Tutorial 1 : SAT AND BDD

CS60030 Formal Systems

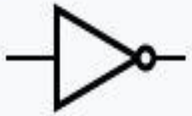
PALLAB DASGUPTA,
FNAE, FASc,
A K Singh Distinguished Professor in AI,
Dept of Computer Science & Engineering
Indian Institute of Technology Kharagpur
Email: pallab@cse.iitkgp.ac.in
Web: http://cse.iitkgp.ac.in/~pallab

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

FORMAL METHODS FOR SAFETY CRITICAL SYSTEMS

Circuit to CNF Representation

Characteristic function

Type	Operation	CNF Sub-expression
 AND	$C = A \cdot B$	$(\overline{A} \vee \overline{B} \vee C) \wedge (A \vee \overline{C}) \wedge (B \vee \overline{C})$
 NAND	$C = \overline{A \cdot B}$	$(\overline{A} \vee \overline{B} \vee \overline{C}) \wedge (A \vee C) \wedge (B \vee C)$
 OR	$C = A + B$	$(A \vee B \vee \overline{C}) \wedge (\overline{A} \vee C) \wedge (\overline{B} \vee C)$
 NOR	$C = \overline{A + B}$	$(A \vee B \vee C) \wedge (\overline{A} \vee \overline{C}) \wedge (\overline{B} \vee \overline{C})$
 NOT	$C = \overline{A}$	$(\overline{A} \vee \overline{C}) \wedge (A \vee C)$
 XOR	$C = A \oplus B$	$(\overline{A} \vee \overline{B} \vee \overline{C}) \wedge (A \vee B \vee \overline{C}) \wedge (A \vee \overline{B} \vee C) \wedge (\overline{A} \vee B \vee C)$

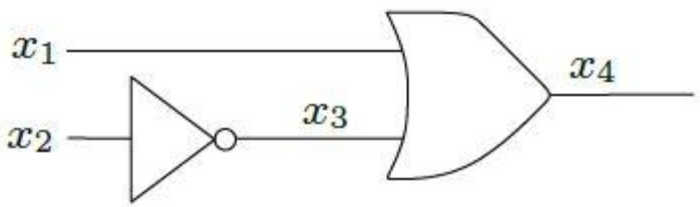
<u>AND Gate to CNF:</u>

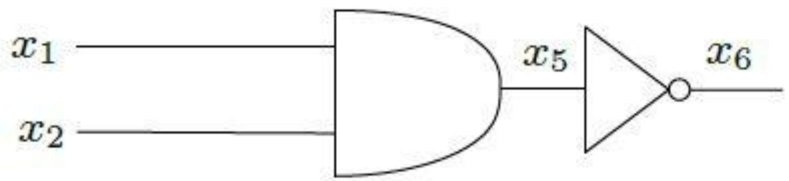
$C \Leftrightarrow A \wedge B$

- ≡ $(C \Rightarrow A \wedge B) \wedge (A \wedge B \Rightarrow C)$
- ≡ $(\neg C \vee (A \wedge B)) \wedge (\neg(A \wedge B) \vee C)$
- ≡ $(\neg C \vee A) \wedge (\neg C \vee B) \wedge (\neg A \vee \neg B \vee C)$

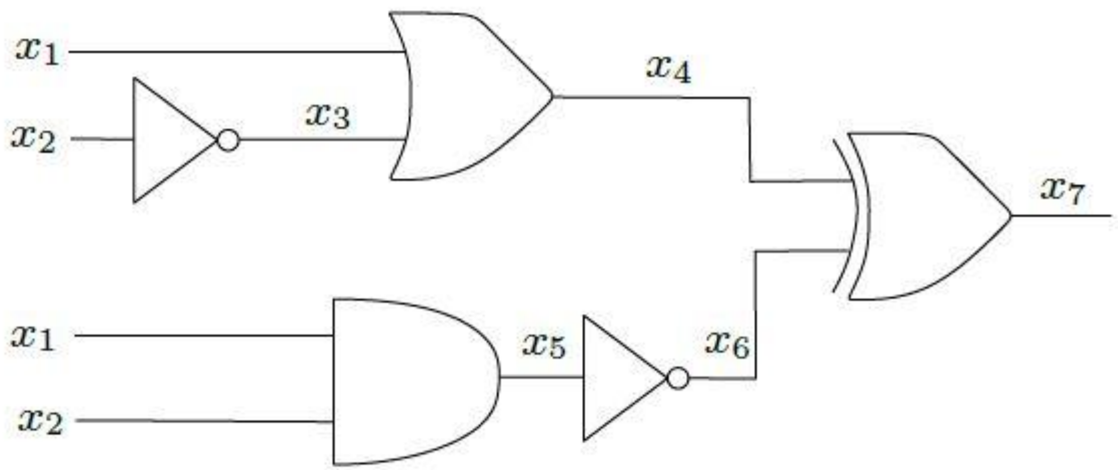
1. Equivalence Checking of Two Circuits Using SAT

- Transform the circuits into CNF.



