Bayesian Networks and Decision Graphs

Chapter 7

Learning the structure of a Bayesian network

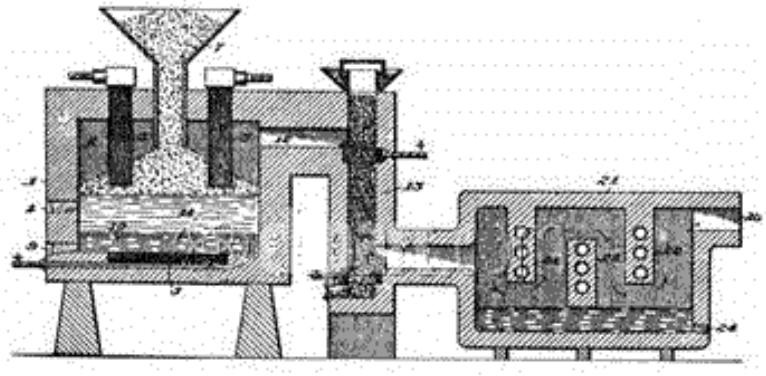
We have:

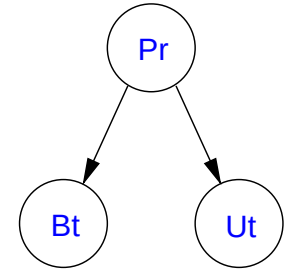
- A complete database of cases over a set of variables.

We want:

- A Bayesian network structure representing the independence properties in the database.

Cases	Pr	Bt	Ut
1.	yes	pos	pos
2.	yes	neg	pos
3.	yes	pos	neg
4.	yes	pos	neg
5.	no	neg	neg



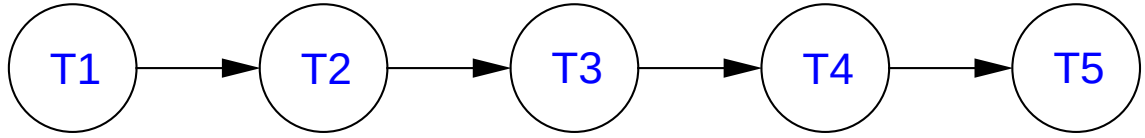


Bayesian networks from databases I

Example: Transmission of symbol strings, $P_{\mathcal{D}}(\mathcal{U})$:

		Last 3							
		aaa	aab	aba	abb	baa	bab	bba	bbb
First 2	aa	0.017	0.021	0.019	0.019	0.045	0.068	0.045	0.068
	ab	0.033	0.040	0.037	0.038	0.011	0.016	0.010	0.015
	ba	0.011	0.014	0.010	0.010	0.031	0.046	0.031	0.045
	bb	0.050	0.060	0.057	0.057	0.016	0.023	0.015	0.023

Consider the model N as representing the database:



- Simpler than the database.
- A representation of the database.

The chain rule (yields the joint probability which we can compare to the actual database):

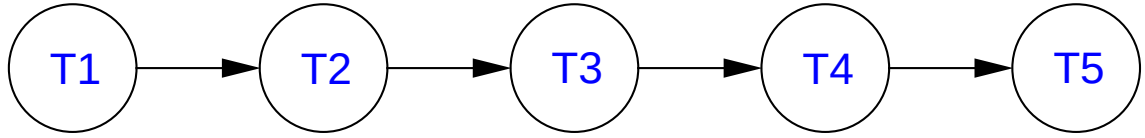
$$P_N(\mathcal{U}) = P(T_1, T_2, T_3, T_4, T_5) = P(T_1)P(T_2|T_1)P(T_3|T_2)P(T_4|T_3)P(T_5|T_4).$$

Bayesian networks from databases II

Example: Transmission of symbol strings, $P_{\mathcal{D}}(\mathcal{U})$:

First 2 \ Last 3	aaa	aab	aba	abb	baa	bab	bba	bbb
aa	0.017	0.021	0.019	0.019	0.045	0.068	0.045	0.068
ab	0.033	0.040	0.037	0.038	0.011	0.016	0.010	0.015
ba	0.011	0.014	0.010	0.010	0.031	0.046	0.031	0.045
bb	0.050	0.060	0.057	0.057	0.016	0.023	0.015	0.023

Consider the model N as representing the database:



First 2 \ Last 3	aaa	aab	aba	abb	baa	bab	bba	bbb
aa	0.016	0.023	0.018	0.021	0.044	0.067	0.050	0.061
ab	0.030	0.044	0.033	0.041	0.011	0.015	0.012	0.014
ba	0.010	0.016	0.012	0.014	0.029	0.045	0.033	0.041
bb	0.044	0.067	0.059	0.061	0.016	0.023	0.017	0.021

Are $P_N(\mathcal{U})$ and $P_{\mathcal{D}}(\mathcal{U})$ sufficiently identical?

A (naïve) way to look at it

Some agent produces samples $\mathcal{D}$ of cases from a Bayesian network M over the universe $\mathcal{U}$.

- ➤ These cases are handed over to you, and you should now reconstruct M from the cases.

Assumptions:

- ➤ The sample is fair ($P_{\mathcal{D}}(\mathcal{U})$ reflects the distribution determined by M).
- ➤ All links in M are essential.

A naïve procedure:

- For each Bayesian network structure N:
  - Calculate the distance between $P_N(\mathcal{U})$ and $P_{\mathcal{D}}(\mathcal{U})$.
- Return the network N that minimizes the distance, and where all links are essential.

But this is hardly feasible!

The space of network structures is huge!

The number of DAG structures (as a function of the number of nodes):

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} \frac{n!}{(n-i)!i!} 2^{i(n-i)} f(n-i).$$

Some example calculations:

Nodes	Number of DAGs	Nodes	Number of DAGs
1	1	13	$1.9 \cdot 10^{31}$
2	3	14	$1.4 \cdot 10^{36}$
3	25	15	$2.4 \cdot 10^{41}$
4	543	16	$8.4 \cdot 10^{46}$
5	29281	17	$6.3 \cdot 10^{52}$
6	$3.8 \cdot 10^{6}$	18	$9.9 \cdot 10^{58}$
7	$1.1 \cdot 10^{9}$	19	$3.3 \cdot 10^{65}$
8	$7.8 \cdot 10^{11}$	20	$2.35 \cdot 10^{72}$
9	$1.2 \cdot 10^{15}$	21	$3.5 \cdot 10^{79}$
10	$4.2 \cdot 10^{18}$	22	$1.1 \cdot 10^{87}$
11	$3.2 \cdot 10^{22}$	23	$7.0 \cdot 10^{94}$
12	$5.2 \cdot 10^{26}$	24	$9.4 \cdot 10^{102}$

Two approaches to structural learning

Score based learning:

- Produces a series of candidate structures.
- Returns the structure with highest score.

Constraint based learning:

- Establishes a set of conditional independence statements for the data.
- Builds a structure with d-separation properties corresponding to the independence statements found.

Constraint based learning

Some notation:

- To denote that A is conditionally independent of B given $\mathcal{X}$ in the database we shall use

$$I(A, B, \mathcal{X}).$$

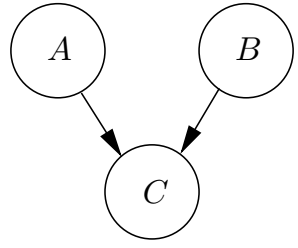
Some assumptions:

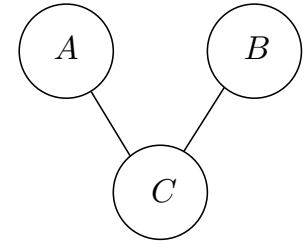
- The database is a faithful sample from a Bayesian network M: A and B are d-separated given $\mathcal{X}$ in M if and only if $I(A, B, \mathcal{X})$.
- We have an oracle that correctly answers questions of the type:

"Is $I(A, B, \mathcal{X})$?"

The algorithm: Use the oracle's answers to first establish a skeleton of a Bayesian network:

- The skeleton is the undirected graph obtained by removing directions on the arcs.





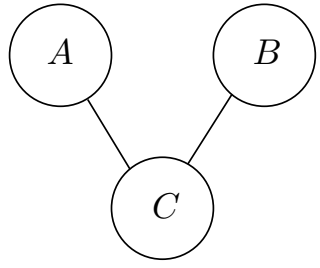
Next, when the skeleton is found we then start looking for the directions on the arcs.

Finding the skeleton I

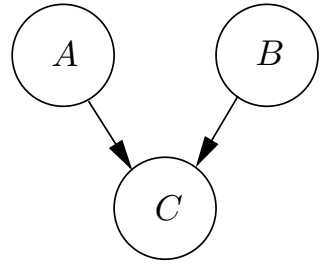
The idea: if there is a link between A and B in M then they cannot be d-separated, and as the data is faithful it can be checked by asking questions to the oracle:

- The link $A - B$ is part of the skeleton if and only if $\neg I(A, B, \mathcal{X})$, for all $\mathcal{X}$.

