Bayesian Networks
Representation and Reasoning

Marco F. Ramoni
Children's Hospital Informatics Program
Harvard Medical School

HST 951 (2003)

Harvard-MIT Division of Health Sciences and Technology
HST.951J: Medical Decision Support

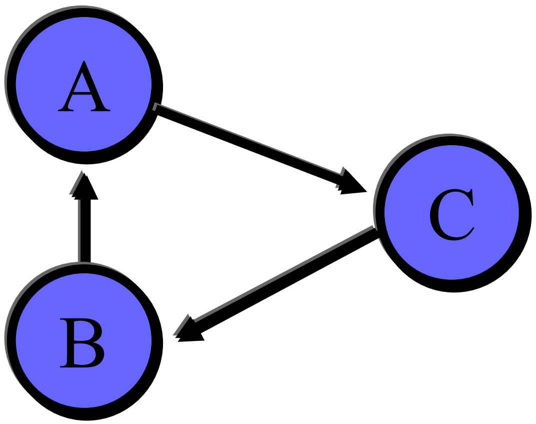
Introduction

- Bayesian network are a knowledge representation formalism for reasoning under uncertainty.
- A Bayesian network is a direct acyclic graph encoding assumptions of conditional independence.
- In a Bayesian network, nodes are stochastic variables and arcs are dependency between nodes.
- Bayesian networks were designed to encode explicitly encode “deep knowledge” rather than heuristics, to simplify knowledge acquisition, provide a firmer theoretical ground, and foster reusability.

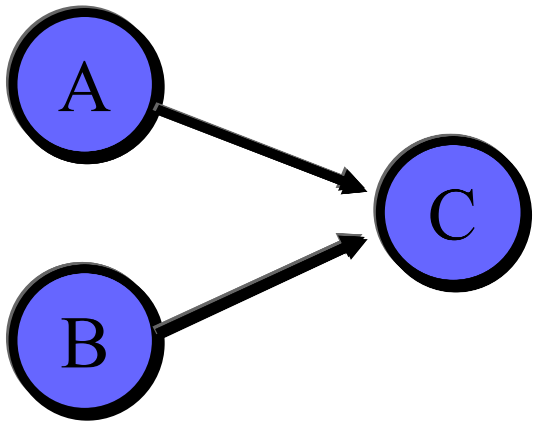
Graph

A graph (network) *G*(*N*,*L*) is defined by:

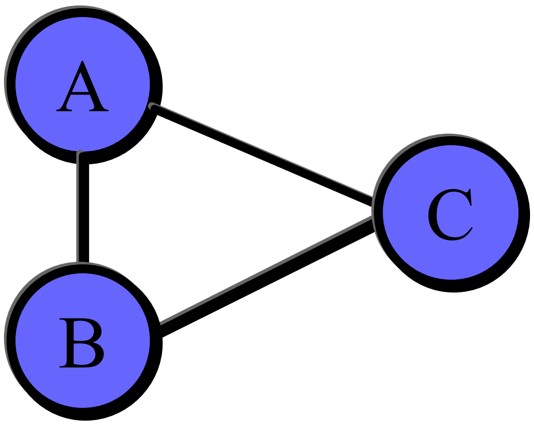
- Nodes: A finite set *N* = {*A*,*B*,...} of nodes (vertices).
- Arcs: A set *L* of arcs (edges): ordered pair of nodes.
- *Set L is a subset of all possible pairs of nodes N.*



