

**DISCRETE**

**MATHEMATICS AND**

**ITS APPLICATIONS Book: Discrete Mathematics and Its Applications Author: Kenneth H. Rosen**

**Sixth Edition**

**McGraw-Hill International Edition**

****

**Chapter 1**

**The Foundations:**

**Logic and Proofs**

****

**Objectives**

● Explain what makes up a correct mathematical argument

● Introduce tools to construct arguments



**Contents**

1.1-Propositional Logic

1.2-Propositonal Equivalences 1.3-Predicates and Quantifiers 1.4-Nested Quantifiers

1.5-Rules of Inference



**1.1- Propositional Logic**

1.1.1- Definitions and Truth Table 1.1.2- Precedence of Logical Operators



**1.1.1- Definitions and Truth Table**

● **Proposition** is a declarative sentence that is either ***true*** or ***false*** but ***not both***.

● Proposition is a sentence that declares a ***fact***. ● Examples:

\* Bà Tưng is one of descendants of Bà Trưng \* Ha Noi is not the capital of Vietnam

\* 1+5 < 4

\* What time is it? \* X+Y=Z

OK

No OK



**1.1.1- Definitions…**

● **Truth table** – I am a girl

|  |
| --- |
| **p**  **True/ T / 1** |
| **False / F / 0** |



**1.1.1- Definitions…**

● Negation of proposition p is the statement “ It is not case that p”.

● Notation: p (or )

*p*

|  |  |
| --- | --- |
| **p**  1 | 0 |
| 0 | 1 |



**1.1.1- Definitions…**

● Conjunction of propositions p and q is the proposition “ p and q” and denoted by p***^***q

|  |  |  |
| --- | --- | --- |
| **p**  **0** | **q**  **0** | **p*^*q**  **0** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |



**1.1.1- Definitions…**

● Disjunction of propositions p and q is the proposition “ p or q” and denoted by p ***v*** q

|  |  |  |
| --- | --- | --- |
| **p**  **0** | **q**  **0** | **pq**  **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **1** |



**1.1.1- Definitions…**

● Exclusive-or (xor) of propositions p and q, denoted by p ⊕ q

|  |  |  |
| --- | --- | --- |
| **p**  **0** | **q**  **0** | **p** ⊕ **q**  **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |

⊕ q



**1.1.1- Definitions…**

● Implication: *p → q* (p implies q) ● p: *hypothesis / antecedent / premise* ● q: *conclusion* / *consequence* ● *p → q can be expressed as:* - *q if p*

- *If p, then q*

- *p is sufficient condition for q* - *q is necessary condition for p*

|  |  |  |
| --- | --- | --- |
| **p**  **0** | **q**  **0** | **p → q**  **1** |
| **0** | **1** | **1** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |

“If 1 + 1 = 3, then dogs can fly” 🡪TRUE

(p  q)

p=0, q=0 ,

so (pq) is true.



**1.1.1- Definitions…**

● Biconditional statement p  q is the proposition “ p if and only if q”

● p → q (p ***only if*** q) and pq (p ***if*** q)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **p**  **0** | **q**  **0** | **p→q**  **1** | **q→p**  **1** | **(p→q) ^ (q→p)**  **1** | **p ↔ q**  **1** |
| **0** | **1** | **1** | **0** | **0** | **0** |
| **1** | **0** | **0** | **1** | **0** | **0** |
| **1** | **1** | **1** | **1** | **1** | **1** |



**1.1.2- Precedence of Logical Operators**

(1) Parentheses from inner to outer

**(2) ¬**

**(3) ^**

(4) v

(5) →

(6) ↔



**1.2- Propositional Equivalences**

1.2.1- Tautology and Contradiction 1.2.2- Logical Equivalences

1.2.3- De Morgan’s Laws



**1.2.1- Tautology and Contradiction**

● Tautology is a proposition that is ***always true*** ● Contradiction is a proposition that is ***always false***

● When p ↔ q is tautology, we say “p and q are called logically equivalence”. Notation: p ≡ q



Example 3 p.23

● *Show that p  q and ¬p v q are logically equivalent.*

**

**

**1.2.2- Logical Equivalences…**

|  |  |
| --- | --- |
| **Equivalence**  p ^ T ≡ p p v F ≡ p | **Name**  Identity laws |
| p v T ≡ T p ^ F ≡ F | Domination Laws |
| p v p ≡ p p ^ p ≡ p | Idempotent Laws |
| ¬(¬p) ≡ p | Double Negation Laws |
| p v q ≡ q v p p ^ q ≡ q ^ p | Commutative Laws |
| (p v q) v r ≡ p v (q v r)  (p ^ q) ^ r ≡p^(q^r) | Associative Laws |
| pv (q^r) ≡ (pvq) ^ (pvr)  p^ (qvr) ≡ (p^q) v (p^r) | Distributive Laws |



