Chapter 1.

The Foundations: Logic and Proofs

Chapter 1, Part 2: Predicate Logic

Dept. of Software
Gachon University
Spring 2022

1.4 Predicates and Quantifiers

Limitation of Propositional Logic

- Statements operate only for constant objects
- No ways to apply to many objects / groups
  - Example:
    - John is a CS graduate — John has passed cs441
    - Ann is a CS graduate — Ann has passed cs441
    - Ken is a CS graduate — Ken has passed cs441
    - …

한꺼번에 묘사 불가

- Statements do not explicitly represent objects, properties and their relations

관계를 명확하게 않을수 있음

Seoul is the capital of S. Korea.

John is a CS Student.

John → a CS Student

object → property

Limitation of Propositional Logic (cont.)

Statements need to define the property of the group of objects

Propositional logic: statements cannot say "All", "there exists", or etc

- **Example:**
  - All new cars must be registered.
  - Some of the CS graduates graduate with honor.

Solution : Using a mathematical model for reasoning

- Example – Consider the following boolean propositions:
  - A (Ava is tall)
  - B (Baily is tall)
  - C (Chris is tall)

  …

  - Z (Zack is tall)

서술부

- We can capture the same set of truth values using a single **predicate** (or boolean function), Tall(x)
  - *x* is a variable representing a person

같은 서술부를 가진 명제들이 있을 때
그 서술부를 사용해 표현 가능

- Tall(x) is true whenever person x is tall, and is false otherwise.
  - Tall(Ava) : true if proposition A above is true.
  - Tall(Baily) : true if proposition B above is true.
  - Tall(Chris) : true if proposition C above is true.

Introducing **Predicate** Logic

Explicit : 명료하게

- **Predicate logic:** represent properties or relations among objects using predicate
  - Explicitly models objects and their properties
  - Allows to make statements with variables and quantify them
- **Predicates :** *P*(*x*), Q(x), *MAX*(*x, y, z*), *Red(1)*, …
  - over one, two or many variables or constants.

  Examples: Red(car2), Student(x), Married(John,Ann)

  함수처럼 사용 가능

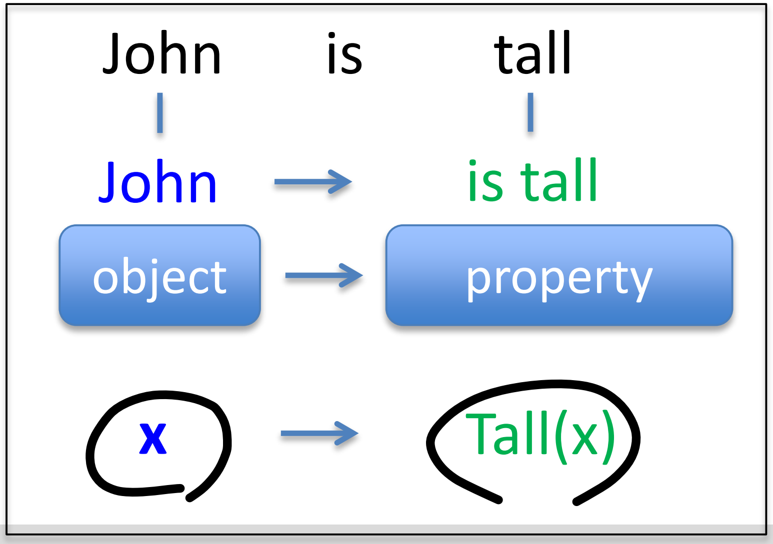
  Variable – represents object of specific type

  Examples: x, y , z

  x is a CS graduate. ***x*** has passed cs441

  Constant –models a specific object

  Examples: "John", "Ahn", "car2", "7"



Explicitly represents properties or relations among objects

Predicates

Predicate P(x) is **a proposition or not?**

P(x)는 참, 거짓이 될수 없어서
proposition이 아님

- P(x) is **not a proposition**

since there are more objects it can be applied to.

