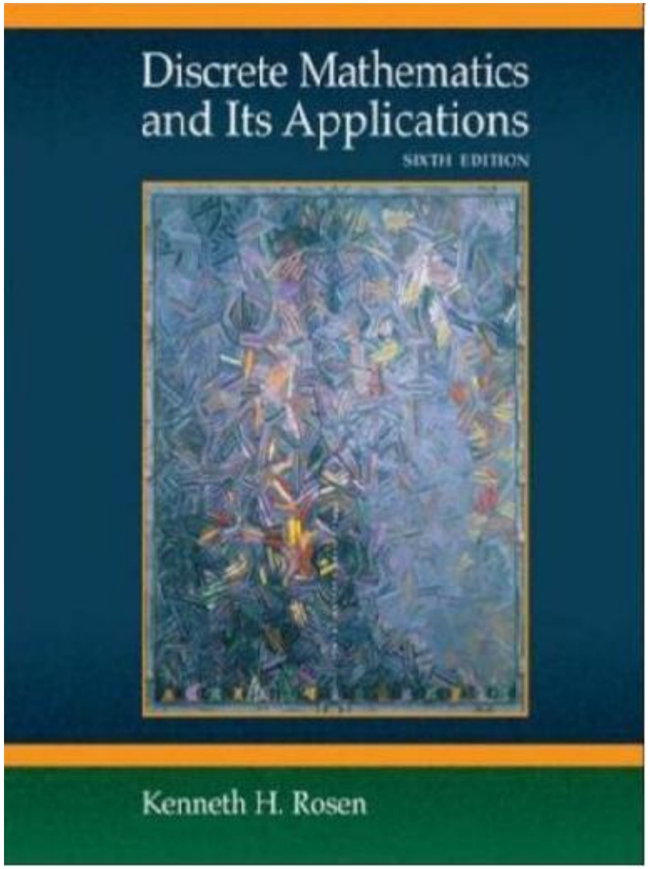
Jiangxi University of Science and Technology

Discrete Mathematics and Its Applications

Lecture 4:

Applications of Propositional Logic



Acknowledgement

- Most of these slides are adapted from ones created by Professor Bart Selman at Cornell University and Dr Johnnie Baker

Chapter 1 Exercises

Pages (28-30)

1(a,b,f)

4(b)

7,8

9(a,f)

11(a,f)

15

26,27

Applications of Propositional Logic: Summary

- Translating English to Propositional Logic
- System Specifications
- Boolean Searching
- Logic Puzzles
- Logic Circuits
- AI Diagnosis Method (Optional)

Translating English Sentences

- Steps to convert an English sentence to a statement in propositional logic
  - Identify atomic propositions and represent using propositional variables.
  - Determine appropriate logical connectives
- "If I go to Harry's or to the country, I will not go shopping."
  - p: I go to Harry's
  - q: I go to the country.
  - r: I will go shopping.

If p or q then not r.

$$(p \vee q) \rightarrow \neg r$$

Example

Problem: Translate the following sentence into propositional logic:

"You can access the Internet from campus only if you are a computer science major or you are not a freshman."

One Solution: Let *a*, *c*, and *f* represent respectively "You can access the internet from campus," "You are a computer science major," and "You are a freshman."

$$a \rightarrow (c \lor \neg f)$$

System Specifications

- System and Software engineers take requirements in English and express them in a precise specification language based on logic.

Example: Express in propositional logic:

"The automated reply cannot be sent when the file system is full"

Solution: One possible solution: Let p denote "The automated reply can be sent" and q denote "The file system is full."

$$q \rightarrow \neg p$$

江西理工大學
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Consistent System Specifications

Definition: A list of propositions is *consistent* if it is possible to assign truth values to the proposition variables so that each proposition is true.

Exercise: Are these specifications consistent?

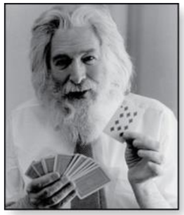
- "The diagnostic message is stored in the buffer or it is retransmitted."
- "The diagnostic message is not stored in the buffer."
- "If the diagnostic message is stored in the buffer, then it is retransmitted."

