

Tautologies, Contradictions, Contingencies

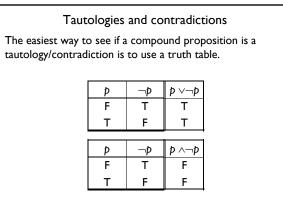
- Tautology:
 - A compound proposition is called a tautology if no matter what truth values its atomic propositions have, its own truth value is T.
 - For example: $p \lor \neg p$ (Law of excluded middle)

Tautologies, Contradictions, Contingencies

- Contradiction:
 - The opposite to a tautology, is a compound proposition that's always false -a contradiction.
 For example: p < ¬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 → ¬p



	Tautology Example									
Exa	mple	e:								
Demonstrate that is a tautology										
	[¬p ∧(p ∨q)]→q									
1.	I. Using a truth table – show that $[\neg p \land (p \lor q)] \rightarrow q$ is always true									
	Р	q	ър	p∨q	ר (p ∨q)∧ קר	[¬p ∧(p ∨q)]→q]			
	Т	Т								
	т	F								
	F	Т								
	F	F								
			_				13			

		Т	autolo	ogy by Truth ⁻	Table
Þ	q	٦þ	¢∨q	י ¢ ∧(¢ ∨q)	[¬ p ∧(p ∨q)]→q
т	т	F			
т	F	F			
F	т	т			
F	F	т			
		-			
					L3

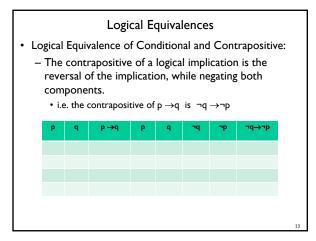
		Т	autolo	ogy by Truth ⁻	Table
Þ	q	p	þ∨q	ף ∧(p ∨q)	[¬ p ∧(p ∨q)]→q
т	т	F	т		
т	F	F	т		
F	т	т	т		
F	F	т	F		
		-			

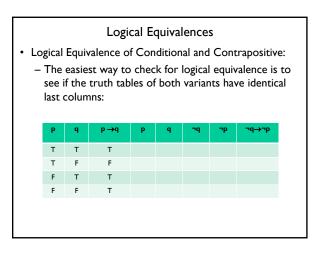
Tautology by Truth Table							
Þ	q	p	¢∨q	¬ p ∧(p ∨q)	[¬ p ∧(p ∨q)]→q		
т	т	F	Т	F			
т	F	F	т	F			
F	т	т	т	т			
F	F	т	F	F			
		-					

		Т	autolo	ogy by Truth ⁻	Table
Þ	q	p	þ∨q	ף ∧(p ∨q)	[¬p ∧(p ∨q)]→q
т	т	F	т	F	т
т	F	F	т	F	т
F	т	т	т	т	т
F	F	т	F	F	т

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.





The see i	easie	valence o st way to truth tab	f Cor chec	' Idition :k for	logical	Cont equiv	'alence is
р	q	p →d	Р	q	P۲	ъΡ	чс→р
т	т	т	т	т	F	F	
т	F	F	т	F	т	F	
F	т	т	F	т	F	т	
F	F	т	F	F	т	т	

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	p →d	р	q	рг	чΡ	ч−+рг
т	т	т	т	т	F	F	т
т	F	F	т	F	т	F	F
F	т	т	F	т	F	т	т
F	F	т	F	F	т	т	т

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:

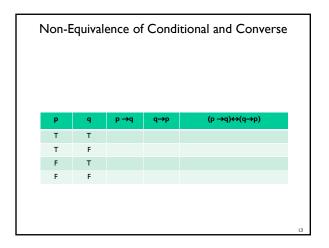
P T	q	p →d	Р	P	P۲	¬р	¬q→¬р
т	_						
	т	т	т	т	F	F	т
т	F	F	т	F	Т	F	F
F	т	т	F	Т	F	т	т
F	F	т	F	F	Т	т	т

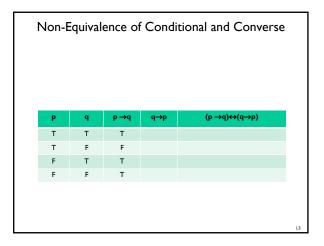
Non-Equivalence of Conditional and Converse

The converse of a logical implication is the reversal of the implication. i.e. the converse of $p \rightarrow q$ is $q \rightarrow p$.

•

 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."