- A **predicate *P(x)*** assigns a value **true or false** to each *x* depending on whether the property holds or not for *x.*
  - The assignment is best viewed as a big table with the variable x substituted for objects from ***the domain (universe) of discourse*** *(*전체집합 U)*
- Example:
  - Assume **Student*(x)*** where the universe of discourse are people
  - **Student(John)** …. T (if John is a student)
  - **Student(Ann)** …. T (if Ann is a student)
  - **Student(Jane)** ….. F (if Jane is not a student)

Predicates

- **Example:**
  - **Assume a predicate *P(x)* that represents the statement: *'x is a prime number'***
- Is P(x) a proposition ?
  - **No. Many possible substitutions are possible**
- What are the truth values of:

전체는
명제 X

  - P(2) T → 각각은 명제이다
  - P(3) T
  - P(4) F
  - P(5) T
  - P(6) F
  - P(7) T
  - **All statements P(2), P(3), P(4), P(5), P(6), P(7) are propositions**

Predicates

- Predicates can have **two or more arguments** which represent the **relations between objects**

Example:

- **Older(x, y)** - ‘x is older than y’
  - not a proposition, but after the substitution it becomes one

- **Older(John, Peter)** denotes ‘John is older than Peter’
  - this is a proposition because it is either true or false

구체화되면 명제임

(cont.)

x, y가 정해져야만 "명제" 라고 부를 수 있음

Example:

- Let Q(x,y) denote ‘x + 5 > y’
  - Is Q(x, y) a proposition? **No!**
  - Is Q(3, 7) a proposition? **Yes.** It is true.
  - What is the truth value of:
  - Q(3, 7) T
  - Q(1, 6) F
  - Q(2, 2) T
  - Is Q(3,y) a proposition? **No!** We cannot say if it is true or false.

혼합된

Compound Expressions in predicate logic

- **Compound statements are obtained via logical connectives**

Examples:

- Student(Ann) ∧ Student(Jane)
  - **Translation:** "Both Ann and Jane are students"
  - **Proposition:** yes.
- Country(Sienna) ∨ River(Sienna)
  - **Translation:** "Sienna is a country or a river"
  - **Proposition:** yes.
- CS-major(x) → Student(x)
  - **Translation:** "if x is a CS-major then x is a student"
  - **Proposition:** no.

Quantified statements

‘ All CS/SW graduates have to take SWMath”

‘Some CS/SW students graduate with honor.’

기본 명제들은 All, Some 사용 불가

그러나 술어 논리에서는 사용 가능!

Predicate logic permits quantified sentences where variables are substituted for statements about the group of objects

Quantified statements

- Predicate logic lets us to make statements about groups ("all", "some") of objects
  - To do this we use special quantified expressions using ***quantifiers***
- **Two most important quantifiers are:**
  - *Universal Quantifier*, "For all," symbol: $\forall$
  - *Existential Quantifier*, "There exists," symbol: $\exists$
- Examples
  - *Universal:* "All SW/CS graduates have to take SW-math"
    - True for all graduates
  - *Existential:* "Some SW/CS students graduate with honor"
    - True for some students

Universal Quantifier ∀

- **Definition:** The universal quantification of P(x) is the proposition:
  - "*P(x) is true for all values of x in the domain of discourse*." The notation $\forall x\ P(x)$ denotes the universal quantification of P(x), and is expressed as **for every x P(x)** **(for all x P(x))**
  - An element for which *P(x)* is false is called a **counterexample** of $\forall x\ P(x)$.

- **Example:**

나타낸다

  - **Question:** Let P(x) denote x > x - 1.
  - What is the truth value of $\forall x\ P(x)$**?**
  - Assume the universe of discourse of x is all real numbers.

Universal Quantifier (cont.)

- **Quantification** converts a propositional function into **a proposition** by binding a variable to a set of values from the universe of discourse.

- **E.g.** Let P(x) denote $x > x - 1$.