Assume that the only conditional independence found is $I(A, B)$:

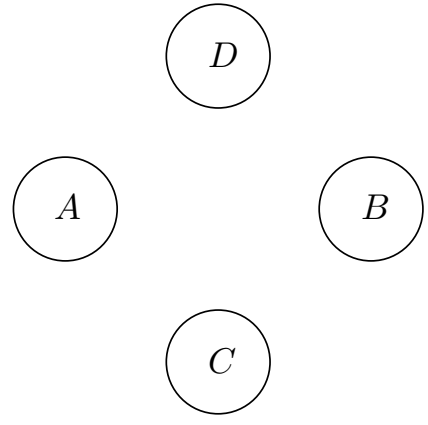


The skeleton



The only possible DAG

Assume that the conditional independences found are $I(A, B, D)$ and $I(C, D, \{A, B\})$:

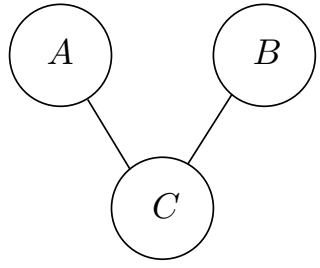


Finding the skeleton II

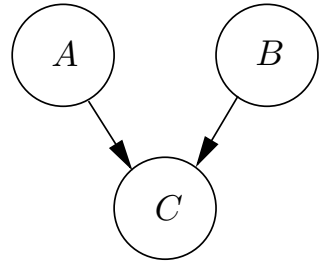
The idea: if there is a link between A and B in M then they cannot be d-separated, and as the data is faithful it can be checked by asking questions to the oracle:

- The link $A - B$ is part of the skeleton if and only if $\neg I(A, B, \mathcal{X})$, for all $\mathcal{X}$.

Assume that the only conditional independence found is $I(A, B)$:

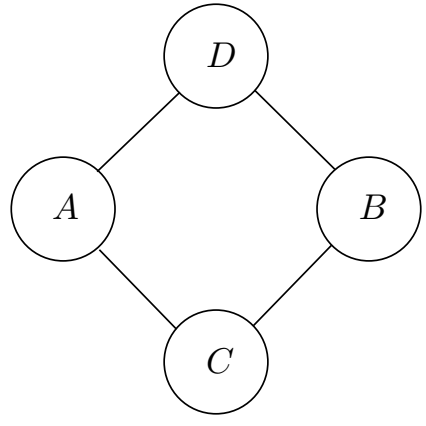


The skeleton



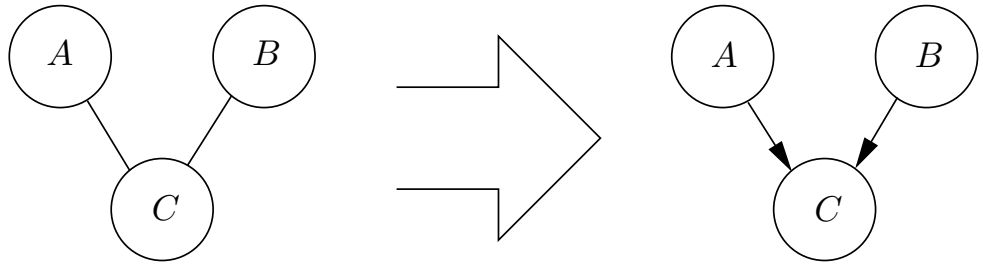
The only possible DAG

Assume that the conditional independences found are $I(A, B, D)$ and $I(C, D, \{A, B\})$:

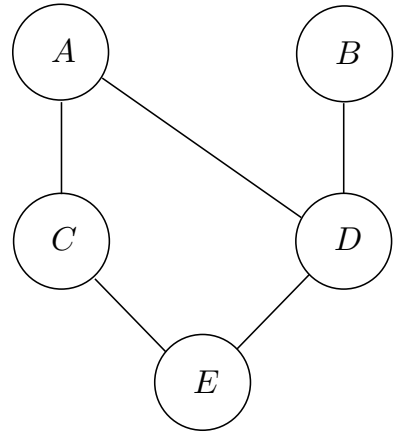


Setting the directions on the links I

Rule 1: If you have three nodes, A, B, C such that $A - C$ and $B - C$, but not $A - B$, then introduce the v-structure $A \rightarrow C \leftarrow B$ if there exists an $\mathcal{X}$ (possibly empty) such that $I(A, B, \mathcal{X})$ and $C \notin \mathcal{X}$.

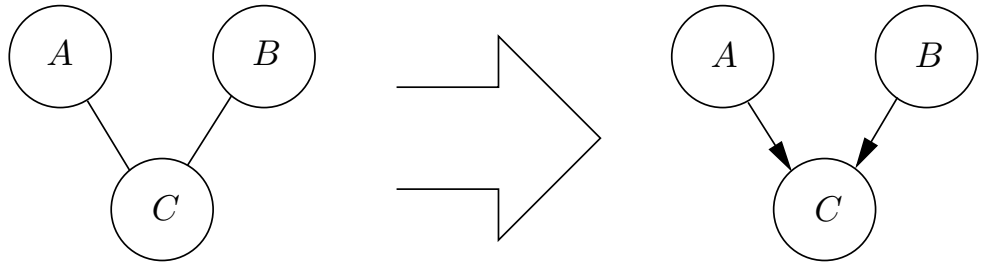


Example: Assume that we get the independences $I(A, B)$, $I(B, C)$, $I(A, B, C)$, $I(B, C, A)$, $I(C, D, A)$, $I(B, C, \{D, A\})$, $I(C, D, \{A, B\})$, $I(B, E, \{C, D\})$, $I(A, E, \{C, D\})$, $I(B, C, \{A, D, E\})$, $I(A, E, \{B, C, D\})$, $I(B, E, \{A, C, D\})$.

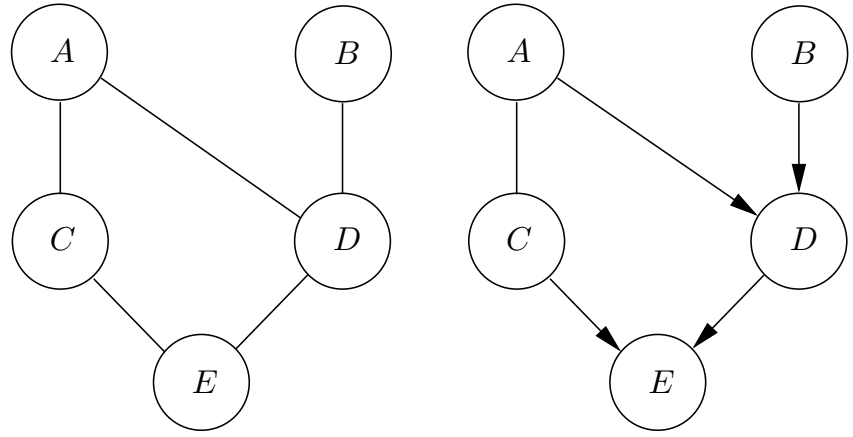


Setting the directions on the links I

Rule 1: If you have three nodes, A, B, C such that $A - C$ and $B - C$, but not $A - B$, then introduce the v-structure $A \rightarrow C \leftarrow B$ if there exists an $\mathcal{X}$ (possibly empty) such that $I(A, B, \mathcal{X})$ and $C \notin \mathcal{X}$.



Example: Assume that we get the independences $I(A, B)$, $I(B, C)$, $I(A, B, C)$, $I(B, C, A)$, $I(C, D, A)$, $I(B, C, \{D, A\})$, $I(C, D, \{A, B\})$, $I(B, E, \{C, D\})$, $I(A, E, \{C, D\})$, $I(B, C, \{A, D, E\})$, $I(A, E, \{B, C, D\})$, $I(B, E, \{A, C, D\})$.



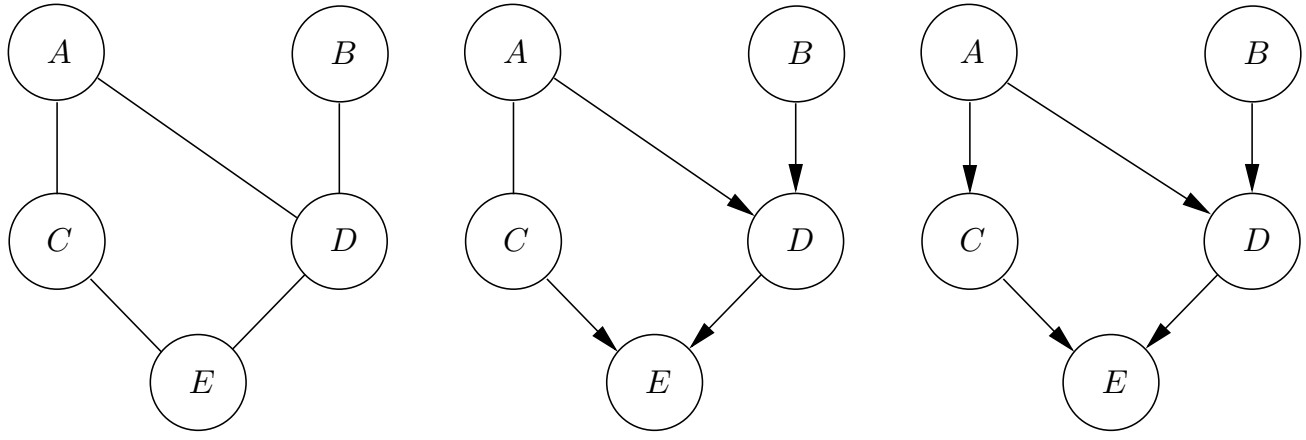
Setting the directions on the links II

Rule 2 [Avoid new v-structures]: When Rule 1 has been exhausted, and you have $A \rightarrow C - B$ (and no link between A and B), then direct $C \rightarrow B$.

