I. Introduction

A. Course information
   1. Basic information on handout sheet
   2. Instructor, office hours, TA, homework, exams, finals, grading

B. Course content, goals, and relevance
   1. This course is basically a mathematics course; it teaches tools that are important in computer science.

   [GOAL: To develop tools for analyzing computer algorithms to see if they do what we want [are valid].]

   2. A computer program is a set of logical operations that yields a "true" [actually, valid] answer or an invalid answer or no answer [e.g., it does not terminate or it terminates in an error]. These operations act on sets of data strings. Thus we focus on operations on sets. [Our starting point.]

   3. There is strong natural connection between symbolic logic, which operates on statements that are "true" or "false" and digital logic which operates on strings of bits and yields a "1" or a "0". Hence we focus on symbolic logic as well.
II. Logic

A. Rules of Logic

1. The goal is to "mechanize" the process of logical inference, so that we can readily determine whether the truth of the first part of a sentence implies the conclusion.

   Forms: If A and B then A
   
   Example: If the sun is shining and children are playing then the sun is shining is intuitively TRUE.

   Forms: If A or B then A
   
   If the sun is shining or children are playing then the sun is shining is intuitively NOT TRUE in general.

2. Consider a less trivial example.

   Form: If (p ∨ q) then 'r'
   
   Therefore, if not r, then not p and not q.

   If program syntax is faulty or program divides by zero then program generates an error message.

   Program does not generate an error message, correct.

   Therefore, the program syntax is not faulty and the program does not divide by zero.

3. We start with the undefined notions: sentence, true, and false and define.

   Note: Rosen uses the word statement.

   Definition: A proposition is a sentence that is true or false, but not both.

   This is also called a statement.
Examples:
"If A and B then A" is a proposition
"If A or B then A" is not a proposition
2 + 2 = 4 is a proposition
2 + 1 = 4 is a proposition
"He is a student" is not a proposition

NOTE: Undefined quantities in a sentence generally mean that it is not a statement, unless it is a tautology as in first example above.

4. Compound statements
   a. Notation
      \[ \neg = \text{not} \quad (\text{often written } \sim), \]
      \[ \land = \text{and}, \quad \lor = \text{or} \]
      \[ \neg \text{ precedes } \land \text{ and } \lor \text{ in order of operation} \]

\[ p = \text{it is hot} \]
\[ q = \text{it is sunny} \]

b. Translating English to symbols
(1) It is not hot but it is sunny
   \[ \neg "\text{but}" \text{ and } "\text{and}" \text{ are treated the same} \]
   Our sentence becomes: \( \neg p \land q \)

(2) It is neither hot nor sunny
   Our sentence becomes: \( \neg p \land \neg q \)

Examples are often taken from number theory for concreteness

c. Translating numbers to symbols
   \( x \leq a \) means \( x < a \text{ or } x = a \)
   \( a \leq x \leq b \) means \( a \leq x \text{ and } x \leq b \)

Example: \( p \) means \( 0 < x \)
\( q \) means \( x < 3 \)
\( r \) means \( x = 3 \)
\( x \leq 3 \) becomes \( q \lor r \)
\( 0 < x < 3 \) becomes \( p \land q \)
\( 0 < x \leq 3 \) becomes \( p \land (q \lor r) \)
5. Truth tables

a. \( \neg p \) has the opposite truth value from \( p \). We summarize that in a truth table. [We will see a lot of them.]

\[
\begin{array}{c|c}
 p & \neg p \\
\hline
 T & F \\
 F & T \\
\end{array}
\]

b. Similar truth tables for \( \land \) and \( \lor \) are:

Note

- Connection to intersection and union
- \( p = x \in A \)
- \( q = x \in B \)
- \( p \land q \implies x \in A \land B \)
- \( p \lor q \implies x \in A \cup B \)

\[
\begin{array}{c|c|c}
 p & q & p \land q \\
\hline
 T & T & T \\
 T & F & F \\
 F & T & F \\
 F & F & F \\
\end{array}
\]

\[
\begin{array}{c|c|c}
 p & q & p \lor q \\
\hline
 T & T & T \\
 T & F & T \\
 F & T & T \\
 F & F & F \\
\end{array}
\]

The use of \( \land \) corresponds to "and" in standard usage.
The use of \( \lor \) corresponds to "or" in a non-exclusive sense.

In standard usage, if you say:
"Is it a boy or a girl?", you don't have the sense that it could be both.

In logic, one must distinguish the exclusive and non-exclusive or. The exclusive or is written \( \oplus \), and we will introduce it later.

6. Evaluating truth

a. Definition: A compound proposition is a proposition made up of variables \( (p, q, r) \) and
Connectives (\( \land, \lor, \neg \))

Example: \((p \lor q) \land \neg(p \land q) \equiv p \oplus q\)

We compute the truth of compound statements by setting up a truth table. The truth table will have \(2^n\) entries where \(n\) is the number of variables.

The variables are proposition variables or statement variables.

