Bayesian Networks

2. Separation in Graphs

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Business Economics and Information Systems
& Institute for Computer Science
University of Hildesheim
http://www.ismll.uni-hildesheim.de

Graphs

Definition 1. Let V be any set and

$$E \subseteq \mathcal{P}^2(V) := \{\{x,y\} \,|\, x,y \in V\}$$

be a subset of sets of unordered pairs of V. Then $G := (V, E)$ is called an **undirected graph**. The elements of V are called **vertices** or **nodes**, the elements of E **edges**.

Let $e = \{x,y\} \in E$ be an edge, then we call the vertices x, y **incident** to the edge e. We call two vertices $x, y \in V$ **adjacent**, if there is an edge $\{x,y\} \in E$.

The set of all vertices adjacent with a given vertex $x \in V$ is called its **fan**:

$$\text{fan}(x) := \{y \in V \,|\, \{x,y\} \in E\}$$

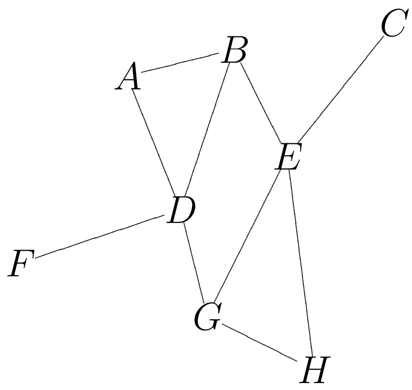


Figure 1: Example graph.

Paths on graphs

Definition 2. Let V be a set. We call $V^* := \bigcup_{i\in\mathbb{N}} V^i$ the **set of finite sequences in** V. The length of a sequence $s \in V^*$ is denoted by $|s|$.

Let $G = (V, E)$ be a graph. We call

$$G^* := V^*_{|G} := \{p \in V^* \,|\, \{p_i, p_{i+1}\} \in E, \\ i = 1, \ldots, |p| - 1\}$$

the **set of paths on** G.

Any contiguous subsequence of a path $p \in G^*$ is called a **subpath of** p, i.e. any path $(p_i, p_{i+1}, \ldots, p_j)$ with $1 \leq i \leq j \leq n$. The subpath $(p_2, p_3, \ldots, p_{n-1})$ is called the **interior of** p. A path of length $|p| \geq 2$ is called **proper**.

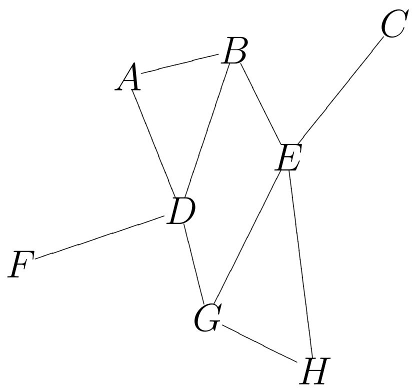


Figure 2: Example graph.

The sequences

$$(A, D, G, H)$$
$$(C, E, B, D)$$
$$(F)$$

are paths on G, but the sequences

$$(A, D, E, C)$$
$$(A, H, C, F)$$

are not.

Separation in graphs (u-separation)

Definition 3. Let $G := (V, E)$ be a graph. Let $Z \subseteq V$ be a subset of vertices. We say, two vertices $x, y \in V$ are **u-separated by Z in G**, if every path from x to y contains some vertex of Z ($\forall p \in G^* : p_1 = x, p_{|p|} = y \Rightarrow \exists i \in \{1, \ldots, n\} : p_i \in Z$).

Let $X, Y, Z \subseteq V$ be three disjoint subsets of vertices. We say, the vertices X and Y are **u-separated by Z in G**, if every path from any vertex from X to any vertex from Y is separated by Z, i.e., contains some vertex of Z.

We write $I_G(X, Y|Z)$ for the statement, that X and Y are u-separated by Z in G.

I_G is called **u-separation relation in G**.

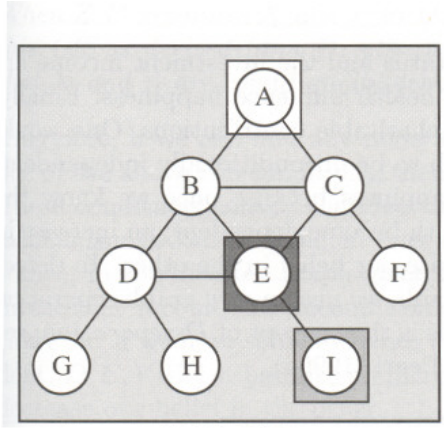


Figure 3: Example for u-separation [CGH97, p. 179].

Separation in graphs (u-separation)