Rule 3 [Avoid cycles]: If $A \rightarrow B$ introduces a directed cycle in the graph, then do $A \leftarrow B$

Rule 4 [Choose randomly]: If none of the rules 1-3 can be applied anywhere in the graph, choose an undirected link and give it an arbitrary direction.

Example:



Skeleton Rule 1 Rule 4

The rules: an overview

Rule 1 [Find v-structures]: If you have three nodes, A, B, C such that $A - C$ and $B - C$, but not $A - B$, then introduce the v-structure $A \rightarrow C \leftarrow B$ if there exists an $\mathcal{X}$ (possibly empty) such that $I(A, B, \mathcal{X})$ and $C \notin \mathcal{X}$.

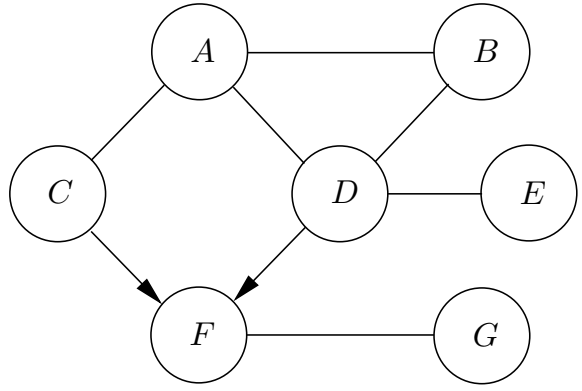
Rule 2 [Avoid new v-structures]: When Rule 1 has been exhausted, and you have $A \rightarrow C - B$ (and no link between A and B), then direct $C \rightarrow B$.

Rule 3 [Avoid cycles]: If $A \rightarrow B$ introduces a directed cycle in the graph, then do $A \leftarrow B$

Rule 4 [Choose randomly]: If none of the rules 1-3 can be applied anywhere in the graph, choose an undirected link and give it an arbitrary direction.

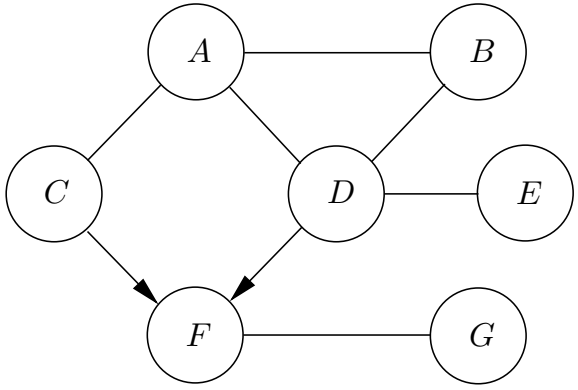
Another example

Consider the graph:



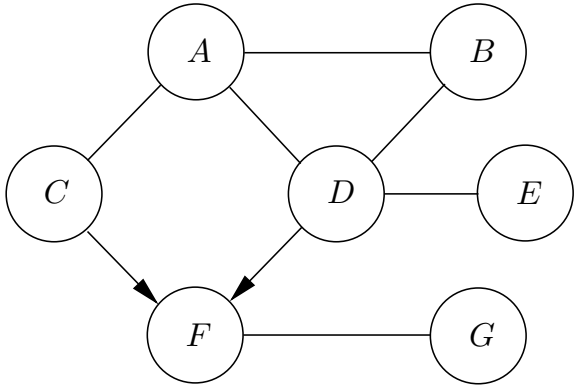
Apply the four rules to learn a Bayesian network structure