Solution: Let p denote "The diagnostic message is not stored in the buffer." Let q denote "The diagnostic message is retransmitted" The specification can be written as: $p \vee q$, $p \rightarrow q$, $\neg p$. When p is false and q is true all three statements are true. So the specification is consistent.

- What if "The diagnostic message is not retransmitted is added."

  Solution: Now we are adding $\neg q$ and there is no satisfying assignment. So the specification is not consistent.

Logic Puzzles



Raymond Smullyan
(Born 1919)

- An island has two kinds of inhabitants, *knights*, who always tell the truth, and *knaves*, who always lie.
- You go to the island and meet A and B.
  - A says "B is a knight."
  - B says "The two of us are of opposite types."

Example: What are the types of A and B?

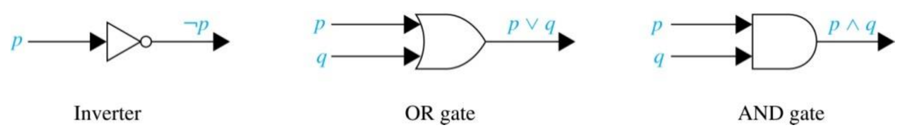
Solution: Let p and q be the statements that A is a knight and B is a knight, respectively. So, then $\neg p$ represents the proposition that A is a knave and $\neg q$ that B is a knave.

- If A is a knight, then p is true. Since knights tell the truth, q must also be true. Then $(p \land \neg q) \lor (\neg p \land q)$ would have to be true, but it is not. So, A is not a knight and therefore $\neg p$ must be true.
- If A is a knave, then B must not be a knight since knaves always lie. So, then both $\neg p$ and $\neg q$ hold since both are knaves.

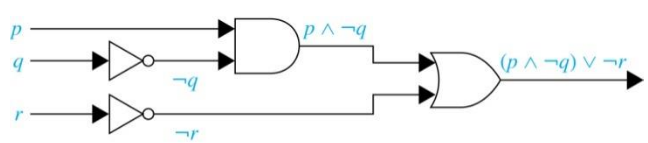
江西理工大學
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Logic Circuits
(Studied in depth in Chapter 12)

- Electronic circuits; each input/output signal can be viewed as a 0 or 1.
  - 0 represents **False**
  - 1 represents **True**
- Complicated circuits are constructed from three basic circuits called gates.



