Network Analysis - Quiz 2

Generating DAGs from a CPDAG

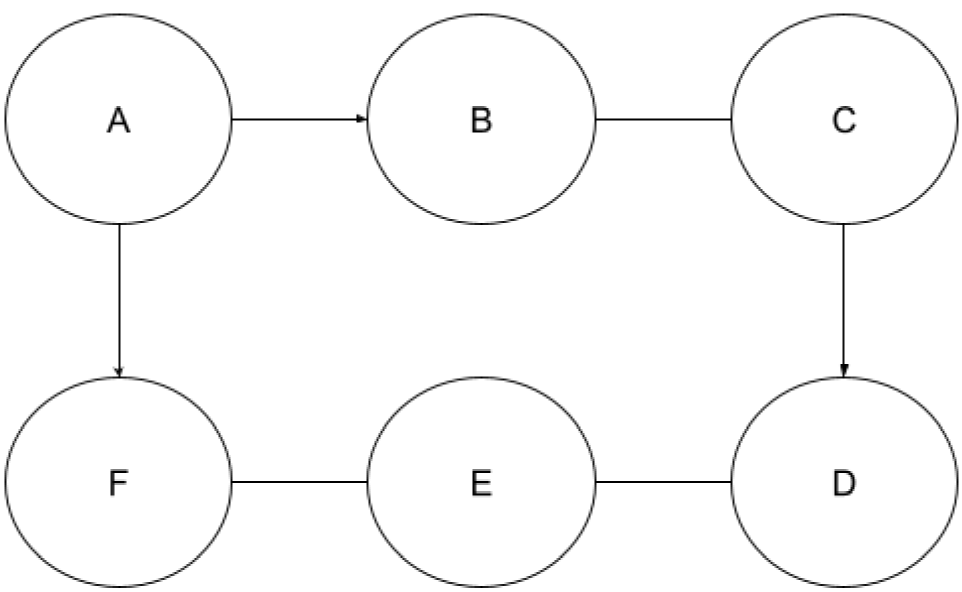
Below, you are given a CPDAG (Completed Partially Directed Acyclic Graph) representing an equivalence class of DAGs. Use this structure to answer the questions below. You must clearly indicate node relationships and provide justification for your answers.

The CPDAG has the following nodes: {A, B, C, D, E, F}. Its edges are:

- A → B
- B — C
- C → D
- D — E
- E — F
- A → F

Note: "—" denotes an undirected edge; "→" denotes a directed edge.

1. Draw the CPDAG



2. For the undirected edges in the CPDAG, explain what constraints (if any) exist when orienting them to form a valid DAG. Specifically, what determines which orientations are valid?

For each undirected edge we may choose either direction **provided that** :

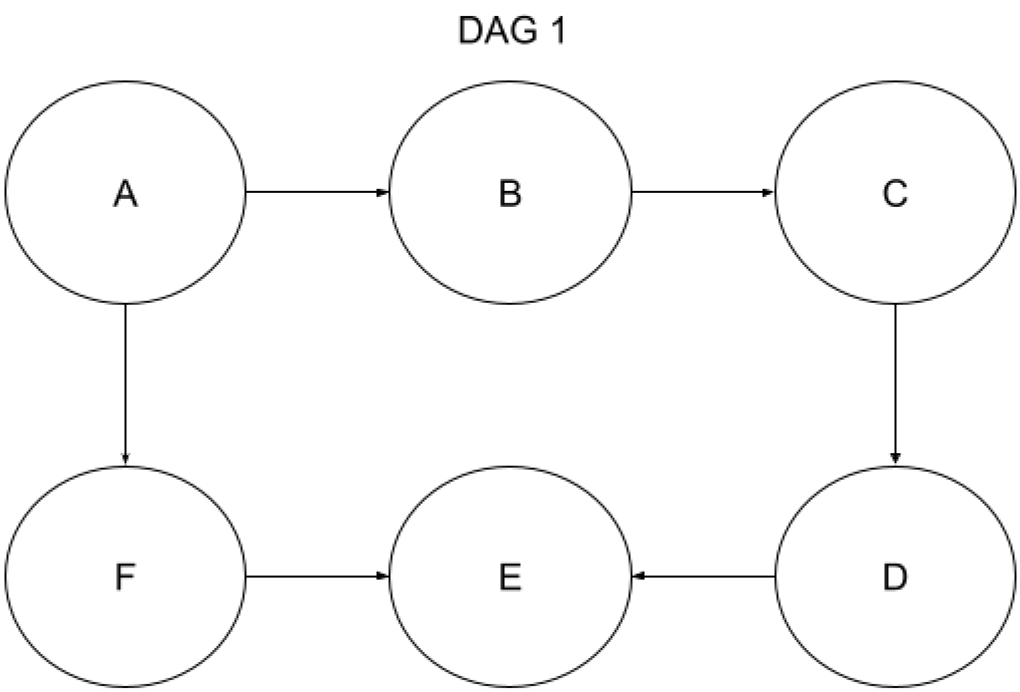
1. **Acyclicity** is preserved (no feedback loops).
2. **Skeleton & v-structures** of the CPDAG are respected: we must **not**