Another example I

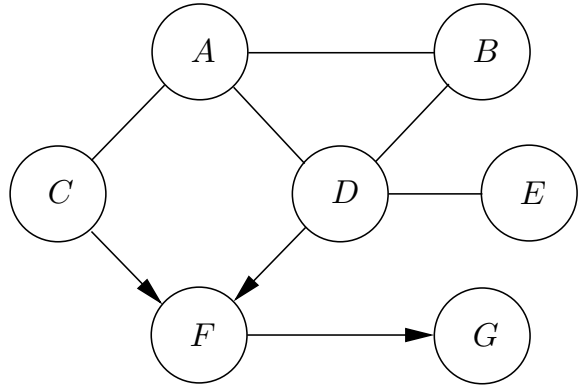


Step 1: Rule 1

Another example I

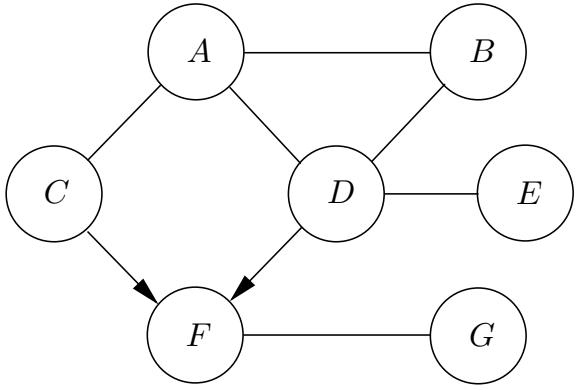


Step 1: Rule 1

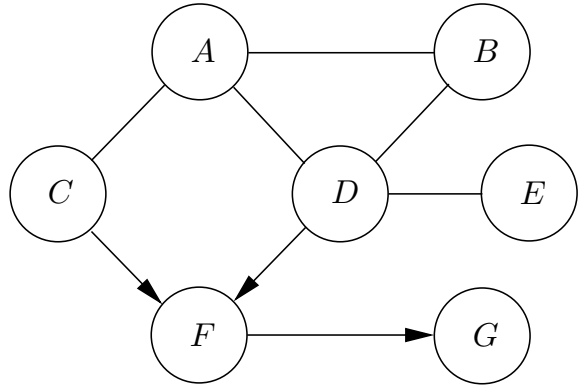


Step 2: Rule 2

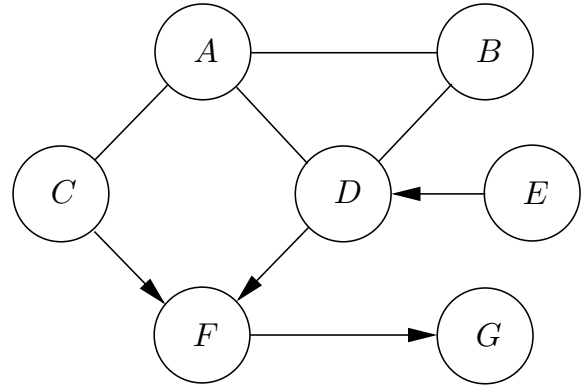
Another example I



Step 1: Rule 1

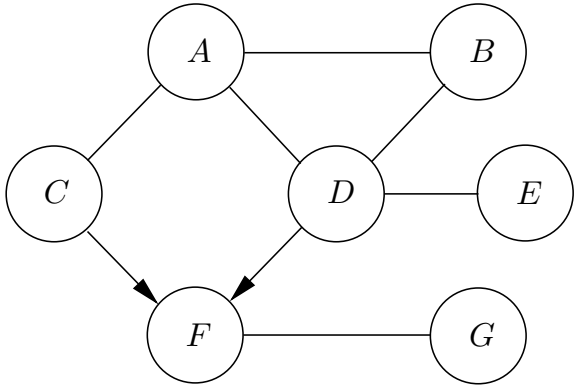


Step 2: Rule 2

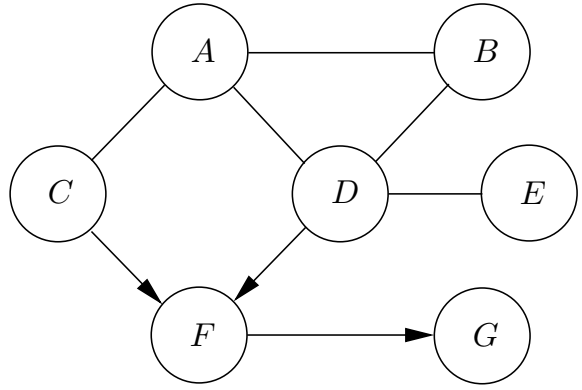


Step 3: Rule 4

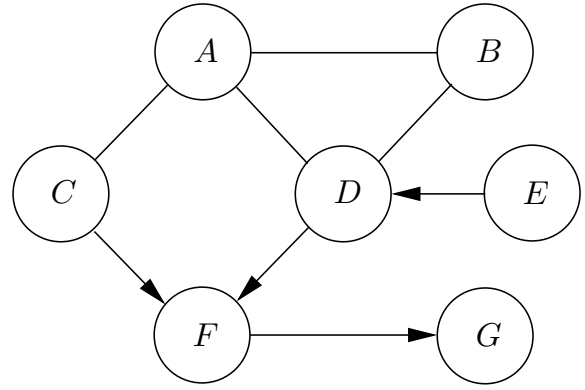
Another example I



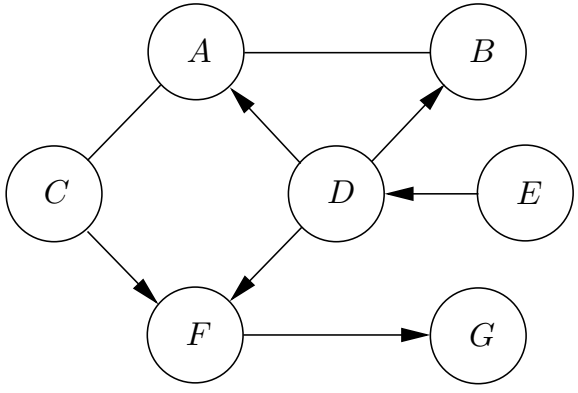
Step 1: Rule 1



Step 2: Rule 2

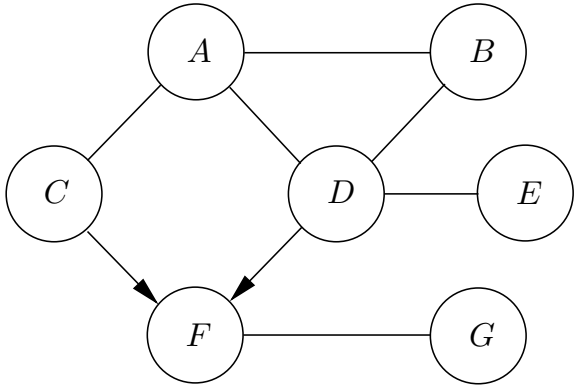


Step 3: Rule 4

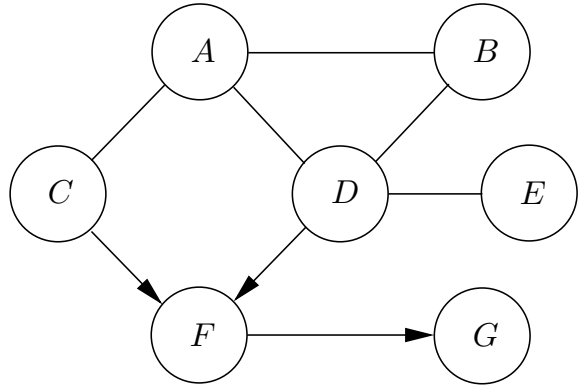


Step 4: Rule 2

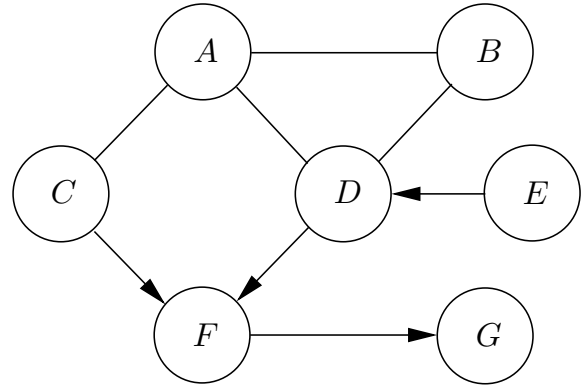