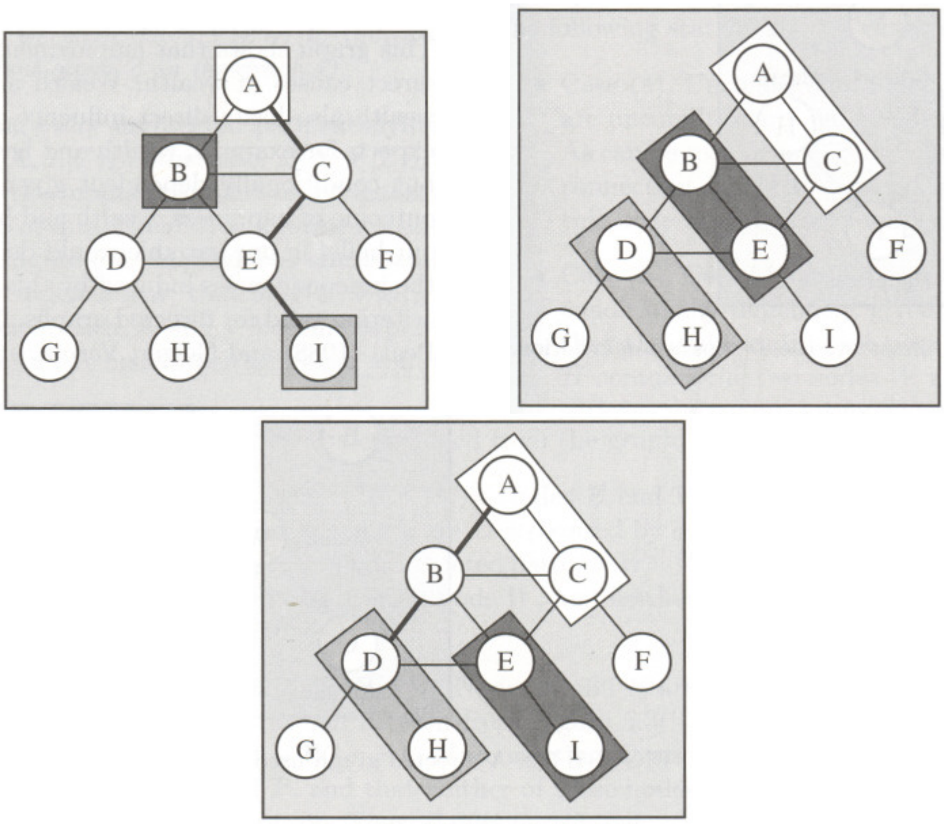


Figure 4: More examples for u-separation [CGH97, p. 179].

Properties of u-separation / no chardality

For u-separation the chordality property does not hold (in general).

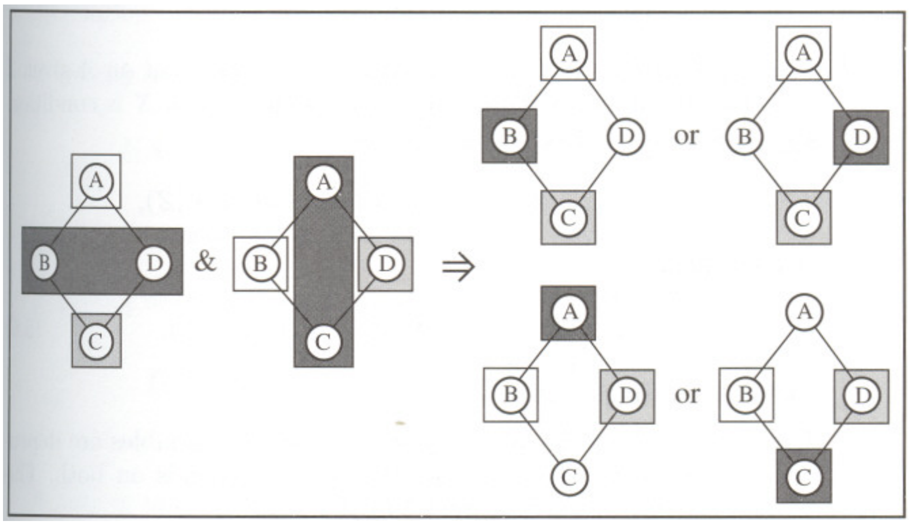


Figure 5: Counterexample for chordality in undirected graphs (u-separation) [CGH97, p. 189].

Properties of u-separation

relation	symmetry	decomposition	composition	strong union	weak union	contraction	intersection	strong transitivity	weak transitivity	chordality
u-separation	+	+	+	+	+	+	+	+	+	−

Checking u-separation

To test, if for a given graph $G = (V, E)$ two given sets $X, Y \subseteq V$ of vertices are u-separated by a third given set $Z \subseteq V$ of vertices, we may use standard breadth-first search to compute all vertices that can be reached from X (see, e.g., [OW02], [CLR90]).

breadth-first search(G, X) :
border $:= X$
reached $:= \emptyset$
while *border* $\neq \emptyset$ **do**
    reached $:=$ *reached* $\cup$ *border*
    border $:= \text{fan}_G($*border*$) \setminus$ *reached*
od
return *reached*

Figure 6: Breadth-first search algorithm for enumerating all vertices reachable from X.

For checking u-separation we have to tweak the algorithm

1. not to add vertices from Z to the border and
2. to stop if a vertex of Y has been reached.

check-u-separation(G, X, Y, Z) :
border $:= X$
reached $:= \emptyset$
while *border* $\neq \emptyset$ **do**
    reached $:=$ *reached* $\cup$ *border*
    border $:= \text{fan}_G($*border*$) \setminus$ *reached* $\setminus Z$
    if *border* $\cap Y \neq \emptyset$
        return *false*
    fi
od
return *true*

Figure 7: Breadth-first search algorithm for checking u-separation of X and Y by Z.

Symmetry

Definition 4. Let V be any set and I a ternary relation on $\mathcal{P}(V)$, i.e., $I \subseteq (\mathcal{P}(V))^3$.

I is called **symmetric**, if

$$I(X, Y|Z) \Rightarrow I(Y, X|Z)$$

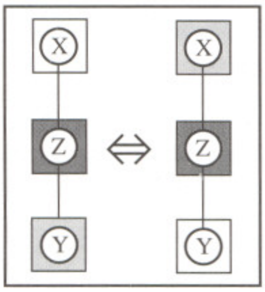


Figure 8: Examples for symmetry [CGH97, p. 186].

Decomposition and Composition

Definition 5. I is called **(right-)decomposable**, if

$$I(X, Y|Z) \Rightarrow I(X, Y'|Z) \quad \text{for any } Y' \subseteq Y$$

I is called **(right-)composable**, if

$$I(X, Y|Z) \text{ and } I(X, Y'|Z) \Rightarrow I(X, Y \cup Y'|Z)$$

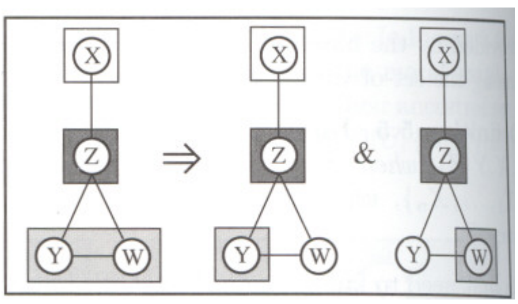


Figure 9: Examples for decomposition [CGH97, p. 186].

