Predicate Logic

- Discrete Mathematics (Kenneth Rosen)
  - 8th edition – 1.4-1.8

Propositional Logic is not enough.

- We have no way to argue about class of entities. For example,
  - Given that all the students are below 25 years of age, and Jill is a student.
  - We do not have any way to deduce that Jill is below 25 years of age.
  - Similarly, if x>2, and 2>1, we have no rules to deduce that x>1.
  - In fact we do not have any way to encode the information, x > 2, in propositional logic, why?

Predicate Logic

- *Predicate logic* is an extension of propositional logic that permits concisely reasoning about whole *classes* of entities.

  E.g., “*x*>y”, “*x*=5”.

- Such statements are neither true or false unless the values of the variables are not specified. Hence, these aren’t propositions.

Proposition vs. Predicates

- More Examples.
  - "Khushraj is Teaching", 2>1
- Both of the above are propositions.
- These propositions have two parts: Subjects, and a "relation/property" about the subjects these subjects.
- For example, Khushraj is a subject, and is teaching is a property or a verb that gives you more information about the subject.
- 2,1 are subjects and > is a relation/property that gives you more info about 2 and 1.

What is a Predicate?

- A **predicate** is a property or relation:
  - Example: *P(x): x is prime*
  - *L(x, y): x loves y*
  - *Friends(x,y,z): x,y,z are friends.*
- Predicates become **propositions** when **variables are instantiated.**
  - *P(5)*
  - *L(Alice, Bob)*
  - *Hence Predicates can be seen as propositional functions. That is, they are a function from the value of the variables, to a proposition.*

What is a Predicate?

- Hence, a *predicate* is modeled as a *function* $P(\cdot)$ from objects to propositions.
  - $P(x)$ = "x is prime" (where x is any object).
- The *result of applying* a predicate P to an object $x=a$ is the *proposition* $P(a)$.
  - *Hence,* $P(3)$ is the *proposition* "3 is a prime."
  - *Similarly, P(4) is the proposition "4 is a prime."*
  - *The truth of these propositions depend on what the meaning of "prime" is for us. If we interpret prime with its usual meaning, theh P(3) is true while P(4) is false.*
- Note: The predicate *P* **itself** (*e.g. P*="is prime") is **not** a proposition (not a complete sentence). Number of arguments that a predicate P takes is its arity.

Proposition vs. Predicates

- Propositions treat statements as whole units, with no insight into internal structure.
- Predicates let you break down propositions into components — e.g., objects and relationships — enabling you to quantify, generalize, and reason about classes of statements.
- This additional structure gives you finer control and expressiveness — hence more fine-grained.
- Hence, predicate logic is more fine-grained than propositional logic.

Applications of Predicate Logic

- Same as propositional logics: automated proofs, solving puzzles, checking correctness of programs, solving complex circuits, querying databases. But are more powerful and can express more type of information.

- In fact, It is *the* formal notation for writing perfectly clear, concise, and unambiguous mathematical *definitions*, *axioms*, and *theorems* for "almost any" branch of mathematics.

Universe/Domain of Discourse

- Notice that predicates can be seen as functions that outputs “true/false”.
- Hence, we need to specify what are the values that the variables can take.
- The collection of values that a variable x can take is called x’s *universe of discourse*.

e.g., let $P(x)$=“$x+1>x$”.

we could define the course of universe as the set of integers.

Quantifiers.

- **Quantifiers** help us express **how many elements** in the universe of discourse satisfy a given condition or predicate.
- **"∀" (the universal quantifier)** means:
  → For **every** element x in the universe, the statement P(x) is true.
  (Symbolically: $\forall x\, P(x)$)
- **"∃" (the existential quantifier)** means:
  → There is **at least one** element x in the universe for which P(x)P(x)P(x) is true.
  (Symbolically: $\exists x\, P(x)$)

Quantifiers Example.