L={(A,C),(B,C),(B,A)}

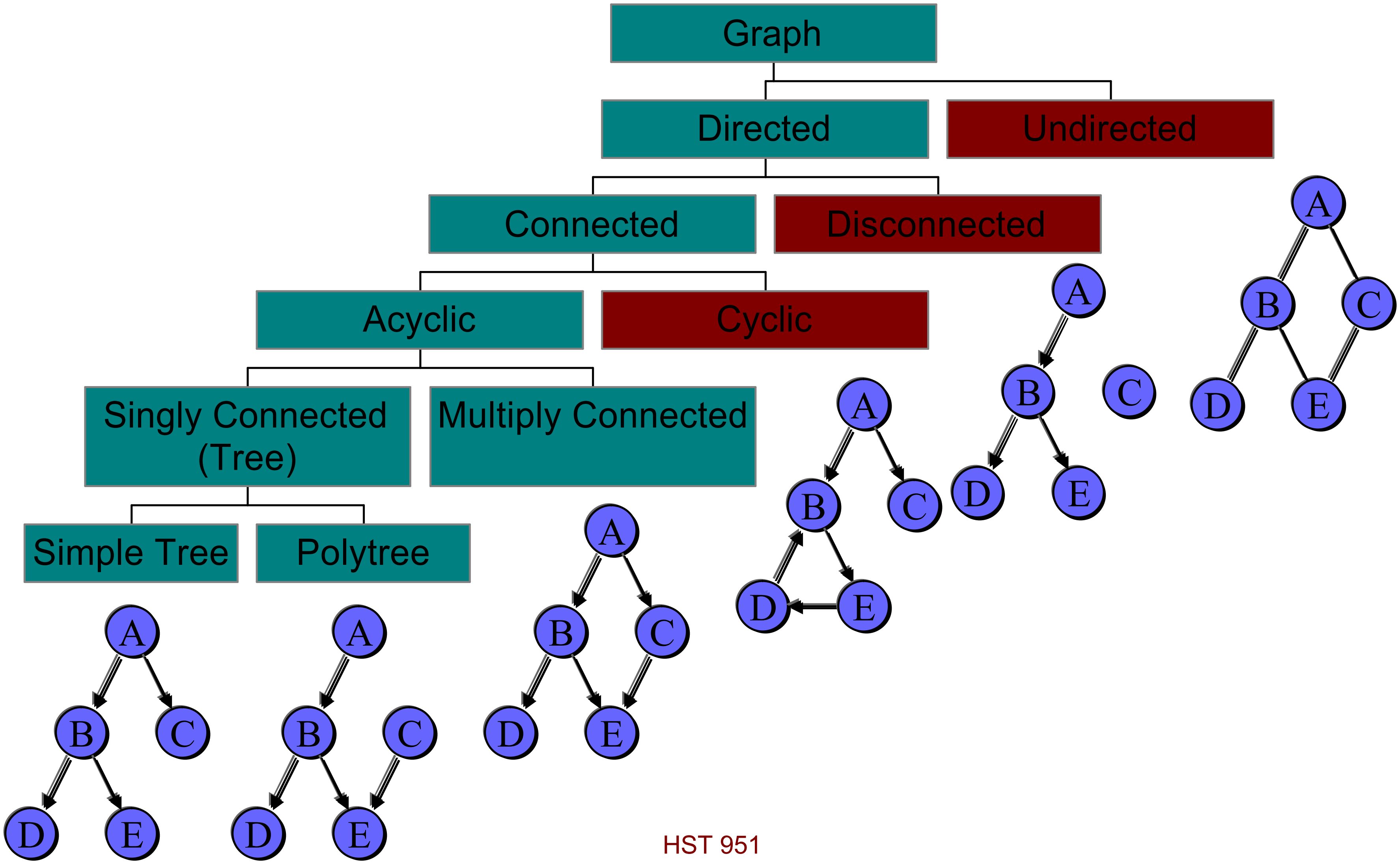


L={(A,C),(B,C)}

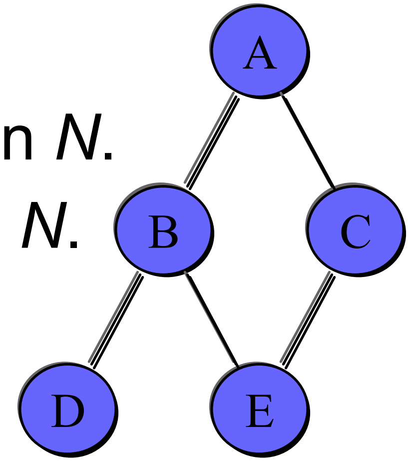


L={(A,C),(B,C),(B,A),(C,A),(C,B),(A,B)}

Types of Graph



Direction



Direction of a link:

- Directed: if (*A*,*B*) is in *N*, then (*B*,*A*) is not in *N*.
- Undirected: if (*A*,*B*) is in N, then (B,A) is in *N*.
- *Note: The link — should be ↔.*

Characters:

- Adjacent set: the nodes one step away from *A:*

$$Adj(A)=\{B|(A,B)\in L\}.$$

- Path: The set of *n* nodes X_i from A to *B* via links:
- Loop: A closed path: $X_1 = X_n$.
- Acyclic graph: A graph with no cycles.

Directed Graphs

Parent: A is parent of B if there is a directed link A→B.

Family: The set made up by a node and its parents.

Ancestor: A is ancestor of B if exists a path from A to B.

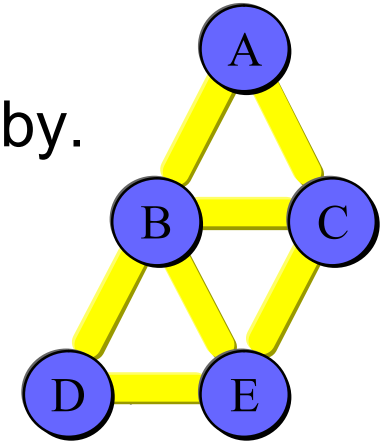
Ancestral set: A set of nodes containing their ancestors.

Cycle: A cycle is a closed loop of directed links.

Associated acyclic graph: The undirected graph obtained by dropping the direction of links.

Moral graph: The undirected graph obtained by.

- ✓ Marring the parents of a common child.
- ✓ Dropping the directions of the links.



Trees

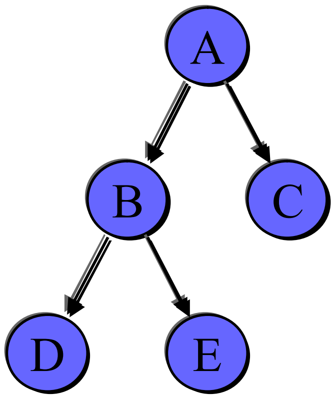
Tree: If every pair of nodes there is at most one path.

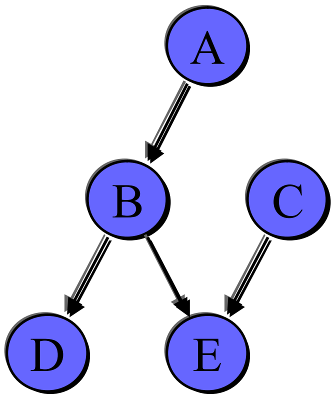
Simple Tree: Each node has at most one parent.

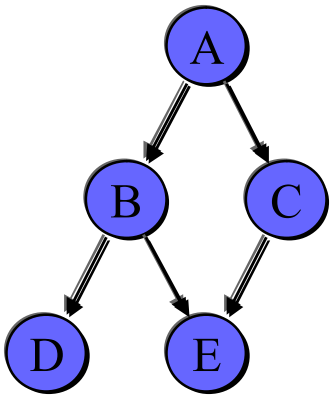
PolyTree: Nodes can have more than one parent.

Multiply Connected Graph: A graph where at least one pair of nodes has more than one path.

Note: The associated undirected graph has a loop.







Bayesian Networks

Qualitative: A dependency graph made by:

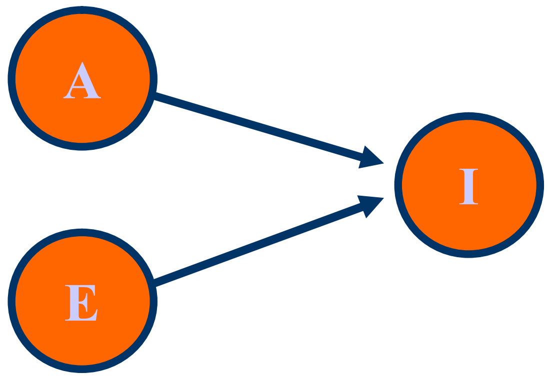
Node: a variable X, with a set of states $\{x_1,\ldots,x_n\}$.

