# Bayesian Networks for Reliability and Risk Analysis

Ibrahim Ahmed
ibrahim.ahmed@polimi.it

## Part 1:

- Basics
- Bayesian Networks: key concepts
- Bayesian Networks for Reliability and Risk Analysis

## Part 2:

- Applications
- Enhancements
- Conclusions

## Part 1:

- Basics
- Bayesian Networks: key concepts
- Bayesian Networks for Reliability and Risk Analysis

Basics

Bayesian Networks: key concepts

Bayesian Networks for Reliability and Risk Analysis

# Probability

**Definition:** Probability *P* is a function that maps all events *A* onto real numbers and satisfies the following three axioms:

1. If S is the set of all possible outcomes, then $P(S) = 1$
2. $0 \le P(A) \le 1$
3. If *A* and *B* are mutually exclusive ($A \cap B = \emptyset$) then

$$P(A \cup B) = P(A) + P(B)$$

From the three axioms it follows that:

I. $P(\emptyset) = 0$

II. If $A \subset B$, then $P(A) \le P(B)$

III. $P(\bar{A}) = 1 - P(A)$

IV. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

**Definition of independence**: Two events A and B are independent if

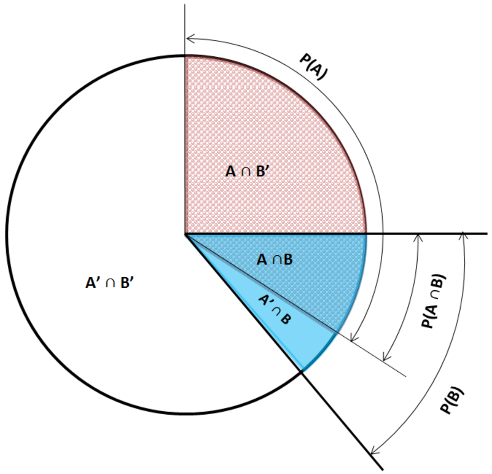
$$P(A \cap B) = P(A)P(B)$$

Conditional probability $P(A|B)$ of *A* given that *B* has occurred is

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

**Note:** If A and B are independent, the probability of *A* (*B*) does not depend on whether *B* (*A*) has occurred or not:

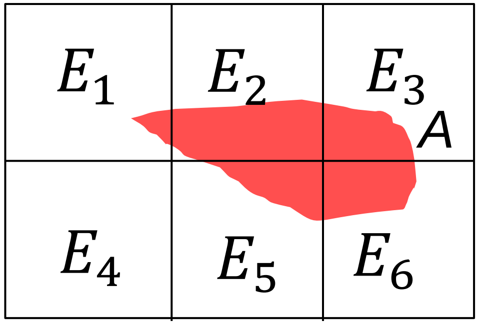
$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)\,P(B)}{P(B)} = P(A)$$



# Law of total probability

If $E_1,\ldots., E_n$ are mutually exclusive and collectively exhaustive events, then

$$P(A) = P(A|E_1)P(E_1) + \cdots + P(A|E_n)P(E_n)$$




Most frequent use of this law:

– Events A and B are mutually exclusive and collectively exhaustive

– Probabilities $P(A|B)$, $P(A|\bar{B})$, and $P(B)$ are known

– These can be used to compute

$$P(A) = P(A|B)P(B) + P(A|\bar{B})P(\bar{B})$$

# Bayes' rule

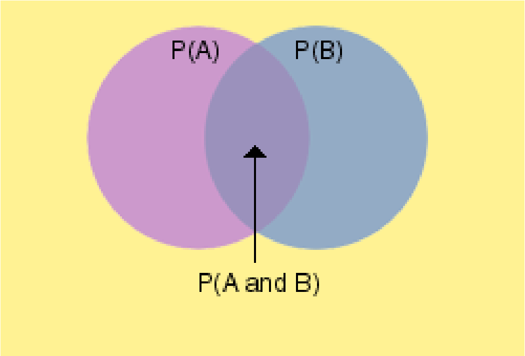
**Bayes' rule**: $P(A|B) = \dfrac{P(B|A) \cdot P(A)}{P(B)}$

It follows from:

- Definition of conditional probability:

$$P(A|B) = \frac{P(A \cap B)}{P(B)};\ P(B|A) = \frac{P(B \cap A)}{P(A)}.$$

- Commutative laws: $P(B \cap A) = P(A \cap B)$.



**Example**:

- The probability of a fire in a certain building is 1/10000 any given day.
- An alarm is activated whenever there is an actual fire, but also once in every 200 days for no reason (false alarm).
- Suppose the alarm is activated. What is the probability that there is a fire?

**Solution:**

$F = Fire,\ \bar{F} = No\ fire,\ A = Alarm,\ \bar{A} = No\ alarm$

$P(F) = 0.0001,\ P(\bar{F}) = 0.9999,\ P(A|F) = 1,\ P(A|\bar{F}) = 0.005$

Bayes: $P(F|A) = \frac{P(A|F)P(F)}{P(A)} = \frac{1 \cdot 0.0001}{0.0051} \approx 2\%$

**Law of total probability**: $P(A) = P(A|F)P(F) + P(A|\bar{F})\ P(\bar{F}) = 0.0051$

# Exercise

A test for diagnosing a particular degradation mechanism is known to be 95% accurate.

The test is performed on a component and the result is positive.

Suppose the component comes from a fleet of 100000, where 2000 suffer from this degradation.

What is the probability that the component is affected by the considered degradation mechanism?

Basics

Bayesian Networks: key concepts

Bayesian Networks for Reliability and Risk Analysis

**Bayesian Network (BN)**: is a directed acyclic graph consisting of:

- **Nodes** $V = \{1, \dots, N\}$, shown as circles, represent the random events whose combination can lead to system failure.

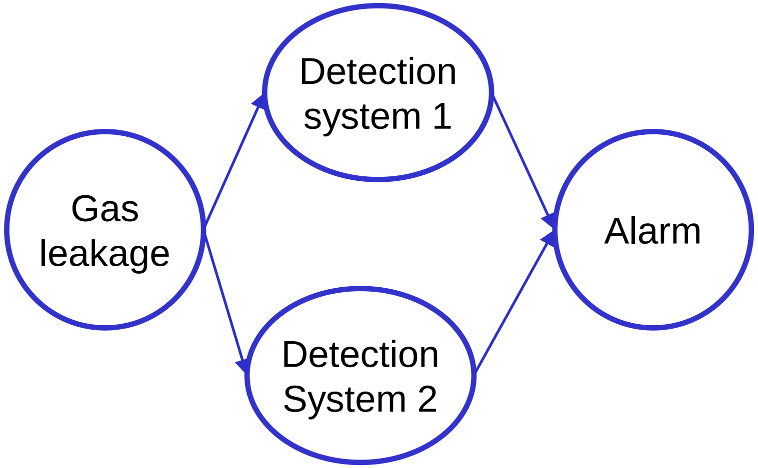
- **Directed arcs** $E \subseteq \{(i,j) | i,j \in V, i \neq j\}$ indicate conditional dependencies among nodes. Specifically, the arc $(i,j) \in E$ which connects node $j \in V$ to node $i \in V$ shows that the event at node $j$ is conditionally dependent to the event at node $i$.

Bayesian Network are also called **Bayesian Belief Networks (BBNs)**

# Bayesian Network Example

| Leakage | DS1: Activated | DS1: Not Activated |
|---|---|---|
| Yes | 0.95 | 0.05 |
| No | 0.05 | 0.95 |

| Leakage yes | Leakage no |
|---|---|
| 0.01 | 0.99 |




| Detection System 1 | Detection System 2 | Alarm: yes | Alarm: no |
|---|---|---|---|
| Activated | Activated | 0.99 | 0.01 |
| Activated | Not Activated | 0.95 | 0.05 |
| Not Activated | Activated | 0.95 | 0.05 |
| Not Activated | Not Activated | 0.01 | 0.99 |

| Leakage | DS2: Activated | DS2: Not Activated |
|---|---|---|
| Yes | 0.95 | 0.05 |
| No | 0.05 | 0.95 |

# Acyclic… what?

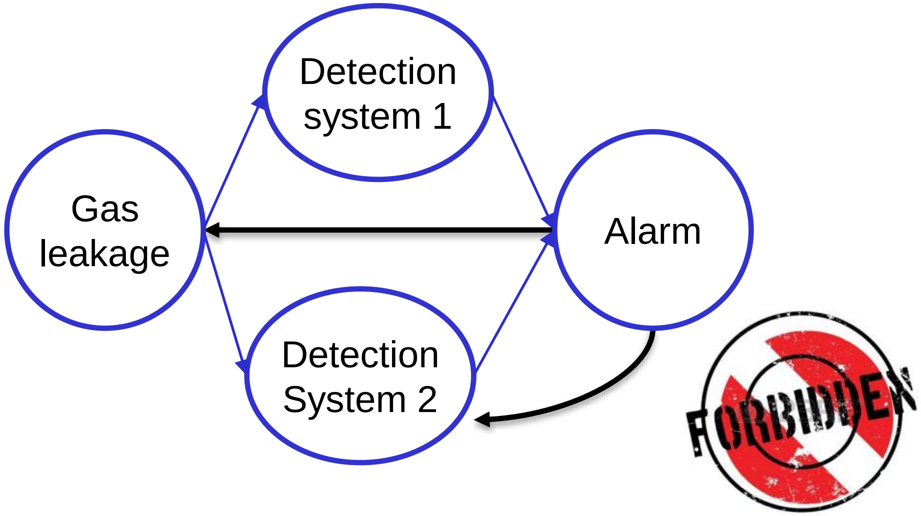


A **path** is a sequence of nodes $(i_1, i_2, \ldots, i_\eta)$, $\eta > 1$ such that

$$(i_j, i_{j+1}) \in E;\ j < \eta$$

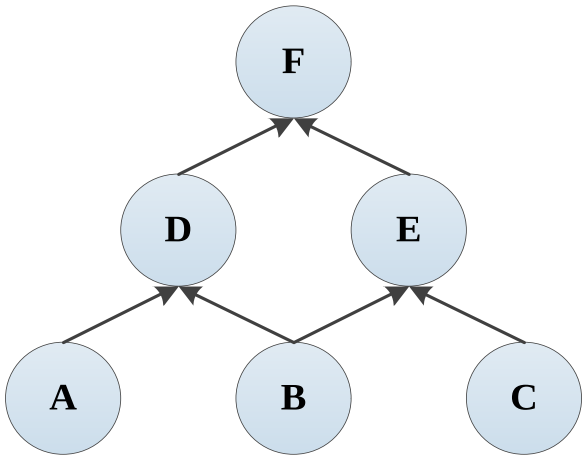
BBN **acyclic** → there is no path $(i_1, i_2, \ldots, i_\eta)$, $\eta > 1$ such that

$$(i_j, i_{j+1}) \in E, j < \eta \text{ and } i_1 = i_\eta$$



# Bayesian Networks: Consist of two parts

**1- qualitative part**, which is its structure,




A is the parent of node D (i.e. node D is the child of node A).