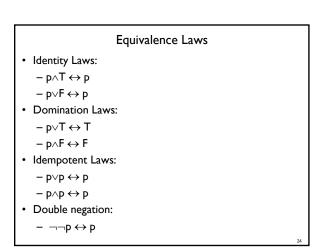


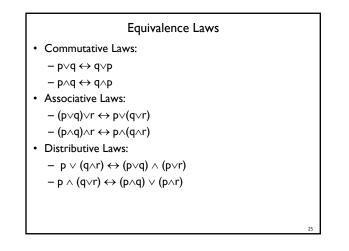
I	Non-E	quival	ence of	Condi	tional and Converse	
	Р	q	p →q	q→p	(p →q)↔(q→p)	
	т	т	т	т		
	т	F	F	т		
	F	т	т	F		
	F	F	т	т		
					21	L
						L

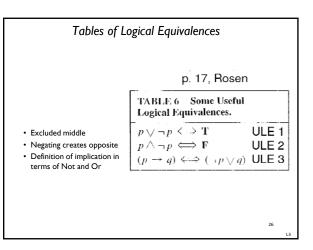
ļ	Non-E	quival	ence of	Condi	tional and Converse	
	Р	q	p →q	q→p	(p →q)↔(q→p)	
	т	т	т	т	т	
	т	F	F	т	F	
	F	т	т	F	F	
	F	F	т	т	Т	
					22	
						Ľ

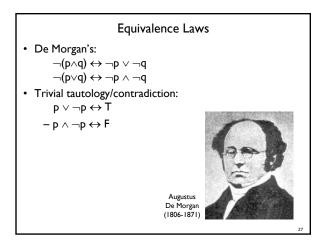
Derivational Proof Techniques

- 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 {\leftrightarrow} (p {\rightarrow} r \)) \land (\neg (s {\wedge} r) \lor \neg t) \) \rightarrow (\neg q {\rightarrow} r \)$
- 32 rows, each additional variable doubles the number of rows









DeMorgan Identities

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

Tautology example

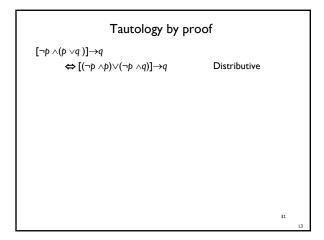
Demonstrate that is a tautology using truth table of logical equivalences

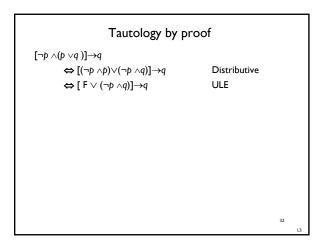
 $[\neg p \land (p \lor q)] {\rightarrow} q$

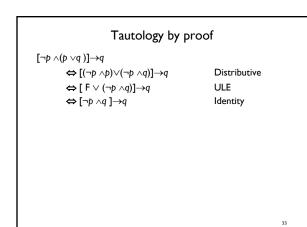
Tautology by proof

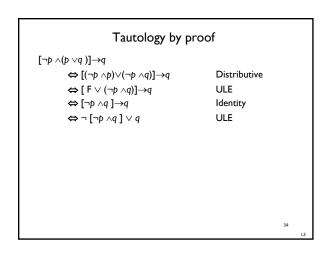
 $[\neg p \land (p \lor q)] \rightarrow q$

30

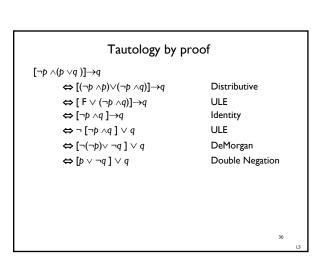








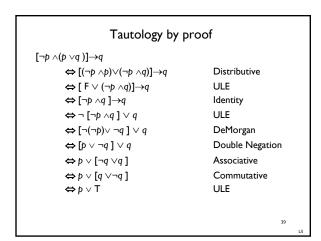
Tautology by p	proof
$[\neg p \land (p \lor q)] \rightarrow q$ $\Leftrightarrow [(\neg p \land p) \lor (\neg p \land q)] \rightarrow q$ $\Leftrightarrow [F \lor (\neg p \land q)] \rightarrow q$ $\Leftrightarrow [\neg p \land q] \rightarrow q$ $\Leftrightarrow \neg [\neg p \land q] \lor q$ $\Leftrightarrow [\neg (\neg p) \lor \neg q] \lor q$	Distributive ULE Identity ULE DeMorgan

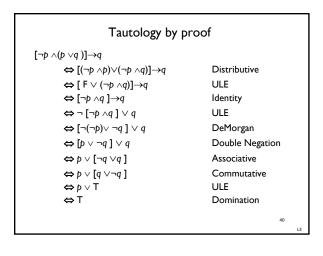


Tautology by proof $[\neg p \land (p \lor q)] \rightarrow q$ Distributive $\Leftrightarrow [(\neg p \land p) \lor (\neg p \land q)] \rightarrow q$ ULE $\Leftrightarrow [\mathsf{F} \lor (\neg p \land q)] {\rightarrow} q$ \Leftrightarrow [¬ $p \land q$] $\rightarrow q$ Identity ULE $\Leftrightarrow \neg [\neg p \land q] \lor q$ $\Leftrightarrow [\neg (\neg p) \lor \neg q \] \lor q$ DeMorgan $\Leftrightarrow [p \lor \neg q] \lor q$ Double Negation $\Leftrightarrow p \vee [\neg q \lor q]$ Associative