Another example I



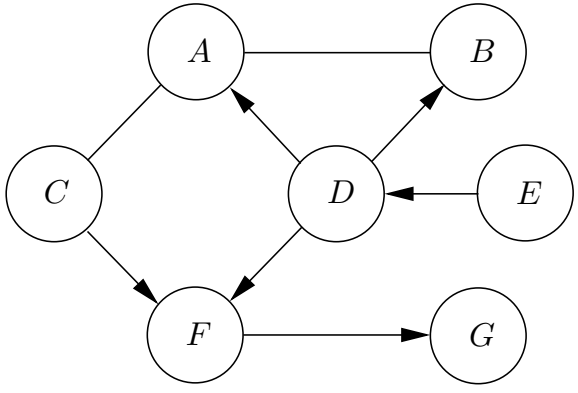
Step 1: Rule 1



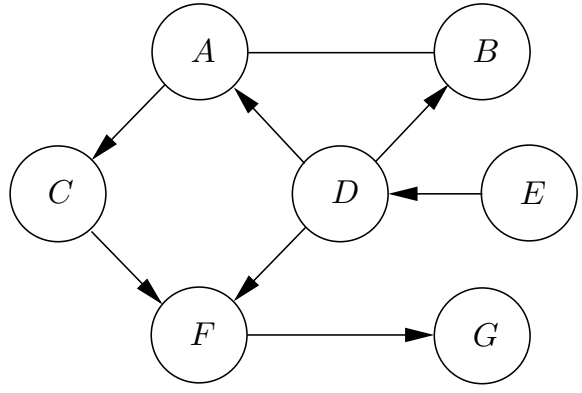
Step 2: Rule 2



Step 3: Rule 4

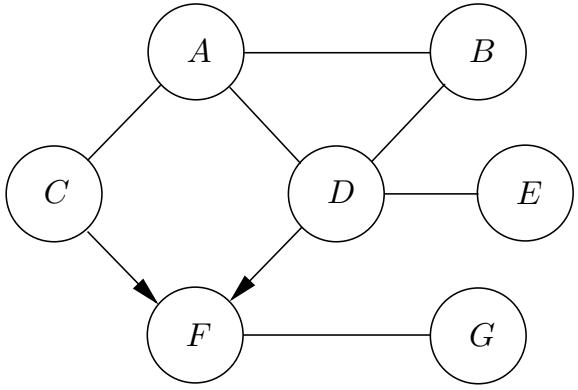


Step 4: Rule 2

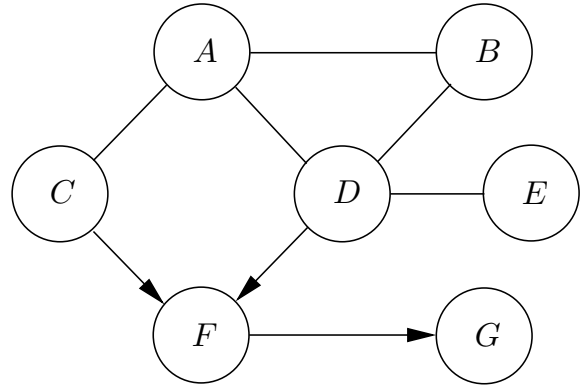


Step 5: Rule 2

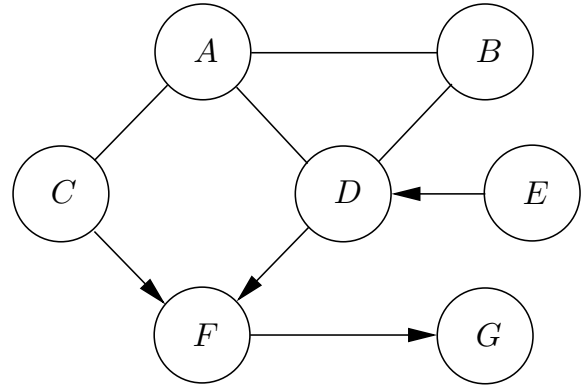
Another example I



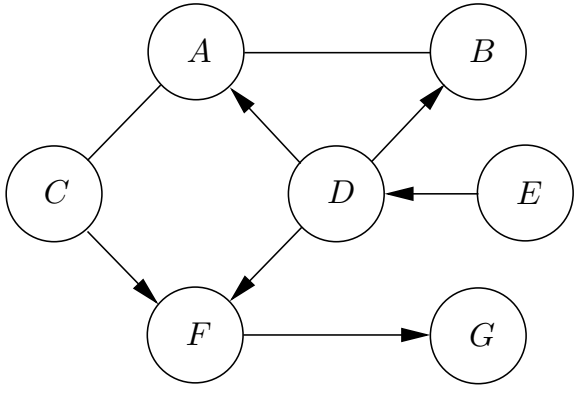
Step 1: Rule 1



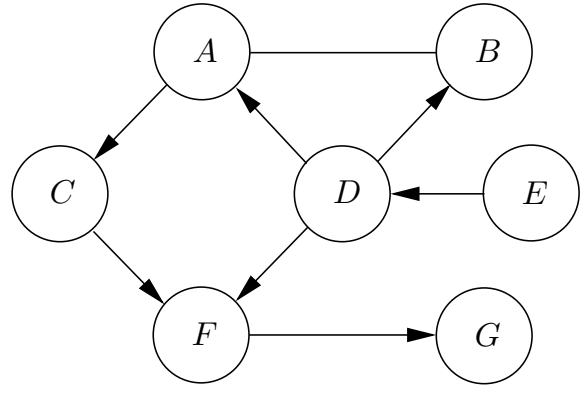
Step 2: Rule 2



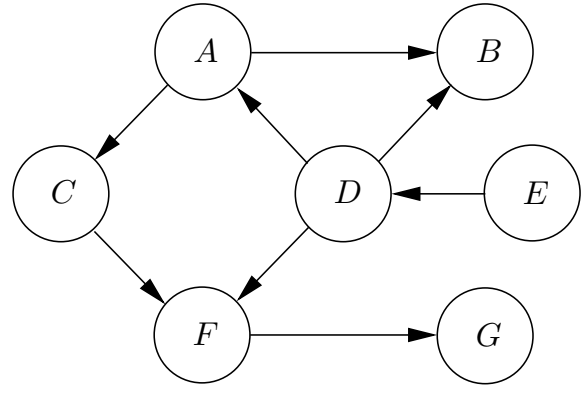
Step 3: Rule 4



Step 4: Rule 2



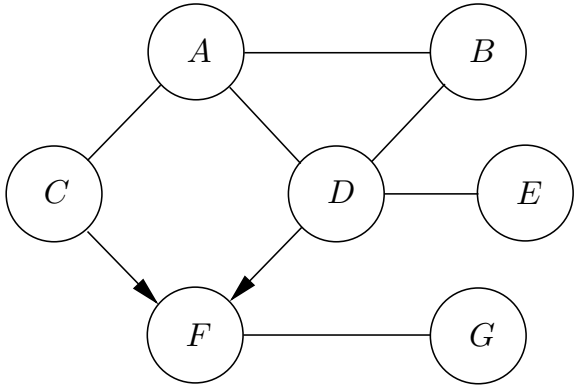
Step 5: Rule 2



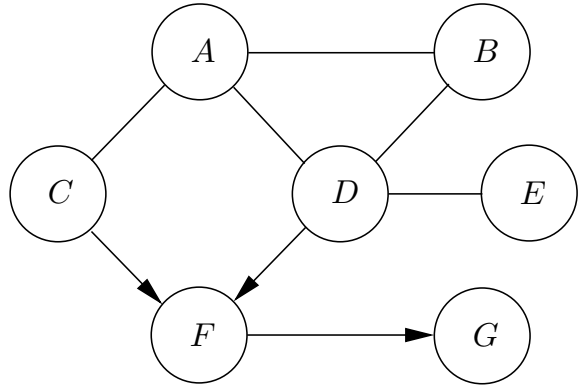
Step 6: Rule 4

However, we are not guaranteed a unique solution!