Nodes A, B, C that have no parents, are root nodes, and Node F that has no child is a leaf node

**2- quantitative part**, which contains its probabilistic parameters.

The parameters of BN are probability distributions (PDs). While a dependency between a node and its parent is expressed by a conditional probability distribution (CPD), root nodes are described by their marginal probability distributions (MPDs).

# Bayesian Networks, why?



Bayesian networks are **probabilistic graphical models**, which offer a convenient and efficient way of generating joint distribution of all its events.

- **Convenient**: causal relationships between events are easy to model.
- **Efficient**: no redundancies in terms of graphical modelling and probability computations.
- **Flexible**: capable of handling imprecise information by capturing quantitative and qualitative data.

"Microsoft's competitive advantage lies in its expertise in Bayesian Networks"
-- Bill Gates, quoted in LA Times, 1996

We define:

- **follower** nodes of $i \in V$: $V_+^i = \{j | (i,j) \in E\}$
- **predecessor** nodes (parent) of $i \in V$: $V_-^i = \{j | (j,i) \in E\}$

All nodes can be partitioned into

- **Leaf nodes** $V^L = \{i \in V | V_-^i = \emptyset\}$
- **Dependent nodes** $V^D = V \setminus V^L = \{i \in V | V_-^i \neq \emptyset\}$

The **depth** of node $i \in V$ in the network can be calculated recursively by

$$d^i = \begin{cases} 0 & V_-^i = \emptyset \\ 1 + \max\limits_{j \in V_-^i} d^j & V_-^i \neq \emptyset \end{cases}$$

# Solving Bayesian Networks

$\boldsymbol{X}^i$ = random variable representing the uncertainty in the state of event at node $i \in V$.

The realization $s$ of $\boldsymbol{X}^i$ belongs to the set of states $S^i = \{0, \dots, |S^i|\}$

$\boldsymbol{X} = [\boldsymbol{X}^1, \dots, \boldsymbol{X}^N]$= BN state vector (where $N$ is the number of node)

$$P(\boldsymbol{X}) = \prod_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N}} P\left(\boldsymbol{X}^i \middle| \boldsymbol{X}^j, j \in V_-^i\right)$$

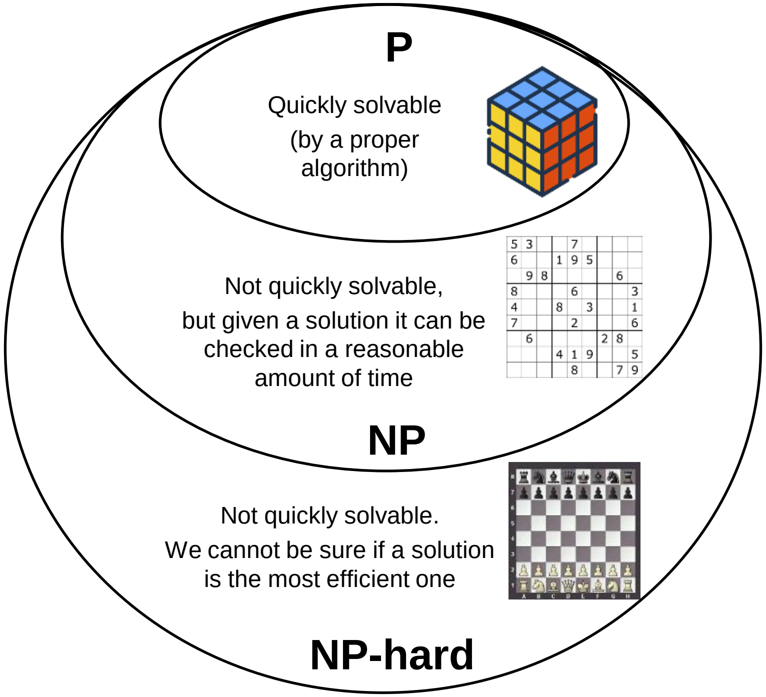
The BN can be solved by propagating the uncertainty from leaf nodes $V^L$ to nodes with the largest depth

# Computational issues

<u>Theorem</u>: Computing event probabilities in a Bayesian network is **NP-hard**.

**NP-hard:** complexity class of problems which cannot be solved by a **N**ondeterministic (ideal) machine in **P**olynomial time (i.e., necessary number of steps upper bounded by a polynomial function of the number of inputs)

That means that there is no general way to solve a NP-hard problem!




Problem complexity classification

# Computational issues



Theorem: Computing event probabilities in a Bayesian network is **NP-hard**

Hardness does not mean it is impossible to perform inference, but:

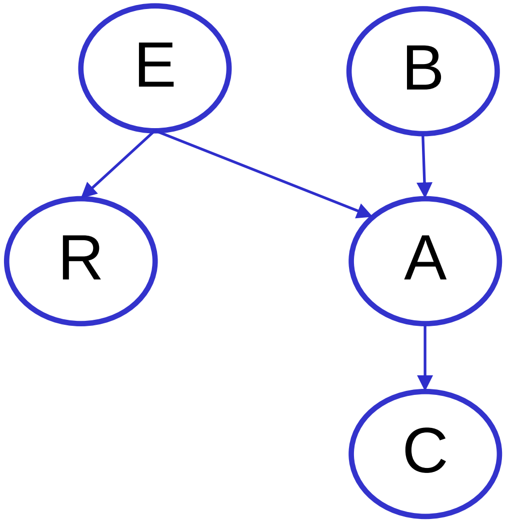
- There is no general procedure that works efficiently for all networks
- For particular families of networks, there are proved efficient procedures
- Different algorithms are developed for inferences in Bayesian networks

There are available software that efficiently perform Bayesian Network inference through a library of functions for several popular algorithms, among those:

- **GeNIe Modeler**: https://www.bayesfusion.com/genie-modeler
- **HUGIN Expert**: https://www.hugin.com/
- **BayesiaLab**: https://www.bayesia.com/

# Computational issues

Factored representation may have exponentially fewer parameters than full joint $P(\boldsymbol{X})$




$$
\begin{aligned}
& P(E,B,R,A,C) \\
& \quad = \quad P(E)P(B|E)P(R|B,E)P(A|R,B,E)P(C|A,R,B,E) \\
& \quad = \quad P(E)P(B)P(R|E)P(A|B,E)P(C|A)
\end{aligned}
$$

If $|S^i|=2$ for every $i$, the number of parameters reduces from $2^5 - 1 = 31$ to $1+1+2+4+2=10$

Factored representation may have exponentially fewer parameters than full joint $P(\boldsymbol{X})$

