CS2020A Discrete Mathematics

Aug-Dec 2025 | Second Test | 30 points

IIT Palakkad

18-Oct-2025 | 8.00 to 9.15 AM

1. Let P be a 3-ary predicate defined on a finite domain D. P can be represented as a binary 3D-Matrix M (a tensor), where $M(i, j, k) = 1$ if $P(i, j, k) = True$ and $M(i, j, k) = 0$ if $P(i, j, k) = False$. The statement

   "M has an all-ones plane perpendicular to its first axis"

   is the first order sentence $\exists x \forall y \forall z P(x, y, z)$ (where x, y, z are domain variables). Write down similar first order sentences for the following statements. (4 points)

   (a) *M has an all-zeros plane perpendicular to its second axis*

   (b) *M has an all-ones line perpendicular to two of its three axes*

   (c) *M has a zero in each plane perpendicular to its third axis*

   (d) *Each plane perpendicular to the first axis in M represents a symmetric binary relation.*

   Answers.

   (a) $\exists y \forall x \forall z \neg P(x, y, z)$

   (b) $\exists x \exists y \forall z P(x, y, z) \vee \exists y \exists z \forall x P(x, y, z) \vee \exists z \exists x \forall y P(x, y, z)$

   (c) $\forall z \exists x \exists y \neg P(x, y, z)$

   (d) $\forall x \forall y \forall z (P(x, y, z) \leftrightarrow P(x, z, y))$

   (1 point for each correct formula)

2. Let $\phi = \forall x \exists y \forall z P(x, y, z)$ be a formula over the domain $D = \{1, \dots, 100\}$.

   a) Write a single python function using nested loops which evaluates ϕ. (2 points)

   b) Write a python program using multiple functions, each using a single loop, to evaluate ϕ. (3 points)

   Note. You can assume that the predicate P is already implemented as a function and you can call it as $P(x, y, z)$.

   The function aeaP() returns $\forall x \exists y \forall z P(x, y, z)$

```
def aeaP() -> bool:
    bool q
    for x in D:
        for y in D:
            for z in D:
                q = P(x,y,z)
                if q == False break
            if q == True break
        if q == False break
    return q
```

   Answer b.

   - The function xyaP(x,y) returns $\forall z P(x, y, z)$
   - The function xeaP(x) returns $\exists y \forall z P(x, y, z)$
   - The function aeaP() returns $\forall x \exists y \forall z P(x, y, z)$

```
def xyaP(x, y) -> bool:
    for z in D:
        if P(x, y, z) == False
            return False
    return True

def xeaP(x) -> bool:
    for y in D:
        if xyaP(x, y) == True
            return True
    return False

def aeaP() -> bool:
    for x in D:
        if xeaP(x) == False
            return False
    return True
```

3. Write a first order formula ϕ which is true if and only if a set P of pigeons can occupy a set H of holes such that every pigeon in P occupies a hole in H and every hole in H is occupied by at most 1 pigeon. Let $A(x, y)$ denote the predicate "x is occupying y".

   (a) Write the formula using shorcuts like $\forall i_{\in P}$, $\exists j_{\in H}$ etc. (2 points)

   Answer:

$$\phi = [(\forall i_{\in P} \exists j_{\in H} A(i,j)] \wedge [\forall j_{\in H} \forall i_{\in P} \forall i'_{\in P} (i \neq i') \rightarrow \neg(A(i,j) \wedge A(i',j))]$$

   (b) Rewrite the above formula if the domain of discourse D contains the pigeons and the holes and you are not allowed to quantify over subsets of D (you cannot use constructs like $\forall i_{\in P}$, $\exists j_{\in H}$ etc.) However, you can use the following two additional predicates