English Statement	Predicate Logic
All humans are mortal	$\forall x\ (Human(x) \rightarrow Mortal(x))$
Some birds can't fly	$\exists x\ (Bird(x) \land \neg CanFly(x))$
Every prime > 2 is odd	$\forall x\ (Prime(x) \land x > 2 \rightarrow Odd(x))$
There is a number divisible by 3	$\exists x\ (Divisible\_By\_3(x))$

Universal Quantifier, $\forall$

- To prove that a statement of the form $\forall x\, P(x)$ is true, we need to check that value of all possible values of x in domain of discourse such that P(x) is true.

- To prove that a statement of the form $\forall x\, P(x)$ is false, it suffices to find a **counterexample** (i.e., one value of x in the universe of discourse such that $P(x)$ is false)
  - e.g., $P(x)$ is the predicate "$x>0$"

Existential Quantifier, ∃

- To prove that a statement of the form $\exists x\, P(x)$ is true, we just need to find one example a in the domain of discourse such that P(a) is true.

- To prove that a statement of the form $\exists x\, P(x)$ is false, we need to check that for every possible value a of x, P(x) is false.

Quantifiers as ∧,∨

- Definitions of quantifiers: If domain ={a,b,c,…}
  $\forall x\ P(x) \Leftrightarrow P(\mathrm{a}) \wedge P(\mathrm{b}) \wedge P(\mathrm{c}) \wedge \ldots$
  $\exists x\ P(x) \Leftrightarrow P(\mathrm{a}) \vee P(\mathrm{b}) \vee P(\mathrm{c}) \vee \ldots$
- We can prove the following 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?

Equivalence Laws

- $\neg\exists x{:}P(x) \Leftrightarrow \forall x{:}\neg P(x)$
  $\neg\forall x{:}P(x) \Leftrightarrow \exists x{:}\neg P(x)$

- $\forall x :(P(x) \wedge Q(x)) \Leftrightarrow (\forall x : P(x)) \wedge (\forall x : Q(x))$
  $\exists x :(P(x) \vee Q(x)) \Leftrightarrow (\exists x : P(x)) \vee (\exists x : Q(x))$

Scope of Quantifier

- The part of a logical formula to which a quantifier is applied is called its scope.

e.g., $(\forall x{:}P(x)) \wedge (\exists y{:}Q(y))$

e.g., $(\forall x{:}P(x)) \wedge (\exists x{:}Q(x))$

Free and Bound Variable

- An expression like $P(x)$ is said to have a *free variable x* (i.e. x is undefined).
- A quantifier (such as ∀ or ∃) applies to an expression containing free variables, and transforms those variables into bound variables, resulting in a statement where the variables are no longer free.
- Notice that formulae containing no free variables can be seen as “propositions”. These formulae are called **closed formula**.
- Any formula containing atleast one free variable is called an **open formula**.

Nested Quantifiers

Exist within the scope of other quantifiers

- Let the domain of *x* & *y* be people.
- Let $P(x,y)$="*x* likes *y*" (a predicate with 2 f.v.'s)
- Then $\exists y{:}P(x,y)$ = "There is someone whom *x* likes." (a predicate with 1 free variable, *x*)
- Then $\forall x{:}(\exists y{:}P(x,y))$ = "Everyone has someone whom they like."
  (A ____________ with ___ free variables.)

Examples

- $P(x,y)$ 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 an actual proposition.
- An expression with one or more free variables is still only a predicate: $\forall x{:}P(x,y)$

Reusing variable names

- $\forall x{:}\exists x{:}P(x)$ - *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".
- $(\forall x{:}P(x)) \wedge (\exists x{:}Q(x))$ - Legal because there are 2 <u>different</u> *x*'s!
- Quantifiers bind as loosely as needed: parenthesize $\forall x \quad P(x) \wedge Q(x)$

Order of Quantifiers

If $P(x,y)$="x likes y," express the following in unambiguous English:

$\forall x:(\exists y:P(x,y))$=

$\exists y:(\forall x:P(x,y))$=

$\exists x:(\forall y:P(x,y))$=

$\forall y:(\exists x:P(x,y))$=

$\forall x:(\forall y:P(x,y))$=

Order of Quantifiers

- A teacher supervises every student.
- What does this mean?
- $\exists x\ \forall y\ [teacher(x) \land student(y)] \rightarrow supervises(x, y)$

OR

- $\forall x\ \exists y[teacher(x) \land student(y)] \rightarrow supervises(x, y)$
- One of the reasons why Natural Language is ambiguous. It sometimes forgets to mention the order of the quantifiers.
- Only possible to switch quantifier without affecting the meaning when they are identical and adjacent.
- $\forall x\ \forall y : P(x,y) \Leftrightarrow \forall y\ \forall x : P(x,y)$
  $\exists x\ \exists y : P(x,y) \Leftrightarrow \exists y\ \exists x : P(x,y)$

Some math examples

- Let domain = 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."
  $\forall x\,\big(E(x) \leftrightarrow (\exists y\ x=2y)\big)$
- "A number is *prime*, $P(x)$, iff it isn't the product of two non-unity numbers."
  $\forall x\,\big(P(x) \leftrightarrow (\neg\exists y,z\ x=y\times z \wedge y\neq 1 \wedge z\neq 1)\big)$

Finite Domain what happens?

Definition of Limit

$$\left(\lim_{x\to a} f(x) = L\right) \Leftrightarrow$$

$$\left(\begin{array}{l}\forall \varepsilon > 0 : \exists \delta > 0 : \forall x : \\ \left(|x-a| < \delta\right) \rightarrow \left(|f(x)-L| < \varepsilon\right)\end{array}\right)$$

Rules of Inference

- Kenneth Rosen, 8 edition, section 1.6

What is an argument?

- An **argument** is a sequence of propositions intended to establish a conclusion.
- It consists of:
  - **Premises**: Propositions assumed to be true.
  - **Conclusion**: The proposition inferred from the premises.
- Written as:
  - Premise_1
  - Premise_2
  - …
  - $\therefore$ Conclusion

Key Definitions

- Premise: A statement assumed to be true for the purpose of the argument.
- Conclusion: A statement that follows logically from the premises.
- Valid Argument: An argument where the conclusion logically follows from the premises. That is if the premises are true then the conclusion is true. In other words, $p_1 \land p_2 \ldots \rightarrow c$ is a tautology.
- Fallacy: An error in reasoning that makes an argument invalid.

Example of a Valid Argument

- Premise 1: If it rains, the ground gets wet. ($p \rightarrow q$)
- Premise 2: It rains. (p)
- ∴ Conclusion: The ground gets wet. (q)
- Valid by Modus Ponens

Common Rules of Inference

- Modus Ponens: $p \rightarrow q, p \vdash q$
- Modus Tollens: $p \rightarrow q, \neg q \vdash \neg p$
- Hypothetical Syllogism: $p \rightarrow q, q \rightarrow r \vdash p \rightarrow r$
- Disjunctive Syllogism: $p \vee q, \neg p \vdash q$
- Addition: $p \vdash p \vee q$
- Simplification: $p \wedge q \vdash p$
- Conjunction: $p, q \vdash p \wedge q$
- Resolution: $p \vee q, \neg p \vee r \vdash q \vee r$

Modus Ponens (Law of Detachment)

- If $p \rightarrow q$ and p are both true, then q must be true.
- Example:
- Premise 1: If I study, I pass.
- Premise 2: I study.
- ∴ I pass.

Modus Tollens

- If $p \rightarrow q$ and $\neg q$, then $\neg p$.
- Example:
- Premise 1: If it is a dog, it has four legs.
- Premise 2: It does not have four legs.
- ∴ It is not a dog.