37

Tautology by proof $[\neg p \land (p \lor q)] \rightarrow q$ Distributive $\Leftrightarrow [(\neg p \land p) \lor (\neg p \land q)] \rightarrow q$ ULE $\Leftrightarrow [\mathsf{F} \lor (\neg p \land q)] {\rightarrow} q$ Identity \Leftrightarrow [$\neg p \land q$] $\rightarrow q$ ULE $\Leftrightarrow \neg [\neg p \land q] \lor q$ $\Leftrightarrow [\neg(\neg p) \lor \neg q] \lor q$ DeMorgan Double Negation $\Leftrightarrow [p \lor \neg q] \lor q$ Associative $\Leftrightarrow p \vee [\neg q \lor q]$ $\Leftrightarrow p \vee [q \vee \neg q]$ Commutative 38





Defining Operators via Equivalences

· Using equivalences, we can define operators in terms of other operators.

- Exclusive or:
$$p \oplus q \leftrightarrow (p \lor q) \land \neg (p \land q)$$

$$p \oplus q \leftrightarrow (p \land \neg q) \lor (q \land \neg p)$$

Implies: $p \rightarrow q \leftrightarrow \neg p \lor q$

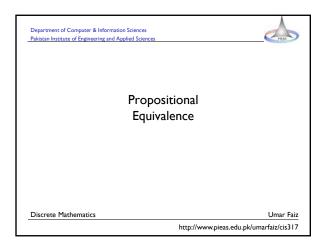
Implies:
$$p \rightarrow q \leftrightarrow -$$

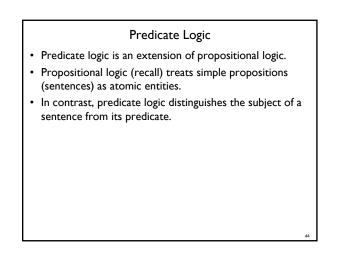
Biconditional:
$$p \leftrightarrow q \leftrightarrow (p \rightarrow q) \land (q \rightarrow p)$$

 $p \leftrightarrow q \leftrightarrow \neg (p \oplus q)$

Review: Propositional Logic

- Atomic propositions: p, q, r, ...
- Boolean operators: $\neg \land \lor \oplus \rightarrow \leftrightarrow$
- Compound propositions: $s := (p \land \neg q) \lor r$
- Equivalences: $p \land \neg q \Leftrightarrow \neg (p \rightarrow q)$
- Proving equivalences using:
 - Truth tables.
 - Symbolic derivations. $p \Leftrightarrow q \Leftrightarrow r \dots$





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 $P(\,\cdot)$ 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.
- "∃" is the ∃XISTS or existential quantifier.
 ∃x P(x) means there exists an x in the u.d. (that is, I or more) such that P(x) is true.

Predicate Logic

- The Universal Quantifier $\forall :$
 - 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), $\forall x P(x)$, 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."

Predicate Logic

- The Existential Quantifier \exists :
 - 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), $\exists x P(x)$, 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."

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)!

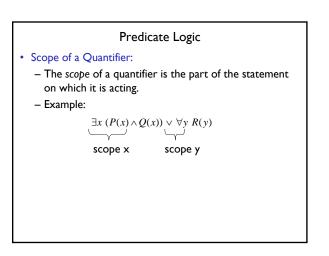
Predicate Logic

• 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.
- $\forall x P(x,y)$ has I 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 $Q(y) = \forall x P(x,y)$

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

• 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:
 - $\neg \forall x \exists y \forall z \ \mathsf{P}(x,y,z) = \exists x \ \forall y \ \exists z \ \neg \mathsf{P}(x,y,z).$

Predicate Logic

• Universes of Discourse (U.D.s):

– The collection of values that a variable \boldsymbol{x} can take is called x's universe of discourse.

- e.g., let P(x)="x+l>x".
- We can then say, "For any number x, P(x) is true" instead of • (0+1>0) \land (1+1>1) \land (2+1>2) \land ...

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 ∃y L(x,y) = "There is someone whom x likes."
 A predicate with one free variable, x
 - Then ∀x (∃y L(x,y)) =
 "Everyone has someone whom they like."
 - (A ______ with ____ free variables.)

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: [∀x P(x)] := "For all x's, P(x)."
 [∃x P(x)] := "There is an x such that P(x)."
- Universes of discourse, bound & free vars.