Department of Computer & Information Sciences
Pakistan Institute of Engineering and Applied Sciences

Propositional Equivalence

Propositional Equivalence

- Two syntactically (i.e., textually) different compound propositions may be the semantically identical (i.e., have the same meaning). We call them equivalent.

Tautologies, Contradictions, Contingencies

- Tautology:
  - A compound proposition is called a tautology if no matter what truth values its atomic propositions have, its own truth value is T.
    - For example: $p \vee \neg p$ (Law of excluded middle)

Tautologies, Contradictions, Contingencies

- Contradiction:
  - The opposite to a tautology, is a compound proposition that's always false –a contradiction.
    - For example: $p \wedge \neg p$

Tautologies, Contradictions, Contingencies

- Contigency:
  - On the other hand, a compound proposition whose truth value isn't constant is called a contingency.
    - For example: $p \rightarrow \neg p$

Tautologies and contradictions

The easiest way to see if a compound proposition is a tautology/contradiction is to use a truth table.

p	$\neg p$	$p \vee \neg p$
F	T	T
T	F	T

p	$\neg p$	$p \wedge \neg p$
F	T	F
T	F	F

Tautology Example

Example:

Demonstrate that is a tautology

$$[\neg p \wedge (p \vee q)] \rightarrow q$$

1. Using a truth table – show that $[\neg p \wedge (p \vee q)] \rightarrow q$ is always true

p	q	¬p	p ∨q	¬p ∧(p ∨q)	[¬p ∧(p ∨q)]→q
T	T				
T	F				
F	T				
F	F				

L3

Tautology by Truth Table

p	*q*	¬*p*	*p* ∨*q*	¬*p* ∧(*p* ∨*q*)	[¬*p* ∧(*p* ∨*q*)]→*q*
T	T	F			
T	F	F			
F	T	T			
F	F	T			

L3

Tautology by Truth Table

p	*q*	¬*p*	*p* ∨*q*	¬*p* ∧(*p* ∨*q*)	[¬*p* ∧(*p* ∨*q*)]→*q*
T	T	F	T		
T	F	F	T		
F	T	T	T		
F	F	T	F		

L3

Tautology by Truth Table

p	*q*	¬*p*	*p* ∨*q*	¬*p* ∧(*p* ∨*q*)	[¬*p* ∧(*p* ∨*q*)]→*q*
T	T	F	T	F	
T	F	F	T	F	
F	T	T	T	T	
F	F	T	F	F	

L3

Tautology by Truth Table

p	*q*	¬*p*	*p* ∨*q*	¬*p* ∧(*p* ∨*q*)	[¬*p* ∧(*p* ∨*q*)]→*q*
T	T	F	T	F	T
T	F	F	T	F	T
F	T	T	T	T	T
F	F	T	F	F	T

L3

Logical Equivalences

- Two compound propositions p, q are logically equivalent if their biconditional joining $p \leftrightarrow q$ is a tautology. Logical equivalence is denoted by $p \Leftrightarrow q$.

Logical Equivalences

- Logical Equivalence of Conditional and Contrapositive:
  - The contrapositive of a logical implication is the reversal of the implication, while negating both components.
    - i.e. the contrapositive of p →q is ¬q →¬p

p	q	p →q	p	q	¬q	¬p	¬q→¬p

Logical Equivalences

- Logical Equivalence of Conditional and Contrapositive:
  - The easiest way to check for logical equivalence is to see if the truth tables of both variants have identical last columns:

p	q	p →q	p	q	¬q	¬p	¬q→¬p
T	T	T					
T	F	F					
F	T	T					
F	F	T					

Logical Equivalences

- Logical Equivalence of Conditional and Contrapositive:
  - The easiest way to check for logical equivalence is to see if the truth tables of both variants have identical last columns:

p	q	p →q	p	q	¬q	¬p	¬q→¬p
T	T	T	T	T	F	F	
T	F	F	T	F	T	F	
F	T	T	F	T	F	T	
F	F	T	F	F	T	T	

Logical Equivalences

- Logical Equivalence of Conditional and Contrapositive:
  - The easiest way to check for logical equivalence is to see if the truth tables of both variants have identical last columns:

p	q	p →q	p	q	¬q	¬p	¬q→¬p
T	T	T	T	T	F	F	T
T	F	F	T	F	T	F	F
F	T	T	F	T	F	T	T
F	F	T	F	F	T	T	T

