COSC 221 - Introduction to Discrete Structures

Lecture - Logic-03

Readings

- Propositional Logic: Sections 3.1, 3.2, 3.3
- Predicate Logic: Sections 3.4
- Computer Science Connections
  1. Computational Complexity (Section 3.3)
  2. Modern Compilers (Section 3.3)
  3. Game Trees (Section 3.4)

- Logic Equivalence
- Logic Laws
- Proving Equivalence

Propositional Logic

Logic Equivalence and Logic Laws (Sec 3.3.2)

- Logic Equivalence
- Logic Laws
- Proving Equivalence

$\varphi \equiv \psi$ if their truth values are the same under every **truth assignment**.

- Logic Equivalence
- Logic Laws
- Proving Equivalence

$\varphi \equiv \psi$ if their truth values are the same under every **truth assignment**.

Greek letters (e.g. phi and psi) for compound propositions

Logic Equivalence and Logic Laws (Sec 3.3.2)

- Logic Equivalence
- Logic Laws
- Proving Equivalence

$\varphi \equiv \psi$ if their truth values are the same under every **truth assignment**.

Greek letters (e.g. phi and psi) for compound propositions

truth values - one for each Boolean variable

- ▷ Logic Equivalence
- ▷ Logic Laws
- ▷ Proving Equivalence

$\varphi \equiv \psi$ if their truth values are the same under every **truth assignment**.

For example, we learned previously that

$$(p \Rightarrow q) \equiv (\neg q \Rightarrow \neg p)$$

(by examining their truth tables).

Truth Table

p	q	$\neg q$	$\neg p$	φ	ψ
T	T	F	F	T	T
T	F	T	F	F	F
F	T	F	T	T	T
F	F	T	T	T	T

- Logic Equivalence
- Logic Laws
- Proving Equivalence

Proven and useful logic equivalences

- Logic Equivalence
- Logic Laws
- Proving Equivalence

Proven and useful logic equivalences

De Morgan's Laws

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

- Logic Equivalence
- Logic Laws
- Proving Equivalence

Proven and useful logic equivalences

De Morgan's Laws

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

Distributive Laws

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

Associative Laws

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

More on Page 3-28 in Textbook

- Truth-Table Method
- Logic-Law Method

De Morgan's Laws

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

- Truth-Table Method
- Logic-Law Method

De Morgan's Laws

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

- Truth-Table Method
- Logic-Law Method

p	q	$\neg(p \wedge q)$	$\neg p \vee \neg q$
T	T	F	F
T	F	T	T
F	T	T	T
F	F	T	T

De Morgan's Laws

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

- Truth-Table Method
- Logic-Law Method

Equivalences Involving Implications

1) $p \Rightarrow q \equiv \neg q \Rightarrow \neg p$
2) $p \Rightarrow q \equiv \neg p \vee q$
3) $\neg(p \Rightarrow q) \equiv p \wedge \neg q$

- Truth-Table Method
- Logic-Law Method

p	q	$p \Rightarrow q$	$\neg p \vee q$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

Equivalences Involving Implications

1) $p \Rightarrow q \equiv \neg q \Rightarrow \neg p$
2) $p \Rightarrow q \equiv \neg p \vee q$
3) $\neg(p \Rightarrow q) \equiv p \wedge \neg q$

- Truth-Table Method
- Logic-Law Method

Using Known Equivalences

$$\begin{aligned}\neg(p \Rightarrow q) &\equiv \neg(\neg p \vee q)\\ &\equiv \neg(\neg p) \wedge (\neg q)\\ &\equiv p \wedge \neg q\end{aligned}$$

Equivalences Involving Implications

1) $p \Rightarrow q \equiv \neg q \Rightarrow \neg p$
2) $p \Rightarrow q \equiv \neg p \vee q$
3) $\neg(p \Rightarrow q) \equiv p \wedge \neg q$

- Truth-Table Method
- Logic-Law Method

Q.1) "If n^2 is even, then n is even". Its negation is

A) If n^2 is not even, then n is not even

B) If n is even, then n^2 is even

C) If n is not even, then n^2 is not even

D) n^2 is even, but n is not even

Equivalences Involving Implications

1) $p \Rightarrow q \equiv \neg q \Rightarrow \neg p$
2) $p \Rightarrow q \equiv \neg p \vee q$
3) $\neg(p \Rightarrow q) \equiv p \wedge \neg q$