Another example II

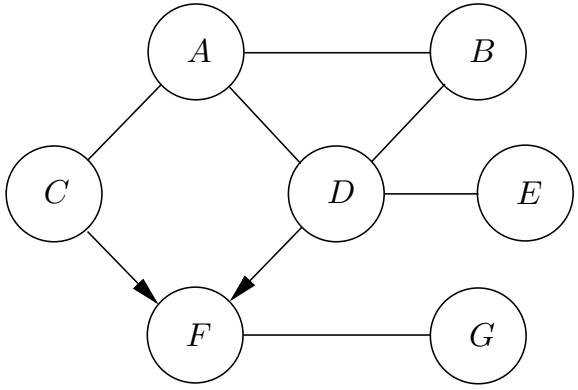


Step 1: Rule 1

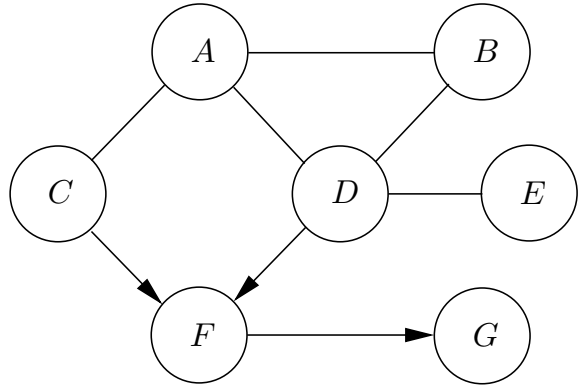


Step 2: Rule 2

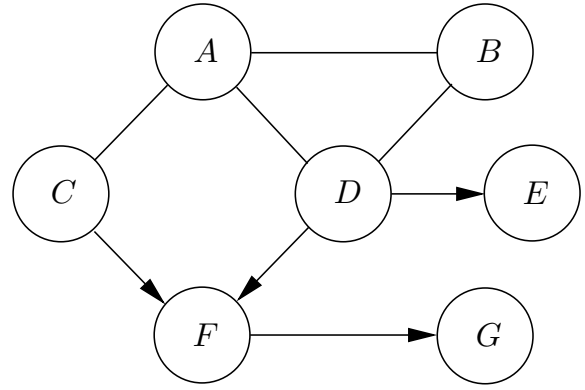
Another example II



Step 1: Rule 1

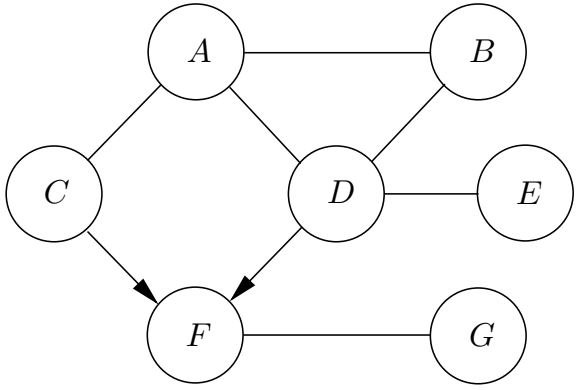


Step 2: Rule 2

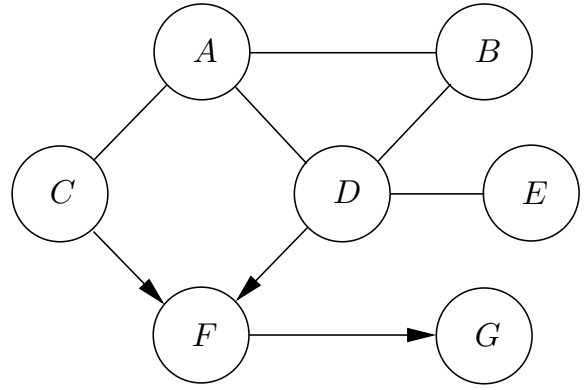


Step 3: Rule 4

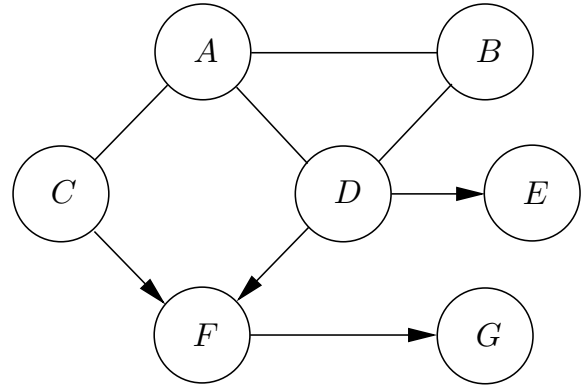
Another example II



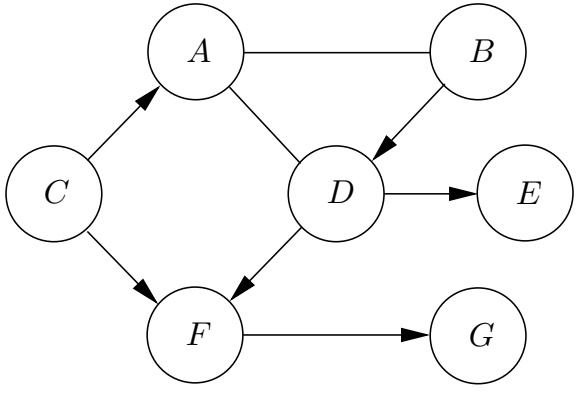
Step 1: Rule 1



Step 2: Rule 2

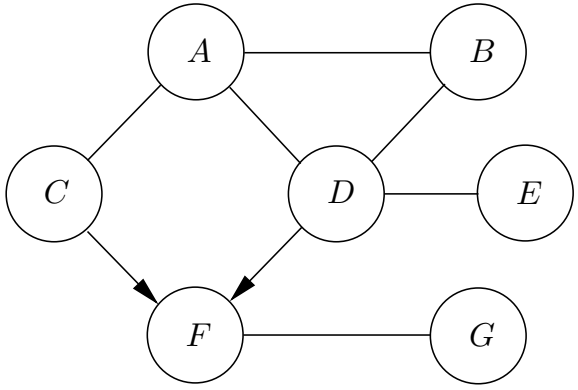


Step 3: Rule 4

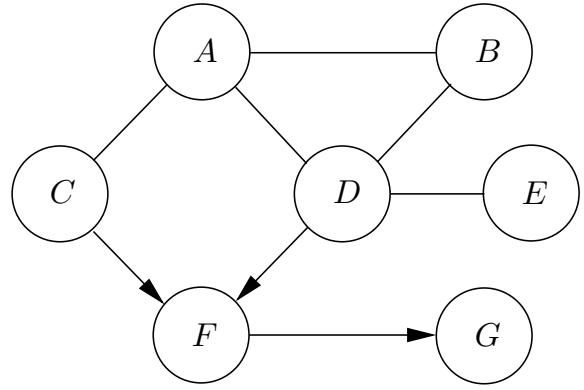


Step 4: Rule 4

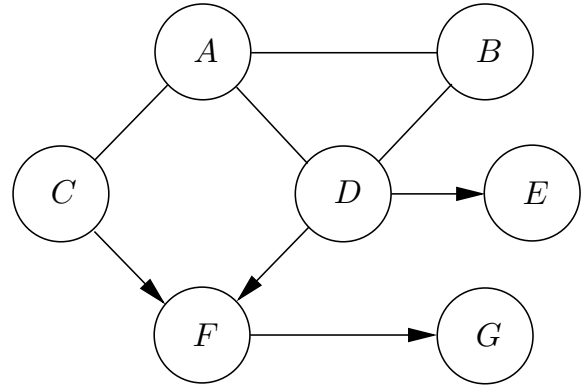
Another example II



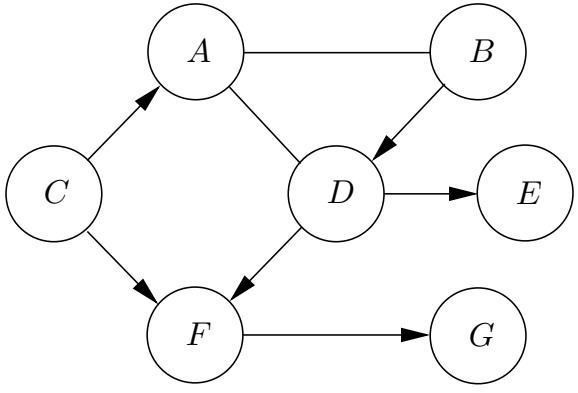
Step 1: Rule 1



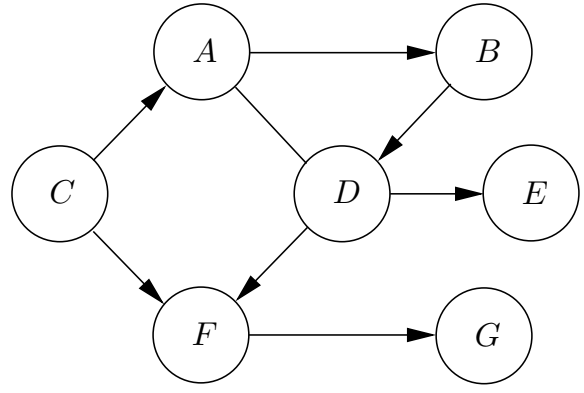
Step 2: Rule 2



Step 3: Rule 4

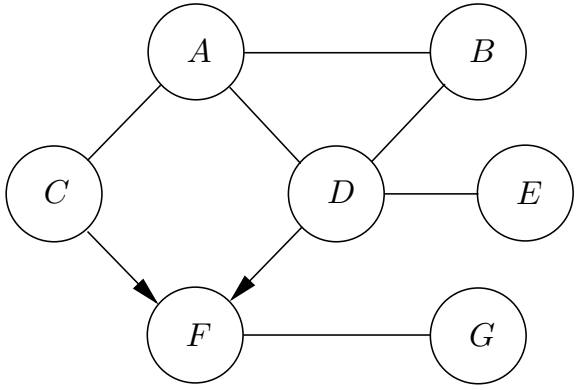


Step 4: Rule 4

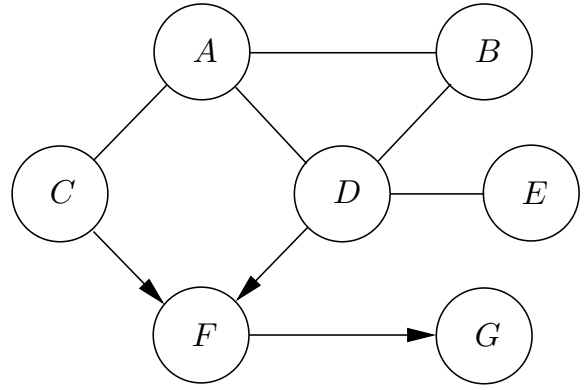


Step 5: Rule 2

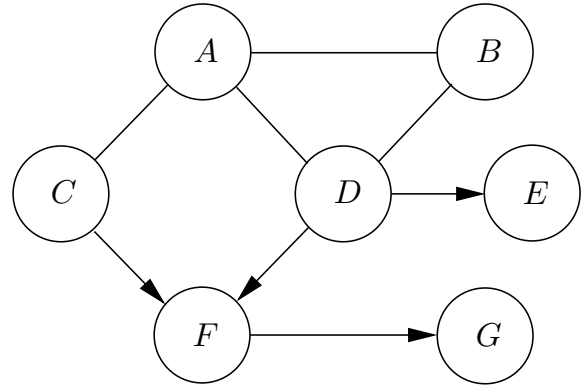
Another example II



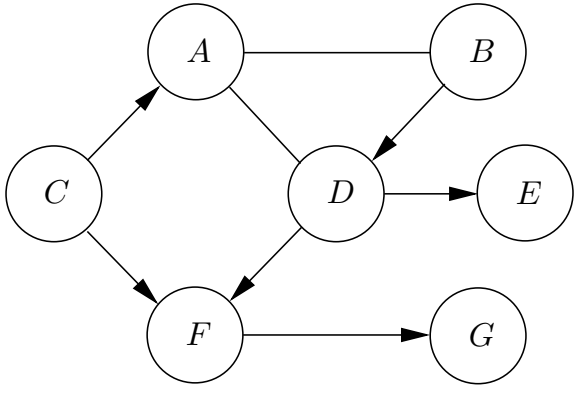
Step 1: Rule 1



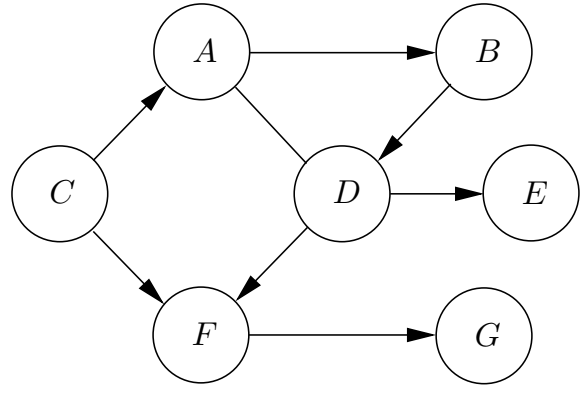
Step 2: Rule 2



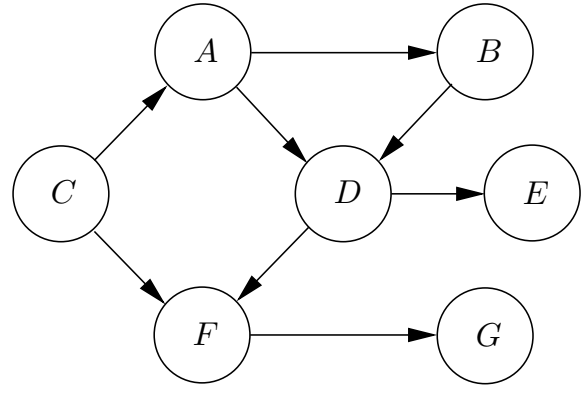
Step 3: Rule 4



Step 4: Rule 4



Step 5: Rule 2



Step 6: Rule 2+3

Although the solution is not necessarily unique, all solutions have the same d-separation properties!

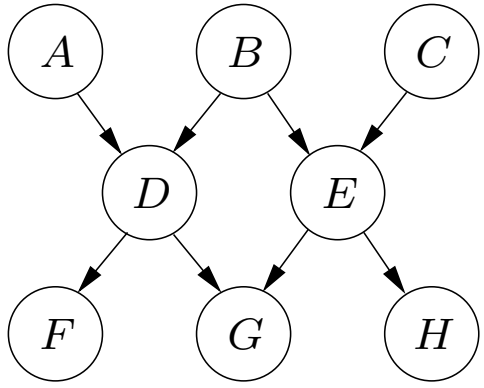
From independence tests to skeleton

Until now, we have assumed that all questions of the form "Is $I(A, B, \mathcal{X})$?" can be answered (allowing us to establish the skeleton). However, questions come at a price, and we would there like to ask as few questions as possible.