Union

Definition 6. I is called **strongly unionable**, if

$$I(X,Y|Z) \Rightarrow I(X,Y|Z \cup Z') \quad \text{for all } Z' \text{ disjunct with } X,Y$$

I is called **(right-)weakly unionable**, if

$$I(X,Y|Z) \Rightarrow I(X,Y'|(Y \setminus Y') \cup Z) \quad \text{for any } Y' \subseteq Y$$

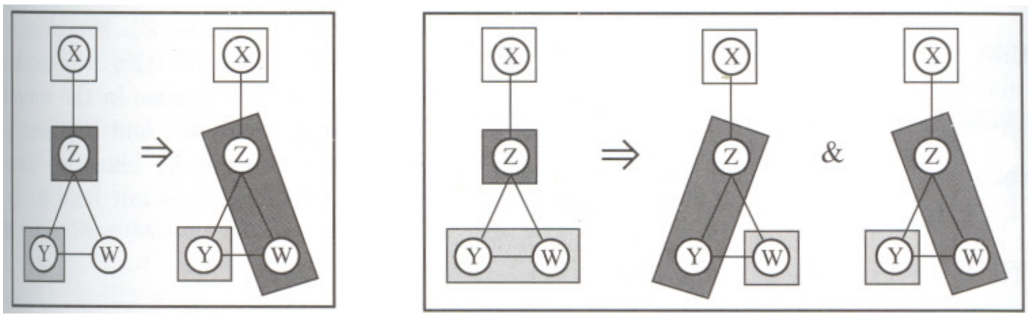


Figure 10: Examples for a) strong union and b) weak union [CGH97, p. 186,189].

Contraction and Intersection

Definition 7. I is called **(right-)contractable**, if

$$I(X,Y|Z) \text{ and } I(X,Y'|Y \cup Z) \Rightarrow I(X,Y \cup Y'|Z)$$

I is called **(right-)intersectable**, if

$$I(X,Y|Y' \cup Z) \text{ and } I(X,Y'|Y \cup Z) \Rightarrow I(X,Y \cup Y'|Z)$$

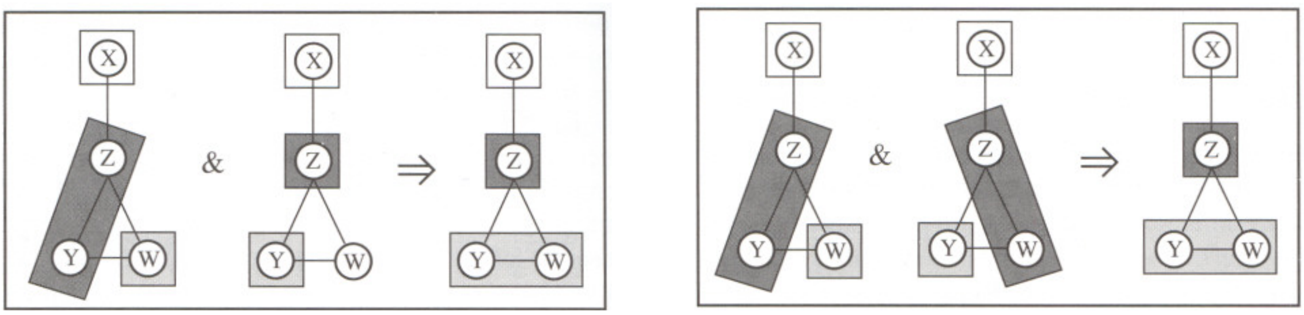


Figure 11: Examples for a) contraction and b) intersection [CGH97, p. 186].

Transitivity

Definition 8. I is called **strongly transitive**, if

$$I(X,Y|Z) \Rightarrow I(X,\{v\}|Z) \text{ or } I(\{v\},Y|Z) \quad \forall v \in V \setminus Z$$

I is called **weakly transitive**, if

$$I(X,Y|Z) \text{ and } I(X,Y|Z\cup\{v\}) \Rightarrow I(X,\{v\}|Z) \text{ or } I(\{v\},Y|Z) \quad \forall v \in V\setminus Z$$

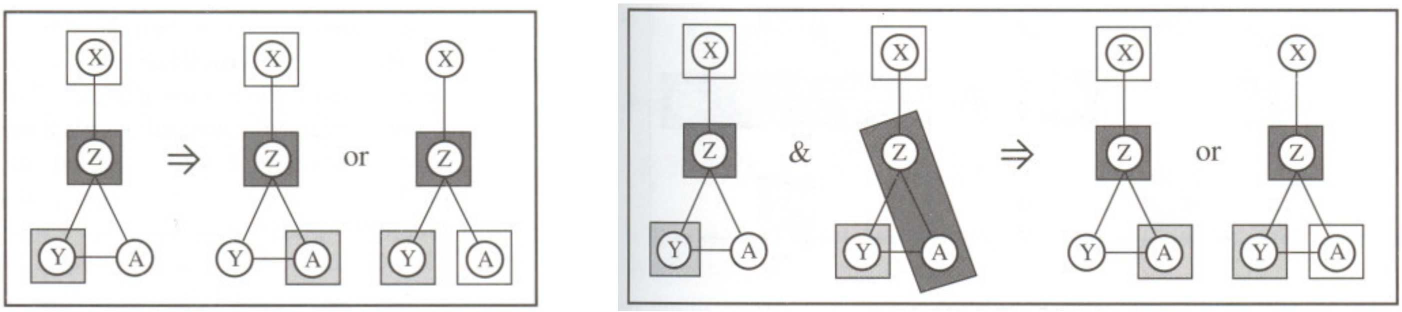


Figure 12: Examples for a) strong transitivity and b) weak transitivity. [CGH97, p. 189]