A real case study: Monitoring Intensive-Care Patients

- 37 variables
- 509 parameters, instead of $2^{37}$

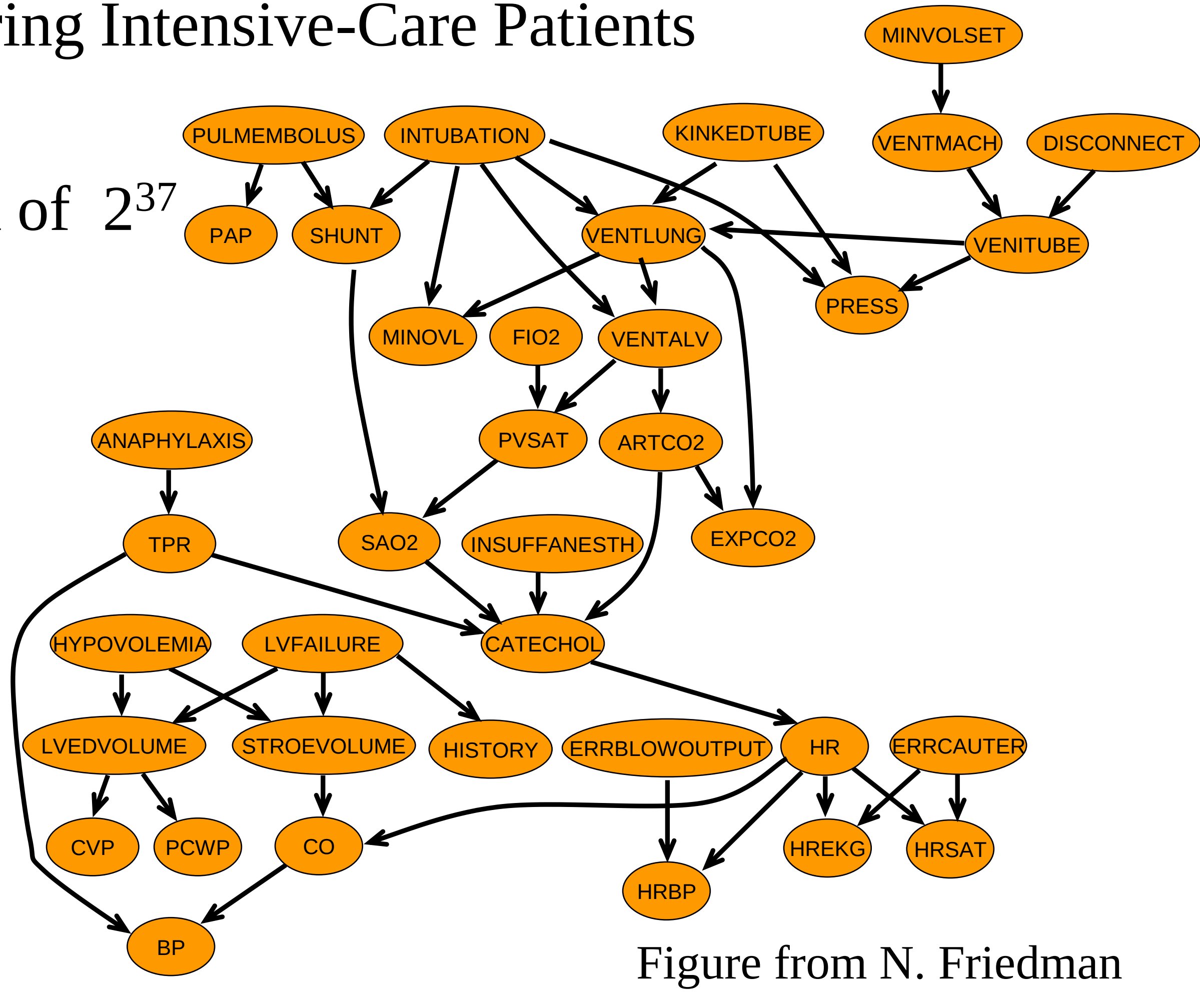


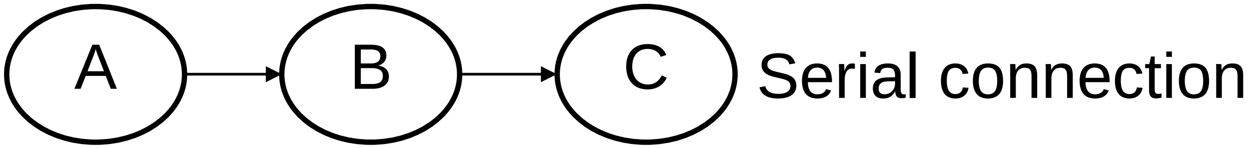

Figure from N. Friedman

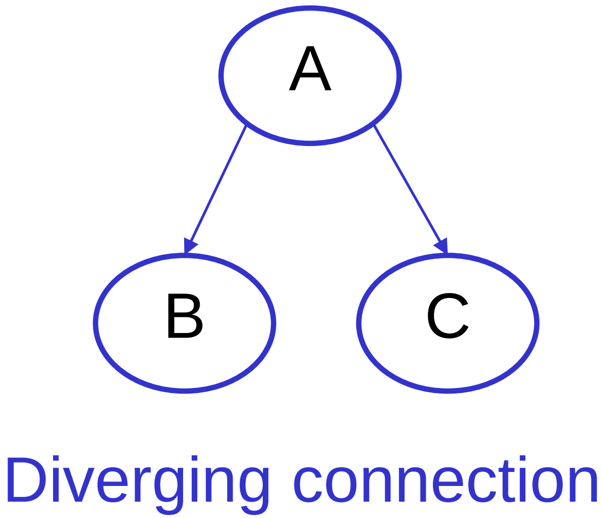
# Inference

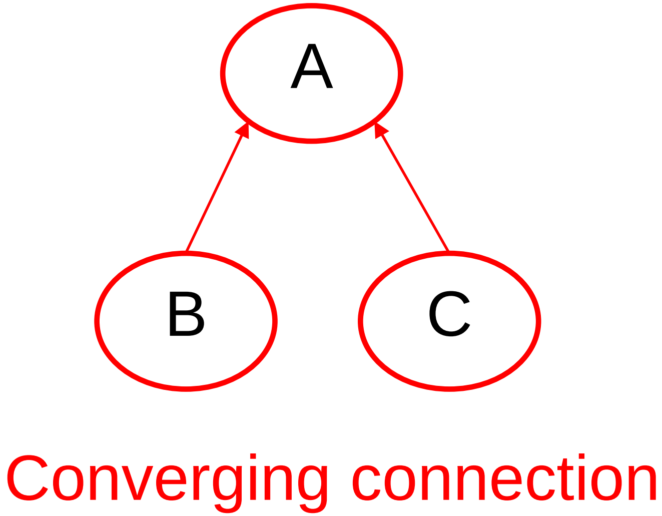
Assume to collect some observation (**evidence**) from the system. In this case the state is said “instantiated”

How would this evidence impact the probabilities of the events?

Consider the following three schemes:



# Serial Connection

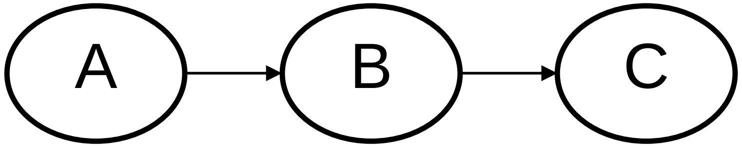
- Evidence on A influences successor nodes
- Evidence on C influences predecessor nodes
- Evidence on B makes A and C independent

Example

A: Failure Event (yes/no)

B: Alarm Activation (yes/no)

C: Plant Evacuation (yes/no)




If we have evidence of a Failure Event→ we change the probability of Alarm Activation and, then, the probability of having a Plant Evacuation.

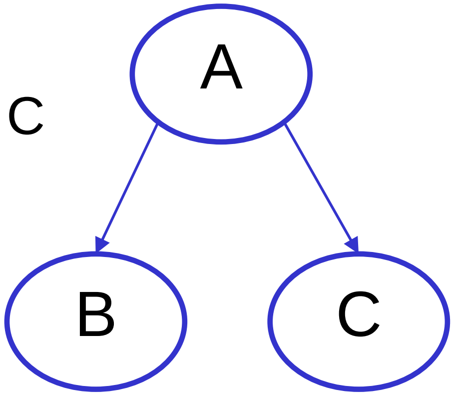
If we have evidence of Plant Evacuation → we change the probability of Alarm Activation and, then, the probability of having a failure event.

If we know that the alarm is activated (or not), we cut the communication between Failure Event and Plant Evacuation

# Diverging Connection



- Evidence on A influences nodes B and C
- Evidence on C influences A and, then, B
- Evidence on A cuts the communication between B and C




Diverging connection

Example

A: Failure Event (yes/no)

B: Detection System 1 (activated/not activated)

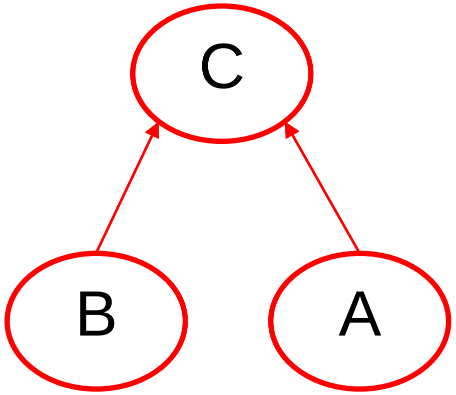
C: Detection System 2 (activated/not activated)

If we have evidence of the activation of Detection System 1 → we change the probability of failure event and, then, the probability of activation of Detection System 2.

If we know that the failure event is occurrred (or not), we know the probability of activation of Detection System 1 and Detection System 2, which do not influence each other

# Converging Connection

- Evidence on C influences nodes B and A
- Evidence on B influences node C,only
- Evidence on C and B, influences node A differently from influence of node C only




Example

A: Fuel level in a car (empty/full)

B: Spark plugs (working/failed)

C: Start (Yes/No)

If we have evidence that the car cannot start→ we change the probability that the fuel level is empty and sparks are failed.

If we have evidence that the car cannot start and that the fuel is full→ we change the probability that sparks are failed

# d-separation

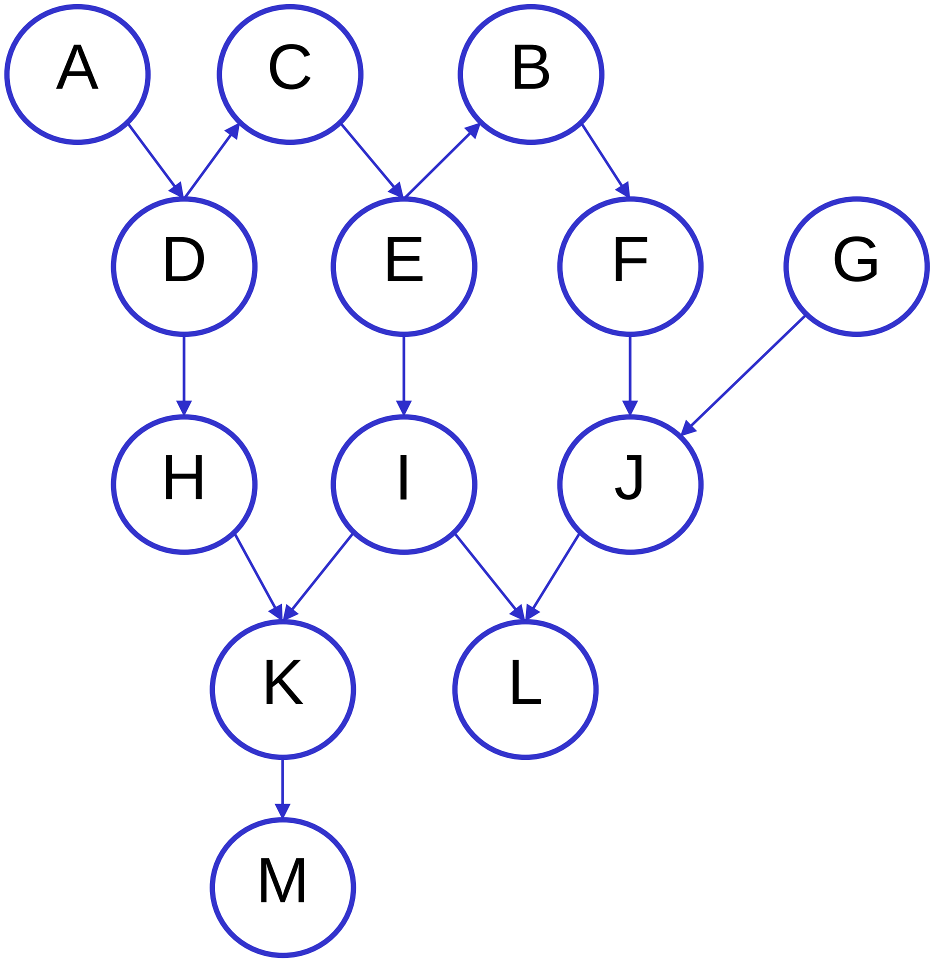


In a Bayesian Network, two nodes A and B are d(irectional)-separated if for all indirected paths between A and B there is a node C such that at least one of the following conditions holds:

- The connection A-B is serial or diverging and C is instantiated
- The connection A-B contains converging structure and neither C nor any of C's successors are instantiated