<table>
<thead>
<tr>
<th>(p)</th>
<th>(q)</th>
<th>(p \lor q)</th>
<th>(p \land q)</th>
<th>(\neg(p \land q))</th>
<th>(p \oplus q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Axioms are needed in math.

Note: Logic does not help determine the truth of component statements, only derived statements.

See Gödel, Escher, Bach by D. R. Hofstadter.

6. Logical equivalence

a. Two statements are logically equivalent if (logically equivalent) - they have the same number of statement variables - the truth tables match exactly.

Example: Dogs bark and cats meow \[ p \equiv q \] are logically equivalent. Cats meow and dogs bark \[ q \equiv p \] are equivalent.

Proof: \( p = \text{dogs bark, } q = \text{cats meow} \)

<table>
<thead>
<tr>
<th>(p)</th>
<th>(q)</th>
<th>(p \land q)</th>
<th>(q \lor p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Both columns are exactly the same.
b. Double negative: \( \neg(\neg p) \equiv p \)

c. An example of non-equivalence
\[ \neg(p \land q) \text{ and } \neg p \land \neg q \text{ are not equivalent} \]

**Proof:**

<table>
<thead>
<tr>
<th>(-p)</th>
<th>(-q)</th>
<th>(p \land q)</th>
<th>((p \lor q))</th>
<th>(-p \lor \neg q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

The columns are not the same.

7. De Morgan's Laws

a. \( \neg(p \land q) \equiv \neg p \lor \neg q \)

**Proof of the first law:**

<table>
<thead>
<tr>
<th>(p)</th>
<th>(q)</th>
<th>(-p)</th>
<th>(-q)</th>
<th>(p \land q)</th>
<th>((p \lor q))</th>
<th>(-p \lor \neg q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Examples:

Negate: "The bus was late or Tom’s watch was slow"
Answer: "The bus was not late and Tom’s watch was not slow"

Negate: \(-1 < x \leq 4\) which is equivalent to \(-1 < x \text{ and } x \leq 4\)
Answer: \(-1 \geq x \text{ or } x > 4\)
8. Tautologies and Contradictions
   a. A tautology is a compound proposition that is always true, regardless of the truth of its propositions.
   b. A contradiction is a compound proposition that is always false.
   c. Any other compound proposition is a contingency.

We designate a tautology with \( \top \), and a contradiction with \( \bot \).

Connection to set theory:

\[
p = x \in A
\]

\[
p \lor \neg p \equiv \top
\]

\[
\neg A \cup \neg A = \emptyset
\]

\[
p \land \neg p \equiv \bot
\]

\[
A \cup A^c = U
\]

Examples:

\[
p \lor \neg p \equiv \top
\]

\[
p \land \neg p \equiv \bot
\]

<table>
<thead>
<tr>
<th>( p )</th>
<th>( \neg p )</th>
<th>( p \lor \neg p )</th>
<th>( p \land \neg p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>( F )</td>
<td>( T )</td>
<td>( F )</td>
</tr>
<tr>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
</tr>
</tbody>
</table>

\( \top \) always true \( \bot \) always false.

d. \( p \land \top \equiv p \) and \( p \land \bot \equiv \bot \)

\( p \lor \top \equiv \top \) and \( p \lor \bot \equiv p \)

<table>
<thead>
<tr>
<th>( p )</th>
<th>( \top )</th>
<th>( \bot )</th>
<th>( p \land \top )</th>
<th>( p \land \bot )</th>
<th>( p \lor \top )</th>
<th>( p \lor \bot )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
<td>( T )</td>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
</tr>
<tr>
<td>( F )</td>
<td>( F )</td>
<td>( T )</td>
<td>( F )</td>
<td>( T )</td>
<td>( F )</td>
<td>( T )</td>
</tr>
</tbody>
</table>
9. Conditional statements

p is the hypothesis
q is the conclusion
\[ p \rightarrow q \]

a. A sentence in the form \( p \rightarrow q \) if \( p \) then \( q \) is called \textit{conditional}
because the truth of \( q \) is conditioned on \( p \).

Examples:

If 30 is divisible by 6 then 30 is divisible by 3

If you drive 60 miles/hour then you will get to Philadelphia before noon

b. If the hypothesis is true and the conclusion is true then the sentence is true.

If the hypothesis is true and the conclusion is false then the sentence is false.

If the hypothesis is false, then we normally say that we cannot tell.
In formal logic, we say that the sentence is true to resolve the ambiguity. This is also referred to as \textit{vacuous truth} or \textit{truth by default}.

We obtain the truth table:

\[
\begin{array}{c|c|c|c}
 p & q & p \rightarrow q \\
 T & T & T \\
 T & F & F \\
 F & T & T \\
 F & F & T \\
\end{array}
\]

c. In the order of operations, \( \rightarrow \) is last. It follows after \( V, \Lambda \), which follow after \( \sim \).
d. Why this unintuitive definition?
   It is useful for quantified statements.