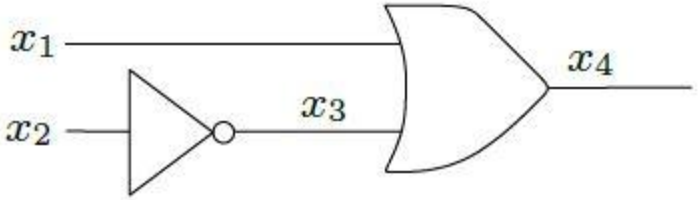


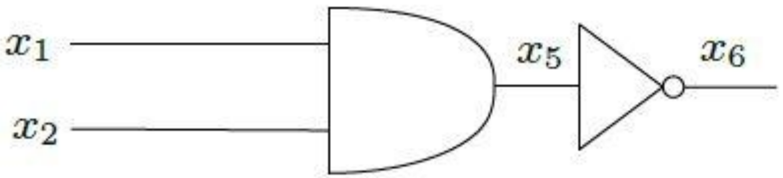
- Are these two circuits equivalent?



x_4	x_6	x_7
0	0	0
0	1	1
1	0	1
1	1	0

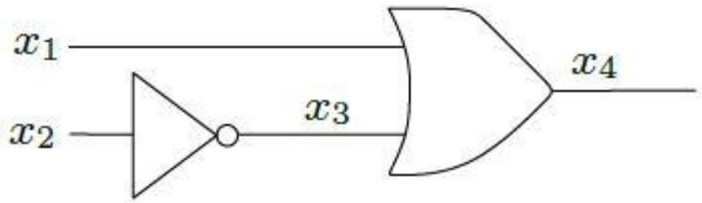
Convert the Circuits to CNF

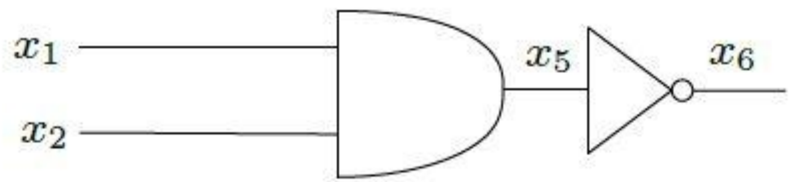




Type	Operation	CNF Sub-expression
AND	$C = A \cdot B$	$(\overline{A} \vee \overline{B} \vee C) \wedge (A \vee \overline{C}) \wedge (B \vee \overline{C})$
NAND	$C = \overline{A \cdot B}$	$(\overline{A} \vee \overline{B} \vee \overline{C}) \wedge (A \vee C) \wedge (B \vee C)$
OR	$C = A + B$	$(A \vee B \vee \overline{C}) \wedge (\overline{A} \vee C) \wedge (\overline{B} \vee C)$
NOR	$C = \overline{A + B}$	$(A \vee B \vee C) \wedge (\overline{A} \vee \overline{C}) \wedge (\overline{B} \vee \overline{C})$
NOT	$C = \overline{A}$	$(\overline{A} \vee \overline{C}) \wedge (A \vee C)$
XOR	$C = A \oplus B$	$(\overline{A} \vee \overline{B} \vee \overline{C}) \wedge (A \vee B \vee \overline{C}) \wedge (A \vee \overline{B} \vee C) \wedge (\overline{A} \vee B \vee C)$

CNF Formulations of the Circuits





1. $X_3 \Leftrightarrow \neg X_2$

   $(X_2 \vee X_3) \wedge (\neg X_2 \vee \neg X_3)$

2. $X_4 \Leftrightarrow X_1 \vee X_3$

   $(X_1 \vee X_3 \vee \neg X_4) \wedge (\neg X_1 \vee X_4) \wedge (\neg X_3 \vee X_4)$

3. $X_5 \Leftrightarrow X_1 \wedge X_2$

   $(\neg X_1 \vee \neg X_2 \vee X_5) \wedge (X_1 \vee \neg X_5) \wedge (X_2 \vee \neg X_4)$

4. $X_6 \Leftrightarrow \neg X_5$

   $(X_6 \vee X_5) \wedge (\neg X_6 \vee \neg X_5)$

5. $X_4 \oplus X_6$

   $(X_4 \vee X_6) \wedge (\neg X_4 \vee \neg X_6)$

2. Solving a Sudoku Puzzle Using SAT

- **Sudoku is combinatorial puzzle where a DXD board has the following constraints:**

1. **Each cell has an unique assignment of number from 0 to D-1**
2. **No number in a row is repeated**
3. **No number in a column is repeated and (No number in a block is repeated)**

- **Model the following 2x2- Sudoku (D=2) as a SAT problem. Mention the variables and the clauses and find a satisfiable assignment. Can you similarly solve the 4X4 Sudoku (D=4) Puzzle?**

Col

Row

	0	1
0	0	-
1	-	-

2 X 2

4 X 4

2	4	.	.
.	.	.	.
.	.	2	3
.	.	.	.