If events A and B are d-separeted, evidence on A does not influence B

# Example

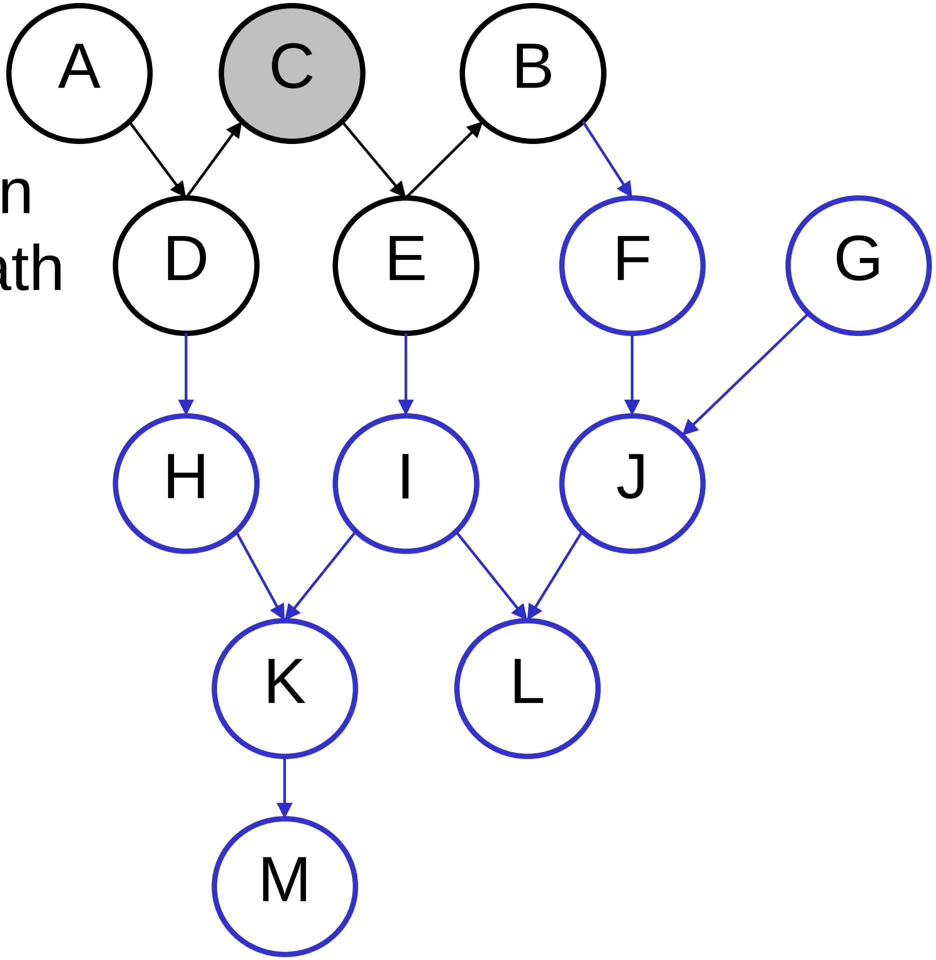



Are A and B d-separated?

# Example

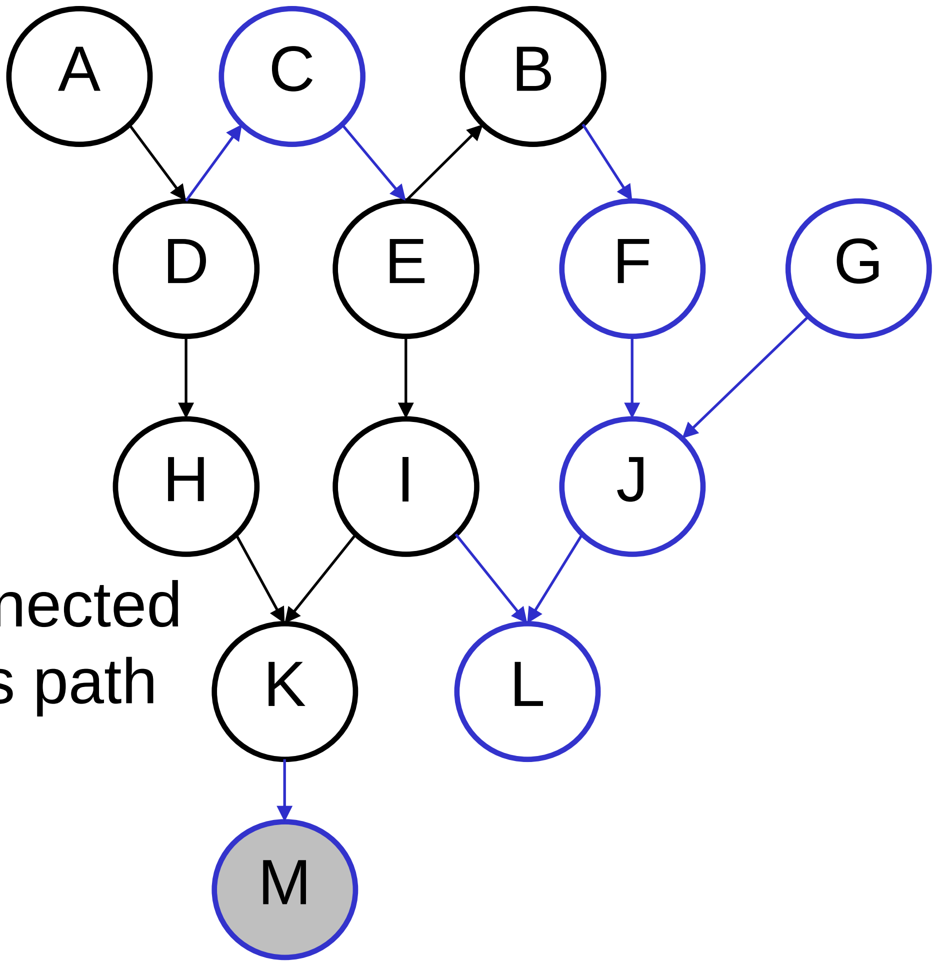


Yes, on this path




Are A and B d-separated?

# Example




d-connected
by this path

Are A and B d-separated?

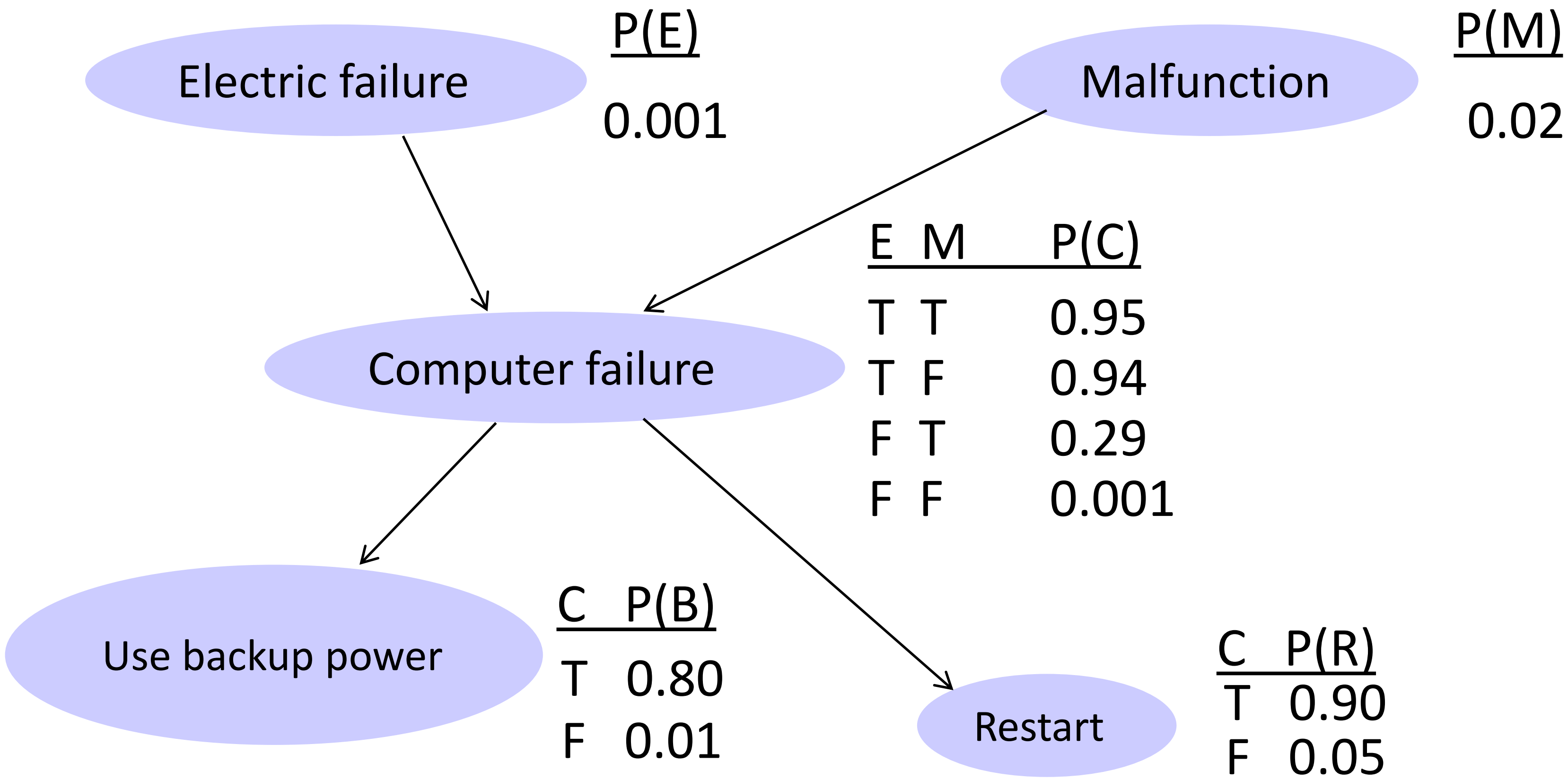
Question: assume to collect some observations (**evidence**) from the system; how would this evidence impact the probabilities of the events?

The conditional probability of a random event given the evidence is known as **a posteriori belief**, useful in case of:

- **Prediction**: computing the probability of an outcome event given the starting condition → Target is a descendent of the evidence!
- **Diagnosis**: computing the probability of disease/fault given symptoms → Target is an ancestor of the evidence!