Chordality

Definition 9. I is called **chordal**, if

$$I(\{a\},\{c\}|\{b,d\}) \text{ and } I(\{b\},\{d\}|\{a,c\}) \Rightarrow I(\{a\},\{c\}|\{b\}) \text{ or } I(\{a\},\{c\}|\{d\})$$

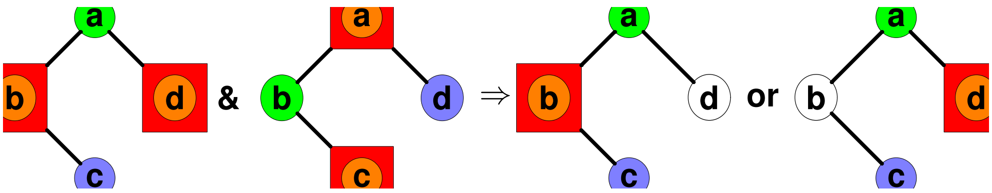


Figure 13: Example for chordality.

1. Separation in Undirected Graphs

2. Properties of Ternary Relations on Sets

3. Separation in Directed Graphs

Directed graphs

Definition 10. Let V be any set and

$$E \subseteq V \times V$$

be a subset of sets of ordered pairs of V. Then $G := (V, E)$ is called a **directed graph**. The elements of V are called **vertices** or **nodes**, the elements of E **edges**.

Let $e = (x, y) \in E$ be an edge, then we call the vertices x, y **incident** to the edge e. We call two vertices $x, y \in V$ **adjacent**, if there is an edge $(x, y) \in E$ or $(y, x) \in E$.

The set of all vertices with an edge from a given vertex $x \in V$ is called its **fanout**:

$$\text{fanout}(x) := \{y \in V \,|\, (x, y) \in E\}$$

The set of all vertices with an edge to a given vertex $x \in V$ is called its **fanin**:

$$\text{fanin}(x) := \{y \in V \,|\, (y, x) \in E\}$$

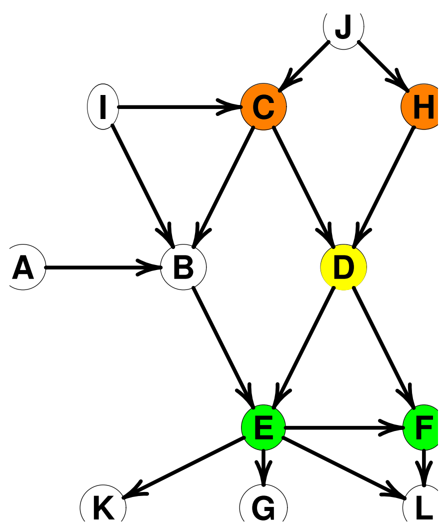


Figure 14: Fanin (orange) and fanout (green) of a node (blue).

Paths on directed graphs

Definition 11. Let $G = (V, E)$ be a directed graph. We call

$$G^* := V^*_{|G} := \{p \in V^* \mid (p_i, p_{i+1}) \in E, \\ i = 1, \ldots, |p| - 1\}$$

the **set of paths on** G. For two vertices $x, y \in V$ we denote by

$$G^*_{[x,y]} := \{p \in V^*_{|G} \mid p_1 = x, p_{|p|} = y\}$$

the **set of paths from** x **to** y.

The notions of **subpath**, **interior**, and **proper path** carry over to directed graphs.

A proper path $p = (p_1, \ldots, p_n) \in G^*$ with $p_1 = p_n$ is called **cyclic**. A path without cyclic subpath is called a **simple path**. A graph without a cyclic path is called **directed acyclig graph (DAG)**.

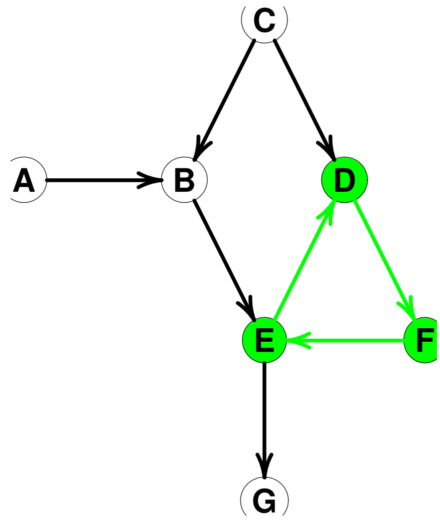


Figure 15: Example for a cycle.

Paths on directed graphs (2/2)

Definition 12. For a DAG G vertices of the fanout are also called **children**

$$\text{child}(x) := \text{fanout}(x) := \{y \in V \mid (x, y) \in E\}$$

and the vertices of the fanin **parents**:

$$\text{pa}(x) := \text{fanin}(x) := \{y \in V \mid (y, x) \in E\}$$

Vertices y with a proper path from y to x are called **ancestors of** x:

$$\text{anc}(x) := \{y \in V \mid \exists p \in G^* : |p| \geq 2, \\ p_1 = y, p_{|p|} = x\}$$

Vertices y with a proper path from x to y are called **descendents of** x:

$$\text{desc}(x) := \{y \in V \mid \exists p \in G^* : |p| \geq 2, \\ p_1 = x, p_{|p|} = y\}$$

Vertices that are not a descendent of x are called **nondescendents of** x.