- Truth-Table Method
- Logic-Law Method

De Morgan (1806 - 1871), a British logician

De Morgan's Laws (three-variable version)

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

- Truth-Table Method
- Logic-Law Method

$$
\begin{aligned}
\neg(p \vee q \vee r) &\equiv \neg(p \vee (q \vee r)) && \text{(by Associative Law)} \\
&\equiv \neg p \wedge \neg(q \vee r) && \text{(by De Morgan's Law)} \\
&\equiv \neg p \wedge (\neg q \wedge \neg r) \\
&\equiv \neg p \wedge \neg q \wedge \neg r
\end{aligned}
$$

De Morgan's Laws (three-variable version)

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

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

- Truth-Table Method
- Logic-Law Method

$$
\begin{aligned}
\neg(p \lor q \lor r) &\equiv \neg(p \lor (q \lor r)) && \text{(by Associative Law)} \\
&\equiv \neg p \land \neg(q \lor r) && \text{(by De Morgan's Law)} \\
&\equiv \neg p \land (\neg q \land \neg r) \\
&\equiv \neg p \land \neg q \land \neg r
\end{aligned}
$$

$$
\begin{aligned}
(p \land q) \land r &\equiv p \land (q \land r) \\
(p \lor q) \lor r &\equiv p \lor (q \lor r)
\end{aligned}
$$

De Morgan's Laws (three-variable version)

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

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

- Truth-Table Method
- Logic-Law Method

Q.2) The truth value of $(p \wedge q) \vee \neg p \vee \neg q$

A) depends on the truth value of p and q

B) is always T

C) is always F

D) can be either T or F

De Morgan's Laws (three-variable version)

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

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

- Truth-Table Method
- Logic-Law Method

Q.3) Logic Equivalence in Programming

```
while( (x > 10) || (x < 3) || (x is even) ) {
        x = random.nextInt(50) - 10;
}
```

The above while-loop terminates if x is

A) 2 B) 6

C) 9 D) 15

De Morgan's Laws (three-variable version)

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

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

More Exercises

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

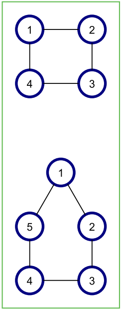
$$\begin{aligned}
p \vee q \rightarrow r &\equiv \neg(p \vee q) \vee r && \text{by definition} \\
&\equiv (\neg p \wedge \neg q) \vee r && \text{De Morgan's laws} \\
&\equiv (\neg p \vee r) \wedge (\neg q \vee r) && \text{Distributive laws} \\
&\equiv (p \rightarrow r) \wedge (q \rightarrow r) && \text{definition}
\end{aligned}$$

Associative Laws

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

Because of the associative laws, the conjunction and disjunctions of more than two statements p, q, r can be be written as $p \wedge q \wedge r$ without causing any ambiguity. For propositions that contain different connectives, the order of operations is important.

See page 310 in the textbook for a detailed discussion on the conventions regarding the precedence of the connectives.



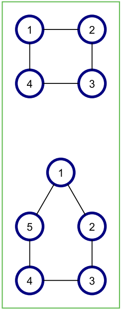
Modelling with Propositional Logic: Graph Coloring as an Example

- ▷ Two colors to use
- ▷ Adjacent nodes colored differently

A graph is a pair $G = (V, E)$, where

V - set of vertices (objects)

E - set of edges (relationships)



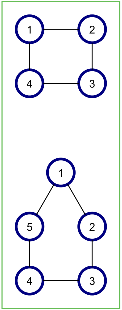
Modelling with Propositional Logic: Graph Coloring as an Example

- Two colors to use
- Adjacent nodes colored differently

A graph is a pair $G = (V, E)$, where

V - set of vertices (objects)

E - set of edges (relationships)



Modelling with Propositional Logic: Graph Coloring as an Example

- ▷ Two colors to use
- ▷ Adjacent nodes colored differently

Variables:

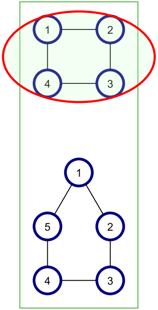
- ▷ Two per node: E.g. x_1^r and x_1^b
- ▷ x_1^r: Node 1 gets "Red"