Logical Equivalences

- Logical Equivalence of Conditional and Contrapositive:
  - The easiest way to check for logical equivalence is to see if the truth tables of both variants have identical last columns:

p	q	p →q	p	q	¬q	¬p	¬q→¬p
T	T	T	T	T	F	F	T
T	F	F	T	F	T	F	F
F	T	T	F	T	F	T	T
F	F	T	F	F	T	T	T

Non-Equivalence of Conditional and Converse

- The converse of a logical implication is the reversal of the implication. i.e. the converse of p →q is q →p.
  - e.g., The converse of "If Donald is a duck then Donald is a bird." is "If Donald is a bird then Donald is a duck."

Non-Equivalence of Conditional and Converse

p	q	p →q	q→p	(p →q)↔(q→p)
T	T			
T	F			
F	T			
F	F			

Non-Equivalence of Conditional and Converse

p	q	p →q	q→p	(p →q)↔(q→p)
T	T	T		
T	F	F		
F	T	T		
F	F	T		

Non-Equivalence of Conditional and Converse

p	q	p →q	q→p	(p →q)↔(q→p)
T	T	T	T	
T	F	F	T	
F	T	T	F	
F	F	T	T	

Non-Equivalence of Conditional and Converse

p	q	p →q	q→p	(p →q)↔(q→p)
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

Derivational Proof Techniques

- When compound propositions involve more and more atomic components, the size of the truth table for the compound propositions increases
  - How many rows are required to construct the truth-table of:
    - ((q↔(p→r)) ∧ (¬(s∧r)∨¬t)) → (¬q→r)
- 32 rows, each additional variable doubles the number of rows

Equivalence Laws

- Identity Laws:
  - p∧T ↔ p
  - p∨F ↔ p
- Domination Laws:
  - p∨T ↔ T
  - p∧F ↔ F
- Idempotent Laws:
  - p∨p ↔ p
  - p∧p ↔ p
- Double negation:
  - ¬¬p ↔ p

Equivalence Laws

- Commutative Laws:
  - $p \vee q \leftrightarrow q \vee p$
  - $p \wedge q \leftrightarrow q \wedge p$
- Associative Laws:
  - $(p \vee q) \vee r \leftrightarrow p \vee (q \vee r)$
  - $(p \wedge q) \wedge r \leftrightarrow p \wedge (q \wedge r)$
- Distributive Laws:
  - $p \vee (q \wedge r) \leftrightarrow (p \vee q) \wedge (p \vee r)$
  - $p \wedge (q \vee r) \leftrightarrow (p \wedge q) \vee (p \wedge r)$

Tables of Logical Equivalences

p. 17, Rosen

- Excluded middle
- Negating creates opposite
- Definition of implication in terms of Not and Or

TABLE 6 Some Useful Logical Equivalences.

$p \vee \neg p \Longleftrightarrow \mathbf{T}$	ULE 1
$p \wedge \neg p \Longleftrightarrow \mathbf{F}$	ULE 2
$(p \rightarrow q) \Longleftrightarrow (\neg p \vee q)$	ULE 3

Equivalence Laws

- De Morgan's:
  $\neg(p \wedge q) \leftrightarrow \neg p \vee \neg q$
  $\neg(p \vee q) \leftrightarrow \neg p \wedge \neg q$
- Trivial tautology/contradiction:
  $p \vee \neg p \leftrightarrow T$
  - $p \wedge \neg p \leftrightarrow F$

Augustus De Morgan (1806-1871)

DeMorgan Identities

- DeMorgan's identities allow for simplification of negations of complex expressions
- Conjunctional negation:
  - $\neg(p1 \wedge p2 \wedge \ldots \wedge pn) \leftrightarrow (\neg p1 \vee \neg p2 \vee \ldots \vee \neg pn)$
  - "It's not the case that all are true iff one is false."
- Disjunctional negation:
  - $\neg(p1 \vee p2 \vee \ldots \vee pn) \leftrightarrow (\neg p1 \wedge \neg p2 \wedge \ldots \wedge \neg pn)$
  - "It's not the case that one is true iff all are false."