Solving a Sudoku Puzzle Using SAT

2 X 2

	0	1
0	0	-
1	-	-

Variables

X_{000}
X_{001}
X_{010}
X_{011}
X_{100}
X_{101}
X_{110}
X_{111}

Clauses

1) All cells must have assignment 0 or 1

$$(X_{000} \vee X_{001}) \wedge (X_{010} \vee X_{011}) \wedge (X_{100} \vee X_{101}) \wedge (X_{110} \vee X_{111})$$

2) A cell can have at most one assignment 0 or 1

$$(\neg X_{000} \vee \neg X_{001}) \wedge (\neg X_{010} \vee \neg X_{011}) \wedge (\neg X_{100} \vee \neg X_{101}) \wedge (\neg X_{110} \vee \neg X_{111})$$

3) No numbers on rows are repeated

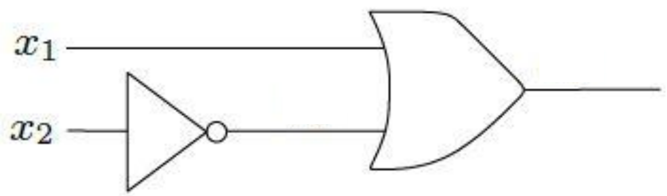
$$(\neg X_{000} \vee \neg X_{010}) \wedge (\neg X_{001} \vee \neg X_{011}) \wedge (\neg X_{100} \vee \neg X_{110}) \wedge (\neg X_{101} \vee \neg X_{111})$$

4) No numbers on columns are repeated

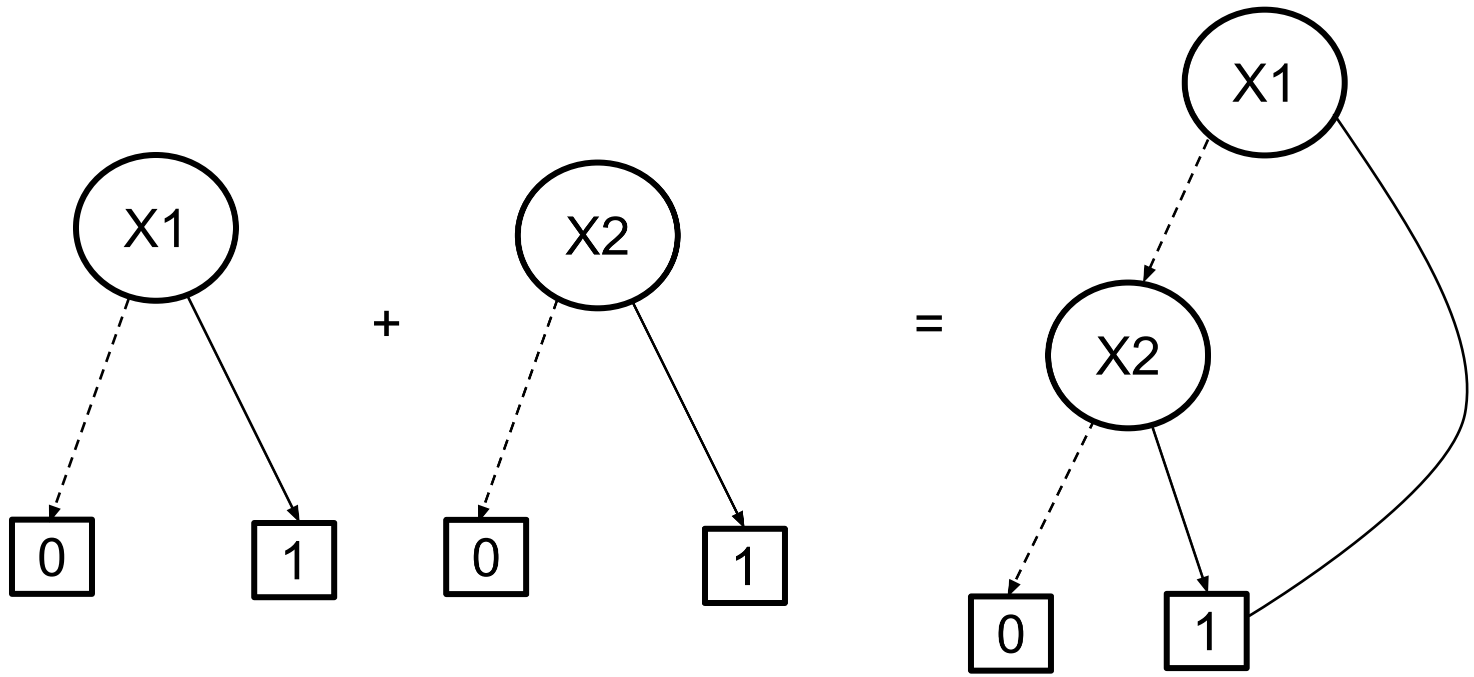
$$(\neg X_{000} \vee \neg X_{100}) \wedge (\neg X_{001} \vee \neg X_{101}) \wedge (\neg X_{010} \vee \neg X_{110}) \wedge (\neg X_{011} \vee$$

Gate Level Circuit to BDDs

- **Each input of the circuit is a BDD.**
- **Each gate becomes an operator that produces a new BDD.**
- **Example:**