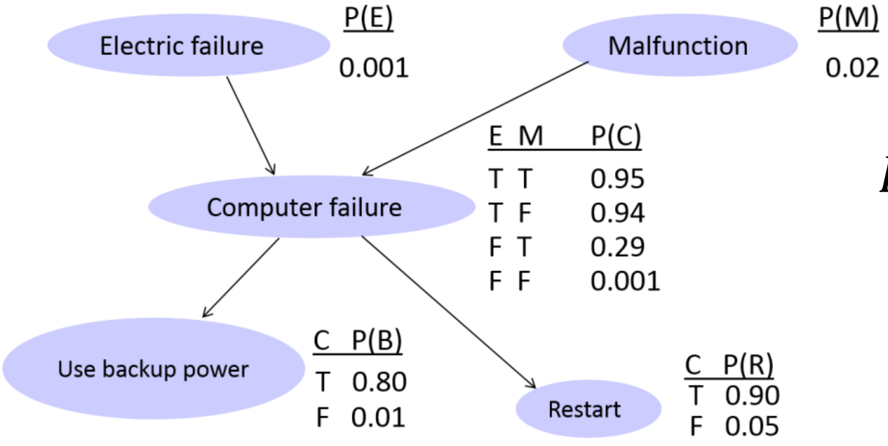
Note: probabilistic inference can propagate and combine evidences from all parts of the network (the directions of arcs do not limit the directions of the queries)

# Computer example



# Examples of inference

What it is the probability that the backup power is working given an electrical failure?




$$P(B|E) = P(B|C)P(C|E) + P(B|\bar{C})P(\bar{C}|E)$$

$$P(C|E) = P(C|\bar{M}, E)P(\bar{M}) + P(C|M, E)P(M)$$

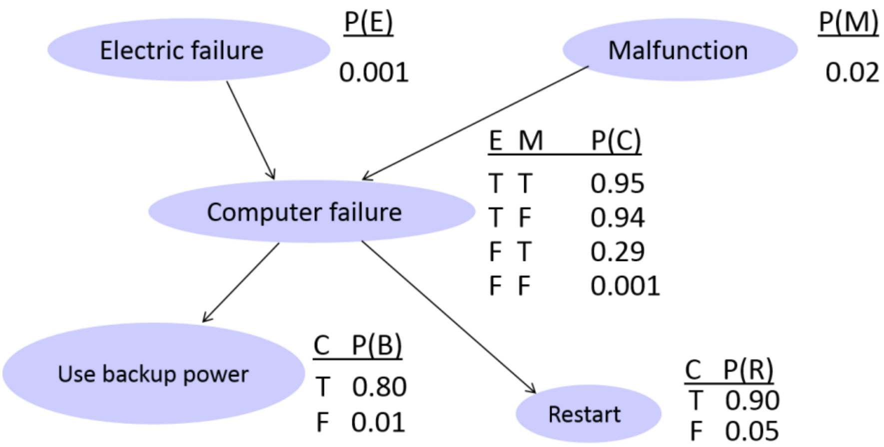
$$P(C|E) = 0.98 \cdot 0.94 + 0.02 \cdot 0.95 = 0.94$$

$$P(B|E) = P(C|E) \cdot 0.8 + P(\bar{C}|E) \cdot 0.01$$

$$P(B|E) = 0.94 \cdot 0.8 + 0.06 \cdot 0.01 = \color{red}{0.75}$$

# Exercise on inference

What it is the probability that the electricity is not working given a backup power event (== you RELY on backup power)?

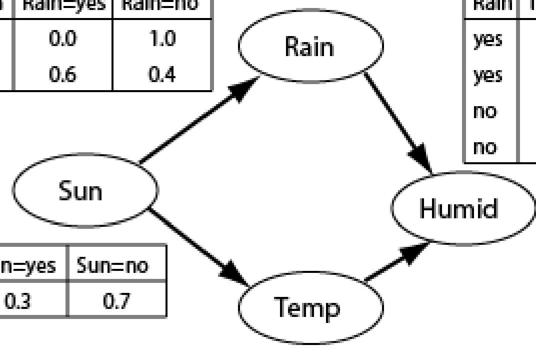


# Weather example



| Sun | Rain=yes | Rain=no |
|---|---|---|
| yes | 0.0 | 1.0 |
| no | 0.6 | 0.4 |

| Rain | Temp | Humid=yes | Humid=no |
|---|---|---|---|
| yes | high | 0.9 | 0.1 |
| yes | low | 0.4 | 0.6 |
| no | high | 0.5 | 0.5 |
| no | low | 0.3 | 0.7 |




| Sun=yes | Sun=no |
|---|---|
| 0.3 | 0.7 |

| Sun | Temp=high | Temp=low |
|---|---|---|
| yes | 0.8 | 0.2 |
| no | 0.5 | 0.5 |

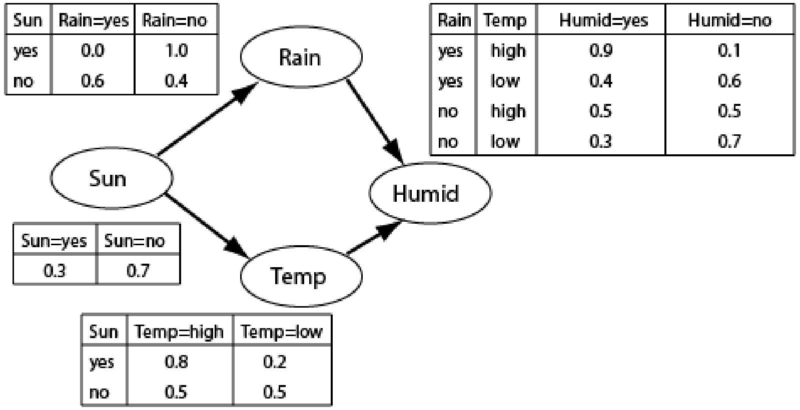
In the above network there are four nodes. Each node has two states. For example the node “sun” has two states: 1- “yes” which represents sunny weather, and “no” which represents non sunny weather. Note that nodes can have more than two states.

Q1. Calculate the probability of having high temperature : p(Temp=high)=?

# Weather example: Solution



Q1. Calculate the probability of having high temperature : p(Temp=high)=?




| Sun | Rain=yes | Rain=no |
|---|---|---|
| yes | 0.0 | 1.0 |
| no | 0.6 | 0.4 |

| Rain | Temp | Humid=yes | Humid=no |
|---|---|---|---|
| yes | high | 0.9 | 0.1 |
| yes | low | 0.4 | 0.6 |
| no | high | 0.5 | 0.5 |
| no | low | 0.3 | 0.7 |

| Sun=yes | Sun=no |
|---|---|
| 0.3 | 0.7 |

| Sun | Temp=high | Temp=low |
|---|---|---|
| yes | 0.8 | 0.2 |
| no | 0.5 | 0.5 |

p(Temp=high)=p(Temp=high/Sun)p(Sun)

p(Temp=high)=p(Temp=high/Sun)p(Sun) =

p(Temp=high/Sun=yes)p(Sun=yes)+ p(Temp=high/Sun=no)p(Sun=no) =

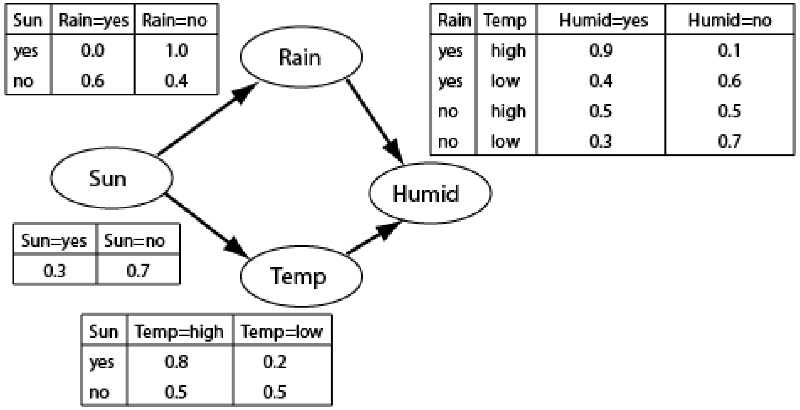
0.8*0.3+0.5*0.7=0.59

# Weather Exercise



Q1. Calculate the probability of having no humid, p(Humid=no)

Q2. Calculate p(Humid=yes/Rain=no)




| Sun=yes | Sun=no |
| --- | --- |
| 0.3 | 0.7 |

| Sun | Rain=yes | Rain=no |
| --- | --- | --- |
| yes | 0.0 | 1.0 |
| no | 0.6 | 0.4 |

| Sun | Temp=high | Temp=low |
| --- | --- | --- |
| yes | 0.8 | 0.2 |
| no | 0.5 | 0.5 |

| Rain | Temp | Humid=yes | Humid=no |
| --- | --- | --- | --- |
| yes | high | 0.9 | 0.1 |
| yes | low | 0.4 | 0.6 |
| no | high | 0.5 | 0.5 |
| no | low | 0.3 | 0.7 |

Basics

Bayesian Networks: key concepts

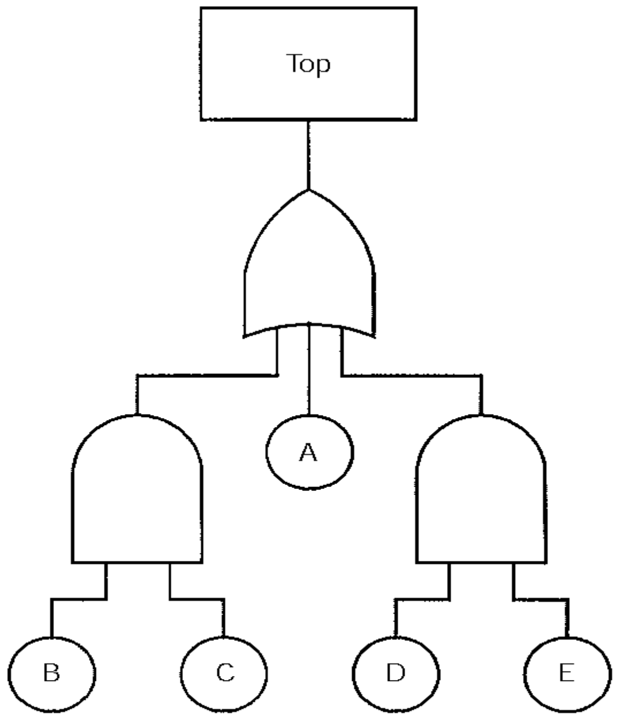
Bayesian Network for Reliability and Risk Analysis

Scenario modeling and quantification are pursued through:

## FAULT TREE ANALYSIS (FTA)

⇩

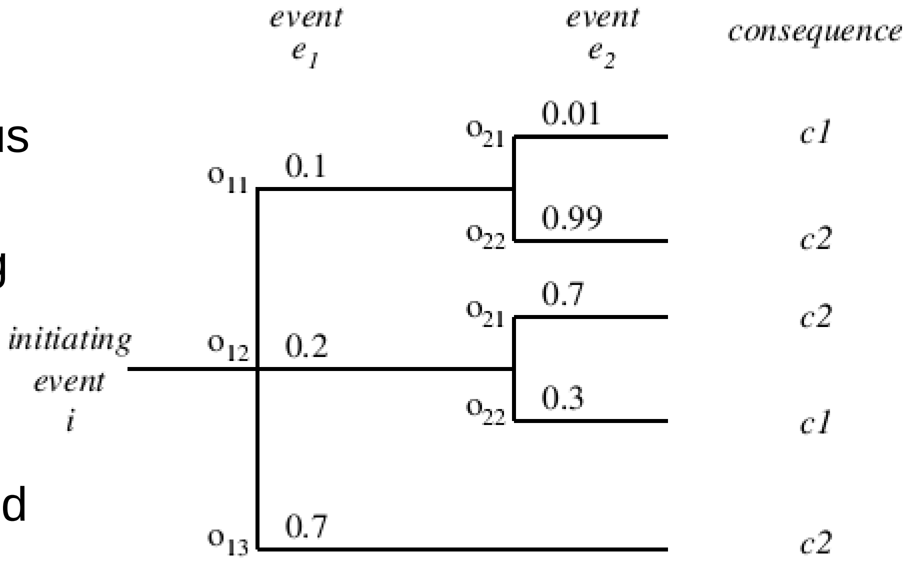
1. Events are binary events (operating/not-operating);
2. Events are statistically independent;
3. Relationships between events and causes are represented by logical gates (e.g., AND and OR for coherent FT);
4. The undesirable event, called Top Event, is postulated and the possible ways for the occurrence of this event are systematically deduced.



