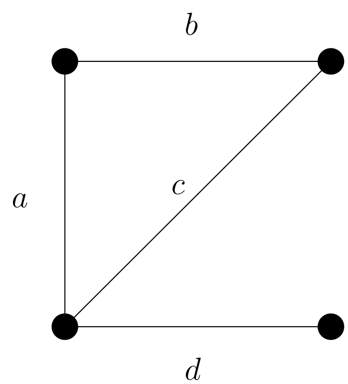
Question 18 **(15 marks)**

Consider the language 3-EDGE-COLOURABILITY, which consists of all graphs G such that we can assign colours to the edges of the graph so that (i) each edge is Red, White or Black, and (ii) any two edges that are incident at a common vertex must get different colours.

Let H be the following graph.



(a) Construct a Boolean expression E_H in Conjunctive Normal Form such that the satisfying truth assignments for E_H correspond to solutions to the 3-EDGE-COLOURABILITY problem on the above graph H (i.e., they correspond to colourings of the edges of H, using at most three colours and which give different colours to incident edges).

To do this, use variable names $a_R, a_W, a_B, b_R, b_W, b_B, c_R, c_W, c_B, d_R, d_W, d_B$, where variable e_X is True if and only if edge e gets colour X (where $e \in \{a, b, c, d\}$ and $X \in \{R, W, B\}$).

The solution needs to describe the *conjunction* of all clauses

$$e_R \vee e_W \vee e_B$$

over all edges e (these clauses ensure that each edge gets at least one colour), together with all clauses

$$\neg e_X \vee \neg f_X$$

over all pairs of incident edges e, f and all $X \in \{R, W, B\}$ (these clauses ensure that incident edges get different colours).

The possible pairs of incident edges are: a, b; a, c; a, d; b, c; c, d. (This is all pairs of edges except b, d.)

$$\bigwedge_{e \in \{a,b,c,d\}} (e_R \vee e_W \vee e_B) \quad \wedge \quad \bigwedge_{\substack{e \text{ incident with } f \\ X \in \{R,W,B\}}} (\neg e_X \vee \neg f_X)$$

OR

$$
\begin{aligned}
&(a_R \vee a_W \vee a_B) \wedge (b_R \vee b_W \vee b_B) \wedge (c_R \vee c_W \vee c_B) \wedge (d_R \vee d_W \vee d_B) \wedge \\
&(\neg a_R \vee \neg b_R) \wedge (\neg a_W \vee \neg b_W) \wedge (\neg a_B \vee \neg b_B) \wedge \\
&(\neg a_R \vee \neg c_R) \wedge (\neg a_W \vee \neg c_W) \wedge (\neg a_B \vee \neg c_B) \wedge \\
&(\neg a_R \vee \neg d_R) \wedge (\neg a_W \vee \neg d_W) \wedge (\neg a_B \vee \neg d_B) \wedge \\
&(\neg b_R \vee \neg c_R) \wedge (\neg b_W \vee \neg c_W) \wedge (\neg b_B \vee \neg c_B) \wedge \\
&(\neg c_R \vee \neg d_R) \wedge (\neg c_W \vee \neg d_W) \wedge (\neg c_B \vee \neg d_B)
\end{aligned}
$$

OR something like ...

$$
\begin{aligned}
&(a_R \vee a_W \vee a_B) \vee \cdots \vee (d_R \vee d_W \vee d_B) \vee \\
&(\neg a_R \vee \neg b_R) \wedge (\neg a_W \vee \neg b_W) \wedge (\neg a_B \vee \neg b_B) \wedge \\
&(\neg a_R \vee \neg c_R) \wedge \quad \cdots \\
&\qquad \cdots \\
&\cdots \quad \wedge (\neg c_B \vee \neg d_B)
\end{aligned}
$$

... provided it is clear which clauses are included, or how they are constructed.