$f = x_1 + \neg x_2$

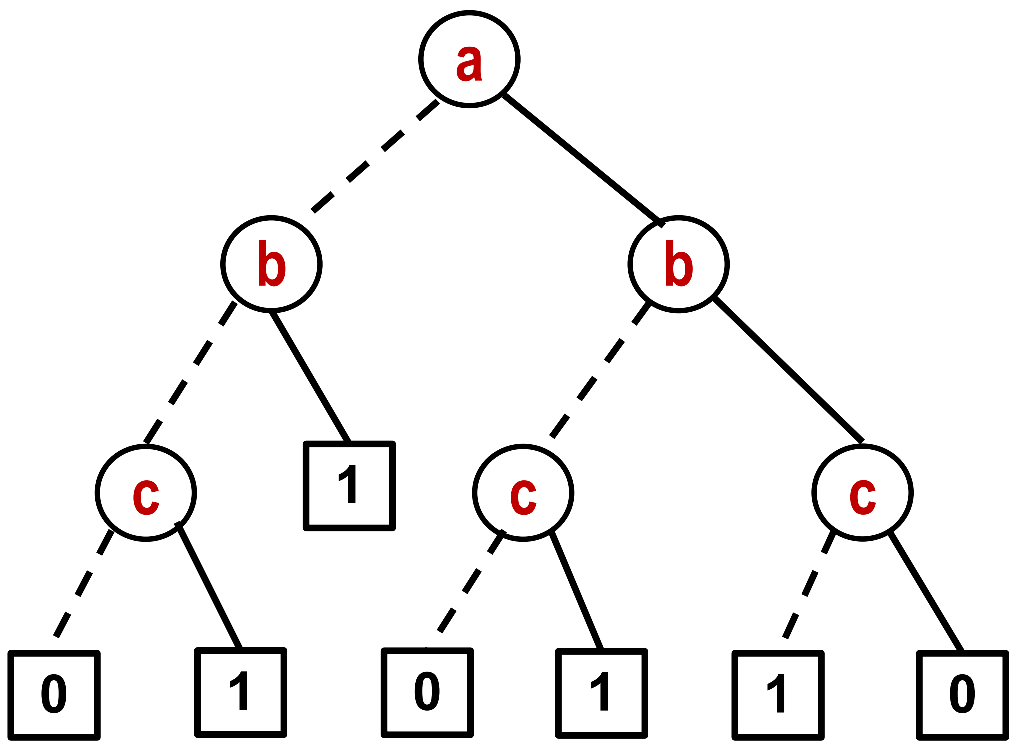


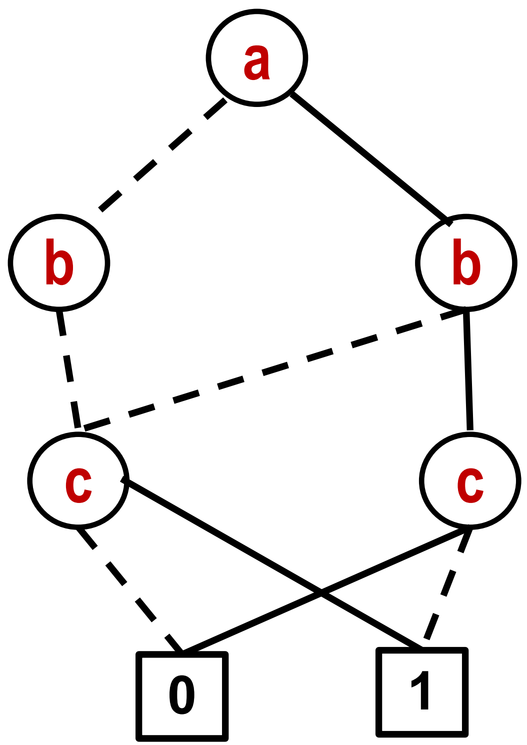
BDD for f

ROBDD - 1

Draw ROBDD for the following function

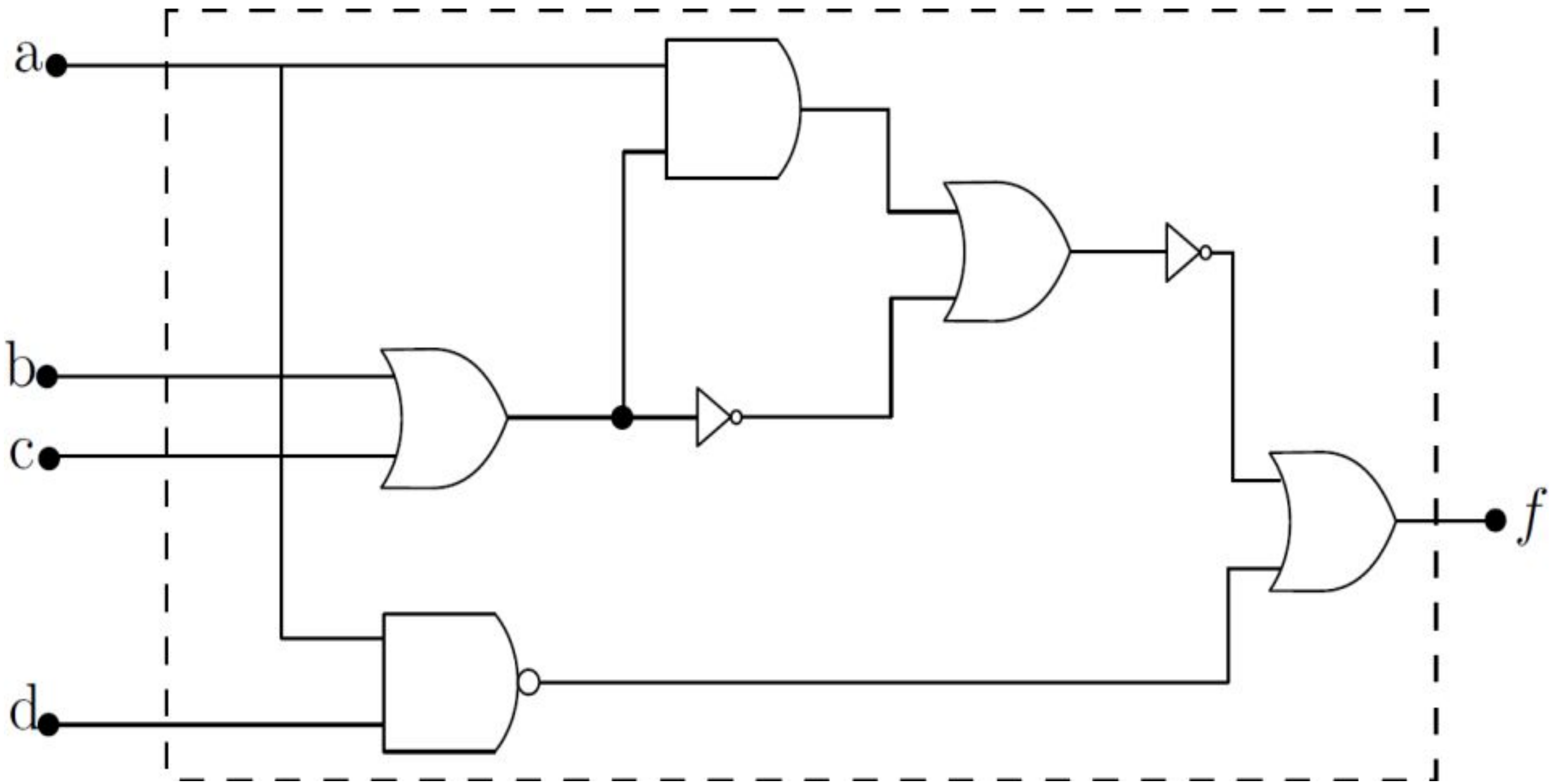