Hypothetical Syllogism

- $p \rightarrow q, q \rightarrow r \vdash p \rightarrow r$
- Example:
- If I win, I'll be happy.
- If I'm happy, I'll celebrate.
- ∴ If I win, I'll celebrate.

Disjunctive Syllogism

- $p \vee q, \neg p \vdash q$
- Example:
- I will eat pizza or pasta.
- I won’t eat pizza.
- ∴ I will eat pasta.

Fallacies

- Invalid Argument.
- Fallacies to Avoid:
- Affirming the Consequent: $p \rightarrow q, q \vdash p$ (invalid)
- Denying the Antecedent: $p \rightarrow q, \neg p \vdash \neg q$ (invalid)

Example Argument Evaluation

- Premises:
- If I go to the party, I will be tired.
- I am tired.
- ∴ I went to the party.
- Fallacy: Affirming the consequent

Summary (Propositional Logic)

- Arguments are made of premises and conclusions.
- Validity means truth-preserving structure.
- Rules of inference help derive conclusions.
- Be careful of fallacies — they look valid but aren't!

What's New in Predicate Logic?

- Propositional logic deals with whole statements.
- Predicate logic analyzes internal structure of statements.
- Introduces:
- - Quantifiers: ∀ (for all), ∃ (there exists)
- - Predicates: Functions mapping objects to truth values
- - Variables and domains

Common Inference Rules in Predicate Logic

- Universal Instantiation (UI):
  $\forall x\, P(x) \vdash P(c)$
- Universal Generalization (UG):
  [For any arbitrary c if P(c)]$\vdash \forall x\, P(x)$
- Existential Instantiation (EI):
  $\exists x\, P(x) \vdash$ [For some element c, P(c)]
- Existential Generalization (EG):
  [For some element c, P(c)] $\vdash \exists x\, P(x)$

Universal Instantiation (UI)

- From a universally quantified statement, infer a specific instance.
- $\forall x\, P(x) \vdash P(a)$
- Example:
- $\forall x\, (Human(x) \rightarrow Mortal(x))$
- $\vdash Human(Socrates) \rightarrow Mortal(Socrates)$

Existential Instantiation (EI)

- From ∃x P(x), infer P(c) for some constant c (assumed fresh).
- ∃x P(x) ⊢ P(c)
- Example:
- ∃x Student(x) ∧ Smart(x)
- ⊢ There is some student,c, Student() ∧ Smart(Alice)

Existential Generalization (EG)

- From a statement about a specific individual, infer existence.
- $P(c) \vdash \exists x\, P(x)$
- Example:
- Smart(Alice)
- $\vdash \exists x\, \text{Smart}(x)$

Universal Generalization (UG)

- From a statement about arbitrary individual, infer universal statement.
- $P(c) \vdash \forall x\, P(x)$ (only if c was arbitrary, not dependent on assumptions)
- Use with caution!
- Example (valid): Assume c is arbitrary and prove $P(c)$, $\vdash \forall x\, P(x)$

Using Predicate Inference Rules

- Predicate logic inference is often done by:
- - Applying UI or EI to eliminate quantifiers
- - Using propositional rules (e.g., Modus Ponens)
- - Generalizing back using EG or UG when allowed

Example Argument

- Premises:
- 1. ∀x (Dog(x) → Mammal(x))
- 2. ∃x Dog(x)
- Conclusion: ∃x Mammal(x)
- Steps:
- - From (1), by UI: Dog(a) → Mammal(a)
- - From (2), by EI: Dog(a)
- - Modus Ponens: ⊢ Mammal(a)
- - EG: ⊢ ∃x Mammal(x)

Common Mistakes

- Applying UG to a constant not known to be arbitrary.
- Assuming that $\exists x\, P(x)$ means P holds for every x.
- Confusing UI and EI scope.

Summary (Predicate Logic)

- Predicate logic inference introduces quantifiers and variables.
- Four key rules: UI, UG, EI, EG.
- Combine with propositional rules for full inference power.
- Carefully manage scope and assumptions for validity.