Logic and Proof

proposition logic True (T) or False (F)

variable to represent proposition

operator $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

Negation (not) $\neg P$

Truth table

P	$\neg P$
T	F
F	T

Conjunction (and) $P \wedge q$

Truth table $P \wedge q$

P	q	$P \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Disjunction (or) $P \vee q$

Truth table $P \vee q$

P	q	$P \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Connection to Boolean algebra

	Logic	Boolean algebra
value	F/T	0/1
operator	$\neg P$	$\overline{P}$
	$P \wedge q$	$P \cdot q$
	$P \vee q$	$P + q$

Implication $P \rightarrow q$

Truth table

P	q	$P \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Contrapositive

$\neg q \rightarrow \neg P = P \rightarrow q$

Truth table same as $P \rightarrow q$

Bioconditional

$P \leftrightarrow q = (P \rightarrow q) \wedge (q \rightarrow P)$

Truth table

P	q	$P \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Tautology always true

Contradiction always false

Equivalence (logically equivalent)

prove $\equiv$ using truth table

$p \equiv q$

De Morgan's Laws

$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$

propositional function $P(x)$

Quantifiers

$\mathbb{N}$ Domain : int positive

$\mathbb{Z}$ Domain : int

$\forall$ all value

$\exists$ some value

De Morgan's Law for Quantifiers

every x $P(x)$ false $\leftarrow$ $\boxed{\neg \exists x \in D \ P(x)} \equiv \boxed{\forall x \in D \ \neg P(x)}$ $\rightarrow$ an x for $P(x)$ true

$\underbrace{\neg \forall x \in D \ P(x)}_{\text{an } x \text{ for } P(x) \text{ false}} \equiv \underbrace{\exists x \in D \ \neg P(x)}_{P(x) \text{ always true for every } x}$

Inference rule

	Premises		Deduce a statement
Rule 1	p	$p \to q$	q
Rule 2	$\neg q$	$p \to q$	$\neg p$
Rule 3	$p \to q$	$q \to r$	$p \to r$
Rule 4	$p \vee q$	$\neg p$	q
Rule 5	$p \vee q$	$\neg p \vee r$	$q \vee r$
Rule 6	p	q	$p \wedge q$
Rule 7	p		$p \vee q$
Rule 8	$p \wedge q$		p

Nested Quantifiers

$\forall x \forall y \ Q(x,y)$

$\exists x \exists y \ Q(x,y)$

$\forall x \exists y \ Q(x,y)$

$\exists x \forall y \ Q(x,y)$

Proof Method

Direct proof $p \to q$

Proof by contrapositive $\neg q \to \neg p$

Proof by contradiction $\neg q \to$ show contradiction

Proof by cases/exhaustion cover all case & prove true in each case

Existence proofs prove $\exists x \in D \ P(x)$ ($x=a$, $P(a)$ is true)

(nonconstructive) (use some property, no value of x)

Disprove by counter example prove $\forall x \in D \ P(x)$ false

Proof by induction base case prove $P(1) \to$ show if $P(k)$ is true, then $P(k+1)$ is also true

Lecture 3 Basic Structures

Set → collection of elements $x \in S$ x in set S

empty set : $\emptyset$ $x \notin S$ x not in set S

set builder

$S = \{x \mid x \in \text{Domain and Predicate}(x)\}$

$S = \{x \in \text{Domain and Predicate}(x)\}$

set operation

$S \subseteq T$	subset	$\forall x (x \in S \to x \in T)$
$S \supseteq T$	superset	$\forall x (x \in T \to x \in S)$
$S = T$	set equality	$\forall x (x \in S \leftrightarrow x \in T)$
$S \cup T$	union	$\{x \mid x \in S \vee x \in T\}$
$S \cap T$	intersection	$\{x \mid x \in S \wedge x \in T\}$
$S - T$	difference	$\{x \mid x \in S \wedge x \notin T\}$
$\lvert S \rvert$	cardinality	number of elements in S

De Morgan's Law of Union

$(A \cup B)' = A' \cap B'$

De Morgan's Law of Intersection

$(A \cap B)' = A' \cup B'$

Cartesian product

$S \times T = \{(s,t) \mid s \in S \wedge t \in T\}$

(s,t) pair

Venn diagram

$|S \cup T| = |S| + |T| - |S \cap T|$

Set identity

$S \cup \emptyset = S$

$S \cup T = T \cup S$

$(S \cup T) \cup R = S \cup (T \cup R)$

$S \cup (T \cap R) = (S \cup T) \cap (S \cup R)$

$S \cap \emptyset = \emptyset$

$S \cap T = T \cap S$

$(S \cap T) \cap R = S \cap (T \cap R)$

$S \cap (T \cup R) = (S \cap T) \cup (S \cap R)$

Sequence ordered collection of element

increasing / decreasing / subsequence

recurrence → Fibonacci sequence

nested summation

Matrix rectangular array

addition

multiplication row × column

Function domain X, range Y

one-to-one function

onto function

Relation set of n-tuples

binary relation

Properties

Reflexive $\forall x \in X \ (x,x) \in R$

Symmetric $\forall x,y \in X \ (x,y) \in R \rightarrow (y,x) \in R$

Transitive $\forall x,y,z \in X \ (x,y) \in R \wedge (y,z) \in R \rightarrow (x,z) \in R$

