COMPSCI 1DM3

1.1-1.2

Winter 2023

Introduction

A proposition is a declarative sentence - meaning it declares a fact - that is either **true** or **false**, but NEVER both.

Example 1:

Here are two propositions:

1. $1 + 2 = 3$
2. $x + 2 = 3$

Proposition 1 is true, but 2 is not a proposition. If we assign a variable to it, it would be a proposition.

We use $p, q, r, s, \ldots,$ as variables to denote propositions. The truth value of a proposition can be T/F.

Types of Propositions

The first type of proposition is an **atomic proposition** which is a proposition that is not composed of other propositions. It is the simplest form of a proposition.

Propositions that are comprised of other propositions (atomic or compound) are called **compound propositions**. They are a collection of propositions that are connected together with logical operators.

Logic Operators

Logical operators are used to connect propositions together. The are used to combine atomic and/or compound propositions with one another to form a logical statement.

Negation

The negation of p is like saying **not** p. It is denoted by $\neg p$ (read 'not' p). The truth value also gets flipped. The negation of p is the statement 'it is not the case that p'.

p	$\neg p$
T	F
F	T

Table 1: Truth table for negation of p

Conjunction

The conjunction of p and q is like saying p **and** q. It is denoted by $p \wedge q$. The truth value is only true when both propositions are true.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Table 2: Truth table for the conjunction of $p \wedge q$

Note you can also say p, but q to represent a conjunction in English.

Disjunction

The disjunction of p and q is like saying p **or** q. It is denoted by $p \vee q$. The truth value is only false when both propositions are false.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Table 3: Truth table for the disjunction of $p \vee q$

This is the **inclusive** or operator. Both p and q can be true, however, if we want either p or q (exclusive), we use XOR.

Exclusive Or

The exclusive or of p and q is like saying p **xor** q. It is denoted by $p \oplus q$. The truth value is only true when one of the propositions are true, otherwise its false.

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Table 4: Truth table for the exclusive or of $p \oplus q$

Conditionals

The conditional of p and q is like saying **if** p, **then** q. It is denoted by $p \to q$. The truth value is only false when p is **true** and q is **false**. Conditionals are also called **implications**. p is often called the *hypothesis* and q is called the *conclusion*.

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Table 5: Truth table for the conditional of $p \to q$

There are multiple ways to represent conditionals.

"if p, then q" "if p, q" "p is sufficient for q" "p implies q" "p only if q"

"a sufficient condition for q is p" "q whenever p" "q is necessary for p" "q follows from p"

"q provided that p" "q if p" "q when p" "a necessary condition for p is q" "q unless $\neg p$"

Converse, Inverse, and Contrapositive

The proposition $q \to p$ is the **converse** of $p \to q$.

The proposition $\neg p \to \neg q$ is the **inverse** of $p \to q$.

The **contrapositive** of $p \to q$ is $\neg q \to \neg p$ and has the same truth value as $p \to q$, therefore $p \to q \equiv \neg q \to \neg p$

Example 2:

"The home team wins whenever it's raining"

Since q when p is a way to represent a conditional, we can find its converse, contrapositive, and inverse.

q = 'the home team wins'
p = 'it's raining'

Converse: $q \to p$ = if the home team wins, then it's raining.
Inverse: $\neg p \to \neg q$ = if it's not raining, then the home team doesn't win.
Contrapositive: $\neg q \to \neg p$ = if the home team doesn't win, then it wasn't raining.

As you can see, only the contrapositive is correct since the team can still win if its not raining, the conditional is just that IF its raining, they will win.

Biconditionals

The biconditional of p and q is like saying p, **if and only if** q. It is denoted by $p \leftrightarrow q$. The truth value is only true when p and q have the same truth values. Biconditionals are also called bi-implications.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Table 6: Truth table for the biconditional of $p \leftrightarrow q$

Other ways to express bi-conditionals:

"p is necessary and sufficient for q" "if p then q, and conversly" "p iff q" "p exactly when q"

Compound Propositions

Now that we have 5 logical connectives – negation, conjunction, disjunction, exclusive or, conditional, biconditional – we can make compound propositions and find its truth value by building from ground up.

Example 3:

Construct the following truth table for the following compound proposition.

$$(p \lor \neg q) \to (p \land q)$$

We break this compound proposition into its most basic components p and q. We should also find $\neg q$ as its used in $p \lor \neg q$. From there we can find the truth values for $p \lor \neg q$ as well as $p \land q$ and solve for the conditional $(p \lor \neg q) \to (p \land q)$.

p	q	$\neg q$	$(p \lor \neg q)$	$(p \land q)$	$(p \lor \neg q) \to (p \land q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	T

Table 7: Truth table for $(p \lor \neg q) \to (p \land q)$

Precedence of Logical Operators

There is a precedence of the logical operators we have learned about. Negation has the highest precedence, meaning a proposition like $\neg p \lor q$ is actually $(\neg p) \lor q$ not $\neg(p \lor q)$.

Next is the rule that conjunction takes more precendence than disjunction. $p \land q \lor r$ means $(p \land q) \lor r$ not $p \land (q \lor r)$.

Lastly, is the rule that conditionals and biconditionals have less precedence than all previously mentioned operators. $p \to q \lor r$ means $p \to (q \lor r)$ not $(p \to q) \lor r$.

Logic and Bit Operations

Bits are symbols with values either 0 or 1. A bit can represents the truth values with 1 being true and 0 being false.

We can also store information in bit strings where the length of said string is the number of bits in the string.

We can apply bit operations on bit strings such as bitwise OR, bitwise XOR, & bitwise AND.

01 1011 0110

11 0001 1101

____

11 1011 1111 bitwise OR

10 1010 1011 bitwise XOR

01 0001 0100 bitwise AND

1.2 Applications of Propositional Logic

Logic Circuits

We can model digital circuits from 3 basic circuits called gates.

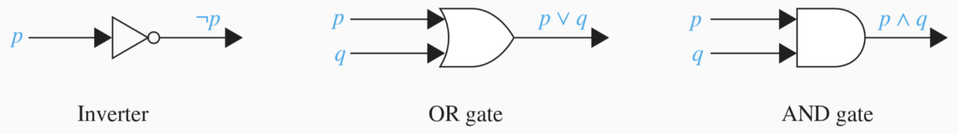

FIGURE 1 Basic logic gates.

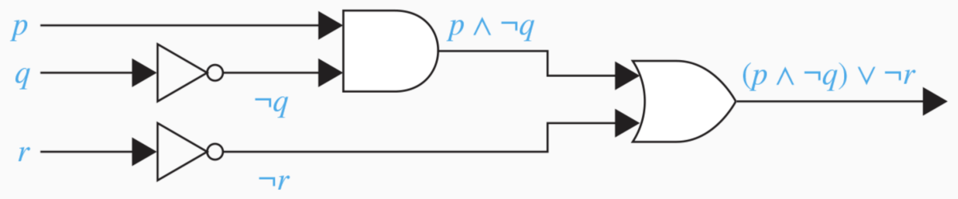

FIGURE 2 A combinatorial circuit.

1.3 Propositional Equivalences

A **compound** proposition that is **always true** (regardless of the truth values given to the propositional variables) is called a **tautology**.

Similarly, a **compound** proposition that is **always false** is called a **contradiction**.

If neither a tautology or contradiction, it is called a **contingency**.

Tautology	Contradiction
$p \vee \neg p$	$p \wedge \neg p$
T	F
T	F

Table 8: Example of tautology and contradiction

Logical Equivalences

Compound propositions that have the same truth tables in all cases are deemed logically equivalent. We can also say the **compound propositions** p and q are logically equivalent if $p \leftrightarrow q$ is a tautology.

Notation: $p \equiv q$ or $p \iff q$

One way to determine if two compound propositions are logically equivalent is to create a truth table for them. Then simply checking if the truth values are the same for both will determine if they are logically equivalent!

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

p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

Table 9: Truth table for $\neg(p \vee q)$ and $\neg p \wedge \neg q$

Important Equivalences

Equivalence	Name
$p \wedge T \equiv p$ $p \vee F \equiv p$	Identity Laws
$p \vee T \equiv T$ $p \wedge F \equiv F$	Domination Laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent Laws
$p \vee \neg p \equiv T$ $p \wedge \neg p \equiv F$	Negation Laws
$\neg(\neg p) \equiv p$	Double Negation Laws
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative Laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative Laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive Laws
$\neg(p \vee q) \equiv \neg p \wedge \neg q$ $\neg(p \wedge q) \equiv \neg p \vee \neg q$	De Morgan's Laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption Laws

Table 10: Logical Equivalences

Equivalence
$p \rightarrow q \equiv \neg p \vee q \equiv \neg q \rightarrow \neg p$
$p \vee q \equiv \neg p \rightarrow q$
$p \wedge q \equiv \neg(p \rightarrow \neg q)$
$\neg(p \rightarrow q) \equiv p \wedge \neg q$
$(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$
$(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$
$(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$
$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$

Table 11: Logical Equivalences Including Conditional Statements

Equivalence
$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Table 12: Logical Equivalences Including Biconditional Statements

General Distributive Law
$(p \vee s) \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r) \vee (s \wedge q) \vee (s \wedge r)$
$(p \wedge s) \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r) \wedge (s \vee q) \wedge (s \vee r)$

Table 13: General Distributive Laws

We can use these equivalences to create new logical equivalences.

Example 4:

Show that $\neg(p \vee (\neg p \wedge q)) \equiv \neg p \wedge \neg q$ using a series of logical equivalences.

$$\begin{aligned}
\neg(p \vee (\neg p \wedge q)) &\equiv \neg p \wedge \neg(\neg p \wedge q) \quad \text{De Morgan Law} \\
&\equiv \neg p \wedge (p \vee \neg q) \quad \text{De Morgan Law} \\
&\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) \quad \text{Distributive Law} \\
&\equiv F \vee (\neg p \wedge \neg q) \quad \text{Negation Law} \\
&\equiv \neg p \wedge \neg q \quad \text{Identity Law}
\end{aligned}$$

Satisfiability

A compound proposition is **satisfiable** if we can assign truth values to the variables to make the proposition true. Therefore, when the compound proposition is false for all assignments of truth values, we call it **unsatisfiable**. This means the negation of an unsatisfiable proposition is a tautology.

1.4 Predicates and Quantifiers

This chapter covers predicate logic which expresses the meaning of a wide range of statements.

Predicates

Statements like $x > 3$ are neither true or false until the variable x is specified/given a value.

This statement $x > 3$ has two parts. The variable x is the subject while the 'is greater than 3' part is the **predicate**. Therefore, predicates are properties the subject can have.

We can denote $x > 3$ as $P(x)$ where P is the predicate and x is the variable. This creates the **propositional function** P at x. When we assign a value for x, $P(x)$ becomes a proposition with a truth value.

Let $P(x)$ denote the statement '$x > 3$'. What is $P(4)$ and $P(2)$?
$P(4)$ is true. $P(2)$ is false.

We can also denote propositional functions for statements with more than one variable. For example, $x+y = 3$ can be denoted as $Q(x, y)$ where Q is the predicate and x, y the variables.

In general, a statement with n variables $x_1, x_2, \ldots, x_n$ can be denoted by $P(x_1, x_2, \ldots, x_n)$. In this case, P is called an n-place predicate or n-ary predicate.

Preconditions and Postconditions

We can use predicates to determine the correctness of a computer program. This means we can show that computers program always produce the desired output when given a valid input. The statements that describe valid inputs are called **preconditions** and the conditions that the output should satisfy are the **postconditions**.

Quantifiers

Another way to create a proposition from a propositional function is through **quantification**. Quantification tells us the extent of which a predicate is true over a range of elements. In English we use words like all, some, many, none, and few quantifications.

Universal Quantification

This tells us that a predicate is true for every element under consideration.

The universal quantification of $P(x)$ for a domain is the proposition that declares $P(x)$ to be true for all values of x in that domain. The universal quantification of $P(x)$ will change when we change the domain – so the domain must always be specified or else the universal qualification is undefined. Domains can also be called the **domain of discourse** or **universe of discourse**.

The universal quantification of $P(x)$ is the statement '$P(x)$ for all values of x in the domain'.

Notation: $\forall x P(x)$ denotes the universal quantification where $\forall$ is called the universal quantifier. We would read this as 'for all $xP(x)$'. An element that makes $P(x)$ false is called a counterexample of $\forall x P(x)$.

Example 5:

Let $P(x)$ be the statement '$x + 1 > x$'. What is the truth value of the quantification $\forall x P(x)$, where the domain consists of all real numbers?

$P(x)$ is true for all real numbers, therefore the quantification $\forall x P(x)$ is true!

Remark: If the domain is empty, $\forall x P(x)$ is true as there are no elements that make it false.

Other ways of expressing universal quantification include ‘all of’, ‘for each’, ‘given any’, ‘for arbitrary’, and ‘for any’.

Example 6:

Let $Q(x)$ be the statement ‘$x < 2$’. What is the truth value of the quantification $\forall x Q(x)$, where the domain consists of all real numbers?

$Q(x)$ is not true for all real numbers. For example $Q(3)$ is false. $x = 3$ is a counterexample for the statement $\forall x Q(x)$, therefore, $\forall x Q(x)$ is false. Note we only need one counterexample to prove $\forall x Q(x)$ is false.

Existential Quantification

This tells us there is one or more element under consideration for which the predicate is true. The existential qualification of $P(x)$ is the proposition ‘there **exists** an x in the domain such that $P(x)$’.

Notation: $\exists x P(x)$ denotes the existential quantification of $P(x)$ where $\exists$ is called the existential quantifier.

Other ways of expressing existential quantification include ‘for some’, ‘for at least one’, and ‘there is’.

Example 7:

Let $P(x)$ be the statement ‘$x > 3$’. What is the truth value of the quantification $\exists x P(x)$, where the domain consists of all real numbers?

$P(x)$ is true sometimes such as $P(4)$. As long as there is at least one x where the statement is true, then $\exists x P(x)$ is true.

Remark: If the domain is empty, $\exists x P(x)$ is false as there are no elements that could make it true.

Uniqueness Quantifier

This tells us that there is a **unique** element that makes $P(x)$ true. It is denoted $\exists!$ or $\exists_1$. The notation $\exists! x P(x)$ or $\exists_1 x P(x)$ states ‘there exists a unique x such that $P(x)$ is true’.

Finite Domains

If the domain is finite, then the elements of the domain $x_1, x_2, \ldots, x_n$ can be listed. The univesal quantification $\forall x P(x)$ is the same as the conjunction $P(x_1) \land P(x_2) \land \cdots \land P(x_n)$.

Similarly, the existential quantification $\exists x P(x)$ is the same as the disjunction $P(x_1) \vee P(x_2) \vee \cdots \vee P(x_n)$ as only one need to be true.

Restricted Domains

If the domain of a universal quantification is restricted, it is the same as the universal quantification of a conditional statement. If the domain of a existential quantification is restricted, it is the same as the existential quantification of a conjunction.

Example 8:

Consider $\forall x < 0(x^2 > 0)$ and $\exists y > 0(y^2 = 0)$. State their restricted domains and equivalent forms.

The statement $\forall x < 0(x^2 > 0)$ has the restricted domain of $x < 0$ and is equivalent to $\forall(x < 0 \to x^2 > 0)$.

The statement $\exists y > 0(y^2 = 0)$ has the restricted domain of $y > 0$ and is equivalent to $\exists(y > 0 \wedge y^2 = 0)$.

Remark: the quantifiers $\forall$ and $\exists$ have higher precendence than ALL logical operators from propositional calculus.

Binding Variables

When a quantifier is used on a variable, we say the occurrence of that varialbe is **bound**. An occurrence of a variable not bound or assigned a value is said to be **free**. The part of the logical expression to which the quantifer is applied to is called the **scope** of the quantifer. Therefore, a variable is free if its outside the scope of all quantifers used.

Example 9:

Consider $\exists x(x + y = 1)$. The variable x is bound by $\exists x$, but y is free as its not assigned a value nor bound.

Consider $\exists x(P(x) \wedge Q(x)) \vee \forall x R(x)$. The scope of $\exists x$ is $P(x) \wedge Q(x)$ since $\exists x$ is only applied to that part and not the whole statement. Likewise, the scope of $\forall x$ is $R(x)$. Notice that $\exists x$ binds x to $P(x) \wedge Q(x)$ and $\forall x$ binds x to $R(x)$.

We could've written it with two variables x, y since the scopes don't overlap, however, its more common to use one letter.

Logical Equivalence

Statements with predicates and quantifers are logically equivalent if and only if they have the same truth value no matter what predicate is substituted and no matter what domain is used.

Example 10:

Show $\forall x(P(x) \land Q(x))$ and $\forall x P(x) \land \forall x Q(x)$ are logically equivalent.

We must show that regardless of domain and what P and Q are, they must take the same truth values. We assume a common domain between P and Q.

We can show that by $\forall x(P(x) \land Q(x))$ and $\forall x P(x) \land \forall x Q(x)$ are logically equivalent by proving if $\forall x(P(x) \land Q(x))$ is true, then $\forall x P(x) \land \forall x Q(x)$ is true, and conversly.

Suppose $\forall x(P(x) \land Q(x))$ is true and that a is in the domain, then $P(a) \land Q(a)$ is true so $P(a)$ is true and $Q(a)$ is true. Therefore, we can conclude that $P(a)$ and $Q(a)$ are true for every a in the domain. Consequently, $\forall x P(x)$ and $\forall x Q(x)$ are both true so $\forall x P(x) \land \forall x Q(x)$ is true.

Negating Quantified Expressions

Consider $P(x)$ to be the statement 'x has taken a course in calculus' where the domain consists of students in your class.

Say we want to negate the phrase 'every student in your class has taken a course in calculus'. That statement is the equivalent of $\forall x P(x)$. Negating it would be equivalent to 'there is a student in your class who hasn't taken a course in calculus' which is the existential quantification of the negation of $P(x)$.

$$\therefore \neg \forall x P(x) \equiv \exists x \neg P(x)$$

Say we want to negate the phrase 'there is a student in your class who has taken a course in calculus'. This is equivalent to $\exists x P(x)$. Negating this would be equivalent to 'every student in your class hasn't taken calculus' which is the universal qualification of the negation of $P(x)$.

$$\therefore \neg \exists x P(x) \equiv \forall x \neg P(x)$$

These are called the **De Morgan's Laws for Quantifiers**. When the domain of $P(x)$ consists of n elements, it follows that $\neg \forall x P(x)$ is the same as $\neg(P(x_1) \land P(x_2) \land \cdots \land P(x_n))$. This is equivalent to $\neg P(x_1) \lor \neg P(x_2) \lor \cdots \lor \neg P(x_n)$ which is the same as $\exists x \neg P(x)$.

Example 11:

What is the negation of the statement $\forall x(x^2 > 2)$?

$\neg \forall x(x^2 > 2) \equiv \exists x \neg(x^2 > 2)$ so we negate $x^2 > 2$ so it becomes $x^2 \leq 2$. This can be written as $\exists x(x^2 \leq 2)$

1.5 Nested Quantifiers

Nested Quantifiers are when we have quantifers inside the scope of another such as $\forall x \exists y(x+y=0)$. We can think of everything within a quantifier as a propositional function. Here $\forall x \exists y(x+y=0)$ is the same as $\forall x Q(x)$, where $Q(x)$ represents $\exists y P(x,y)$ where $P(x,y)$ is $(x+y=0)$.

Order of Quantifiers

The order of quantifiers matters. We should think of nested quantifiers as nested loops. For example, $\forall x \forall y$ is the same as $\forall y \forall x$.

Statement	When True?
$\forall x \forall y$ $\forall y \forall x$	$P(x,y)$ is true for all pairs of x,y
$\exists x \exists y$ $\exists y \exists x$	There is a pair of x,y that makes $P(x,y)$ true
$\forall x \exists y$	For every x there is a y that makes $P(x,y)$ true
$\exists x \forall y$	There is a x that makes $P(x,y)$ true for every y

Statement	When False?
$\forall x \forall y$ $\forall y \forall x$	There's a pair x,y that makes $P(x,y)$ false
$\exists x \exists y$ $\exists y \exists x$	$P(x,y)$ is false for every pair x,y
$\forall x \exists y$	There is an x where every y makes $P(x,y)$ false
$\exists x \forall y$	For every x there is a y that makes $P(x,y)$ false

Table 14: Quantification of Two Variables

Negating Nested Quantifiers

To negate nested quantifiers, we successively apply the same De Morgan rules we learned in 1.3! To remember it, negate and move inward!

Example 12:

Negate the following $\forall x \exists y(xy=1)$

$$\begin{aligned}\forall x \exists y(xy=1) &= \neg \forall x \exists y(xy=1)\\ &= \exists x \neg \exists y(xy=1)\\ &= \exists x \forall y \neg(xy=1)\\ &= \exists x \forall y(xy \neq 1)\end{aligned}$$

1.8 Proof Methods and Strategy

In this section, we study more proof methods as well as advice on how to find proofs of theorems.

Exhaustive Proof and Proof by Cases

This proof method will allow us to prove a theorem through the use of cases. With this, we can prove a conditional statement of form

$$(p_1 \vee p_2 \vee \cdots \vee p_n) \to q$$

This method is based on a new rule of inference which is the tautology

$$[(p_1 \vee p_2 \vee \cdots \vee p_n) \to q] \leftrightarrow [(p_1 \to q) \vee (p_2 \to q) \vee \ldots (p_n \to q)]$$

This is a **proof by cases** where the cases cover all possibilities.

Exhaustive proofs or **proofs by exhaustion** are when we can prove a theorem by examining a small number of examples (we exhaust all possibilites). Each case involves checking one example.

Example 13: Proof by Exhaustion

Prove that $(n+1)^3 \geq 3^n$ if n is a positive integer with $n \leq 4$.
Here all we need to do is simply verify the inequality where $n = 1, 2, 3$, and 4.

For $n = 1$ we have $(n+1)^3 = 8$ and $3^n = 3$ so $(n+1)^3 \geq 3^n$. If we check 2, 3, 4, we'll see the same result.

Example 14: Proof by Cases

Prove that if n is an integer, then $n^2 \geq n$.

Consider 3 cases, where $n > 0, n = 0, n < 0$. This way we cover all possibilities.

This inequality will hold for all 3 cases, so it is valid.

Without Loss of Generality (WLOG)

This is when you can make an assumption in a proof which makes it possible to prove a theorem by reducing the number of cases.

Example 15:

Show that if $x, y \in \mathbb{Z}$ and both xy and $x + y$ are even, then x and y are even.

We start this proof by proof by contraposition. This means we assume the conclusion is false. Assume x or y is odd. With WLOG we only need to consider one of these cases since the other can be proven by swapping roles.

$$x = 2m + 1$$

Now we show that xy or $x + y$ are odd. For this there are two cases, case 1 where y is even so $y = 2n$, and case 2 where y is odd so $y = 2k + 1$.

$$\begin{aligned} (1) \quad x + y &= 2m + 1 + 2n \\ &= 2(m + n) + 1 \\ &= 2k + 1 \text{ is odd.} \end{aligned}$$

$$\begin{aligned} (2) \quad xy &= (2m + 1)(2n + 1) \\ &= 4mn + 2m + 2n + 1 \\ &= 2(2mn + m + n) + 1 \\ &= 2k + 1 \text{ is odd.} \end{aligned}$$

Remark: the most common type of errors occur when not all possible cases are considered.

Existence Proofs

Theorems commonly assert that an object of a particular type exists. These are modeled in the form $\exists x P(x)$ and a proof of this is called an existence proof.

Constructive existence proofs are when we prove $\exists x P(x)$ by finding an element a, called the **witness**, where $P(a)$ is true.

Nonconstructive existence proofs are when we prove that $\exists x P(x)$ is true in a different way. Most commonly, we use a proof by contradiction and show that the negation of $\exists x$ implies a contradiction.

Example 16: Constructive Proof

Show that there is a positive integer that can be written as the sum of cubes of positive integers in two ways.

Using computers,

$$1729 = 10^3 + 9^3 = 12^3 + 1^3$$

Uniquenesss Proofs

Some theorems commonly assert the existence of a unique element with a particular property. In other words, there's exactly one element with this property.

The two parts to a uniqness proof

(1) Existence: show there IS an element x that has the desired property

(2) Uniqueness: show there if x and y have that desired property, then $x = y$

Essentially, we are proving the statement $\exists x(P(x) \wedge \forall y(y \neq x \rightarrow \neg P(y)))$. Wonderful!

Example 17: Uniqueness Proof

If a and b are real numbers where $a \neq 0$, show that there is a unique number r such that $ar + b = 0$

Existence

$$r = -\frac{b}{a}$$
$$a(-\frac{b}{a}) + b = 0$$
$$-b + b = 0$$
$$\therefore \text{ there exists a real number } r$$

Unique – suppose s is a real number

$$ar + b = as + b$$
$$ar = as$$
$$r = s$$

Proof Strategies

Forward reasoning – we begin a direct proof of a conditional statement by using premises and then axioms to reach a conclusion.

Backward reasoning – to prove a statement q we find a statement p such that $p \rightarrow q$.

2.4 Sequences and Summations

A **sequence** is a discrete structure to represent an **ordered list.** We use a_n to denote the *image* of the integer n. We call a_n a **term** of the sequence. Sequences are typically denoted by $\{a_n\}$.

Geometric Progression

This is a sequence in the form

$$a, ar, ar^2, \dots ar^n, \dots$$

where there is an **intial term** a and a **common ratio** r. It is the discrete analogue of $f(x) = ar^x$

Example 18:

Find the inital term and common ratio r for this sequence

$$a_n = 2 \cdot 5^n$$

The intial term is 2 and the common ratio is 5. To find it consider the first few elements...

$$a_n = 2, 10, 50, 250, 1250$$

If there is a common ratio to get from 2 to 10, it would be 5. From 10 to 50 is also 5, and so on.

Arithmetic Progression

This is a sequence in the form

$$a, a + d, a + 2d, \dots, a + nd, \dots$$

where there is an **intial term** a and a **common difference** d. It is the discrete analogue of $f(x) = dx + a$

Example 19:

Find the inital term and common difference d for this sequence

$$a_n = 7 - 3n$$

The intial term is 7 and the common ratio is -3. To find it consider the first few elements...

$$a_n = 7, 4, 1, -2, \dots$$

we can see we are subtracting 3 each time.

Strings

Finite sequences in the form $a_1, a_2, \ldots, a_n$ are called **strings**. The **length** of a string is the number of terms in the string. An **empty string** is denoted by λ and has length 0.

Recurrence Relations

A recurrence relation for the sequence $\{a_n\}$ takes on the form

$$a_0, a_1, \ldots, a_{n-1}$$

as it expresses a_n in one or more previous terms. A recurrence relation recursively defines a sequence. A sequence is a solution of a recurrence relation if its terms satisfy the recurrence relation.

Example 20:

Let $\{a_n\}$ be a sequence that satisfies the reccurence relation $a_n = a_{n-1} + 3$ for $n = 1, 2, 3, \ldots$. If $a_0 = 2$, what is a_1, a_2, a_3?

$$\begin{aligned} a_1 &= a_0 + 3 \\ &= 2 + 3 \\ &= 5 \end{aligned}$$

$$\begin{aligned} a_2 &= a_1 + 3 \\ &= (a_0 + 3) + 3 \\ &= (2 + 3) + 3 \\ &= 8 \end{aligned}$$

$$\begin{aligned} a_3 &= a_2 + 3 \\ &= (a_1 + 3) + 3 \\ &= ((a_0 + 3) + 3) + 3 \\ &= ((2 + 3) + 3) + 3 \\ &= 11 \end{aligned}$$

Intial Conditions: these specify the terms that come before the first term where the recurrence relation takes effect. In our example, it was $a_0 = 2$.

Fibonacci Sequence

Defined by the intial conditions $f_0 = 0, f_1 = 1$, the reccurence relation is

$$f_n = f_{n-1} + f_{n-2}$$

We say we have solved a reccurence relation with its initial conditions when we find an explicit formula (called a **closed formula**). Investment questions are modelled by the formula $a_0 \cdot (1 + r)^t$

Iteration

This is a straightfoward method of solving reccurence relations.

Example 21:

Let $\{a_n\}$ be a sequence that satisfies the reccurence relation $a_n = a_{n-1} + 3$ for $n = 1, 2, 3, \ldots$. If the initial condition is $a_1 = 2$, use iteration to solve the reccurence relation.

$$\begin{aligned}
a_1 &= 2 \\
a_2 &= 2 + 3 \\
a_3 &= (2 + 3) + 3 \\
&= 2 + 3 \cdot 2 \\
a_4 &= ((2 + 3) + 3) + 3 \\
&= 2 + 3 \cdot 3 \\
\\
a_n &= 2 + 3 \cdot (n - 1)
\end{aligned}$$

Special Integer Sequences

To deduce a possible pattern for the next terms of a sequence, ask yourself

- does the same value occur many times in a row?
- are terms obtained from previous terms (by an operation of the same amount or an operation with the index)
- are terms obtained from combining previous terms?

n^{th} Term	First 6 Terms
n^2	1, 4, 9, 16, 25, 36, ...
2^n	2, 4, 8, 16, 32, 64, ...
$n!$	1, 2, 6, 24, 120, 720, ...

Table 15: Common & Useful Sequences

Example 22:

Find a formula for $a_n = 1, 7, 25, 79, 241, \ldots$

There is no obvious pattern, but notice a common ratio is 3 (not constant). So compare with $\{3^n\}$

$$\{3^n\} = 3, 9, 27, 81, 243, \ldots$$

We can see these sequences are off by 2. $\therefore\ a_n = 3^n - 2$

Summation

This is mostly review. Consider the addition of the terms of sequences $a_m + a_{m+1} + \cdots + a_n$ can be represented by

$$\sum_{i=m}^{n} a_i$$

where i is the **index of summation**, m is the **lower limit**, and n is the **upper limit**.

Index shifting – what comes up must come down

$$\sum_{i=1}^{5} i^2 = \sum_{i=0}^{4} (i+1)^2 \tag{1}$$

$$\sum_{i=1}^{5} i^2 = \sum_{i=2}^{6} (i-1)^2 \tag{2}$$

Geometric summations (also known as geometric series) have the formula

$$\sum_{i=0}^{n} ar^i = \begin{cases} \dfrac{ar^i + 1 - a}{r-1}\ , & \text{if } r \neq 1 \\ (n+1)a\ , & \text{if } r = 1 \end{cases}$$

Double summations – similar to nested for loops in computer science. You start by evaluating the inner summation, then the outer.

$$\begin{aligned} \sum_{i=1}^{4}\sum_{j=1}^{3} ij &= \sum_{i=1}^{4} (i + 2i + 3i) \\ &= \sum_{i=1}^{4} (6i) \\ &= (6 + 12 + 18 + 24) \\ &= 60 \end{aligned}$$

We can also add all values of a **function**, or terms of an indexed set where the index of summations loops through every element in the set. Here we write

$$\sum_{s \in S} f(s)$$

to represent the sum of the values $f(s)$ for all elements/members s of S.

$$\begin{aligned} \sum_{s \in 0,2,4} s &= 0 + 2 + 4 \\ &= 6 \end{aligned}$$

Infinite Series – useful infinite summation formulas to know.

$$\begin{aligned}\sum_{i=0}^{\infty} x^i &= \frac{1}{1-x}\\ \sum_{i=0}^{\infty} i x^{i-1} &= \frac{1}{1-x^2}\end{aligned}$$

Note: $|x| < 1$ for both of these formulas.

3.3 Complexities of Algorithms

For an algorithm to be good, it should always produce the correct answer and be **efficient**. We measure efficiency in 2 ways

- **time complexity** – time required to solve a problem relative to input size
- **space complexity** – amount of memory required relative to input size

Time Complexity

This is defined as the **number of operations** used by the algorithm when the input is a particular size.

For example, lets calculate the number of steps for this linear search algorithm:

```
$i := 1$
while ($i \leq n$ and $x \neq a_i$)
    $i := i + 1$
if $i \leq n$ then
    $location := 1$
else $location := 0$
return $location$
```

At each step of the loop, 2 comparisons ($2i$) are made ($i \leq n$ **and** $x \neq a_i$). Outside the loop an additional (+1) comparison is made when $i \leq n$. Therefore $2i + 1$ comparisons are made so its $O(n)$ time complexity.

Worst Case Complexity

When we do a worst case analysis, it tells us how many operations an algorithm requires to **guarantee** that it will produce a solution.

Binary Search Complexity

It has a time complexity of $O(\log n)$. Say we have an array size n. Everytime we divide it $\frac{n}{2}, \frac{n}{4}, \ldots$ times, we are dividing it $n/2^k$ times. Now we divide the array until we find the element (size 1), so until $n/2^k = 1$ which leads to $n = 2^k$. Solving for k is as simple as $\log n = k$.

Bubble Sort & Insertion Sort

These sorting algorithms have a double loop so it is $O(n^2)$.

Matrix Mulitiplication & Boolean Product

You have to go through each row and column of the matrix, and then loop over again to put it for the product matrix so they are both $O(n^3)$.

Matrix Chain Mulitiplication

Say we have 3 matrices A_1, A_2, A_3 which are 30×20, 20×40, and 40×10 respectively. What is the best order to multiply them to minimize the number of mulitiplications?

Note: it takes $m_1m_2m_3$ mulitiplications to multiply 2 matrices that are $m_1 \times m_2$ and $m_2 \times m_3$.

(1) $(A_1A_2)A_3 - 30 \cdot 20 \cdot 40 = 24000$ for A_1A_2. Then to multiply that to A_3 its $30 \cdot 40 \cdot 10 = 12000$ so $24000 + 12000 = 36000$ total.

(2) $A_1(A_2A_3) - 20 \cdot 40 \cdot 10 = 8000$ for A_2A_3. Then to multiply that to A_1 its $30 \cdot 20 \cdot 10 = 6000$ so $8000 + 6000 = 14000$ total.

Algorithm Paradigms

A general approach for constructing algorithms based on a particular concept.

Brute Force

Naive approaches based on going through every possibility without concern of computer resources.

Complexity of Algorithms

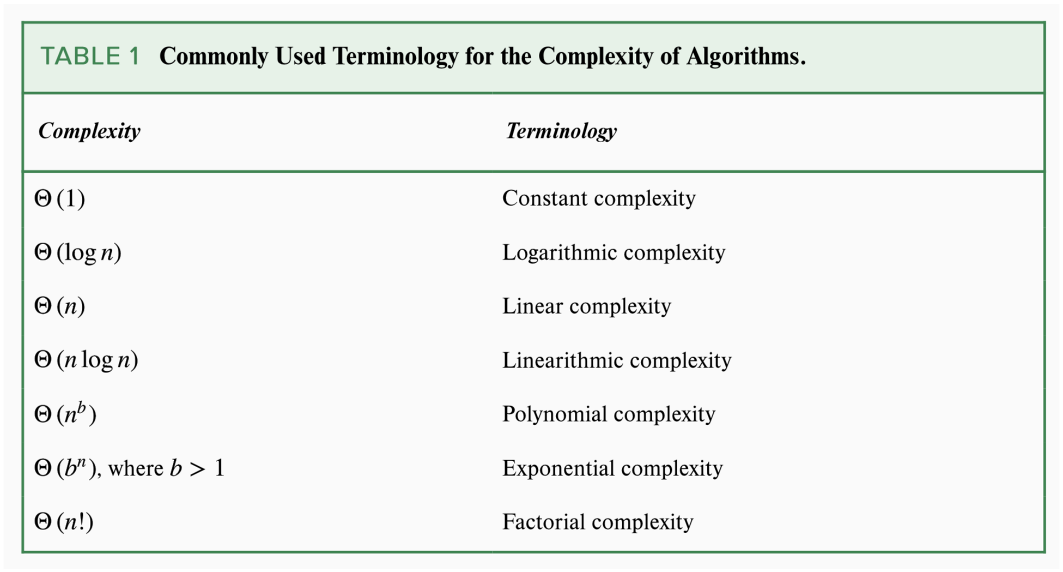

TABLE 1 **Commonly Used Terminology for the Complexity of Algorithms.**

Complexity	*Terminology*
$\Theta(1)$	Constant complexity
$\Theta(\log n)$	Logarithmic complexity
$\Theta(n)$	Linear complexity
$\Theta(n \log n)$	Linearithmic complexity
$\Theta(n^b)$	Polynomial complexity
$\Theta(b^n)$, where $b > 1$	Exponential complexity
$\Theta(n!)$	Factorial complexity

Tractability

A problem solved with an algorithm with polynomial or better complexity is called tractable.

COMPSCI 1DM3

Gulkaran Singh

Winter 2023

1.1 Propositional Logic

Introduction

A proposition is a declarative sentence – meaning it declares a fact – that is either **true** or **false**, but NEVER both. We use **propositional variables** denoted as $p, q, r, s, \ldots$ to represent these propositional statements. The **truth value** of a proposition is either T/F.

Types of Propositions

- **atomic proposition**: simplest form of a proposition (not formed by other propositions)
- **compount propositions**: collection of propositions connected together with logical operators

Logical Operators

Logical operators are used to connect propositions together. The are used to combine atomic and/or compound propositions with one another to form a logical statement.

Negation

The negation of p is like saying **not** p. It is denoted by $\neg p$ (read 'not' p). The truth value also gets flipped. The negation of p is the statement 'it is not the case that p'.

p	$\neg p$
T	F
F	T

Table 1: Truth table for negation of p

Conjunction

The conjunction of p and q is like saying p **and** q. It is denoted by $p \wedge q$. The truth value is only true when both propositions are true.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Table 2: Truth table for the conjunction of $p \wedge q$

Remark: you can also say 'p, but q' to represent a conjunction in English.

Disjunction

The disjunction of p and q is like saying p **or** q. It is denoted by $p \vee q$. The truth value is only false when both propositions are false.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Table 3: Truth table for the disjunction of $p \vee q$

This is the **inclusive** or operator. Both p and q can be true, however, if we want either p or q (exclusive), we use XOR.

Exclusive Or

The exclusive or of p and q is like saying p **xor** q. It is denoted by $p \oplus q$. The truth value is only true when one of the propositions are true, otherwise its false.

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Table 4: Truth table for the exclusive or of $p \oplus q$

Conditionals

The conditional of p and q is like saying **if** p, **then** q. It is denoted by $p \to q$. The truth value is only false when p is **true** and q is **false**. Conditionals are also called **implications**. p is often called the *hypothesis* and q is called the *conclusion*.

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Table 5: Truth table for the conditional of $p \to q$

There are multiple ways to represent conditionals.

"if p, then q" "if p, q" "p is sufficient for q" "p implies q" "p only if q"

"a sufficient condition for q is p" "q whenever p" "q is necessary for p" "q follows from p"

"q provided that p" "q if p" "q when p" "a necessary condition for p is q" "q unless $\neg p$"

Converse, Inverse, and Contrapositive

- The proposition $q \to p$ is the **converse** of $p \to q$.
- The proposition $\neg p \to \neg q$ is the **inverse** of $p \to q$.
- The **contrapositive** of $p \to q$ is $\neg q \to \neg p$ and has the same truth value as $p \to q$, therefore $p \to q \equiv \neg q \to \neg p$.

Biconditionals

The biconditional of p and q is like saying p, **if and only if** q. It is denoted by $p \leftrightarrow q$. The truth value is only true when p and q have the same truth values. Biconditionals are also called bi-implications.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Table 6: Truth table for the biconditional of $p \leftrightarrow q$

Other ways to express bi-conditionals:

"p is necessary and sufficient for q" "if p then q, and conversly" "p iff q" "p exactly when q"

Compound Propositions

We can find the truth value of compound propositions by creating **truth tables**.

p	q	$\neg q$	$(p \vee \neg q)$	$(p \wedge q)$	$(p \vee \neg q) \to (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	T

Table 7: Truth table for $(p \vee \neg q) \to (p \wedge q)$

Precedence of Logical Operators

The order of precedence (from highest to lowest) is negation, conjunction, disjunction, and lastly conditionals/biconditionals.

Therefore this statement $\neg p \to q \vee r \wedge s$ is really $(\neg p) \to (q \vee (r \wedge s))$.

Logic and Bit Operations

A bit can represent truth values with 1 being true and 0 being false. We can apply bit operations on bit strings such as bitwise OR, bitwise XOR, & bitwise AND.

```
01 1011 0110
11 0001 1101
____________
11 1011 1111 bitwise OR
10 1010 1011 bitwise XOR
01 0001 0100 bitwise AND
```

1.2 Applications of Propositional Logic

Logic Circuits

We can model digital circuits from 3 basic circuits called gates.

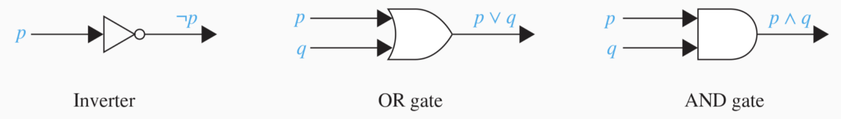

FIGURE 1 **Basic logic gates.**

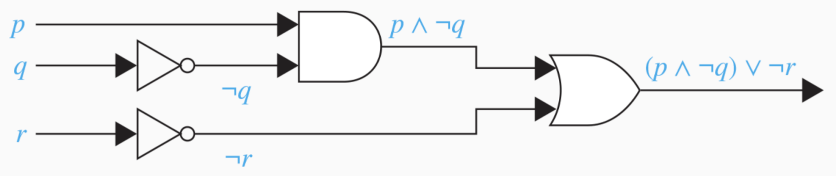

FIGURE 2 **A combinatorial circuit.**

1.3 Propositional Equivalences

- **tautology**: compound proposition that is always true
- **contradiction**: compound proposition that is always false
- **contingency**: neither a tautology nor contradiction

Logical Equivalences

Compound propositions p and q are **logically equivalent** if $p \leftrightarrow q$ is a tautology. The notation for this is $p \equiv q$ or $p \iff q$.

Important Equivalences

Equivalence	Name
$p \wedge T \equiv p$ $p \vee F \equiv p$	Identity Laws
$p \vee T \equiv T$ $p \wedge F \equiv F$	Domination Laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent Laws
$p \vee \neg p \equiv T$ $p \wedge \neg p \equiv F$	Negation Laws
$\neg(\neg p) \equiv p$	Double Negation Laws
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative Laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative Laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive Laws
$\neg(p \vee q) \equiv \neg p \wedge \neg q$ $\neg(p \wedge q) \equiv \neg p \vee \neg q$	De Morgan's Laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption Laws

Table 8: Logical Equivalences

Equivalence
$p \to q \equiv \neg p \vee q \equiv \neg q \to \neg p$
$p \vee q \equiv \neg p \to q$
$p \wedge q \equiv \neg(p \to \neg q)$
$\neg(p \to q) \equiv p \wedge \neg q$
$(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$
$(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$
$(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$
$(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$

Table 9: Logical Equivalences Including Conditional Statements

Equivalence
$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Table 10: Logical Equivalences Including Biconditional Statements

General Distributive Law
$(p \vee s) \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r) \vee (s \wedge q) \vee (s \wedge r)$
$(p \wedge s) \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r) \wedge (s \vee q) \wedge (s \vee r)$

Table 11: General Distributive Laws

Satisfiability

- **satisfiable**: we can assign truth values to the variables to make the proposition true
- **unsatisfiable**: false for all assignments of truth values

1.4 Predicates & Quantifiers

Predicates

Statements that are neither true or false until assigned a value are predicates. For example $x > 3$ is neither T or F until x is specified.

- **propositional function**: P at x where P is the predicate and x is the variable. For example $P(x) = x > 3$

Quantifiers

Quantification tells us the **extent** of which a predicate is true over a range of elements. Note, domains must always be specified.

Universal Quantification

This tells us that **for all** x, the predicate is true. If there is one instance where x is false, the universal quantification is false.

- **notation:** $\forall x P(x)$ reads 'for all x, $P(x)$'
- **counterexample:** an element that makes $\forall x P(x)$ false
- **domain:** typically called domain/universe of discourse. If the domain is empty, $\forall x P(x)$ is true as there are no elements to make it false.
- **finite domain:** $\forall x P(x) \equiv P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_n)$
- **restricted domain:** conditional – $\forall x < 0 (x^2 > 0) \equiv \forall (x < 0 \rightarrow x^2 > 0)$.
- **synonyms:** 'all of', 'for each', 'given any', 'for arbitrary', and 'for any'.

Existential Quantification

This tells us that there is **atleast** one x that makes the predicate is true. If no elements make the predicate true, then the existential quantification is false.

- **notation:** $\exists x P(x)$ reads 'there exists an x such that $P(x)$'
- **domain:** if the domain is false, $\exists x P(x)$ is false as there are no elements to make it true.
- **finite domain:** $\exists x P(x) \equiv P(x_1) \vee P(x_2) \vee \cdots \vee P(x_n)$
- **restricted domain:** conjunction – $\exists x < 0 (x^2 > 0) \equiv \exists (x < 0 \wedge x^2 > 0)$.
- **synonyms:** 'for some', 'for at least one', and 'there is'.

Remark: the quantifiers $\forall$ and $\exists$ have higher precendence than ALL logical operators from propositional calculus.

Negating Quantified Expressions

Flip the quantifier and bring the negation inwards

$$\neg\forall x P(x) \equiv \exists x \neg P(x)$$
$$\neg\exists x P(x) \equiv \forall x \neg P(x)$$

These are called the **De Morgan's Laws for Quantifers**.

1.5 Nested Quantifiers

Just think about these logically and it becomes common sense. Negation of nested quantifers works the same way as before.

Statement	When True?
$\forall x\forall y$ $\forall y\forall x$	$P(x,y)$ is true for all pairs of x,y
$\exists x\exists y$ $\exists y\exists x$	There is a pair of x,y that makes $P(x,y)$ true
$\forall x\exists y$	For every x there is a y that makes $P(x,y)$ true
$\exists x\forall y$	There is a x that makes $P(x,y)$ true for every y

Statement	When False?
$\forall x\forall y$ $\forall y\forall x$	There's a pair x,y that makes $P(x,y)$ false
$\exists x\exists y$ $\exists y\exists x$	$P(x,y)$ is false for every pair x,y
$\forall x\exists y$	There is an x where every y makes $P(x,y)$ false
$\exists x\forall y$	For every x there is a y that makes $P(x,y)$ false

Table 12: Quantification of Two Variables

1.6 Rules of Inference

Proofs are **valid** arguments that establish the truth of statements.

- **valid** $\rightarrow$ conclusion must follow the truth of the **premises**.

Rules of inference act as building blocks to build towards more complicated statements. For example

$$\begin{array}{l} p \\ p \rightarrow q \\ \hline \therefore q \end{array}$$

is the same as $((p \rightarrow q) \wedge p) \rightarrow q$

Fallacies

These are assumptions that lead to incorrect arguments.

- **fallacy of affirming the conclusion**: $((p \to q) \land q) \to p$ is false
- **fallacy of denying the hypothesis**: $((p \to q) \land \neg p) \to \neg q$ is false

1.7 Proofs

Key Definitions

- **theorem**: a mathematical statement sohwn to be true
- **proof**: valid arguments that estable the truth of a theorem
- **axioms/postulates**: statements assumed to be true
- **lemma**: less important theorem
- **conjecture**: mathematical statement assumed to be true, but hasn't been proven yet
- **even integer**: $n = 2k$
- **odd integer**: $n = 2k + 1$
- **perfect square**: $a = b^2$
- **rational**: $r = p/q$ where $q \neq 0$ and $p, q \in \mathbb{Z}$

Direct Proof

Proof method for proving conditionals $p \to q$

(1) assume p to be True

(2) directly 'sub' in p into q

(3) if q is true after assuming p is true, then we have proved $p \to q$

Example: if n is odd, then n^2 is odd.

$$\begin{gathered}\text{assume } p \text{ is true so } n = 2k + 1 \\ \text{then } n^2 \text{ is odd so } n^2 \\ = (2k+1)^2 \\ = 4k^2 + 4k + 1 \\ = 2(2k^2 + 2k) + 1 \\ = 2r + 1 \text{ where } r = 2k^2 + 2k \text{ and } r \in \mathbb{Z} \\ \therefore n^2 \text{ is odd}\end{gathered}$$

Proof by Contraposition

Prove $p \to q$ by proving an equivalent statement $\neg q \to \neg p$

(1) assume q is false

(2) if we get that $\neg p$ is true, then we've proved $\neg q \to \neg p$

Example: if $3n + 2$ is odd, then n is odd.

$$\begin{gathered}
\text{assume } q \text{ is false so } n = 2k \\
\text{then sub into } 3n + 2 \text{ so } 3(2k) + 2 \\
= 6k + 2 \\
= 2(3k + 1) \\
= 2r \text{ where } r = 3k + 1 \text{ and } r \in \mathbb{Z} \\
\neg p \text{ (that 3n+2 is even) is true so} \\
\neg q \to \neg p \text{ is true} \\
\therefore p \to q \text{ is true}
\end{gathered}$$

Basically show that you get $\neg p$ when you assume $\neg q$

Proof by Contradiction

Assume $pand\neg q$ is true, and show that $\neg p$ is true. Since p and $\neg p$ can't both be true, we get a contradiction.

Take our previous example, we assumed $\neg q$ to be false and proved $\neg p$ to be true. With a proof of contradiction we are assuming p to be true, so by proving $\neg p$ to be true, we created a contradiction.

Mistakes in Proofs

- counterexamples will expose mistakes
- **begging the question:** proof is based on the truth of a statement that is being proved

2.1 Sets

- **Roster Method**: list all elements $\{a, b, c, d\}$
- **Set Builder**: $\{x \in \mathbb{Z}^+ \mid x < 10\}$
- **Equal Sets**: 2 sets are equal if they have the same elements $\forall x(x \in A \leftrightarrow x \in B)$. Repetition and order doesn't matter.
- **Empty Sets**: Denoted by $\emptyset$ or $\{\}$, it has no elements in it.

Subsets

To show $A \subseteq B$: if $x \in A$ then x also $\in B$. Simply, every element of A is in B.

Theorem: every nonempty set S has 2 subsets

- the empty set $\emptyset$
- the set S itself

Proper Subsets

To show $A \subset B$

- if $A \subseteq B$ is true
- $A \neq B$ so there is an $x \in B$ not in A

Cardinality

This is the size of a set. The number of distinct elements. It is denoted by $|S|$.

Power Sets

This is the **set** of all subsets of S. Denoted by $\mathcal{P}(S)$. If a set has n elements, its power set has 2^n elements. Think of it as all possible combinations of elements as sets.

Example: Find $\mathcal{P}(0, 1, 2)$ The first two are easy, $\emptyset$ and $\{0, 1, 2\}$ from Theorem 1. Next you can make these combinations: $\{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}$.

$$\therefore \mathcal{P}(0, 1, 2) = \{\emptyset, \{0, 1, 2\}, \{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}\}$$

We can check our answer by seeing we have $2^3 = 8$ elements in the power set.

Cartesian Product

Denoted as $A \times B$, it is the set of all ordered pairs (a, b). Think of it as the distributive property when multiplying.

Example: If $A = \{1, 2\}$ and $B = \{a, b, c\}$, what is $A \times B$

$$A \times B = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\}$$

Notice the cardinality of $A \times B$ is their two individual cardinalities multiplied.

Also denote $A \times A$ as A^2 and $A \times A \times A$ as A^3

Truth Sets

The truth set of a predicate P is the set of elements in the domain D that makes $P(x)$ true

Example: $P(x)$ is $|x| = 1$ and D is $\mathbb{Z}$. What is the truth set?

The truth set is denoted $\{x \in \mathbb{Z} \mid |x| = 1\}$ so it is $\{-1, 1\}$

2.2 Set Operations

- **union:** elements in A or in B so $A \cup B$
- **intersection:** elements in A **and** in B so $A \cap B$
- **difference:** elements in A **not** in B so $A - B$
- **complement:** elements in U not in A so $\overline{A} = U - A$

Multisets

Denoted as $\{m_1 \cdot a_1, m_2 \cdot a_2, \ldots, m_n \cdot a_n\}$

- **cardinality:** sum of the multiplicities $(m_1 + m_2 + \cdots + m_n)$
- **union:** $\max(m_{Ai}, m_{Bi})$
- **intersection:** $\min(m_{Ai}, m_{Bi})$
- **difference:** $\max(m_{Ai} - m_{Bi},\, 0)$
- **complement:** $m_{Ai} + m_{Bi}$

2.3 Functions

$f : A \to B$

- **domain** A is the domain of f
- **codomain** B is the codomain of f
- **image** for $f(a) = b$, b is the image, a is the preimage
- **range** the set of all b's (always a subset of the codomain)

Example: $f : \mathbb{Z} \to \mathbb{Z}$ and $f(x) = x^2$

- domain and codomain are $\mathbb{Z}$
- range is the set of perfect squares $\{1, 4, 9, \ldots\}$

One-to-One / Injective

Think of the Horizontal Bar Test. More formally, it is if f never takes the same value twice. To prove if a function is injective, we must show that

$$\text{if } f(a) = f(b) \text{ then } a = b$$
$$\text{or}$$
$$\text{if } f(a) \neq f(b) \text{ then } a \neq b$$

If a function is strictly increasing, it is injective.

Onto / Surjective

If the range of f is the same as the codomain of f.

Bijective: both one-to-one and onto

Ceiling and Floor Functions

- **Ceiling**: $\lceil x \rceil$ rounds x up
- **Floor**: $\lfloor x \rfloor$ rounds x down

Useful Properties

- $\lceil -x \rceil = -\lfloor x \rfloor$
- $\lfloor -x \rfloor = -\lceil x \rceil$
- $\lceil x + n \rceil = \lceil x \rceil + n$ if $n \in \mathbb{Z}$
- $\lfloor x + n \rfloor = \lfloor x \rfloor + n$ if $n \in \mathbb{Z}$

Know how to graph these for the midterm.

COMPSCI 1DM3

Gulkaran Singh

Winter 2023

4.1 – Divisibility and Modular Arithmetic

We say that a **divides** b if $\frac{b}{a}$ is an integer.

- a is a factor/divisor of b
- b is a multiple of a
- **notation:** $a|b$ (a divides b) or $a\not|b$ (a doesn't divide in b)

Theorem 1 Properties of Divisibility

Let a, b, and c be integers, where $a \neq 0$. Then

(i) if $a|b$ and $a|c$ then $a|(b+c)$
(ii) if $a|b$, then $a|bc$ for all $c \in \mathbb{Z}$
(iii) if $a|b$ and $b|c$, then $a|c$

4.1.3 – Division Algorithm

When we divide an integer by a positive number, there is a quotient and remainder.

Theorem 2 The Division Algorithm

Let $a \in \mathbb{Z}$ and d a positive number, then $a = dq + r$

- d is called the divisor
- a is called the dividend
- q is called the quotient
- r is called the remainder

Theorem 3 Notation

$$q = a \text{ div } d, \quad r = a \text{ mod } d$$

Example 1: What is q and d when -11 is divided by 3

Solution: -11 = 3(-4) + 1 so q = -4 and r = 1

Remark: r must be $0 \leq r < d$, it can never be negative.

4.1.4 – Modular Arithmetic

If a positive integer m divides $a - b$ then we say a is congruent to b modulo m.

- a $\equiv b(\text{mod } m)$ is a congruence and m is its modulus

Theorem 4

$$a \equiv b(\text{mod } m) \leftrightarrow a \text{ mod } m = b \text{ mod } m$$

Example 2: Determine if 17 is congruent to 5 modulo 6

Since 6 divides into $17 - 5 = 12$, then $17 \equiv 5(\text{mod} 6)$

Theorem 5
a and b are only congruent to m if and only if there's an integer k such that $a = b + km$

The set of all integers congruent to a modulo m is called its congruence class.

Theorem 6
If $a \equiv b(\text{mod } m)$ and $c \equiv d(\text{mod } m)$ then

$$a + c \equiv b + d(\text{mod } m) \quad \text{and} \quad ac \equiv bd(\text{mod } m)$$

Notice:

$$(a + b)\text{mod } m = [(a \text{ mod } m) + (b \text{ mod } m)]\text{mod } m$$

$$ab\text{mod } m = [(a \text{ mod } m)(b \text{ mod } m)]\text{mod } m$$

4.1.5 – Arithmetic Modulo m

We can do arithmetic on $\mathbb{Z}_m$, the set of all nonnegative integers less than m.

When we use these operations, we are doing arithmetic modulo m.

- **addition:** $a +_m b = (a+b) \text{mod } m$
- **multiplication** $a \cdot_m b = ab \text{mod } m$

4.2 Integer Representations and Algorithms

We will show given a base b and an integer n, we will construct the base b representations of this number.

4.2.2 Representation of Integers

Base 10 integers are represented as $a_0 \cdot 10^0 + a_1 \cdot 10^1 + a_2 \cdot 10^2 + \ldots$. So an integer like 965 can be broken down into $9 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0$. We can generalize this to ANY base b.

Theorem 7 Base b Expansion of n

$$n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$$

Remark: the base b expansion of n is also often represented as $(a_k a_{k-1} \cdots a_1 a_0)_b$. For example $(1010)_2$ is the base 2 expansion of 10.

HEXADECIMAL EXPANSION

Typically, the base b represents the numbers - 1 that can be included as a_k. For example, for binary the only integers allowed are 0 and 1 to represent binary. For hexadecimal we allow $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$, A, B, C, D, E, and F where A = 10, B = 11, and so on.

Example 3: What is the decimal expansion of $(2AE0B)_{16}$

$(2AE0B)_{16} = 2 \cdot 16^4 + 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16^1 + 11 \cdot 16^0 = 175627$

We can also represent any hexademical digit with 4 bits. For instance, $(11100101)_2 = (E5)_{16}$ because $(1110)_2 = (E)_{16}$ and so on. Note, bytes are 2 hexadecimal digits or 8 bits.

BASE CONVERSION

The general algorithm is to divide by base b and continuously divide the quotients you get by b. Keep track of the remainders as reversing them will be the result.

Example 4: Convert 12345 to base octal expansion

Divide 12345 by 8 to get $12345 = 8 \cdot 1543 + 1$ where 1543 is the next quotient to divide by 8.

Keep doing this and we'll get the remainders 1, 7, 0, 0, and 3. Reverse them and you get your answer $(30071)_8$.

PSEUDOCODE

```
$q := n$
$k := 0$
**while** $q \neq 0$
    $a_k := q$ **mod** $b$
    $q := q$ **div** $b$
    $k := k + 1$
**return** $(a_{k-1}, \cdots, a_1, a_0)$
```

4.2.3 Algorithms for Integer Operations

Defining arithmetic for base 2 expansion

ADDITION

- $1 + 0 = 1$
- $1 + 1 = 0$ carry 1
- $1 + 1$ (and carry 1) $= 1$ carry 1

MULTIPLICATION – works the same as normal multiplication, just remember the addition rules

4.2.4 Modular Exponentiation

- Learning how to compute $b^n \bmod m$ efficiently where b, n, m are large integers.

To compute 3^{11}, notice $11 = (1011)_2$ in binary. So we can do $3^{11} = 3^8 \cdot 3^2 \cdot 3^1$

4.3 Primes and GCDS

- a prime only has factors 1 and itself
- otherwise its called a *composite*

Theorem 8 The Fundamental Theorem of Arithmetic
Every integer greator than 1 can be written as a product of primes (in increasing size).

Examples include $100 = 2 \cdot 2 \cdot 5 \cdot 5 \cdot 5$

4.3.3 Trial Division

- brute force algorithm that determines whether an integer n is prime based on if it is not divisble by any prime $\leq \sqrt{n}$

Theorem 9
If n is a composite integer, then n has a prime divisor less than or equal to $\sqrt{n}$

Using this, if n is a prime, we know it won't be divisible by any primes less than $\sqrt{n}$

You can also find an integers prime factorization by dividing by successive prime factors:

Example 5: Find a prime factorization of 7007

Try 2, 3, and 5 and notice none of them divide 7007. 7 divides 7007 into 1001. Now start at 7, it divides 1001 into 143. Start at 7 again, it doesn't divide 143, but 11 does. We get 11 divides 143 into 13. Since our result (13) is prime, we are done. $7007 = 7 \cdot 7 \cdot 11 \cdot 13$

4.3.4 The Sieve of Eratosthenes

To find all primes not exceeding 100 for example, you start with 2. All integers divisible by 2 other than 2 are deleted. Then 3, 5, and 7. All the numbers left are the primes not exceeding 100.

Theorem 10
There are infinitely many primes

Theorem 11 The Prime Number Theory
The ratio of $\pi(x)$, the number of prime numbers not exceeding x, and $\frac{x}{\ln x}$ approaches 1 as x grows without bound.

$x/\ln x$ tells us approximately how many primes there are not exceeding x. For example,

$$\frac{\pi(10^{10})}{\left(\dfrac{10^{10}}{\ln 10^{10}}\right)} = 1.01729$$

This can help us estimate the probability that a randomly chosen number is prime, that is with $1/\ln n$. So the possibility a number near 10^{1000} is prime is $1/\ln 10^{1000}$ (note, picking an odd number doubles the chances).

4.3.6 Greatest Common Divisor

- the largest integer d such that $d|a$ and $d|b$
- **notation:** $\gcd(a, b)$
- simple method: take the highest common divisors
- **relatively prime:** if the $\gcd(a, b) = 1$
- **pairwise relatively prime** compare all pairs of integers and all of them $= 1$

Example 6: Are 10, 17, and 21 relatively pairwise prime?

gcd(10, 17) = 1 and gcd(10, 21) = 1 so yes

Another way to find gcd is with prime factorizations.

Theorem 12

$$\gcd(a, b) = p_1^{\min(a_1,b_1)} p_2^{\min(a_2,b_2)} \cdots p_n^{\min(a_n,b_n)}$$

Example 7: Find gcd(120, 500)

$$\begin{aligned}
120 &= 2^3 \cdot 3 \cdot 5 \\
500 &= 2^2 \cdot 5^3 \\
\gcd(120, 500) &= 2^{\min(3,2)} 3^{\min(1,0)} 5^{\min(1,3)} \\
\gcd(120, 500) &= 2^2 3^0 5^1 \\
&= 20
\end{aligned}$$

We can also find the least common multiple with prime factorizations

Theorem 13

$$\operatorname{lcm}(a,b) = p_1^{\max(a_1,b_1)} p_2^{\max(a_2,b_2)} \cdots p_n^{\max(a_n,b_n)}$$

Theorem 14 Relationship between gcd and lcm
Let $a, b \in \mathbb{Z}^+$, then

$$ab = \gcd(a,b) \cdot \operatorname{lcm}$$

4.3.7 The Euclidean Algorithm

- more efficient algorithm to compute gcds
- let $a, b, q, r \in \mathbb{Z}$, then $a = bq + r$. Then $\gcd(a,b) = \gcd(a,r)$.
- you continue with successive divisions and use the remainder as b in the gcd.

4.3.8 GCDs as Linear Combinations

Theorem 15 Bezout's Theorem
There exists integers s and t such that $\gcd(a,b) = sa + tb$

- these are called the bezout's coefficients
- bro idek refer to god for help

5.1 Mathematical Induction

Theorem 16 The Principle of Mathematical Induction
Basis Step: Verify $P(1)$ is true.
Inductive Step: prove that $P(k) \rightarrow P(k+1)$ is true.

- $(P(1) \land \forall k(P(k) \rightarrow P(k+1))) \rightarrow \forall n P(n)$
- show that if a random kth case is true AND $k+1$ is true, then its always true

COMPSCI 1DM3

Gulkaran Singh

Winter 2023

Contents

1 Discrete Probability

As always we start with some key definitions

- **sample space**: the *set* of all possible outcomes
- **event**: a *subset* of the sample space

1.1 Finite Probability

Theorem 1.1

If we have a set of *equally* likely events, then we can define the probability of one of those events as such

$$p(E) = \frac{|E|}{|S|}$$

where S is a finite sample size and E an event.

Remember, whenever we calculate probability it is always between 0 and 1 so

$$0 \leq p(E) \leq 1$$

1.2 Probability of Complements

Now, if we've defined the possibility of an event from the sample size occuring, what about the possibility of the **other** events? Well, we can calculate the probability of a *complementary* event (denoted as $\bar{E}$) with the following

Theorem 1.2

Let $\bar{E} = S - E$ denote the complementary event of E, then the probability is given by

$$p(\bar{E}) = 1 - p(E)$$

Proof. Notice $|\bar{E}| = |S| - |E|$,

$$\begin{aligned} p(\bar{E}) &= \frac{|\bar{E}|}{|S|} \\ &= \frac{|S| - |E|}{|S|} \\ &= \frac{|S|}{|S|} - \frac{|E|}{|S|} \\ &= 1 - \frac{|E|}{|S|} \\ &= 1 - p(E) \end{aligned}$$

Recall how the complement of a set A is $U - A$. You should be able to draw the connection with what's going on here. The probablity of the other events occuring is just 1 minus the probability of the chosen event occuring!

1.3 Union of Events

We've been talking about the probability of just one event happening. What about two events? How do we calculate the probability that one *or* another event will occur? What about *both* events occuring?

Theorem 1.3

Let E_1, E_2 be events,

$$p(E_1 \cup E_2) = p(E_1) + p(E_2) - p(E_1 \cap E_2)$$

Comment. You may be confused on where that minus $p(E_1 \cap E_2)$ section of the formula comes from. Well realize that when we calculate the probability of two events happening seperately, we need to account for an overlap of when both events occur. This is made clear with the following example.

Example 1.3.1. What is the probability that a random positive integer less than 100 ($x \in \mathbb{Z}^+ < 100$), is divisble by 9 **or** 11.

We begin by defining E_1 as the event that the number is divisible by 9, and E_2 as being divisble by 11.

$$p(E_1 \cup E_2) = p(E_1) + p(E_2) - p(E_1 \cap E_2)$$

$p(E_1)$ is simple enough to calculate. Starting at 0, 9, 18, ..., 99 we can see there are 12/100 numbers divisible by 9. $p(E_2)$ is also simple. Starting at 0, 11, 22, ..., 99 we can see there are 10/100 numbers divisible by 11. But notice how there are numbers divisible by **both** 9 and 11. Both 0 and 99 are divisble by 9 and 11. So we need to remove this overlap when we calculate the probablity.

$$\begin{aligned} p(E_1 \cup E_2) &= \frac{12}{100} + \frac{10}{100} - \frac{2}{100} \\ &= \frac{20}{100} = \frac{1}{5} \end{aligned}$$

Quickly note that by rearranging Theorem 1.3, we can find the probability of $p(E_1 \cap E_2)$

Theorem 1.4

Let E_1, E_2 be events,

$$p(E_1 \cap E_2) = p(E_1) + p(E_2) - p(E_1 \cup E_2)$$

2 Probability Theory

We've been discussing probabilities of equally likely events, but in this chapter we tackle events that have different chances of occuring. For example, imagine a world where tails was more likely than heads.

2.1 Probability Model/Distribution

The probability distribution is simply a function p from the set of all outcomes. It's made clear with the following example.

Example 2.1.1. What is the probablility distribution of a coin that is biased to land tails twice as much as heads?

Notice that $P(T) = 2P(H)$, from here we can the fact both of these probabilities must add to 1.

$$\begin{aligned} P(T) + P(H) &= 1 \\ (2P(H)) + P(H) &= 1 \\ 3P(H) &= 1 \\ \therefore P(H) &= \frac{1}{3} \\ \therefore P(T) &= \frac{2}{3} \end{aligned}$$

2.2 Union of Disjoint Events

We revisit the union of events, but this time we discuss events that are *pairwise disjoint* events which really just means there's no overlap between them. The textbook gives us this fancy formula

$$p(\bigcup_i E_i) = \sum_i p(E_i)$$

but all this really means is the following

Theorem 2.1

Let E_1, E_2 be pairwise disjoint events,

$$p(E_1 \cup E_2) = p(E_1) + p(E_2)$$

A good example for this is a die. You can only roll one number at a time, there's no overlap as there's no way you can roll two numbers on one die.

2.3 Conditional Probability

Here we discuss the likelihood or probability of an event occuring *based* on a previous event occuring. For example if I generated a random bit string of starting with 0, what's the probability of there being 2 consecutive 0's?

Theorem 2.2

Let E and F be events where $p(F) > 0$, then the conditional probability of E given F is

$$p(E|F) = \frac{p(E \cap F)}{p(F)}$$

Example 2.3.1. Given a random bit string of length 4, starting with 0, what's the probability of there being 2 consecutive 0's?

$2^4 = 16$ means there are 16 possibilities for this random bit string. If the bit string starts with 0, then that narrows us down to 8/16 bit strings as half start with 0 and half start with 1. Notice how $E \cap F = \{0000, 0001, 0010, 0011, 0100\}$ so 5/16 bit strings have a starting bit of 0 and two consecutive 0's somewhere within. Therefore,

$$\begin{aligned} p(E|F) &= \frac{p(E \cap F)}{p(F)} \\ &= \frac{5/16}{8/16} \\ &= \frac{5}{6} \end{aligned}$$

2.4 Independence

A common mistake that people make with conditional probability is trying to calculate with two independent events, where one event happening has no effect on the other. If you flip heads 3 times, what's the probability of the 4th time being tails? Still 1/2 as flipping a coin and the previous results are independent events!

Theorem 2.3

We can say two events are independent if and only if

$$p(E \cap F) = p(E)p(F)$$

Example 2.4.1. Assume a family has 2 children given by {BB, BG, GB, GG}. Are the events of having 2 boys and having atleast 1 boy independent?

We can simply just plug the probabilities into our formula and check!

$$p(E \cap F) = p(E)p(F)$$
$$\frac{1}{4} = \frac{1}{4} \cdot \frac{3}{4}$$
$$\frac{1}{4} \neq \frac{3}{16}$$

so no, these two events are not independent.

2.5 Random Variable

A random variable is essentially a function $p(x = r)$ which is the probability x takes on the value r. This definition is made clear with the following example.

Example 2.5.1. If we flip a coin 3 times, find the distribution of $x(t)$ if $x(t)$ is the number of times heads is flipped.

Flipping a coin 3 times means there are $2^3 = 8$ possibilities {HHH, HHT, HTH, HTT, TTT, TTH, THT, THH}. Let's try to find $p(x = 3)$. This is essentially saying what is the probability of heads being flipped exactly 3 times (as indicated by $x = 3$). If we had $p(x = 2)$, this is the probability of heads being flipped exactly 2 times.

$$\{HHH, HHT, HTH, HTT, TTT, TTH, THT, THH\}$$
$$p(x = 2) = 3/8$$

Now we'd do this for $x = 0, 1, 2,$ and 3 but I'm lazy so do it yourself.

2.6 Binomial Distribution

Say we have only two possible events like flipping a coin. We can label these outcomes as a success (denoted by p) and a failure (denoted by q). We know that $p+q = 1$ and rearranging we get that $q = 1-p$, similar to section 1.2. Now we can calculate the probability of having k successful events occurring from a total n amount of trials.

Theorem 1

The probability of having k successes from n trials is determined by

$$\binom{n}{k} p^k q^{n-k}$$

Note, sometimes this is also denoted as $b(k; n, p)$

Example 2.6.1. Suppose the probability of passing 1DM3 is 92%. What is the probability that exactly 4/5 students pass?

$$p(x = 4) = (0.92)^4(0.08)^1$$

Typically, this is where we would begin. We have a random variable that looks at exactly 4 students passing. We calculated the probability of the first 4 people passing as $(0.92)^4$. Now using $q = 1 - p$ or your brain, you'll calculate the final person failing as having a probability of (0.08). The one aspect we forgot to consider are the possible combinations. For example we can have PPFPP or FPPPP, and so on. That's where the binomial coefficient comes into play.

$$p(x = 4) = \binom{5}{4}(0.92)^4(0.08)$$

Notice how we have 5 total trials, and 4 of those we want to be successes. Now we simply calculate.

$$\begin{aligned}&= \frac{5!}{4! \cdot 1!}(0.92)^4(0.08)\\&= 0.2866\end{aligned}$$

COMPSCI 1DM3

Test Questions

Winter 2023

Problem 1

Prove the following theorem

$$ab = \gcd(a, b) \cdot \operatorname{lcm}(a, b)$$

Start with noting the prime factorization definition of gcd and lcm

$$ab = \left(p_1^{\min(a_1,b_1)} p_2^{\min(a_2,b_2)} \cdots p_n^{\min(a_n,b_n)}\right)\left(p_1^{\max(a_1,b_1)} p_2^{\max(a_2,b_2)} \cdots p_n^{\max(a_n,b_n)}\right)$$

Notice how we can factor out each p_k with $k = 1, 2, \ldots n$.

$$ab = p_1^{\min(a_1,b_1)+\max(a_1,b_1)} p_2^{\min(a_2,b_2)+\max(a_2,b_2)} \cdots p_n^{\min(a_n,b_n)+\max(a_n,b_n)}$$

Now let's notice that we can write the integers a and b in their respective prime factorization form.

$$\begin{aligned} ab &= \left(p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}\right)\left(p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n}\right) \\ &= p_1^{a_1+b_1} p_2^{a_2+b_2} \cdots p_n^{a_n+b_n} \end{aligned}$$

With the same base, we can compare the exponents. For each $k = 1, 2, \ldots n$, we see that

$$\begin{aligned} p_1^{a_1+b_1} p_2^{a_2+b_2} \cdots p_n^{a_n+b_n} &= p_1^{\min(a_1,b_1)+\max(a_1,b_1)} p_2^{\min(a_2,b_2)+\max(a_2,b_2)} \cdots p_n^{\min(a_n,b_n)+\max(a_n,b_n)} \\ a_k + b_k &= \min(a_k, b_k) + \max(a_k, b_k) \end{aligned}$$

which is trivially true, but to complete this proof consider the two cases

$$\begin{aligned} \text{Case 1:} \quad & a_k > b_k \\ & a_k + b_k = b_k + a_k \\ \\ \text{Case 2:} \quad & a_k < b_k \\ & a_k + b_k = a_k + b_k \end{aligned}$$

Problem 2

Use mathematical induction to prove the following

$$\sum_{j=2}^{n} C(j,2) = C(n+1,3) \qquad n \geq 2$$

Let $P(n)$ be the propositional statement that $\sum_{j=2}^{n} C(j,2) = C(n+1,3)$

Basis Step: Prove $P(2)$

$$\sum_{j=2}^{2} C(j,2) = C(2,2) = 1 = C(2+1,3) = 1$$

Inductive Hypothesis:

$$\begin{aligned} P(k) &= \sum_{j=2}^{k} C(j,2) = C(k+1,3) \\ P(k+1) &= \sum_{j=2}^{k+1} C(j,2) = C(k+2,3) \end{aligned}$$

We'll start with $P(k)$ and add the $k+1^{\text{th}}$ term to prove this summation formula.

$$\begin{aligned} P(k+1) &= P(k) + C(k+1,2) \\ &= C(k+1,3) + C(k+1,2) \\ &= \binom{k+1}{3} + \binom{k+1}{2} \end{aligned}$$

Notice at this step, we can use pascal's identity to get our result

$$\begin{aligned} \binom{n}{r} + \binom{n}{r-1} &= \binom{n+1}{r} \\ \binom{k+1}{3} + \binom{k+1}{2} &= \binom{k+2}{3} \\ &= C(k+2,3) \\ &= P(k+1) \\ \therefore P(k) &\rightarrow P(k+1) \end{aligned}$$