Tautology example

Demonstrate that is a tautology using truth table of logical equivalences

$$[\neg p \wedge (p \vee q)] \rightarrow q$$

Tautology by proof

$[\neg p \wedge (p \vee q)] \rightarrow q$

Tautology by proof

$[\neg p \wedge (p \vee q)] \to q$

$\Leftrightarrow [(\neg p \wedge p) \vee (\neg p \wedge q)] \to q$ — Distributive

31

L3

Tautology by proof

$[\neg p \wedge (p \vee q)] \to q$

$\Leftrightarrow [(\neg p \wedge p) \vee (\neg p \wedge q)] \to q$ — Distributive

$\Leftrightarrow [\text{F} \vee (\neg p \wedge q)] \to q$ — ULE

32

L3

Tautology by proof

$[\neg p \wedge (p \vee q)] \to q$

$\Leftrightarrow [(\neg p \wedge p) \vee (\neg p \wedge q)] \to q$ — Distributive

$\Leftrightarrow [\text{F} \vee (\neg p \wedge q)] \to q$ — ULE

$\Leftrightarrow [\neg p \wedge q] \to q$ — Identity

33

L3

Tautology by proof

$[\neg p \wedge (p \vee q)] \to q$

$\Leftrightarrow [(\neg p \wedge p) \vee (\neg p \wedge q)] \to q$ — Distributive

$\Leftrightarrow [\text{F} \vee (\neg p \wedge q)] \to q$ — ULE

$\Leftrightarrow [\neg p \wedge q] \to q$ — Identity

$\Leftrightarrow \neg [\neg p \wedge q] \vee q$ — ULE

34

L3

Tautology by proof

$[\neg p \wedge (p \vee q)] \to q$

$\Leftrightarrow [(\neg p \wedge p) \vee (\neg p \wedge q)] \to q$ — Distributive

$\Leftrightarrow [\text{F} \vee (\neg p \wedge q)] \to q$ — ULE

$\Leftrightarrow [\neg p \wedge q] \to q$ — Identity

$\Leftrightarrow \neg [\neg p \wedge q] \vee q$ — ULE

$\Leftrightarrow [\neg(\neg p) \vee \neg q] \vee q$ — DeMorgan

35

L3

Tautology by proof

$[\neg p \wedge (p \vee q)] \to q$

$\Leftrightarrow [(\neg p \wedge p) \vee (\neg p \wedge q)] \to q$ — Distributive

$\Leftrightarrow [\text{F} \vee (\neg p \wedge q)] \to q$ — ULE

$\Leftrightarrow [\neg p \wedge q] \to q$ — Identity

$\Leftrightarrow \neg [\neg p \wedge q] \vee q$ — ULE

$\Leftrightarrow [\neg(\neg p) \vee \neg q] \vee q$ — DeMorgan

$\Leftrightarrow [p \vee \neg q] \vee q$ — Double Negation

36

L3

Tautology by proof

$[\neg p \wedge (p \vee q)] \to q$

$\Leftrightarrow [(\neg p \wedge p) \vee (\neg p \wedge q)] \to q$	Distributive
$\Leftrightarrow [\, F \vee (\neg p \wedge q)] \to q$	ULE
$\Leftrightarrow [\neg p \wedge q\,] \to q$	Identity
$\Leftrightarrow \neg\, [\neg p \wedge q\,] \vee q$	ULE
$\Leftrightarrow [\neg(\neg p) \vee \neg q\,] \vee q$	DeMorgan
$\Leftrightarrow [p \vee \neg q\,] \vee q$	Double Negation
$\Leftrightarrow p \vee [\neg q \vee q\,]$	Associative

37

L3

Tautology by proof

$[\neg p \wedge (p \vee q)] \to q$

$\Leftrightarrow [(\neg p \wedge p) \vee (\neg p \wedge q)] \to q$	Distributive
$\Leftrightarrow [\, F \vee (\neg p \wedge q)] \to q$	ULE
$\Leftrightarrow [\neg p \wedge q\,] \to q$	Identity
$\Leftrightarrow \neg\, [\neg p \wedge q\,] \vee q$	ULE
$\Leftrightarrow [\neg(\neg p) \vee \neg q\,] \vee q$	DeMorgan
$\Leftrightarrow [p \vee \neg q\,] \vee q$	Double Negation
$\Leftrightarrow p \vee [\neg q \vee q\,]$	Associative
$\Leftrightarrow p \vee [q \vee \neg q\,]$	Commutative