Scenario modeling and quantification are pursued through:

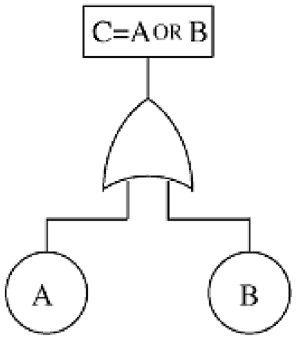
## EVENT TREE ANALYSIS (ETA)

1. System evolution following the hazardous occurrence is divided into discrete events;
2. System evolution starts from an initiating event;
3. Each event has a finite set of outcomes (commonly there are two outcomes: occurring event or not occurring) associated with the occurrence probabilities;
4. The leafs of the event tree represent the event scenarios to be analyzed.

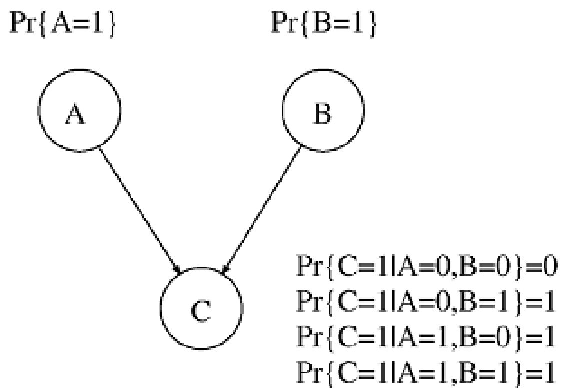


- Procedures:
  - Create a leaf node in BN instead of each basic event in FT. Although if more basic events represent a component, we can create just one leaf node for BN. In fact, we can create a leaf node in BN for each basic event, component, subsystem and … of the system.
  - For each gate of FT create a node in BN.
  - Connect nodes of BN as their respective basic events, components, subsystems, gates and … in FT are connected.
  - The direction of the connections (i.e. edges) is from the nodes that impose effect toward the nodes that receive the effect.

# Mapping FT into Bayesian Network

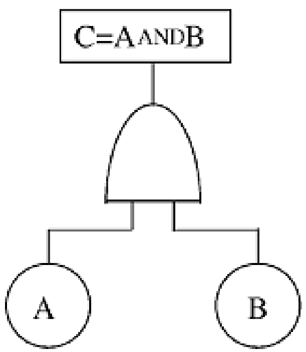



FAULT - TREE: OR Gate

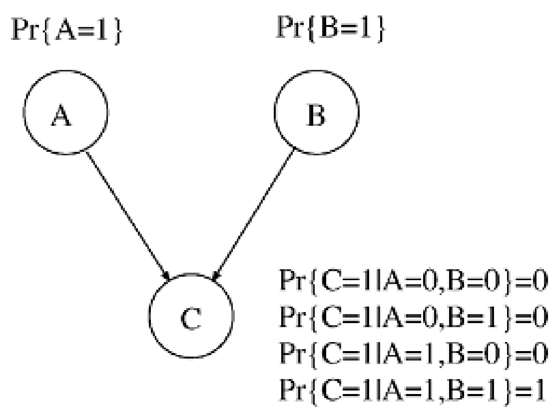



BAYESIAN NETWORK: OR Node

**Switching from binary logic to probabilistic logic!**




FAULT - TREE: AND Gate




BAYESIAN NETWORK: AND Node

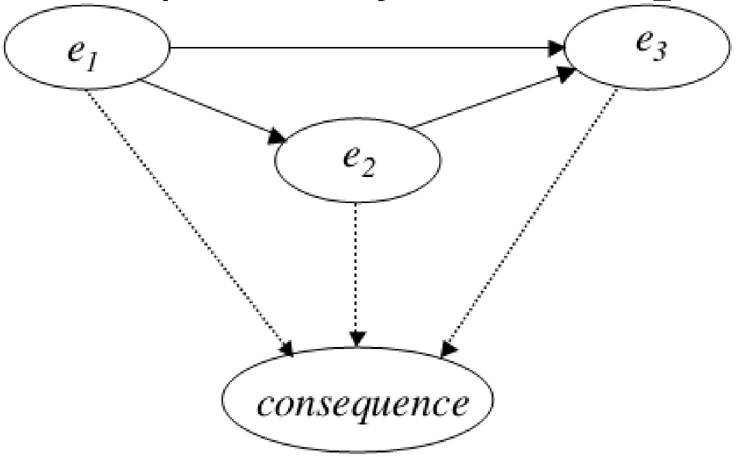
***Reference**:* Bobbio A., Portinale L., Minichino M., Ciancamerla E., "*Improving the analysis of dependable systems by mapping fault trees into Bayesian networks",* Reliability Engineering and System Safety 71, pp. 249–260 (2001) .

# Mapping ET into Bayesian Network

**Any event tree with three events $e_1$, $e_2$, and $e_3$ can be represented by the BN shown below.**

Two types of directed arc complete the network:

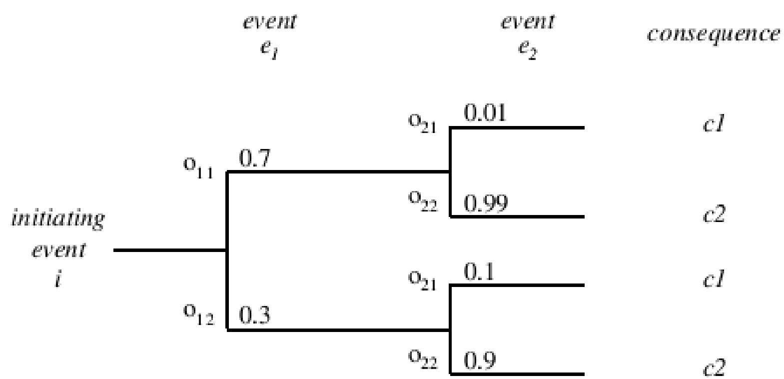
- Consequence arcs (shown as dotted lines) connect each event node to the consequence node. This relationship is deterministic: the probability table for the consequence node encodes the logical relationship between the events and the consequences.
- Causal arcs (shown as solid lines) connect each event node to all events later in time. For instance, event $e_1$ is a causal factor for event $e_2$, thus it influences the probability of event $e_2$.

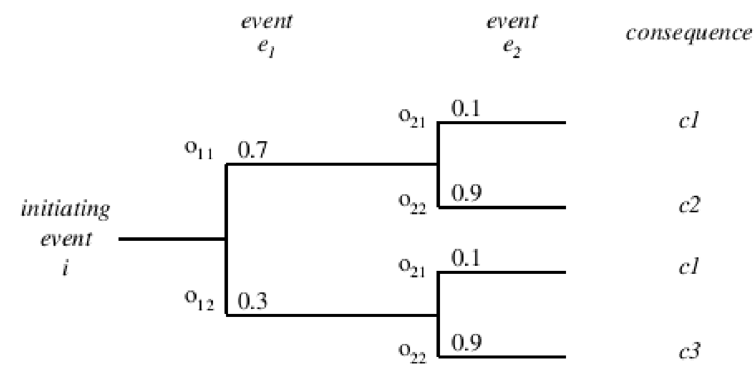


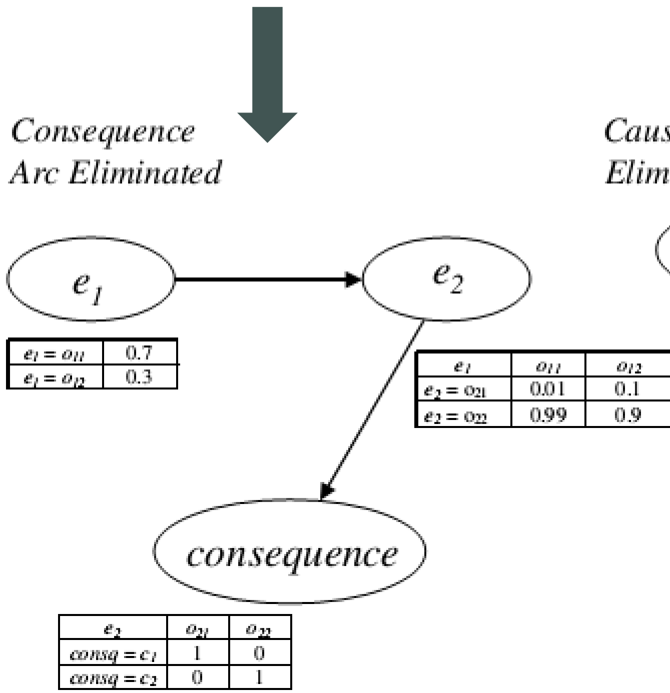

***Reference**: Bearfield G., Marsh W., "Generalizing Event Trees Using Bayesian Networks with a Case Study of Train Derailment"*, Computer Safety, Reliability, and Security (2005).

