CS 2305: Discrete Mathematics for Computing I

Lecture 04

- KP Bhat

Key Logical Equivalences 1

Identity Laws: $p \wedge T \equiv p,\qquad p \vee F \equiv p$

Domination Laws: $p \vee T \equiv T,\qquad p \wedge F \equiv F$

Idempotent laws: $p \vee p \equiv p,\qquad p \wedge p \equiv p$

Double Negation Law: $\neg(\neg p) \equiv p$

Negation Laws: $p \vee \neg p \equiv T,\qquad p \wedge \neg p \equiv F$

Key Logical Equivalences 2

Commutative Laws: $p \vee q \equiv q \vee p$, $p \wedge q \equiv q \wedge p$

Associative Laws: $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$

$(p \vee q) \vee r \equiv p \vee (q \vee r)$

Distributive Laws: $(p \vee (q \wedge r)) \equiv (p \vee q) \wedge (p \vee r)$

$(p \wedge (q \vee r)) \equiv (p \wedge q) \vee (p \wedge r)$

Absorption Laws: $p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$

More Logical Equivalences

TABLE 7 Logical Equivalences Involving Conditional Statements.

$$p \to q \equiv \neg p \lor q$$
$$p \to q \equiv \neg q \to \neg p$$
$$p \lor q \equiv \neg p \to q$$
$$p \land q \equiv \neg(p \to \neg q)$$
$$\neg(p \to q) \equiv p \land \neg q$$
$$(p \to q) \land (p \to r) \equiv p \to (q \land r)$$
$$(p \to r) \land (q \to r) \equiv (p \lor q) \to r$$
$$(p \to q) \lor (p \to r) \equiv p \to (q \lor r)$$
$$(p \to r) \lor (q \to r) \equiv (p \land q) \to r$$

TABLE 8 Logical Equivalences Involving Biconditional Statements.

$$p \leftrightarrow q \equiv (p \to q) \land (q \to p)$$
$$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$$
$$p \leftrightarrow q \equiv (p \land q) \lor (\neg p \land \neg q)$$
$$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$$

Limitations of Truth Table Approach to Establish Logical Equivalences

- To prove the equivalence of two compound propositions in n variables, using a truth table, requires a truth table with 2^n rows
  - Infeasible beyond a handful of variables
    - for each additional propositional variable added, number of rows in the truth table doubles
  - An alternate approach is needed

2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^{10}	1,024
2^{15}	32,768
2^{20}	1,048,576
2^{25}	33,554,432
2^{30}	1,073,741,824
2^{35}	34,359,738,368
2^{40}	1,099,511,627,776
2^{45}	35,184,372,088,832
2^{50}	1,125,899,906,842,620

When there are 1000 variables, checking every one of the 2^{1000} possible combinations … cannot be done by a computer in even trillions of years.

Alternate Approach to Establish Logical Equivalences

- Simplify logical expressions using some well known results ("logical identities")
- More practical, since it can handle an arbitrarily large number of inputs
- Show that **¬(p ∨ (¬p ∧ q))** and **¬p ∧ ¬q** are logically equivalent

1. $\neg(p \lor (\neg p \land q)) \equiv \neg p \land \neg(\neg p \land q)$
2. $\equiv \neg p \land (\neg(\neg p) \lor \neg q)$
3. $\equiv \neg p \land (p \lor \neg q)$
4. $\equiv (\neg p \land p) \lor (\neg p \land \neg q)$
5. $\equiv \mathbf{F} \lor (\neg p \land \neg q)$
6. $\equiv (\neg p \land \neg q) \lor \mathbf{F}$
7. $\equiv \neg p \land \neg q$

1. Second De Morgan's Law
   - $\neg(p \lor q) \equiv \neg p \land \neg q$
2. First De Morgan's Law
   - $\neg(p \land q) \equiv \neg p \lor \neg q$
3. Double Negation Law
   - $\neg(\neg p) \equiv p$
4. Second Distributive Law
   - $p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$
5. $(\neg p \land p) \equiv \mathbf{F}$
6. Commutative Law for Disjunction
   - $p \lor q \equiv q \lor p$
7. Identity Law for Contradiction
   - $p \lor \mathbf{F} \equiv p$

Important Logical Equivalences ("Logical Identities")

TABLE 6 Logical Equivalences.

Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

Note: The *Equivalence* column of this chart by itself (i.e. without the *Name* column) will be provided to the students in the mid-term exam

Propositional Satisfiability

- A compound proposition is ***satisfiable*** if there is an assignment of truth values to its variables that make it true. When no such assignments exist, the compound proposition is ***unsatisfiable***
  - At least one row in the truth table has output value of T
- A compound proposition is unsatisfiable if and only if its negation is a tautology.

Questions on Propositional Satisfiability

Example: Determine the satisfiability of the following compound propositions:

$$\left(p \vee \neg q\right) \wedge \left(q \vee \neg r\right) \wedge \left(r \vee \neg p\right)$$

Solution: Satisfiable. Assign **T** to *p, q,* and *r*.

$$\left(p \vee q \vee r\right) \wedge \left(\neg p \vee \neg q \vee \neg r\right)$$

Solution: Satisfiable. Assign **T** to *p* and ***F*** to *q*.

$$(x \vee y) \wedge (x \vee \neg y) \wedge \neg x$$

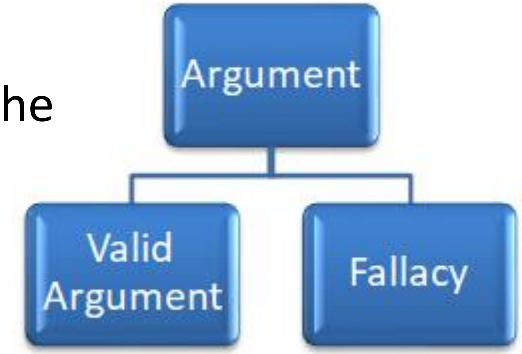
Solution: Not satisfiable. Check each possible assignment of truth values to the propositional variables and none will make the proposition true.

Applications of Satisfiability

- Many real world problems can be modeled in terms of propositional satisfiability
  - robotics, software testing, artificial intelligence planning, computer-aided design, machine vision, integrated circuit design, scheduling, computer networking, genetics etc.
- Foundational concept in theoretical computer science and advanced algorithms
- Lot of research has been done on developing reasonably efficient methods for solving satisfiability problems
  - Progressively eliminate potential solutions, given the constraints of the problem
    - Analogous to striking out entire sets of rows from an exhaustive truth table

Arguments

- An argument is a sequence of statements (premises) that end with a conclusion
- A valid argument is a conclusion that follows from the truth of the premises of the argument
  - an argument is valid if and only if it is impossible for all the premises to be true and the conclusion to be false
    - But no claim that if the conclusion is true then the premises are true as well
- An argument which is not valid is called a fallacy



Test for Argument Validity (1)

p q

P1: If John eats peanuts, he falls sick $p \rightarrow q$

P2: John did not eat peanuts $\neg p$

∴ John did not fall sick $\neg q$

Does $((p \rightarrow q) \land \neg p) \rightarrow \neg q$?

p	q	¬p	¬q	p → q	(p → q) ∧ ¬p
T	T	F	F	T	F
T	F	F	T	F	F
F	T	T	F	T	T
F	F	T	T	T	T

Premise true but conclusion false: a fallacy

Test for Argument Validity (2)

p q

P1: If John eats peanuts, he falls sick $p \rightarrow q$

P2: John did not fall sick $\neg q$

∴ John did not eat peanuts $\neg p$

Does $((p \rightarrow q) \wedge \neg q) \rightarrow \neg p$?

p	q	¬p	¬q	p → q	(p → q) ∧ ¬q
T	T	F	F	T	F
T	F	F	T	F	F
F	T	T	F	T	F
F	F	$\boxed{T}$	T	T	$\boxed{T}$

Modus Tollens

Conclusion true when premise true: valid argument

Argument Forms

Is there anything in common between these two arguments?

P1: If John eats peanuts, he falls sick — $\underbrace{\text{John eats peanuts}}_{p}$, $\underbrace{\text{he falls sick}}_{q}$

P2: John did not fall sick ¬q

∴ John did not eat peanuts ¬p

$((p \to q) \land \neg q) \to \neg p$

P1: If you work hard, you will get a good raise — $\underbrace{\text{you work hard}}_{p}$, $\underbrace{\text{you will get a good raise}}_{q}$

P2: You did not get a good raise ¬q

∴ You did not work hard ¬p

$((p \to q) \land \neg q) \to \neg p$

Although the two arguments are different, they have the same **argument form**

- same symbolic representation
- Belong to the same "family" of arguments

Argument vs Argument Form

- **Argument**
  - sequence of propositions
  - valid if the truth of all its premises implies that the conclusion is true
  - a specific instance of an argument form
- **Argument Form**
  - sequence of compound propositions involving propositional variables
  - valid if no matter which particular propositions are substituted for the propositional variables in its premises, the conclusion is true if the premises are all true
  - represents an entire family of arguments

Establishing Validity of Argument Forms

- Brute force approach
    - truth table
  - Major issue
    - Number of rows in the truth table increases exponentially with the number of propositional variables
- Rules of inference based approach
  - first establish the validity of some relatively simple argument forms (“rules of inference”)
  - use rules of inference as building blocks to validate more complicated valid argument forms