- Is P(x) a proposition ?
  - **No. Many possible substitutions are possible**

- Is $\forall$**x P(x)** a proposition ?
  - **True if for all x from the universe of discourse P(x) is true.**

?

Universal Quantifier (cont.)

- $\forall x\, P(x)$ is read as "For all x, P(x)" or "For every x, P(x)"

  Examples:

  1) If $P(x)$ denotes "$x > 0$" and U is the integers, then $\forall x\, P(x)$ is false.
  2) If $P(x)$ denotes "$x > 0$" and U is the positive integers, then $\forall x\, P(x)$ is true.
  3) If $P(x)$ denotes "x is even" and U is the integers, then $\forall x\, P(x)$ is false.

Aha!

Existential Quantifier ∃

- **Definition:** The **existential quantification** of P(x) is the proposition
- "*There exists an element in the domain (universe) of discourse such that P(x) is true*." The notation **∃x P(x)** denotes the existential quantification of P(x), and is expressed as **there is an x such that P(x) is true**.

$\exists x P(x)$ 는, P(x)를 T로 만드는 어떤 하나의 수만 찾으면 T가 된다

- **Example 1:**
  - **Question:** Let **T(x)** denote x > 5, where x is from Real numbers.
  - What is the truth value of **∃ x P(x)?** T

전체집합 *U* :
real number (실수)

x>5 이면 T.

Existential Quantifier (cont.)

- **Example 2:**
  - **Question:** Let **Q(x)** denote x = x + 2 where x is a real number.
  - What is the truth value of **∃ x Q(x)?**

F

Summary of Quantified Statements

- $P(x)$: is not a proposition.
- $\forall x\, P(x)$ and $\exists x P(x)$: both are propositions.

  - When $\forall x\, P(x)$ and $\exists x P(x)$ are true and false?

Statement	When True?	When False?
$\forall x\, P(x)$	P(x) true for all x	There exists an x where P(x) is false
$\exists x P(x)$	There exist(s) some x for which P(x) is true	P(x) is false for all x

  - If the domain U consists of $x_1, x_2, \ldots, x_N$ then
    - $\forall x\, P(x)$ is true whenever $P(x_1) \wedge P(x_1) \wedge \ldots P(x_N)$ is true
    - $\exists x\, P(x)$ is true whenever $P(x_1) \vee P(x_1) \vee \ldots P(x_N)$ is true

Thinking about Quantifiers

- When the domain of discourse is finite, we can think of quantification as looping through the elements of the domain.

- To evaluate $\forall x\, P(x)$ loop through all x in the domain.
  - If at every step P(x) is true, then $\forall x\, P(x)$ is true.
  - If at a step P(x) is false, then $\forall x\, P(x)$ is false and the loop terminates.

- To evaluate $\exists x\, P(x)$ loop through all x in the domain.
  - If at some step, P(x) is true, then $\exists x\, P(x)$ is true and the loop terminates.
  - If the loop ends without finding an x for which P(x) is true, then $\exists x\, P(x)$ is false.

- C.f. Even if the domains are infinite, we can still think of the quantifiers this fashion, but the loops will not terminate in some cases.

Properties of Quantifiers

- The truth values of $\exists x\, P(x)$ and $\forall x\, P(x)$ depend on both the **propositional function *P(x)*** and on the **domain *U***.
- **Examples**:

전체집합 고려!

P(x)	Domain *U*	$\exists x\, P(x)$	$\forall x\, P(x)$
	$x \geq 0$ Positive integers	0, 1, 2 True	False
$x < 2$	$x < 0$ Negative integers	−1, −2, ... True	True
	3, 4, 5	False	False
$x > 2$	3, 4, 5	True	True

Precedence of Quantifiers

$\forall x\, P(x) \vee Q(x) = ?$

$(\forall x\, P(x)) \vee Q(x)$?

$\cancel{\forall x\, (P(x) \vee Q(x))}$?

- The quantifiers ∀ and ∃ have higher precedence than all the logical operators.