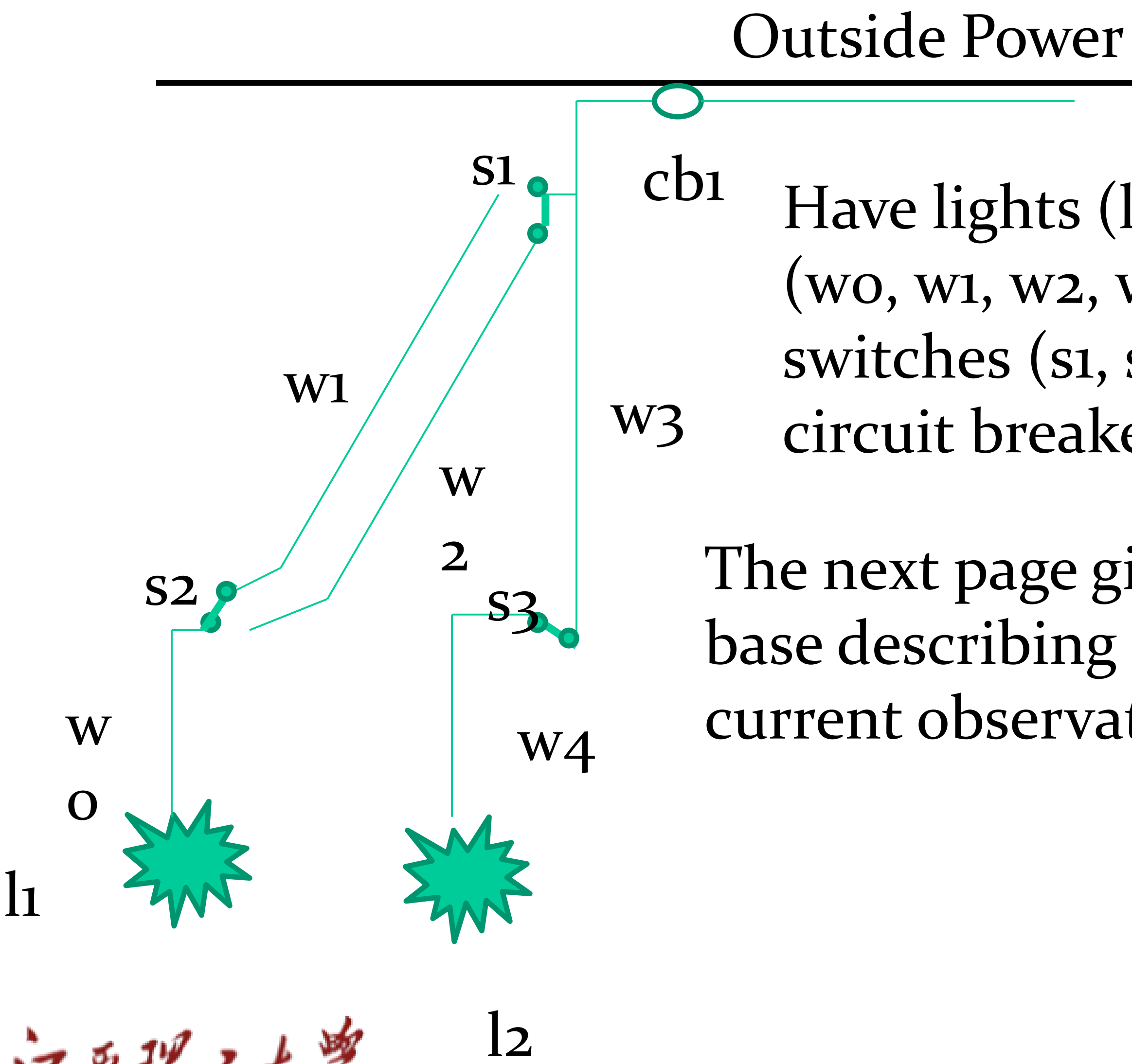
  - The inverter (**NOT gate**)takes an input bit and produces the negation of that bit.
  - The **OR gate** takes two input bits and produces the value equivalent to the disjunction of the two bits.
  - The **AND gate** takes two input bits and produces the value equivalent to the conjunction of the two bits.
- More complicated digital circuits can be constructed by combining these basic circuits to produce the desired output given the input signals by building a circuit for each piece of the output expression and then combining them. For example:



Diagnosis of Faults in an Electrical System (*Optional*)

- AI Example (from *Artificial Intelligence: Foundations of Computational Agents* by David Poole and Alan Mackworth, 2010)
- Need to represent in propositional logic the features of a piece of machinery or circuitry that are required for the operation to produce observable features. This is called the **Knowledge Base (KB)**.
- We also have observations representing the features that the system is exhibiting now.

Electrical System Diagram (optional)



Have lights (l1, l2), wires (w0, w1, w2, w3, w4), switches (s1, s2, s3), and circuit breakers (cb1)

The next page gives the knowledge base describing the circuit and the current observations.

Representing the Electrical System in Propositional Logic

- We need to represent our common-sense understanding of how the electrical system works in propositional logic.
- For example: “If l1 is a light and if l1 is receiving current, then l1 is lit.
  - lit_l1 → light_l1 ∧ live_l1 ∧ ok_l1
- Also: “If w1 has current, and switch s2 is in the up position, and s2 is not broken, then w0 has current.”
  - live_w0 → live_w1 ∧ up_s2 ∧ ok_s2
- This task of representing a piece of our common-sense world in logic is a common one in logic-based AI.

Knowledge Base (*opt*)

- live_outside
- light_l1
- light_l2
- live_l1 → live_w0
- live_w0 → live_w1 ∧ up_s2 ∧ ok_s2
- live_w0 → live_w2 ∧ down_s2 ∧ ok_s2
- live_w1 → live_w3 ∧ up_s1 ∧ ok_s1
- live_w2 → live_w3 ∧ down_s1 ∧ ok_s1
- live_l2 → live_w4
- live_w4 → live_w3 ∧ up_s3 ∧ ok_s3
- live_w3 → live_outside ∧ ok_cb1
- lit_l1 → light_l1 ∧ live_l1 ∧ ok_l1
- lit_l2 → light_l2 ∧ live_l2 ∧ ok_l2

We have outside power.
Both l1 and l2 are lights.