f(a, b, c) = $\overline{a}b + \overline{b}c + b\overline{c}$





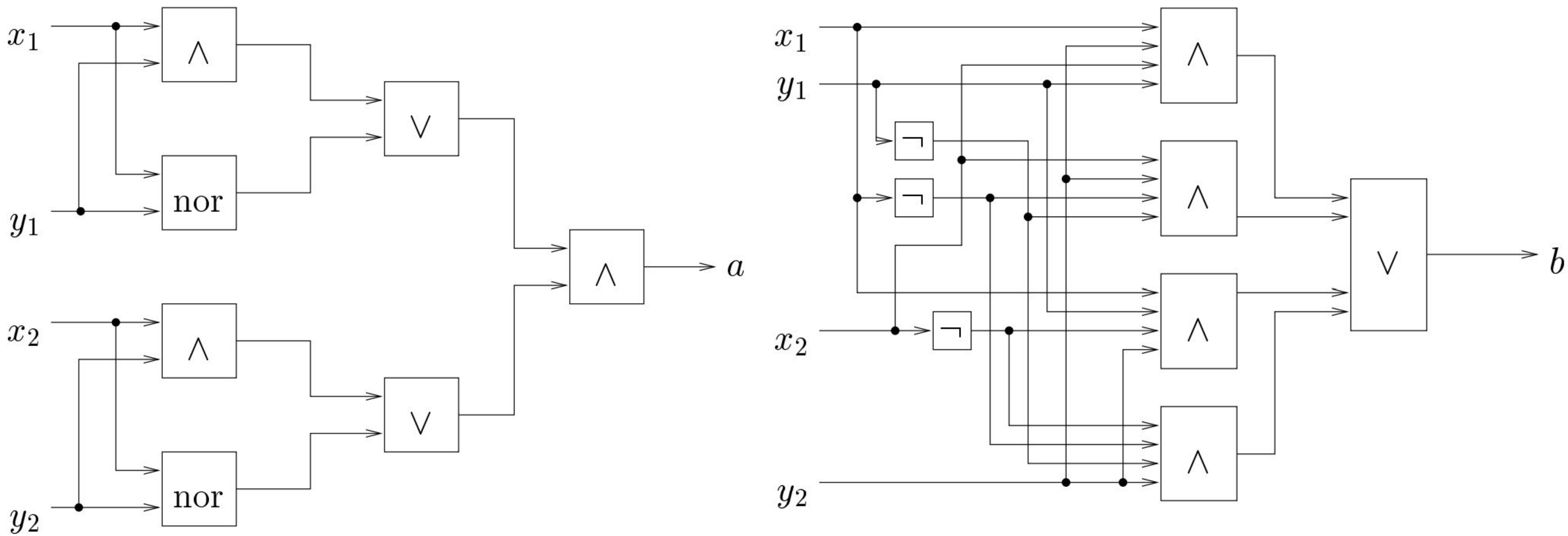
ROBDD – 2

Draw the ROBDD for *f* using the ordering a>b>c>d, for the circuit given below.