Coloring proposition in CNF

$x_1^r \vee x_1^b$

$$(x_1^r \vee x_1^b) \wedge (x_2^r \vee x_2^b) \wedge (x_3^r \vee x_3^b) \wedge (x_4^r \vee x_4^b)$$
$$\wedge (\neg x_1^r \vee \neg x_2^r) \wedge (\neg x_1^b \vee \neg x_2^b)$$
$$\wedge (\neg x_2^r \vee \neg x_3^r) \wedge (\neg x_2^b \vee \neg x_3^b)$$
$$\wedge (\neg x_3^r \vee \neg x_4^r) \wedge (\neg x_3^b \vee \neg x_4^b)$$
$$\wedge (\neg x_4^r \vee \neg x_1^r) \wedge (\neg x_4^b \vee \neg x_1^b)$$

True iff 1 is colored



Modelling with Propositional Logic: Graph Coloring as an Example

- Two colors to use
- Adjacent nodes colored differently

Variables:

- Two per node: E.g. x_1^r and x_1^b
- x_1^r: Node 1 gets "Red"

Coloring proposition in CNF

$$(x_1^r \vee x_1^b) \wedge (x_2^r \vee x_2^b) \wedge (x_3^r \vee x_3^b) \wedge (x_4^r \vee x_4^b)$$

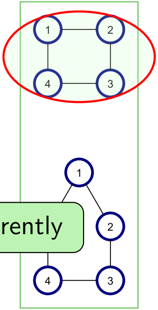
$$(\neg x_1^r \vee \neg x_2^r) \wedge (\neg x_1^b \vee \neg x_2^b)$$

$$\wedge\ (\neg x_2^r \vee \neg x_3^r) \wedge (\neg x_2^b \vee \neg x_3^b)$$

$$\wedge\ (\neg x_3^r \vee \neg x_4^r) \wedge (\neg x_3^b \vee \neg x_4^b)$$

$$\wedge\ (\neg x_4^r \vee \neg x_1^r) \wedge (\neg x_4^b \vee \neg x_1^b)$$

True iff 1 and 2 colored differently

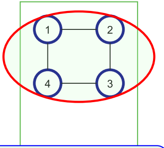


Modelling with Propositional Logic: Graph Coloring as an Example

- ▷ Two colors to use
- ▷ Adjacent nodes colored differently

Variables:

- ▷ Two per node: E.g. x_1^r and x_1^b
- ▷ x_1^r: Node 1 gets "Red"



Coloring proposition in CNF

$$(x_1^r \lor x_1^b) \land (x_2^r \lor x_2^b) \land (x_3^r \lor x_3^b) \land (x_4^r \lor x_4^b)$$
$$\land (\neg x_1^r \lor \neg x_2^r) \land (\neg x_1^b \lor \neg x_2^b)$$
$$\land (\neg x_2^r \lor \neg x_3^r) \land (\neg x_2^b \lor \neg x_3^b)$$
$$\land (\neg x_3^r \lor \neg x_4^r) \land (\neg x_3^b \lor \neg x_4^b)$$
$$\land (\neg x_4^r \lor \neg x_1^r) \land (\neg x_4^b \lor \neg x_1^b)$$

Truth Assignments to Coloring

- Satisfying Assignment
  $x_1^r = T, x_2^b = T, x_3^r = T, x_4^b = T$
- Proper Coloring
  Red, Blue, Red, Blue

Modelling with Propositional Logic: Graph Coloring as an Example

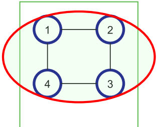
- Two colors to use
- Adjacent nodes colored differently

Variables:

- Two per node: E.g. x_1^r and x_1^b
- x_1^r: Node 1 gets "Red"

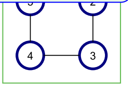
Coloring proposition in CNF

$$(x_1^r \vee x_1^b) \wedge (x_2^r \vee x_2^b) \wedge (x_3^r \vee x_3^b) \wedge (x_4^r \vee x_4^b)$$
$$\wedge (\neg x_1^r \vee \neg x_2^r) \wedge (\neg x_1^b \vee \neg x_2^b)$$
$$\wedge (\neg x_2^r \vee \neg x_3^r) \wedge (\neg x_2^b \vee \neg x_3^b)$$
$$\wedge (\neg x_3^r \vee \neg x_4^r) \wedge (\neg x_3^b \vee \neg x_4^b)$$
$$\wedge (\neg x_4^r \vee \neg x_1^r) \wedge (\neg x_4^b \vee \neg x_1^b)$$



Satisfiable CNF Formula

CNF: Conjunctive Normal Form



Modelling with Propositional Logic: Graph Coloring as an Example

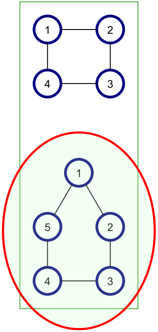
- Two colors to use
- Adjacent nodes colored differently

Variables:

- Two per node: E.g. x_1^r and x_1^b
- x_1^r: Node 1 gets "Red"

Coloring proposition in CNF

$$
\begin{aligned}
\varphi = {} & (x_1^r \vee x_1^b) \wedge (x_2^r \vee x_2^b) \wedge (x_3^r \vee x_3^b) \wedge (x_4^r \vee x_4^b) \wedge (x_5^r \vee x_5^b) \\
& \wedge (\neg x_1^r \vee \neg x_2^r) \wedge (\neg x_1^b \vee \neg x_2^b) \\
& \wedge (\neg x_2^r \vee \neg x_3^r) \wedge (\neg x_2^b \vee \neg x_3^b) \\
& \wedge (\neg x_3^r \vee \neg x_4^r) \wedge (\neg x_3^b \vee \neg x_4^b) \\
& \wedge (\neg x_4^r \vee \neg x_5^r) \wedge (\neg x_4^b \vee \neg x_5^b) \\
& \wedge (\neg x_5^r \vee \neg x_1^r) \wedge (\neg x_5^b \vee \neg x_1^b)
\end{aligned}
$$



Modelling with Propositional Logic: Graph Coloring as an Example

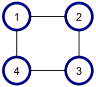
- ▷ Two colors to use
- ▷ Adjacent nodes colored differently

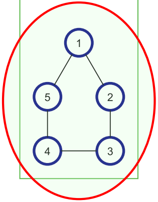
Variables:

- ▷ Two per node: E.g. x_1^r and x_1^b
- ▷ x_1^r: Node 1 gets "Red"

Coloring proposition in CNF

$$
\begin{aligned}
\varphi = {} & (x_1^r \vee x_1^b) \wedge (x_2^r \vee x_2^b) \wedge (x_3^r \vee x_3^b) \wedge (x_4^r \vee x_4^b) \wedge (x_5^r \vee x_5^b) \\
& \wedge (\neg x_1^r \vee \neg x_2^r) \wedge (\neg x_1^b \vee \neg x_2^b) \\
& \wedge (\neg x_2^r \vee \neg x_3^r) \wedge (\neg x_2^b \vee \neg x_3^b) \\
& \wedge (\neg x_3^r \vee \neg x_4^r) \wedge (\neg x_3^b \vee \neg x_4^b) \\
& \wedge (\neg x_4^r \vee \neg x_5^r) \wedge (\neg x_4^b \vee \neg x_5^b) \\
& \wedge (\neg x_5^r \vee \neg x_1^r) \wedge (\neg x_5^b \vee \neg x_1^b)
\end{aligned}
$$





Modelling with Propositional Logic: C

- ▷ Two colors to use
- ▷ Adjacent nodes colored differentl

Contradiction and Tautology

Unsatisfiable Formula: a **contradiction**

$\neg\varphi$: a **tautology**

Additional Notes

- **Tautology**: A proposition that is always true
- **Contradiction**: A proposition that is always false
- **Satisfiable Propositions**: there exist satisfying truth assignments

Algorithmic Problem: Satisfiability PROBLEM

INSTANCE: A logic statement F.

QUESTION: Is F satisfiable?

E.g., $F = (x_1 \vee x_2) \wedge\ (x_8 \vee \neg x_{15}) \cdots\ (\neg x_{99} \vee x_{100})$

Additional Notes

Brackets and Precedence of Connectives

- $\neg$ (highest), $\cdots$, $\Rightarrow$ (lowest)
- Breaking ties: left-to-right rule
- When uncertain, always use parentheses
- Section 3.2.4

$$(p \Rightarrow q) \wedge \neg q \Rightarrow \neg p$$

References I