Arc: a dependency of a variable X on its parents Π.

Quantitative: The distributions of a variable X given each combination of states π_i of its parents Π.

A	p(A)
Y	0.3
O	0.7

E	p(E)
L	0.8
H	0.2



A	E	I	p(I|A,E)
Y	L	L	0.9
Y	L	H	0.1
Y	H	L	0.5
Y	H	H	0.5
O	L	L	0.7
O	L	H	0.3
O	H	L	0.2
O	H	H	0.8

A=Age; E=Education; I=Income

Independence

- Perfect dependence between Disease and Test:

Test	Disease	
	0	1
0	100	0
1	0	100

Test	Disease	
	0	1
0	1	0
1	0	1

- Independence between Disease and Test:

Test	Disease	
	0	1
0	50	50
1	40	60

Test	Disease	
	0	1
0	0.5	0.5
1	0.4	0.6

Exercise: Compute the CPT for Test given Disease.

Why Do We Care?

- Independence simplifies models: if two variables are independent, I do not need to model their interaction but I can reason about them separately.
- In this form of independence, called marginal independence, however, a variable will tell me nothing about another variable, by design.
- There is another, more useful, form of independence, which maintains the connection between variables but, at the same time, breaks down the whole system in separate regions: conditional independence.
- This is independence used by Bayesian networks.

Conditional Independence

- When two variables are independent given a third, they are said to be conditionally independent.

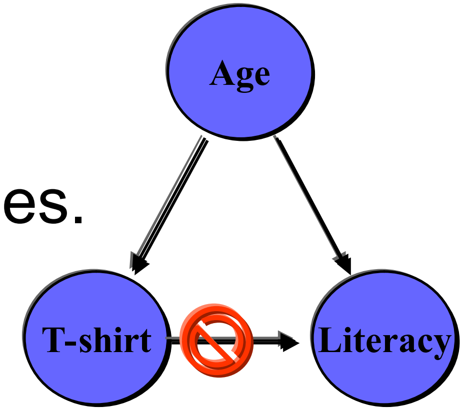
$$p(A|B \wedge C)=p(A \wedge B \wedge C)/p(B \wedge C)=p(A|C).$$

	Literacy	
T-shirt	Yes	No
Small	0.32	0.68
Large	0.35	0.65

		Literacy	
Age	T-shirt	Yes	No
<5	Small	0.3	0.7
<5	Large	0.3	0.7
>5	Small	0.4	0.6
>5	Large	0.4	0.6

Bayesian Networks

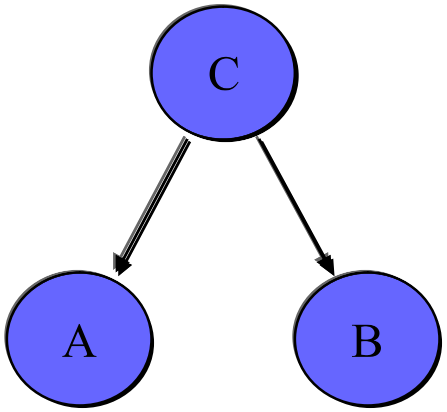
- Bayesian networks use graphs to capture these statement of conditional independence.
- A Bayesian network (BBN) is defined by a graph:
  - ✓ Nodes are stochastic variables.
  - ✓ Links are dependencies.
  - ✓ No link means independence given a parent.
- There are two components in a BBN:
  - ✓ Qualitative graphical structure.
  - ✓ Quantitative assessment of probabilities.



Decomposition

- BBNs decompose the joint probability distribution with the graph of conditional independence.
- Therefore, the graphical structure factorizes the joint probability distribution:

$$p(A \wedge B \wedge C) = p(A|C) \times p(B|C) \times p(C).$$

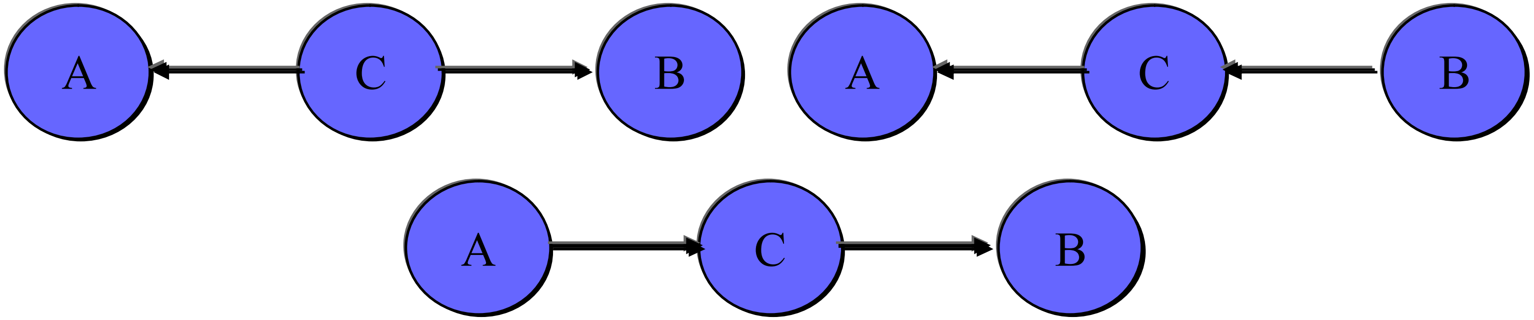


Markov Equivalence

- Different network structures may encode the same conditional independence statements:

  A and B are conditionally independent given C.

  can be encoded by 3 different network structures.
- In all these network structures, the information flow running between A and B along the direction of the arrows is mediated by the node C.