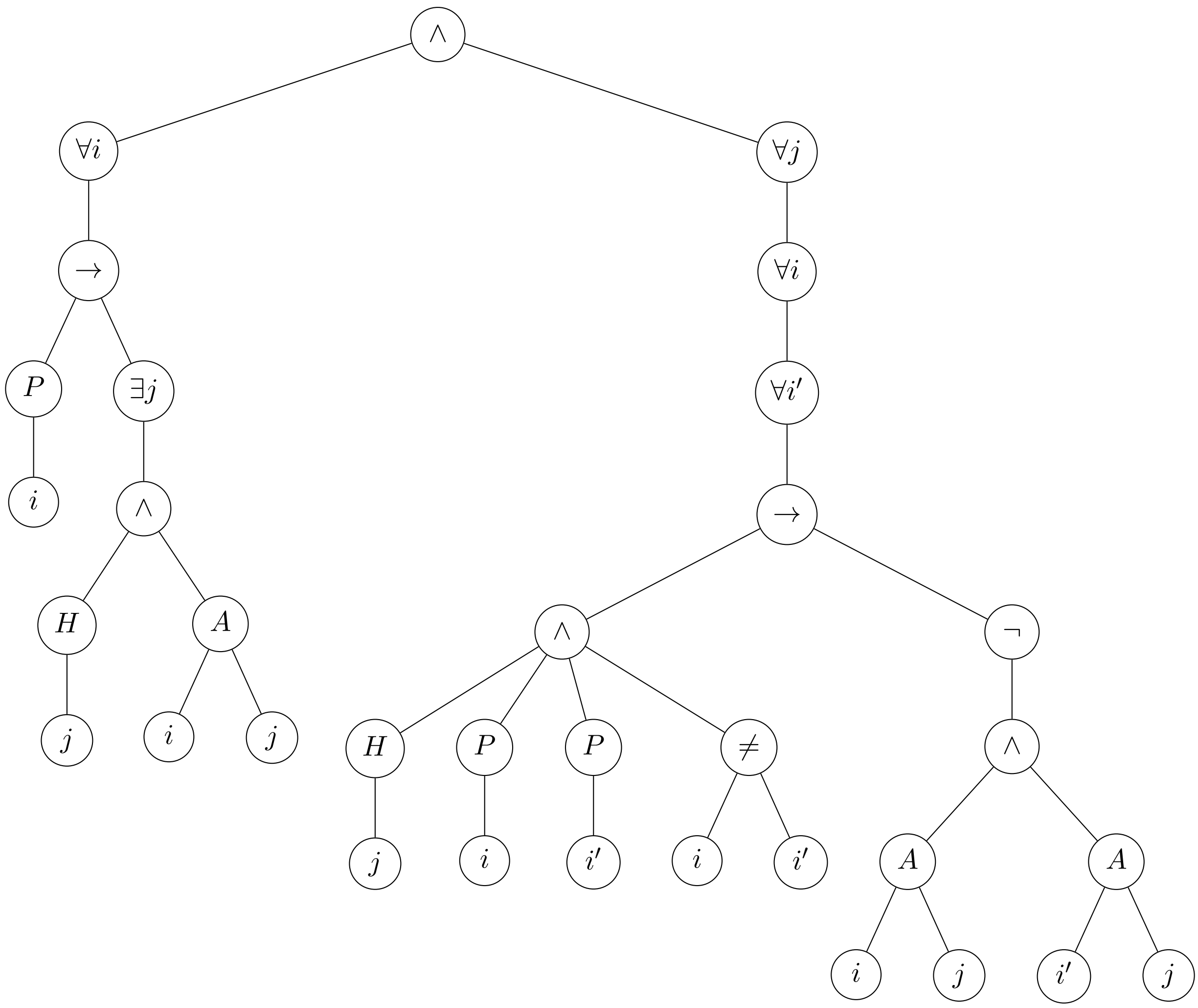
   - P(x) : x is a pigeon
   - H(x) : x is a hole (2 points)

   Answer:

$$\phi = [(\forall i (P(i) \rightarrow \exists j (H(j) \wedge A(i,j))] \wedge [\forall j \forall i \forall i' (H(j) \wedge P(i) \wedge P(i') \wedge (i \neq i')) \rightarrow \neg(A(i,j) \wedge A(i',j))]$$

   (c) Draw the parse tree for the above formula. (2 points)

   Answer



4. When do we say that two simple undirected graphs are *isomorphic*? (2 points)

   Ans: Two simple undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic* if there exists a bijection f from V_1 to V_2 such that

$$\{x, y\} \in E_1 \iff \{f(x), f(y)\} \in E_2$$

5. Look at the three graphs G_1, G_2, G_3 in Figure 1.

   (a) Is G_1 isomorphic to G_2.
   If yes, give an isomorphism from G_1 to G_2. If no, give a justification. (2 points)

   Ans: Yes G_1 is isomorphic to G_2. The following bijection can be verified to be an isomorphism. $(a, e, f, d) \mapsto (v, w, z, y)$, $(d \mapsto u)$, and $(b \mapsto x)$.

