- \emptyset , $\{(1, a), (1, b)\}$, and $\{(1, a), (2, b), (3, a)\}$ are relations from A to B .

Properties of relations

- A relation R over A is called
 - *Reflexive* iff $\forall a \text{ in } A, (a, a) \in R$
 - *Symmetric* iff $\forall a, b \in A, (a, b) \in R \Leftrightarrow (b, a) \in R$
 - *Transitive* iff $\forall a, b, c \in A, (a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R$
- A relation with all the above properties is called an *equivalence* relation.
- An equivalence relation divides a set into partitions called *equivalent classes*.

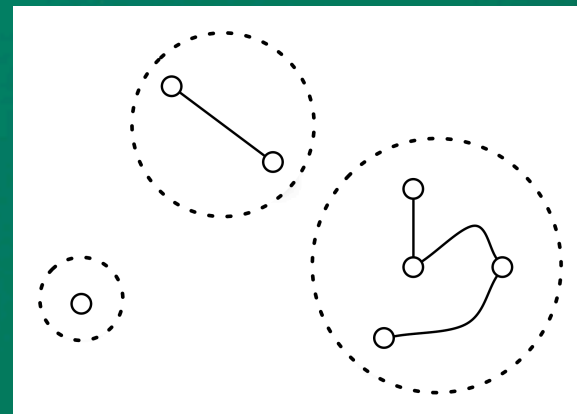
An equivalence relation

Example: connectedness is an equivalence relation over undirected graphs.



What are the equivalence classes called?

Connected Components

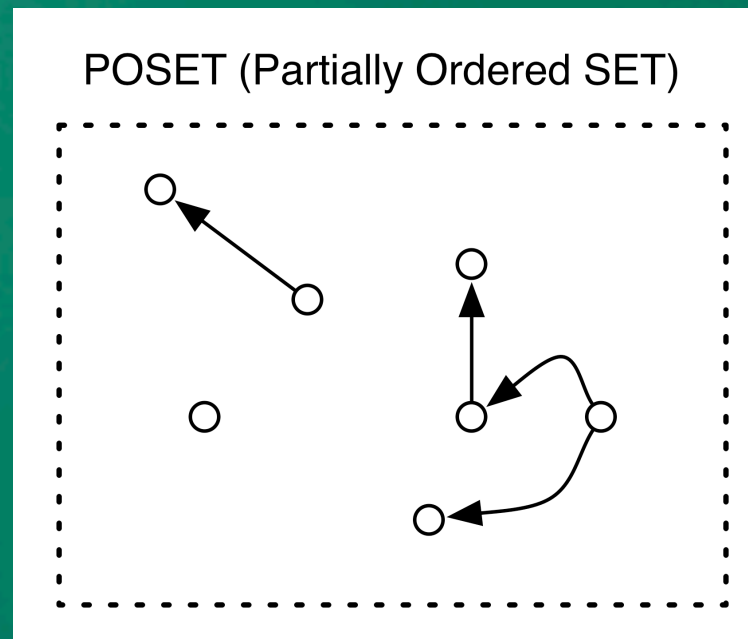


Partial order relations

- A relation R over A is called *anti-symmetric* iff
$$\forall a, b \in A, (a, b) \in R \wedge (b, a) \in R \Leftrightarrow a=b$$
- As an example, \leq is a reflexive, transitive, anti-symmetric relation over \mathbb{Z} .
- Every reflexive, transitive anti-symmetric relation over A is called a *partial order*.
- The partiality comes from the fact that nothing in this definition forces every two elements of the set to be ordered by the relation.

An example of a partial order

- Give a subset S of digraphs (directed graphs) such that directed connectedness is a partial order for every G in S .
 - ✓ DAGs (Directed Acyclic Graphs)
- A set paired with a partial order relation is called a *partially ordered set* or *poset*.
- **Be Precise:** $V(G)$ is the POSET!



b Total order

- Adding totality restriction: A relation R over A is called a *total order* iff it is transitive and anti-symmetric and

$$\forall a, b \in A, (a, b) \in R \vee (b, a) \in R$$

- Therefore \leq is a total order over \mathbb{Z} .
- Note that reflexivity results from totality!
- A set paired with a total order is called a *totally ordered set*.