Example

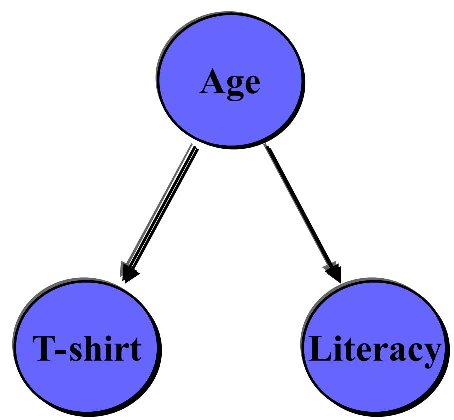
Background knowledge: General rules of behavior.

p(Age=<5)=0.3
p(T-shirt=small| Age=<5)=0.5
p(T-shirt=small|Age=>5)=0.3
p(Literacy=yes|Age=>5)=0.6
p(Literacy=yes|Age=<5)=0.2.

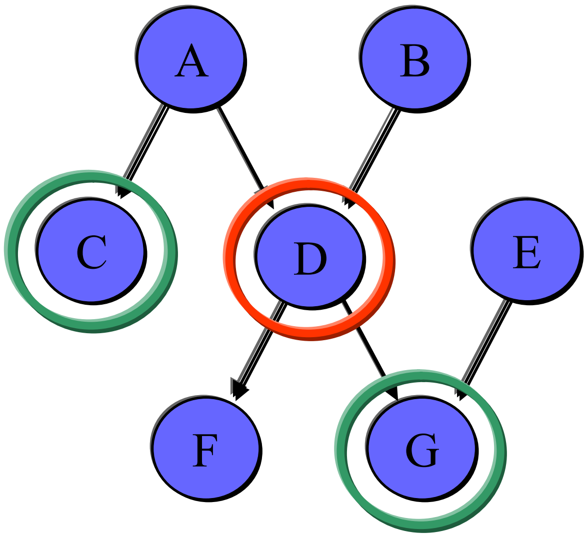
Evidence: Observation *p(T-shirt=small).*

Solution: The posterior probability distribution of the unobserved nodes given evidence: *p(Literacy| T-shirt=small)* and *p(Age| T-shirt=small).*

p(Age=<5,T-shirt=small,Literacy=yes)
p(Age=<5,T-shirt=small,Literacy=no)
p(Age=<5,T-shirt=large,Literacy=yes)
p(Age=<5,T-shirt=large,Literacy =no)
p(Age=>5,T-shirt=small,Literacy=yes)
p(Age=>5,T-shirt=small,Literacy=no)
p(Age=>5,T-shirt=large,Literacy=yes)
p(Age=>5,T-shirt=large, Literacy=no).



Reasoning



Components of a problem:

- Knowledge: graph and numbers.
- Evidence: e={c and g}.
- Solution: p(d|c,g)=?

Note: Lower case is an instance.

A	p(A)
0	0.3
1	0.7

B	p(B)
0	0.6
1	0.4

E	p(E)
0	0.1
1	0.9

A	C	p(C|A)
0	0	0.25
0	1	0.75
1	0	0.50
1	1	0.50

D	F	p(F|D)
0	0	0.80
0	1	0.20
1	0	0.30
1	1	0.70

A	B	D	p(D|A,B)
0	0	0	0.40
0	0	1	0.60
0	1	0	0.45
0	1	1	0.55
1	0	0	0.60
1	0	1	0.40
1	1	0	0.30
1	1	1	0.70

D	E	G	p(G|D,E)
0	0	0	0.90
0	0	1	0.10
0	1	0	0.70
0	1	1	0.30
1	0	0	0.25
1	0	1	0.75
1	1	0	0.15
1	1	1	0.85

Brute Force

- Compute the Joint Probability Distribution:

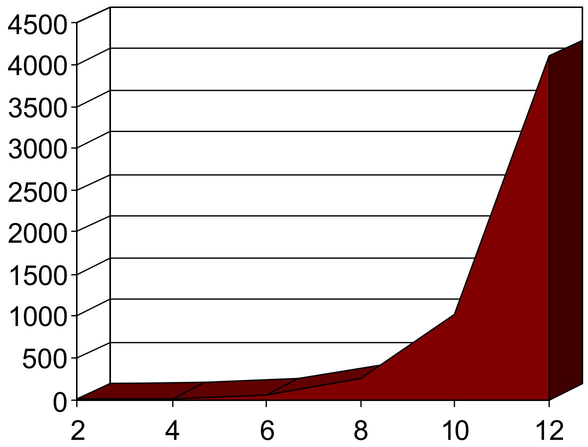
p(a,b,c,d,e,f,g)=p(a)p(b)p(c|d)p(d|a,b)p(e)p(f|d)p(g|d,e).

- Marginalize out the variable of interest:

$$p(d)=\Sigma\ p(a,b,c,e,f,g).$$

Note: We have replaced $\wedge$ with ,

Cost: 2^n probabilities ($2^6 = 64$).



Decomposition

Decomposition: D breaks the BBN into two BBNs:

$$p(d)= \Sigma\, p(a)p(b)p(c|a)p(d|a,b)p(e)p(f|d)p(g|d,e)=.$$

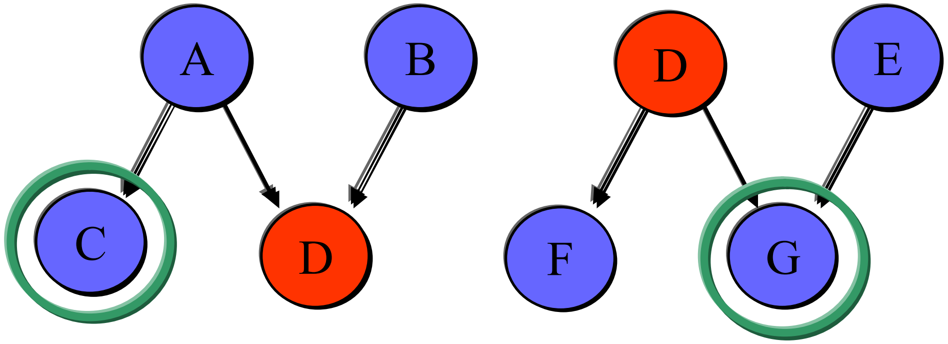
$$= (\Sigma\, p(a)p(b)p(c|a)p(d|a,b))\ (\Sigma\, p(e)p(f|d)p(g|d,e)).$$

Saving: We move from 64 to $2^3 + 2^3=16$, and most of all the terms move from 7 to 4 and from 7 to 3.