- create a new v-structure that is absent from the CPDAG, or
- destroy a v-structure that is already compelled.

Edges whose reversal would violate either rule are *compelled*; the rest are *reversible*. Meek's orientation rules implement these checks algorithmically.

3. Generate DAGs from this CPDAG that represents this equivalence class. For each DAG, list all directed edges explicitly and draw them.

Below are **two** legal orientations of the CPDAG (many others exist). Both are acyclic and obey the constraints above.



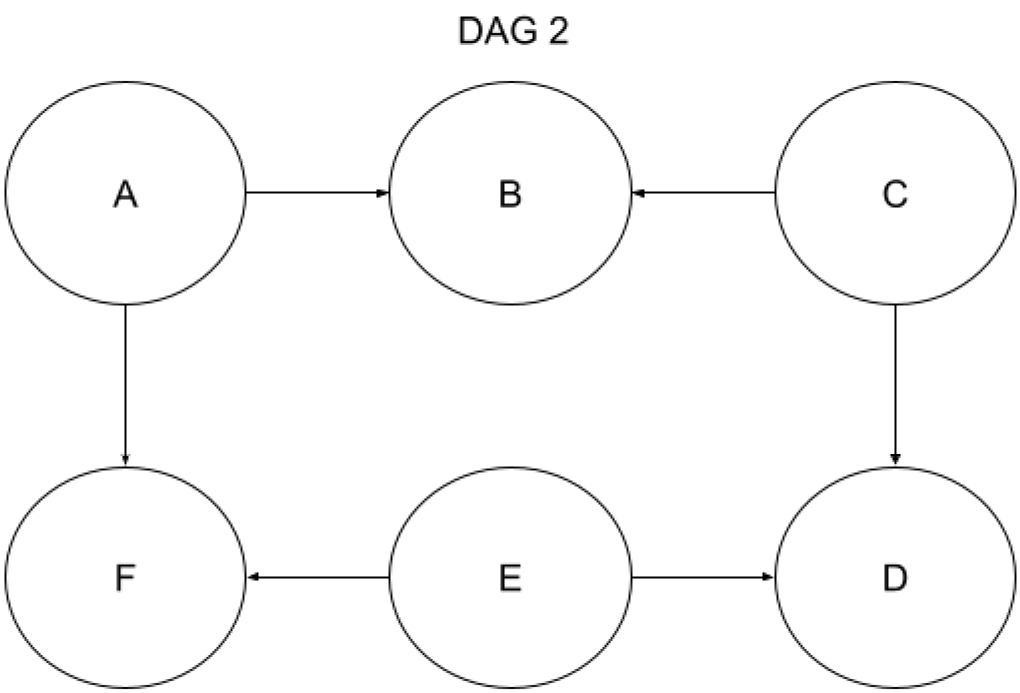
Directed Edge List:

```
A→B, B→C, C→D, D→E, F→E, A→F
```

This is valid because:

1. No new v-structures are introduced beyond what is allowed by the CPDAG.

E has parents D and F, forming D → E ← F, which is a valid v-structure since the CPDAG did not specify a direction for D—E or E—F, allowing either.
2. No cycles are introduced (acyclic property maintained).
F → E → … does not loop back to F.
A → F → E → … never loops back to A.



Directed Edge List:

```
A→B, C→B, C→D, E→D, E→F, A→F
```

This is valid because:
1. No new v-structures introduced:
A→B←C is a valid v-structure (the CPDAG allowed B—C to be oriented either way, so this is fine).
E→D←C is also a valid new v-structure because D—E was undirected and can be oriented this way.
2. No directed cycles:
All paths progress forward without looping back.