우선순위 높음

- For example,

  $\forall x\, P(x) \vee Q(x)$ means $(\forall x\, P(x)) \vee Q(x)$
- $\forall x\, (P(x) \vee Q(x))$ means something different
  – Unfortunately, often people write $\forall x\, P(x) \vee Q(x)$ when they mean $\forall x\, (P(x) \vee Q(x))$.

Equivalences in Predicate Logic

- Statements involving predicates and quantifiers are *logically equivalent* if and only if they have the same truth value
  - for every predicate substituted into these statements and
  - for every domain of discourse used for the variables in the expressions.

- **Example 1**: $\forall x \neg\neg S(x) \equiv \forall x\, S(x)$
- **Example 2**: If U consists of the integers 1,2, and 3:

$$\forall x P(x) \equiv P(1) \wedge P(2) \wedge P(3)$$
$$\exists x P(x) \equiv P(1) \vee P(2) \vee P(3)$$

Negation of Quantified Expressions

- Consider $\forall x\, J(x)$

  "**Every student in your class has taken a course in Java**."

  - $J(x)$: "x has taken a course in Java"

  - the domain U : students in your class.

- Negating the original statement gives :

  "**It is not the case that every student in your class has taken Java.**"

  Implies → "**There is a student in your class who has not taken Java**."

- Symbolically $\neg\forall x\, J(x)$ and $\exists x\, \neg J(x)$ are equivalent, i.e., $\neg\forall x\, J(x) \equiv \exists x\, \neg J(x)$

Not all = Some

De Morgan's Laws for Quantifiers

- The rules for negating quantifiers are:

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.

- The reasoning in the table shows that:

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

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

- These are important. You will use these.

Negation of Quantifiers

- **English statement:**
  - **Nothing is perfect.**
  - Translation: $\neg\exists x\ Perfect(x)$

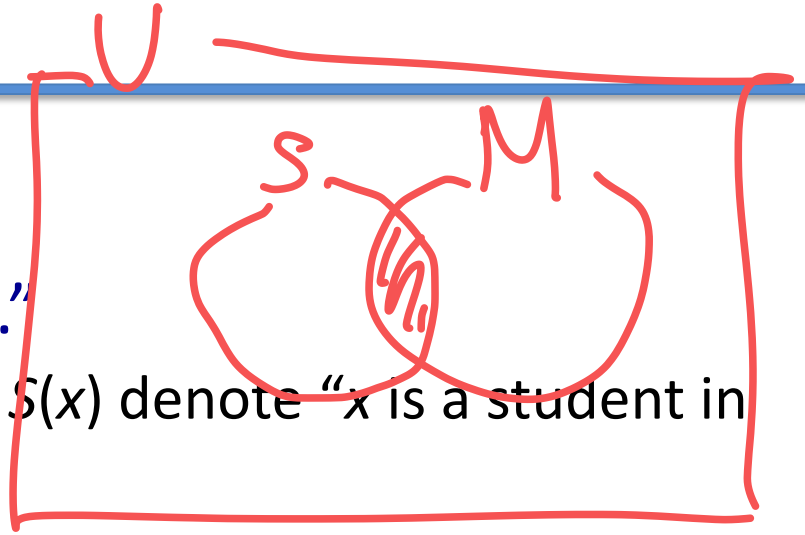
  - **Everything is imperfect.**
  - Translation: $\forall x\ \neg Perfect(x)$

- Conclusion (DeMorgan's Laws for quantifiers)
  $\neg\exists x\ Perfect(x)$ is equivalent to $\forall x\ \neg Perfect(x)$

Translation from English to Logic

Examples: $\exists x$ $S(x)$ $M(x)$

1. "~~Some~~ student in this class has visited Mexico."



Solution: Let $M(x)$ denote "x has visited Mexico" and $S(x)$ denote "x is a student in this class," and U be all people.

?

$\exists x(S(x) \land M(x))$ $S(x) \rightarrow M(x)$

2. "Every student in this class has visited Canada or Mexico."