   For all real \( x \), if \( x > 2 \) then \( x^2 > 4 \)

   We want this statement to be true for all real \( x \). There are three cases:
   
   (1) \( x > 2 \) (hypothesis true), \( x^2 > 4 \) (conclusion true)
   (2) \(-2 \leq x \leq 2 \) (hypothesis false), \( x^2 < 4 \) (conclusion false)
   (3) \( x < -2 \) (hypothesis false), \( x^2 > 4 \) (conclusion true)

   The only way to have this statement be true for all \( x \) is to have the definition we have given.

   e. We also have the equivalence
   \[ p \rightarrow q \equiv \neg p \lor q \]

   Either you get to work on time or you are fired
   \( p \) = You don't get to work on time
   \( q \) = You are fired

   The equivalent statement is:
   If you don't get to work on time, then you will be fired

10. Other conditional forms
a. Negation
   \[ \neg (p \rightarrow q) \equiv p \land \neg q \]

   Proof: \[ \neg (p \rightarrow q) \equiv \neg (p \land \neg q) \equiv p \land \neg q \]
   definition de Morgan's law
Example: If I drive 60 miles/hr then I will get to Philadelphia.

I drive 60 miles per hour and I do not get to Philadelphia.

6. Contrapositive
The contrapositive of \( p \rightarrow q \)
is \( \neg q \rightarrow \neg p \)
\[ p \rightarrow q \equiv \neg q \rightarrow \neg p \]

If I do not get to Philadelphia then I do not drive 60 miles/hr.

C. Converse and inverse
(1) Converse of \( p \rightarrow q \): \( q \rightarrow p \)
Inverse of \( p \rightarrow q \): \( \neg p \rightarrow \neg q \)

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
p & q & p \rightarrow q & \neg q \rightarrow \neg p & q \rightarrow p & \neg p \rightarrow \neg q \\
\hline
T & T & T & T & T & T \\
T & F & F & F & T & T \\
F & T & T & T & F & T \\
F & F & T & T & T & T \\
\hline
\end{array}
\]

(2) Statement and contrapositive are equivalent, inverse and converse are equivalent.

BUT

(Statement and contrapositive) are not equivalent to the (inverse and converse).

Converse: If I get to Philadelphia then I drive 60 miles/hr.
Inverse: If I do not drive 60 miles/hr then I do not get to Philadelphia.
3. Only if and the biconditional
   a. \( p \text{ only if } q \) means \( \neg q \rightarrow \neg p \)
   or \( p \rightarrow q \)

   By contrast, \( p \text{ if } q \) means \( q \rightarrow p \)
   or \( \neg p \rightarrow \neg q \)

   These are not equivalent.

b. The biconditional of \( p \) and \( q \) is
   \( p \text{ if and only if } q \) or \( p \iff q \)

   True when \( p \) and \( q \) have the same truth values
   False when they have opposite truth values

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>( p \iff q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

\( p \iff q \) is designated \( p \leftrightarrow q \)

The order of \( \leftrightarrow \) is the same as \( \rightarrow \); it appears after \( \land \) and \( \lor \)

Note \( p \leftrightarrow q = (p \rightarrow q) \land (q \rightarrow p) \)

In practice, one proves a biconditional statement by proving \( p \rightarrow q \) and \( q \rightarrow p \) separately.

This terminology is often used in mathematics:

C. \( r \) is a sufficient condition for \( s \)
   means \( r \rightarrow s \)

\( r \) is a necessary condition for \( s \)
   means \( \neg r \rightarrow \neg s \)

\( r \) is a necessary and sufficient condition for \( s \)
   means \( r \leftrightarrow s \)
Examples:
Your birth is a sufficient condition for your death.
If you are born, then you will die.

Your birth is a necessary condition for your death.
If you are never born, then you will never die.

d. Note: In common language, we often say "if" when we mean "if and only if."

"If you eat dinner, then you will get dessert."

When we say that to a child, we mean that they have to eat dessert.
B. Logic of Quantified Statements

1. How do we deal with an argument like:
   All human beings are mortal
   Socrates is a human being
   Therefore Socrates is mortal
   Intuitively, it is correct, but we have to learn how to deal with quantifiers like all and some.

2. A predicate is a sentence that contains a finite number of variables and becomes a statement when specific values are substituted for the variables. The domain is the set of all values that may be substituted for the variables.

   a. Example: "x is a student at y"
   The domain of x is the set of all students in the United States
   The domain of y is the set of all universities in the United States.
   We would write this statement as Q(x, y). It is a logical function whose values are T or F for every x and y. [It is exactly like a mathematical function in that produces a unique value for every x and y.]
The predicate is not well-defined until its domain has been defined.

b. If \( P(x) \) is a predicate and \( x \) has domain \( D \), then the truth set is all elements of \( D \) that make \( P(x) \) true:

\[
\text{truth set} = \{ x \in D \mid P(x) \}
\]

c. Notation: Let \( P(x) \) and \( Q(x) \) have a common domain \( D \)

\( P(x) \Rightarrow Q(x) \)

means that the truth set of \( P(x) \) is a subset of the truth set of \( Q(x) \)