   Marks. 1 point for saying Yes. 1 point for any correct isomorphism.

   (b) Is G_1 isomorphic to G_3.
   If yes, give an isomorphism from G_2 to G_3. If no, give a justification. (2 points)

   Ans: No. Number of edges in G_3 is different from the number of edges in G_1.

   Marks. 1 point for saying No. 1 point for pointing out any graph invariant on which G_1 and G_3 differs.

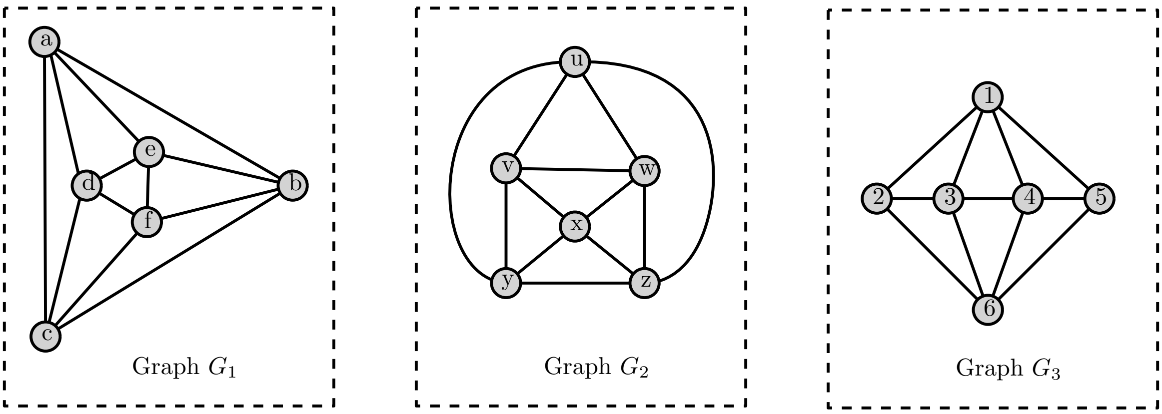


Figure 1: Three Graphs

6. Prove that the number of odd degree vertices in any graph is a even. (3 points)

Answer.

- Let V be the vertex set of any simple undirected graph G.
- $\sum_{v \in V} d(v)$ (where $d(v)$ denotes degree of v) is equal to twice the number of edges in G and hence an even number.
- If there are are odd number of vertices in V with odd degree, then the above sum is an odd number.
- Hence the number of odd degree vertices in G is even.

7. Define the following (assume simple undirected graphs)

   (a) Connected
   (b) Cycle
   (c) Tree (3 points)

Answer.

(a) A simple undirected graph G is *connected* if there is a path between every pair of vertices in G.

(b) A *cycle* in a simple undirected graph G is an alternating sequence of vertices and edges starting and ending with the same vertex, no other vertex repeating, having at least one edge and such that every edge in the sequence is preceded by one of its end points and succeedded by the other.

(c) A *tree* is a connected acyclic undirected graph.

8. Prove that every tree on n vertices ($n \geq 1$) has exactly $n - 1$ edges. (5 points)

Answer. Proof by ordinary induction on n.

- When $n = 1$, the number of edges is 0 and hence the claim is true.
- Let T be a tree on $k + 1$ vertices for some $k \geq 1$ and assume that the claim is true for every tree on k vertices.
- Take the last vertex v of any maximal path in T. Since a tree is acyclic by definition, v has degree exactly 1.
- Remove v and the single edge touching it from T to get a new graph T'. T' is acyclic since removing a vertex or edge from an acyclic graph (T in this case) will not create a cycle.
- T' remains connected since v, being a degree one vertex was not an intermediate vertex in a path connecting any pair of vertices in T.
- Hence T' is a tree and by induction hypothesis it has $k - 1$ edges. So T had exactly k edges.
- By principle of mathematical induction, any tree on n vertices has exactly $n - 1$ edges.