ROBDD - 3

Verify if the two circuits are equivalent or not using ROBDD



Graph Colouring to SAT Formulation

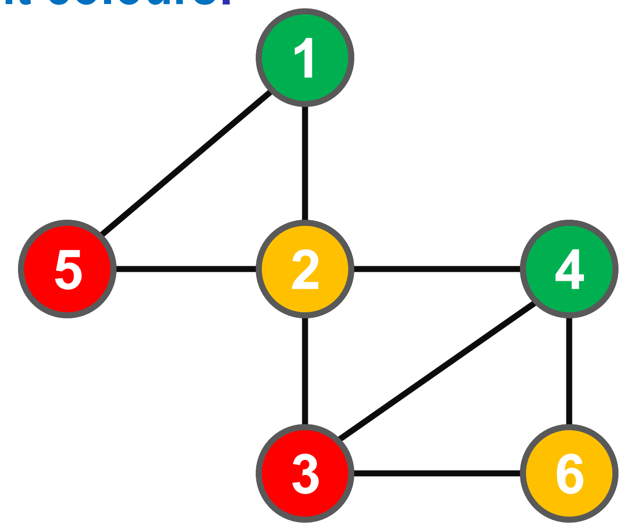
We are given a graph G = (V,E)

A colouring of the *n* vertices of the graph with *k* colours is a mapping; f: V -- > {1, .., *k*}

- f(*v*) denotes the color of vertex *v*

A coloring is a *proper colouring*, if, adjacent vertices must receive different colours.

Write a SAT formulation for *proper colouring*.



Solution

1. Each vertex must be coloured
2. Each vertex should have only one colour
3. Neighboring vertices should not have same colour

Practice Problems : Graph Coloring : SAT Formulation

We are given a graph G = (V,E)

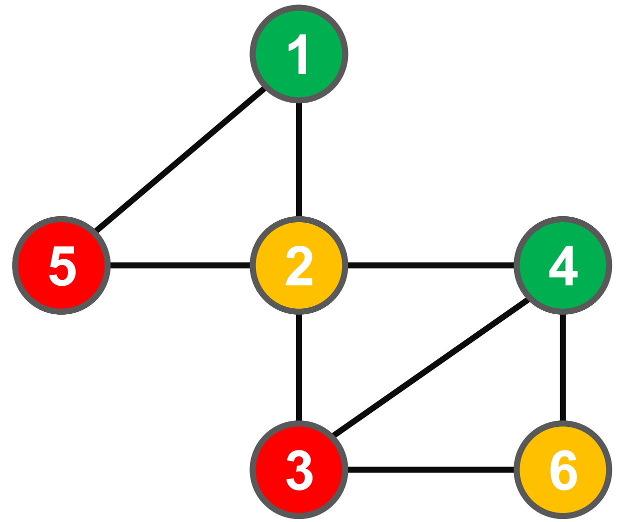
A coloring of the *n* vertices of the graph with *k* colors is a map; f: V→{1, .., *k*}

- f(*v*) denotes the color of vertex *v*

A coloring is a *proper coloring*, if, adjacent vertices must receive different colors.

PROBLEM

- To find the minimum *k* such that a proper *k*-coloring of G is possible

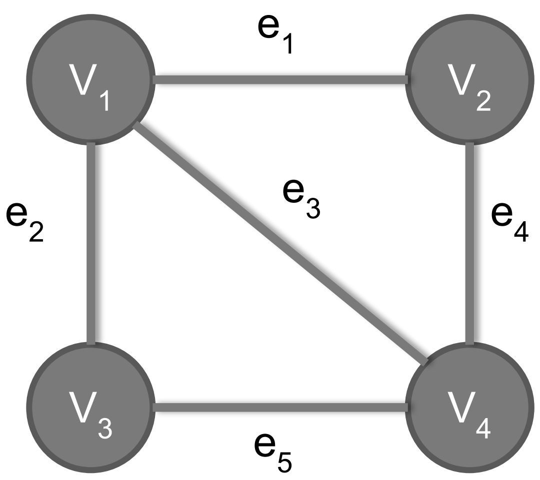


In how many ways can we color the *n* vertices with *k* colors?

Each vertex may receive one of the *k* colors

Number of colorings (not necessarily proper colorings) = k^n

Graph Colouring



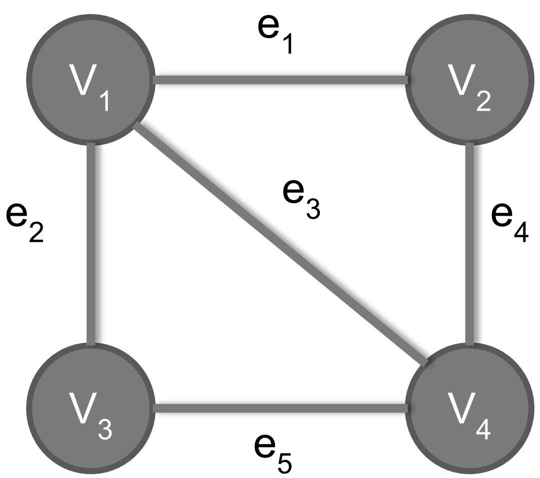
Types of Constraints:

1. Vertex Constraints: A vertex must get exactly one color.
2. Edge Constraints: No two *adjacent* vertices should be colored with the same color

Boolean State Encoding:

- Each color is given a number "i" – assume N colors
- Each vertex is given a number "j"
- For "k" colors, each vertex has "i" Boolean variables. Vertex "j" has variables numbered as [(j-1)*N + i]: For N = 3 colors, Vertex V_3 is represented as the three Boolean variables x_7, x_8 and x_9 respectively representing that the vertex V_3 is colored by colors "1", "2" or "3".

Graph Colouring



Vertex Constraints:

For Vertex V_1:

Assign it a color : $(x_1 \lor x_2 \lor x_3)$

Exactly one color : $(\neg x_1 \lor \neg x_2) \land (\neg x_1 \lor \neg x_3)\ (\neg x_2 \lor \neg x_3)$

Edge Constraints:

For Vertex V_1: edge e_1

Color 1: $(\neg x_1 \lor \neg x_4)$

Color 2: $(\neg x_2 \lor \neg x_5)$

Color 3: $(\neg x_3 \lor \neg x_6)$

What about with two colors?

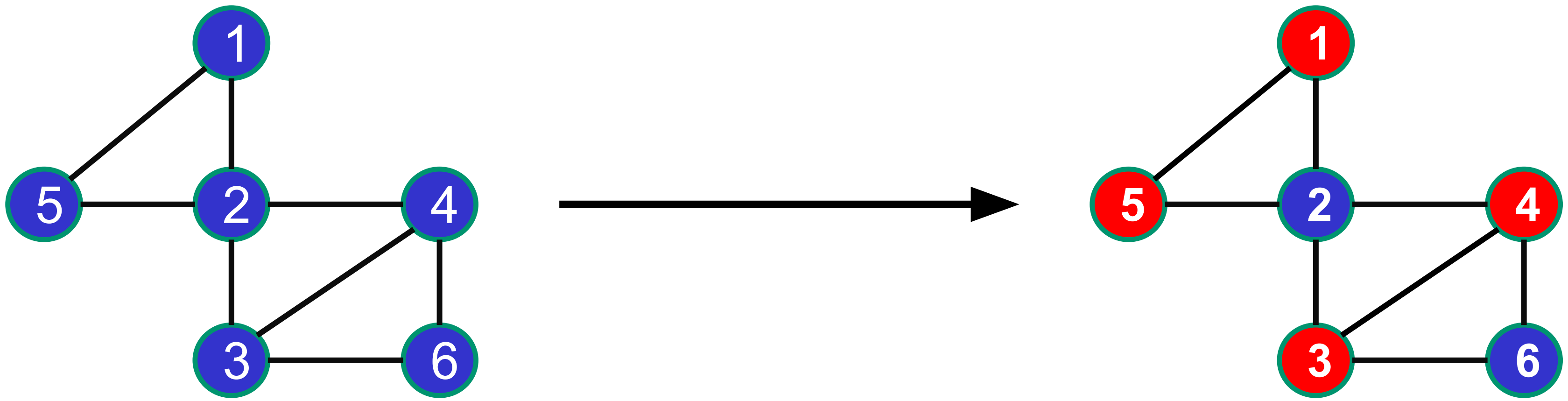
Frequency Allocation

In mobile telephony, the frequency allocation problem is stated as follows. There are a number of transmitters deployed and each of them can transmit on any of a given set of frequencies. Different transmitters have different frequency sets. Some transmitters are so close that they cannot transmit at the same frequency, because then they would interfere with each other. You are given the frequency range of each transmitter and the pairs of transmitters that can interfere if they use the same frequency. The problem is to determine if there is any possible choice of frequencies so that no transmitter interferes with any other.

Minimum Vertex Cover

A *vertex cover* of a graph G is a set *S* of vertices such that *S* contains at least one endpoint of every edge of G.