**1.2.2- Logical Equivalences…**

|  |  |
| --- | --- |
| **Equivalence**  ¬ (p^q) ≡ ¬pv¬q ¬(pvq) ≡ ¬p^¬q | **Name**  De Morgan Laws |
| pv (p^q)≡ p p^(pvq)≡ p | Absorption Laws |
| pv¬p ≡ T p^¬p≡ F | Negation Laws |



**1.2.2- Logical Equivalences…**

|  |  |
| --- | --- |
| **Equivalences**  p→q ≡ ¬pvq | **Equivalences**  p↔q ≡ (p→q) ^ (q→p) |
| p→q ≡ ¬q → ¬p | p↔q ≡ ¬p ↔ ¬q |
| pvq ≡ ¬ p → q | p↔q ≡ (p ^ q) v (¬p ^ ¬q) |
| p^q ≡ ¬ (p → ¬q) | ¬ (p↔q) ≡ p↔ ¬q |
| ¬(p→q) ≡ p^¬q |  |
| (p→q) ^(p→r) ≡ p → (q^r) |  |
| (p→r) ^ (q→r) ≡ (pvq) → r |  |
| (p→q) v (p→r) ≡ p→ (qvr) |  |
| (p→r) v (q→r) ≡ (p^q) → r |  |



**1.3- Predicates and Quantifiers**

● Introduction

● Predicates

● Quantifiers



**1.3.1- Introduction**

● A type of logic 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 > 0

**1.3.2- Predicates – vị từ**

● P(X)=“X is a prime number” , called propositional function at X.

● P(2)=”2 is a prime number” ≡True ● P(4)=“4 is a prime number” ≡False



**1.3.2- Predicates – vị từ**

● Q(X1,X2,…,Xn) , n-place/ n-ary predicate ● Example: “x=y+3” 🡺 Q(x,y)

Q(1,2) ≡ “1=2+3” ≡ false

Q(5,2) ≡ “5=2+3” ≡ true



**1.3.2- Predicates…**

● Predicates are pre-conditions and post

conditions of a program. ● If x>0 then x:=x+1 – Predicate: “x>0” 🡺 P(x) – Pre-condition: P(x)

– Post-condition: P(x)

● T:=X;

X:=Y;

Y:=T;

Pre-condition (P(…)) : condition describes valid input.

Post-condition (Q(…)) : condition describes valid output of the codes. **Show the verification that a program always produces the desired output:** P(…) is true

Executing Step 1.

Executing Step 2.

…..

Q(…) is true

- Pre-condition: “x=a and y=b” 🡺 P(x, y) - Post-condition: “x=b and y=a” 🡺 Q(x, y)



**1.3.3- Quantifiers – Lượng từ**

● The words in natural language: all, some, many, none, few, ….are used in quantifications.

● Predicate Calculus : area of logic that deals with predicates and quantifiers.

● The ***universal quantification*** *of P(x) is the statement “P(x) for all values of x in the domain”.* Notation : ∀xP(x) ● The ***existential quantification*** *of P(x) is the statement “There exists an element x in the domain such that P(x)”.* Notation : ∃xP(x)

● **Uniqueness quantifier**: ∃!x P(x) or ∃1xP(x)

● ∀xP(x) v Q(y) :

● x is a bound variable

● y is a free variable

**1.3.4- Quantifiers and Restricted Domains**

∀x<0(x2 > 0), ∀y ≠ 0(y3 ≠ 0), ∃z>0(z2 =2) 🡸🡺

∀x(x<0  x2 > 0), ∀y(y ≠ 0  y3 ≠ 0), ∃z(z>0 ^ z2 =2)

Restricted domains



**1.3.5- Precedence of Quantifiers**

● Quantifiers have higher precedence than all logical operators from propositional calculus.

● ∀xP(x) v Q(x) 🡺 (∀xP(x)) v Q(x)

● ∀ has higher precedence. So, ∀ affects on P(x) only.