prove reflexive: $y = x \Rightarrow$ prove relation

prove symmetric: y, x swap $\Rightarrow$ prove / counter example

prove transitive: two $\begin{matrix} x & y \\ \downarrow & \downarrow \\ y & z \end{matrix}$ conversion

equivalence ← reflexive, symmetric, transitive

partial ordering ← reflexive, antisymmetric, transitive

Lecture 4 Algorithm

Algorithm Computation problem steps

$f(n) = O(g(n))$ $\quad f(n) \le c'g(n)$ $\quad$ upper bound

$f(n) = \Omega(g(n))$ $\quad f(n) \ge cg(n)$ $\quad$ lower bound

$f(n) = \Theta(g(n))$ $\quad cg(n) \le f(n) \le c'g(n)$ $\quad$ tight bound

Complexity		*Algorithm*
O(1)	Constant time	Compare two numbers
O(log n)	Logarithmic	Binary search (on a sorted array)
O(n)	Linear time	Search (on a unsorted array)
O(n log n)		Merge sort
O(n^2)	Quadratic	Selection sort
O(n^3)	Cubic	Matrix multiplication
O(2^n)	Exponential	Brute-force search on boolean satisfiability problem
O(n!)	Factorial	Brute-force search on traveling salesman problem

Running time

Lecture 5 Induction and Recursion

mathematical induction

Prove $P(n)$ is true for $\forall n$

basis step — show $P(1)$ is true.

inductive step — Assume $P(k)$ is true $\Rightarrow$ $P(k+1)$ is true

strong induction

basis step — show $P(1)$ is true.

inductive step — Assume $P(j)$ is true for all int j with $2 \le j \le k$

$\Rightarrow P(k+1)$ is true

Recursive

factorial
- basic step: $F(0) = 1$
- recursive step: $F(n) = n \times F(n-1)$

binomial coefficient
- basic step: $C(n,0) = 1$
- recursive step: $C(n,k) = C(n-1,k-1) + C(n-1,k)$

Tower of Hanoi

```
SolveHanoi(integer n, rod S, rod T, rod R)
    if n>0
        SolveHanoi(n-1, S, R, T)
        move to top disk from S to T
        SolveHanoi(n-1, R, T, S)
```

Merge-sort

```
MergeSort(Array A[1...n])
    if n>1
        m ← ⌊n/2⌋
        B[1...m] ← A[1...m]
        C[1...n-m] ← A[m+1...n]
        MergeSort(B[1...m])
        MergeSort(C[1...n-m])
        A[1...n] ← merge(B[1...m], C[1...n-m])
```

```
merge(Array L[1...l], Array R[1...r])
    n ← l+r
    create new array A[1...n]
    i ← 1; j ← 1
    for k ← 1 to n
        if i ≤ l and (j > r or L[i] ≤ R[j])
            A[k] ← L[i]; i = i+1
        else
            A[k] ← R[j]; j = j+1
    return A.
```

Lecture 6 Counting

Product rule $|S_1 \times S_2| = |S_1| \times |S_2|$

Sum rule $|S_1 \cup S_2| = |S_1| + |S_2|$ if $S_1 \cap S_2 = \emptyset$

Subtraction rule $|S_1 \cup S_2| = |S_1| + |S_2| - |S_1 \cap S_2|$

Division rule $n = \frac{|S|}{d}$

Principle of Inclusion-Exclusion $|S_1 \cup S_2 \cup \cdots \cup S_n| = \sum_{1 \le i \le n} |S_i| - \sum_{1 \le i \le j \le n} |S_i \cap S_j|$
$+ \sum_{1 \le i < j < k \le n} |S_i \cap S_j \cap S_k|$
$+ (-1)^{n-1} |S_1 \cap S_2 \cap \cdots \cap S_n|$

Tree Diagram

Pigeonhole principle (prove by contrapositive) $\forall k \in \mathbb{Z}^+$

$(k+1)$ (or more) objects are placed into k boxes at least 1 box has at least 2 objects.

Generalize N & $k \in \mathbb{Z}^+$ $(N > k)$ N objects placed in k boxes,

at least 1 box contains at least $\lceil N/k \rceil$ objects

Permutation / Combination

$P(n,r) = \frac{n!}{(n-r)!}$ ↳ prove by product rule

$C(n,r) = C(n,n-r) = \frac{n!}{r!(n-r)!}$

algebraic proof
or
combinatorial proof

↳ r-permutation & division rule

Binomial theorem $(x+y)^n = \sum_{r=0}^{n} \binom{n}{r} x^{n-r} y^r$

Pascal's identity $\binom{n+1}{r} = \binom{n}{r-1} + \binom{n}{r}$

Generalized permutation and combination

combination with repetition $C(n+r-1, r)$

permutations with indistinguishable objects $\frac{n!}{n_1! n_2! \cdots n_k!}$