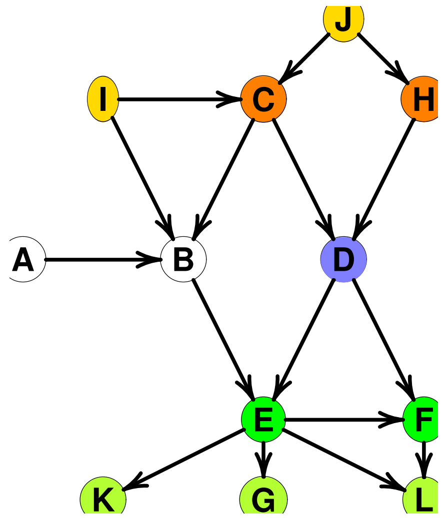


Figure 16: Parents/Fanin (orange) and additional ancestors (light orange), children/fanout (green) and additional descendants (light green) of a node (blue).

Chains

Definition 13. Let $G := (V, E)$ be a directed graph. We can construct an **undirected skeleton** $u(G) := (V, u(E))$ **of** G by dropping the directions of the edges:

$$u(E) := \{\{x, y\} \mid (x, y) \in E \text{ or } (y, x) \in E\}$$

The paths on $u(G)$ are called **chains of** G:

$$G^{\blacktriangle} := u(G)^*$$

i.e., a chain is a sequence of vertices that are linked by a forward or a backward edge. If we want to stress the directions of the linking edges, we denote a chain $p = (p_1, \ldots, p_n) \in G^{\blacktriangle}$ by

$$p_1 \leftarrow p_2 \rightarrow p_3 \leftarrow \cdots \leftarrow p_{n-1} \rightarrow p_n$$

The notions of **length**, **subchain**, **interior** and **proper** carry over from undirected paths to chains.

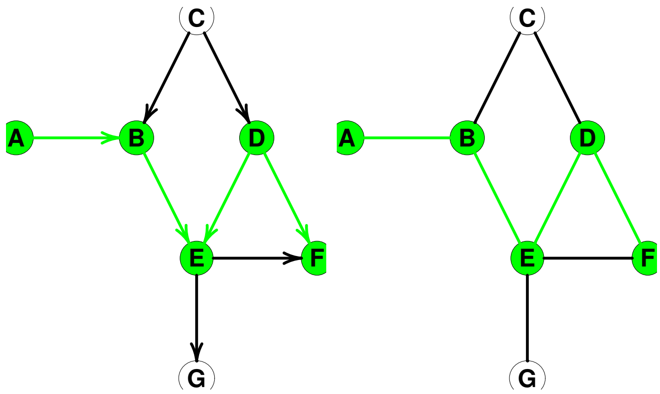


Figure 17: Chain (A, B, E, D, F) on directed graph and path on undirected skeleton.

Blocked chains

Definition 14. Let $G := (V, E)$ be a directed graph. We call a chain

$$p_1 \rightarrow p_2 \leftarrow p_3$$

a **head-to-head meeting.**

Let $Z \subseteq V$ be a subset of vertices. Then a chain $p \in G^{\blacktriangle}$ is called **blocked at position** i **by** Z, if for its subchain (p_{i-1}, p_i, p_{i+1}) there is

$$\begin{cases} p_i \in Z, & \text{if not } p_{i-1} \rightarrow p_i \leftarrow p_{i+1} \\ p_i \notin Z \cup \operatorname{anc}(Z), & \text{else} \end{cases}$$

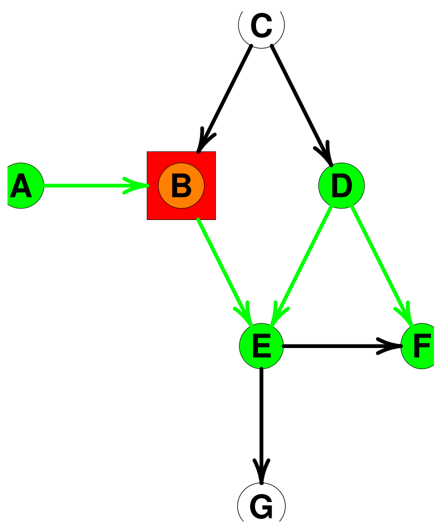


Figure 18: Chain (A, B, E, D, F) is blocked by $Z = \{B\}$ at 2.

Blocked chains / more examples

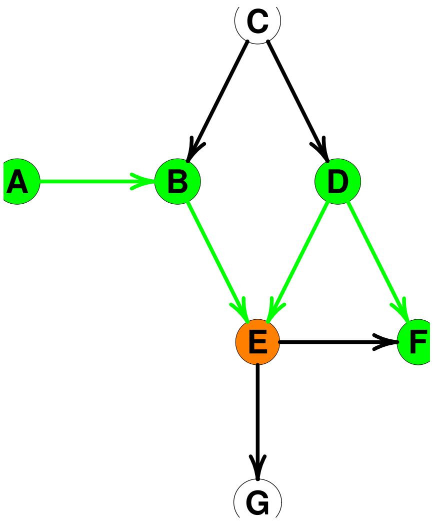


Figure 19: Chain (A, B, E, D, F) is blocked by $Z = \emptyset$ at 3.

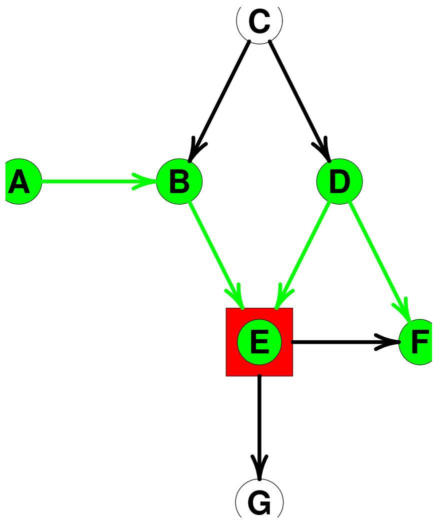
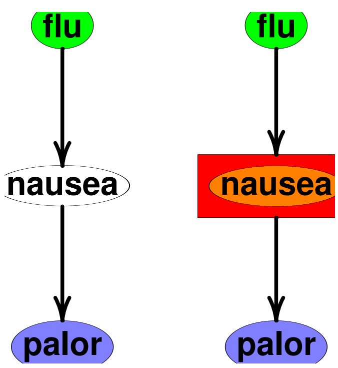


Figure 20: Chain (A, B, E, D, F) is **not** blocked by $Z = \{E\}$ at 3.

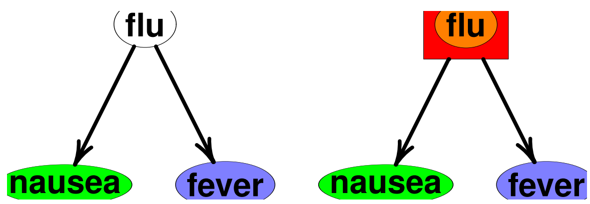
Blocked chains / rationale

The notion of blocking is choosen in a way so that chains model "flow of causal influence" through a causal network where the states of the vertices Z are already know.

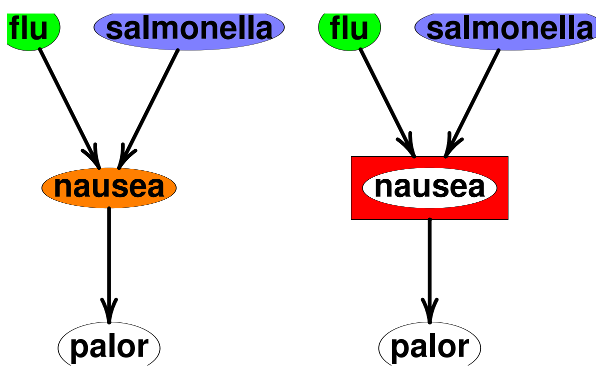
1) Serial connection / intermediate cause:



2) Diverging connection / common cause:



3) Converging connection / common effect:



Models "discounting" [Nea03, p. 51].

The moral graph

Definition 15. Let $G := (V, E)$ be a DAG.

As the **moral graph of** G we denote the undirected skeleton graph of G plus additional edges between each two parents of a vertex, i.e. $\text{moral}(G) := (V, E')$ with

$$E' := u(E) \cup \{\{x, y\} \mid \exists z \in V : x, y \in \text{pa}(z)\}$$

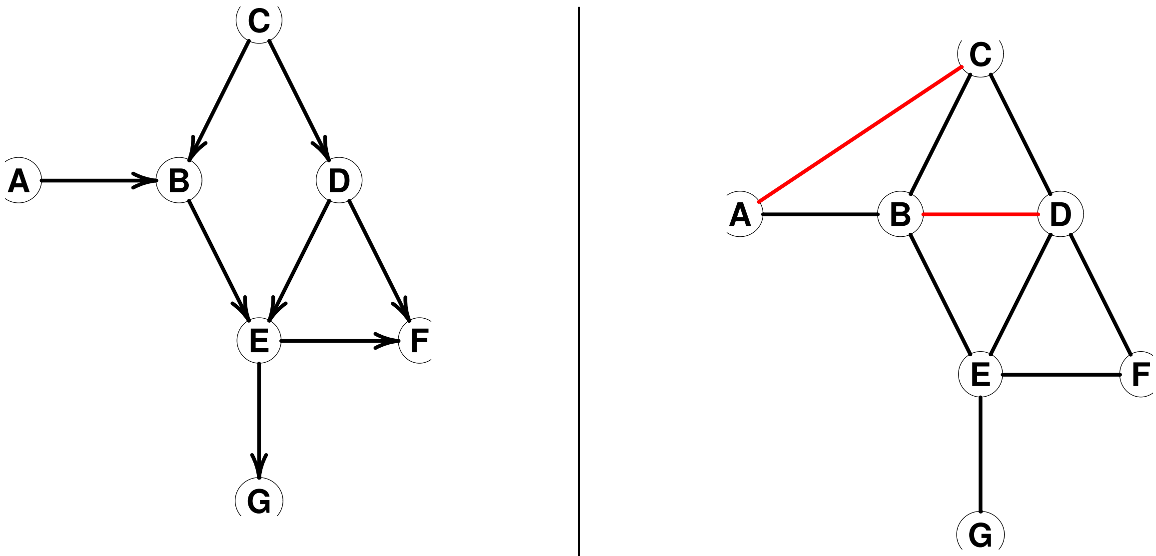


Figure 22: DAG and its moral graph.

Separation in DAGs (d-separation)

Let $G := (V, E)$ be a DAG.

Let $X, Y, Z \subseteq V$ be three disjoint subsets of vertices. We say, the vertices X and Y are **separated by** Z **in** G, if

(i) every chain from any vertex from X to any vertex from Y is blocked by Z

or equivalently

(ii) X and Y are u-separated by Z in the moral graph of the ancestral hull of $X \cup Y \cup Z$.

We write $I_G(X, Y | Z)$ for the statement, that X and Y are separated by Z in G.

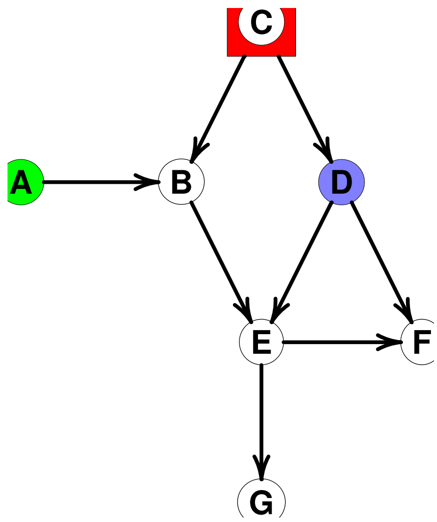


Figure 23: Are the vertices A and D separated by C in G?

Separation in DAGs (d-separation) / examples

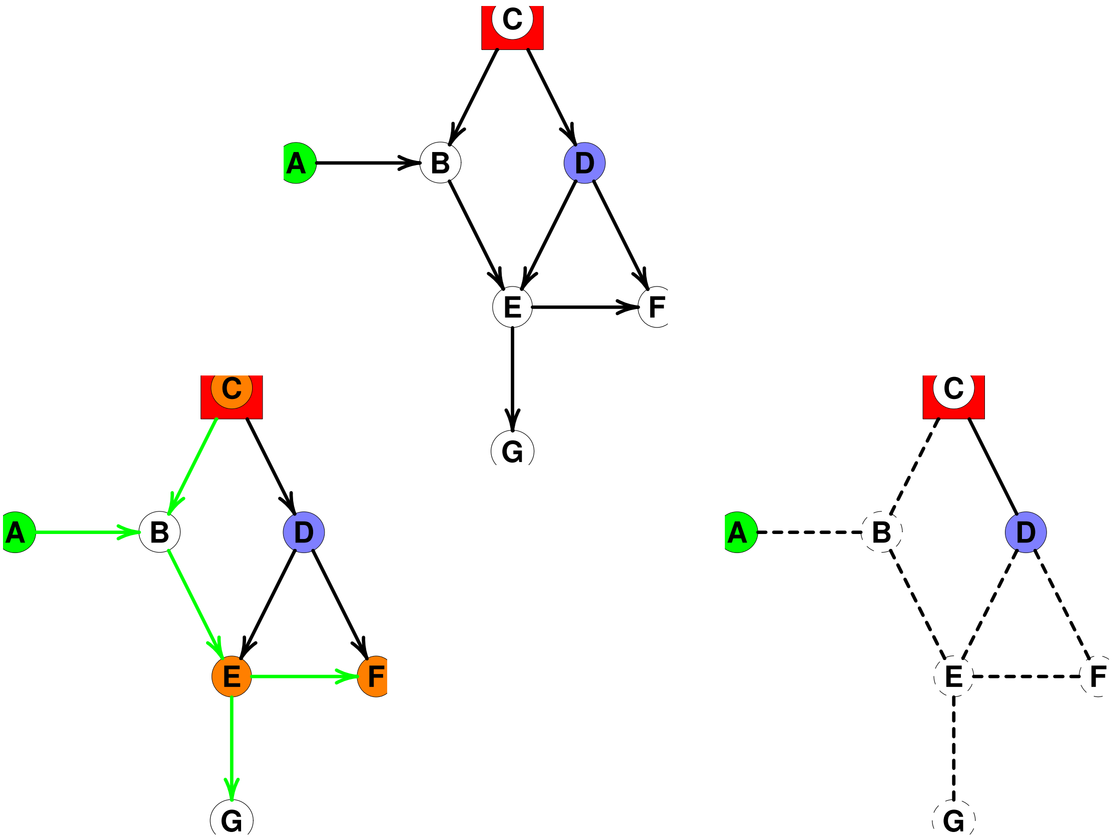


Figure 24: A and D are separated by C in G.

Separation in DAGs (d-separation) / more examples

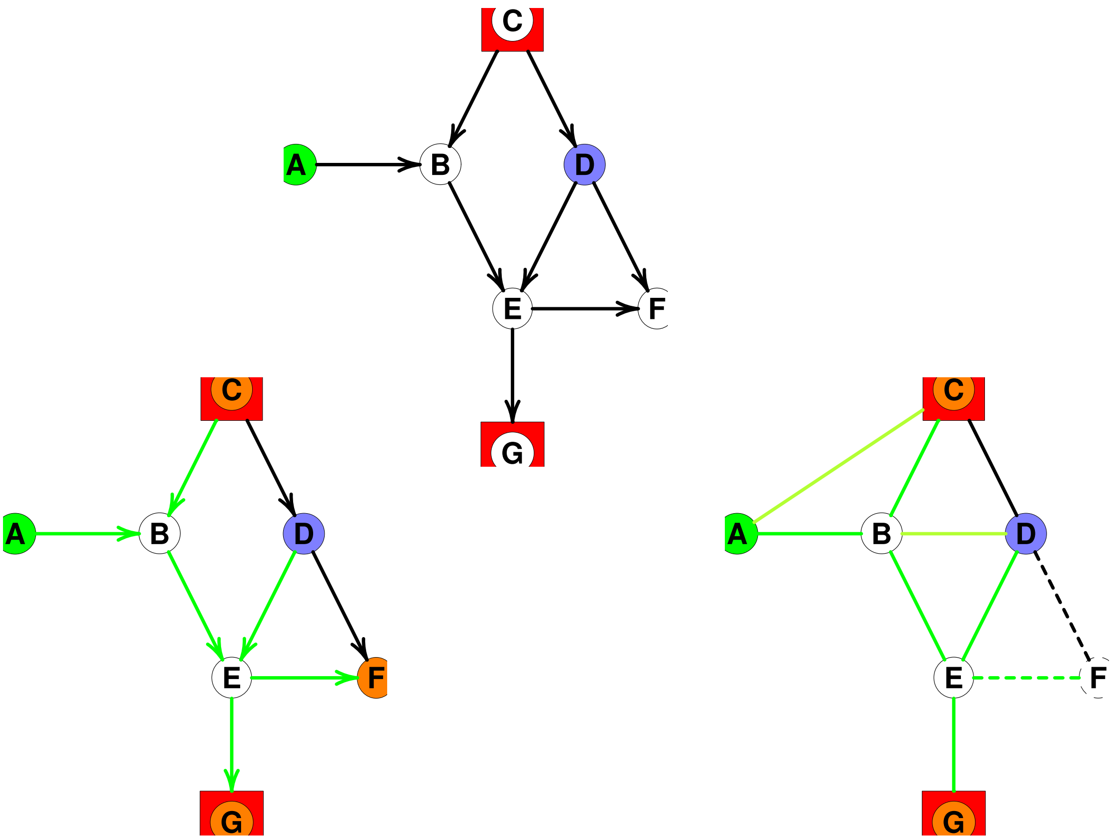


Figure 25: A and D are not separated by $\{C, G\}$ in G.