4. Do two DAGs from the same CPDAG encode the same set of conditional independencies? Why or why not?
Yes.
By definition, all DAGs represented by a given CPDAG share **(i) the same underlying undirected skeleton** and **(ii) the same collection of v-structures**.

When two DAGs agree on both of these features they are *Markov-equivalent*, meaning they imply an identical set of conditional-independence relations. Consequently, any pair of DAGs taken from the same CPDAG will encode exactly the same conditional independencies.

5. Are the moral graphs of DAGs from the same equivalence class the same? Provide a toy example.

Yes.

Moralisation only needs two ingredients that every DAG in a Markov-equivalence class shares:

1. the **skeleton** (which edges exist, ignoring arrowheads), and
2. the set of **v-structures** (colliders).

Since those are fixed for the whole class, marrying the parents of each collider and then dropping arrowheads produces exactly the same undirected graph for every member of the class.

Toy example:

Take four nodes {A, B, C, D} and the common skeleton A — C — B and C — D.

DAG 1 : A → C ← B, **C** → **D**

DAG 2 : A → C ← B, **D** → **C**

The two DAGs differ only in the orientation of the C–D edge, so they are Markov-equivalent.

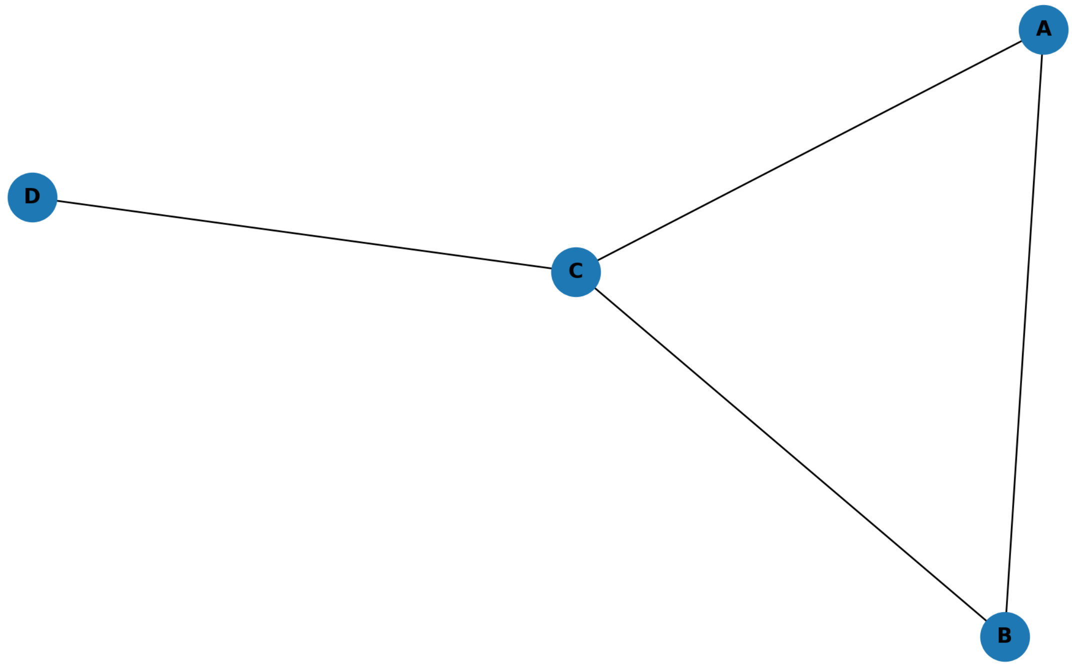
Moralise each DAG

1. Drop arrowheads: A–C, B–C, C–D

2. Marry collider parents: add edge A–B (they both point to C)

Resulting moral graph (identical for both):

Moral graph with nodes A, B, C, D



Thus, even though the original DAGs are different, their moral graphs coincide, confirming the statement.

6. TRUE or FALSE -
Suppose you use the junction tree to do exact inference for probabilistic queries in the two graphs you developed. All probabilistic queries performed would produce equivalent results across the graphs.

TRUE.

Because the moral graphs are identical, the junction trees (built from triangulated moral graphs) are also the same. Therefore, any exact inference (like via junction tree algorithm) will yield identical results across these DAGs.

7. Briefly describe an advantage that likelihood weighting approximate inference has over logical sampling.

Likelihood weighting keeps all samples by fixing evidence variables and reweighting them, whereas logical sampling often rejects many samples that don't match the evidence. This means likelihood weighting has much lower variance and is far more efficient, especially when the evidence is rare.