D-separation: the basic idea is based on a property of graphs called d-separation (directed-separation).

Propagation in Polytrees

- In a polytree, each node breaks the graph into two independent sub-graphs and evidence can be independently propagated in the two graphs:
  - ✓ E+: evidence coming from the parents (E+ = {c}).
  - ✓ E-: evidence coming from the children (E- = {g}).



Message Passing

- Message passing algorithm (Kim & Pearl 1983) is a local propagation method for polytrees.
- The basic idea is that p(d) is actually made up by parent component π(d) and a south component λ(d).
- The basic idea is to loop and pass π and λ messages between nodes until no message can be passed.
- In this way, the propagation is entirely distributed and the computations are locally executed in each node.

Algorithm

Input: A BBN with a set of variables X and a set of evidential statements ε = {A=a,B=b,…}.

Output: Conditional probability distribution $p(X|\varepsilon)$ for each non evidential variable X.

Initialization Step:

Each evidential variable X,

if $x \in$ e $p(x)=1$, else $p(x)=0$.

if $x \in$ e $l(x)=1$, else $l(x)=0$.

Each non evidential root variable X, $p(x) = \pi(x)$.

Each non evidential childless variable X, $\lambda(x)=1$.

Algorithm II

- Iteration Step (on non evidential variables X/e):
  - If X has all the π-messages from its parents, $\pi(x)$.
  - If X has all the λ-messages from its children, $\lambda(x)$.
  - If $\pi(x)$, for each child, if λ -messages from all other children are in, send π-message to child.
  - If $\lambda(x)$, for each parent, if π-messages from all other parents are in, send λ-message to parent.
  - Repeat until no message is sent.
- Closure:
  - ✓ For each X/e, compute $\beta(x)= \pi(x)\ \lambda(x)$.
  - ✓ For each X/e, compute $p(x)= \beta(x)/\Sigma\ \lambda(x_i)$.

Properties

Distributed: Each node does not need to know about the others when it is passing the information around.

Parallel architecture: Each node can be imagined as a separate processor.

Complexity: Linear in the number of nodes.

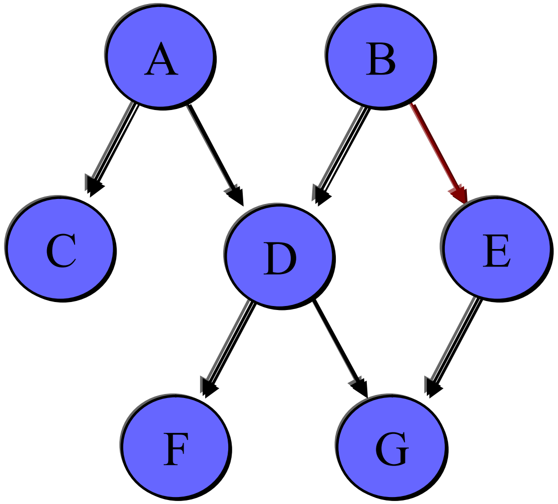
Limitations: Confined to a restricted class of graphs and, most of all, unable to represent an important class of problems.

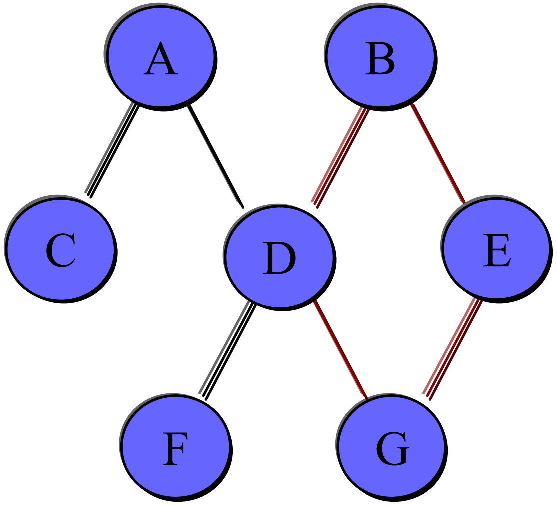
Importance: Proof of feasibility - Bayesians are not just dreamers after all.

Multiply Connected BBN

When the BBN is a multiply connected graph.

- The associated undirected graph contains a loop.
- Each node does not break the network into 2 parts.
- Information may flow through more than one paths.
- Pearl's Algorithm is no longer applicable.