To reduce the number of questions we exploit the following property:

Theorem: The nodes A and B are not linked if and only if $I(A, B, \mathrm{pa}(A))$ or $I(A, B, \mathrm{pa}(B))$.

It is sufficient to ask questions $I(A, B, \mathcal{X})$, where $\mathcal{X}$ is a subset of A's or B's neighbors.



An active path from A to B must go through a parent of B.

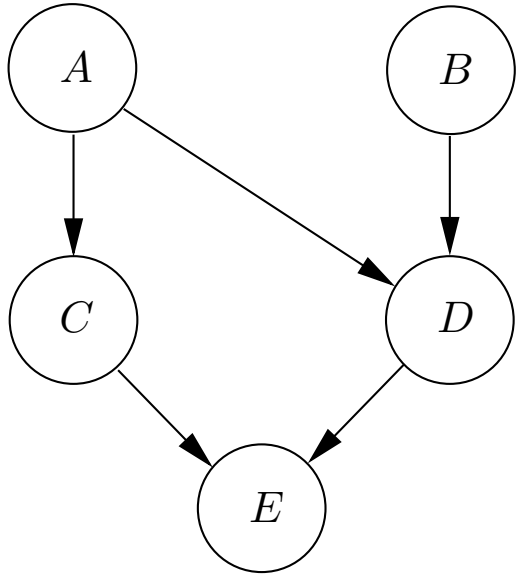
The PC algorithm

The PC algorithm:

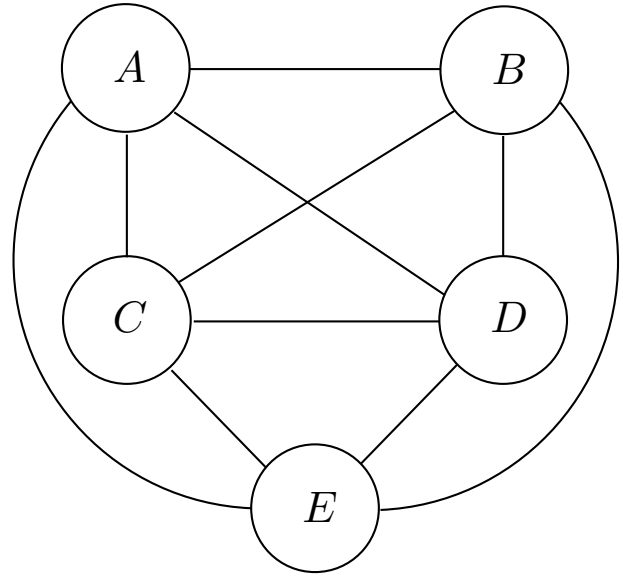
1. Start with the complete graph;
2. $i := 0$;
3. **while** a node has at least $i + 1$ neighbors
   - **for all** nodes A with at least $i + 1$ neighbors
     - **for all** neighbors B of A
       - **for all** neighbor sets $\mathcal{X}$ such that $|\mathcal{X}| = i$ and $\mathcal{X} \subseteq (\mathrm{nb}(A) \setminus \{B\})$
         - **if** $I(A, B, \mathcal{X})$ **then** remove the link $A - B$ and store "$I(A, B, \mathcal{X})$"
   - $i := i + 1$

Example

We start with the complete graph and ask the questions $I(A, B)$?, $I(A, C)$?, $I(A, D)$?, $I(A, E)$?, $I(B, C)$?, $I(B, D)$?, $I(B, E)$?, $I(C, D)$?, $I(C, E)$?, $I(D, E)$?



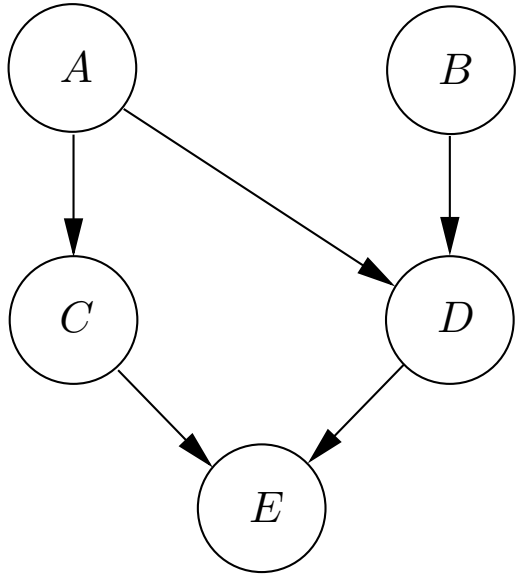
The original model



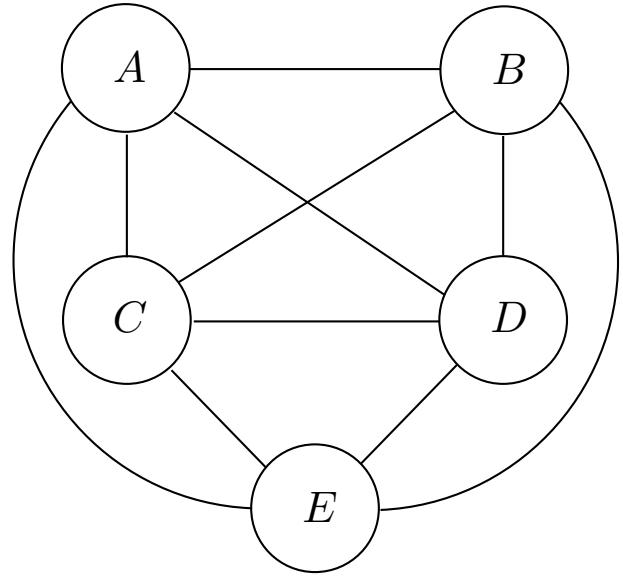
The complete graph

Example

We start with the complete graph and ask the questions $I(A, B)$?, $I(A, C)$?, $I(A, D)$?, $I(A, E)$?, $I(B, C)$?, $I(B, D)$?, $I(B, E)$?, $I(C, D)$?, $I(C, E)$?, $I(D, E)$?.



The original model



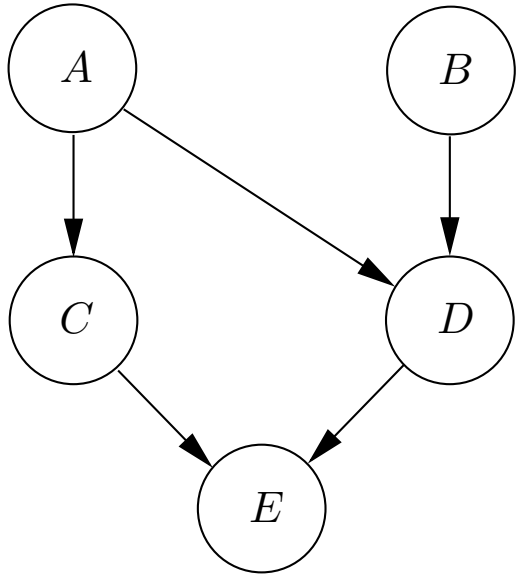
The complete graph

We get a “yes” for $I(A, B)$? and $I(B, C)$?:

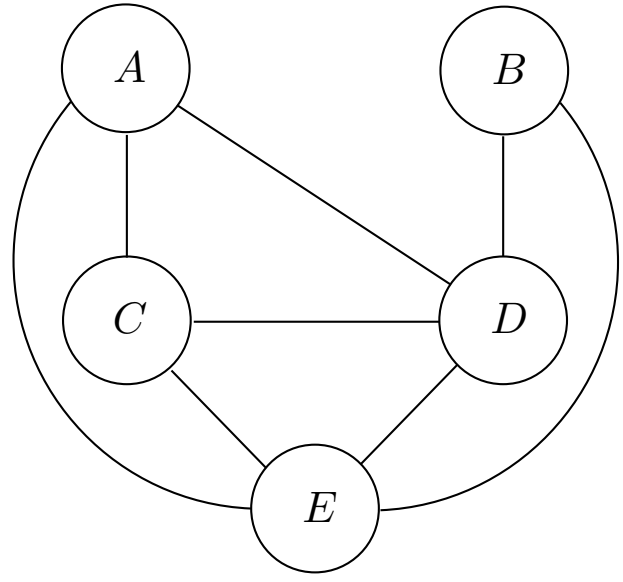
- the links $A - B$ and $B - C$ are therefore removed.

Example

We now condition on one variable and ask the questions $I(A, C, E)?$, $I(A, E, C)?$, $I(B, C, D)?$, $I(B, C, E)?$, $I(B, D, C)?$, $I(B, D, E)?$, $I(B, E, C)?$, $I(B, E, D)?$, $I(C, B, A)?$, $\ldots$,$I(C, D, A)?$, $I(C, D, B)?$.



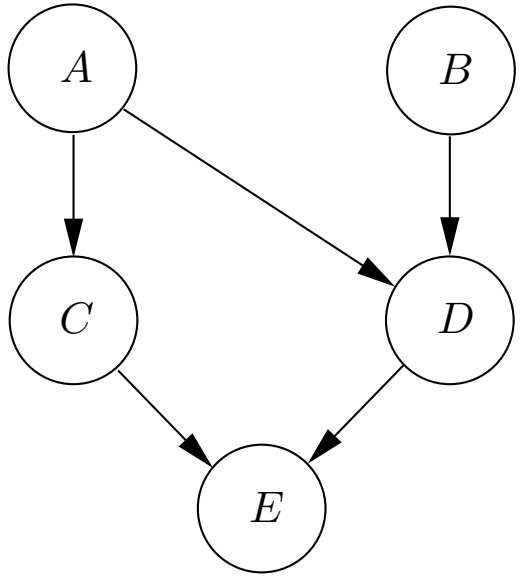
The original model



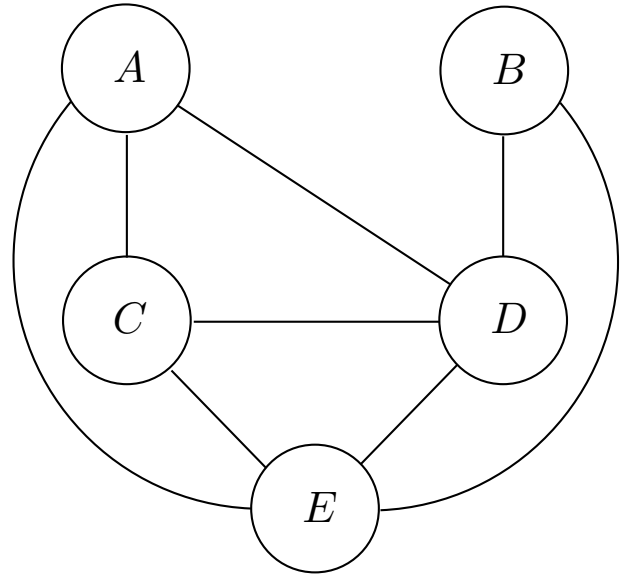
After one iteration

Example

We now condition on one variable and ask the questions $I(A, C, E)?$, $I(A, E, C)?$, $I(B, C, D)?$, $I(B, C, E)?$, $I(B, D, C)?$, $I(B, D, E)?$, $I(B, E, C)?$, $I(B, E, D)?$, $I(C, B, A)?$, $\ldots$,$I(C, D, A)?$, $I(C, D, B)?$.



The original model



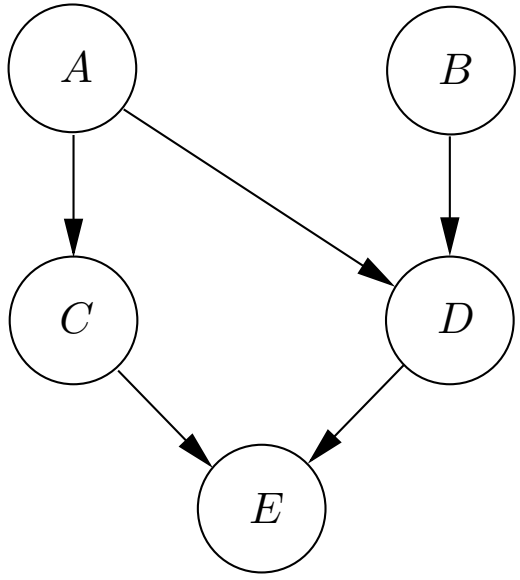
After one iteration

The question $I(C, D, A)?$ has the answer "yes":

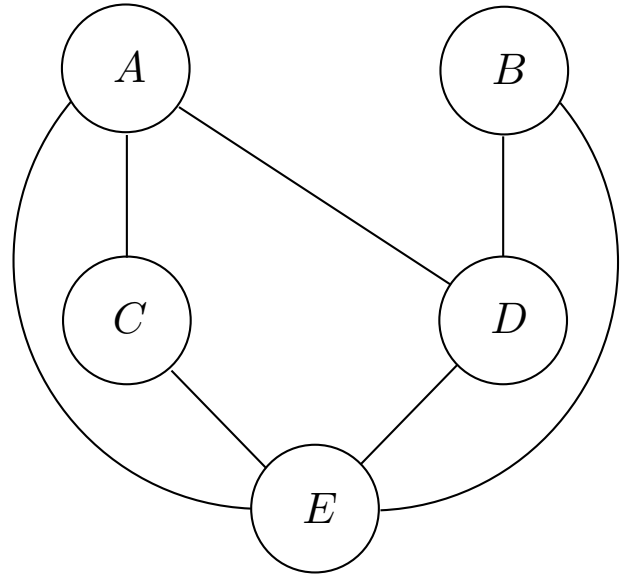
- we therefore remove the link $C - D$.

Example

We now condition on two variable and ask questions like $I(B, C, \{D, E\})?$.



The original model



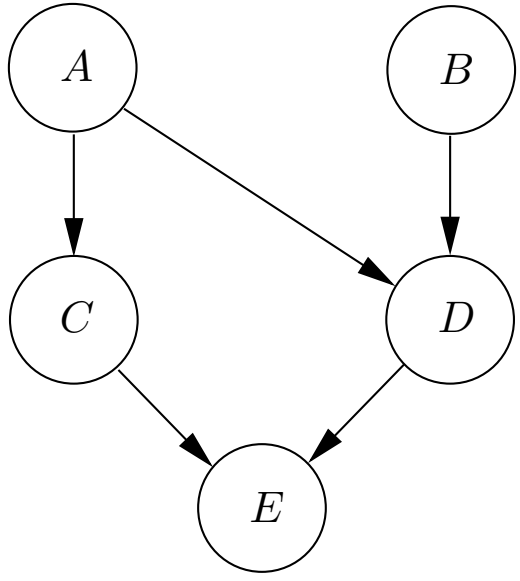
After two iterations

The questions $I(B, E, \{C, D\})?$ and $I(E, A, \{D, C\})?$ have the answer “yes”:

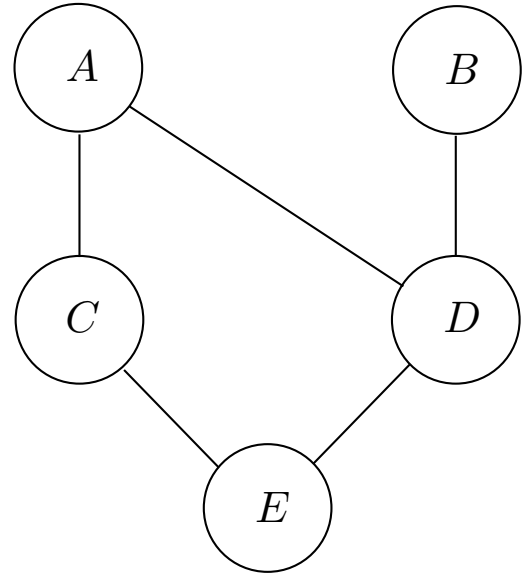
- we therefore remove the links $B - E$ and $A - E$.

Example

We now condition on three variables, but since no nodes have four neighbors we are finished.



The original model



After three iterations

The identified set of independence statements are then $I(A, B)$, $I(B, C)$, $I(C, D, A)$, $I(A, E, \{C, D\})$, and $I(B, E, \{C, D\})$. They are sufficient for applying rules 1-4.

Real world data

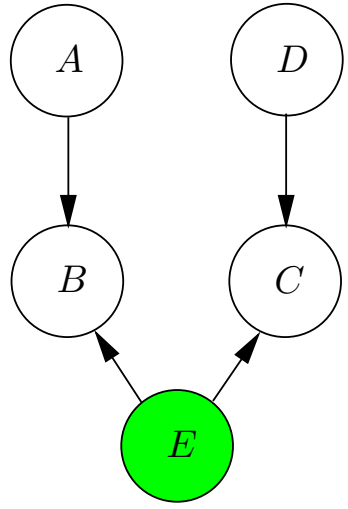
The oracle is a statistical test, e.g. conditional mutual information:

$$CE(A,B|X) = \sum_X P(X) \sum_{A,B} P(A,B|X) \log \frac{P(A,B|X)}{P(A|X)P(B|X)}.$$

$$I(A,B,X) \Leftrightarrow CE(A,B|X) = 0.$$

However, all tests have false positives and false negatives, which may provide false results/causal relations!

Similarly, false results may also be caused to hidden variables:



Properties

- If the database is a faithful sample from a Bayesian network and the oracle is reliable, then a solution exists.
- You can only infer causal relations if no hidden variables exists.