An example of a total order

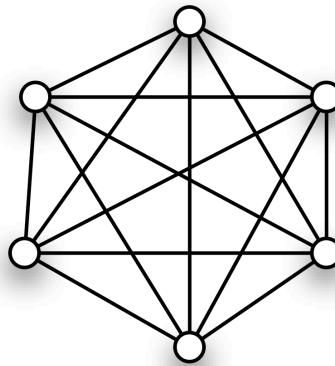
- Give a subset S of digraphs such that directed connectedness is a total order for every G in S .
 - ✓ Directed Acyclic Tournaments

Be Precise:

Actually, $V(G)$ is the totally ordered set, not G . G is not even a set!

A complete graph:

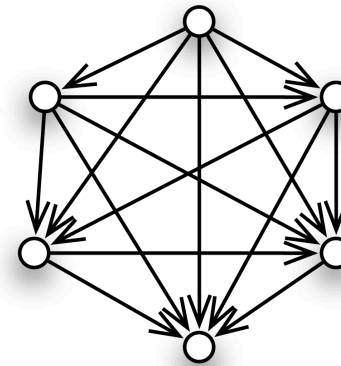
K_6



Assigning direction to every edge



A tournament



Functions

- A relation f from A to B , in which for every a in A there is at most one (a, b) in f is called a *(unary) function*.
- If $(a, b) \in f$, we write $f(a)=b$ and say that $f(a)$ is *defined* (otherwise it is *undefined*). (a is called argument of f and b is called the value of f at point a)
- *Domain* of f or $Dom(f)$ is the set of all a in A such that $f(a)$ is defined.
- A function from A to B is *total* if $Dom(f)=A$, otherwise it is *partial*. A total function f from A to B is shown as $f: A \rightarrow B$.

Functions

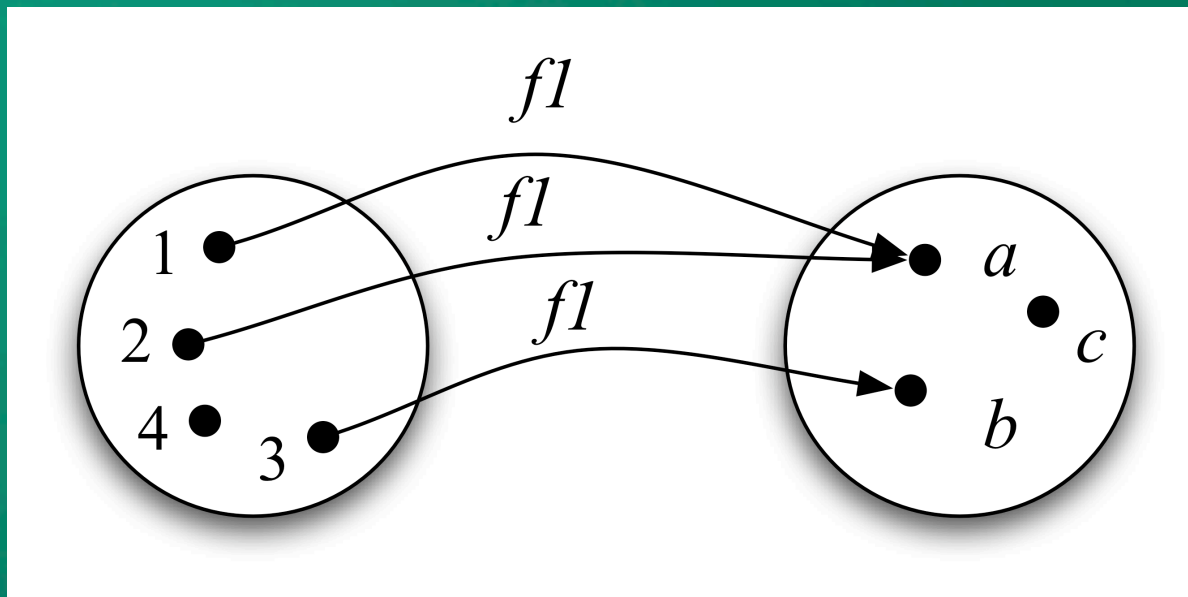
- If f is a function from A to B , B is called the *codomain* of f .
- The set of all b in B such that $(a, b) \in f$ for some $a \in A$ is called the *image* of f .
- The term *range* has been used in the literature to refer to both codomain and image. Throughout the course, we use the term range equivalent to codomain.