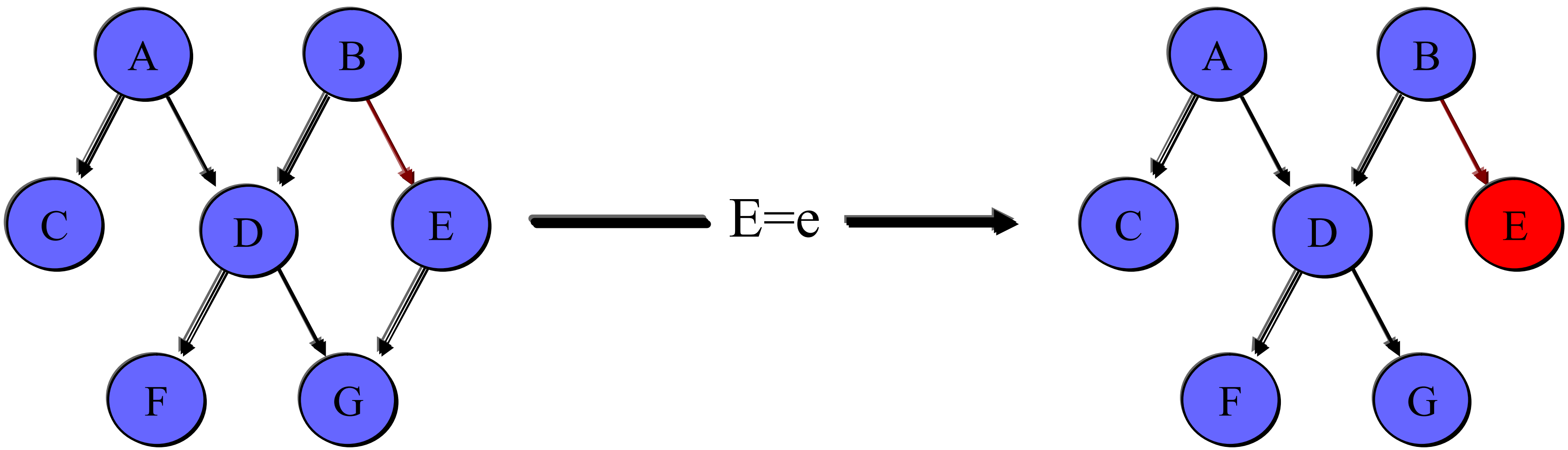
Methods

- Main stream methods:
  - ✓ Conditioning Methods.
  - ✓ Clustering Methods.
- The basic strategy is:
  - ✓ Turn multiply connected graph in something else.
  - ✓ Use Pearl's algorithm to propagate evidence.
  - ✓ Recover the conditional probability p(x|e) for X.
- Methods differ in the way in which.
  - ✓ What they transform the graph into.
  - ✓ The properties they exploit for this transformation.

Conditioning Methods

The transformation strategy is:

- ✓ Instantiate a set of nodes (cutset) to cut the loops.
- ✓ Absorb evidence and change the graph topology.
- ✓ Propagate each BBN using Pearl's algorithm.
- ✓ Marginalize with respect to the loop cutset.



Algorithm

Input: a (multiply connected) BBN and evidence e.

Output: the posterior probability p(x|e) for each X.

Procedure:

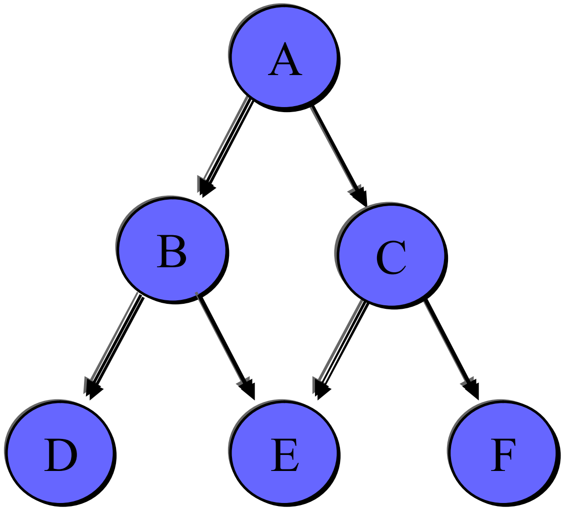
1. Identify a loop cutset $C=(C_1, \ldots, C_n)$.
2. For each member of combinations $c=(c_1, \ldots, c_n)$.
   - ☞ Generate a polytree BBNs for each c.
   - ☞ Use Pearl's Algorithm to compute $p(x|\varepsilon,c_1,\ldots,c_n)$.
   - ☞ Compute $p(c_1,\ldots, c_n| \varepsilon) = p(\varepsilon |c_1,\ldots,c_n)p(c_1,\ldots,c_n) /p(\varepsilon)$.
3. For each node X,
   - ☞ $\alpha=p(x|\varepsilon) \propto \Sigma_c p(x|\varepsilon,c_1,\ldots,c_n)p(\varepsilon|c_1,\ldots,c_n)p(c_1,\ldots,c_n)$,
   - ☞ Compute $p(x|e)= \alpha/\Sigma_x p(x)$.

Complexity

- The computational complexity is exponential in the size of the loop cutset, as we must generate and propagate a BBN for each combination of states of the loop cutset.
- The identification of the minimal loop cutset of a BBN is NP-hard, but heuristic methods exist to make it feasible.
- The computational complexity is a problem common to all methods moving from polytrees to multiply connected graphs.

Example

- A Multiply connected BBN
- No evidence



A	p(A)
0	0.3
1	0.7

A	B	p(B|A)
0	0	0.4
0	1	0.6
1	0	0.1
1	1	0.9

A	C	p(C|A)
0	0	0.2
0	1	0.8
1	0	0.50
1	1	0.50

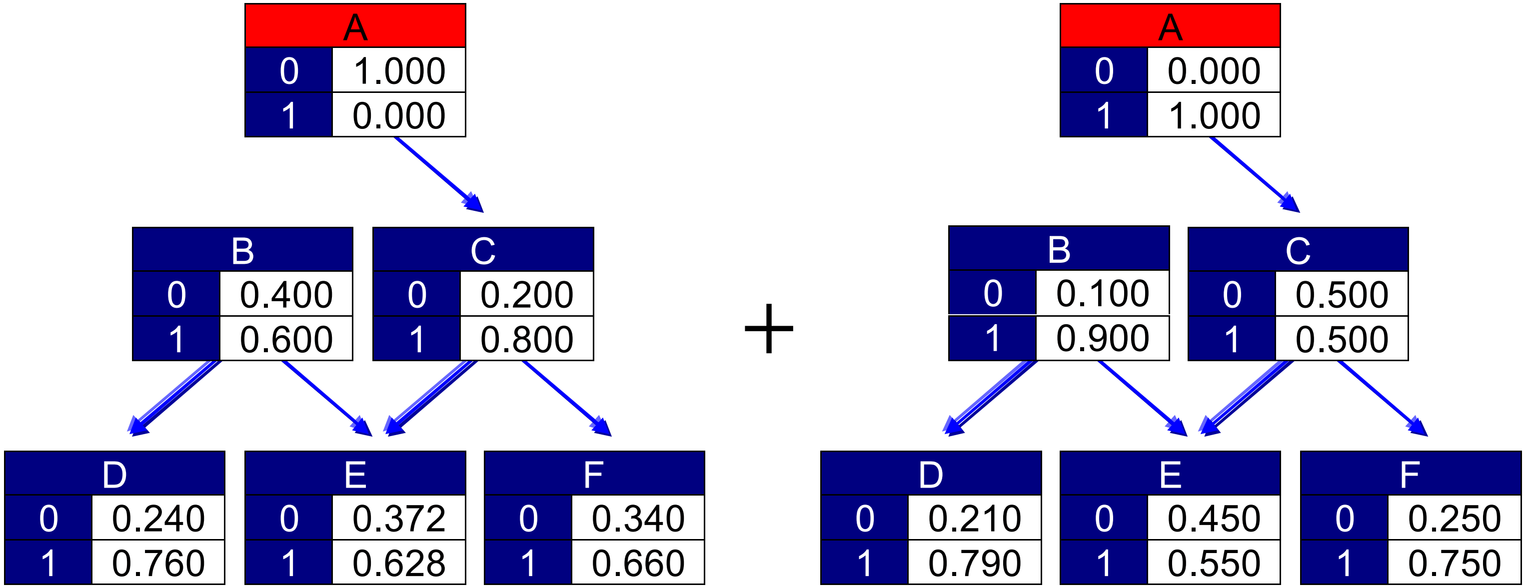
B	D	p(D|B)
0	0	0.3
0	1	0.7
1	0	0.2
1	1	0.8