**1.3.6- Logical Equivalences Involving Quantifiers**

Statements involving predicates and quantifiers are ***logically equivalent if and only if they have the same truth value*** no matter which predicates are substituted into the statements and which domain of discourse is used for the variables in these propositional functions. ● ∀x (P(x) ^ Q(x)) ≡ ∀xP(x) ^ ∀xQ(x)

– Proof: page 39

|  |  |  |  |
| --- | --- | --- | --- |
| **Expression**  ¬∃xP(x) | **Equivalence**  ∀x ¬P(x) | **Expression**  ∃xP(x) | **Negation**  ∀x ¬P(x) |
| ¬ ∀xP(x) | ∃x ¬P(x) | ∀xP(x) | ∃x ¬P(x) |



**1.3.7- Translating**

● For every student in the class has studied calculus

● For every student in the class, that student has studied calculus

● For every student x in the class, x has studied calculus

● ∀x (S(x) → C(x))



**Negating nested quantifiers**

¬ ∀x∃y(xy=1) ≡ ∃x ¬∃y (xy=1) // De Morgan laws ≡ (∃x) (∀y) ¬(xy=1)

≡ (∃x) (∀y) (xy ≠ 1)





**1.5- Rules of Inference**

● Definitions

● Rules of Inferences



**1.5.1- Definitions**

● Proposition 1 // Hypothesis

● Proposition 2 ● Proposition 3 ● Proposition 4 ● Proposition 5 ● ………

● Conclusion

Arguments 2,3,4 are premises of argument 5

Arguments

Propositional Equivalences



**1.5.2- Rules Inferences**

|  |  |  |
| --- | --- | --- |
| **Rule**  p  p →q  ∴q | **Tautology**  [p^ (p→q)] → q  **You work hard**  If **you work hard** then **you will pass the examination**  ∴**you will pass the examination** | **Name**  Modus ponen |
| ¬q  p → q  ∴¬p | [¬q ^(p → q)] → ¬p  **She did not get a prize**  If **she is good at learning she will get a prize**  ∴**She is not good at learning** | Modus tollen |



**1.5.2- Rules Inferences**

|  |  |  |
| --- | --- | --- |
| **Rule**  p  →q  q →r  ∴p →r | **Tautology**  [(p →q) ^(q →r)] →(p→r)  If **the prime interest rate goes up** then **the stock prices go down**.  If **the stock prices go down** then **most people are unhappy**.  If **the prime interest rate goes up** then **most people are unhappy**. | **Name**  Hypothetical syllogism |

**Rules Inferences…**

|  |  |  |
| --- | --- | --- |
| **Rule**  pvq  ¬p  ∴q | **Tautology**  **[(**pvq**)** ^¬p**]** → **q**  **Power puts off** or **the lamp is**  **malfunctional**  **Power doesn’t put off**  **the lamp is malfunctional** | **Name**  Disjunctive  syllogism |
| p  ∴pvq | **p →(pvq)**  **It is below freezing now**  **It is below freezing now or raining now** | Addition |
| p^q  ∴p | (p^q) →p  **It is below freezing now and raining now**  **It is below freezing now** | Simplication |



**Rules Inferences…**

|  |  |  |
| --- | --- | --- |
| **Rule**  p  q  ∴p^q | **Tautology**  [(p) ^(q)) → (p^q) | **Name**  Conjunction |
| pvq  ¬pvr  ∴qvr | [(pvq) ^(¬pvr)] →(qvr)  **Jasmin is skiing OR it is not snowing It is snowing OR Bart is playing hockey Jasmin is skiing OR Bart is playing hockey** | Resolution |



**1.5.3- Fallacies**

● If **you do every problem in this book** then you will learn discrete mathematic

You learned mathematic

(p → q) ^q

=(¬ p v q) ^ q

(absorption law)

= q

 No information for p

p can be true or false 🡺 You may learn discrete mathematic but you might do some problems only.



**Fallacies…**

● (p → q)^q  p is not a tautology ( it is false when p = 0, q = 1) ● (p  q)^¬p  ¬q is not a tautology (it is false when p = 0, q = 1)

**1.5.4- Rules of Inference for Quantified Statements**

|  |  |
| --- | --- |
| **Rule**  ∀xP(x)  ∴P(c) | **Name**  Universal Instantiation |
| P(c) for arbitrary c  ∴∀xP(x) | Universal generalization |
| ∃xP(x)  ∴P(c) for some element c | Existential instantiation |
| P(c) for some element c ∴∃xP(x) | Existential generalization |



**Rules of Inference for Quantified Statements…**

● “All student are in this class had taken the course PFC”

● “HB is in this class”

● “Had HB taken PFC?”

● ∀x(P(x) → Q(x)) ● P(HB) → Q(HB) ● P(HB)

● Q(HB) // conclusion

Premise

Universal Instantiation Modus ponens



**Summary**

● Propositional Logic

● Propositional Equivalences ● Predicates and Quantifiers ● Nested Quantifiers

● Rules and Inference



**THANK YOU**