Checking d-separation

To test, if for a given graph $G = (V, E)$ two given sets $X, Y \subseteq V$ of vertices are d-separated by a third given set $Z \subseteq V$ of vertices, we may

- build the moral graph of the ancestral hull and
- apply the u-separation criterion.

```
check-d-separation(G, X, Y, Z) :
G' := moral(anc_G(X ∪ Y ∪ Z))
return check-u-separation(G', X, Y, Z)
```

Figure 26: Algorithm for checking d-separation via u-separation in the moral graph.

A drawback of this algorithm is that we have to rebuild the moral graph of the ancestral hull whenever X or Y changes.

Checking d-separation

Instead of constructing a moral graph, we can modify **a breadth-first search for chains** to find all vertices not d-separated from X by Z in G.

The breadth-first search must not hop over head-to-head meetings with the middle vertex not in Z nor having an descendent in Z.

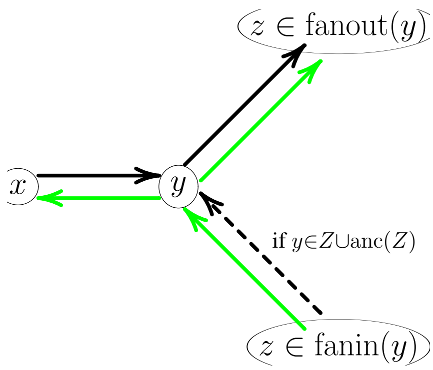


Figure 27: Restricted breadth-first search of non-blocked chains.

```
enumerate-d-separation(G = (V, E), X, Z) :
borderForward := ∅
borderBackward := X \ Z
reached := ∅
while borderForward ≠ ∅ or borderBackward ≠ ∅ do
    reached := reached ∪ (borderForward \ Z) ∪ borderBackward
    borderForward := fanout_G(borderBackward ∪ (borderForward \ Z)) \ reached
    borderBackward := fanin_G(borderBackward ∪ (borderForward ∩ (Z ∪ anc(Z)))) \ Z \ reached
od
return V \ reached
```

Figure 28: Algorithm for enumerating all vertices d-separated from X by Z in G via restricted breadth-first search (see [Nea03, p. 80–86] for another formulation).

Properties of d-separation / no strong union

For d-separation the strong union property does not hold.

I is called **strongly unionable**, if

$$I(X, Y|Z) \Rightarrow I(X, Y|Z \cup Z') \quad \text{for all } Z' \text{ disjunct with } X, Y$$

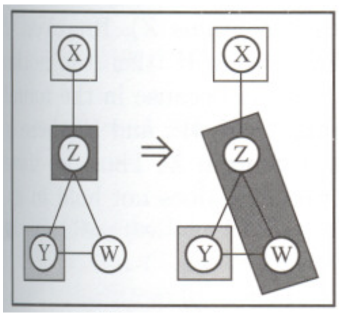


Figure 29: Example for strong union in undirected graphs (u-separation) [CGH97, p. 189].

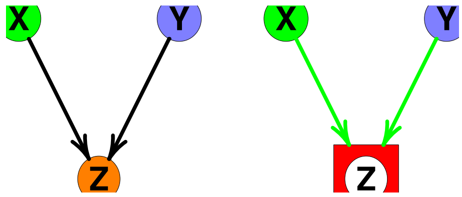


Figure 30: Counterexample for strong unions in DAGs (d-separation).

Properties of d-separation / no strong transitivity

For d-separation the strong transitivity property does not hold.

I is called **strongly transitive**, if

$$I(X, Y|Z) \Rightarrow I(X, \{v\}|Z) \text{ or } I(\{v\}, Y|Z) \quad \forall v \in V \setminus Z$$

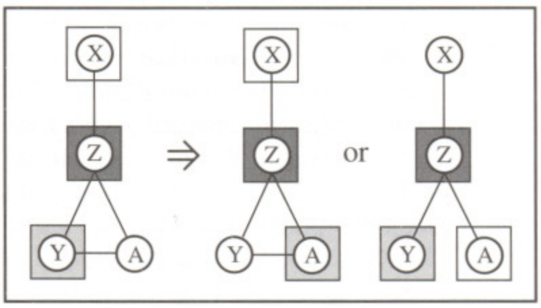


Figure 31: Example for strong transitivity in undirected graphs (u-separation) [CGH97, p. 189].

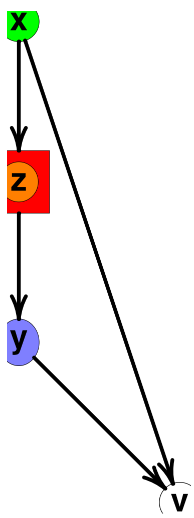


Figure 32: Counterexample for strong transitivity in DAGs (d-separation).

Properties of d-separation

relation	symmetry	decomposition	composition	strong union	weak union	contraction	intersection	strong transitivity	weak transitivity	chordality
u-separation	+	+	+	+	+	+	+	+	+	−
d-separation	+	+	+	−	+		+	−	+	+

References

[CGH97] Enrique Castillo, José Manuel Gutiérrez, and Ali S. Hadi. *Expert Systems and Probabilistic Network Models*. Springer, New York, 1997.

[CLR90] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, 1990.

[Nea03] Richard E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2003.

[OW02] Thomas Ottmann and Peter Widmayer. *Algorithmen und Datenstrukturen*. Spektrum Verlag, Heidelberg, 2002.

[VP90] Thomas Verma and Judea Pearl. Causal networks: semantics and expressiveness. In Ross D. Shachter, Tod S. Levitt, Laveen N. Kanal, and John F. Lemmer, editors, *Uncertainty in Artificial Intelligence 4*, pages 69–76. North-Holland, Amsterdam, 1990.