\( P(x) \Leftrightarrow Q(x) \)

means that the truth sets of both are identical

Example: Let \( T[P(x)] \equiv \text{truth set of } P(x) \)

\( P(x) = \text{"x is a factor of 8"} \)
\( Q(x) = \text{"x is a factor of 4"} \)
\( R(x) = \text{"x < 5 and x \neq 3"} \)
\( D = \mathbb{Z}^+ \) (the set of positive integers)

\( C = \text{is a proper subset of...} \)
3. Universal and existential statements
   a. A universal statement has the form: \( \forall x \in D, Q(x) \)
      where \( Q(x) \) is a predicate with domain \( D \). A value of \( x \) for which \( Q(x) \) is false is called a counterexample.

   b. Example: \( Q(x) = \forall x \in D, x^2 \geq x \)
      Suppose \( D = \{1, 2, 3, 4\} \)
      then this statement is true.
      We may proceed exhaustively:
      \( 1^2 = 1 \geq 1, \ 2^2 = 4 \geq 2, \ 3^2 = 9 \geq 3, \ 4^2 = 16 \geq 4 \)
      Suppose \( D = \mathbb{R} \), then this statement is false.
      We proceed by finding a counterexample:
      \( \exists \frac{1}{2} \in \mathbb{R}; \ (\frac{1}{2})^2 = \frac{1}{4} < \frac{1}{2} \).

   c. Note: The method above is called the method of exhaustion and is applicable when the domain is finite.

   d. An existential statement has the form: \( \exists x \in D \) such that \( Q(x) \)
      It is true iff \( Q(x) \) is true for at least one element in \( D \). It is false iff \( Q(x) \) is false for all elements in \( D \).

   e. Example: \( Q(m) = \exists m \in D, m^2 = m \)
Proposition
Suppose $D = \mathbb{Z}$, then this statement is true.
$1 \in \mathbb{Z}$ and $1^2 = 1$

Proposition
Suppose $D = \{5, 6, 7\}$, then this statement is false.
We proceed exhaustively.
$5^2 = 25 \neq 5$, $6^2 = 36 \neq 6$, $7^2 = 49 \neq 7$

f. Translating between formal and informal language.
It is important to develop facility with this.

Examples: Formal to informal

$\forall x \in \mathbb{R}, x^2 \geq 0$ becomes
All real numbers have non-negative squares.

$\exists m \in \mathbb{Z}, m^2 = m$ becomes
There is an integer whose square equals itself.

Informal to formal:
All triangles have three sides
$\forall t \in T$, it has three sides (T = set of all triangles).

Some dogs have spots
$\exists d \in D$, it has spots (D = set of all dogs).

4. Universal conditional statements

$\forall x, P(x) \rightarrow Q(x)$
These are arguably the most important kind of statements in mathematics.

Example:
$\forall x \in \mathbb{R}, x > 2 \rightarrow x^2 > 4$
Or if a real number is greater than 2,
then its square is greater than 4.
5. Equivalent forms
   a. \( \forall x \in U, \ P(x) \rightarrow Q(x) \) is equivalent to
      \( \forall x \in D, \ Q(x) \)
      if \( D \) is the truth set of \( P(x) \)

   Example:
   \( \forall x \in \mathbb{R}, \ x \in \mathbb{Z} \rightarrow x \in \mathbb{Q} \)
   and \( \forall x \in \mathbb{Z}, \ x \in \mathbb{Q} \) are equivalent

   b. \( \exists x \in U, \ P(x) \land Q(x) \) is equivalent to
      \( \exists x \in D, \ Q(x) \)
      if \( D \) is the truth set of \( P(x) \)

   Example:
   "There is an integer that is both prime and even" becomes "There is a prime that is even"

6. Implicit quantification.
   Quantifiers like "for all", "there exists" and so on are often left out of problem statements and
   must be inferred - even in mathematical writing

   "If \( x > 2 \) then \( x^2 > 4 \)" really means
   \( \forall x \in \mathbb{R}, \ x > 2 \rightarrow x^2 > 4 \)

   "24 can be written as the sum of two integers." really means \( \exists m, n \in \mathbb{Z}, \ m + n = 24 \)

   One must look out for hidden quantifiers!
7. Negation
   a. The negation of $\forall x \in D, Q(x)$ is $\exists x \in D, \neg Q(x)$
      Formally: $\neg(\forall x \in D, Q(x)) \equiv \exists x \in D, \neg Q(x)$
      The negation of a universal statement "all are..." is equivalent to "some are not..."

   b. The negation of $\exists x \in D, Q(x)$ is $\forall x \in D, \neg Q(x)$
      Formally: $\neg(\exists x \in D, Q(x)) \equiv \forall x \in D, \neg Q(x)$
      The negation of an existential statement "some are..." is equivalent to "all or not..."