# Mapping ET into Bayesian Network (alternatives)





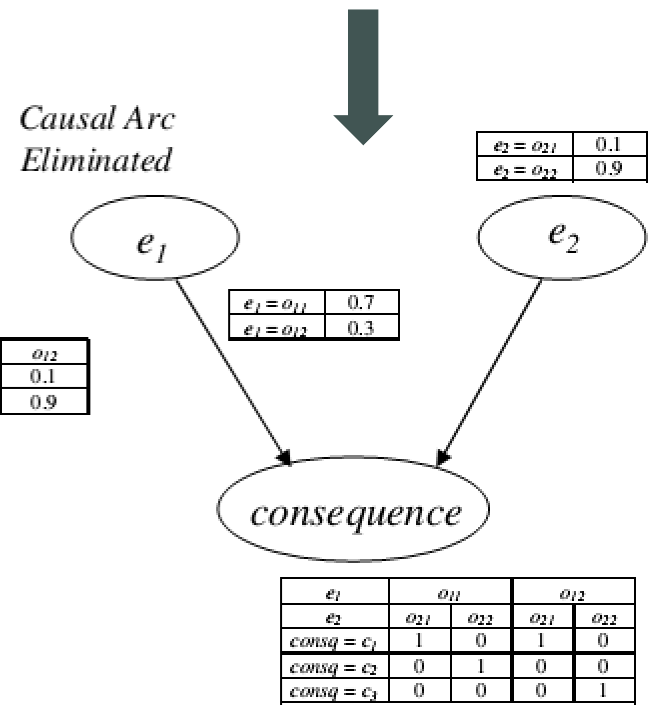




| | |
|---|---|
| $e_1 = o_{11}$ | 0.7 |
| $e_1 = o_{12}$ | 0.3 |

| $e_1$ | $o_{11}$ | $o_{12}$ |
|---|---|---|
| $e_2 = o_{21}$ | 0.01 | 0.1 |
| $e_2 = o_{22}$ | 0.99 | 0.9 |

| $e_2$ | $o_{21}$ | $o_{22}$ |
|---|---|---|
| $consq = c_1$ | 1 | 0 |
| $consq = c_2$ | 0 | 1 |

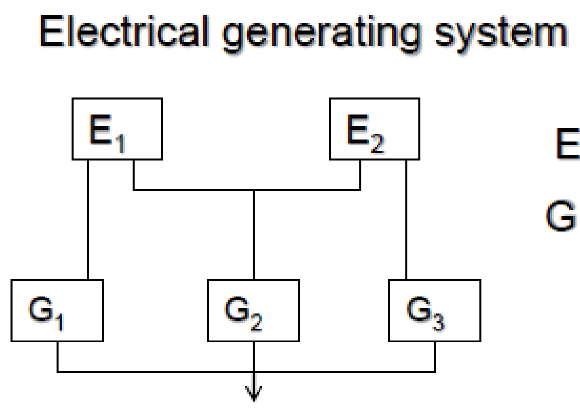



| | |
|---|---|
| $e_2 = o_{21}$ | 0.1 |
| $e_2 = o_{22}$ | 0.9 |

| | |
|---|---|
| $e_1 = o_{11}$ | 0.7 |
| $e_1 = o_{12}$ | 0.3 |

| $e_1$ | $o_{11}$ | | $o_{12}$ | |
|---|---|---|---|---|
| $e_2$ | $o_{21}$ | $o_{22}$ | $o_{21}$ | $o_{22}$ |
| $consq = c_1$ | 1 | 0 | 1 | 0 |
| $consq = c_2$ | 0 | 1 | 0 | 0 |
| $consq = c_3$ | 0 | 0 | 0 | 1 |

# Reliability Modeling through BN: example

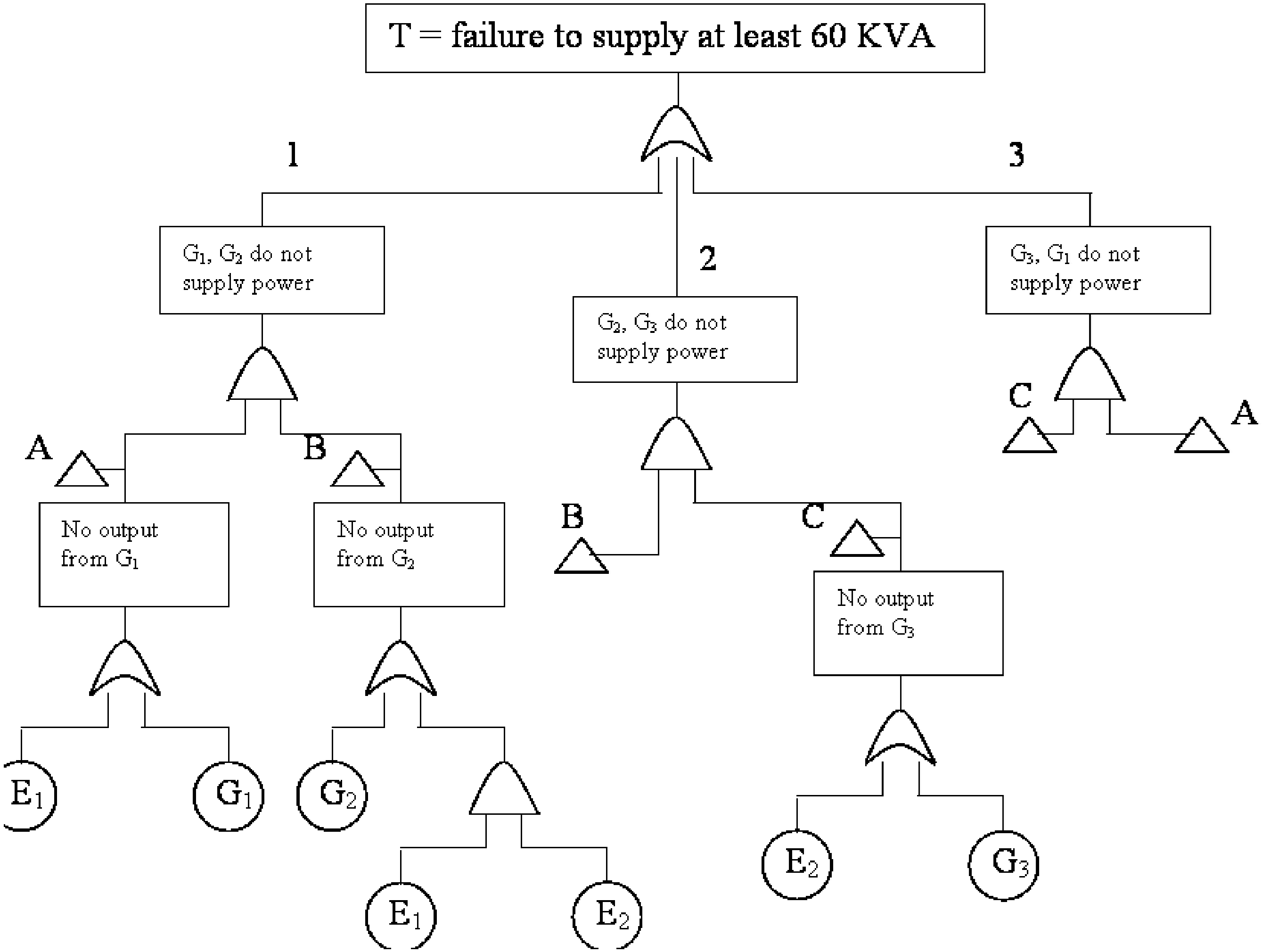


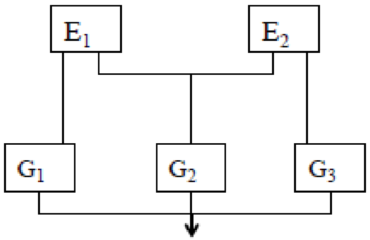

E1, E2 = engines

G1, G2, G3 = generators, each one is rated at 30 KVA

The generation system fails when the delivered power is under 60KVA

# Fault Tree

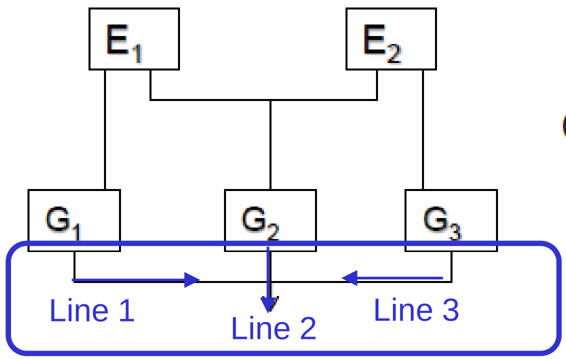

T = failure to supply at least 60 KVA
1
2
3
G1, G2 do not supply power
G2, G3 do not supply power
G3, G1 do not supply power
A
B
C
B
C
A
No output from G1
No output from G2
No output from G3
E1
G1
G2
E1
E2
E2
G3


E1
E2
G1
G2
G3

# Reliability Modeling through BN: example

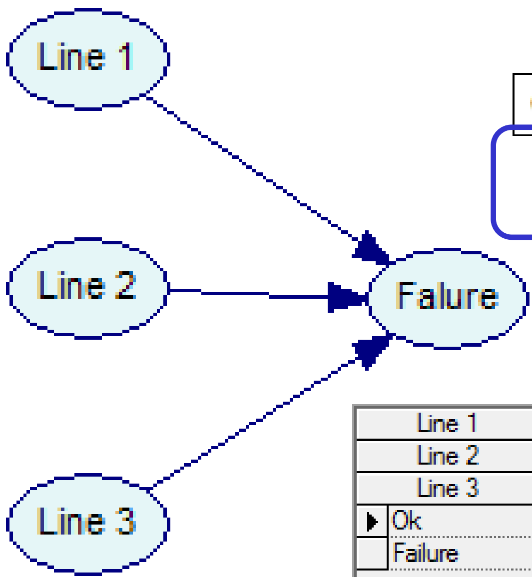


Electrical generating system




E1, E2 = engines

G1, G2, G3 = generators, each one is rated at 30 KVA

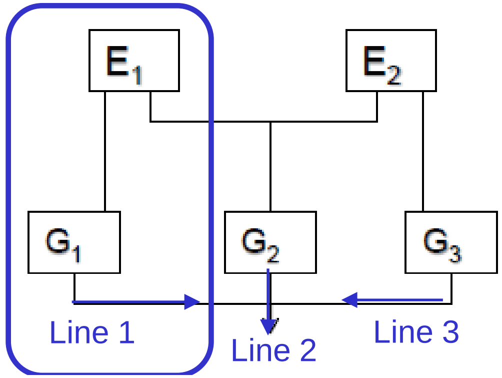



| Line 1 | PowerOk | | | | PowerKo | | | |
|---|---|---|---|---|---|---|---|---|
| Line 2 | PowerOk | | PowerKo | | PowerOk | | PowerKo | |
| Line 3 | PowerOk | PowerKo | PowerOk | PowerKo | PowerOk | PowerKo | PowerOk | PowerKo |
| Ok | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| Failure | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