C	F	p(F|C)
0	0	0.1
0	1	0.9
1	0	0.4
1	1	0.6

B	C	E	p(E|B,C)
0	0	0	0.4
0	0	1	0.6
0	1	0	0.5
0	1	1	0.5
1	0	0	0.7
1	0	1	0.3
1	1	0	0.2
1	1	1	0.8

Example

- Loop cutset: {A}.
- p(B=0)=p(B=0|A=0)p(A=1) + p(B=0|A=1)p(A=1).



	A	B	C	D	E	F
0	0.300	0.190	0.410	0.219	0.427	0.277
1	0.700	0.810	0.590	0.781	0.573	0.723

Clustering Methods

The basic strategy (Lauritzen & Spiegelhalter 1988) is:

1. Convert a BBN in a undirected graph coding the same conditional independence assumptions.
2. Ensure the resulting graph is decomposable.
3. This operation clusters nodes in locally independent subgraphs (cliques).
4. These cliques are joint to each other via a single nodes.
5. Produce a perfect numbering of nodes.
6. Recursively propagate evidence.

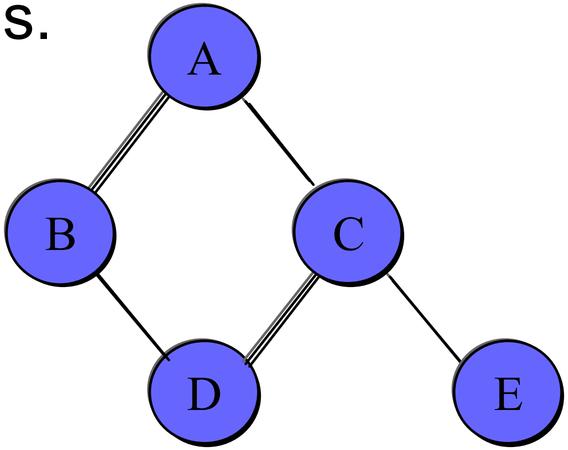
Markov Networks

- A Markov network is a based on undirected graphs:

  BBN : DAG = Markov Network : Undirected Graph.

- Markov networks encode conditional independence assumptions (as BBNs) using a Undirected Graph:
  1. A link between A and B means dependency.
  2. A variable is independent of all not adjacent variables given the adjacent ones.

Example: E is independent from (A,B,D) given C.



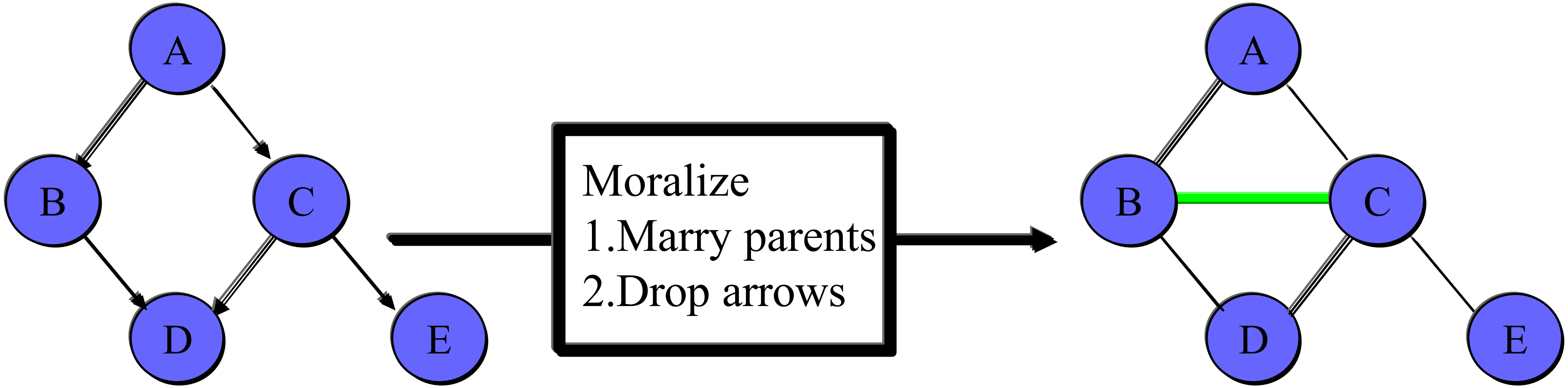
Decomposable

- Decomposable Markov networks lead to efficiency:
  - ✓ A Markov network is said to be decomposable when it contains no cycle with longer than 3 (there is no unbroken cycle with more than 3 nodes).
- The joint probability distribution of the graph can be factorized by the marginal distributions of the cliques:
  - ✓ A clique is the largest sub-graph in which nodes are all adjacent to each other.
  - ✓ Therefore, a clique cannot be further simplified by conditional independence assumptions.