Normally, in reducing 3-EDGE-COLOURABILITY to SATISFIABILITY, you need extra clauses to ensure that each edge gets *at most* one colour. Such clauses have the form

$$\neg e_X \vee \neg e_Y$$

for each edge e and each pair X, Y of distinct colours from the colour set $\{R, W, B\}$. BUT, in this case, it so happens that the structure of the graph ensures that this constraint is imposed by the other clauses anyway. So these clauses are not required in this case.

(b) Give a polynomial-time reduction from 3-EDGE-COLOURABILITY to SATISFIABILITY.

Input: Graph G.
For each edge e of G, create three new variables, and put them in a clause, $e_R \vee e_W \vee e_B$, and also create the three clauses $\neg e_R \vee \neg e_W$, $\neg e_R \vee \neg e_B$, $\neg e_W \vee \neg e_B$.
For each pair of incident edges e, f of G:
{
    For each colour X:
    {
        Create the new clause: $\neg e_X \vee \neg f_X$.
    }
}
Output: the conjunction of all the clauses created so far.

Official use only
15

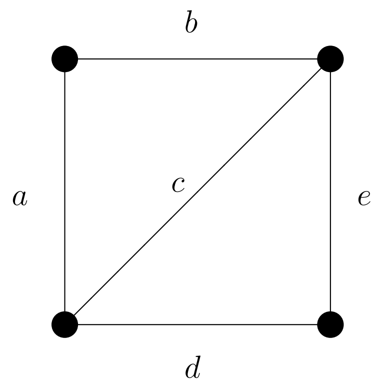
Question 18 **(13 marks)**

A **perfect matching** in a graph G is a subset X of the edge set of G that meets each vertex exactly once. In other words, no two edges in X share a vertex, and each vertex of G is incident with exactly one edge in X.

The PERFECT MATCHING decision problem is as follows.

PERFECT MATCHING
Input: Graph G.
Question: Does G have a perfect matching?

For example, in the following graph, the edge set $\{a, e\}$ is a perfect matching. But $\{a, b, e\}$ is not a perfect matching (since, for example, a and b share a vertex), and $\{a\}$ is not a perfect matching (since some vertices are not incident with the edge in this set).



Let W be the above graph.

(a) Construct a Boolean expression E_W in Conjunctive Normal Form such that the satisfying truth assignments for E_W correspond to perfect matchings in the above graph W.

$$
\begin{aligned}
&(a \vee b) \wedge (\neg a \vee \neg b) \wedge \\
&(a \vee c \vee d) \wedge (\neg a \vee \neg c) \wedge (\neg a \vee \neg d) \wedge (\neg c \vee \neg d) \wedge \\
&(b \vee c \vee e) \wedge (\neg b \vee \neg c) \wedge (\neg b \vee \neg e) \wedge (\neg c \vee \neg e) \wedge \\
&(d \vee e) \wedge (\neg d \vee \neg e).
\end{aligned}
$$

(b) Give a polynomial-time reduction from PERFECT MATCHING to SATISFIABILITY.

Input: graph G.
For each vertex v:
{
    Let the edges incident with v be $e_1, e_2, \ldots, e_d$.
        We need to create clauses which, when AND-ed together, say:
        Exactly one of these edges is in the matching.
    Create a new clause, $e_1 \vee \cdots \vee e_d$.
        This says: at least one of $e_1, \ldots, e_d$ is in the matching.
    For each pair e_i, e_j of edges incident with v:
        Create a new clause, $\neg e_i \vee \neg e_j$.
            This says: at least one of e_i, e_j is *not* in the matching.
}
Combine all these clauses using $\wedge$, to create the conjunction of all of them.
Output the resulting expression.

Official use only
13

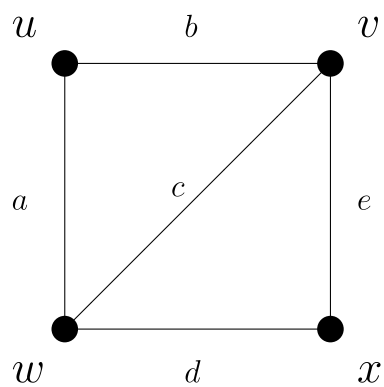
Question 17 **(11 marks)**

A **vertex cover** in a graph G is a set X of vertices that meets every edge of G. So, every edge is incident with at least one vertex in X.

The VERTEX COVER decision problem is as follows.

VERTEX COVER
Input: Graph G.
Question: Does G have a vertex cover?

For example, in the following graph, the vertex set $\{u, w, x\}$ is a vertex cover, and so is $\{v, w\}$. But $\{u, x\}$ is not a vertex cover, since it does not meet every edge. (Specifically, it misses the diagonal edge c.)



Let W be the above graph.

(a) Construct a Boolean expression E_W in Conjunctive Normal Form such that the satisfying truth assignments for E_W correspond to vertex covers in the above graph W.

$$(u \vee v) \wedge (u \vee w) \wedge (v \vee w) \wedge (v \vee x) \wedge (w \vee x)$$

(b) Give a polynomial-time reduction from VERTEX COVER to SATISFIABILITY.

Input: graph G.
For each vertex of G, create a variable to represent that vertex.
For each edge e of G:
{
Create a clause consisting of a disjunction of the two variables associated with the endpoints of e.
}
Combine all the clauses using conjunction.
Output the expression so formed.

Alternative solution:

This variant of VERTEX COVER has no constraint on the size of the vertex cover. So, in fact, for any input graph G, the answer is always YES, since every graph has a vertex cover: take the entire vertex set of the graph, for example.[1]

So, for a polynomial-time reduction, it's sufficient to just map any input graph to some fixed satisfiable CNF expression. (Any input string that is not even a graph could be mapped to a fixed **un**satisfiable CNF expression.)

Official use only
11

[1]Normally, the VERTEX COVER problem includes a second input, being a positive integer k, and the problem asks if there is a vertex cover *size at most* k. That version is NP-complete, as shown in the last lecture.

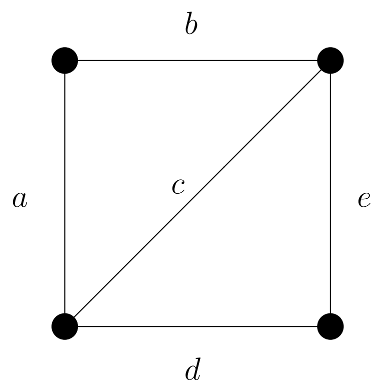
Question 17 **(12 marks)**

An **edge cover** in a graph G is a set X of edges that meets every vertex of G. So, every vertex is incident with at least one edge in X.

The EDGE COVER decision problem is as follows.

EDGE COVER
Input: Graph G.
Question: Does G have an edge cover?

For example, in the following graph, the edge set $\{a, b, c, d\}$ is an edge cover, and so is $\{a, e\}$. But $\{a, b, c\}$ is not an edge cover, since it does not meet every vertex. (Specifically, it misses the bottom right vertex.)



Let W be the above graph.

(a) Construct a Boolean expression E_W in Conjunctive Normal Form such that the satisfying truth assignments for E_W correspond to edge covers in the above graph W.

$$(a \vee b) \wedge (a \vee c \vee d) \wedge (b \vee c \vee e) \wedge (d \vee e).$$

(b) Give a polynomial-time reduction from EDGE COVER to SATISFIABILITY.

Input: graph G.
For each vertex v:
{
    Let the edges incident with v be $e_1, e_2, \ldots, e_d$.
    Create a new clause, $e_1 \vee \cdots \vee e_d$.
        This says: at least one of $e_1, \ldots, e_d$ is in the edge cover.
}
Combine all these clauses using $\wedge$, to create the conjunction of all of them.
Output the resulting expression.

Official use only
12