   Examples: $\forall$ primes $p$, $p$ is odd
              Neg: $\exists$ a prime $p$, $p$ is not odd

              No politicians are honest $\iff$ $\forall$ politicians $x$, $x$ is not honest
              Neg: Some politicians are honest: $\exists$ a politician $x$, $x$ is honest

   C. In spoken language, "not" is ambiguous, it can mean "not some" or "not all".
      You have to be careful when translating into formal language.
d. Negations of universal equivalent statements

\[ \neg (\forall x, P(x) \rightarrow Q(x)) \equiv \exists x, \neg (P(x) \rightarrow Q(x)) \]

So much is obvious from the foregoing.

From the relationship \( \neg (P \rightarrow Q) \equiv P \land \neg Q \)

We also have

\[ \neg (\forall x, P(x) \rightarrow Q(x)) \equiv \exists x, P(x) \land \neg Q(x) \]

Example:

Any person who is blond has blue eyes

Neg: There are some people who are blond and don’t have blue eyes.

8. Vacuous truth of universal statements

“All Martians are green”

Neg: There exists a Martian that isn’t green.

But Martians don’t exist, so the negation must be false.

But, then, the statement must be true. We refer to it as “vacuously true.”

9. Multiple quantifiers

a. Many statements in mathematics contain more than one quantifier, as do statements in everyday language.

Positive

\[ \forall \text{ positive numbers } x, \exists \text{ another } y, y < x \]

This is true, which becomes: “Given any positive number, we can find a smaller positive number.”

\[ \exists \text{ a positive number } x, \forall \text{ positive numbers } y, y < x \]

This is not true, which becomes: There is a positive number that is larger than all other positive numbers.
Clearly, these statements are very different. Consider another example in everyday language:

"Everybody loves somebody"

"Somebody loves everybody"

Which translate to informal language:

\[ \forall \text{people } x, \exists \text{ a person } y, \text{ such that } x \text{ loves } y \]

\[ \exists \text{ a person } x, \forall \text{ people } y, \ x \text{ loves } y \]

b. Example: Three students, Adam, Betty, and Claire, register for five subjects (Computer Science, Physics, History, English, French).

Adam takes Computer Science, History, French

Betty takes Physics, English, French

Claire takes Computer Science, English, French

\[ S = \{ A, B, C \} = \text{set of all students} \]

\[ T = \{ CS, Ph, Hi, En, Fr \} = \text{set of all subjects} \]

\[ \exists x \in T, \forall y \in S, y \text{ takes } x \]

Proof: By example from \( T \)

By exhaustion from \( S \)

Set \( x = Fr \)

\( y = A \), \( A \) takes \( Fr \);

\( y = B \), \( B \) takes \( Fr \);

\( y = C \), \( C \) takes \( Fr \);

Hence, the statement is true.

c. Negation

\[ \neg (\forall x, \exists y, P(x,y)) \equiv \exists x, \forall y, \neg P(x,y) \]

\[ \neg (\exists x, \forall y, P(x,y)) \equiv \forall x, \exists y, \neg P(x,y) \]
Examples: The negation of "everybody love somebody" is "There is somebody who does not love anybody" (first case).
The negation of "somebody loves everybody" is "Everybody has somebody that they don't love."

10. Relationship of $\forall$ to $\land$ and $\exists$ to $\lor$

a. With a finite domain $D = \{x_1, x_2, \ldots, x_n\}$, we have the equivalence relationship
\[\forall x \in D, \ Q(x) \equiv Q(x_1) \land Q(x_2) \land \ldots \land Q(x_n)\]

Example: $D = \{0, 1\}$
\[\forall x \in D, \ x^2 = x \equiv (0^2 = 0) \land (1^2 = 1)\]

b. With a finite domain $D = \{x_1, x_2, \ldots, x_n\}$, we also have the equivalence relationship
\[\exists x \in D, \ Q(x) \equiv Q(x_1) \lor Q(x_2) \lor \ldots \lor Q(x_n)\]

Example: $D = \{0, 1\}$
\[\exists x \in D, \ x + x = x \equiv (0 + 0 = 0) \lor (1 + 1 = 0)\]

II. Variants of universal conditional statements proposition

As is the case for standard conditional statements, universal conditional statements have a contrapositive, converse, and inverse statement:

statement: $\forall x \in D, \ P(x) \rightarrow Q(x)$
contrapositive: $\forall x \in D, \ \neg Q(x) \rightarrow \neg P(x)$
converse: $\forall x \in D, \ Q(x) \rightarrow P(x)$
inverse: $\forall x \in D, \ \neg P(x) \rightarrow \neg Q(x)$
Example:

- Statement: \( \forall x \in \mathbb{R}, \ x^2 \leq 4 \rightarrow x \leq 2 \)
- Contrapositive: \( \forall x \in \mathbb{R}, \ x > 2 \rightarrow x^2 > 4 \)
- Converse: \( \forall x \in \mathbb{R}, \ x \leq 2 \rightarrow x^2 \leq 4 \)
- Inverse: \( \forall x \in \mathbb{R}, \ x^2 > 4 \rightarrow x > 2 \)

Proposition

As before, statement \( \equiv \) contrapositive and
Proposition

converse \( \equiv \) inverse, but statement \( \neq \) converse
Proposition

and statement \( \neq \) inverse.
Proposition

In this case, the statement and contrapositive
Proposition

are true. The converse and inverse are false.
Proposition

12. The notion of necessary and sufficient
Proposition

conditions can be extended to universal
Proposition

conditional statements:
Proposition

\( \forall x, \ r(x) \text{ is a sufficient condition for } s(x) \)
Proposition

mean: \( \forall x, \ r(x) \rightarrow s(x) \)
Proposition

\( \forall x, \ r(x) \text{ is a necessary condition for } s(x) \)
Proposition

means \( \forall x, \neg r(x) \rightarrow \neg s(x) \)
Proposition

\( \forall x, \ r(x) \text{ only if } s(x) \)
Proposition

means \( \forall x, \neg s(x) \rightarrow \neg r(x) \equiv \forall x, \ r(x) \rightarrow s(x) \)
Proposition

(Hence, \( r(x) \) is a sufficient condition for \( s(x) \))

Example: To play chess, it is necessary to know the rules of
Proposition

chess: \( \forall \text{ people } x, \ x \text{ plays chess } \rightarrow x \text{ knows the rules of chess} \)
Proposition

Note: the reversal from: knowing the rules of chess
Proposition

is not a necessary condition for playing chess.
C. Methods of Proof

1. The goal of mathematics is to obtain exact statements by reasoning in a chain that starts from axioms or postulates and uses valid rules of inference to obtain theorems. Fallacies are forms of incorrect reasoning — some of which are very common. Lemmas are theorems that must be proved prior to a major result. Corollaries are theorems that follow almost immediately from a major result. The boundary between theorems, lemmas, and corollaries is in the eye of the beholder. A conjecture is a proposition whose truth is unknown.

2. Rules of inference (Summarized in Rosen Table 1, p. 58)
   a. Modus ponens
     \[
     \begin{align*}
     p & \Rightarrow q \\
     p \Rightarrow q & \\
     \therefore q
     \end{align*}
     \]
     Example: \( p: n > 3 \)
     \( p\Rightarrow q: n > 3 \Rightarrow n^2 > 9 \)
     \( \therefore q: \therefore n^2 > 9 \)
   b. Modus tollens (proof by contradiction)
     \[
     \begin{align*}
     \neg q & \\
     p \Rightarrow q & \\
     \therefore \neg q
     \end{align*}
     \]
     Note that modus tollens works on, while modus ponens works on \( p \) in the upper half.
c. Other rules

Addition       Simplification       Conjunction
\[ \frac{p}{p \lor q} \qquad \frac{p \land q}{p} \qquad \frac{p}{p \land q} \]

Hypothetical syllogism       Disjunctive syllogism       Resolution
\[ \frac{p \Rightarrow q \qquad q \Rightarrow r}{p \Rightarrow r} \qquad \frac{p \lor q}{\neg p} \qquad \frac{p \lor q \lor r}{\neg p} \]

3. Valid arguments

A. An argument is valid whenever the truth of the hypotheses \( p_1, p_2, \ldots, p_n \) proves the truth of the conclusion \( q \) \((p_1, p_2, \ldots, p_n) \Rightarrow q\)

6. Example:

Hypotheses
(1) It is not sunny this afternoon and it is colder than yesterday.
(2) We will go swimming only if it is sunny.
(3) If we do not go swimming, then we will take a canoe trip.
(4) If we take a canoe trip we will be home by sunset.

Conclusion:
We will be home by sunset.

Propositions:
p: It is sunny this afternoon.
q: It is colder than yesterday.
V: We will go swimming  
S: We will take a canoe trip  
T: We will be home by sunset

Hypotheses:  
(1) \( \neg p \land q \)  
(2) \( r \rightarrow p \)  
(3) \( \neg r \rightarrow s \)  
(4) \( s \rightarrow t \)  

Conclusion: \( t \)

Proof: 
1. \( \neg p \land q \) hypothesis  
2. \( \neg p \) simplification using 1  
3. \( r \rightarrow p \) hypothesis  
4. \( \neg r \)  
5. \( \neg r \rightarrow s \) hypothesis  
6. \( s \) modus ponens from 4 and 5  
7. \( s \rightarrow t \) hypothesis  
8. \( t \) modus ponens from 6 and 7

4. Fallacies:  
   a. Affirming the conclusion or Begging the question or circular reasoning  
      Is the doctor a man?  
      Doctors must be human  
      Men are human  
      Therefore doctors are men

   b. Denying the hypothesis or False generalization  
      All men are human  
      Women aren't men  
      Women aren't human  
      Women aren't doctors
5. Rules of inference for quantified statements

Universal instantiation

\[ \forall x \ P(x) \]
\[ \therefore \ P(c) \]

Existential instantiation

\[ \exists x \ P(x) \]
\[ \therefore \ P(c) \text{ for some } c \]

Universal generalization

\[ P(c) \text{ for an arbitrary } c \in D \]
\[ \therefore \ \forall x \ P(x) \]

Existential generalization

\[ P(c) \text{ for some element } c \]
\[ \therefore \ \exists x \ P(x) \]

Example:

\[ \forall x \in \mathbb{R}, \ x^2 > 0 \text{ (hypothesis)} \]
\[ 1 = 1^2 > 0 \text{ (follows by universal instantiation)} \]

6. Methods of proof

a. Direct proof – example

Definition: An integer \( n \) is even if \( \exists \) an integer \( k \) such that \( n = 2k \). An integer \( n \) is odd if it is not even (which implies \( n = 2k + 1 \) for some \( k \)).

Theorem: If \( n \) is an odd integer, then \( n^2 \) is an odd integer.

Proof:

\[ n = 2k + 1 \text{ for some } k \]

Hence,

\[ n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 \]

\[ \Rightarrow n^2 = 4(k^2 + k) + 1 \]

Since \( k^2 + k \) is an integer (Why?),

\[ n^2 \text{ is odd} \]

b. Indirect proof – an example

We prove the contrapositive. If we want \( p \Rightarrow q \), we prove \( \neg q \Rightarrow \neg p \).

Theorem: If \( 3n + 2 \) is odd, then \( n \) is odd.
Proof: Suppose \( n \) is even, then \( n = 2k \) for some integer. Thus, \( 3n + 2 = 6k + 2 = 2(3k + 1) \) is even. Hence, the contrapositive holds and the theorem must hold.

### c. Proof by contradiction

Suppose we can show \( \neg p \rightarrow F \) then \( p \) must hold. For example, if we can show \( \neg p \rightarrow (r \land \neg r) \) for any statement, then \( p \) must hold.

**Theorem:** \( \sqrt{2} \) is irrational

Suppose \( \neg p \), i.e., \( \sqrt{2} \) is rational.

Then, we must have \( \sqrt{2} = \frac{a}{b} \) where \( a \) and \( b \) are relatively prime. Then, we have \( 2b^2 = a^2 \).

Hence, \( a^2 \) is even, which means that \( a \) is even. Since \( a \) is even, \( a = 2c \) \( \Rightarrow a^2 = 4c^2 \) and \( b^2 = 2c^2 \).

Hence, we cannot express \( \sqrt{2} \) as the quotient of two relatively prime integers and it cannot be rational.

Note: An indirect proof can always be rewritten as a proof by contradiction. The opposite is not true.

### d. Proof by cases

\[
[(p \lor p \lor \ldots \lor p) \rightarrow q] \iff [(p \rightarrow q) \land (p \rightarrow q) \land \ldots \land (p \rightarrow q)]
\]

### e. Proof of equivalence

\[(p \iff q) \iff [(p \rightarrow q) \land (q \rightarrow p)]\]
7. Theorems with quantifiers
   a. Existence proof: \( \exists x \ P(x) \)
      Example: There is a positive integer that can be written as the sum of a cube in two different ways.
      Proof: \( 1729 = 10^3 + 9^3 = 12^3 + 1^3 \)
      This proof is constructive.
      It is also possible to have non-constructive proofs, at the end of which you know that it exists, but you don't know what it is.

   b. Uniqueness proofs: We show that only one element has the property of interest.
      Example: Theorem: Every integer has a unique additive inverse.
      Proof: Suppose the opposite.
      Given a \( p \), there are two integers \( r \) and \( s \) such that \( r + s = 0 \) and \( p + r = 0 \) and \( p + s = 0 \).
      In this case, \( p + r = p + s \) which implies \( r = s \), which contradicts our hypothesis.
      Note: Most uniqueness proofs use contradiction.
D. Digital Logic
1. Modern computers are based on transistors.
   a. These transistors are forced by using positive feedback into putting out high voltage (e.g., 5 volts) or low voltage (e.g., 0 volts).
   b. These states are conventionally designated 1 and 0 and are used to represent digital bits.
   c. Central processing units are built to carry out complex logical operations to add, subtract, multiply, divide, as well as carry out the logical operations that we used.
   d. Design is not based on transistors; instead it is based on digital logic gates. These gates correspond to our basic logical operations with $T \to 1$ and $F \to 0$. They have standard graphical representations.

2. Basic logic gates
   a. AND gate
      \[
      \begin{array}{ccc}
      & P & \rightarrow & R \\
      Q & \rightarrow & R
      \end{array}
      \]
      \[
      \begin{array}{c|cc|cc}
      P & 0 & 0 & 0 & 0 \\
      Q & 0 & 1 & 0 & 0 \\
      & 1 & 0 & 0 & 0 \\
      & 1 & 1 & 1 & 1
      \end{array}
      \]
   b. OR gate
      \[
      \begin{array}{ccc}
      & P & \rightarrow & R \\
      Q & \rightarrow & R
      \end{array}
      \]
      \[
      \begin{array}{c|cc|cc}
      P & 0 & 0 & 0 & 0 \\
      Q & 0 & 1 & 0 & 0 \\
      & 1 & 0 & 1 & 1 \\
      & 1 & 1 & 1 & 1
      \end{array}
      \]
c. NOT gate

\[
P \rightarrow \overline{R}
\]

<table>
<thead>
<tr>
<th>P</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

d. Just as with logic expressions, we can trace the behavior for any input by constructing a truth table or input/output table.

Note \( \overline{A} \) is the opposite of \( A \):

- \( A = 1 \Rightarrow \overline{A} = 0 \)
- \( A = 0 \Rightarrow \overline{A} = 1 \)
- \( \overline{\overline{A}} = A \)

3. One can construct Boolean expressions (logical expressions) with \( \land \lor \neg \) to express the output of a combination of digital gates in complete analogy to our construction of logical statements. Example:

\[
\begin{align*}
P & \\
\overline{Q} & \\
\overline{\overline{(P \lor Q) \land \neg (PAQ)}}
\end{align*}
\]

Truth table or input/output table:

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P\overline{Q}</th>
<th>\overline{PAQ}</th>
<th>\neg(\overline{PAQ})</th>
<th>(P\lor Q) \land \neg(\overline{PAQ})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
4. Conversely, one can design circuits to carry out logical operations which is the essence of computer design.

a. from a Boolean expression

Example 1:

\((\neg P \land Q) \lor \neg Q\)

1. Make \(\neg P\) from \(P\) using NOT gate
2. Make \(\neg P \land Q\) using AND gate
3. Make \(\neg Q\) from \(Q\) using NOT gate
4. Make output using OR gate

Example 2: \((P \land Q) \lor (R \land S)) \lor T\)

The order corresponds to:

One represents this case usually with a multiple input AND gate.

Similarly, there are multiple input OR gates that yield 0 iff all inputs are 0.
b. From a truth table

1. Look for rows where \( S = 1 \)
   (In this case, row 1, 3, and 7)

2. In an AND statement, combine
   \( X \) if \( X = 1 \) and \( -X \) if \( X = 0 \)

   Rows 1: \((P \land Q \land R)\)
   Rows 3: \((P \land -Q \land R)\)
   Rows 7: \((P \land -Q \land -R)\)

   These combinations produce
   1 only if \( P, Q, R \) equal the
   values in the rows

3. In an OR statement, combine the row statements

   \[ S = (P \land Q \land R) \lor (P \land -Q \land R) \lor (P \land -Q \land -R) \]

   This approach always works but it does not usually
   yield the simplest representation.

   It is called the **disjunctive normal form**.

5. Simplification

   a. This is an art! There is no deductive procedure
      that yields the simplest possible circuit.

      How much energy to put into simplification
      depends on 1) cost of transistors, 2) cost of
      transistor real estate, 3) cost of development.

      For VLSI, simplification is less important.

      For high-speed circuits, it is more important

b. Example:

\[ (P \land -Q) \lor (P \land Q) \]

Circuit (a)

\[ ((P \land -Q) \lor (P \land Q)) \land Q \]

Circuit (b)

\[ P \land Q \]
2.32

Proof of equivalence:

(1) Truth Table

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P∧¬Q</th>
<th>PΛQ</th>
<th>(P∧¬Q)V(PΛQ)</th>
<th>((P∧¬Q)V(PΛQ))ΛQ</th>
<th>PΛQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

(2) Boolean logic

\[
((P∧¬Q)V(PΛQ))ΛQ \\
≡ (P∧¬Q)VQ ΛQ \quad \text{(distribution law)} \\
≡ PΛQ \quad \text{(negation law)} \\
≡ PΛQ \quad \text{(identity law)}
\]

6. Computer addition

a. In ordinary arithmetic, we use the decimal system

\[
321 = 3 \times 100 + 2 \times 10 + 1
\]

This system has 10 digits 0, 1, \ldots, 9

b. Computers use binary arithmetic, because everything is represented in groups of 1 and 0

\[
110101 = 1 \times 32 + 1 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 + 1 = 53_{10}
\]

6. The basic addition of two digits is given by

\[
\begin{align*}
1_2 + 1_2 &= 10_2 \\
1_2 + 0_2 &= 01_2 \\
0_2 + 1_2 &= 01_2 \\
0_2 + 0_2 &= 00_2
\end{align*}
\]

which has a carry and a sum

Representing these two digits as \( P \) and \( Q \), we see that the carry = \( PΛQ \)

\[
\text{sum} = (P∧Q)∧(PΛQ) = P⊕Q
\]
b. To implement a full adder, we must include a carry from a previous stage. So, we must add three digits. The largest possible number is $1_2 + 1_2 + 1_2 = 11_2$, in which case $C = 1$ and $S = 1$; we do not need a third digit.

**Full Adder**

![Diagram of Full Adder](image)

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>C1</th>
<th>S1</th>
<th>C2</th>
<th>S</th>
<th>C1V C2</th>
<th>C</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Verification from the truth table.