If s2 is ok and s2 is in a down position and w2 has current, then w0 has current.

Observations (*opt*)

- Observations need to be added to the KB
  - Both Switches up
    - up_s1
    - up_s2
  - Both lights are dark
    - ¬lit_l1
    - ¬ lit_l2

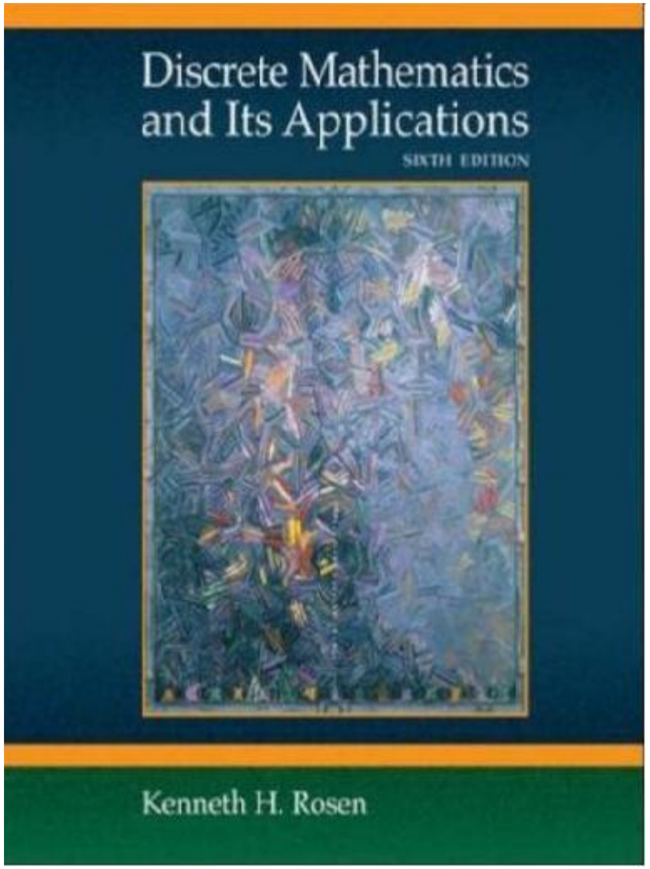
江西理工大學
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Diagnosis (*opt*)

- We assume that the components are working ok, unless we are forced to assume otherwise. These atoms are called *assumables*.
- The assumables (ok_cb1, ok_s1, ok_s2, ok_s3, ok_l1, ok_l2) represent the assumption that we assume that the switches, lights, and circuit breakers are ok.
- If the system is working correctly (all assumables are true), the observations and the knowledge base are consistent (i.e., satisfiable).
- The augmented knowledge base is clearly not consistent if the assumables are all true. The switches are both up, but the lights are not lit. Some of the assumables must then be false. This is the basis for the method to diagnose possible faults in the system.
- A diagnosis is a minimal set of assumables which must be false to explain the observations of the system.

Jiangxi University of Science and Technology

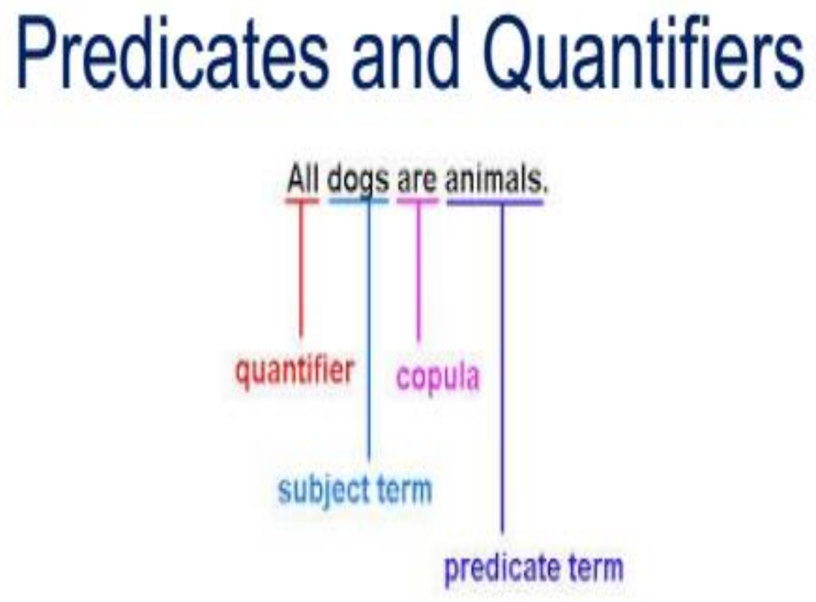
Predicates & Quantifiers



江西理工大學
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Predicates and Quantifiers

- Propositional logic, cannot adequately express the meaning of all statements in mathematics and in natural language.
- For example, suppose that we know that "Every computer connected to the university network is functioning properly"



Limitations of proposition logic

•No rules of propositional logic allow us to conclude the truth of the statement: “MATH3 is functioning properly,”

•where MATH3 is one of the computers connected to the university network. Likewise, we cannot use the rules of propositional logic to conclude from the statement: **“CS2 is under attack by an intruder,”** where CS2 is a computer on the university network, to conclude the truth of: “**There is a computer on the university network that is under attack by an intruder.**

Limitations of proposition logic

- Proposition logic cannot adequately express the meaning of statements
  - Suppose we know

"*Every* computer connected to the university network is functioning property"

  - No rules of propositional logic allow us to conclude

"*MATH3 is functioning property*"

where MATH3 is one of the computers connected to the university network

Predicates and Quantifiers

Predicates

"x is greater than 3"

This statement is neither true nor false when the value of the variable is not specified.

This statement has two pats

The fist part (subject) is the variable x

The second (predicate) is "is greater than 3"

We can denote this statement by P(x)

Where P denotes the **predicate** "is greater than 3"

Once a value has been assigned to x, the statement P(x) becomes a proposition and has a truth value. P is called Proposition function.

PREDICATE

- we will introduce a more powerful type of logic called predicate logic.
- We will see how predicate 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.

Predicates

- Statements involving variables, such as
  "$x > 3$," "$x = y + 3$," "$x + y = z$,"
  and
  "computer x is under attack by an intruder,"
  and
  "computer x is functioning properly,

are often found in mathematical assertions, in computer programs, and in system specifications.

These statements are neither true nor false when the values of the variables are not specified.

Predicates

we discuss the ways that propositions can be produced from such statements.

The statement "*x* is greater than 3" has two parts.

The first part, the variable *x*, is the subject of the statement.

The second part—the 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.**
- **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 a proposition and has a truth value.**

Quantifiers

Express the extent to which a predicate is true

- In English, *all*, *some*, *many*, *none*, *few*
- Focus on two types:

– Universal: a predicate is true for every element under consideration

– Existential: a predicate is true for there is one or more elements under consideration

- Predicate calculus:

the area of logic that deals with predicates and quantifiers

Predicates and Quantifiers

Predicates

let P(x) denote "is greater than 3"
What are the truth values of P(4) and P(2)?

let Q(x,y) denote "x=y+3"
What are the truth values of Q(1,2) and Q(3,0)?

let R(x,y,z) denote "x+y=z"
What are the truth values of R(1,2,3) and R(0,0,1)?

$P(x_1,x_2,x_3,\ldots\ldots\ldots,x_n)$

P is called n-place(n-ary) predicate.

Predicates and Quantifiers

Predicates

let **A**(c,n) denote “computer c is connected to network n”

Suppose that the computer **MATH1** is connected to network **CAMPUS2**, but not to network **CAMPUS1**

What are the truth values of
A(MATH1, CAMPUS1) and A(MATH1, CAMPUS2) ?

Predicates and Quantifiers

Predicates

Proposition functions(Predicates) occur in computer programs.

If $x>0$ then $x:=x+1$

$P(x)$: “$x>0$”

If $P(x)$ is true the assignment is executed

If $P(x)$ is false the assignment is not executed

江西理工大學
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Predicates and Quantifiers

Universal quantification
Which tell us that a predicate is true for every element under consideration.

existential quantification
Which tell us that there is one or more element under consideration for which the predicate is true.

The **area of logic** that deals with **predicates** and **quantifiers** is called **predicate calculus**.

Predicates and Quantifiers

The universal quantification of P(x) is the statement
"P(x) for all values of x in the domain"
$\forall x\, P(x)$ read as "for all x P(x)"
"for every x P(x)"

$\forall$ is called universal quantifier
The existential quantification of P(x) is the statement
"there exists an element x in the domain such that P(x)"
$\exists x\, P(x)$ read as "there is an x such that P(x)"
"there is at least one x such that P(x)"
"for some x P(x)"
$\exists$ is called existential quantifier

Predicates and Quantifiers

$\forall x\ P(x)$

When true $P(x)$ is true for every x.

When false there is an x for which $P(x)$ is false.

$\exists x\ P(x)$

When true there is an x for which $P(x)$ is true.

When false $P(x)$ is false for every x.

Predicates and Quantifiers

Let **Q(x) "x<2"**

What is the truth value of $\forall x\, Q(x)$ when the domain consists of all real numbers?

Q(x) is not true for every real number x,

for example Q(3) is false.

x=3 is a **counterexample** for the statement $\forall x\, Q(x)$

Thus $\forall x\, Q(x)$ is false.

江西理工大学
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Predicates and Quantifiers

Let Q(x) "$x^2>0$"

What is the truth value of $\forall x\, Q(x)$ when the domain consists of all integers?

Q(x) is not true for every integer number x, for example Q(0) is false.

x=0 is a counterexample for the statement

$\forall x\, Q(x)$: Thus $\forall x\, Q(x)$ is false.

Predicates and Quantifiers

What is the truth value of $\forall x\,(x^2 \geq x)$ when the domain

a) consists of all real number?

b) consists of all integers?

a) is **false** because $(0.5)^2 \not\geq 0.5$

$x^2 \geq x$ is false for all real numbers in the range $0<x<1$

b) is **true** because there are no integer x with $0<x<1$

Predicates and Quantifiers

Let Q(x) "x>3"

What is the truth value of ∃x Q(x) when the domain consists of all real numbers?

Q(x) is sometimes true , for example Q(4) is true.

Thus ∃x Q(x) is true.

∃x Q(x) is false iif there is no elements in the domain for which Q(x) is true

Predicates and Quantifiers

Let Q(x) “x=x+1”

What is the truth value of ∃x Q(x) when the domain consists of all real numbers?

Q(x) is false for every real number.

Thus ∃x Q(x) is **false**.

∃x Q(x) is false iif there is **no elements** in the domain for which **Q(x) is true** or the **domain** is **empty**.

Predicates and Quantifiers

When all the elements in the domain can be listed

$$x_1, x_2, x_3, x_4, \ldots\ldots., x_n$$

$\forall x\, Q(x)$ is the same as the conjunction

$$Q(x_1) \wedge Q(x_2) \wedge \ldots. \wedge Q(x_n)$$

$\exists x\, Q(x)$ is the same as the disjunction

$$Q(x_1) \vee Q(x_2) \vee \ldots. \vee Q(x_n)$$

Predicates and Quantifiers

Let Q(x) "x^2<10"

What is the truth value of ∀x Q(x) when the domain consists of the positive integers not exceeding 4?

∀x Q(x) is the same as the conjunction

$$Q(1) \wedge Q(2) \wedge Q(3) \wedge Q(4).$$

Q(4) is false

Thus ∀x Q(x) is false.

Predicates and Quantifiers

Let Q(x) “$x^2>10$”

What is the truth value of $\exists x\, Q(x)$ when the domain consists of the positive integers not exceeding 4?

$\exists x\, Q(x)$ is the same as the disjunction

$$Q(1) \vee Q(2) \vee Q(3) \vee Q(4).$$

Q(4) is true

Thus $\exists x\, Q(x)$ is true.

Predicates and Quantifiers

If domain consists of n (finite) object and we need to determine the truth value of $\forall x\, Q(x)$

- Loop through all n values of x to see if Q(x) is always true
- If you encounter a value x for which Q(x) is false exit the loop with $\forall x\, Q(x)$ is false
- otherwise $\forall x\, Q(x)$ is true

$\exists x\, Q(x)$

- Loop through all n values of x to see if Q(x) is true
- If you encounter a value x for which Q(x) is true exit the loop with $\exists x\, Q(x)$ is true
- Otherwise $\exists x\, Q(x)$ is false

Predicates and Quantifiers

You can define other quantifiers, for example

“there are exactly two”

“there are at least 100”

Uniqueness quantifier $\exists!$ $\exists 1$

“there exists a unique x such that P(x) is true”

“there is exactly one”

“there is one and only one”

Precedence of quantifiers

$\forall$ $\exists$ have higher precedence than all logical operators $\neg \wedge \vee \rightarrow \leftrightarrow$

Quantifiers with restricted domain

What do these statements mean (domain real)

- $\forall x<0\ (x2>0)$ same as $\forall x\ (x<0 \rightarrow x2>0)$

"the square of a negative real number is positive"

- $\forall y\neq 0\ (y^3 \neq 0)$ same as $\forall y(y\neq 0 \rightarrow y^3 \neq 0)$

"the cube of every nonzero real number is nonzero"

- $\exists z>0\ (z2=2)$ same as $\exists z(z>0 \wedge z2=2)$

"there is a positive square root of 2"

Binding variables

- $\exists x\ (x+y=1)$

The variable x is bounded by the existential quantification $\exists x$ and the variable y is **free**

- $\exists x\ (P(x) \land Q(x)) \lor \forall x\ R(x)$

All variables are bounded

The **scope** of the **first** quantifier $\exists x$ is the expression $P(x) \land Q(x)$

second quantifier $\forall x$ is the expression $R(x)$

existential quantifier binds the variable x in $P(x) \land Q(x)$

Universal quantifier binds the variable x in $R(x)$

Logical equivalence involving quantifiers

Statements involving predicates and quantifier are logically equivalent iff they have the same truth value

Show that $\forall x\, (P(x) \wedge Q(x)) \equiv \forall x\, P(x) \wedge \forall x\, Q(x)$

- suppose that $\forall x\, (P(x) \wedge Q(x))$ is true,
  this means that if a is in the domain then $P(a) \wedge Q(a)$ is true

Hence, both P(a) and Q(a) are true for every element in the domain

So $\forall x\, P(x)$ and $\forall x\, Q(x)$ are both true.

This means that $\forall x\, P(x) \wedge \forall x\, Q(x)$ is true.

江西理工大學
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Logical equivalence involving quantifiers

Show that $\forall x\, (P(x) \wedge Q(x)) \equiv \forall x\, P(x) \wedge \forall x\, Q(x)$

- suppose that $\forall x\, P(x) \wedge \forall x\, Q(x)$ is true, it follows that both $\forall x\, P(x)$ and $\forall x\, Q(x)$ are true hence,
- if a is in the domain then P(a) is true and Q(a) is true

This means that $\forall x\, (P(x) \wedge Q(x))$ is true.

Now we can conclude that

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

Logical equivalence involving quantifiers

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

- The universal quantifier can be distributed over the a conjunction ($\wedge$).
- The universal quantifier can be distributed over the a disjunction ($\vee$).
- The existential quantifier can not be distributed over the a conjunction ($\wedge$) and disjunction ($\vee$).

Negating quantified expression

"Every student in your class has taken a course in calculus" $\forall x\, P(x)$
Where, $P(x)$ is "x has taken a course in calculus"
and the domain consists of the student in your class

Negation
"It is not the case that every student in your class has taken a course in calculus"
or
"There is a student in your class who has not taken a course in calculus" $\exists x \neg P(x)$

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

Negating quantified expression

"There is a student in this class who has taken a course in calculus" $\exists x\, Q(x)$

Where, $Q(x)$ is "x has taken a course in calculus"

and the domain consists of the student in your class

Negation

"It is not the case that there is a student in this class who has taken a course in calculus"

or

"Every student in this class has not taken a course in calculus" $\forall x \neg Q(x)$

"Not all students in this class have taken a course in calculus" is not used.

$$\neg \exists x\, Q(x) \equiv \forall x \neg Q(x)$$

江西理工大學

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Negating quantified expression

Show that (homework)

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

Negating quantified expression

De Morgan’s laws for quantifiers

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

The negation is true when for every x, P(x) is false

The negation is false when there is an x for which P(x) is true

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

The negation is true when there is an x for which P(x) is false

The negation is false when for every x, P(x) is true

Negating quantified expression

De Morgan's laws for quantifiers

When the domain of a predicate $Q(x)$ consists of n elements x_1, x_2 ,x_3, x_4, ……., x_n

$\forall x\ Q(x)$ is the same as the conjunction

$$Q(x_1) \wedge Q(x_2) \wedge \ldots. \wedge Q(x_n)$$

$\neg\forall x\ Q(x)$ is the same as the disjunction

$$\neg Q(x_1) \vee \neg Q(x_2) \vee \ldots. \vee \neg Q(x_n)$$

$\exists x\ Q(x)$ is the same as the disjunction

$$Q(x_1) \vee Q(x_2) \vee \ldots. \vee Q(x_n)$$

$\neg\ \exists x\ Q(x)$ is the same as the conjunction

$$\neg Q(x_1) \wedge \neg Q(x_2) \wedge \ldots. \wedge \neg Q(x_n)$$

Negating quantified expression

Examples: what are the negation of

- $\forall x\ (x^2 > x)$
- $\exists x\ (x^2 = x)$

- $\forall x\ (x^2 > x)$

$\neg\forall x\ (x^2 > x) \equiv \exists x\ \neg(x^2 > x) \equiv \exists x\ (x^2 \le x)$

- $\exists x\ (x^2 = x)$

$\neg\exists x\ (x^2 = x) \equiv \forall x\ \neg(x^2 = x) \equiv \forall x\ (x^2 \ne x)$

Negating quantified expression

Show that

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

$$
\begin{aligned}
\neg\forall x\,[P(x) \rightarrow Q(x)] &\equiv \exists x\,\neg[P(x) \rightarrow Q(x)] \\
&\equiv \exists x\,\neg[\neg P(x) \vee Q(x)] \\
&\equiv \exists x\,[\neg\neg P(x) \wedge \neg Q(x)] \\
&\equiv \exists x\,[P(x) \wedge \neg Q(x)]
\end{aligned}
$$

Translating from English into Logical Expressions

Express the statement “Every student in the class has studied calculus” using predicates and quantifiers.

- Rewrite the statement to identify the appreciate quantifiers to use

“For every student in the class,that student has studied calculus”

- Introduce the variable x

“For every student x in the class,x has studied calculus”

- introduce C(x) “x has studied calculus”

$\forall x\ C(x)$

The domain for x consists of the students in the class

Translating from English into Logical Expressions

If we change the domain to consists of people

"For every person x, if person x is a student in the class,then x has studied calculus"

C(x) "x has studied calculus"
S(x) "person x is a student in the class"

$\forall x\ (S(x) \rightarrow C(x))$

Note $\forall x\ (S(x) \wedge C(x))$ is wrong
All people are students in this class and have studied calculus

Translating from English into Logical Expressions

Express the statements “some students in the class has visited Cairo” , “every student in the class has visited either Aswan or Cairo ” using predicates and quantifiers.

- “some students in the class has visited Cairo”

some students ~ there is a student

C(x) “x has visited Cairo”

$$\exists x\, C(x)$$

- “every student in the class has visited either Aswan or Cairo ”

A(x) “x has visited Aswan ”

$$\forall x\, (C(x) \vee A(x))$$

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Translating from English into Logical Expressions

System specifications

Express the statement

using predicates and quantifiers

S(m,y) “Mail message m is larger than y megabyte”

Where m has the domain of all mail message

y is a positive real number

C(m) “Mail message m will be compressed”

$\forall m\ (S(m,1) \rightarrow C(m))$

江西理工大学
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Translating from English into Logical Expressions

System specifications

Express the statement “If a user is active, at least one network link will be available” using predicates and quantifiers.

A(u) “user u is active”

Where u has the domain of all users

S(n,x) “network link n is in state x”

Where n has the domain of all network links

x has the domain of all possible states

for a network link

$$\exists u\, A(u) \rightarrow \exists n\, S(n, \text{available})$$

Chapter 1 Exercises

Pages (46-50)

1-4

5(a,d)

8(b,c)

10

12(b,d,g)

17

20(e)

21-22

24, 29, 31 ,36

40-42

43, 48