Solution: Add $C(x)$ denoting "x has visited Canada." and U still be all people.

?

$\forall x \; S(x) \rightarrow (M(x) \lor C(x))$

Lewis Carroll Example

Charles Lutwidge Dodgson
(AKA Lewis Caroll)
(1832-1898)
The author of "Alice in Wonderland"

1. **"All lions are fierce(사나운)."**
2. **"Some lions do not drink coffee."**
3. **"Some fierce creatures do not drink coffee."**
   – Let P(x), Q(x), and R(x) be the statements "x is a lion," "x is fierce," and "x drinks coffee," respectively.

1. $\forall x\,(P(x) \rightarrow Q(x))$
2. $\exists x\,(P(x) \wedge \neg R(x))$
3. $\exists x\,(Q(x) \wedge \neg R(x))$

1.5 Nested Quantifiers

Nested Quantifiers

- More than one quantifier may be necessary to capture the meaning of a statement in the predicate logic.

Example: "Every real number has an inverse" is

$$\forall x \, \exists y (x + y = 0)$$

where the domains of x and y are the real numbers.

- We can also think of nested propositional functions:

  $\forall x \, \exists y (x + y = 0)$ can be viewed as $\forall x \; Q(x)$ where $Q(x)$ is $\exists y \; P(x, y)$ where $P(x, y)$ is $(x + y = 0)$

Nested Quantifiers

Translation Example:

- **There is a person who loves everybody.**
- Assume:
  - Variables x and y denote people
  - A predicate L(x,y) denotes: “x loves y”

- Then we can write in the predicate logic:

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

Order of Quantifiers

- The order of nested quantifiers does not matter if quantifiers are of the same type

- **Example:**
  - For all x and y, if x is a parent of y then y is a child of x
  - **Assume:**
    - Parent(x,y) denotes “x is a parent of y”
    - Child(x,y) denotes “x is a child of y”
  - Two equivalent ways to represent the statement:
    - $\forall x \forall y \text{Parent}(x, y) \rightarrow \text{Child}(y, x)$
    - $\forall y \forall x \text{Parent}(x, y) \rightarrow \text{Child}(y, x)$

Order of Quantifiers

- Let **Q(x,y)** be the statement "$x + y = 0$."
  - Assume that U is the real numbers. Then, $\forall x \exists y P(x,y)$ is true.
  - Then, is $\exists y \forall x P(x,y)$ also true ?

Order of Quantifiers

- The order of nested quantifiers matters if quantifiers are of different type

- Example: A predicate L(x,y) denotes: “x loves y”

  $\forall x\ \exists y\ L(x,y)$ is not the same as $\exists y \forall x\ L(x, y)$

- $\forall x\ \exists y\ L(x,y)$
  – Everybody loves someone
- $\exists y \forall x\ L(x, y)$
  – There is someone who is loved by everyone

Questions on Order of Quantifiers

Example 1: Let *U* be the real numbers,

Define $P(x,y) : x \cdot y = 0$

What is the truth value of the following:

1. $\forall x \forall y P(x,y)$

   Answer: False

2. $\forall x \exists y P(x,y)$

   Answer: True

3. $\exists x \forall y\ P(x,y)$

   Answer: True

4. $\exists x\ \exists\ y\ P(x,y)$

   Answer: True

Questions on Order of Quantifiers

Example 2: Let *U* be the real numbers,

Define $P(x,y) : x / y = 1$

What is the truth value of the following:

1. $\forall x \forall y P(x,y)$

   Answer: False

2. $\forall x \exists y P(x,y)$

   Answer: False (x=0)

3. $\exists x \forall y\, P(x,y)$

   Answer: False

4. $\exists x \exists y\, P(x,y)$

   Answer: True

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*

Section Summary

- Nested Quantifiers
- Order of Quantifiers
- Translating from Nested Quantifiers into English
- Translating Mathematical Statements into Statements involving Nested Quantifiers.
- Negating Nested Quantifiers.