38

L3

Tautology by proof

$[\neg p \wedge (p \vee q)] \to q$

$\Leftrightarrow [(\neg p \wedge p) \vee (\neg p \wedge q)] \to q$	Distributive
$\Leftrightarrow [\, F \vee (\neg p \wedge q)] \to q$	ULE
$\Leftrightarrow [\neg p \wedge q\,] \to q$	Identity
$\Leftrightarrow \neg\, [\neg p \wedge q\,] \vee q$	ULE
$\Leftrightarrow [\neg(\neg p) \vee \neg q\,] \vee q$	DeMorgan
$\Leftrightarrow [p \vee \neg q\,] \vee q$	Double Negation
$\Leftrightarrow p \vee [\neg q \vee q\,]$	Associative
$\Leftrightarrow p \vee [q \vee \neg q\,]$	Commutative
$\Leftrightarrow p \vee T$	ULE

39

L3

Tautology by proof

$[\neg p \wedge (p \vee q)] \to q$

$\Leftrightarrow [(\neg p \wedge p) \vee (\neg p \wedge q)] \to q$	Distributive
$\Leftrightarrow [\, F \vee (\neg p \wedge q)] \to q$	ULE
$\Leftrightarrow [\neg p \wedge q\,] \to q$	Identity
$\Leftrightarrow \neg\, [\neg p \wedge q\,] \vee q$	ULE
$\Leftrightarrow [\neg(\neg p) \vee \neg q\,] \vee q$	DeMorgan
$\Leftrightarrow [p \vee \neg q\,] \vee q$	Double Negation
$\Leftrightarrow p \vee [\neg q \vee q\,]$	Associative
$\Leftrightarrow p \vee [q \vee \neg q\,]$	Commutative
$\Leftrightarrow p \vee T$	ULE
$\Leftrightarrow T$	Domination

40

L3

Defining Operators via Equivalences

- Using equivalences, we can define operators in terms of other operators.
  - Exclusive or: $p \oplus q \leftrightarrow (p \vee q) \wedge \neg(p \wedge q)$
    $p \oplus q \leftrightarrow (p \wedge \neg q) \vee (q \wedge \neg p)$
  - Implies: $p \to q \leftrightarrow \neg p \vee q$
  - Biconditional: $p \leftrightarrow q \leftrightarrow (p \to q) \wedge (q \to p)$
    $p \leftrightarrow q \leftrightarrow \neg(p \oplus q)$

41

Review: Propositional Logic

- Atomic propositions: p, q, r, …
- Boolean operators: $\neg\ \wedge\ \vee\ \oplus\ \to\ \leftrightarrow$
- Compound propositions: $s :\equiv (p \wedge \neg q) \vee r$
- Equivalences: $p \wedge \neg q \Leftrightarrow \neg(p \to q)$
- Proving equivalences using:
  - Truth tables.
  - Symbolic derivations. $p \Leftrightarrow q \Leftrightarrow r$ …

42

Department of Computer & Information Sciences
Pakistan Institute of Engineering and Applied Sciences

Propositional Equivalence

Predicate Logic

- Predicate logic is an extension of propositional logic.
- Propositional logic (recall) treats simple propositions (sentences) as atomic entities.
- In contrast, predicate logic distinguishes the subject of a sentence from its predicate.

Predicate Logic

- A predicate is a property or description of subjects in the universe of discourse.
  - Aslam is tall.
  - The building is structurally sound.
  - 17 is a prime number.

Predicate Logic

- In the sentence "The dog is sleeping":
  - The phrase "the dog" denotes the subject - the object or entity that the sentence is about.
  - The phrase "is sleeping" denotes the predicate- a property that is true of the subject.
- In predicate logic, a predicate is modeled as a function P(·) from objects to propositions.
  - P(x) = "x is sleeping" (where x is any object).

Predicate Logic

- Convention: Lowercase variables x, y, z... denote objects/entities; uppercase variables P, Q, R… denote propositional functions (predicates).
- Applying a predicate P to an object x is the proposition P(x). But the predicate P itself (e.g. P="is sleeping") is not a proposition (not a complete sentence).
  - e.g. if P(x) = "x is a prime number", P(3) is the proposition "3 is a prime number."