Triangulation

- When a Markov network is not decomposable, we triangulate the graph by including the missing links.
- The product of the joint probability of each clique, divided by the product of their intersection:

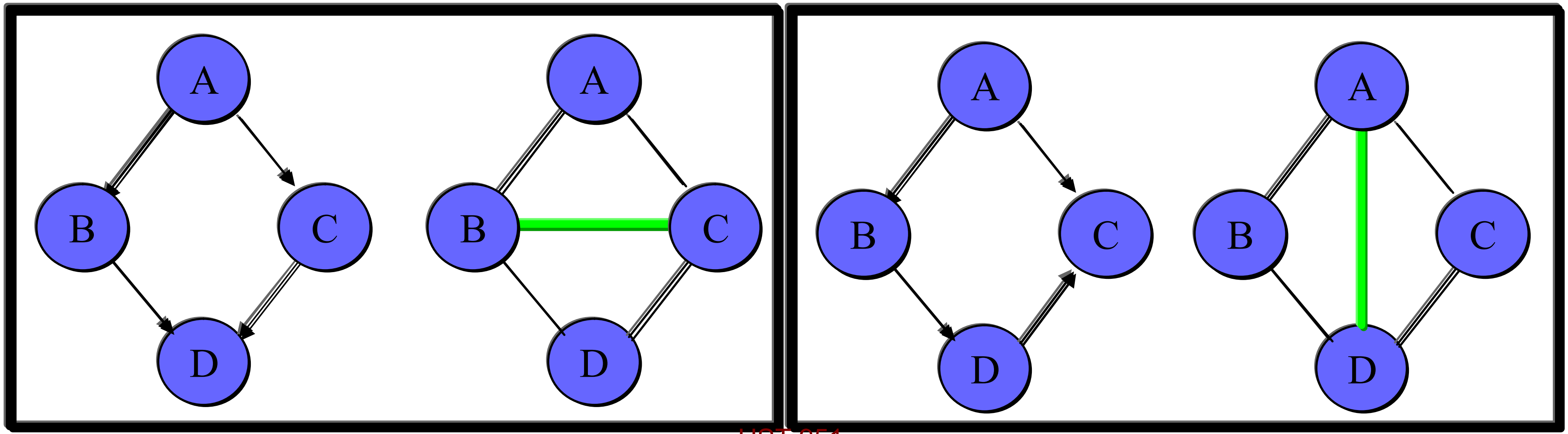
$$p(a,b,c)=p(c|a)p(b|a)p(a).$$



Reading Independence

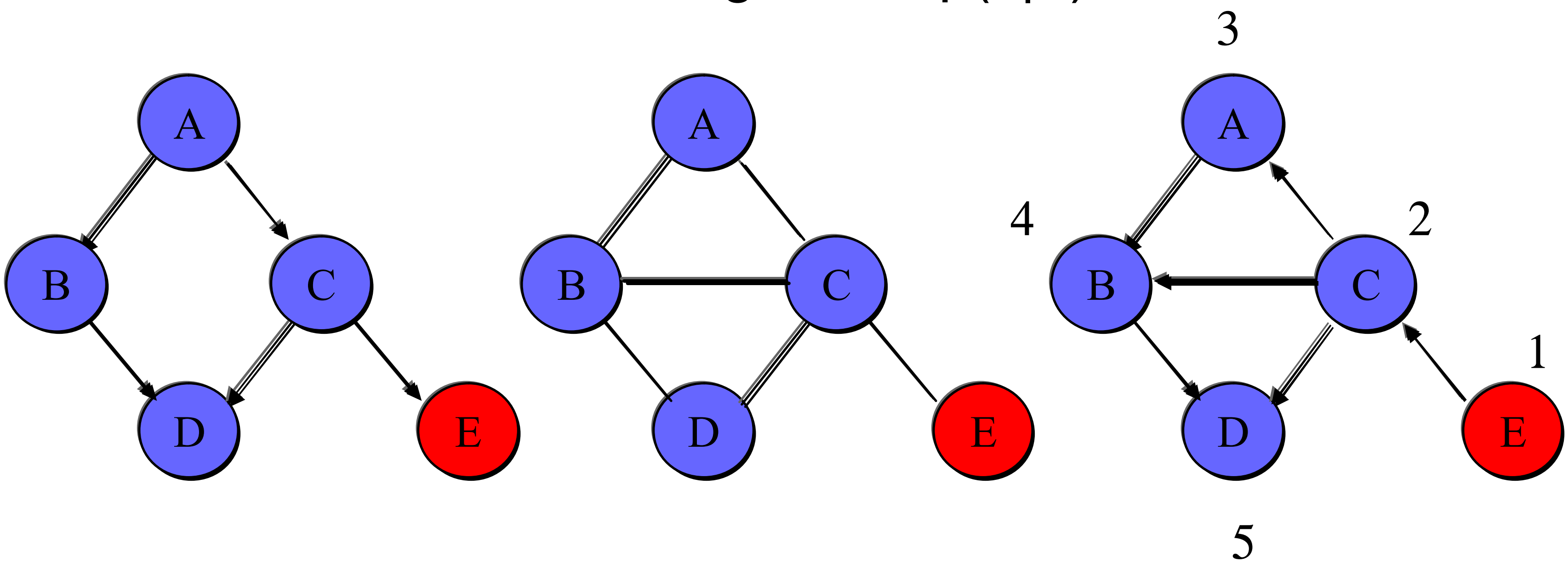
- The translation method via moralization reads the conditional independence statements in BBN.
- DAGs cannot encode any arbitrary set of conditional independence assumptions.

I(D,A|(B,C)) I(C,B|(A,D))



Propagation

- Compile the BBN into a moralized Markov network.
- Maximum cardinality search:
- For each clique Q compute p(q|e).
- Within each cluster, marginalize p(x|e).



Who is the Winner?

- Clustering is also NP-complete. The source of computational complexity is the size of the larger clique in the graph.
- Global conditioning (Shachter, Andersen & Szolovits 1994) shows that:
  1. Conditioning is a special case of Clustering.
  2. Conditioning is better at trading off memory-time.
  3. Conditioning is better suited for parallel implementations.