Discrete Computational Structures

12140204

Anas Melhem
Palestine Technical University

The Foundations: Logic and Proofs

Chapter 1, Part II: Predicate Logic

Predicates and Quantifiers

Section 1.4

Section Summary

- Predicates
- Variables
- Quantifiers
  - Universal Quantifier
  - Existential Quantifier
- Negating Quantifiers
  - De Morgan's Laws for Quantifiers
- Translating English to Logic
- Logic Programming (*optional*)

Propositional Logic Not Enough

- If we have:

  "Every computer connected to the university network is functioning properly."

No rules of propositional logic allow us to conclude the truth of the statement:

"MATH-PC2 is functioning properly."

- Can't be represented in propositional logic. Need a language that talks about objects, their properties, and their relations.

Definition of Predicate Logic

- ***Predicate logic:*** *is a more powerful type of logic* can be used to express the meaning of a wide range of statements in mathematics and computer science in ways that permit us to reason and explore relationships between **objects**.
- "$x > 3$," "$x = y + 3$," "$x + y = z$,"

Predicate Logic

- The statement "$x > 3$" has **two parts.**
  1. The first part, the variable *x*, is the subject of the statement.
  2. The second part, Predicate, "is greater than 3"—refers to a property that the subject of the statement can have.
- We can denote the statement "*x* is greater than 3" by $P(x)$, where *P* denotes the predicate "is greater than 3" and *x* is the variable.

Propositional Function

- *Propositional functions* contain variables and a predicate, e.g., *P*(*x*). Variables can be replaced by elements from their *domain*.
- The statement *P(x)* is also said to be the value of the **propositional function** *P* at *x*. Once a value has been assigned to the variable *x*, the statement *P(x)* becomes proposition and has a truth value.
- Let *P(x)* denote "$x > 0$" and the domain be the integers. Then:

  P(-3) is false.

  P(0) is false.

  P(3) is true.

Universes of Discourse (U.D.s)

- E.g., let $P(x)$="x+1>x". We can then say, "For *any* number x, $P(x)$ is true" instead of $(\mathbf{0}+1>\mathbf{0}) \land (\mathbf{1}+1>\mathbf{1}) \land (\mathbf{2}+1>\mathbf{2}) \land \ldots$
- The collection of values that a variable x can take is called x's ***universe of discourse***.
- We call **D** the Domain of discourse (**universe of discourse**) of P

Propositional Function

- **Example:** let P(x) denoted the statement " x is an even integer" what are the truth values of?
- **P(5), 5 is even integer**
- **P(4), 4 is even integer**
- **P(6), 6 is even integer**
- **P(7), 7 is even integer**

Propositional Function

- **Example:** let $R(x, y, z)$ denote the statement " $x + y = z$." what are the truth values of?
- ***R*(1, 2, 3) , the statement is**
- ***R*(0, 0, 1), the statement is**

Quantifier Expressions

- **Definition 1:**

  Let *P* be a propositional function with universe of discourse *D*. The *universal quantification* of *P(x)* is the proposition *"P(x) is true for all values of x in D"*

- Notation for universal quantification: $\forall x\ P(x)$

- The symbol ∀ means "for all" or "for every" or "for any".

Quantifier Expressions

- **Definition 2:**

  Let *P* be a propositional function with universe of discourse *D*. The *existential quantification* of *P(x)* is the proposition *"There exists an element x in D such that P(x) is true."*

- Notation for universal quantification: $\exists x \, P(x)$

- The symbol∃means "exists" or "fat least one" or "for some".

Quantifier Expressions

TABLE 1 Quantifiers.

Statement	*When True?*	*When False?*
$\forall x P(x)$	$P(x)$ is true for every x.	There is an x for which $P(x)$ is false.
$\exists x P(x)$	There is an x for which $P(x)$ is true.	$P(x)$ is false for every x.

Quantifiers

- **Example:** Let P(x) denote the statement "x > 3", and let the universe of discourse be *the set of real numbers*. What is the truth value of the quantification ∃xP(x) and ∀xP(x)?

- **Solution**
  - ∃xP(x) is true since there exists at least one value in the real numbers set.
    - E.g.: x = 4 which can fulfill the statement "x > 3".
  - ∀xP(x) is false since not all values in the real numbers set can fulfill the statement "x > 3".
    - E.g.: x = 1 which can not fulfill the statement "x > 3".

Quantifiers with Restricted Domains

What do the statements $\forall x < 0\,(x^2 > 0)$, $\forall y \neq 0\,(y^3 \neq 0)$, and $\exists z > 0\,(z^2 = 2)$ mean, where the domain in each case consists of the real numbers?

Solution: The statement $\forall x < 0\ (x^2 > 0)$ states that for every real number x with $x < 0$, $x^2 > 0$. That is, it states "The square of a negative real number is positive." This statement is the same as $\forall x(x < 0 \rightarrow x^2 > 0)$.

The statement $\forall y \neq 0\,(y^3 \neq 0)$ states that for every real number y with $y \neq 0$, we have $y^3 \neq 0$. That is, it states "The cube of every nonzero real number is nonzero." Note that this statement is equivalent to $\forall y(y \neq 0 \rightarrow y^3 \neq 0)$.

Finally, the statement $\exists z > 0\,(z^2 = 2)$ states that there exists a real number z with $z > 0$ such that $z^2 = 2$. That is, it states "There is a positive square root of 2." This statement is equivalent to $\exists z(z > 0 \wedge z^2 = 2)$. ◀

Precedence of Quantifiers

- The quantifiers ∀ and ∃ have higher precedence than all logical operators from propositional calculus.
- Example: $\forall x P(x) \vee Q(x)$ is the disjunction of $\forall x P(x)$ and $Q(x)$. In other words, it means $(\forall x P(x)) \vee Q(x)$.

- $\forall x(P(x) \wedge Q(x)) \equiv \forall x P(x) \wedge \forall x Q(x)$

Free and Bound Variables

- An expression like $P(x)$ is said to have a *free variable* x (meaning, x is undefined).
- A quantifier (either $\forall$ or $\exists$) *operates* on an expression having one or more free variables, and *binds* one or more of those variables, to produce an expression having one or more *bound variables*.

Example of Binding

- $P(x,y)$ has 2 free variables, x and y.
- $\forall x\ P(x,y)$ has 1 free variable, and one bound variable. [Which is which?]
- "$P(x)$, where $x=3$" is another way to bind x.
- An expression with <u>zero</u> free variables is a bona-fide (actual) proposition.
- An expression with <u>one or more</u> free variables is still only a predicate: $\forall x\ P(x,y)$

Negating Quantified Expressions

TABLE 2 **De Morgan's Laws for Quantifiers.**

Negation	*Equivalent Statement*	*When Is Negation True?*	*When False?*
$\neg\exists x P(x)$	$\forall x \neg P(x)$	For every x, $P(x)$ is false.	There is an x for which $P(x)$ is true.
$\neg\forall x P(x)$	$\exists x \neg P(x)$	There is an x for which $P(x)$ is false.	$P(x)$ is true for every x.

Negating Quantified Expressions

What are the negations of the statements $\forall x (x^2 > x)$ and $\exists x (x^2 = 2)$?

Solution: The negation of $\forall x (x^2 > x)$ is the statement $\neg \forall x (x^2 > x)$, which is equivalent to $\exists x \neg (x^2 > x)$. This can be rewritten as $\exists x (x^2 \leq x)$. The negation of $\exists x (x^2 = 2)$ is the statement $\neg \exists x (x^2 = 2)$, which is equivalent to $\forall x \neg (x^2 = 2)$. This can be rewritten as $\forall x (x^2 \neq 2)$. The truth values of these statements depend on the domain. ◀

Translating from English into Logical Expressions

- **Example 23** : Express the statement "**Every student in this class has studied calculus**" using predicates and quantifiers.
- "For every student in this class, that student has studied calculus."
- Next, we introduce a **variable *x*** so that our statement becomes
- "**For every student *x* in this class, *x* has studied calculus.**"
- we introduce $C(x)$, which is the statement "***x* has studied calculus**." Consequently, if the domain for x consists of the students in the class, we can translate our statement as $\forall x C(x)$.
- **See Example 24, Page 49**

Using Quantifiers in System Specifications

- Use predicates and quantifiers to express the system specifications "**Every mail message larger than one megabyte will be compressed**".
- ***Solution***: Let $S(m, y)$ be "**Mail message *m* is larger than *y* megabytes**," where the **variable *m*** has the domain of all **mail messages** and the **variable *y*** is a **positive real number**, and let **$C(m)$ denote** "**Mail message *m* will be compressed**." Then the specification "Every mail message larger than one megabyte will be compressed" can be represented as $\forall m(S(m, 1) \rightarrow C(m))$.

Using Quantifiers in System Specifications

- Use predicates and quantifiers to express the system specifications "**If a user is active, at least one network link will be available**."
- ***Solution***: Let *A(u)* represent "**User *u* is active**," where the **variable *u*** has the domain of **all users**, let ***S(n, x)*** denote "**Network link *n* is in state *x***," where ***n*** has the domain of **all network links** and ***x*** has the domain of **all possible states** for a network link. Then the specification "If a user is active, at least one network link will be available" can be represented by $\exists u\, A(u) \rightarrow \exists n\, S(n, \text{available})$.

Logic Programming

- An important type of programming language is designed to reason using the rules of predicate logic.
- Prolog (from *Pro*gramming in *Log*ic), developed in the 1970s by computer scientists working in the area of artificial intelligence, is an example of such a language.
- Prolog programs include a set of declarations consisting of two types of statements, **Prolog facts** and **Prolog rules**.
- Prolog facts define predicates by specifying the elements that satisfy these predicates.
- Prolog rules are used to define new predicates using those already defined by Prolog facts.

Logic Programming

- **EXAMPLE :** Consider a **Prolog** program given facts telling it the **instructor** of each class and in which classes **students** are enrolled.
- Such a program could use the **predicates** ***instructor(p, c)*** and ***enrolled(s, c)*** to represent that professor *p* is the instructor of course *c* and that student *s* is enrolled in course *c*, respectively.
- A new **predicate *teaches(p, s)***, representing that professor *p* teaches student *s*, can be defined using the Prolog rule

```
teaches(P,S) :- instructor(P,C), enrolled(S,C)
```

Logic Programming

- For example, the Prolog facts in such a program might include:
    1. **instructor(chan,math273)**
    2. **instructor(patel,ee222)**
    3. **instructor(grossman,cs301)**
    4. **enrolled(kevin,math273)**
    5. **enrolled(juana,ee222)**
    6. **enrolled(juana,cs301)**
    7. **enrolled(kiko,math273)**
    8. **enrolled(kiko,cs301)**

Logic Programming

- Prolog answers queries using the facts and rules it is given. For example, using the facts and rules listed, the query ?
- **?enrolled(kevin,math273)** produces the response
  - **yes**
- because the fact *enrolled*(kevin, math273) was provided as input. The query
- **?enrolled(X,math273)** produces the response
  - **kevin**
  - **Kiko**
- to find all the professors who are instructors in classes being taken by Juana, we use the query **?teaches(X,juana)** This query returns
  - **patel**
  - **grossman**

Nested Quantifiers

Section 1.5

Nesting of Quantifiers

- We can have nested quantifiers on a statements such as
  - $\forall x \exists y P(x, y)$
    - "for all x, there exists a y such that $\mathrm{P}(x, y)$"
    - Example: $\forall x \exists y(x + y == 0)$
  - $\exists x \forall y P(x, y)$
    - "There exists an x such that for all y $\mathrm{P}(x, y)$ is true"
    - Example: $\exists x \forall y(x \cdot y == 0)$

1.5 Nesting of Quantifiers

TABLE 1 Quantifications of Two Variables.

Statement	*When True?*	*When False?*
$\forall x \forall y P(x, y)$ $\forall y \forall x P(x, y)$	$P(x, y)$ is true for every pair x, y.	There is a pair x, y for which $P(x, y)$ is false.
$\forall x \exists y P(x, y)$	For every x there is a y for which $P(x, y)$ is true.	There is an x such that $P(x, y)$ is false for every y.
$\exists x \forall y P(x, y)$	There is an x for which $P(x, y)$ is true for every y.	For every x there is a y for which $P(x, y)$ is false.
$\exists x \exists y P(x, y)$ $\exists y \exists x P(x, y)$	There is a pair x, y for which $P(x, y)$ is true.	$P(x, y)$ is false for every pair x, y.

Translating English Sentences into Logical Expressions

- Ex: Any student either has a computer or has a friend that has a computer.
- "For any student x, x has a computer or there is some student y where y has a computer and x and y are friends."
- $\forall x[C(x) \lor \exists y[C(y) \land F(x,y)]]$, where
  - $C(x)$ is "x has a computer" and
  - $F(x,y)$ is "x and y are friends".
  - Universe discourse for x and y is all students.

The Order of Quantifiers

- The order of the quantifiers in the nested quantifier is important specially in cases like:
  - Combination between Universal and Existential quantifiers

  - $\exists x \forall y$ and $\forall x \exists y$ are **not equivalent**!
    - $\exists x \forall y P(x, y)$
      - $P(x, y) = (x + y == 0)$ is false

    - $\forall x \exists y P(x, y)$
      - $P(x, y) = (x + y == 0)$ is true

Negating Nested Quantifiers

- Recall negation rules for single quantifiers:
  - $\neg\forall x P(x) = \exists x \neg P(x)$
  - $\neg\exists x P(x) = \forall x \neg P(x)$
  - Essentially, you change the quantifier(s), and negate what it's quantifying
- Examples:
  - $\neg(\forall x \exists y P(x, y))$
  - $\equiv \exists x \neg\exists y P(x, y)$
  - $\equiv \exists x \forall y \neg P(x. y)$

  - $\neg(\forall x \exists y \forall z\, P(x, y, z))$
  - $\equiv \exists x \neg\exists y \forall z P(x, y, z)$
  - $\equiv \exists x \forall y \neg\forall z P(x, y, z)$
  - $\equiv \exists x \forall y \exists z \neg P(x, y, z)$

Translation

- "The sum of two positive integers is always positive."
  - $\forall x \forall y (x > 0 \land y > 0 \rightarrow x + y > 0)$
  - $\forall x \forall y (O(x) \land O(y) \rightarrow P(x, y))$, where
    - $O(x) = x > 0$
    - $P(x, y) = x + y > 0$
- "There exists an additive identity for any real number."
  - $\exists x \forall y (x + y = y)$

Translation

- $\forall x \forall y \left(\left((x \geq 0) \land (y < 0) \right) \rightarrow (x - y > 0) \right)$
  - A non-negative number minus a negative number is greater than zero
- $\exists x \exists y \left(\left((x \leq 0) \land (y \leq 0) \right) \rightarrow (x - y > 0) \right)$
  - The difference between two non-positive numbers is not necessarily non-positive (i.e. can be positive)
- $\forall x \forall y \left(\left((x \neq 0) \land (y \neq 0) \right) \leftrightarrow (xy \neq 0) \right)$
  - The product of two non-zero numbers is non-zero iff both factors are non-zero

Defining New Quantifiers

As per their name, quantifiers can be used to express that a predicate is true of any given *quantity* (number) of objects.

Define $\exists!x\, P(x)$ to mean "$P(x)$ is true of *exactly one* x in the universe of discourse."

$$\exists!x\, P(x) \Leftrightarrow \exists x\, \big(P(x) \wedge \neg\exists y\, (P(y) \wedge y \neq x)\big)$$

"There is an x such that $P(x)$, where there is no y such that P(y) and y is other than x."

Review: More Equivalence Laws

- $\forall x\ \forall y\ P(x,y) \Leftrightarrow \forall y\ \forall x\ P(x,y)$
  $\exists x\ \exists y\ P(x,y) \Leftrightarrow \exists y\ \exists x\ P(x,y)$
- $\forall x\ (P(x) \wedge Q(x)) \Leftrightarrow (\forall x\ P(x)) \wedge (\forall x\ Q(x))$
  $\exists x\ (P(x) \vee Q(x)) \Leftrightarrow (\exists x\ P(x)) \vee (\exists x\ Q(x))$
- Exercise:
  See if you can prove these yourself.
  - What propositional equivalences did you use?

More to Know About Binding

- $\forall x\, \exists x\, P(x)$ - x is not a free variable in $\exists x\, P(x)$, therefore the $\forall x$ binding isn't used.
- $(\forall x\, P(x)) \wedge Q(x)$ - The variable x is outside of the *scope* of the $\forall x$ quantifier, and is therefore free. Not a proposition!
- $(\forall x\, P(x)) \wedge (\exists x\, Q(x))$ – This is legal, because there are 2 different x's!

Some Number Theory Examples

- Let u.d. = the *natural numbers* 0, 1, 2, …
- "A number x is *even*, $E(x)$, if and only if it is equal to 2 times some other number."
  $\forall x\ (E(x) \leftrightarrow (\exists y\ \ x{=}2y))$
- "A number is *prime*, $P(x)$, iff it's greater than 1 and it isn't the product of two non-unity numbers."
  $\forall x\ \big(P(x) \leftrightarrow (x{>}1 \land \neg\exists yz\ \ x{=}yz \land y{\neq}1 \land z{\neq}1)\big)$

Review: Predicate Logic

- Objects *x*, *y*, *z*, …
- Predicates *P*, *Q*, *R*, … are functions mapping objects *x* to propositions *P*(*x*).
- Multi-argument predicates *P*(*x*, *y*).
- Quantifiers: [$\forall x\, P(x)$] :≡ "For all *x*'s, *P*(*x*)."
  [$\exists x\, P(x)$] :≡ "There is an *x* such that *P*(*x*)."
- Universes of discourse, bound & free vars.

Review: Natural language is ambiguous!

- "Everybody likes somebody."
  - For everybody, there is somebody they like,
    - $\forall x\ \exists y\ Likes(x,y)$ [Probably more likely.]
  - or, there is somebody (a popular person) whom everyone likes?
    - $\exists y\ \forall x\ Likes(x,y)$
- "Somebody likes everybody."
  - Same problem: Depends on context, emphasis.

Review: Quantifier Exercise

If $R(x,y)$=“x relies upon y,” express the following in unambiguous English:

$\forall x(\exists y\, R(x,y))$=

$\exists y(\forall x\, R(x,y))$=

$\exists x(\forall y\, R(x,y))$=

$\forall y(\exists x\, R(x,y))$=

$\forall x(\forall y\, R(x,y))$=

Everyone has *someone* to rely on.

There’s a poor overburdened soul whom *everyone* relies upon (including himself)!

There’s some needy person who relies upon *everybody* (including himself).

Everyone has *someone* who relies upon them.

Everyone relies upon *everybody*, (including themselves)!

Review: Quantifier Equivalence Laws

- Definitions of quantifiers: If u.d.=a,b,c,…
  $\forall x\, P(x) \Leftrightarrow P(\mathrm{a}) \wedge P(\mathrm{b}) \wedge P(\mathrm{c}) \wedge \ldots$
  $\exists x\, P(x) \Leftrightarrow P(\mathrm{a}) \vee P(\mathrm{b}) \vee P(\mathrm{c}) \vee \ldots$
- From those, we can prove the laws:
  $\forall x\, P(x) \Leftrightarrow \neg\exists x\, \neg P(x)$
  $\exists x\, P(x) \Leftrightarrow \neg\forall x\, \neg P(x)$
- Which *propositional* equivalence laws can be used to prove this? **DeMorgan's**

More Notational Conventions

- Quantifiers bind as loosely as needed: parenthesize $\forall x \big(P(x) \wedge \mathrm{Q}(x) \big)$
- Consecutive quantifiers of the same type can be combined: $\forall x\ \forall y\ \forall z\ P(x,y,z) \Leftrightarrow$ $\forall x,y,z\ P(x,y,z)$ or even $\forall xyz\ P(x,y,z)$
- All quantified expressions can be reduced to the canonical *alternating* form $\forall x_1 \exists x_2 \forall x_3 \exists x_4 \ldots P(x_1, x_2, x_3, x_{4,} \ldots)$

Rules of Inference

Section 1.6

Proof Terms

- Theorem
  - A statement that has been proven to be true.
- Axioms, postulates, hypotheses, premises
  - Assumptions (often unproven) defining the structures about which we are reasoning.
- Rules of inference
  - Patterns of logically valid deductions from hypotheses to conclusions.

Inference Rules and Implications

- Each logical inference rule corresponds to an implication that is a tautology.
- antecedent 1
  antecedent 2 …
  ∴ consequent

  Inference rule

$$\frac{\text{antecedent 1} \quad \text{antecedent 2} \ \ldots}{\therefore \text{consequent}}$$

- Corresponding tautology:
  $((\text{ante. } 1) \wedge (\text{ante. } 2) \wedge \ldots) \rightarrow \text{consequent}$

Inference Rules - General Form

- Inference Rule-
  - Pattern establishing that if we know that a set of antecedent statements of certain forms are all true, then a certain related consequent statement is true.
- antecedent 1
  antecedent 2 …
  ―――――――――
  ∴ consequent

  "∴" means "therefore"

Rules of Inference for Propositional Logic

- A theorem often has two parts
  - Conditions (premises, hypotheses)
  - Conclusion
- A correct (deductive) proof is to establish that
  - If the conditions are true then the conclusion is true
  - i.e., conditions $\rightarrow$ conclusion is a tautology
- Often there are missing pieces between conditions and conclusion. Fill it by an argument
  - Using conditions and axioms
  - Statements in the argument connected by proper rules of inference

Rules of Inference for Propositional Logic

- Rules of inference provide the justification of the steps used in a proof.
- One important rule is called **Modus Ponens** or the law of detachment. It is based on the tautology $(p \wedge (p \to q)) \to q$ We write it in the following way:

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

The two hypotheses p and $q \to q$ are written in a column, and the conclusion q below a bar, where $\therefore$ means "therefore".

Rules of Inference

- The rule states that if p_1 and p_2 and … and p_n are all true, then q is true as well.
- Each rule is an established tautology of
- $p_1 \land p_2 \land \cdots \land p_n \rightarrow q$
- These rules of inference can be used in any mathematical argument and do not require any proof.

$$\begin{array}{l} p_1 \\ p_2 \\ . \\ . \\ . \\ p_n \\ \hline \therefore q \end{array}$$

Valid Arguments in Propositional Logic

- An argument Just like a rule of inference, it consists of one or more hypotheses (or premises) and a conclusion.
  - We say that an argument is valid, if whenever all its hypotheses are true, its conclusion is also true.
  - However, if any hypotheses is false, even a valid argument can lead to an incorrect conclusion.
  - Proof: show that hypothesis → conclusion is true using rules of inference

Example on Argument Validity

- Check the validity of the following arguments

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

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

Argument Validity

- Check the validity of the following arguments
  - "If I like DM, I will study it. Either I study DM or I fail the course. Therefore, If I fail the course, then I do not like DM".

p = I like DM
q = I study it
r = I fail the course

$$\begin{array}{l} p \rightarrow q \\ q \vee r \\ \hline r \rightarrow (\neg p) \end{array}$$

It is an **invalid** argument

p	q	r	(¬p)	p → q	q ∨ r	r → (¬p)
T	T	T	F	T	T	F
T	T	F	F	T	T	T
T	F	T	F	F	T	F
T	F	F	F	F	F	T
F	T	T	T	T	T	T
F	T	F	T	T	T	T
F	F	T	T	T	T	T
F	F	F	T	T	F	T

TABLE 1 Rules of Inference.

Rule of Inference	*Tautology*	*Name*
p $p \to q$ $\therefore q$	$(p \land (p \to q)) \to q$	Modus ponens
$\neg q$ $p \to q$ $\therefore \neg p$	$(\neg q \land (p \to q)) \to \neg p$	Modus tollens
$p \to q$ $q \to r$ $\therefore p \to r$	$((p \to q) \land (q \to r)) \to (p \to r)$	Hypothetical syllogism
$p \lor q$ $\neg p$ $\therefore q$	$((p \lor q) \land \neg p) \to q$	Disjunctive syllogism
p $\therefore p \lor q$	$p \to (p \lor q)$	Addition
$p \land q$ $\therefore p$	$(p \land q) \to p$	Simplification
p q $\therefore p \land q$	$((p) \land (q)) \to (p \land q)$	Conjunction
$p \lor q$ $\neg p \lor r$ $\therefore q \lor r$	$((p \lor q) \land (\neg p \lor r)) \to (q \lor r)$	Resolution

p $\therefore \overline{p \vee q}$	$p \rightarrow (p \vee q)$	Addition
$p \wedge q$ $\therefore \overline{p}$	$(p \wedge q) \rightarrow p$	Simplification
p q $\therefore \overline{p \wedge q}$	$((p) \wedge (q)) \rightarrow (p \wedge q)$	Conjunction

Rule of Inference	*Tautology*	*Name*
p $p \rightarrow q$ $\therefore q$	$(p \land (p \rightarrow q)) \rightarrow q$	Modus ponens
$\neg q$ $p \rightarrow q$ $\therefore \neg p$	$(\neg q \land (p \rightarrow q)) \rightarrow \neg p$	Modus tollens

$p \to q$ $q \to r$ $\therefore p \to r$	$((p \to q) \wedge (q \to r)) \to (p \to r)$	Hypothetical syllogism
$p \vee q$ $\neg p$ $\therefore q$	$((p \vee q) \wedge \neg p) \to q$	Disjunctive syllogism

Argument Validity

- "If it rains today, then we will not have a barbeque today. If we do not have a barbeque today, then we will have a barbeque tomorrow. Therefore, if it rains today, then we will have a barbeque tomorrow."
  - p: "It is raining today."
  - q: "We will not have a barbecue today."
  - r: "We will have a barbecue tomorrow."

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

Hypothetical syllogism (chaining) rule of inference

  - It is a **valid** argument.

Argument Validity

- Gary is either intelligent or a good actor.
- If Gary is intelligent, then he can count from 1 to 10.
- Gary can only count from 1 to 3.
- Therefore, Gary is a good actor.
    - **p**: "Gary is intelligent."
    - **q**: "Gary is a good actor."
    - **r**: "Gary can count from 1 to 10."

Hypothesis : $p \lor q$
Hypothesis : $p \to r$
Hypothesis : $\neg r$
Conclusion : q

Step 1:	$\neg r$	Hypothesis
Step 2:	$p \to r$	Hypothesis
Step 3:	$\neg p$	*Modus tollens* 1&2
Step 4:	$q \lor p$	Hypothesis
Step 5:	q	Disjunctive Syllogism 3&4

Conclusion: q ("Gary is a good actor.")

Argument Validity

- Show that the **premises** "If you send me an e-mail message, then I will finish writing the program," "If you do not send me an e-mail message, then I will go to sleep early," and "If I go to sleep early, then I will wake up feeling refreshed" lead to the conclusion "If I do not finish writing the program, then I will wake up feeling refreshed."
- ***Solution:*** Let p be the proposition "You send me an e-mail message," q the proposition "I will finish writing the program," r the proposition "I will go to sleep early," and s the proposition "I will wake up feeling refreshed." Then the **premises are $p \rightarrow q$, $\neg p \rightarrow r$, and $r \rightarrow s$**.
- The desired conclusion is $\neg q \rightarrow s$. We need to give a valid argument with premises $p \rightarrow q$, $\neg p \rightarrow r$, and $r \rightarrow s$ and conclusion $\neg q \rightarrow s$.

Argument Validity

- ***Solution:*** This argument form shows that the premises lead to the desired conclusion.

Step	Reason
1. $p \to q$	Premise
2. $\neg q \to \neg p$	Contrapositive of (1)
3. $\neg p \to r$	Premise
4. $\neg q \to r$	Hypothetical syllogism using (2) and (3)
5. $r \to s$	Premise
6. $\neg q \to s$	Hypothetical syllogism using (4) and (5)

Rules of Inference for Quantified Statements

TABLE 2 Rules of Inference for Quantified Statements.

Rule of Inference	*Name*
$\forall x P(x)$ $\therefore P(c)$	Universal instantiation
$P(c)$ for an arbitrary c $\therefore \forall x P(x)$	Universal generalization
$\exists x P(x)$ $\therefore P(c)$ for some element c	Existential instantiation
$P(c)$ for some element c $\therefore \exists x P(x)$	Existential generalization

Rules of Inference for Quantified Statements

- **Example:** Show that the premises "Everyone in this discrete mathematics class has taken a course in computer science" and "Marla is a student in this class" imply the conclusion "Marla has taken a course in computer science."
- ***Solution***: Let $D(x)$ denote "x is in this discrete mathematics class," and let $C(x)$ denote "x has taken a course in computer science." Then the premises are $\forall x(D(x) \rightarrow C(x))$ and $D(\text{Marla})$. The conclusion is $C(\text{Marla})$. The following steps can be used to establish the conclusion from the premises.

Step	Reason
1. $\forall x(D(x) \rightarrow C(x))$	Premise
2. $D(\text{Marla}) \rightarrow C(\text{Marla})$	Universal instantiation from (1)
3. $D(\text{Marla})$	Premise
4. $C(\text{Marla})$	Modus ponens from (2) and (3)

Rules of Inference for Quantified Statements

- **Example:** Show that the premises "A student in this class has not read the book," and "Everyone in this class passed the first exam" imply the conclusion "Someone who passed the first exam has not read the book."
- ***Solution***: Let $C(x)$ be "x is in this class," $B(x)$ be "x has read the book," and $P(x)$ be "x passed the first exam." The premises are $\exists x(C(x) \wedge \neg B(x))$ and $\forall x(C(x) \rightarrow P(x))$. The conclusion is $\exists x(P(x) \wedge \neg B(x))$. These steps can be used to establish the conclusion from the premises.

Step	Reason
1. $\exists x(C(x) \wedge \neg B(x))$	Premise
2. $C(a) \wedge \neg B(a)$	Existential instantiation from (1)
3. $C(a)$	Simplification from (2)
4. $\forall x(C(x) \rightarrow P(x))$	Premise
5. $C(a) \rightarrow P(a)$	Universal instantiation from (4)
6. $P(a)$	Modus ponens from (3) and (5)
7. $\neg B(a)$	Simplification from (2)
8. $P(a) \wedge \neg B(a)$	Conjunction from (6) and (7)
9. $\exists x(P(x) \wedge \neg B(x))$	Existential generalization from (8)

Rules of Inference for Quantified Statements

- Suppose we have the following premises:
  "It is not sunny and it is cold."
  "We will swim only if it is sunny."
  "If we do not swim, then we will play volley."
  "If we play volley, then we will be home early."
- prove the theorem
  "We will be home early" using inference rules.

Rules of Inference for Quantified Statements

- *Solution :*
  - ❖ *Sunny = it is sunny*
  - ❖ *Cold = it is cold*
  - ❖ *Swim = we will swim*
  - ❖ *Volley = we will play volley*
  - ❖ *Early = we will be home early*
- Then the premises can be written as:

1. **~Sunny ∧ *Cold***
2. ***Swim → Sunny***
3. ***~Swim → Volley***
4. ***Volley → Early***

Rules of Inference for Quantified Statements

Step	Proved by
1. $\neg sunny \wedge cold$	Premise #1.
2. $\neg sunny$	Simplification of 1.
3. $swim \rightarrow sunny$	Premise #2.
4. $\neg swim$	Modus tollens on 2,3.
5. $\neg swim \rightarrow volley$	Premise #3.
6. $volley$	Modus ponens on 4,5.
7. $volley \rightarrow early$	Premise #4.
8. $early$	Modus ponens on 6,7.