Predicate Logic

- Quantifier Expressions:
  - Quantifiers provide a notation that allows us to quantify (count) how many objects in the universe of discourse satisfy a given predicate.
  - "$\forall$" is the FOR$\forall$LL or universal quantifier. $\forall$x P(x) means for all x in the u.d., P holds.
  - "$\exists$" is the $\exists$XISTS or existential quantifier. $\exists$x P(x) means there exists an x in the u.d. (that is, 1 or more) such that P(x) is true.

Predicate Logic

- The Universal Quantifier ∀:
  - Example:
    Let the u.d. of x be parking spaces at PIEAS.
    Let P(x) be the predicate "x is full."
    Then the universal quantification of P(x), ∀x P(x), is the proposition:
    - "All parking spaces at PIEAS are full."
    - i.e., "Every parking space at PIEAS is full."
    - i.e., "For each parking space at PIEAS, that space is full."

49

Predicate Logic

- The Existential Quantifier ∃:
  - Example:
    Let the u.d. of x be parking spaces at PIEAS.
    Let P(x) be the predicate "x is full."
    Then the existential quantification of P(x), ∃x P(x), is the proposition:
    - "Some parking space at PIEAS is full."
    - "There is a parking space at PIEAS that is full."
    - "At least one parking space at PIEAS is full."

50

Predicate Logic

- If R(x,y)="x relies upon y," express the following in unambiguous English:
  - ∀x(∃y R(x,y))= Everyone has *someone* to rely on.
  - ∃y(∀x R(x,y))= There's someone whom everyone relies upon (including himself)!
  - ∃x(∀y R(x,y))= There's some needy person who relies upon everybody (including himself).
  - ∀y(∃x R(x,y))= Everyone has someone who relies upon them.
  - ∀x(∀y R(x,y))= Everyone relies upon everybody, (including themselves)!

51

Predicate Logic

- Free Variables:
  - An expression like P(x) is said to have a free variable x (meaning, x is undefined).
  - P(x,y) has 2 free variables, x and y.
  - ∀x P(x,y) has 1 free variable, and one bound variable.
  - An expression with zero free variables is a bona-fide (actual) proposition.
  - An expression with one or more free variables is still only a predicate: e.g. let Q(y) = ∀x P(x,y)

52

Predicate Logic

- Binding:
  - A variable is *bound* if it has a value or a quantifier is "acting" on it $\exists x\ Q(x, y)$
  - A quantifier (either ∀ or ∃) 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: $\exists x\ (P(x) \land Q(x)) \lor \forall y\ R(y)$
  - x is bound, y is free.

Predicate Logic

- Scope of a Quantifier:
  - The *scope* of a quantifier is the part of the statement on which it is acting.
  - Example:

$$\underbrace{\exists x\ (P(x) \land Q(x))}_{\text{scope x}} \lor \underbrace{\forall y\ R(y)}_{\text{scope y}}$$

Predicate Logic

- Negations:
  - We can also negate propositions with quantifiers.
  - Two important equivalences:

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

    $\neg\exists x\, P(x) \equiv \forall x \neg P(x)$
  - It is not the case that for all x P(x) is true = there must be an x for which P(x) is not true
  - It is not true that there exists an x for which P(x) is true = P(x) must be false for all x

Predicate Logic

- Negation of Nested Quantifiers:
  - To negate a quantifier, move negation to the right, changing quantifiers as you go.
  - Example:

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

56

Predicate Logic

- Universes of Discourse (U.D.s):
  - 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 can then say, "For any number x, P(x) is true" instead of
    - $(0+1>0) \wedge (1+1>1) \wedge (2+1>2) \wedge \ldots$

57

Predicate Logic

- Nesting of Quantifiers:
  - Example: Let the u.d. of x & y be people.
  - Let L(x,y)="x likes y"
    - A predicate with two free variables
  - Then $\exists y$ L(x,y) = "There is someone whom x likes."
    - A predicate with one free variable, x
  - Then $\forall x$ ($\exists y$ L(x,y)) = "Everyone has someone whom they like."
    - (A __________ with ___ free variables.)

58

Summary: 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.

59