Functions: some examples

- Consider following sets and relations.
 - $A = \{1, 2, 3, 4\}$, $B = \{a, b, c\}$
 - $f1 = \{(1, a), (2, a), (3, b)\}$, $f2 = \{(1, a), (1, b), (2, c)\}$
- $f1$ is a function, but $f2$ is not.
- $Dom(f1) = \{1, 2, 3\}$, $Range(f1) = B$, $Image(f1) = \{a, b\}$.

Functions as mappings

- Functions are sometimes called mappings.



Functions: another example

- Consider the following function.

$$f: \mathbb{R} \rightarrow \mathbb{R}$$

$$f(x) = 1/(x^2 - 1)^{1/2}$$

- $Dom(f) = \{x \in \mathbb{R} \mid |x| > 1\}$, $Range(f) = \mathbb{R}$, $Image(f) = \mathbb{R}^+$

n-ary functions

- An *n-ary* function is a function from $A_1 \times A_2 \times \dots \times A_n$ to B .
- The generalization of the previously defined terms to n-ary case is straightforward.
- Throughout the course, we sometimes use the term *an (n-ary) operator over A* to call a total (n-ary) function from A^n to A ($n > 0$).

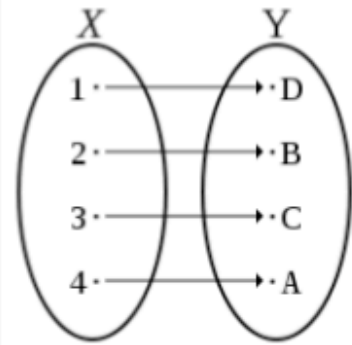
Functions: some examples

- Arithmetic negation is a unary operator over R ($-: R \rightarrow R$).
- Arithmetic addition and multiplication are binary operators over R (i.e. $+: R \rightarrow R$).
- Division is a binary operation over $R - \{0\}$.
- In-fix notation for binary operations: $a+b$ instead of $+(a, b)$.

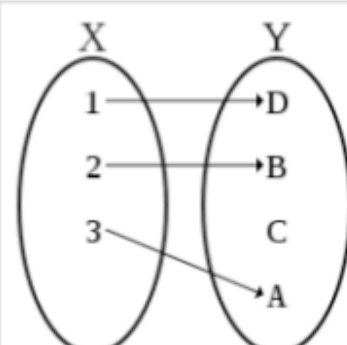
Properties of functions

- The function $f:A \rightarrow B$ is called *surjective* (or onto) iff $\text{Image}(f)=B$.
- The function $f:A \rightarrow B$ is called *injective* (or one-to-one) iff $\forall a, a' \in A, f(a)=f(a') \Rightarrow a=a'$
- A function $f:A \rightarrow B$ is called a *bijective* function (or a *bijection*) if it is both surjective and injective.
- A bijection is also called a *one-to-one correspondence*.

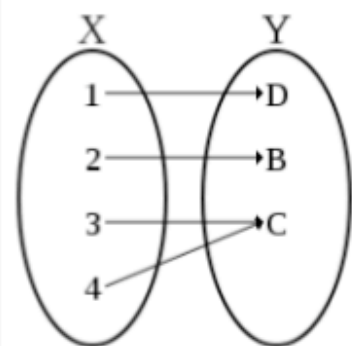
Properties of functions



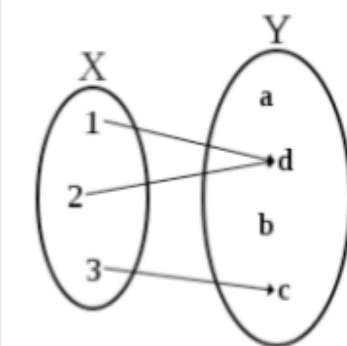
Injective and surjective
(bijective).



Injective and non-surjective.



Non-injective and surjective.



Non-injective and non-surjective.

Properties of functions

- Consider

$$f1: \mathbb{R} \rightarrow \mathbb{R}$$

$$f1(x) = x^2$$

$$f2: \mathbb{R}^+ \rightarrow \mathbb{R}^+$$

$$f2(x) = 1/x^{1/2}$$

- Are $f1$ and $f2$ surjective, injective, or bijective?