PROBLEM: ***To find the <u>minimum</u> size vertex cover***



Airline Operation

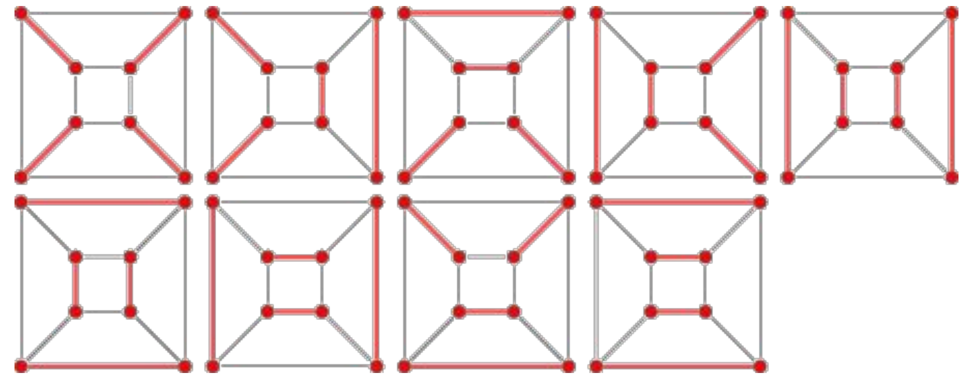
An airline company operates flights between various small (Class C/D/E) and large airports (Class B – like Chicago ORD). It wants to identify the least number of airport hubs from which it needs to operate its large aircrafts like the Boeing 747/777/787 or A-380/A-350. Come up with a SAT formulation that can help them.

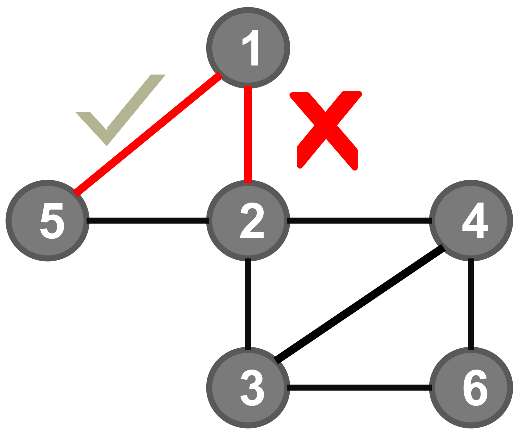
- You want the minimum number of airport hubs to operate from, so that all small airports are covered.
- We discriminate between airports (some cannot act as hubs) - Large aircrafts cannot land at all airports.
- By minimizing these hubs, the aircraft saves on operating costs.

Perfect Matching

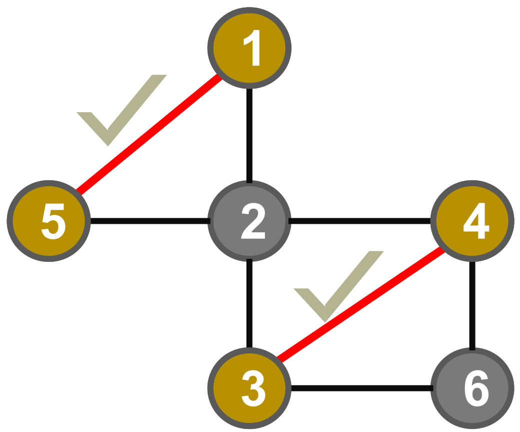
Matching: A choice of edges, every vertex has at most one edge of the matching incident on it.

Perfect Matching: A matching that covers all vertices

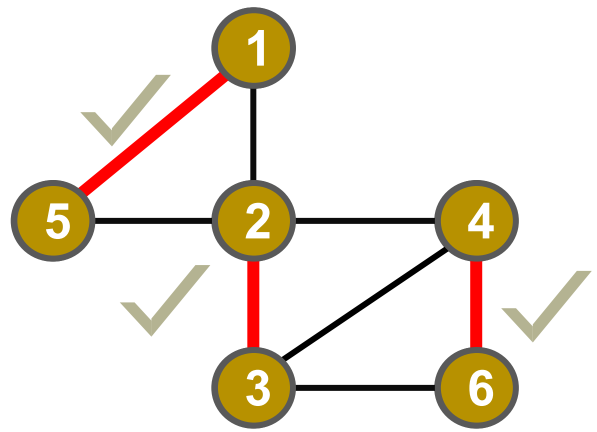




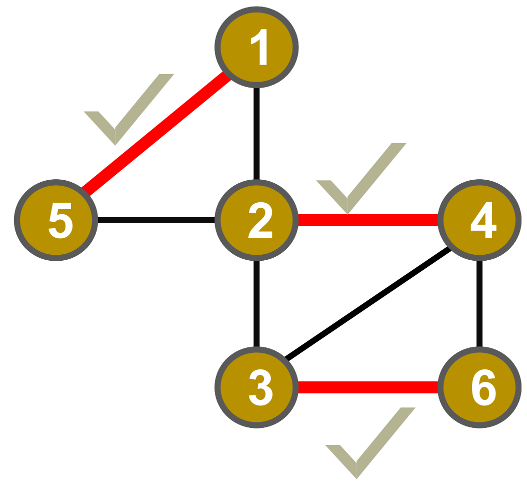
NOT a Matching
(1)



A Matching
(2)



Perfect Matching
(3)



Perfect Matching
(4)

Scheduling a Conference

Scheduling Speakers at a conference. There are N speakers and N time slots planned for a conference. Every speaker has a set of time slots in which there are available/unavailable. You wish to check if there is a way to assign a speaker to a preferred time slot, such that every speaker is able to speak at the conference.