# Reliability Modeling through BN: example

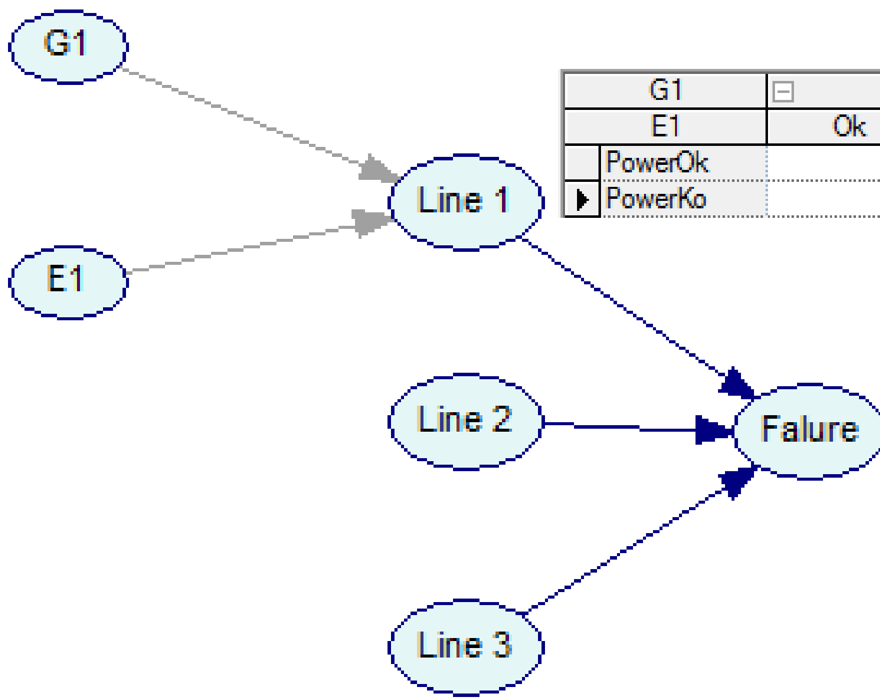


Electrical generating system



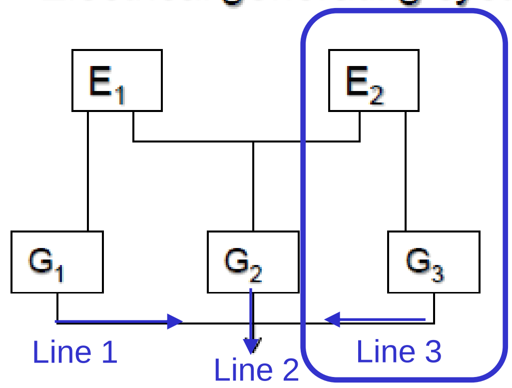

E1, E2 = engines

G1, G2, G3 = generators, each one is rated at 30 KVA



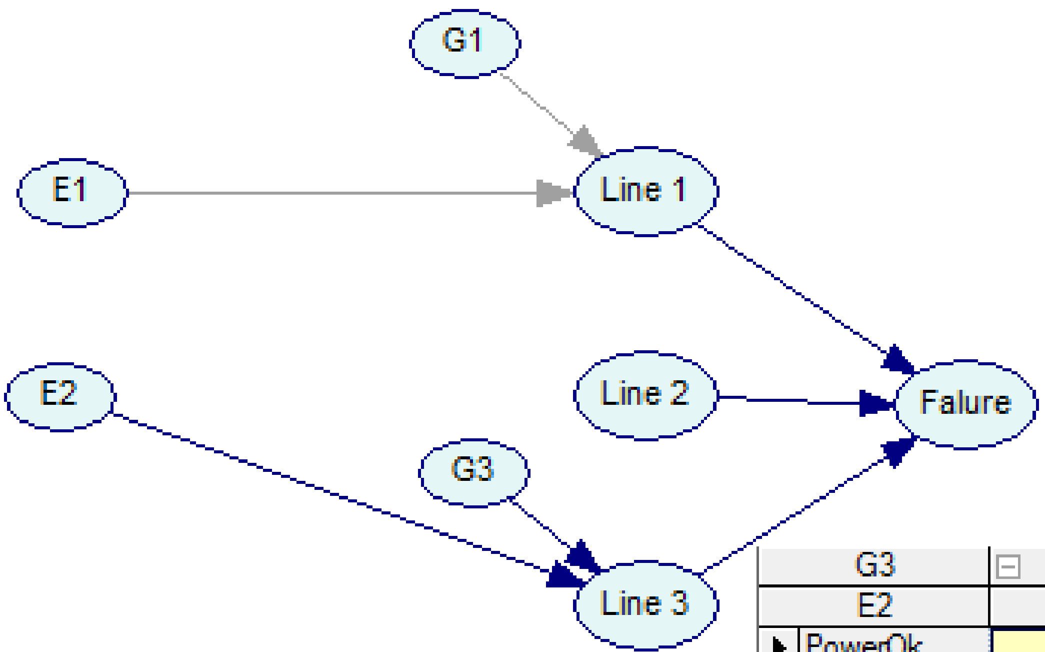

| G1 | Ok | | Failed | |
|---|---|---|---|---|
| E1 | Ok | Failed | Ok | Failed |
| PowerOk | 1 | 0 | 0 | 0 |
| PowerKo | 0 | 1 | 1 | 1 |

Electrical generating system




E1, E2 = engines

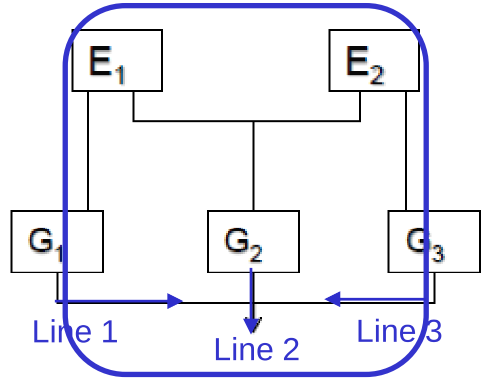
G1, G2, G3 = generators, each one is rated at 30 KVA




| G3 | Ok | | Failed | |
|---|---|---|---|---|
| E2 | Ok | Failed | Ok | Failed |
| PowerOk | 1 | 0 | 0 | 0 |
| PowerKo | 0 | 1 | 1 | 1 |

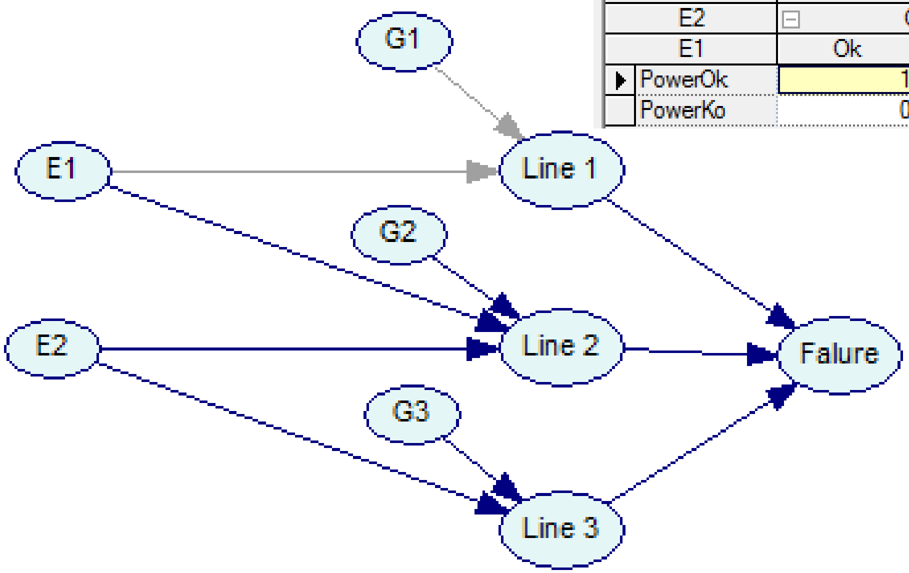
# Reliability Modeling through BN: example

Electrical generating system

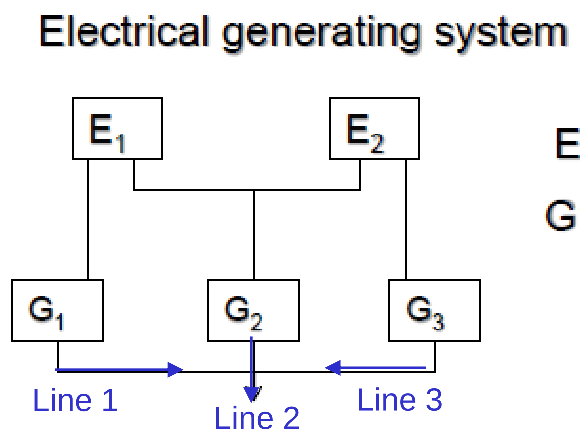



E1, E2 = engines

G1, G2, G3 = generators, each one is rated at 30 KVA

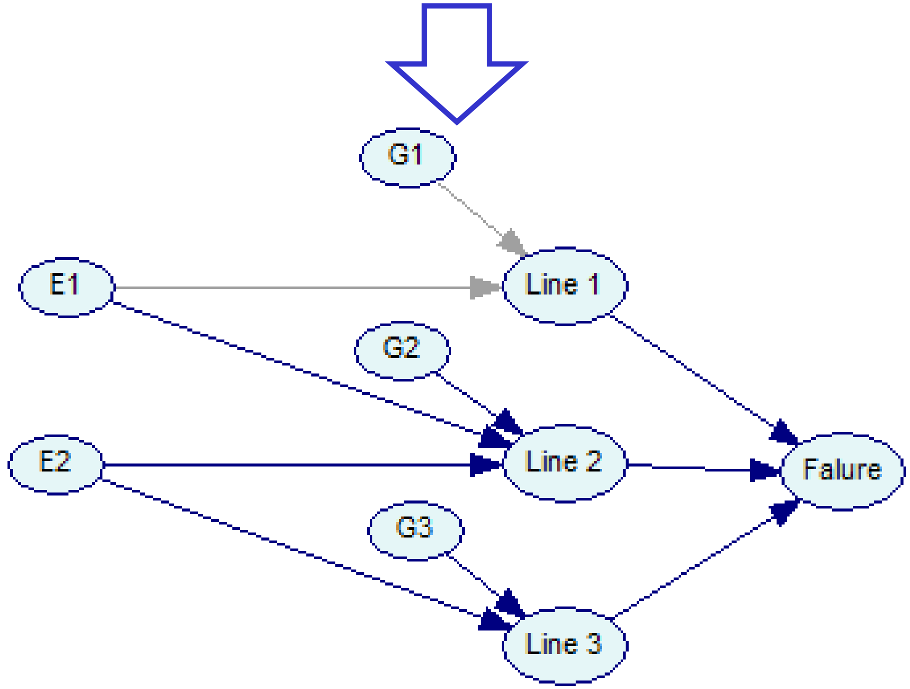



| G2 | Ok | | | | Failed | | | |
|---|---|---|---|---|---|---|---|---|
| E2 | Ok | | Failed | | Ok | | Failed | |
| E1 | Ok | Failed | Ok | Failed | Ok | Failed | Ok | Failed |
| PowerOk | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| PowerKo | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |




E1, E2 = engines

G1, G2, G3 = generators, each one is rated at 30 KVA



# Advantages of BN model

- Multi-state modeling

| G1 | Ok | | | Degraded | | | Failed | | |
|---|---|---|---|---|---|---|---|---|---|
| E1 | Ok | Degraded | Failed | Ok | Degraded | Failed | Ok | Degraded | Failed |
| PowerOk | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PowerDegraded | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| PowerKo | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |