Bayesian Networks

Prof. Dr. Rudolf Kruse,
Pascal Held

Computational Intelligence Group
Department of Knowledge Processing and Language Engineering
Faculty of Computer Science
kruse@iws.cs.uni-magdeburg.de

About me: Rudolf Kruse

- in 1979 diploma in mathematics (minor computer science) at TU Braunschweig
- there dissertation in 1980, rehabilitation in 1984
- 2 years full-time employee at Fraunhofer Institute
- in 1986 offer of professorship for computer science at TU Braunschweig
- since 1996 professor at the University of Magdeburg
- **research:** data mining, explorative data analysis, fuzzy systems, neuronal networks, evolutionary algorithms, Bayesian networks
- `mailto:kruse@iws.cs.uni-magdeburg.de`
- office: G29-008, telephone: 0391 67-58706
- consultation: Wednesdays, 11 a.m. – 12 noon

About the working group Computational Intelligence

teaching:

- Intelligent Systems — Bachelor (2 V + 2 Ü, 5 CP)
- Evolutionary Algorithms — Bachelor (2 V + 2 Ü, 5 CP)
- Neuronal Networks — Bachelor (2 V + 2 Ü, 5 CP)
- Fuzzy Systems — Master (2 V + 2 Ü, 6 CP)
- Bayesian Network — Master (2 V + 2 Ü, 6 CP)
- Intelligent Data Analysis — Master (2 V + 2 Ü, 6 CP)
- (pro-)seminars: Information Mining, Computational Intelligence

research examples:

- dynamic graph analysis in brain networks (C. Moewes)
- analysis of social networks (P. Held)
- planet search by astronomical data analysis (C. Braune)

About the lecture

- lecture dates: Thursday, 3:15 p.m.–4:45 p.m., G22A-218
- information about the course:
  `http://fuzzy.cs.ovgu.de/wiki/pmwiki.php?n=Lehre.BN1213`
  - weekly lecture slides as PDF
  - also assignment sheets for the exercise
  - important announcements and date!

Content of the lecture

- Introduction
- Rule-based Systems
- Elements of Graph Theory
- Decomposition
- Probability Foundations
- Applied Probability Theory
- Probabilistic Causal Networks
- Propagation in Belief Networks
- Learning Graphical Models
- Decision Graphs / Influence Diagrams
- Frameworks of Imprecision and Uncertainty

About the exercise

- active participation and explanations of your solutions
- tutor will call attention to mistakes and answer questions
- pure 'calculations' of sample solution is not the purpose
- tutor: Pascal Held `mailto:pheld@ovgu.de`
- consultation: Just knock on the door and see if he is there :-)
- first assignment due October 18, 2012
- Thursday, 3:15 p.m.–4:45 p.m., G22A-208

Conditions for Certificate (“Schein”) and Exam

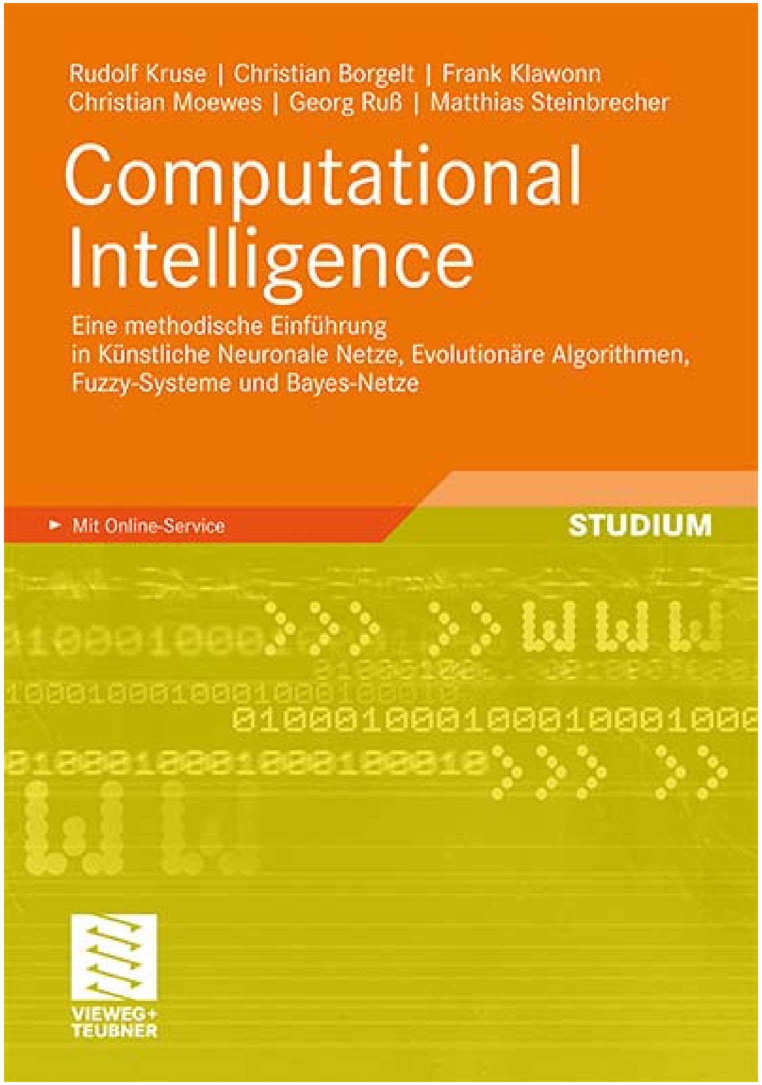
Certificate will get who...

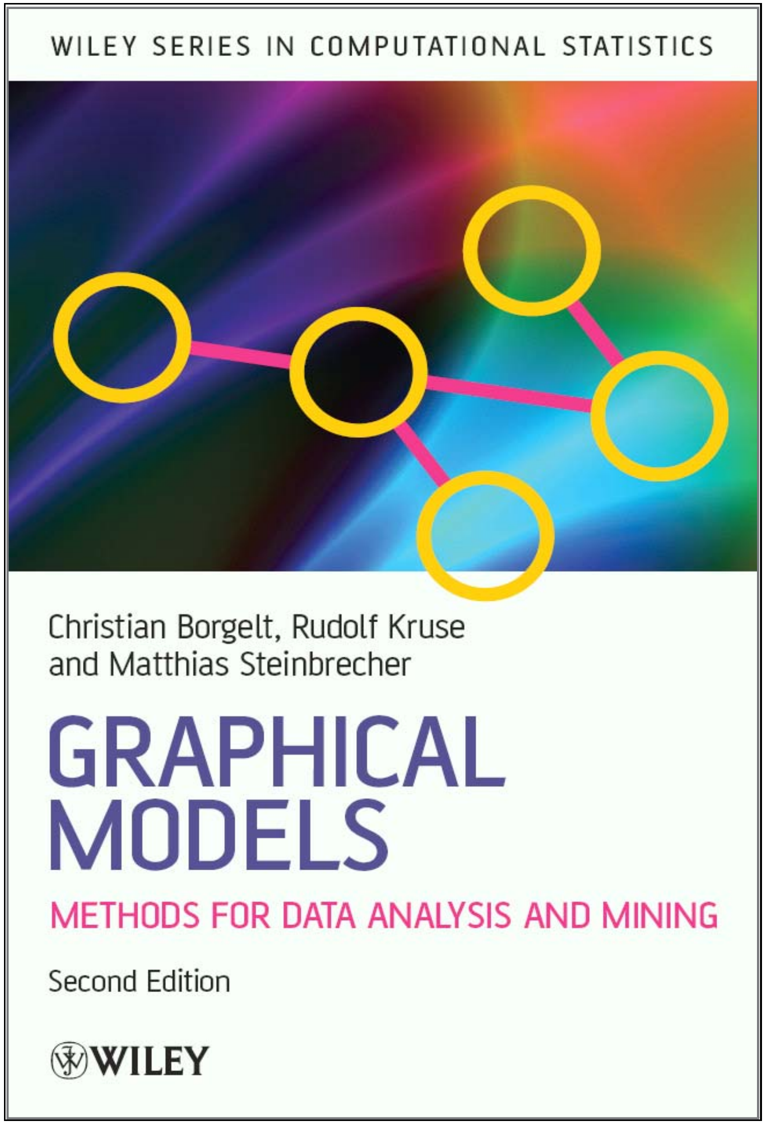
- contribute well in exercises every week,
- present ≥ 2 solutions to written assignment during exercises.
- tick off $\geq 66\%$ of all written assignments,
- small colloquium (≈ 10 min.) or written test (if > 20 students).

Exam or marked certificate will get who...

- just pass the oral exam (≈ 25 minutes) or written exam (if > 20 students).
- active participation in the exercises will help getting a good grade ;-)

Books about the course





http://www.computational-intelligence.eu/

- **Human Expert**

  A human *expert* is a specialist for a specific differentiated application field who creates solutions to customer problems in this respective field and supports them by applying these solutions.

- **Requirements**
  - Formulate precise problem scenarios from customer inquiries
  - Find correct and complete solution
  - Understandable answers
  - Explanation of solution
  - Support the deployment of solution

Knowledge Based Systems (2)

- **"Intelligent" System**

  An intelligent system is a program that models the knowledge and inference methods of a human expert of a specific field of application.

- **Requirements for construction:**
  - Knowledge Representation
  - Knowledge Acquisition
  - Knowledge Modification

Qualities of Knowledge

In most cases our knowledge about the present world is

- **incomplete/missing** (knowledge is not comprehensive)
  - e. g. "I don't know the bus departure times for public holidays because I only take the bus on working days."
- **vague/fuzzy/imprecise** (knowledge is not exact)
  - e. g. "The bus departs roughly every full hour."
- **uncertain** (knowledge is unreliable)
  - e. g. "The bus departs probably at 12 o'clock."

We have to decide nonetheless!

- Reasoning under Vagueness
- Reasoning with Probabilities
- . . . and Cost/Benefit

Example

Objective: *Be at the university at 9:15 to attend a lecture.*

- There are several plans to reach this goal:
  - P_1: Get up at 8:00, leave at 8:55, take the bus at 9:00 ...
  - P_2: Get up at 7:30, leave at 8:25, take the bus at 8:30 ...
  - ...
- All plans are *correct*, but
  - they imply different *costs* and different *probabilities* to *actually* reach that goal.
  - P_2 would be the plan of choice as the lecture is important and the success rate of P_1 is only about 80–95%.
- Question: *Is a computer capable of solving these problems involving uncertainty?*

Uncertainty and Rules (1)

- Example: We are given a simple expert system for dentists that may contain the following rule:

$$\forall p : [\text{Symptom}(p, \textsf{toothache}) \Rightarrow \text{Disease}(p, \textsf{cavity})]$$

- This rule is *incorrect*! Better:

$$\forall p : \Big[\text{Symptom}(p, \textsf{toothache}) \Rightarrow \text{Disease}(p, \textsf{cavity}) \lor \text{Disease}(p, \textsf{gumdisease}) \lor \ldots \Big]$$

- Maybe take the *causal* rule?

$$\forall p : \Big[\text{Disease}(p, \textsf{cavity}) \Rightarrow \text{Symptom}(p, \textsf{toothache})\Big]$$

- Incorrect, too.

Uncertainty and Rules (2)

Problems with propositional logic:

- We cannot enumerate all possible causes, even though ...
- We do not know the (medical) cause-effect interactions, and even though ...
- Uncertainty about the patient remains:
  - Caries and toothache may co-occurr by chance.
  - Were (exhaustively) all examinations conducted?
    — If yes: correctly?
  - Did the patient answer all questions?
    — If yes: appropriately?
- Without perfect knowledge no correct logical rules!

Example:

- We would like to support a robot's localization by fixed landmarks. From the presence of a landmark we may infer the location.

Problem:

- Sensors are imprecise!
    - We cannot conclude definitely a location simply because there was a landmark detected by the sensors.
    - The same holds true for undetected landmarks.
    - Only probabilities are being increased or decreased.

Degrees of Belief

- We (or other agents) are only believing facts or rules to some extent.
- One possibility to express this *partial belief* is by using *probability theory*.
- "The agent believes the sensor information to 0.9" means:
  In 9 out of 10 cases the agent trusts in the correctness of the sensor output.
- Probabilities gather the "uncertainty" that originates due to ignorance.
- Probabilities $\neq$ Vagueness/Fuzziness!
  - The predicate "large" is fuzzy whereas "This might be Peter's watch." is uncertain.

Rational Decisions under Uncertainty

- Choice of several *actions* or *plans*
- These may lead to different results with different *probabilities*.
- The *actions* cause different (possibly subjective) *costs*.
- The *results* yield different (possibly subjective) *benefits*.
- It would be rational to choose that action that yields the largest total benefit.

$$\text{Decision Theory} = \text{Utility Theory} + \text{Probability Theory}$$

Decision-theoretic Agent

input perception
output action

$K \leftarrow$ a set of probabilistic beliefs about the state of the world
calculate updated probabilities for current state based on available evidence including current percept and previous action
calculate outcome probabilities for actions, given action descriptions and probabilities of current states
select action A with highest expected utility given probabilities of outcomes and utility information
return A

Decision Theory: An agent is rational if and only if it chooses the action yielding the largest utility averaged over all possible outcomes of all actions.

Rule-based Systems

Modi of usage:

- **Query:** Facts are retrieved from database or user is interrogated
- **Explanation:** System answers questions how a decision was concluded

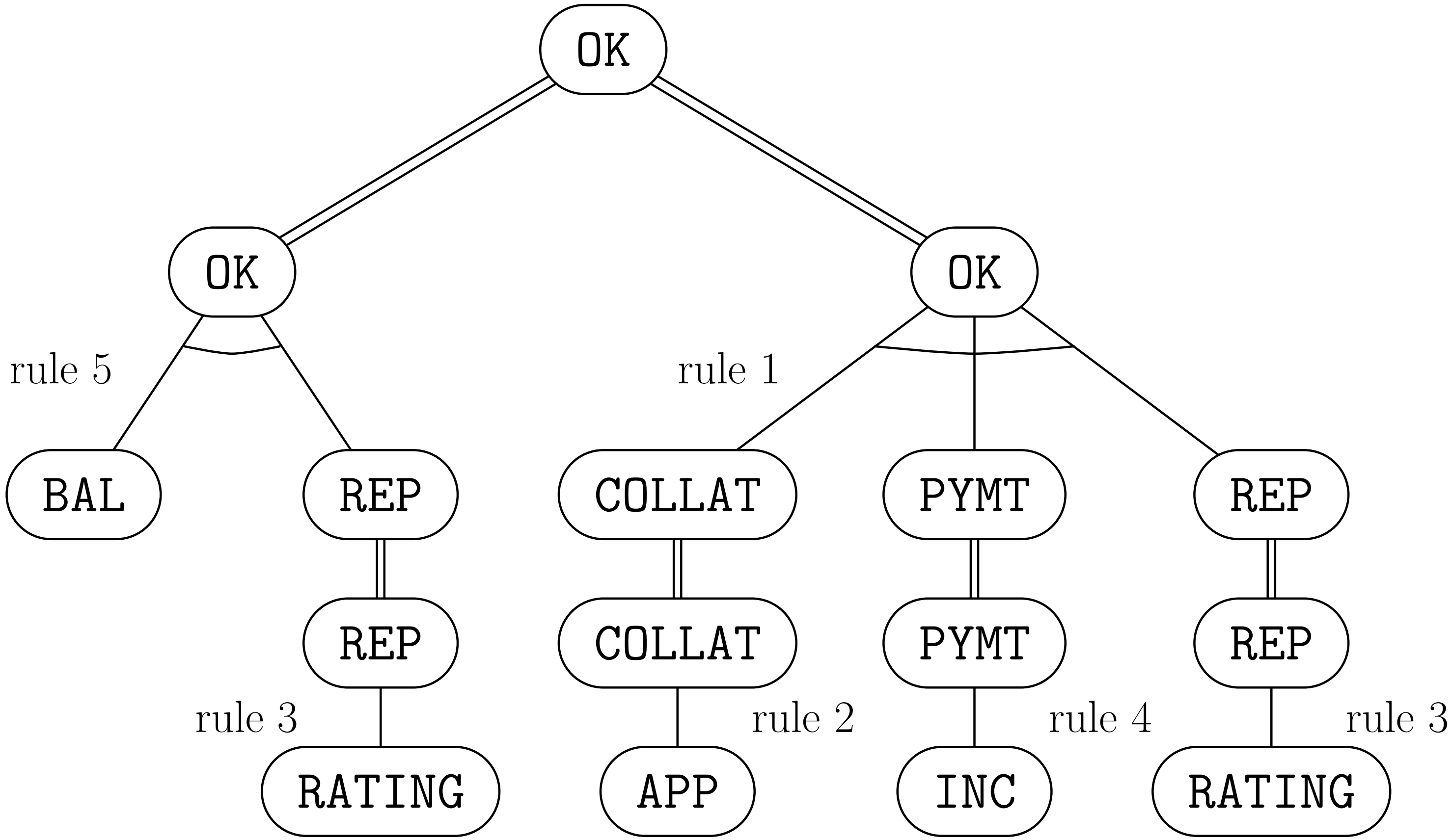
Example rule base:

1. COLLAT $\wedge$ PYMT $\wedge$ REP $\Rightarrow$ OK
2. APP $\Rightarrow$ COLLAT
3. RATING $\Rightarrow$ REP
4. INC $\Rightarrow$ PYMT
5. BAL $\wedge$ REP $\Rightarrow$ OK

Description:

COLLAT	satisfactory collateral
PYMT	payments undisputed
REP	good reputation
APP	high appraisal
RATING	good rating
INC	positive income
BAL	excellent balance
OK	loan will be approved

Corresponding `and/or`-Tree:



Example Queries

- **User:** Why do you believe that the loan should be approved?
- **System:** Because the collateral for the loan is satisfactory, and the applicant is able to make the loan payments, and the applicant has a good financial reputation.
- **User:** Why did you ask me about whether or not the applicant's income exceeds his/her expenses?
- **System:** Because I was trying to establish whether or not the applicant is able to make the loan payments.
- **User:** Why do you believe that the applicant has a good financial reputation?
- **System:** Because the credit rating service said that the applicant has a good credit rating.

Rules

- A rule in general is a *if-then*-construct consisting of a *condition* and an *action*.

$$\boxed{\text{If } \textit{condition} \text{ then } \textit{conclusion}}$$

- These two parts may be interpreted differently according to the context:
  - **Inference rules**: If *premise* then *conclusion*
  - **Hypotheses:** If *evidence* then *hypothesis*
  - **Productions:** If *condition* then *action*

- Rules are often referred to as *productions* or *production rules*.

Rules

- A rule in the ideal case represents a unit of knowledge.
- A set of rules together with an execution/evaluation strategy comprises a program to find solutions to specific problem classes.
- Prolog program: rule-based system
- Rule-based systems are historically the first types of AI systems and were for a long time considered prototypical expert systems.
- Nowadays, not every expert systems uses rules as its core inference mechanism.
- Rising importance in the field of business process rules.

Rule Evaluation

Forward chaining

- Expansion of knowledge base: as soon as new facts are inserted the system also calculates the conclusions/consequences.
- Data-driven behavior
- Premises-oriented reasoning: the chaining is determined by the left parts of the rules.

Backward chaining

- Answering queries
- Demand-driven behavior
- Conclusion-oriented reasoning: the chaining is determined by the right parts of the rules.

Components of a Rules-based System

Data base

- Set of structured data objects
- Current state of modeled part of world

Rule base

- Set of rules
- Application of a rule will alter the data base

Rule interpreter

- Inference machine
- Controls the program flow of the system

Rule Interpretation

- Main scheme forward chaining
  - Select and apply rules from the set of rules with valid antecedences. This will lead to a modified data base and the possibility to apply further rules.
- Run this cycle as long as possible.
- The process terminates, if
  - there is no rule left with valid antecendence
  - a solution criterion is satisfied
  - a stop criterion is satisfied (e. g. maximum number of steps)
- Following tasks have to be solved:
  - Identify those rules with a valid condition
    $\Rightarrow$ **Instantiation** or **Matching**
  - Select rules to be executed
    $\Rightarrow$ need for **conflict resolution**
    (e. g. via partial or total orderings on the rules)

Certainty Factors

Mycin (1970)

- **Objective:** Development of a system that supports physicians in diagnosing bacterial infections and suggesting antibiotics.
- **Features:** Uncertain knowledge was represented and processed via *uncertainty factors*.
- **Knowledge:** 500 (uncertain) decision rules as static knowledge base.
- **Case-specific knowledge:**
  - static: patients' data
  - dynamic: intermediate results (facts)
- **Strengths:**
  - diagnosis-oriented interrogation
  - hypotheses generation
  - finding notification
  - therapy recommendation
  - explanation of inference path

Uncertainty Factors

- Uncertainty factor $\quad \mathrm{CF} \in [-1, 1] \quad \approx \quad$ degree of belief.

- Rules:

$$\mathrm{CF}(A \to B) \begin{cases} = 1 & B \text{ is certainly true given } A \\ > 0 & A \text{ supports } B \\ = 0 & A \text{ has no influence on } B \\ < 0 & A \text{ provides evidence against } B \\ = -1 & B \text{ is certainly false given } A \end{cases}$$

A Mycin Rule

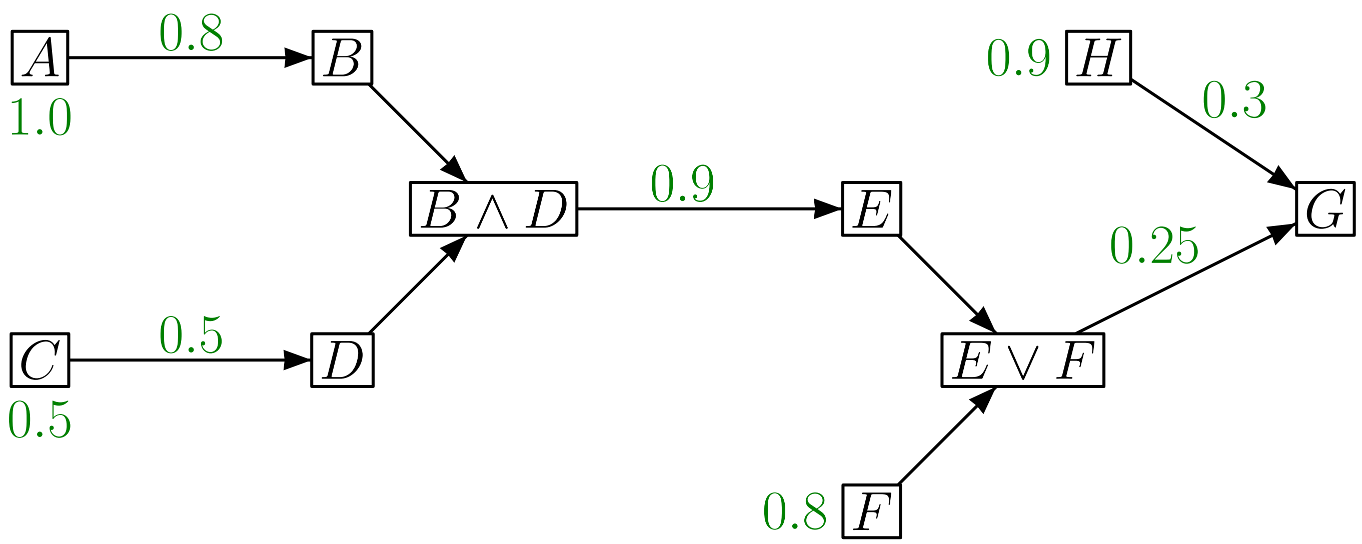
```
RULE035

PREMISE:   ($AND    (SAME CNTXT GRAM GRAMNEG)
                    (SAME CNTXT MORPH ROD)
                    (SAME CNTXT AIR ANAEROBIC))
ACTION:    (CONCL.CNTXT IDENTITY BACTEROIDES TALLY .6)
```

If	1) the *gram stain* of the organism is *gramneg*, and 2) the *morphology* of the organism is *rod*, and 3) the *aerobicity* of the organism is *anaerobic*
then	there is suggestive evidence (0.6) that the *identity* of the organism is *bacteroides*

Example

$$
\begin{aligned}
A &\to B\,[0.80] \\
C &\to D\,[0.50] \\
B \wedge D &\to E\,[0.90] \\
E \vee F &\to G\,[0.25] \\
H &\to G\,[0.30]
\end{aligned}
\qquad\qquad
\begin{aligned}
A\,&[1.00] \\
C\,&[0.50] \\
F\,&[0.80] \\
H\,&[0.90]
\end{aligned}
$$



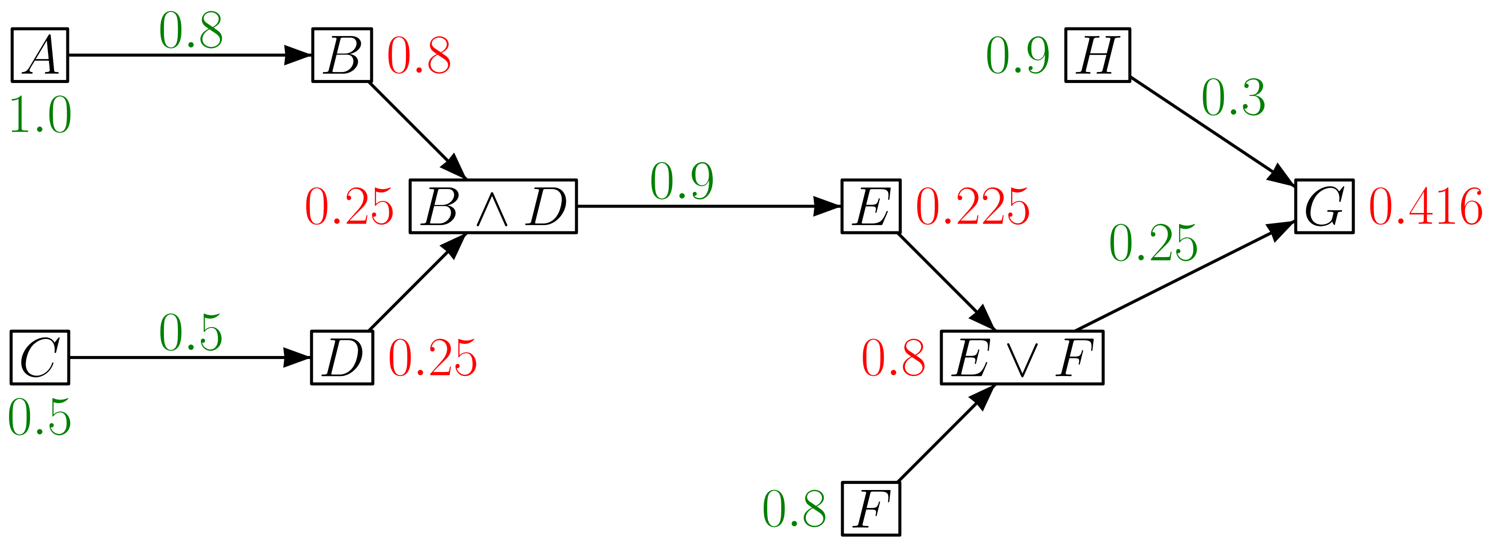
Propagation Rules

- **Conjunction:** $\mathrm{CF}(A \wedge B) = \min\{\mathrm{CF}(A), \mathrm{CF}(B)\}$
- **Disjunction:** $\mathrm{CF}(A \vee B) = \max\{\mathrm{CF}(A), \mathrm{CF}(B)\}$
- **Serial Combination:** $\mathrm{CF}(B, \{A\}) = \mathrm{CF}(A \to B) \cdot \max\{0, \mathrm{CF}(A)\}$
- **Parallel Combination:** for $n > 1$:
  $\mathrm{CF}(B, \{A_1, \ldots, A_n\}) =$
  $\quad f(\mathrm{CF}(B, \{A_1, \ldots, A_{n-1}\}), \mathrm{CF}(B, \{A_n\}))$

with

$$f(x, y) = \begin{cases} x + y - xy & \text{if} \quad x, y > 0 \\ x + y + xy & \text{if} \quad x, y < 0 \\ \dfrac{x + y}{1 - \min\{|x|, |y|\}} & \text{otherwise} \end{cases}$$

Example (cont.)



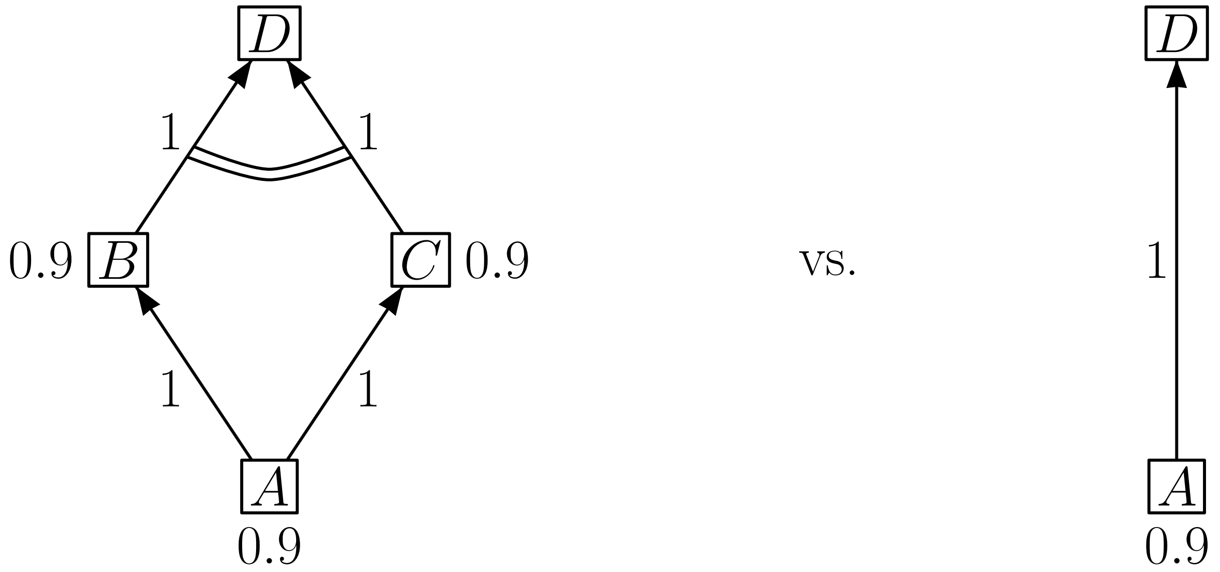
$$f(0.3 \cdot 0.9, 0.25 \cdot 0.8) \;=\; 0.27 + 0.2 - 0.27 \cdot 0.2 = 0.416$$

Was Mycin a failure?

- It worked in the Mycin case because the rules had tree-like structure.
- It can be shown that the rule combination scheme is inconsistent in general.

Example: $\mathrm{CF}(A) = 0.9$, $\mathrm{CF}(D) = ?$

$\mathrm{CF}(D) = 0.9 + 0.9 - 0.9 \cdot 0.9 = 0.99$ $\qquad$ $\mathrm{CF}(D) = 0.9$



Certainty factor is increased just because (the same) evidence is transferred over different (parallel) paths!

Was Mycin a failure?

Mycin was never used for its intended purpose, because

- physicians were distrustful and not willing to accept Mycin's recommendations.
- Mycin was too good.

However,

- Mycin was a milestone for the development of expert systems.
- it gave rise to impulses for expert system development in general.

Probabilistic Rules

How to assign probabilities to rules (implications)?

$$\boxed{P(B \mid A) \;\leq\; P(A \to B) = P(\neg A \lor B)}$$

A	B	$P(\cdot)$
0	0	0.04
0	1	0.95
1	0	0.01
1	1	0

$P(B \mid A) = 0$, but $P(A \to B) = 0.99$!

In the following, probabilistic rules are evaluated with conditional probabilities.

Elements of Graph Theory

Simple Graph

A simple graph (or just: graph) is a tuple $\mathcal{G} = (V, E)$ where

$$V = \{A_1, \ldots, A_n\}$$

represents a finite set of **vertices** (or **nodes**) and

$$E \subseteq (V \times V) \setminus \{(A, A) \mid A \in V\}$$

denotes the set of **edges**.
It is called simple since there are no self-loops and no multiple edges.

Edge Types

Let $\mathcal{G} = (V, E)$ be a graph. An edge $e = (A, B)$ is called

- **directed** if $(A, B) \in E \;\Rightarrow\; (B, A) \notin E$
  Notation: $A \rightarrow B$
- **undirected** if $(A, B) \in E \;\Rightarrow\; (B, A) \in E$
  Notation: $A - B$ or $B - A$

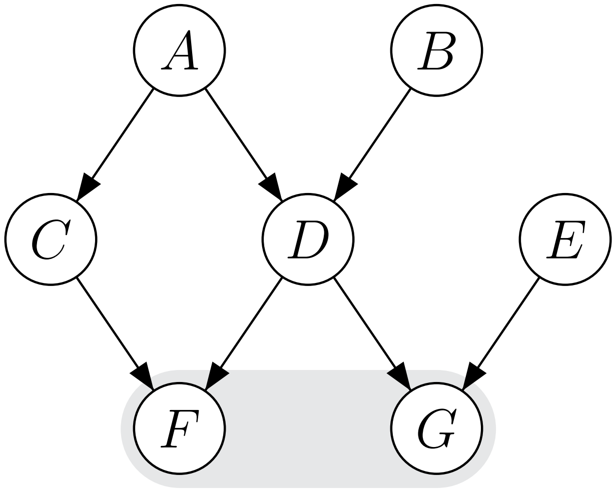
(Un)directed Graph

A graph with only (un)directed edges is called an (un)directed graph.

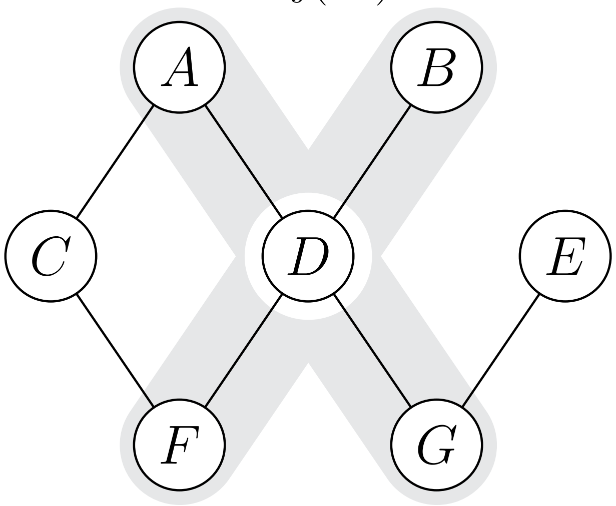
Adjacency Set

Let $\mathcal{G} = (V, E)$ be a graph. The set of nodes that is accessible via a given node $A \in V$ is called the **adjacency set** of A:

$$\mathrm{adj}(A) \;=\; \{B \in V \mid (A, B) \in E\}$$



$\mathrm{adj}(D)$



Paths

Let $\mathcal{G} = (V, E)$ be a graph. A series ρ of r pairwise different nodes

$$\rho = \left\langle A_{i_1}, \ldots, A_{i_r} \right\rangle$$

is called a **path** from A_i to A_j if

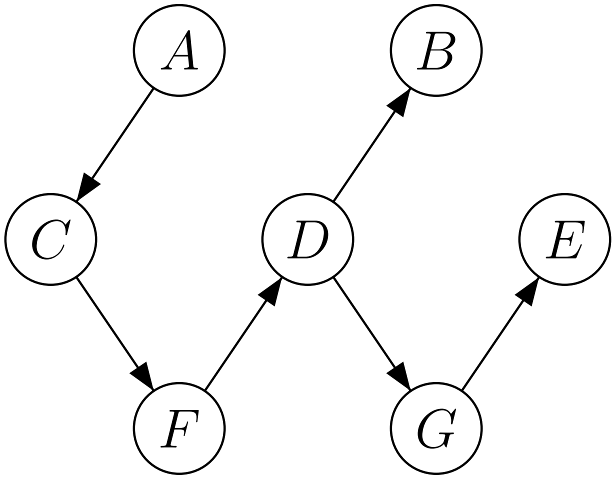
- $A_{i_1} = A_i, \quad A_{i_r} = A_j$
- $A_{i_{k+1}} \in \mathrm{adj}(A_{i_k}), \quad 1 \leq k < r$

A path with only undirected edges is called an **undirected path**

$$\rho = A_{i_1} - \cdots - A_{i_r}$$

whereas a path with only directed edges is referred to as a **directed path**

$$\rho = A_{i_1} \to \cdots \to A_{i_r}$$



If there is a directed path ρ from node A to node B in a directed graph $\mathcal{G}$ we write

$$A \overset{\rho}{\underset{\mathcal{G}}{\rightsquigarrow}} B.$$

If the path ρ is undirected we denote this with

$$A \overset{\rho}{\underset{\mathcal{G}}{\leftrightsquigarrow}} B.$$

Graph Types

Loop

Let $\mathcal{G} = (V, E)$ be an undirected graph. A path

$$\rho = X_1 - \cdots - X_k$$

with $X_k - X_1 \in E$ is called a loop.

Cycle

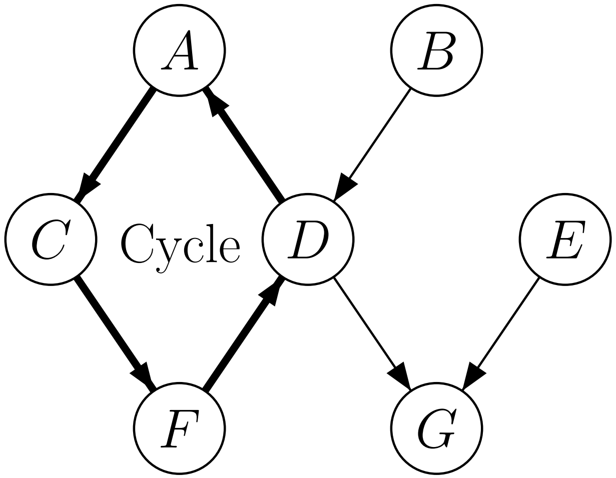
Let $\mathcal{G} = (V, E)$ be a directed graph. A path

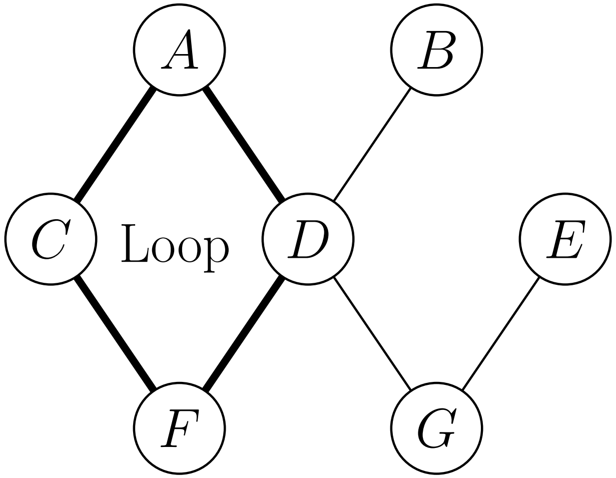
$$\rho = X_1 \to \cdots \to X_k$$

with $X_k \to X_1 \in E$ is called a cycle.

Directed Acyclic Graph (DAG)

A directed graph $\mathcal{G} = (V, E)$ is called **acyclic** if for every path $X_1 \to \cdots \to X_k$ in $\mathcal{G}$ the condition $X_k \to X_1 \notin E$ is satisfied, i. e. it contains no cycle.





Parents, Children and Families

Let $\mathcal{G} = (V, E)$ be a directed graph. For every node $A \in V$ we define the following sets:

- **Parents:**

  $$\text{parents}_{\mathcal{G}}(A) = \{B \in V \mid B \to A \in E\}$$

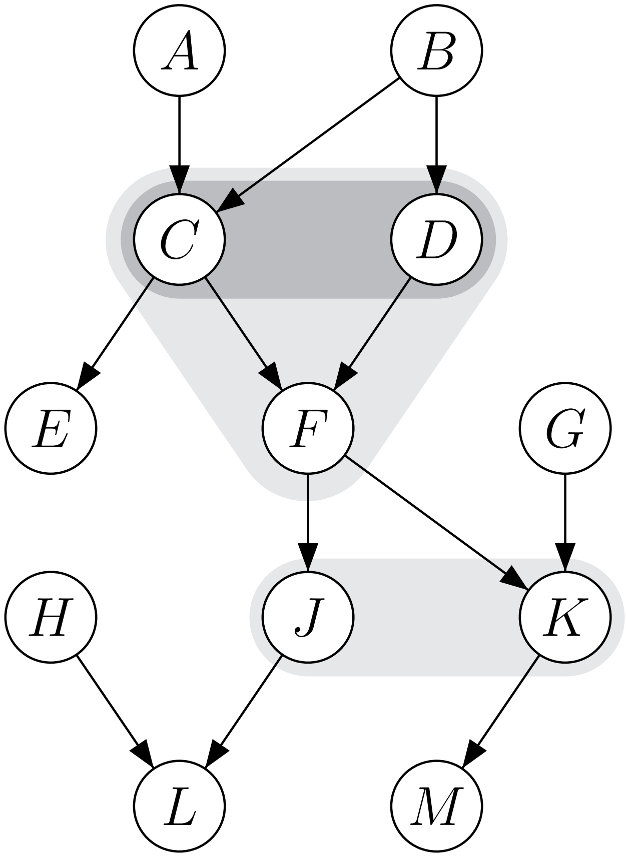
- **Children:**

  $$\text{children}_{\mathcal{G}}(A) = \{B \in V \mid A \to B \in E\}$$

- **Family:**

  $$\text{family}_{\mathcal{G}}(A) = \{A\} \cup \text{parents}_{\mathcal{G}}(A)$$

If the respective graph is clear from the context, the index $\mathcal{G}$ is omitted.



$$\begin{aligned} \text{parents}(F) &= \{C, D\} \\ \text{children}(F) &= \{J, K\} \\ \text{family}(F) &= \{C, D, F\} \end{aligned}$$

Ancestors, Descendants, Non-Descendants

Let $\mathcal{G} = (V, E)$ be a DAG. For every node $A \in V$ we define the following sets:

- **Ancestors:**

$$\mathrm{ancs}_{\mathcal{G}}(A) \;=\; \{B \in V \mid \exists \rho : B \overset{\rho}{\underset{\mathcal{G}}{\rightsquigarrow}} A\}$$

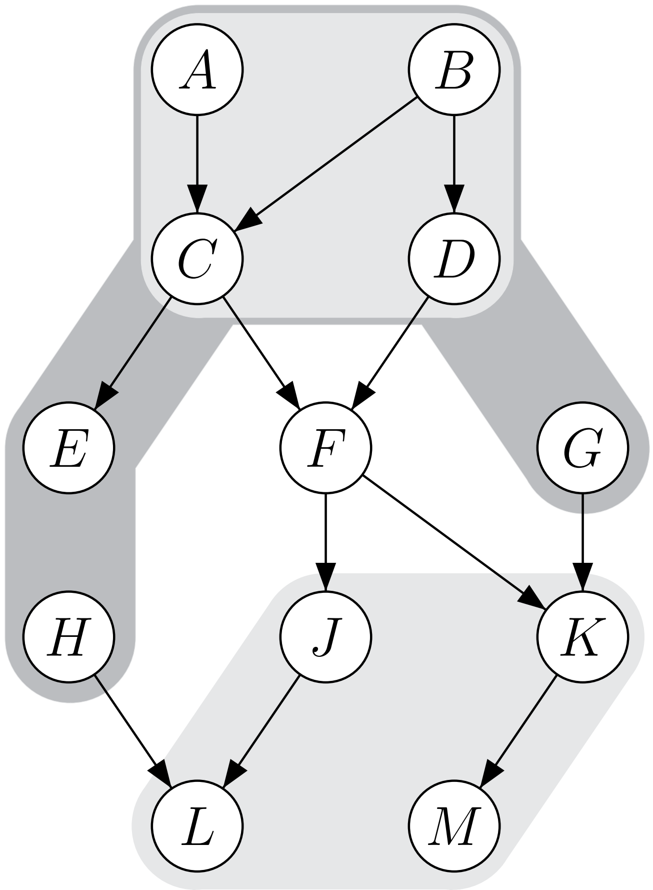
- **Descendants:**

$$\mathrm{descs}_{\mathcal{G}}(A) \;=\; \{B \in V \mid \exists \rho : A \overset{\rho}{\underset{\mathcal{G}}{\rightsquigarrow}} B\}$$

- **Non-Descendants:**

$$\text{non-descs}_{\mathcal{G}}(A) \;=\; V \setminus \{A\} \setminus \mathrm{descs}_{\mathcal{G}}(A)$$

If the respective graph is clear from the context, the index $\mathcal{G}$ is omitted.



$$\begin{aligned}\mathrm{ancs}(F) &= \{A, B, C, D\}\\ \mathrm{descs}(F) &= \{J, K, L, M\}\\ \text{non-descs}(F) &= \{A, B, C, D, E, G, H\}\end{aligned}$$

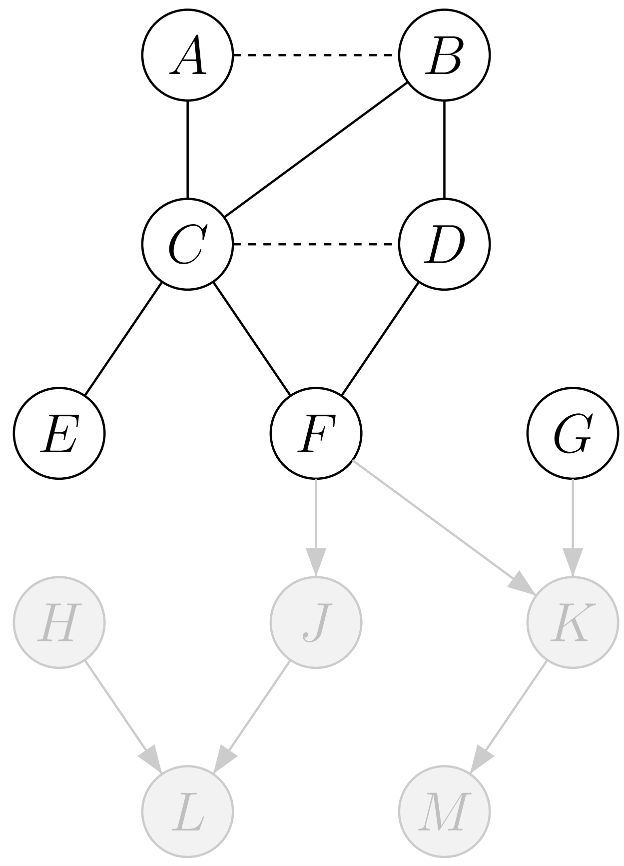
Operations on Graphs

Let $\mathcal{G} = (V, E)$ be a DAG.

The **Minimal Ancestral Subgraph** of $\mathcal{G}$ given a set $M \subseteq V$ of nodes is the smallest subgraph that contains all ancestors of all nodes in M.

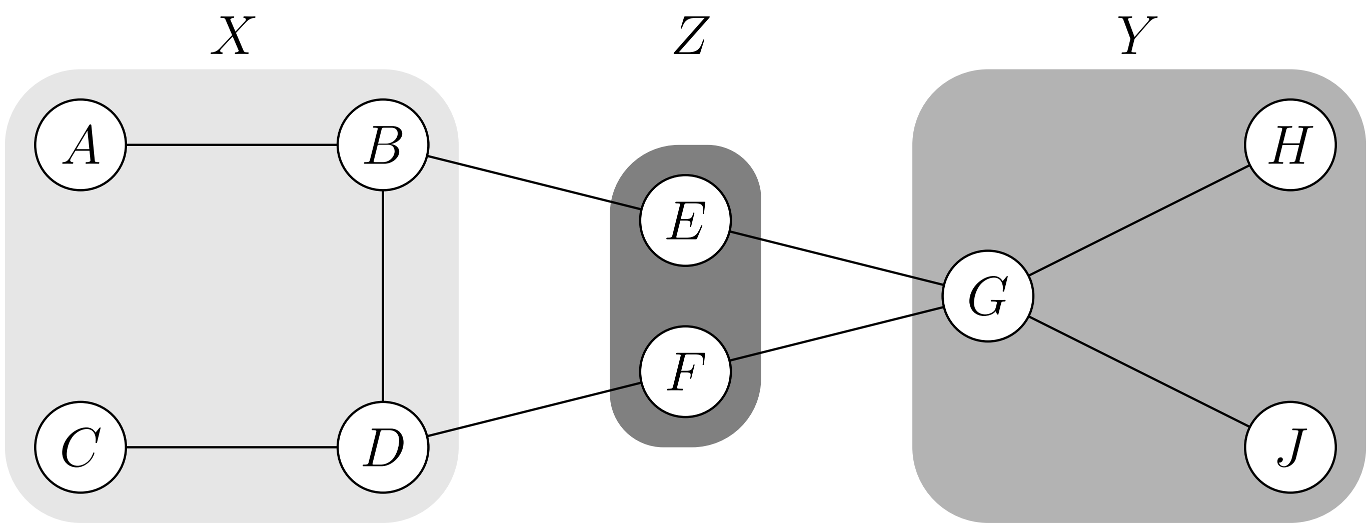
The **Moral Graph** of $\mathcal{G}$ is the undirected graph that is obtained by

1. connecting nodes that share a common child with an arbitrarily directed edge and,
2. converting all directed edges into undirected ones by dropping the arrow heads.



Moral graph of ancestral graph induced by the set $\{E, F, G\}$.

u-Separation



Let $\mathcal{G} = (V, E)$ be an undirected graph and $X, Y, Z \subseteq V$ three disjoint subsets of nodes. We agree on the following separation criteria:

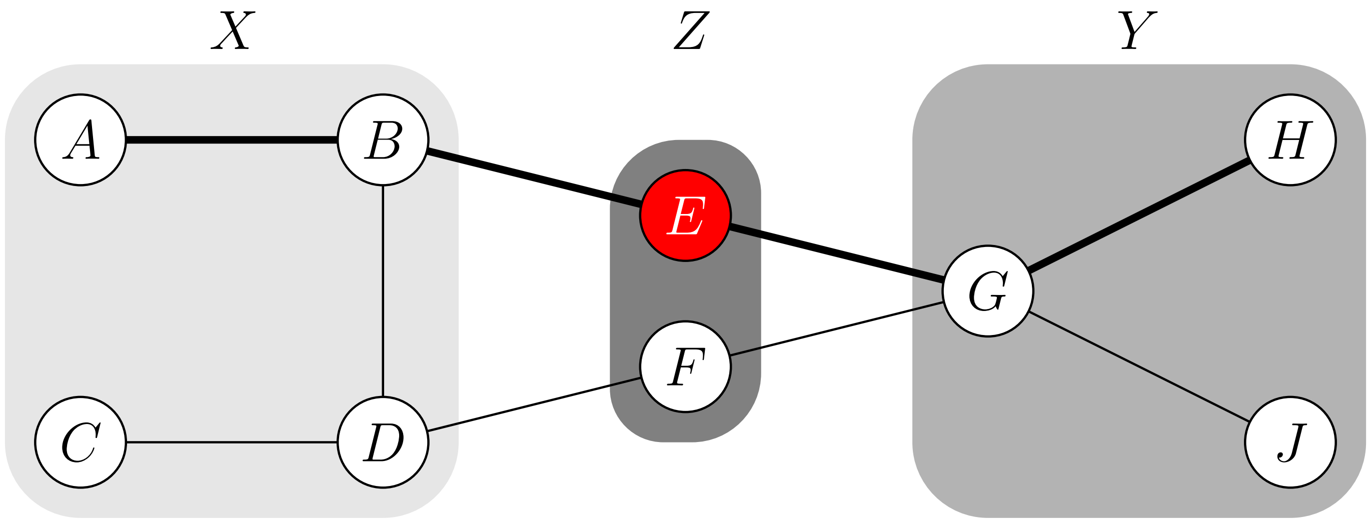
1. Z u-separates X from Y — written as

$$X \perp\!\!\!\perp_{\mathcal{G}} Y \mid Z,$$

   if every possible path from a node in X to a node in Y is blocked.

2. A path is blocked if it contains one (or more) **blocking nodes**.
3. A node is a blocking node if it lies in Z.

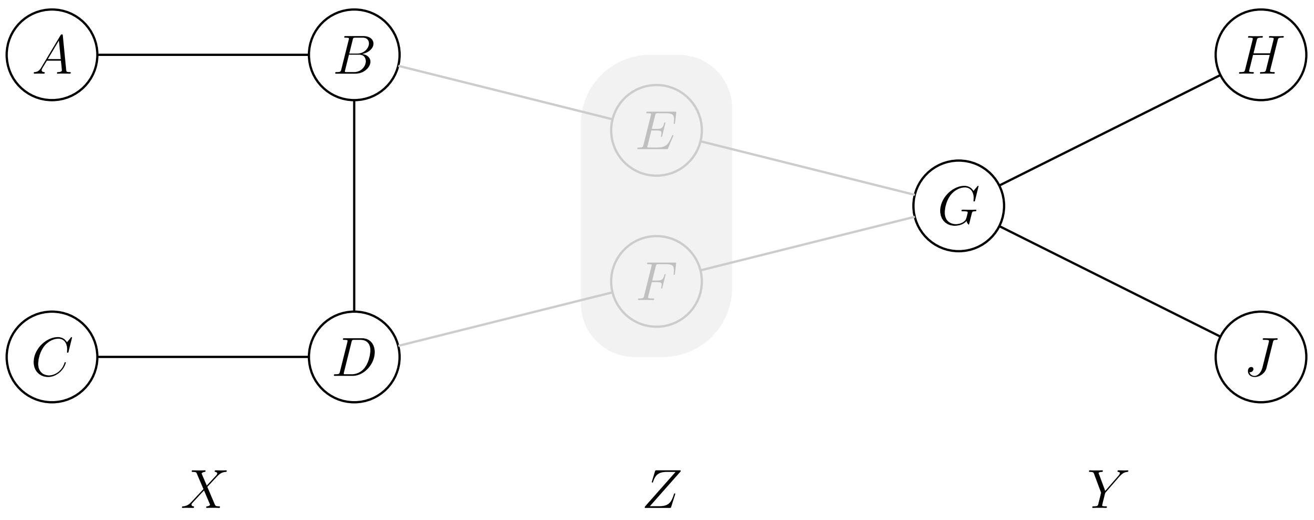
u-Separation



E. g. path $A - B - E - G - H$ is blocked by $E \in Z$. It can be easily verified, that every path from X to Y is blocked by Z. Hence we have:

$$\{A, B, C, D\} \perp\!\!\!\perp_{\mathcal{G}} \{G, H, J\} \mid \{E, F\}$$

u-Separation



Another way to check for u-separation: Remove the nodes in Z from the graph (and all the edges adjacent to these nodes). X and Y are u-separated by Z if the remaining graph is disconnected with X and Y in separate subgraphs.

d-Separation

Now: Separation criterion for directed graphs.

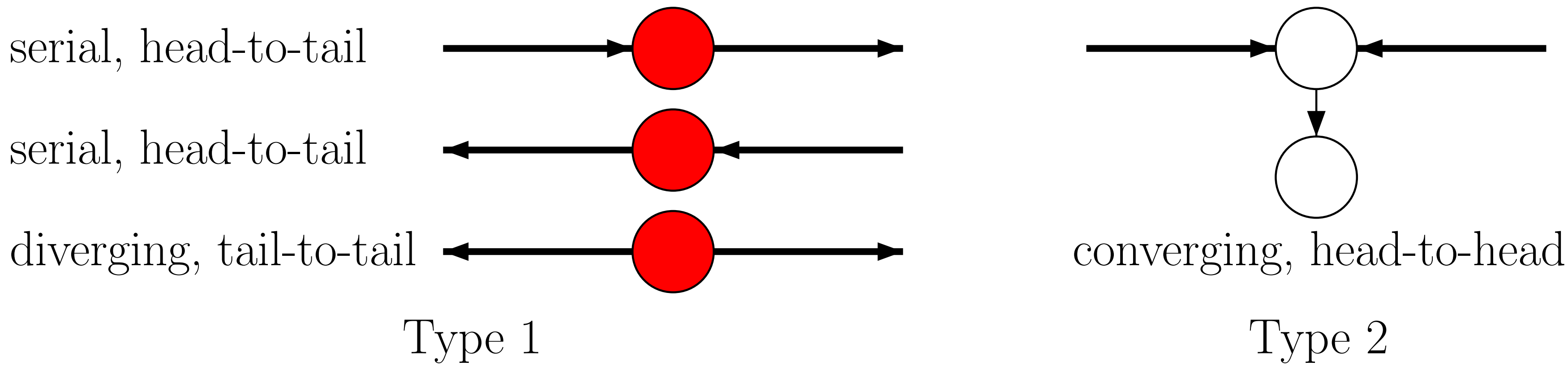
We use the same principles as for u-separation. Two modifications are necessary:

- Directed paths may lead also in reverse to the arrows.
- The blocking node condition is more sophisticated.

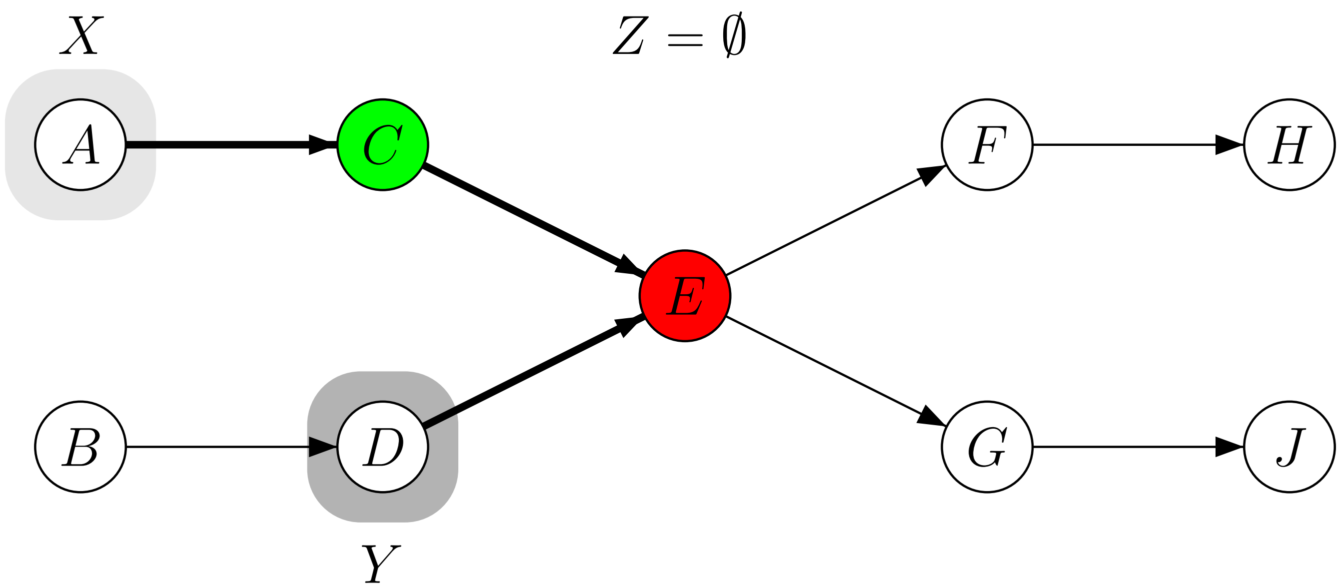
Blocking Node (in a directed path)

A node A is blocked if its edge directions **along the path**

- are of type 1 and $A \in Z$, or
- are of type 2 and neither A nor one of its descendants is in Z.



d-Separation



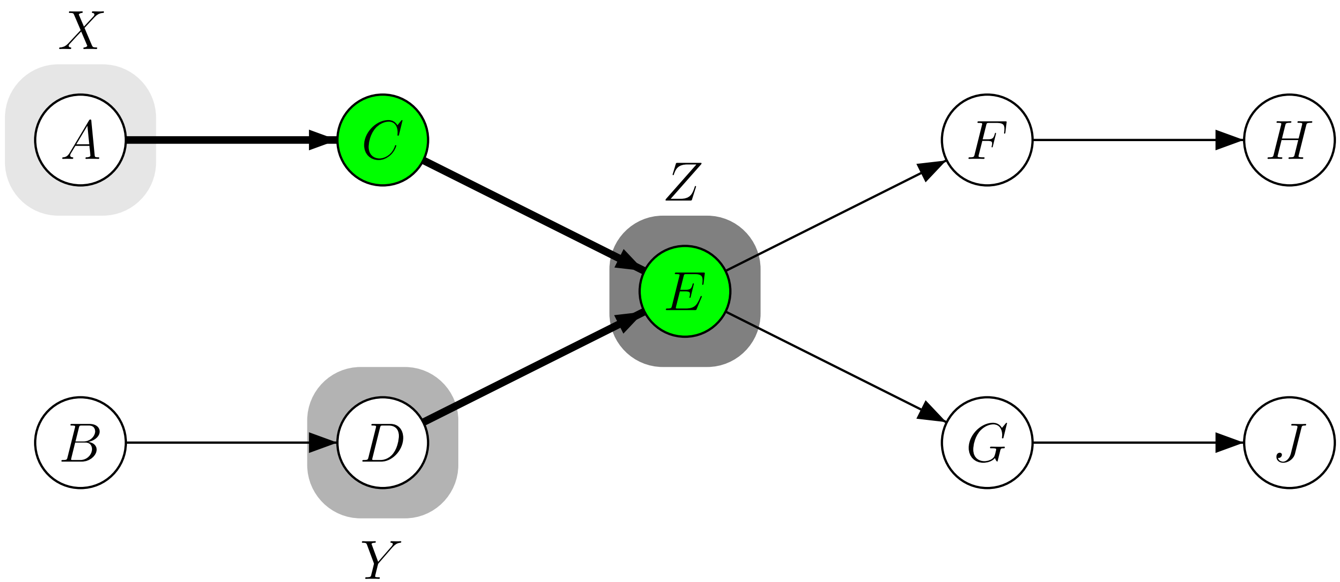
Checking path $A \to C \to E \leftarrow D$:

- C is **serial** and not in Z: non-blocking
- E is **converging** and not in Z, neither is F, G, H or J: **blocking**

$\Rightarrow$ Path is blocked

$$A \perp\!\!\!\perp D \mid \emptyset$$

d-Separation



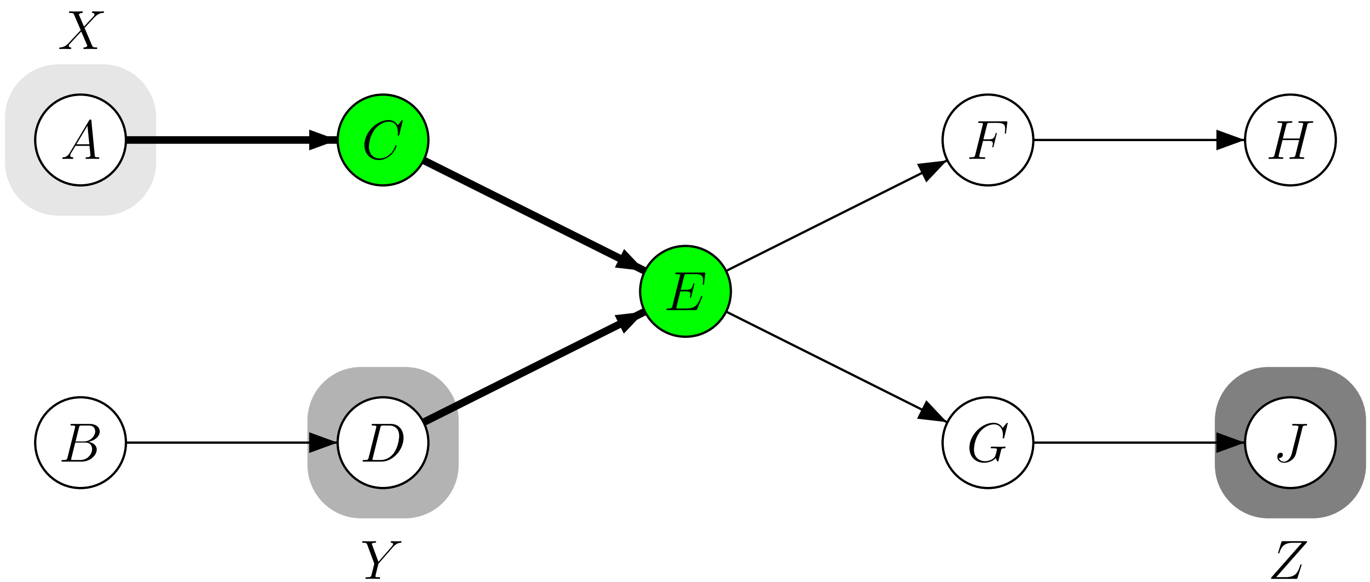
Checking path $A \rightarrow C \rightarrow E \leftarrow D$:

- C is **serial** and not in Z: non-blocking
- E is **converging** and in Z: non-blocking

$\Rightarrow$ Path is not blocked

$$A \not\perp\!\!\!\perp D \mid E$$

d-Separation



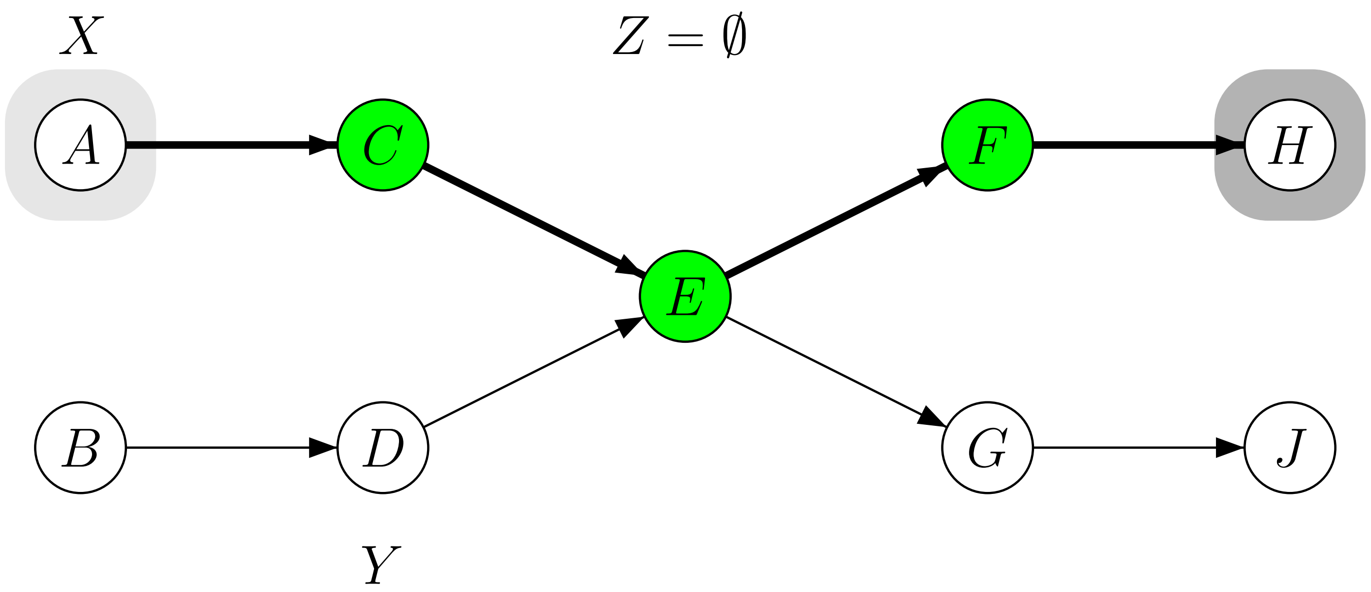
Checking path $A \to C \to E \leftarrow D$:

- C is **serial** and not in Z: non-blocking
- E is **converging** and not in Z but one of its descendants (J) is in Z: non-blocking

$\Rightarrow$ Path is not blocked

$$A \not\perp\!\!\!\perp D \mid J$$

d-Separation



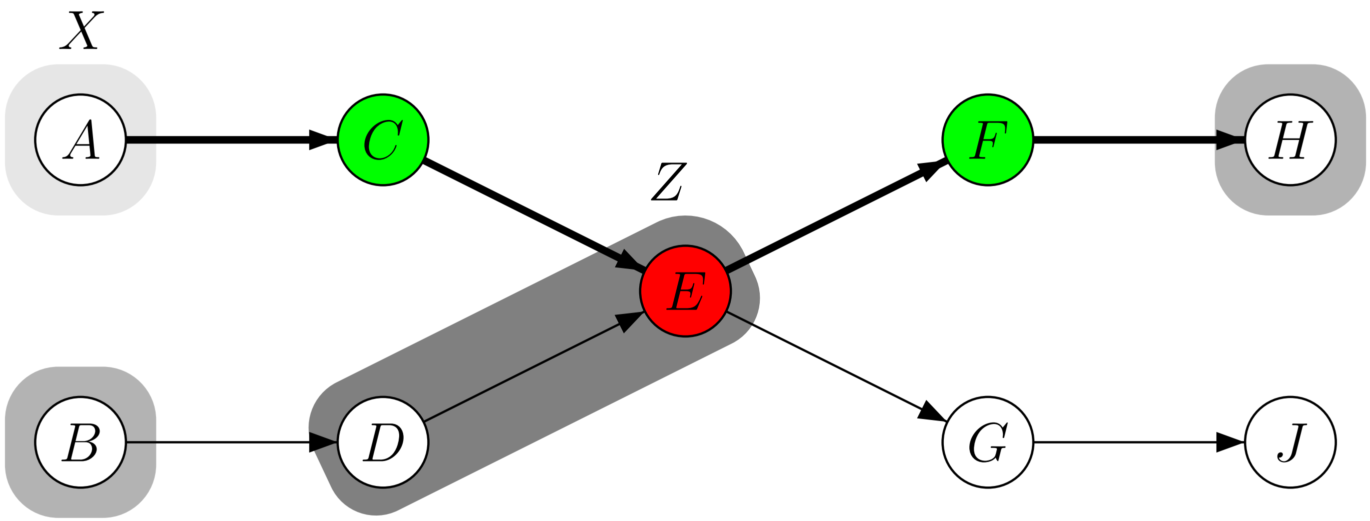
Checking path $A \to C \to E \to F \to H$:

- C is **serial** and not in Z: non-blocking
- E is **serial** and not in Z: non-blocking
- F is **serial** and not in Z: non-blocking

$\Rightarrow$ Path is not blocked

$$A \not\perp\!\!\!\perp H \mid \emptyset$$

d-Separation



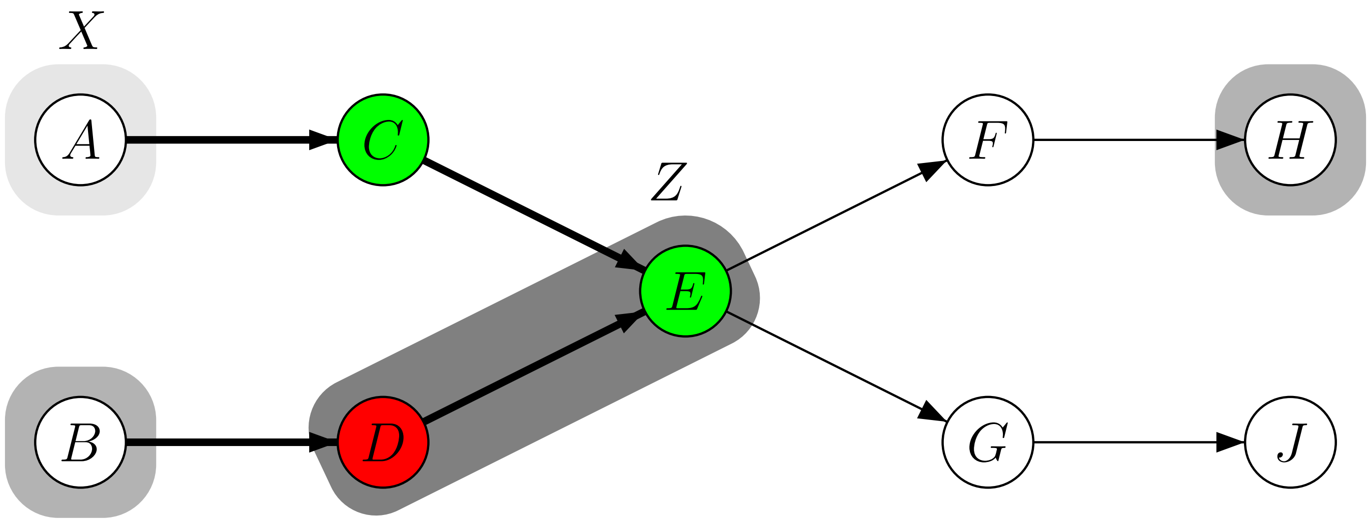
$Y = \{B, H\}$

Checking path $A \to C \to E \to F \to H$:

- C is **serial** and not in Z: non-blocking
- E is **serial** and in Z: **blocking**
- F is **serial** and not in Z: non-blocking

$\Rightarrow$ Path is blocked

d-Separation



$$Y = \{B, H\}$$

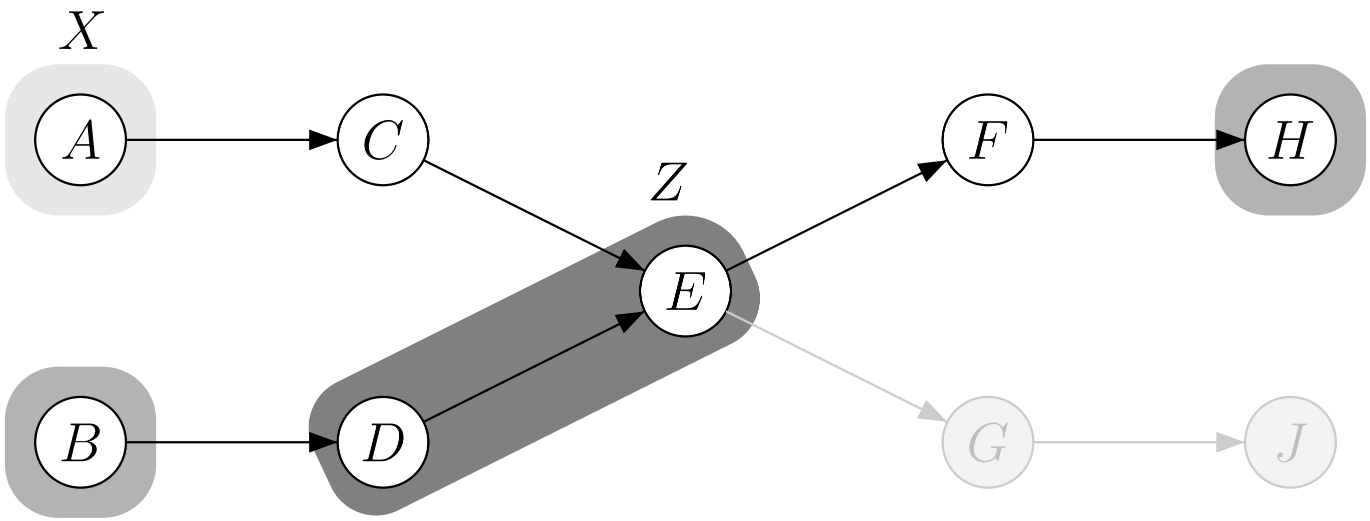
Checking path $A \to C \to E \leftarrow D \to B$:

- C is **serial** and not in Z: non-blocking
- E is **converging** and in Z: non-blocking
- D is **serial** and in Z: **blocking**

$\Rightarrow$ Path is blocked

$$A \perp\!\!\!\perp H, B \mid D, E$$

d-Separation: Alternative Way for Checking

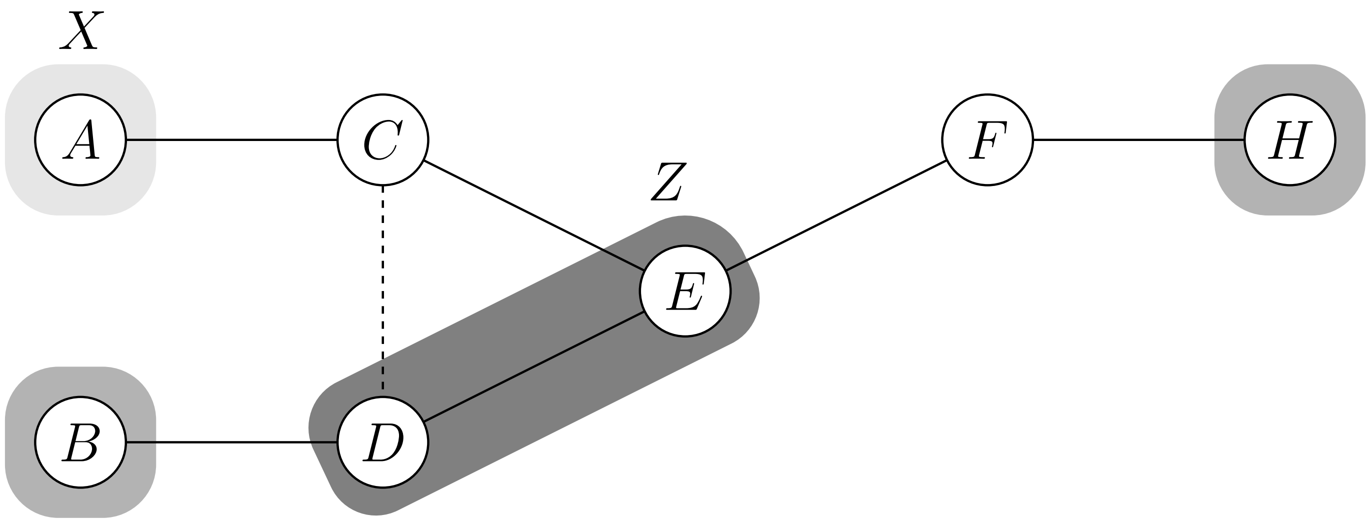


$Y = \{B, H\}$

Steps

- Create the minimal ancestral subgraph induced by $X \cup Y \cup Z$.

d-Separation: Alternative Way for Checking

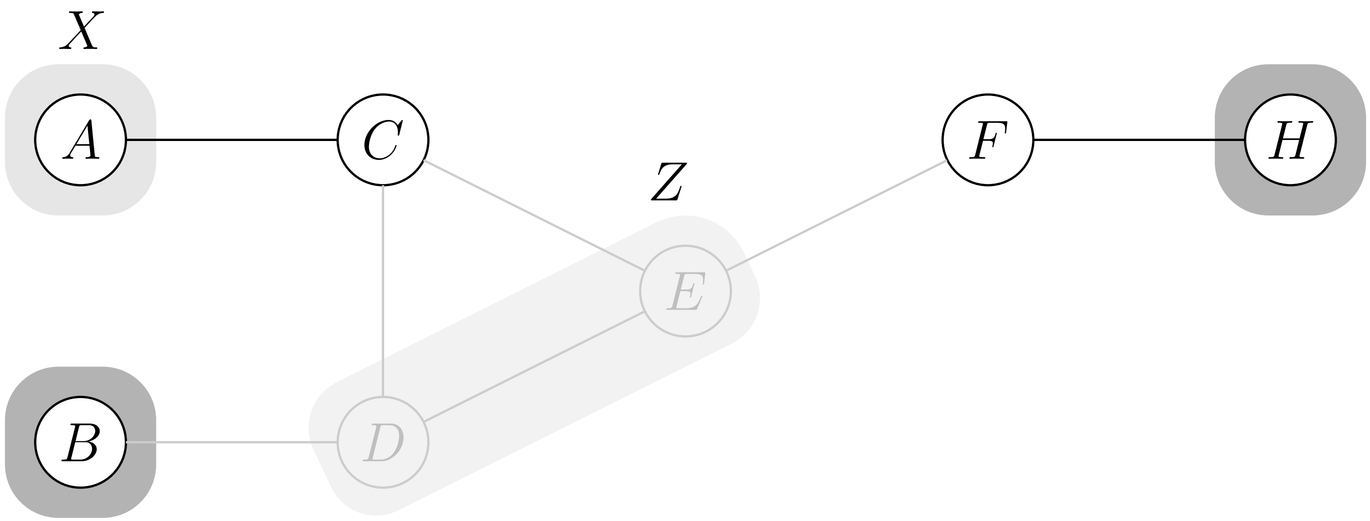


$$Y = \{B, H\}$$

Steps

- Create the minimal ancestral subgraph induced by $X \cup Y \cup Z$.
- Moralize that subgraph.

d-Separation: Alternative Way for Checking



$$Y = \{B, H\}$$

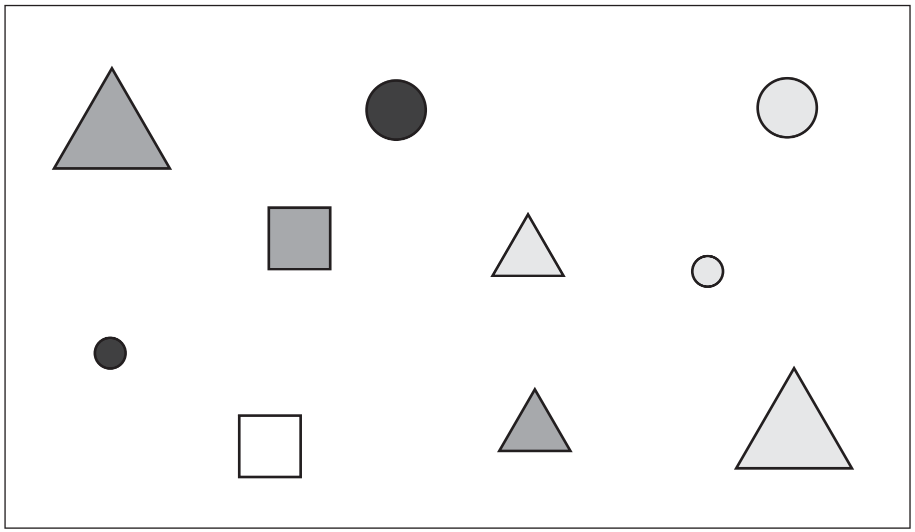
Steps:

- Create the minimal ancestral subgraph induced by $X \cup Y \cup Z$.
- Moralize that subgraph.
- Check for u-Separation in that undirected graph.

$$A \perp\!\!\!\perp H, B \mid D, E$$

Decomposition

Example World



- 10 simple geometric objects
- 3 attributes

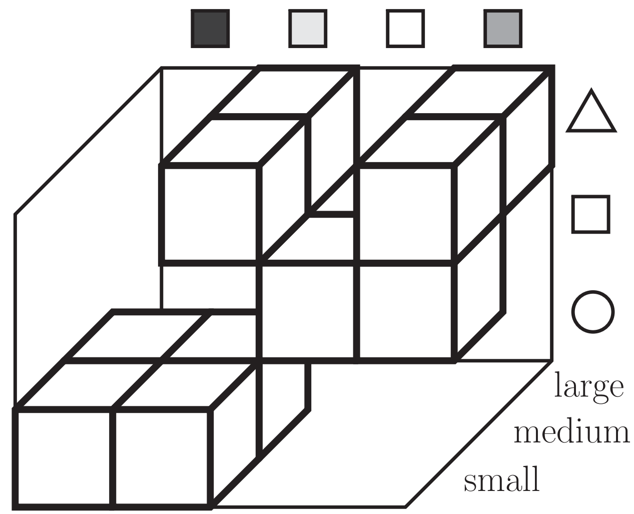
Relation

color	shape	size
■	○	small
■	○	medium
▢	○	small
▢	○	medium
▢	△	medium
▢	△	large
□	□	medium
▩	□	medium
▩	△	medium
▩	△	large

Relation

color	shape	size
■	○	small
■	○	medium
▫	○	small
▫	○	medium
▫	△	medium
▫	△	large
□	□	medium
▪	□	medium
▪	△	medium
▪	△	large

Geometric Representation



Object Representation

- **Universe of Discourse**: Ω
- $\omega \in \Omega$ represents a single abstract object.
- A subset $E \subseteq \Omega$ is called an **event**.
- For every event we use the function R to determine whether E is possible or not.

$$R: \ 2^{\Omega} \to \{0, 1\}$$

- We claim the following properties of R:
  1. $R(\emptyset) = 0$
  2. $\forall E_1, E_2 \subseteq \Omega: \ R(E_1 \cup E_2) \ = \ \max\{R(E_1), R(E_2)\}$
- For example:

$$R(E) \ = \ \begin{cases} 0 & \text{if} \ \ E = \emptyset \\ 1 & \text{otherwise} \end{cases}$$

Object Representation

- Attributes or Properties of these objects are introduced by functions: (later referred to as **random variables**)

$$A: \ \Omega \to \mathrm{dom}(A)$$

where $\mathrm{dom}(A)$ is the domain (i. e., set of all possible values) of A.

- A set of attibutes $U = \{A_1, \ldots, A_n\}$ is called an **attribute schema**.

- The **preimage** of an attribute defines an **event**:

$$\forall a \in \mathrm{dom}(A): \ A^{-1}(a) = \{\omega \in \Omega \mid A(\omega) = a\} \subseteq \Omega$$

- Abbreviation: $A^{-1}(a) \quad = \quad \{\omega \in \Omega \mid A(\omega) = a\} \quad = \quad \{A = a\}$

- We will index the function R to stress on which events it is defined. R_{AB} will be short for $R_{\{A,B\}}$.

$$R_{AB}: \bigcup_{a \in \mathrm{dom}(A)} \bigcup_{b \in \mathrm{dom}(B)} \Big\{\{A = a, B = b\}\Big\} \to \{0, 1\}$$

Formal Representation

A = color	B = shape	C = size
a_1 = ■	b_1 = ○	c_1 = small
a_1 = ■	b_1 = ○	c_2 = medium
a_2 = □	b_1 = ○	c_1 = small
a_2 = □	b_1 = ○	c_2 = medium
a_2 = □	b_3 = △	c_2 = medium
a_2 = □	b_3 = △	c_3 = large
a_3 = □	b_2 = □	c_2 = medium
a_4 = ■	b_2 = □	c_2 = medium
a_4 = ■	b_3 = △	c_2 = medium
a_4 = ■	b_3 = △	c_3 = large

$$
\begin{aligned}
R_{ABC}&(A = a, B = b, C = c)\\
&= R_{ABC}(\{A = a, B = b, C = c\})\\
&= R_{ABC}(\{\omega \in \Omega \mid A(\omega) = a \wedge\\
&\qquad\qquad B(\omega) = b \wedge\\
&\qquad\qquad C(\omega) = c\})\\
&= \begin{cases} 0 & \text{if there is no tuple } (a, b, c)\\ 1 & \text{else}\end{cases}
\end{aligned}
$$

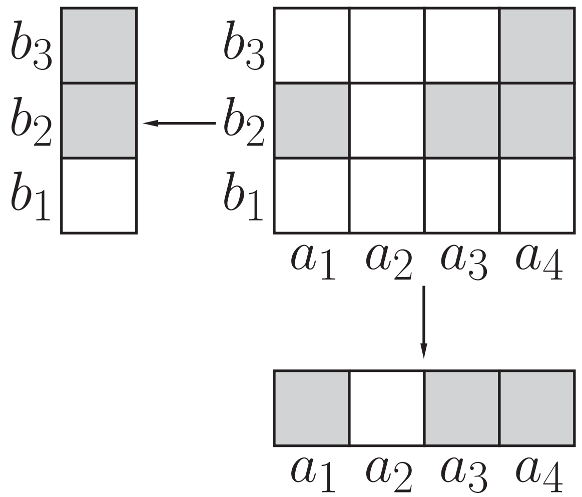
R serves as an indicator function.

Projection / Marginalization

Let R_{AB} be a relation over two attributes A and B. The projection (or marginalization) from schema $\{A, B\}$ to schema $\{A\}$ is defined as:

$$\forall a \in \mathrm{dom}(A): \; R_A(A = a) \; = \max_{\forall b \in \mathrm{dom}(B)} \{R_{AB}(A = a, B = b)\}$$

This principle is easily generalized to sets of attributes.

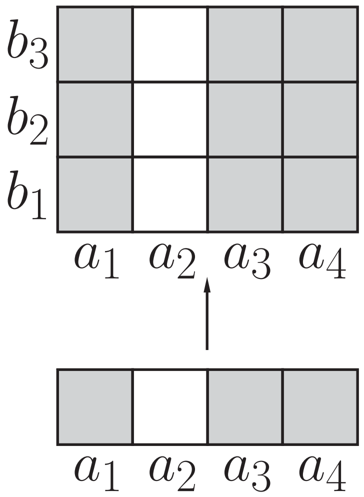


Cylindrical Extention

Let R_A be a relation over an attribute A. The cylindrical extention R_{AB} from $\{A\}$ to $\{A, B\}$ is defined as:

$$\forall a \in \mathrm{dom}(A) : \forall b \in \mathrm{dom}(B) : \; R_{AB}(A = a, B = b) \; = \; R_A(A = a)$$

This principle is easily generalized to sets of attributes.

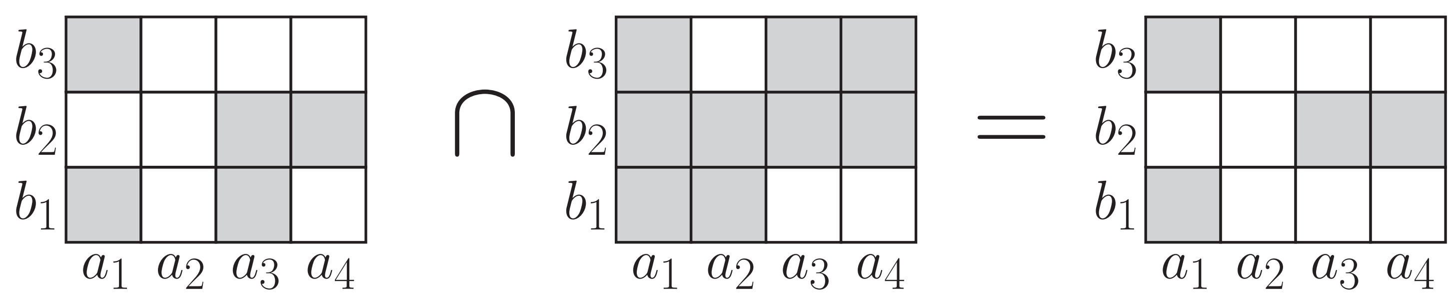


Intersection

Let $R_{AB}^{(1)}$ and $R_{AB}^{(2)}$ be two relations with attribute schema $\{A, B\}$. The intersection R_{AB} of both is defined in the natural way:

$$\forall a \in \mathrm{dom}(A) : \forall b \in \mathrm{dom}(B) :$$
$$R_{AB}(A = a, B = b) \;=\; \min\{R_{AB}^{(1)}(A = a, B = b), R_{AB}^{(2)}(A = a, B = b)\}$$

This principle is easily generalized to sets of attributes.

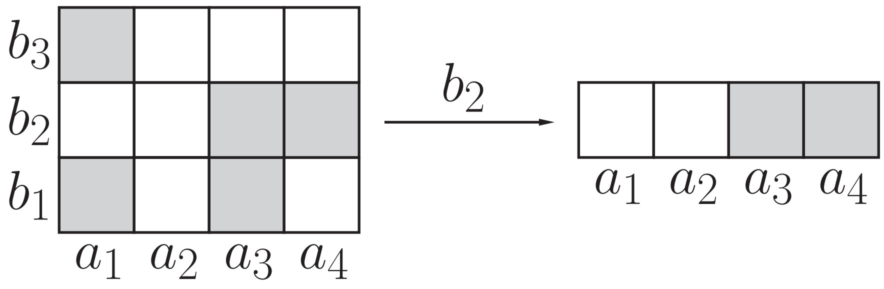


Conditional Relation

Let R_{AB} be a relation over the attribute schema $\{A, B\}$. The conditional relation of A given B is defined as follows:

$$\forall a \in \mathrm{dom}(A) : \forall b \in \mathrm{dom}(B) : R_A(A = a \mid B = b) \ = \ R_{AB}(A = a, B = b)$$

This principle is easily generalized to sets of attributes.

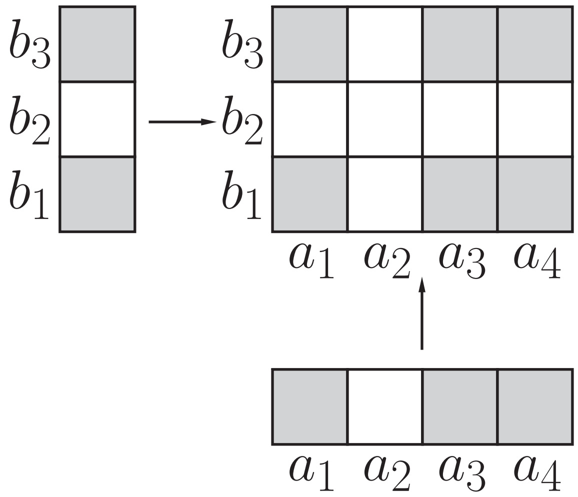


(Unconditional) Independence

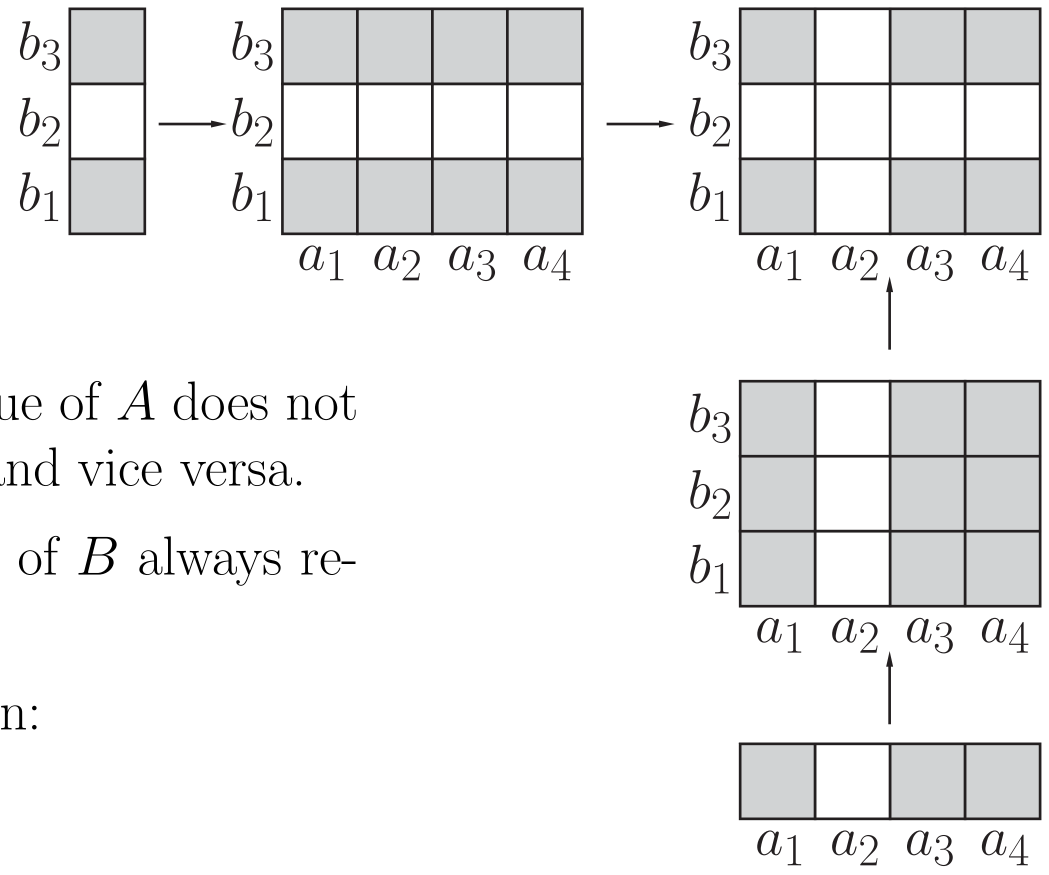
Let R_{AB} be a relation over the attribute schema $\{A, B\}$. We call A and B relationally independent (w. r. t. R_{AB}) if the following condition holds:

$$\forall a \in \mathrm{dom}(A) : \forall b \in \mathrm{dom}(B) : R_{AB}(A = a, B = b) \;=\; \min\{R_A(A = a), R_B(B = b)\}$$

This principle is easily generalized to sets of attributes.



(Unconditional) Independence



Intuition: Fixing one (possible) value of A does not restrict the (possible) values of B and vice versa.

Conditioning on any possible value of B always results in the same relation R_A.

Alternative independence expression:

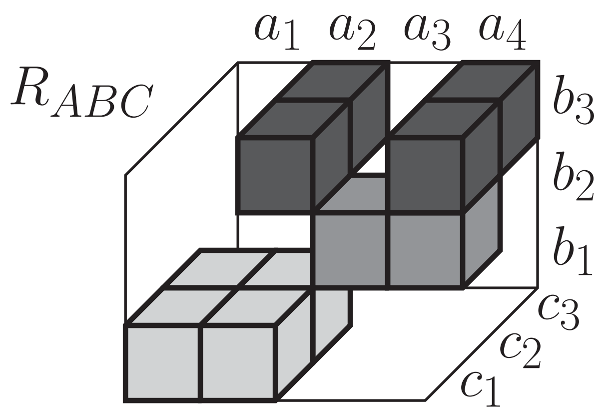
$$\forall b \in \mathrm{dom}(B): \quad R_B(B = b) = 1:$$
$$R_A(A = a \mid B = b) \;=\; R_A(A = a)$$

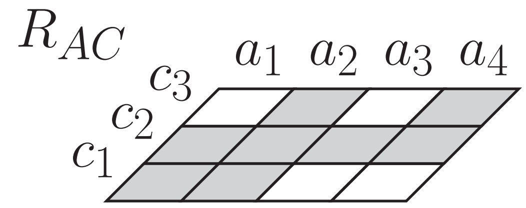
Decomposition

- Obviously, the original two-dimensional relation can be reconstructed from the two one-dimensional ones, if we have (unconditional) independence.
- The definition for (unconditional) independence already told us how to do so:

$$R_{AB}(A = a, B = b) \;=\; \min\{R_A(A = a), R_B(B = b)\}$$

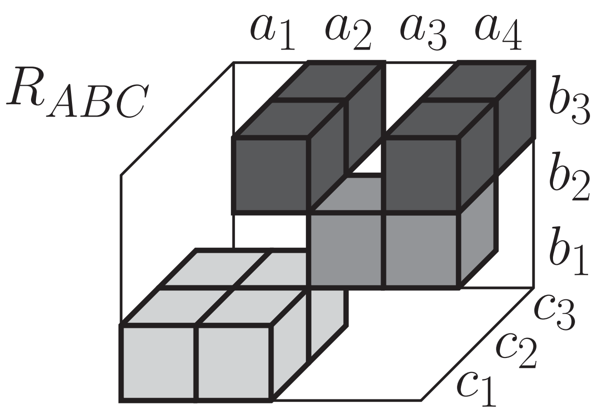
- Storing R_A and R_B is sufficient to represent the information of R_{AB}.
- **Question:** The (unconditional) independence is a rather strong restriction. Are there other types of independence that allow for a decomposition as well?

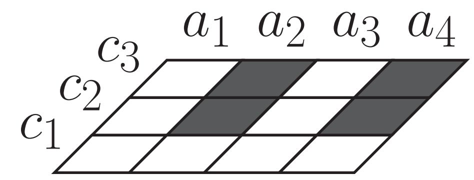


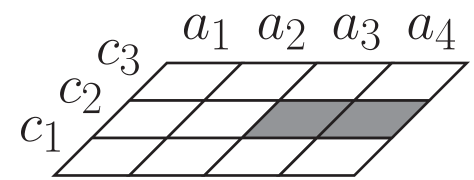


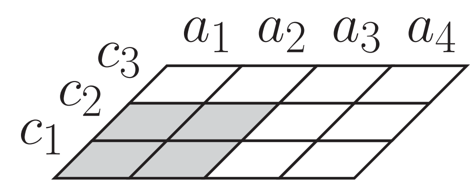
Clearly, A and C are unconditionally dependent, i. e. the relation R_{AC} cannot be reconstructed from R_A and R_C.

Conditional Relational Independence









However, given all possible values of B, all respective conditional relations R_{AC} show the independence of A and C.

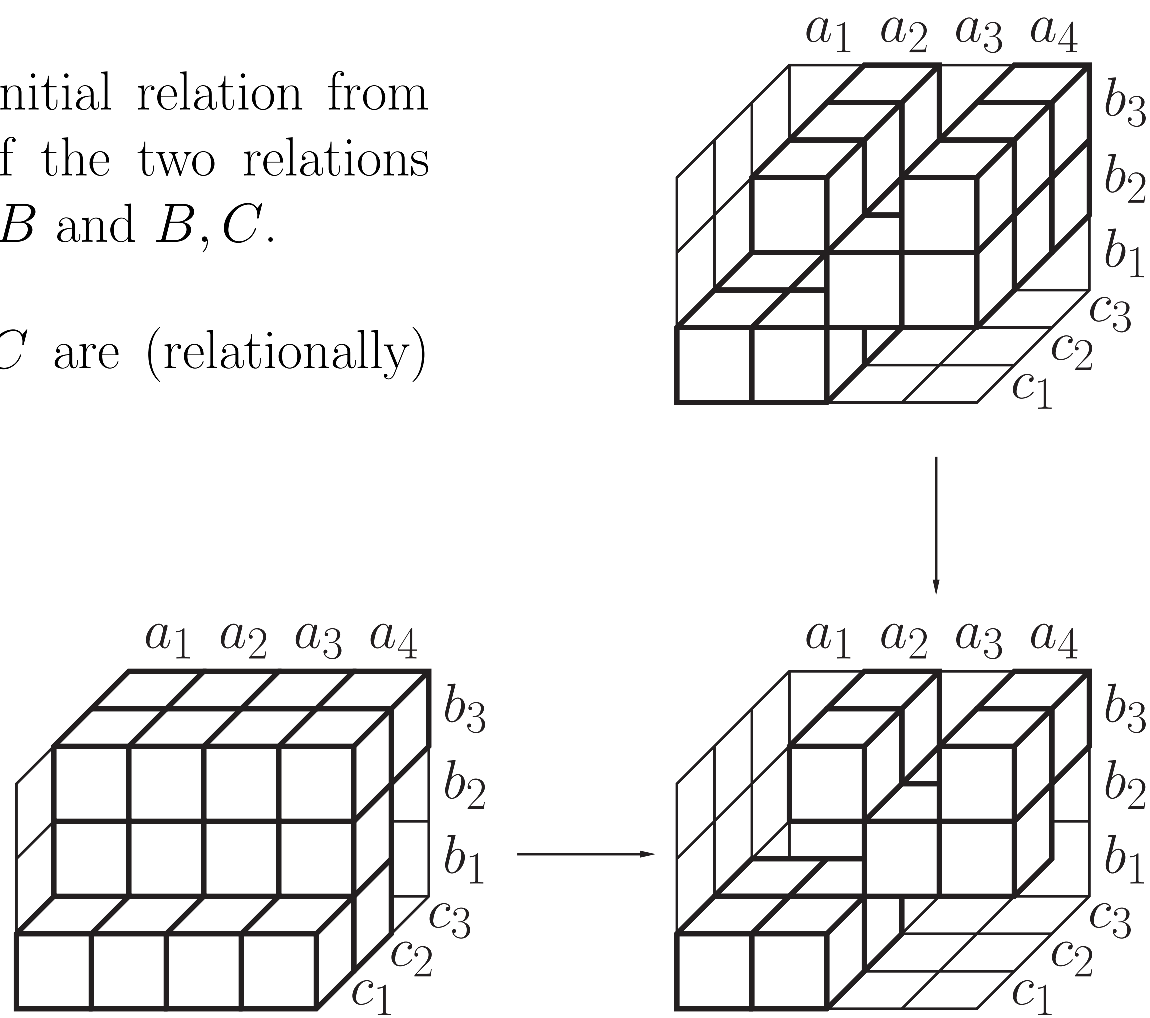
$$R_{AC}(a, c \mid b) \;=\; \min\{R_A(a \mid b), R_C(c \mid b)\}$$

With the definition of a conditional relation, the decomposition description for R_{ABC} reads:

$$R_{ABC}(a, b, c) \;=\; \min\{R_{AB}(a, b), R_{BC}(b, c)\}$$

Again, we reconstruct the initial relation from the cylindrical extentions of the two relations formed by the attributes A, B and B, C.

It is possible since A and C are (relationally) independent given B.



Probability Foundations

Reminder: Probability Theory

- **Goal**: Make statements and/or predictions about results of physical processes.
- Even processes that seem to be simple at first sight may reveal considerable difficulties when trying to predict.
- Describing real-world physical processes always calls for a simplifying mathematical model.
- Although everybody will have some intuitive notion about probability, we have to formally define the underlying mathematical structure.
- Randomness or chance enters as the incapability of precisely modelling a process or the inability of measuring the initial conditions.
    - *Example*: Predicting the trajectory of a billard ball over more than 9 banks requires more detailed measurement of the initial conditions (ball location, applied momentum etc.) than physically possible according to Heisenberg's uncertainty principle.

Formal Approach on the Model Side

- We conduct an experiment that has a set Ω of possible outcomes.
  E. g.:
    - Rolling a die ($\Omega = \{1, 2, 3, 4, 5, 6\}$)
    - Arrivals of phone calls ($\Omega = \mathbb{N}_0$)
    - Bread roll weights ($\Omega = \mathbb{R}_+$)
- Such an outcome is called an **elementary event**.
- All possible elementary events are called the **frame of discernment** Ω (or sometimes **universe of discourse**).
- The set representation stresses the following facts:
    - All possible outcomes are covered by the elements of Ω.
      (**collectively exhaustive**).
    - Every possible outcome is represented by exactly one element of Ω.
      (**mutual disjoint**).

Events

- Often, we are interested in *higher-level* events
(e. g. casting an odd number, arrival of at least 5 phone calls or
purchasing a bread roll heavier than 80 grams)

- Any subset $A \subseteq \Omega$ is called an **event** which **occurs**, if the outcome $\omega_0 \in \Omega$ of the random experiment lies in A:

$$\text{Event } A \subseteq \Omega \;\text{ occurs} \quad \Leftrightarrow \quad \bigvee_{\omega \in A} (\omega = \omega_0) \;=\; \mathsf{true} \quad \Leftrightarrow \quad \omega_0 \in A$$

- Since events are sets, we can define for two events A and B:
  - $A \cup B$ occurs if A or B occurs; $A \cap B$ occurs if A and B occurs.
  - $\overline{A}$ occurs if A does not occur (i. e., if $\Omega \backslash A$ occurs).
  - A and B are *mutually exclusive*, iff $A \cap B = \emptyset$.

Event Algebra

- A family of sets $\mathcal{E} = \{E_1, \dots, E_n\}$ is called an **event algebra**, if the following conditions hold:
  - The **certain event** Ω lies in $\mathcal{E}$.
  - If $E \in \mathcal{E}$, then $\overline{E} = \Omega \backslash E \in \mathcal{E}$.
  - If E_1 and E_2 lie in $\mathcal{E}$, then $E_1 \cup E_2 \in \mathcal{E}$ and $E_1 \cap E_2 \in \mathcal{E}$.
- If Ω is uncountable, we require the additional property:

  For a series of events $E_i \in \mathcal{E}, i \in \mathbb{N},$ the events $\bigcup_{i=1}^{\infty} E_i$ and $\bigcap_{i=1}^{\infty} E_i$ are also in $\mathcal{E}$.

  $\mathcal{E}$ is then called a **σ-algebra**.

Side remarks:

- Smallest event algebra: $\mathcal{E} = \{\emptyset, \Omega\}$
- Largest event algebra (for finite or countable Ω): $\mathcal{E} = 2^\Omega = \{A \subseteq \Omega \mid \mathsf{true}\}$

Probability Function

- Given an event algebra $\mathcal{E}$, we would like to assign every event $E \in \mathcal{E}$ its probability with a **probability function** $P: \mathcal{E} \to [0,1]$.

- We require P to satisfy the so-called **Kolmogorov Axioms**:

  - $\forall E \in \mathcal{E}: \ 0 \leq P(E) \leq 1$
  - $P(\Omega) = 1$
  - If $E_1, E_2 \in \mathcal{E}$ are mutually exclusive, then $P(E_1 \cup E_2) = P(E_1) + P(E_2)$.

- From these axioms one can conclude the following (incomplete) list of properties:

  - $\forall E \in \mathcal{E}: \ P(\overline{E}) = 1 - P(E)$
  - $P(\emptyset) = 0$
  - For pairwise disjoint events $E_1, E_2, \ldots \in \mathcal{E}$ holds:

$$P\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} P(E_i)$$

    Note that for $|\Omega| < \infty$ the union and sum are finite also.

Elementary Probabilities and Densities

Question 1: How to calculate P?

Question 2: Are there "default" event algebras?

- Idea for question 1: We have to find a way of distributing (thus the notion *distribution*) the unit mass of probability over all elements $\omega \in \Omega$.
  - If Ω is finite or countable a **probability mass function** p is used:

$$p: \ \Omega \to [0,1] \quad \text{and} \quad \sum_{\omega \in \Omega} p(\omega) = 1$$

  - If Ω is uncountable (i. e., continuous) a **probability density function** f is used:

$$f: \ \Omega \to \mathbb{R} \quad \text{and} \quad \int_{\Omega} f(\omega)\, \mathrm{d}\omega = 1$$

“Default” Event Algebras

- Idea for question 2 (“default” event algebras) we have to distinguish again between the cardinalities of Ω:
  - Ω finite or countable: $\mathcal{E} = 2^\Omega$
  - Ω uncountable, e. g. $\Omega = \mathbb{R}$: $\mathcal{E} = \mathcal{B}(\mathbb{R})$
- $\mathcal{B}(\mathbb{R})$ is the **Borel Algebra**, i. e., the smallest σ-algebra that contains all closed intervals $[a, b] \subset \mathbb{R}$ with $a < b$.
- $\mathcal{B}(\mathbb{R})$ also contains all open intervals and single-item sets.
- It is sufficient to note here, that all intervals are contained

$$\{[a,b]\,,\,]a,b]\,,\,]a,b[\,,\,[a,b[\,\subset \mathbb{R} \mid a < b\} \subset \mathcal{B}(\mathbb{R})$$

  because the event of a bread roll having a weight between 80 g and 90 g is represented by the interval $[80, 90]$.

Probability Spaces

- For a sample space A, an event algebra B (over A) and a probability function C, we call the triple

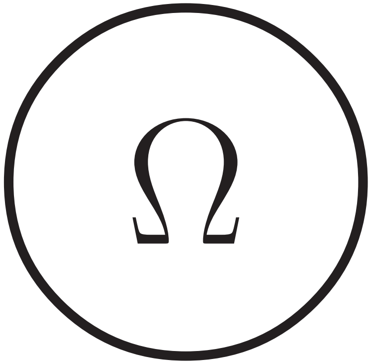
$$(A, B, C)$$

a **probability space**.

Real World

$(\Xi, \mathcal{X}, Q)$

Model



$(\Omega, \mathcal{E}, P)$

Reminder: Preimage of a Function

- Let $f: \; D \to M$ be a function that assigns to every value of D a value in M.
- For every value of $y \in M$ we can ask which values of $x \in D$ are mapped to y:

$$D \supseteq \{x \in D \mid f(x) = y\} \stackrel{\text{Def}}{=} f^{-1}(y)$$

- $f^{-1}(y)$ is called the **preimage** of y under f, denoted also as $\{f = y\}$.
- The notion can be generalized from $y \in M$ to sets $B \subseteq M$:

$$D \supseteq \{x \in D \mid f(x) \in B\} \stackrel{\text{Def}}{=} f^{-1}(B)$$

- If f is bijective then $\forall y \in M : \left|f^{-1}(y)\right| = 1$.
- Examples:
  - $\sin^{-1}(0) = \{k \cdot \pi \mid k \in \mathbb{Z}\}$
  - $\exp^{-1}(1) = \{0\}$
  - $\operatorname{sgn}^{-1}(1) = (0, +\infty) \subset \mathbb{R}$

Random Variable

We still need a means of mapping real-world outcomes in Ξ to our space Ω.

- A function $X : D \to M$ is called a **random variable** iff the preimage of any value of M is an event (in some probability space).

- If X maps Ξ onto Ω, we define

$$P_X(X \in A) \;=\; Q(\{\xi \in \Xi \mid X(\xi) \in A\}).$$

- X may also map from Ω to another domain: $X : \Omega \to \mathrm{dom}(X)$. We then define:

$$P_X(X \in A) \;=\; P(\{\omega \in \Omega \mid X(\omega) \in A\}).$$

- If X is numeric, we call $F(x)$ with

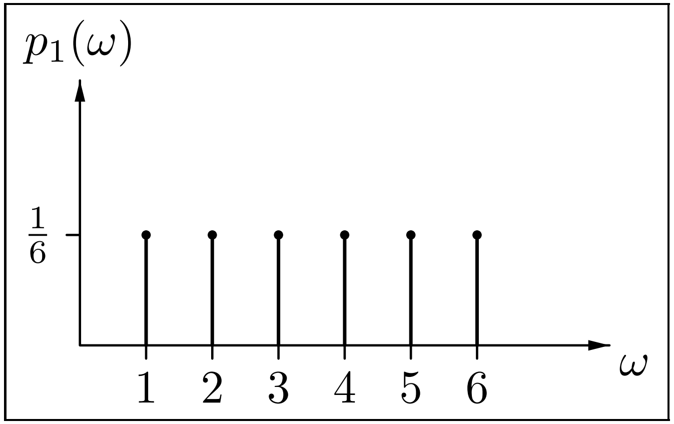
$$F(x) \;=\; P(X \leq x)$$

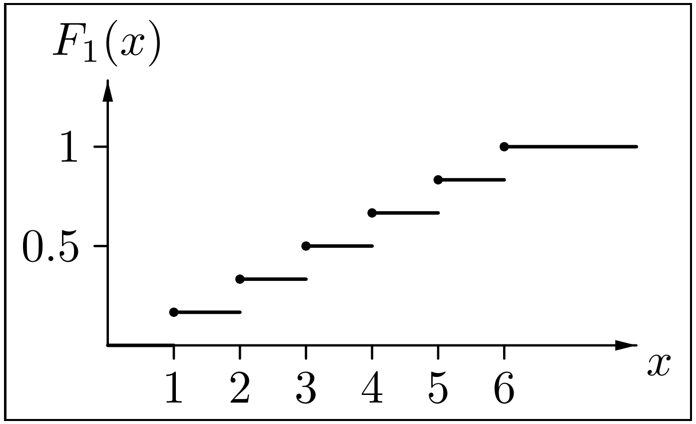
the **distribution function** of X.

Example: Rolling a Die

$\Omega = \{1,2,3,4,5,6\} \quad X = \mathrm{id}$

$$p_1(\omega) = \tfrac{1}{6} \qquad\qquad F_1(x) = P(X \le x)$$



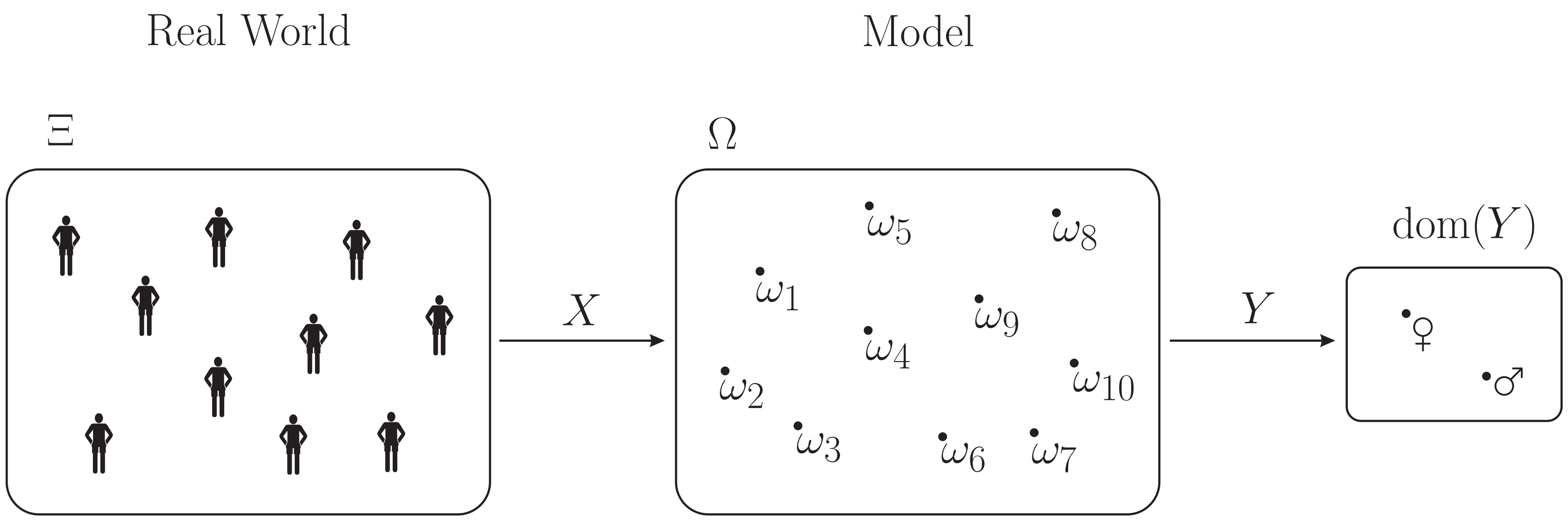


$$\sum_{\omega\in\Omega} p_1(\omega) = \sum_{i=1}^{6} p_1(\omega_i) = \sum_{i=1}^{6} \frac{1}{6} \; = \; 1$$

$$P(X \le x) = \sum_{x' \le x} P(X = x')$$

$$P(a < X \le b) = F_1(b) - F_1(a)$$

$$P(X = x) \; = \; P(\{X = x\}) \; = \; P(X^{-1}(x)) \; = \; P(\{\omega \in \Omega \mid X(\omega) = x\})$$



$X : \Xi \to \Omega$ $\qquad$ $Y : \Omega \to \{♀, ♂\}$

$$Q\Big(\{\xi \in \Xi \mid X(\xi) \in Y^{-1}(♀)\}\Big) \quad = \quad P\Big(\{\omega \in \Omega \mid Y(\omega) = ♀\}\Big) \quad = \quad P\Big(Y = ♀\Big) \quad = \quad P\Big(♀\Big)$$

Applied Probability Theory

Why (Kolmogorov) Axioms?

- If P models an *objectively* observable probability, these axioms are obviously reasonable.
- However, why should an agent obey formal axioms when modeling degrees of (subjective) belief?
- Objective vs. subjective probabilities
- Axioms constrain the set of beliefs an agent can abide.
- Finetti (1931) gave one of the most plausible arguments why subjective beliefs should respect axioms:

  "When using contradictory beliefs, the agent will eventually fail."

Unconditional Probabilities

- $P(A)$ designates the *unconditioned* or *a priori* probability that $A \subseteq \Omega$ occurs if *no* other additional information is present.
  For example:

  $$P(\mathsf{cavity}) \;=\; 0.1$$

  Note: Here, cavity is a proposition.

- A formally different way to state the same would be via a binary random variable Cavity:

  $$P(\mathsf{Cavity} = \mathsf{true}) \;=\; 0.1$$

- A priori probabilities are derived from statistical surveys or general rules.

Unconditional Probabilities

- In general a random variable can assume more than two values:

$$\begin{aligned}
P(\ \mathsf{Weather} = \mathsf{sunny}\ ) &= 0.7 \\
P(\ \mathsf{Weather} = \mathsf{rainy}\ ) &= 0.2 \\
P(\ \mathsf{Weather} = \mathsf{cloudy}) &= 0.02 \\
P(\ \mathsf{Weather} = \mathsf{snowy}) &= 0.08 \\
P(\mathsf{Headache} = \mathsf{true}\ ) &= 0.1
\end{aligned}$$

- $P(X)$ designates the vector of probabilities for the (ordered) domain of the random variable X:

$$\begin{aligned}
P(\mathsf{Weather}) &= \langle 0.7, 0.2, 0.02, 0.08 \rangle \\
P(\mathsf{Headache}) &= \langle 0.1, 0.9 \rangle
\end{aligned}$$

- Both vectors define the respective probability distributions of the two random variables.

Conditional Probabilities

- New evidence can alter the probability of an event.
- Example: The probability for cavity increases if information about a toothache arises.
- With additional information present, the a priori knowledge must not be used!
- $P(A \mid B)$ designates the *conditional* or *a posteriori* probability of A *given* the sole observation (*evidence*) B.

$$P(\mathsf{cavity} \mid \mathsf{toothache}) \;=\; 0.8$$

- For random variables X and Y $P(X \mid Y)$ represents the set of conditional distributions for each possible value of Y.

Conditional Probabilities

- $P(\mathsf{Weather} \mid \mathsf{Headache})$ consists of the following table:

	$\mathsf{h} \mathrel{\hat{=}} \mathsf{Headache} = \mathsf{true}$	$\neg\mathsf{h} \mathrel{\hat{=}} \mathsf{Headache} = \mathsf{false}$
$\mathsf{Weather} = \mathsf{sunny}$	$P(\mathsf{W} = \mathsf{sunny} \mid \mathsf{h})$	$P(\mathsf{W} = \mathsf{sunny} \mid \neg\mathsf{h})$
$\mathsf{Weather} = \mathsf{rainy}$	$P(\mathsf{W} = \mathsf{rainy} \mid \mathsf{h})$	$P(\mathsf{W} = \mathsf{rainy} \mid \neg\mathsf{h})$
$\mathsf{Weather} = \mathsf{cloudy}$	$P(\mathsf{W} = \mathsf{cloudy} \mid \mathsf{h})$	$P(\mathsf{W} = \mathsf{cloudy} \mid \neg\mathsf{h})$
$\mathsf{Weather} = \mathsf{snowy}$	$P(\mathsf{W} = \mathsf{snowy} \mid \mathsf{h})$	$P(\mathsf{W} = \mathsf{snowy} \mid \neg\mathsf{h})$

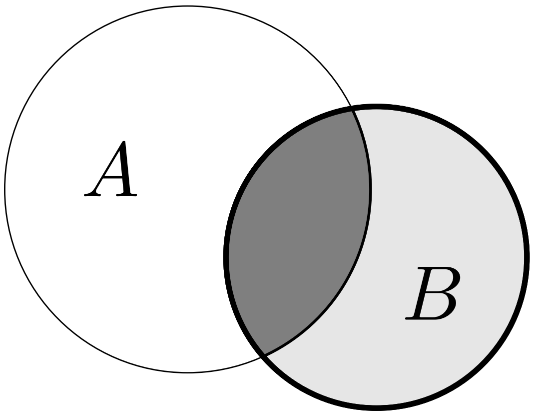
- Note that we are dealing with *two* distributions now!
  Therefore each column sums up to unity!

- Formal definition:

$$P(A \mid B) \;=\; \frac{P(A \wedge B)}{P(B)} \qquad \text{if} \quad P(B) > 0$$

Conditional Probabilities

$$\boxed{P(A \mid B) \;=\; \frac{P(A \wedge B)}{P(B)}}$$



- Product Rule: $P(A \wedge B) \;=\; P(A \mid B) \cdot P(B)$
- Also: $P(A \wedge B) \;=\; P(B \mid A) \cdot P(A)$
- A and B are *independent* iff

$$P(A \mid B) \;=\; P(A) \qquad \text{and} \qquad P(B \mid A) \;=\; P(B)$$

- Equivalently, iff the following equation holds true:

$$P(A \wedge B) \;=\; P(A) \cdot P(B)$$

Interpretation of Conditional Probabilities

Caution! Common misinterpretation:

"$P(A \mid B) = 0.8$ means, that $P(A) = 0.8$, given B holds."

This statement is wrong due to (at least) two facts:

- $P(A)$ is *always* the a-priori probability,
  never the probability of A given that B holds!

- $P(A \mid B) = 0.8$ is only applicable as long as no other evidence except B is present.
  If C becomes known, $P(A \mid B \wedge C)$ has to be determined.

  In general we have:

  $$P(A \mid B \wedge C) \neq P(A \mid B)$$

  E. g. $C \rightarrow A$ might apply.

Joint Probabilities

- Let $X_1, \dots, X_n$ be random variables over the same framce of descernment Ω and event algebra $\mathcal{E}$. Then $\vec{X} = (X_1, \dots, X_n)$ is called a *random vector* with

$$\vec{X}(\omega) \;=\; (X_1(\omega), \dots, X_n(\omega))$$

- Shorthand notation:

$$P(\vec{X} = (x_1, \dots, x_n)) \;=\; P(X_1 = x_1, \dots, X_n = x_n) \;=\; P(x_1, \dots, x_n)$$

- Definition:

$$\begin{aligned} P(X_1 = x_1, \dots, X_n = x_n) \;&=\; P\Big(\Big\{\; \omega \in \Omega \;\Big|\; \bigwedge_{i=1}^{n} X_i(\omega) = x_i \;\Big\}\Big) \\ &=\; P\Big(\; \bigcap_{i=1}^{n} \{X_i = x_i\} \;\Big) \end{aligned}$$

Joint Probabilities

- Example: $P(\mathsf{Headache}, \mathsf{Weather})$ is the *joint probability distribution* of both random variables and consists of the following table:

	$\mathsf{h} \mathrel{\widehat{=}} \mathsf{Headache} = \mathsf{true}$	$\neg\mathsf{h} \mathrel{\widehat{=}} \mathsf{Headache} = \mathsf{false}$
$\mathsf{Weather} = \mathsf{sunny}$	$P(\mathsf{W} = \mathsf{sunny} \wedge \mathsf{h})$	$P(\mathsf{W} = \mathsf{sunny} \wedge \neg\mathsf{h})$
$\mathsf{Weather} = \mathsf{rainy}$	$P(\mathsf{W} = \mathsf{rainy} \wedge \mathsf{h})$	$P(\mathsf{W} = \mathsf{rainy} \wedge \neg\mathsf{h})$
$\mathsf{Weather} = \mathsf{cloudy}$	$P(\mathsf{W} = \mathsf{cloudy} \wedge \mathsf{h})$	$P(\mathsf{W} = \mathsf{cloudy} \wedge \neg\mathsf{h})$
$\mathsf{Weather} = \mathsf{snowy}$	$P(\mathsf{W} = \mathsf{snowy} \wedge \mathsf{h})$	$P(\mathsf{W} = \mathsf{snowy} \wedge \neg\mathsf{h})$

- All table cells sum up to unity.

Calculating with Joint Probabilities

All desired probabilities can be computed from a joint probability distribution.

	toothache	¬toothache
cavity	0.04	0.06
¬cavity	0.01	0.89

- Example: $$\begin{aligned} P(\mathsf{cavity} \lor \mathsf{toothache}) &= P(\mathsf{cavity} \land \mathsf{toothache}) \\ &+ P(\neg\mathsf{cavity} \land \mathsf{toothache}) \\ &+ P(\mathsf{cavity} \land \neg\mathsf{toothache}) = 0.11 \end{aligned}$$

- Marginalizations: $$\begin{aligned} \mathrm{P}(\mathsf{cavity}) &= P(\mathsf{cavity} \land \mathsf{toothache}) \\ &+ P(\mathsf{cavity} \land \neg\mathsf{toothache}) = 0.10 \end{aligned}$$

- Conditioning:

$$P(\mathsf{cavity} \mid \mathsf{toothache}) = \frac{P(\mathsf{cavity} \land \mathsf{toothache})}{P(\mathsf{toothache})} = \frac{0.04}{0.04 + 0.01} = 0.80$$

Problems

- Easiness of computing all desired probabilities comes at an unaffordable price:

  Given n random variables with k possible values each, the joint probability distribution contains k^n entries which is infeasible in practical applications.

- Hard to handle.
- Hard to estimate.

Therefore:

1. Is there a more *dense* representation of joint probability distributions?
2. Is there a more *efficient* way of processing this representation?

- The answer is *no* for the general case, however, certain dependencies and independencies can be exploited to reduce the number of parameters to a practical size.

Stochastic Independence

- Two events A and B are *stochastically independent* iff

$$
\begin{aligned}
P(A \wedge B) \;&=\; P(A) \cdot P(B) \\
&\Leftrightarrow \\
P(A \mid B) \;=\; P(A) \;&=\; P(A \mid \overline{B})
\end{aligned}
$$

- Two random variables X and Y are *stochastically independent* iff

$$
\begin{aligned}
\forall x \in \mathrm{dom}(X) : \forall y \in \mathrm{dom}(Y) : \qquad P(X = x, Y = y) \;&=\; P(X = x) \cdot P(Y = y) \\
&\Leftrightarrow \\
\forall x \in \mathrm{dom}(X) : \forall y \in \mathrm{dom}(Y) : \qquad P(X = x \mid Y = y) \;&=\; P(X = x)
\end{aligned}
$$

- Shorthand notation: $P(X, Y) = P(X) \cdot P(Y)$.
  Note the formal difference between $P(A) \in [0, 1]$ and $P(X) \in [0, 1]^{|\mathrm{dom}(X)|}$.

Conditional Independence

- Let X, Y and Z be three random variables. We call X *and* Y *conditionally independent given* Z, iff the following condition holds:

$$\forall x \in \mathrm{dom}(X) : \forall y \in \mathrm{dom}(Y) : \forall z \in \mathrm{dom}(Z) :$$
$$P(X = x, Y = y \mid Z = z) \;=\; P(X = x \mid Z = z) \cdot P(Y = y \mid Z = z)$$

- Shorthand notation: $X \perp\!\!\!\perp_P Y \mid Z$

- Let $\boldsymbol{X} = \{A_1, \ldots, A_k\}$, $\boldsymbol{Y} = \{B_1, \ldots, B_l\}$ and $\boldsymbol{Z} = \{C_1, \ldots, C_m\}$ be three disjoint sets of random variables. We call $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *conditionally independent given* $\boldsymbol{Z}$, iff

$$P(\boldsymbol{X}, \boldsymbol{Y} \mid \boldsymbol{Z}) \;=\; P(\boldsymbol{X} \mid \boldsymbol{Z}) \cdot P(\boldsymbol{Y} \mid \boldsymbol{Z}) \Leftrightarrow P(\boldsymbol{X} \mid \boldsymbol{Y}, \boldsymbol{Z}) \;=\; P(\boldsymbol{X} \mid \boldsymbol{Z})$$

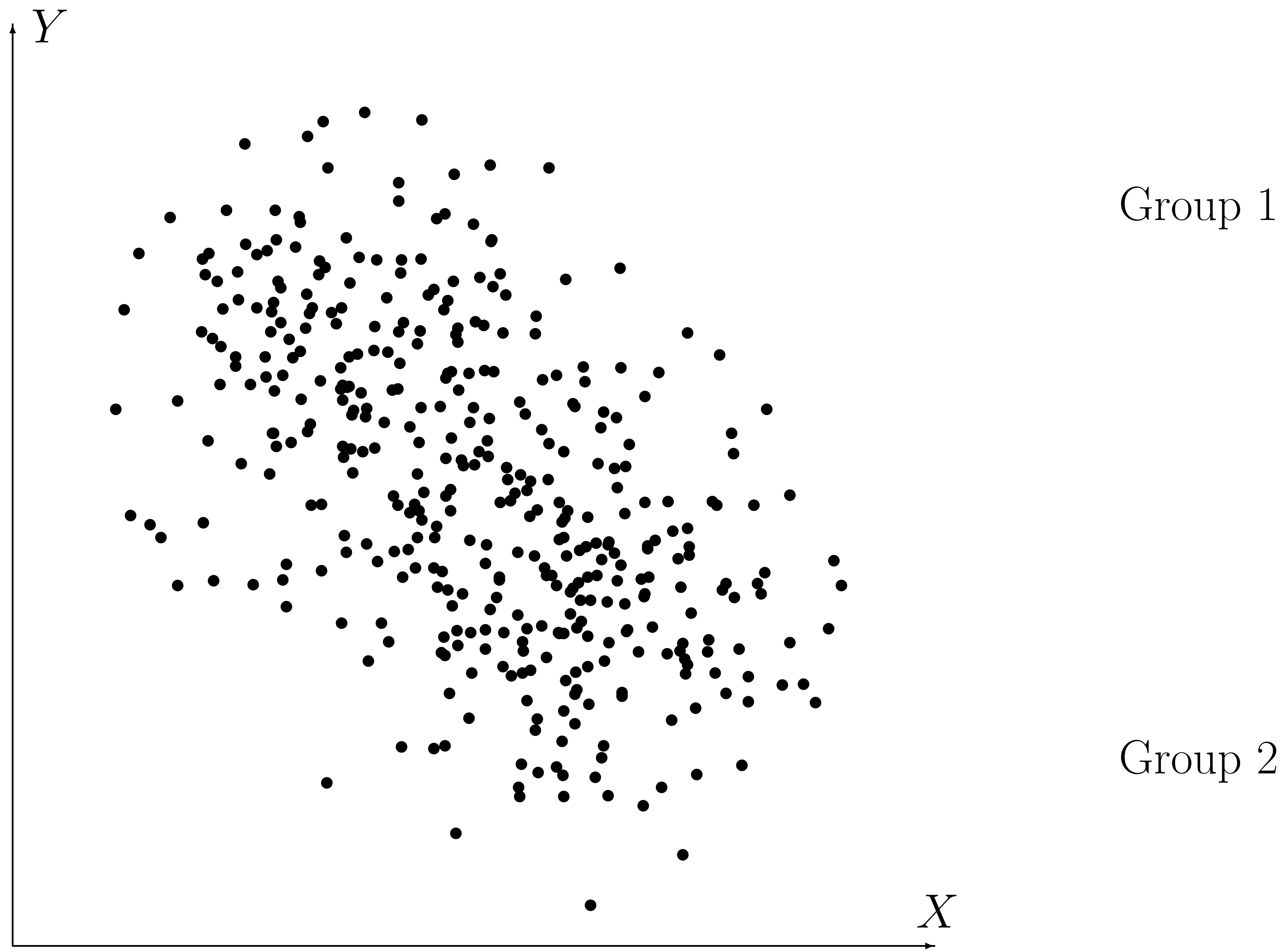
- Shorthand notation: $\boldsymbol{X} \perp\!\!\!\perp_P \boldsymbol{Y} \mid \boldsymbol{Z}$

Conditional Independence

- The complete condition for $\boldsymbol{X} \perp\!\!\!\perp_P \boldsymbol{Y} \mid \boldsymbol{Z}$ would read as follows:

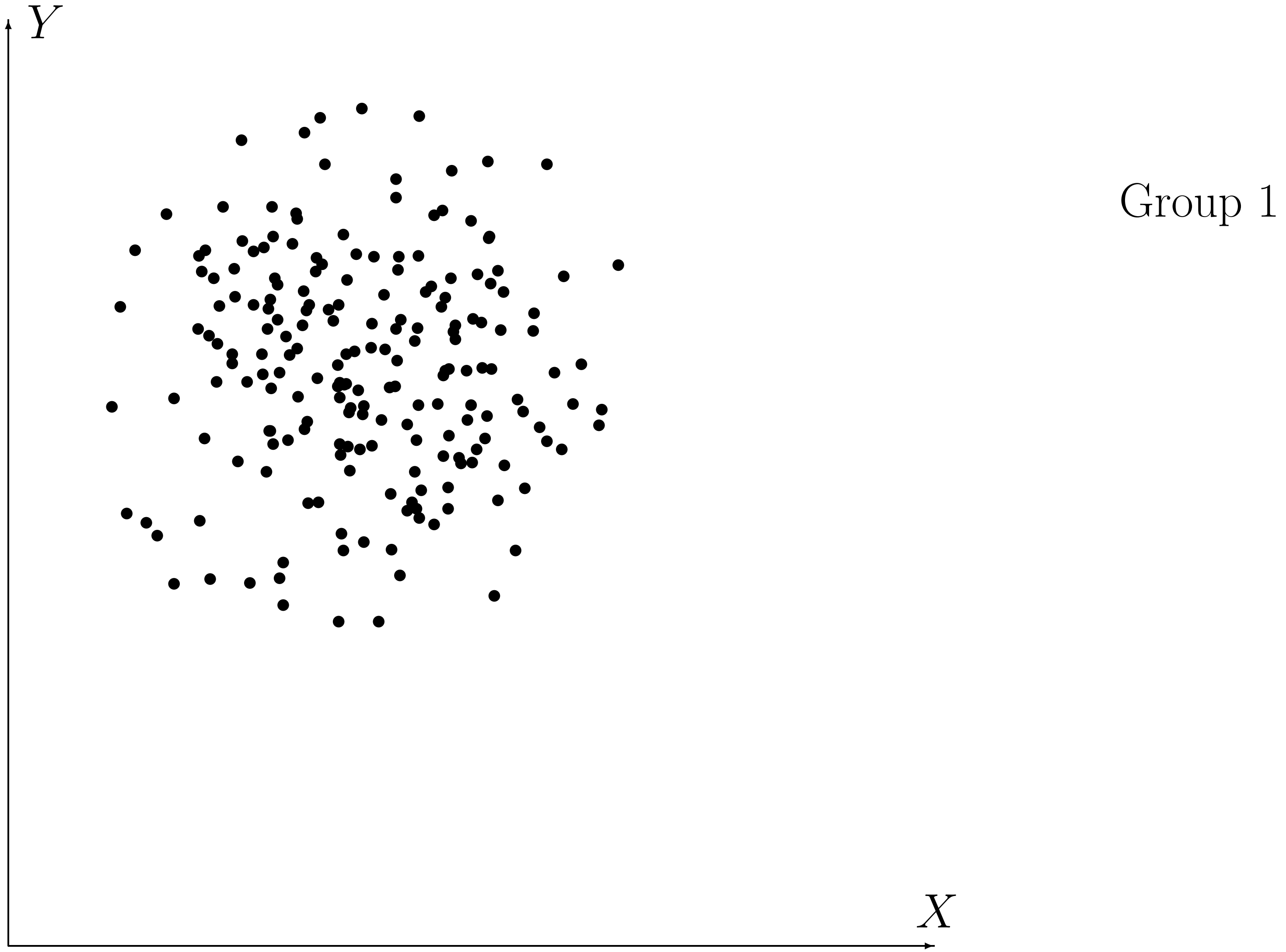
$$
\begin{aligned}
&\forall a_1 \in \mathrm{dom}(A_1) : \cdots \forall a_k \in \mathrm{dom}(A_k) : \\
&\quad \forall b_1 \in \mathrm{dom}(B_1) : \cdots \forall b_l \in \mathrm{dom}(B_l) : \\
&\quad\quad \forall c_1 \in \mathrm{dom}(C_1) : \cdots \forall c_m \in \mathrm{dom}(C_m) : \\
&\quad\quad\quad P(A_1 = a_1, \ldots, A_k = a_k, B_1 = b_1, \ldots, B_l = b_l \mid C_1 = c_1, \ldots, C_m = c_m) \\
&\quad\quad\quad\quad = P(A_1 = a_1, \ldots, A_k = a_k \mid C_1 = c_1, \ldots, C_m = c_m) \\
&\quad\quad\quad\quad\; \cdot P(B_1 = b_1, \ldots, B_l = b_l \mid C_1 = c_1, \ldots, C_m = c_m)
\end{aligned}
$$

- Remarks:
  1. If $\boldsymbol{Z} = \emptyset$ we get (unconditional) independence.
  2. We do not use curly braces ($\{\}$) for the sets if the context is clear. Likewise, we use X instead of $\boldsymbol{X}$ to denote sets.



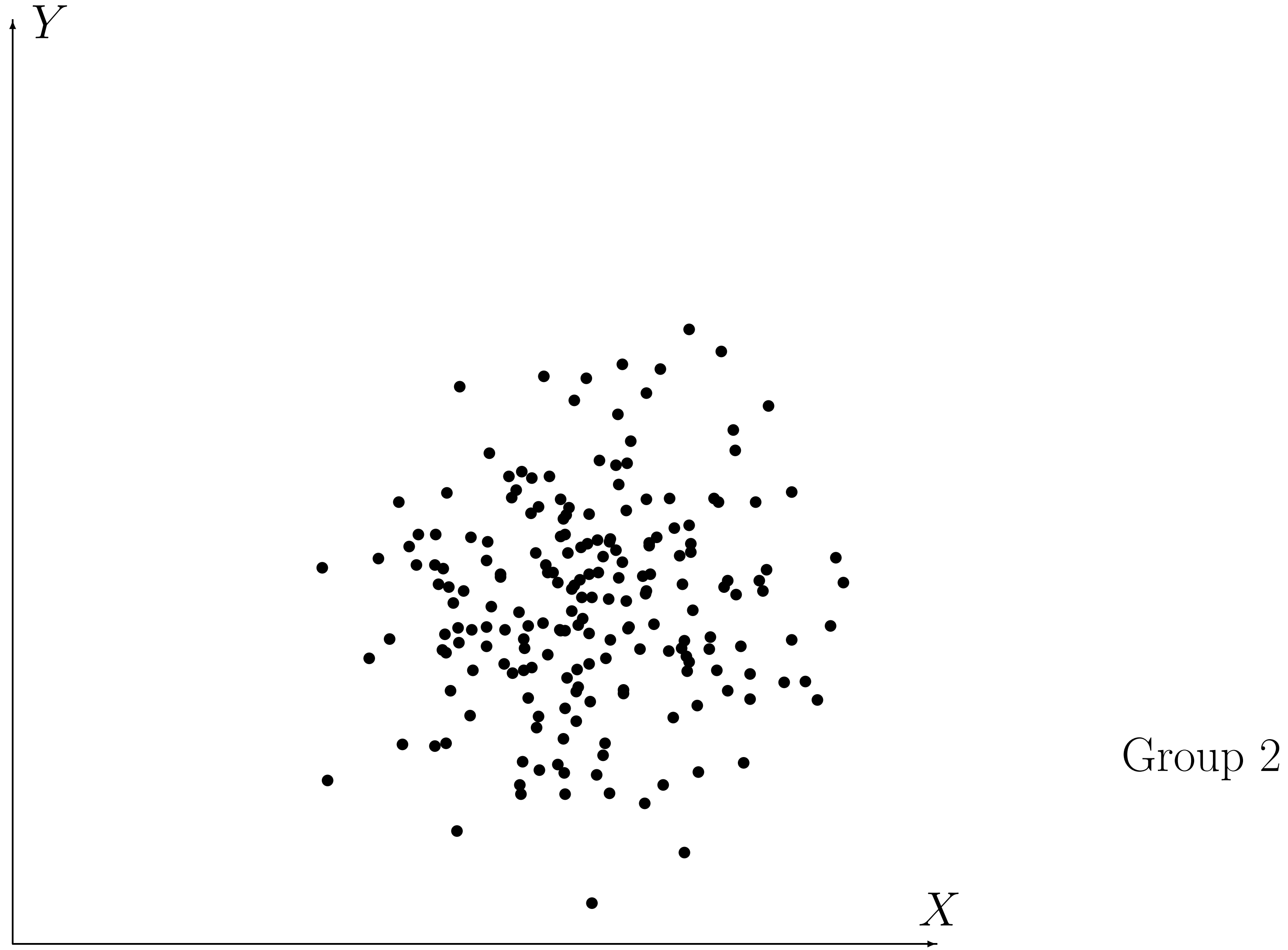
(Weak) Dependence in the entire dataset: X and Y dependent.

Conditional Independence — Example 1



No Dependence in Group 1: X and Y conditionally independent given Group 1.

Conditional Independence — Example 1



No Dependence in Group 2: X and Y conditionally independent given Group 2.

Conditional Independence — Example 2

- $\mathrm{dom}(G) = \{\mathsf{mal}, \mathsf{fem}\}$ Geschlecht (gender)
- $\mathrm{dom}(S) = \{\mathsf{sm}, \overline{\mathsf{sm}}\}$ Raucher (smoker)
- $\mathrm{dom}(M) = \{\mathsf{mar}, \overline{\mathsf{mar}}\}$ Verheiratet (married)
- $\mathrm{dom}(P) = \{\mathsf{preg}, \overline{\mathsf{preg}}\}$ Schwanger (pregnant)

p_{GSMP}		G = mal		G = fem	
		S = sm	S = $\overline{\mathsf{sm}}$	S = sm	S = $\overline{\mathsf{sm}}$
M = mar	P = preg	0	0	0.01	0.05
	P = $\overline{\mathsf{preg}}$	0.04	0.16	0.02	0.12
M = $\overline{\mathsf{mar}}$	P = preg	0	0	0.01	0.01
	P = $\overline{\mathsf{preg}}$	0.10	0.20	0.07	0.21

Conditional Independence — Example 2

$$P(\mathsf{G}=\mathsf{fem}) = P(\mathsf{G}=\mathsf{mal}) = 0.5 \qquad P(\mathsf{P}=\mathsf{preg}) = 0.08$$
$$P(\mathsf{S}=\mathsf{sm}) = 0.25 \qquad P(\mathsf{M}=\mathsf{mar}) = 0.4$$

- **Gender** and **Smoker** are not independent:

$$P(\mathsf{G}=\mathsf{fem} \mid \mathsf{S}=\mathsf{sm}) = 0.44 \neq 0.5 = P(\mathsf{G}=\mathsf{fem})$$

- **Gender** and **Marriage** are marginally independent but conditionally dependent given **Pregnancy**:

$$P(\mathsf{fem}, \mathsf{mar} \mid \overline{\mathsf{preg}}) \approx 0.152 \quad \neq \quad 0.169 \approx P(\mathsf{fem} \mid \overline{\mathsf{preg}}) \cdot P(\mathsf{mar} \mid \overline{\mathsf{preg}})$$

Bayes Theorem

- Product Rule (for events A and B):

$$P(A \cap B) = P(A \mid B)P(B) \quad \text{and} \quad P(A \cap B) = P(B \mid A)P(A)$$

- Equating the right-hand sides:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

- For random variables X and Y:

$$\forall x \forall y : \quad P(Y = y \mid X = x) = \frac{P(X = x \mid Y = y)P(Y = y)}{P(X = x)}$$

- Generalization concerning background knowledge/evidence E:

$$P(Y \mid X, E) = \frac{P(X \mid Y, E)P(Y \mid E)}{P(X \mid E)}$$

Bayes Theorem — Application

$$P(\mathsf{toothache} \mid \mathsf{cavity}) = 0.4$$
$$P(\mathsf{cavity}) = 0.1$$
$$P(\mathsf{toothache}) = 0.05$$

$$P(\mathsf{cavity} \mid \mathsf{toothache}) = \frac{0.4 \cdot 0.1}{0.05} = 0.8$$

Why not estimate $P(\mathsf{cavity} \mid \mathsf{toothache})$ right from the start?

- Causal knowledge like $P(\mathsf{toothache} \mid \mathsf{cavity})$ is more robust than diagnostic knowledge $P(\mathsf{cavity} \mid \mathsf{toothache})$.
- The causality $P(\mathsf{toothache} \mid \mathsf{cavity})$ is independent of the a priori probabilities $P(\mathsf{toothache})$ and $P(\mathsf{cavity})$.
- If $P(\mathsf{cavity})$ rose in a caries epidemic, the causality $P(\mathsf{toothache} \mid \mathsf{cavity})$ would remain constant whereas both $P(\mathsf{cavity} \mid \mathsf{toothache})$ and $P(\mathsf{toothache})$ would increase according to $P(\mathsf{cavity})$.
- A physician, after having estimated $P(\mathsf{cavity} \mid \mathsf{toothache})$, would not know a rule for updating.

Relative Probabilities

Assumption:
We would like to consider the probability of the diagnosis $\mathsf{GumDisease}$ as well.

$$\begin{aligned} P(\mathsf{toothache} \mid \mathsf{gumdisease}) &= 0.7 \\ P(\mathsf{gumdisease}) &= 0.02 \end{aligned}$$

Which diagnosis is more probable?

If we are interested in *relative probabilities* only (which may be sufficient for some decisions), $P(\mathsf{toothache})$ needs not to be estimated:

$$\begin{aligned} \frac{P(C \mid T)}{P(G \mid T)} &= \frac{P(T \mid C)P(C)}{P(T)} \cdot \frac{P(T)}{P(T \mid G)P(G)} \\ &= \frac{P(T \mid C)P(C)}{P(T \mid G)P(G)} = \frac{0.4 \cdot 0.1}{0.7 \cdot 0.02} \\ &= 28.57 \end{aligned}$$

Normalization

If we are interested in the absolute probability of $P(C \mid T)$ but do not know $P(T)$, we may conduct a complete case analysis (according C) and exploit the fact that $P(C \mid T) + P(\neg C \mid T) = 1$.

$$P(C \mid T) = \frac{P(T \mid C)P(C)}{P(T)}$$

$$P(\neg C \mid T) = \frac{P(T \mid \neg C)P(\neg C)}{P(T)}$$

$$1 = P(C \mid T) + P(\neg C \mid T) = \frac{P(T \mid C)P(C)}{P(T)} + \frac{P(T \mid \neg C)P(\neg C)}{P(T)}$$

$$P(T) = P(T \mid C)P(C) + P(T \mid \neg C)P(\neg C)$$

Normalization

- Plugging into the equation for $P(C \mid T)$ yields:

$$P(C \mid T) = \frac{P(T \mid C)P(C)}{P(T \mid C)P(C) + P(T \mid \neg C)P(\neg C)}$$

- For general random variables, the equation reads:

$$P(Y = y \mid X = x) = \frac{P(X = x \mid Y = y)P(Y = y)}{\sum\limits_{\forall y' \in \mathrm{dom}(Y)} P(X = x \mid Y = y')P(Y = y')}$$

- Note the "loop variable" y'. Do not confuse with y.

Multiple Evidences

- The patient complains about a toothache. From this first evidence the dentist infers:

$$P(\mathsf{cavity} \mid \mathsf{toothache}) = 0.8$$

- The dentist palpates the tooth with a metal probe which catches into a fracture:

$$P(\mathsf{cavity} \mid \mathsf{fracture}) = 0.95$$

- Both conclusions might be inferred via Bayes rule. But what does the combined evidence yield? Using Bayes rule further, the dentist might want to determine:

$$P(\mathsf{cavity} \mid \mathsf{toothache} \land \mathsf{fracture}) = \frac{P(\mathsf{toothache} \land \mathsf{fracture} \mid \mathsf{cavity}) \cdot P(\mathsf{cavity})}{P(\mathsf{toothache} \land \mathsf{fracture})}$$

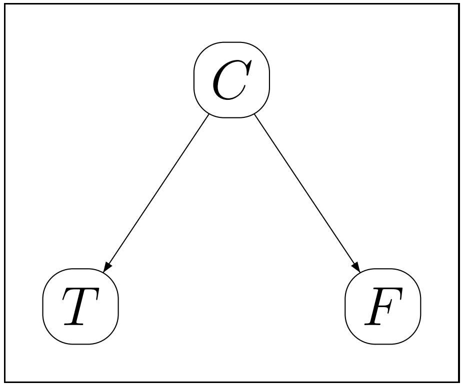
Multiple Evidences

Problem:
He needs $P(\textsf{toothache} \wedge \textsf{catch} \mid \textsf{cavity})$, i. e. diagnostics knowledge for all combinations of symptoms in general. Better incorporate evidences step-by-step:

$$P(Y \mid X, E) = \frac{P(X \mid Y, E)P(Y \mid E)}{P(X \mid E)}$$

Abbreviations:

- C — cavity
- T — toothache
- F — fracture



Objective:
Computing $P(C \mid T, F)$ with just causal statements of the form $P(\,\cdot \mid C)$ and under exploitation of independence relations among the variables.

Multiple Evidences

- A priori: $P(C)$
- Evidence toothache: $P(C \mid T) = P(C)\,\dfrac{P(T \mid C)}{P(T)}$
- Evidence fracture: $P(C \mid T, F) = P(C \mid T)\,\dfrac{P(F \mid C, T)}{P(F \mid T)}$

$$T \perp\!\!\!\perp F \mid C \quad \Leftrightarrow \quad P(F \mid C, T) = P(F \mid C)$$

$$P(C \mid T, F) \;=\; P(C)\,\frac{P(T \mid C)}{P(T)}\,\frac{P(F \mid C)}{P(F \mid T)}$$

Seems that we still have to cope with symptom inter-dependencies?!

Multiple Evidences

- Compound equation from last slide:

$$\begin{aligned} P(C \mid T, F) &= P(C)\,\frac{P(T \mid C)\; P(F \mid C)}{P(T)\; P(F \mid T)} \\ &= P(C)\,\frac{P(T \mid C)\; P(F \mid C)}{P(F, T)} \end{aligned}$$

- $P(F, T)$ is a normalizing constant and can be computed if $P(F \mid \neg C)$ and $P(T \mid \neg C)$ are known:

$$P(F, T) = \underbrace{P(F, T \mid C)}_{P(F|C)P(T|C)}\; P(C) + \underbrace{P(F, T \mid \neg C)}_{P(F|\neg C)P(T|\neg C)}\; P(\neg C)$$

- Therefore, we finally arrive at the following solution...

Multiple Evidences

$$P(C \mid F, T) = \frac{\boxed{P(C)}\ \boxed{P(T \mid C)}\ \boxed{P(F \mid C)}}{\boxed{P(F \mid C)}\ \boxed{P(T \mid C)}\ \boxed{P(C)} + \boxed{P(F \mid \neg C)}\ \boxed{P(T \mid \neg C)}\ \boxed{P(\neg C)}}$$

Note that we only use causal probabilities $P(\ \cdot \mid C)$ together with the a priori (marginal) probabilities $P(C)$ and $P(\neg C)$.

Multiple Evidences — Summary

Multiple evidences can be treated by reduction on

- a priori probabilities
- (causal) conditional probabilities for the evidence
- under assumption of conditional independence

General rule:

$$P(Z \mid X, Y) \;=\; \alpha \; P(Z) \; P(X \mid Z) \; P(Y \mid Z)$$

for X and Y conditionally independent given Z and with normalizing constant α.

Monty Hall Puzzle

Marylin Vos Savant in her riddle column in the New York Times:

You are a candidate in a game show and have to choose between three doors. Behind one of them is a Porsche, whereas behind the other two there are goats. After you chose a door, the host Monty Hall (who knows what is behind each door) opens another (not your chosen one) door with a goat. Now you have the choice between keeping your chosen door or choose the remaining one.

Which decision yields the best chance of winning the Porsche?

Monty Hall Puzzle

G	You win the Porsche.
R	You revise your decision.
A	Behind your initially chosen door is (and remains) the Porsche.

$$
\begin{aligned}
P(G \mid R) &= P(G, A \mid R) + P(G, \overline{A} \mid R) \\
&= P(G \mid A, R) P(A \mid R) + P(G \mid \overline{A}, R) P(\overline{A} \mid R) \\
&= 0 \cdot P(A \mid R) + 1 \cdot P(\overline{A} \mid R) \\
&= P(\overline{A} \mid R) = P(\overline{A}) = \frac{2}{3}
\end{aligned}
$$

$$
\begin{aligned}
P(G \mid \overline{R}) &= P(G, A \mid \overline{R}) + P(G, \overline{A} \mid \overline{R}) \\
&= P(G \mid A, \overline{R}) P(A \mid \overline{R}) + P(G \mid \overline{A}, \overline{R}) P(\overline{A} \mid \overline{R}) \\
&= 1 \cdot P(A \mid \overline{R}) + 0 \cdot P(\overline{A} \mid \overline{R}) \\
&= P(A \mid \overline{R}) = P(A) = \frac{1}{3}
\end{aligned}
$$

Example: C = Patient takes medication, E = patient recovers

	E	$\neg E$	$\sum$	Recovery rate
C	20	20	40	50%
$\neg C$	16	24	40	40%
$\sum$	36	44	80	

Men	E	$\neg E$	$\sum$	Rec.rate	Women	E	$\neg E$	$\sum$	Rec.rate
C	18	12	30	60%	C	2	8	10	20%
$\neg C$	7	3	10	70%	$\neg C$	9	21	30	30%
	25	15	40			11	29	40	

$$\begin{array}{rrcl} & P(E \mid C) & > & P(E \mid \neg C) \\ \text{but} & P(E \mid C, M) & < & P(E \mid \neg C, M) \\ & P(E \mid C, W) & < & P(E \mid \neg C, W) \end{array}$$

Probabilistic Reasoning

- Probabilistic reasoning is difficult and may be problematic:
  - $P(A \wedge B)$ is not determined simply by $P(A)$ and $P(B)$:
    $P(A) = P(B) = 0.5 \quad \Rightarrow \quad P(A \wedge B) \in [0, 0.5]$
  - $P(C \mid A) = x, P(C \mid B) = y \quad \Rightarrow \quad P(C \mid A \wedge B) \in [0, 1]$
    Probabilistic logic is *not truth functional*!
- Central problem: How does additional information affect the current knowledge?
  I. e., if $P(B \mid A)$ is known, what can be said about $P(B \mid A \wedge C)$?
- High complexity: n propositions $\rightarrow 2^n$ full conjunctives
- Hard to specify these probabilities.

Summary

- Uncertainty is inevitable in complex and dynamic scenarios that force agents to cope with ignorance.
- Probabilities express the agent's inability to vote for a definitive decision. They model the degree of belief.
- If an agent violates the axioms of probability, it may exhibit irrational behavior in certain circumstances.
- The Bayes rule is used to derive unknown probabilities from present knowledge and new evidence.
- Multiple evidences can be effectively included into computations exploiting conditional independencies.

Probabilistic Causal Networks

The Big Objective(s)

In a wide variety of application fields two main problems need to be addressed over and over:

1. **How can (expert) knowledge of complex domains be efficiently represented?**
2. **How can inferences be carried out within these representations?**
3. **How can such representations be (automatically) extracted from collected data?**

We will deal with all three questions during the lecture.

Example 1: Planning in car manufacturing

Available information

- "Engine type e_1 can only be combined with transmission t_2 or t_5."
- "Transmission t_5 requires crankshaft c_2."
- "Convertibles have the same set of radio options as SUVs."

Possible questions/inferences:

- "Can a station wagon with engine e_4 be equipped with tire set y_6?"
- "Supplier S_8 failed to deliver on time. What production line has to be modified and how?"
- "Are there any peculiarities within the set of cars that suffered an aircondition failure?"

Example 2: Medical reasoning

Available information:

- "Malaria is much less likely than flu."
- "Flu causes cough and fever."
- "Nausea can indicate malaria as well as flu."
- "Nausea never indicated pneunomia before."

Possible questions/inferences

- "The patient has fever. How likely is he to have malaria?"
- "How much more likely does flu become if we can exclude malaria?"

Common Problems

Both scenarios share some severe problems:

- **Large Data Space**
  It is intractable to store all value combinations, i. e. all car part combinations or inter-disease dependencies.

  (Example: VW Bora has 10^{200} theoretical value combinations*)

- **Sparse Data Space**
  Even if we could handle such a space, it would be extremely sparse, i. e. it would be impossible to find good estimates for all the combinations.

  (Example: with 100 diseases and 200 symptoms, there would be about 10^{62} different scenarios for which we had to estimate the probability.*)

* The number of particles in the observable universe is estimated to be between 10^{78} and 10^{85}.

Idea to Solve the Problems

- **Given:** A large (high-dimensional) distribution δ representing the domain knowledge.

- **Desired:** A set of smaller (lower-dimensional) distributions $\{\delta_1, \ldots, \delta_s\}$ (maybe overlapping) from which the original δ *could* be reconstructed with no (or as few as possible) errors.

- With such a decomposition we can draw any conclusions from $\{\delta_1, \ldots, \delta_s\}$ that could be inferred from δ — without, however, actually reconstructing it.

Example: Car Manufacturing

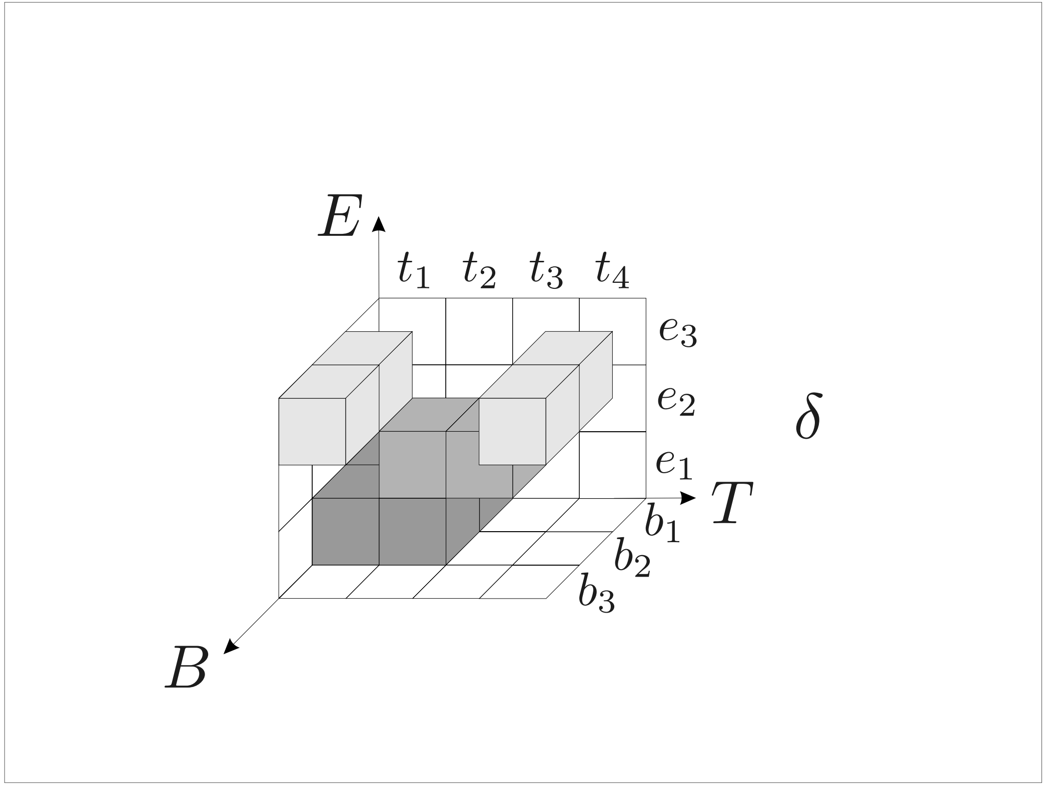
- Let us consider a car configuration is described by three attributes:
  - Engine E, $\mathrm{dom}(E) = \{e_1, e_2, e_3\}$
  - Breaks B, $\mathrm{dom}(B) = \{b_1, b_2, b_3\}$
  - Tires T, $\mathrm{dom}(T) = \{t_1, t_2, t_3, t_4\}$
- Therefore the set of all (theoretically) possible car configurations is:

$$\Omega = \mathrm{dom}(E) \times \mathrm{dom}(B) \times \mathrm{dom}(T)$$

- Since not all combinations are technically possible (or wanted by marketing) a set of rules is used to cancel out invalid combinations.

Example: Car Manufacturing

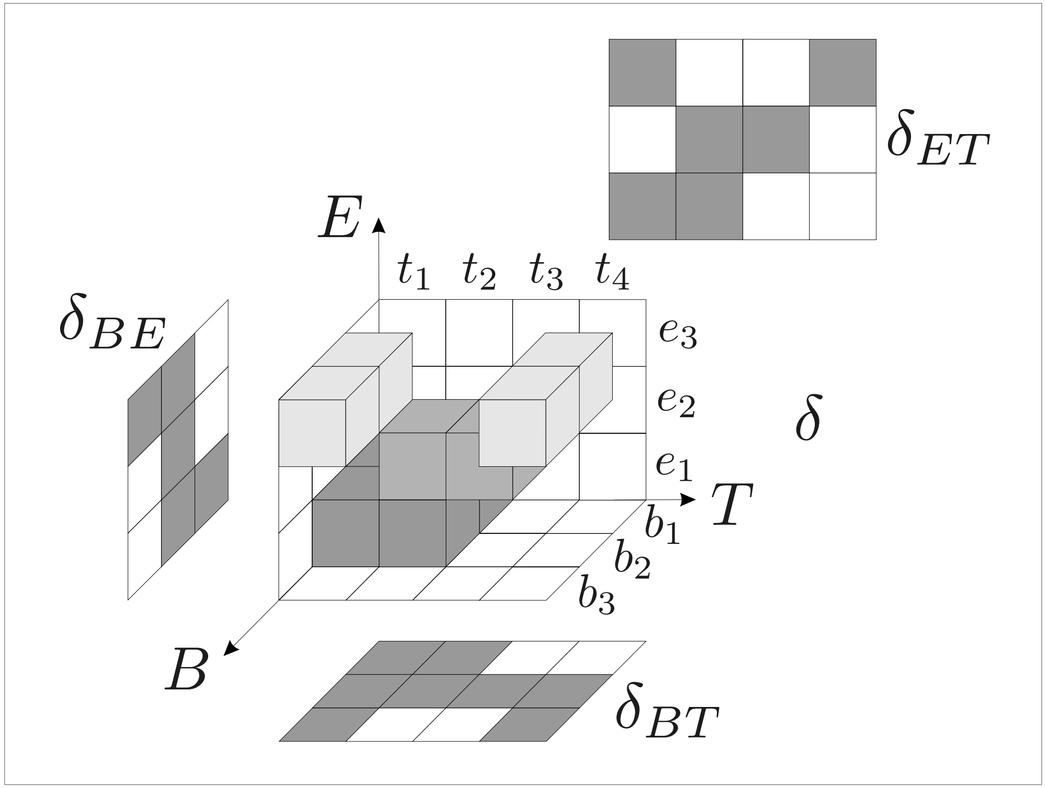
Possible car configurations



- Every cube designates a valid value combination.
- 10 car configurations in our model.
- Different colors are intended to distinguish the cubes only.

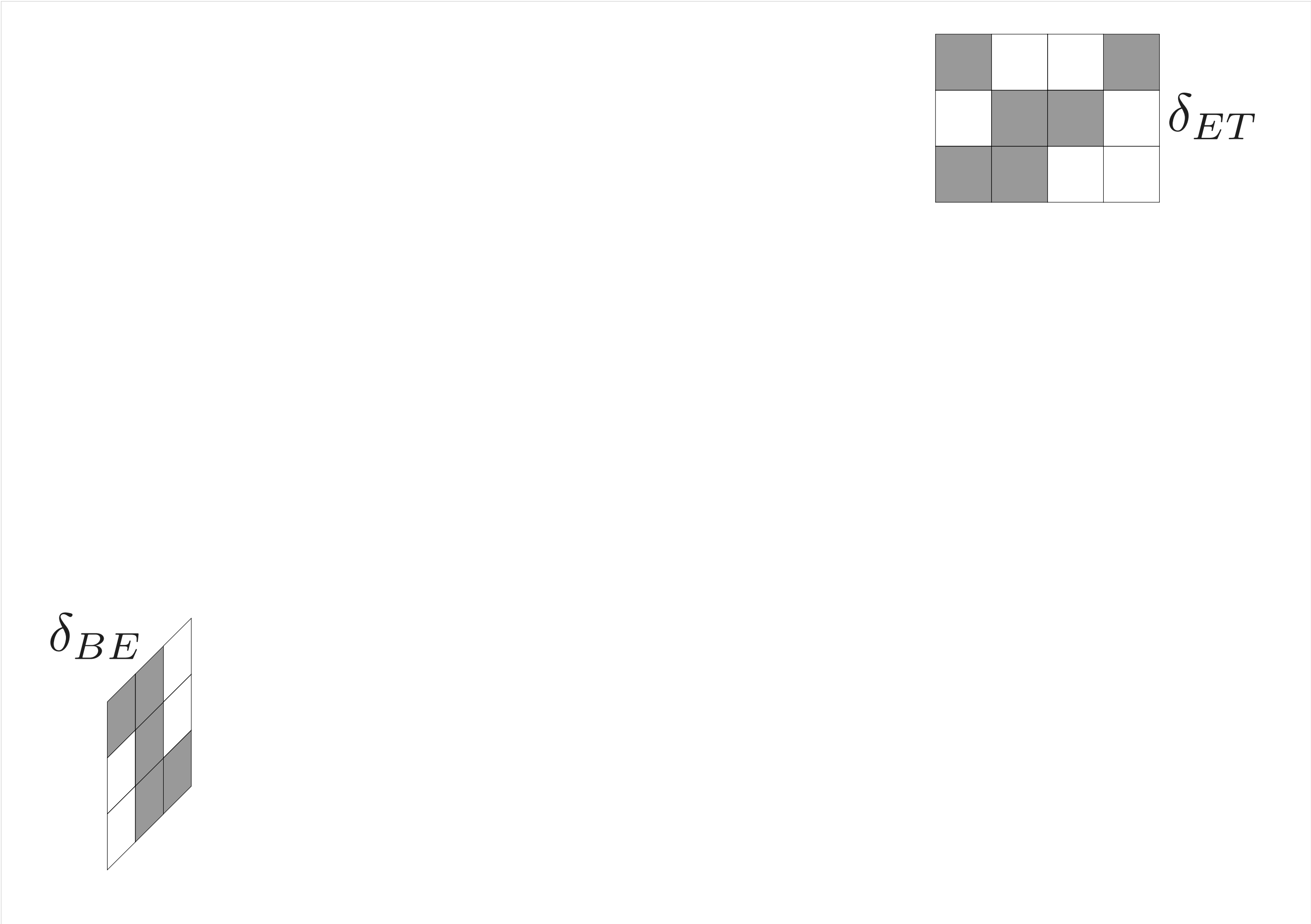
Example

2-D projections

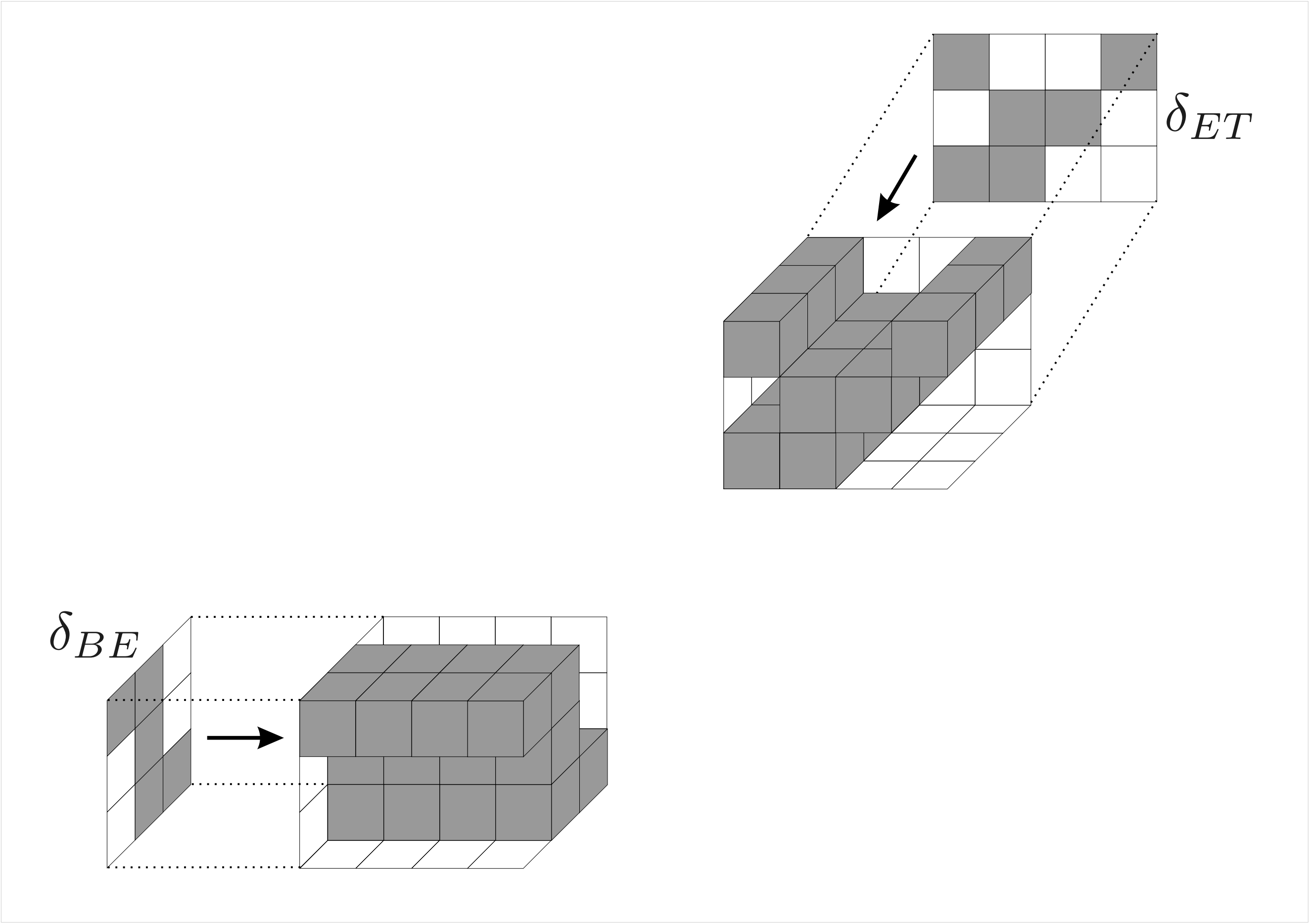


- Is it possible to reconstruct δ from the δ_i?

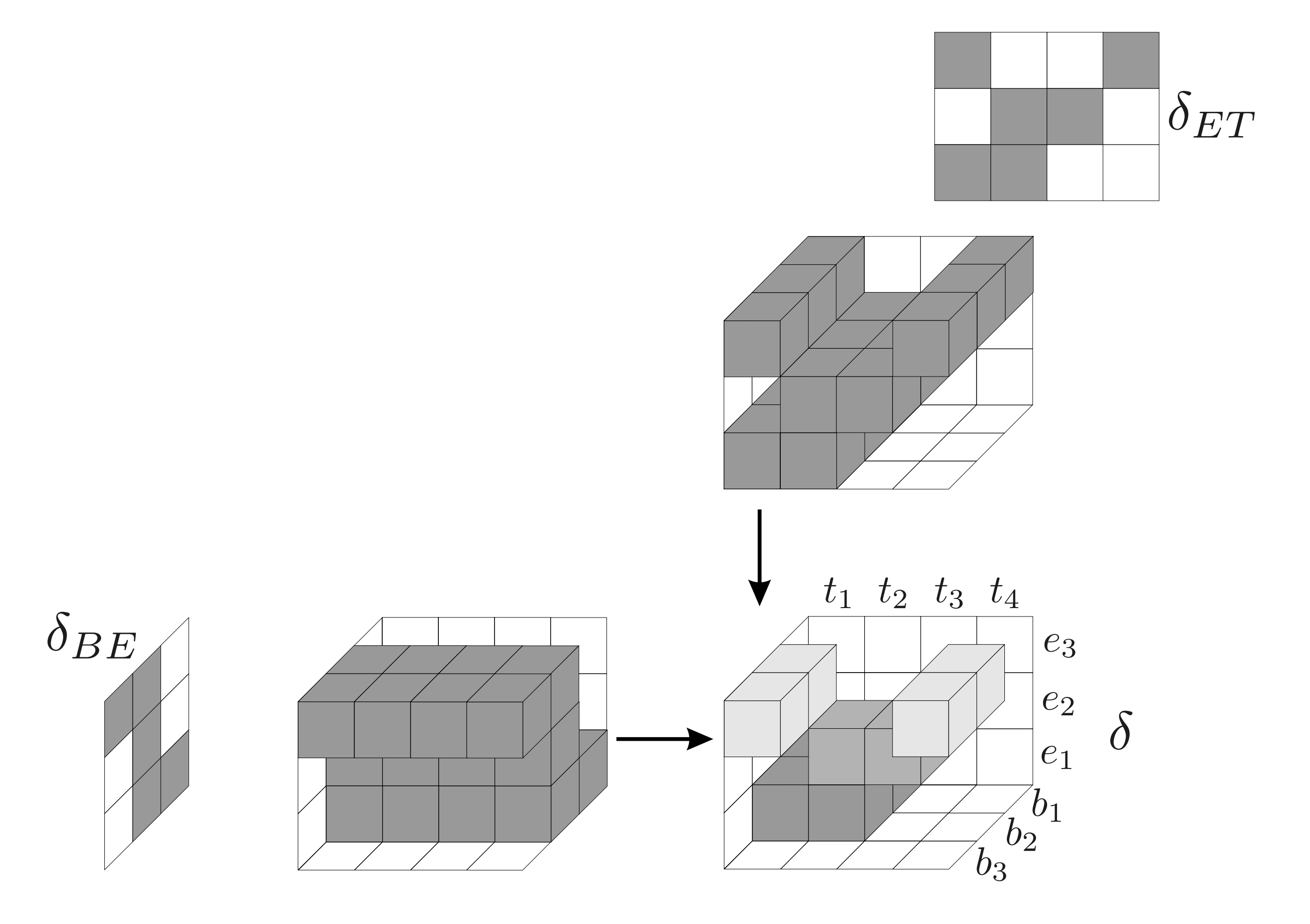
Example: Reconstruction of δ with δ_{BE} and δ_{ET}



Example: Reconstruction of δ with δ_{BE} and δ_{ET}



Example: Reconstruction of δ with δ_{BE} and δ_{ET}

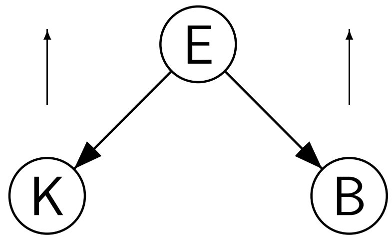


Example — Qualitative Aspects

- Lecture theatre in winter: Waiting for Mr. K and Mr. B.
  Not clear whether there is ice on the roads.

- 3 variables:
  - E road condition: $\mathrm{dom}(\mathsf{E}) = \{\mathsf{ice}, \neg\mathsf{ice}\}$
  - K K had an accident: $\mathrm{dom}(\mathsf{K}) = \{\mathsf{yes}, \mathsf{no}\}$
  - B B had an accident: $\mathrm{dom}(\mathsf{B}) = \{\mathsf{yes}, \mathsf{no}\}$

- Ignorance about these states is modelled via the observer's belief.

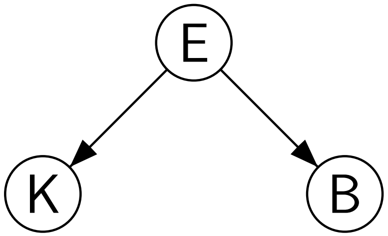


$\downarrow$ E influences K and B
(the more ice the more accidents)

$\uparrow$ Knowledge about accident increases belief in ice

Example

A priori knowledge	Evidence	Inferences
E unknown	B has accident	$\Rightarrow$ E $=$ ice more likely
		$\Rightarrow$ K has accident more likely
E $= \neg$ice	B has accident	$\Rightarrow$ no change in belief about E
		$\Rightarrow$ no change in belief about accident of K
E unknown		K and B dependent
E known		K and B independent



Causal Dependence vs. Reasoning

Rule: A entails B with certainty x: $\boxed{A \xrightarrow{x} B}$

- **Deduction** ($\rightarrow$):
  A and $A \xrightarrow{x} B$, therefore B more likely as effect (causality)

- **Abduction** ($\leftarrow$):
  B and $A \xrightarrow{x} B$, therefore A more likely as cause (no causality)

For this reason, the notion "dependency model" is to be preferred to "causal network".

Objective

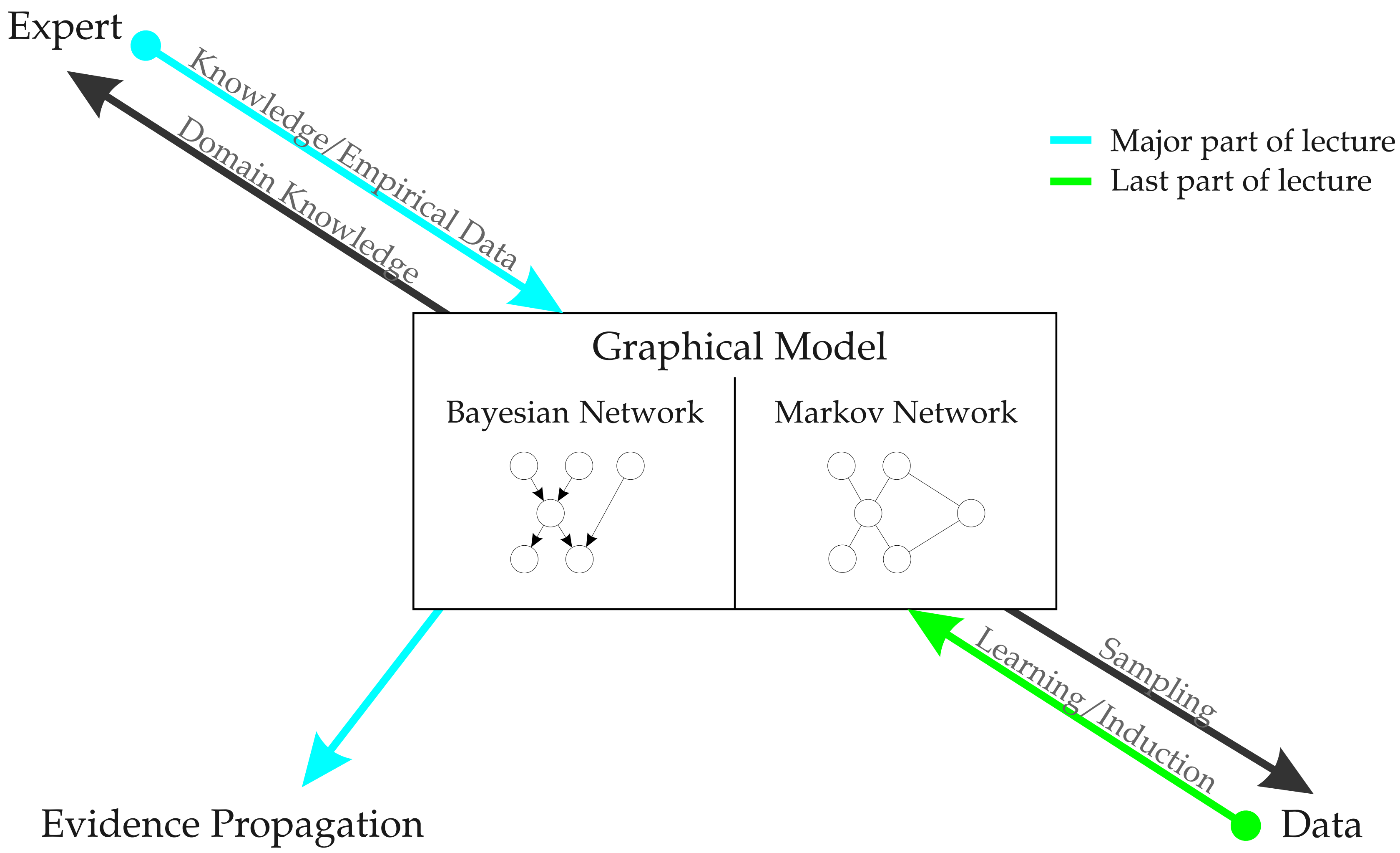
Is it possible to exploit local constraints (wherever they may come from — both structural and expert knowledge-based) in a way that allows for a decomposition of the large (intractable) distribution $P(X_1, \ldots, X_n)$ into several sub-structures $\{C_1, \ldots, C_m\}$ such that:

- The collective size of those sub-structures is much smaller than that of the original distribution P.
- The original distribution P is recomposable (with no or at least as few as possible errors) from these sub-structures in the following way:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{m} \Psi_i(c_i)$$

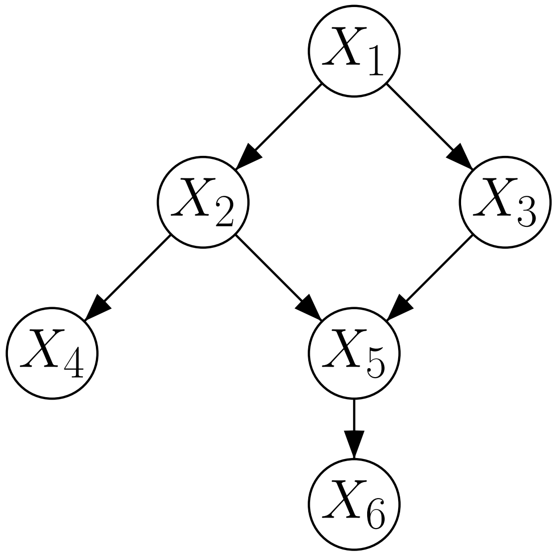
where c_i is an instantiation of C_i and $\Psi_i(c_i) \in \mathbb{R}^+$ a *factor potential.*

The Big Picture / Lecture Roadmap



Probabilistic Causal Networks

Probabilistic causal networks are directed acyclic graphs (DAGs) where the nodes represent propositions or variables and the directed edges model a direct causal dependence between the connected nodes. The strength of dependence is defined by conditional probabilities.

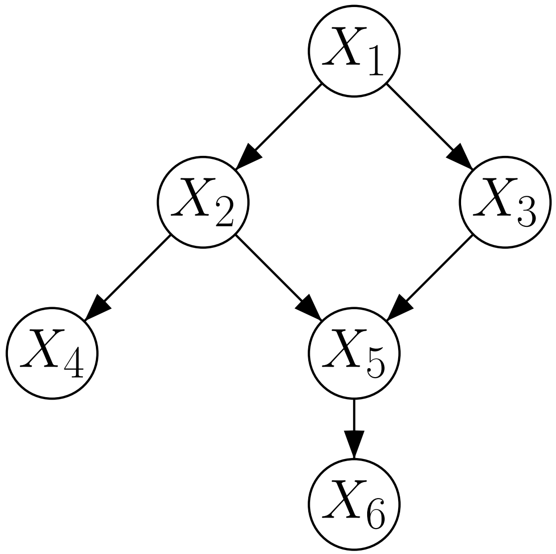


In general (according chain rule):

$$
\begin{aligned}
P(X_1, \ldots, X_6) = \; & P(X_6 \mid X_5, \ldots, X_1) \cdot \\
& P(X_5 \mid X_4, \ldots, X_1) \cdot \\
& P(X_4 \mid X_3, X_2, X_1) \cdot \\
& P(X_3 \mid X_2, X_1) \cdot \\
& P(X_2 \mid X_1) \cdot \\
& P(X_1)
\end{aligned}
$$

Probabilistic Causal Networks

Probabilistic causal networks are directed acyclic graphs (DAGs) where the nodes represent propositions or variables and the directed edges model a direct causal dependence between the connected nodes. The strength of dependence is defined by conditional probabilities.



According graph (independence structure):

$$\begin{aligned} P(X_1, \ldots, X_6) = \; & P(X_6 \mid X_5) \cdot \\ & P(X_5 \mid X_2, X_3) \cdot \\ & P(X_4 \mid X_2) \cdot \\ & P(X_3 \mid X_1) \cdot \\ & P(X_2 \mid X_1) \cdot \\ & P(X_1) \end{aligned}$$

Formal Framework

Nomenclature for the next slides:

- $X_1, \ldots, X_n$ Variables
  (properties, attributes, random variables, propositions)

- $\Omega_1, \ldots, \Omega_n$ respective finite domains
  (also designated with $\mathrm{dom}(X_i)$)

- $\Omega = \bigtimes_{i=1}^{n} \Omega_i$ Universe of Discourse (tuples that characterize objects described by $X_1, \ldots, X_n$)

- $\Omega_i = \{x_i^{(1)}, \ldots, x_i^{(n_i)}\}$ $n = 1, \ldots, n,\ n_i \in \mathbb{N}$

Formal Framework

- Let Ω^* be the real universe of objects under consideration (e. g. population of people, collection of cars, customer transactions, etc.). Then the random vector $\vec{X} = (X_1, \ldots, X_n)$ *describes* each element $\omega^* \in \Omega^*$ in terms of the universe of discourse Ω:

$$\vec{X} : \Omega^* \to \Omega \quad \text{with} \quad \vec{X}(\omega^*) = (X_1(\omega^*), \ldots, X_n(\omega^*))$$

- If $(\Omega^*, \mathcal{E}, Q)$ is an intrinsic probability space acting in the background, then it induces — in combination with $\vec{X}$ — a probability measure P over Ω:

$$\begin{aligned}
&\forall (x_1, \ldots, x_n) \in \Omega : \\
&\qquad P(\{(x_1, \ldots, x_n)\}) \;=\; P(X_1 = x_1, \ldots, X_n = x_n) \\
&\qquad\qquad\qquad\qquad\quad\;\; =\; Q(\{\omega^* \in \Omega^* \mid \bigwedge_{i=1}^{n} X_i = x_i\})
\end{aligned}$$

Formal Framework

- The product space $(\Omega, 2^\Omega, P)$ is unique iff $P(\{(x_1, \ldots, x_n)\})$ is specified for all $x_i \in \{x_i^{(1)}, \ldots, x_i^{(n_i)}\}$, $i = 1, \ldots, n$.

- When the distribution $P(X_1, \ldots, X_n)$ is given in tabular form, then $\prod_{i=1}^n |\Omega_i|$ entries are necessary.

- For variables with $|\Omega_i| \geq 2$ at least 2^n entries.

- The application of DAGs allows for the representation of existing (in)dependencies.

Constructing a DAG

input $P(X_1, \dots, X_n)$
output a unique DAG G

1: Set the nodes of G to $\{X_1, \dots, X_n\}$.

2: Choose a total ordering on the set of variables (e. g. $X_1 \prec X_2 \prec \cdots \prec X_n$)

3: For X_i find the smallest (uniquely determinable) set $S_i \subseteq \{X_1, \dots, X_n\}$ sucht that $P(X_i \mid S_i) = P(X_i \mid X_1 \dots, X_{i-1})$.

4: Connect all nodes in S_i with X_i and store $P(X_i \mid S_i)$ as quantization of the dependencies for that node X_i (given its parents).

5: **return** G

Belief Network

- A *Belief Network* (V, E, P) consists of a set $V = \{X_1, \ldots, X_n\}$ of random variables and a set E of directed edges between the variables.
- Each variable has a finite set of mutual exclusive and collectively exhaustive states.
- The variables in combination with the edges form a directed, acyclich graph.
- Each variable with parent nodes $B_1, \ldots, B_m$ is assigned a potential table $P(A \mid B_1, \ldots, B_m)$.
- Note, that the connections between the nodes not necessarily express a causal relationship.
- For every belief network, the following equation holds:

$$P(V) = \prod_{v \in V : P(c(v)) > 0} P(v \mid c(v))$$

with $c(v)$ being the parent nodes of v.

Example

- Let a_1, a_2, a_3 be three blood groups and b_1, b_2, b_3 three indications of a blood group test.

$$\begin{array}{lll} \text{Variables:} & A \text{ (blood group)} & B \text{ (indication)} \\ \text{Domains:} & \Omega_A = \{a_1, a_2, a_3\} & \Omega_B = \{b_1, b_2, b_3\} \end{array}$$

- It is conjectured that there is a causal relationship between the variables.

- A and B constitute random variables w. r. t. $(\Omega^*, \mathcal{E}, Q)$.

$$\Omega = \Omega_A \times \Omega_B \qquad A : \Omega^* \to \Omega_A, \quad B : \Omega^* \to \Omega_B$$

- A, B and $(\Omega^*, \mathcal{E}, Q)$ induce the probability space $(\Omega, 2^\Omega, P)$ with

$$P\Big(\{(a, b)\}\Big) = Q\Big(\{\omega^* \in \Omega^* \mid A(\omega^*) = a \wedge B(\omega^*) = b\}\Big) :$$

$P(\{(a_i, b_j)\})$	b_1	b_2	b_3	$\sum$
a_1	0.64	0.08	0.08	0.8
a_2	0.01	0.08	0.01	0.1
a_3	0.01	0.01	0.08	0.1
$\sum$	0.66	0.17	0.17	1

$$A \longrightarrow B$$

$P(A, B) = P(B \mid A) \cdot P(A)$

We are dealing with a belief network.

Example

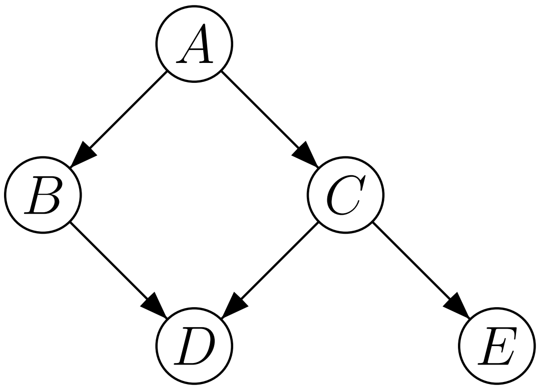
Choice of universe of discourse

	Variable	Domain
A	metastatic cancer	$\{a_1, a_2\}$
B	increased serum calcium	$\{b_1, b_2\}$
C	brain tumor	$\{c_1, c_2\}$
D	coma	$\{d_1, d_2\}$
E	headache	$\{e_1, e_2\}$

$(\cdot_1$ — present, $\cdot_2$ — absent$)$

$$\Omega = \{a_1, a_2\} \times \cdots \times \{e_1, e_2\}$$
$$|\Omega| = 32$$

Analysis of dependencies



Example

Choice of probability parameters

$$P(a,b,c,d,e) \overset{\text{abbr.}}{=} P(A=a, B=b, C=c, D=d, E=e)$$
$$= P(e \mid c)P(d \mid b,c)P(c \mid a)P(b \mid a)P(a)$$

Shorthand notation

- 11 values to store instead of 31
- Consult experts, textbooks, case studies, surveys, etc.

Calculation of conditional probabilities

Calculation of marginal probabilities

Crux of the Matter

- Knowledge acquisition (Where do the numbers come from?)
  $\rightarrow$ learning strategies

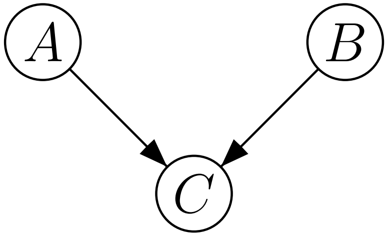
- Computational complexities
  $\rightarrow$ exploit independencies

Problem:

- When does the independency of X and Y given Z hold in (V, E, P)?

- How can we determine $P(X, Y \mid Z) = P(X \mid Z)P(Y \mid Z)$ solely using the graph structure?

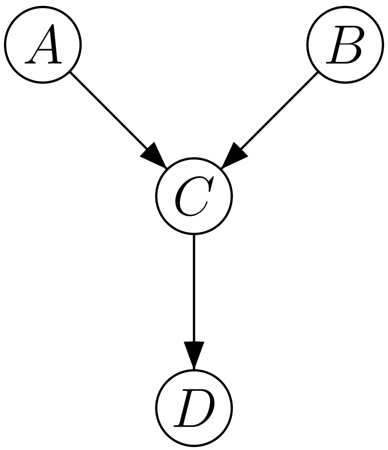
Converging Connection



Meal quality	
A	quality of ingredients
B	cook's skill
C	meal quality

- If C is not instantiated (i. e., no value specified/observed), A and B are marginally independent.
- After instantiation (observation) of C the variables A and B become conditionally dependent given C.
- Evidence can only be transferred over a converging connection if the variable in between (or one of its successors) is initialized.

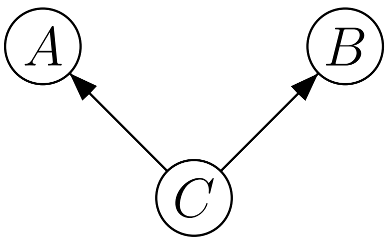
Converging Connection (cont.)



Meal quality	
A	quality of ingredients
B	cook's skill
C	meal quality
D	restaurant success

- If nothing is known about the restaurant success or meal quality or both, the cook's skills and quality of the ingredients are unrelated, thas is, *independent*.
- However, if we observe that the restaurant has no success, we can infer that the meal quality might be bad.
- If we further learn that the ingredients quality is high, we will conclude that the cook's skills must be low, thus rendering both variables *dependent*.

Diverging Connection

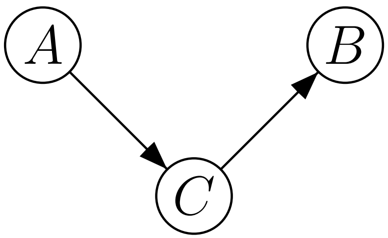


Diagnosis	
A	body temperature
B	cough
C	disease

- If C is unknown, knowledge about A ist relevant for B and vice versa, i. e. A and B are marginally dependent.
- However, if C is observed, A and B become conditionally independent given C.
- A influences B via C. If C is known it in a way blocks the information from flowing from A to B, thus rendering A and B (conditionally) independent.

Dependencies

Serial Connection

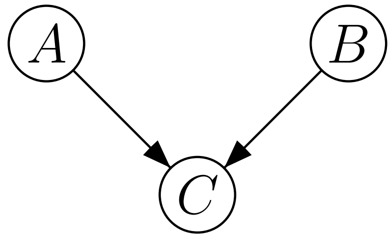


Accidents	
A	rain
B	accident risk
C	road conditions

- Analog scenario to case 2
- A influences C and C influences B. Thus, A influences B.
  If C is known, it blocks the path between A and B.

Formal Representation

Converging Connection: Marginal Independence



- Decomposition according to graph:

$$P(A,B,C) = P(C \mid A,B) \cdot P(A) \cdot P(B)$$

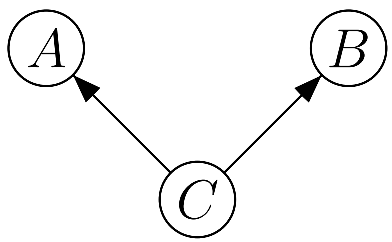
- Embedded Independence:

$$P(A,B,C) = \frac{P(A,B,C)}{P(A,B)} \cdot P(A) \cdot P(B) \qquad \text{with} \quad P(A,B) \neq 0$$

$$P(A,B) = P(A) \cdot P(B)$$

$$\Rightarrow \; A \perp\!\!\!\perp B \mid \emptyset$$

Formal Representation



Diverging Connection: Conditional Independence

- Decomposition according to graph:

$$P(A, B, C) = P(A \mid C) \cdot P(B \mid C) \cdot P(C)$$

- Embedded Independence:

$$P(A, B \mid C) = P(A \mid C) \cdot P(B \mid C)$$

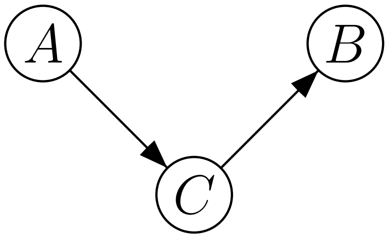
$$\Rightarrow A \perp\!\!\!\perp B \mid C$$

- Alternative derivation:

$$P(A, B, C) = P(A \mid C) \cdot P(B, C)$$

$$P(A \mid B, C) = P(A \mid C)$$

$$\Rightarrow A \perp\!\!\!\perp B \mid C$$



Serial Connection: Conditional Independence

- Decomposition according to graph:

$$P(A, B, C) \;=\; P(B \mid C) \cdot P(C \mid A) \cdot P(A)$$

- Embedded Independence:

$$\begin{aligned} P(A, B, C) \;&=\; P(B \mid C) \cdot P(C, A) \\ P(B \mid C, A) \;&=\; P(B \mid C) \\ &\Rightarrow\; A \perp\!\!\!\perp B \mid C \end{aligned}$$

Trivial Cases:

- Marginal Independence:

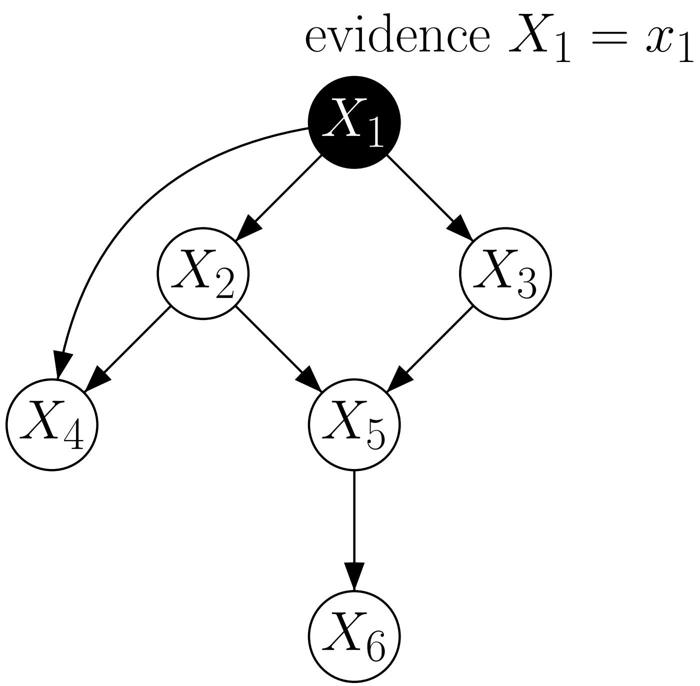
$$A \qquad\qquad B \qquad\qquad P(A, B) \;=\; P(A) \cdot P(B)$$

- Marginal Dependence:

$$A \longrightarrow B \qquad\qquad P(A, B) \;=\; P(B \mid A) \cdot P(A)$$

Question

Question: Are X_2 and X_3 independent given X_1?



d-Separation

Let $G = (V, E)$ a DAG and $X, Y, Z \in V$ three nodes.

a) A set $S \subseteq V\backslash\{X, Y\}$ *d-separates* X and Y, if S blocks all paths between X and Y. (paths may also route in opposite edge direction)

b) A path π is d-separated by S if at least one pair of consecutive edges along π is blocked. There are the following blocking conditions:

1.	$X \leftarrow Y \rightarrow Z$	tail-to-tail
2.	$X \leftarrow Y \leftarrow Z$ $X \rightarrow Y \rightarrow Z$	head-to-tail
3.	$X \rightarrow Y \leftarrow Z$	head-to-head

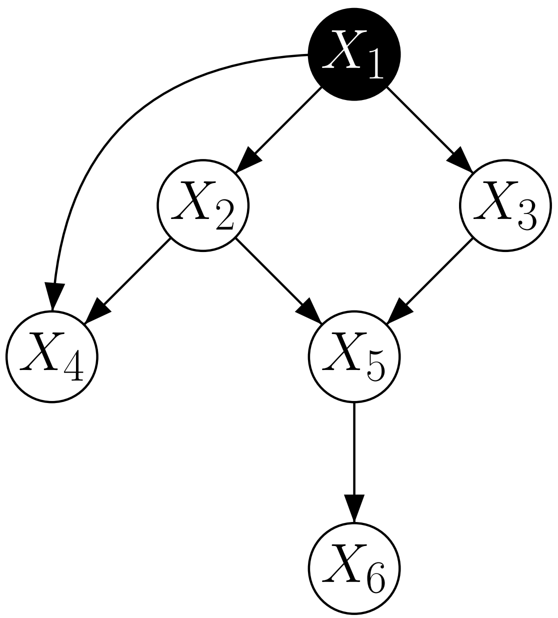
c) Two edges that meet tail-to-tail or head-to-tail in node Y are blocked if $Y \in S$.

d) Two edges meeting head-to-head in Y are blocked if neither Y nor its successors are in S.

If $S \subseteq V \backslash \{X, Y\}$ d-separates X and Y in a Belief network (V, E, P) then X and Y are conditionally independent given S:

$$P(X, Y \mid S) = P(X \mid S) \cdot P(Y \mid S)$$

Application to the previous example:



Paths: $\pi_1 = \langle X_2 - X_1 - X_3 \rangle,\ \pi_2 = \langle X_2 - X_5 - X_3 \rangle$
$\pi_3 = \langle X_2 - X_4 - X_1 - X_3 \rangle,\ S = \{X_1\}$

π_1 $X_2 \leftarrow X_1 \rightarrow X_3$ tail-to-tail
$X_1 \in S \Rightarrow \pi_1$ is blocked by S

π_2 $X_2 \rightarrow X_5 \leftarrow X_3$ head-to-head
$X_5, X_6 \notin S \Rightarrow \pi_2$ is blocked by S

π_3 $X_4 \leftarrow X_1 \rightarrow X_3$ tail-to-tail
$X_2 \rightarrow X_4 \leftarrow X_1$ head-to-head
both connections are blocked $\Rightarrow \pi_3$ is blocked

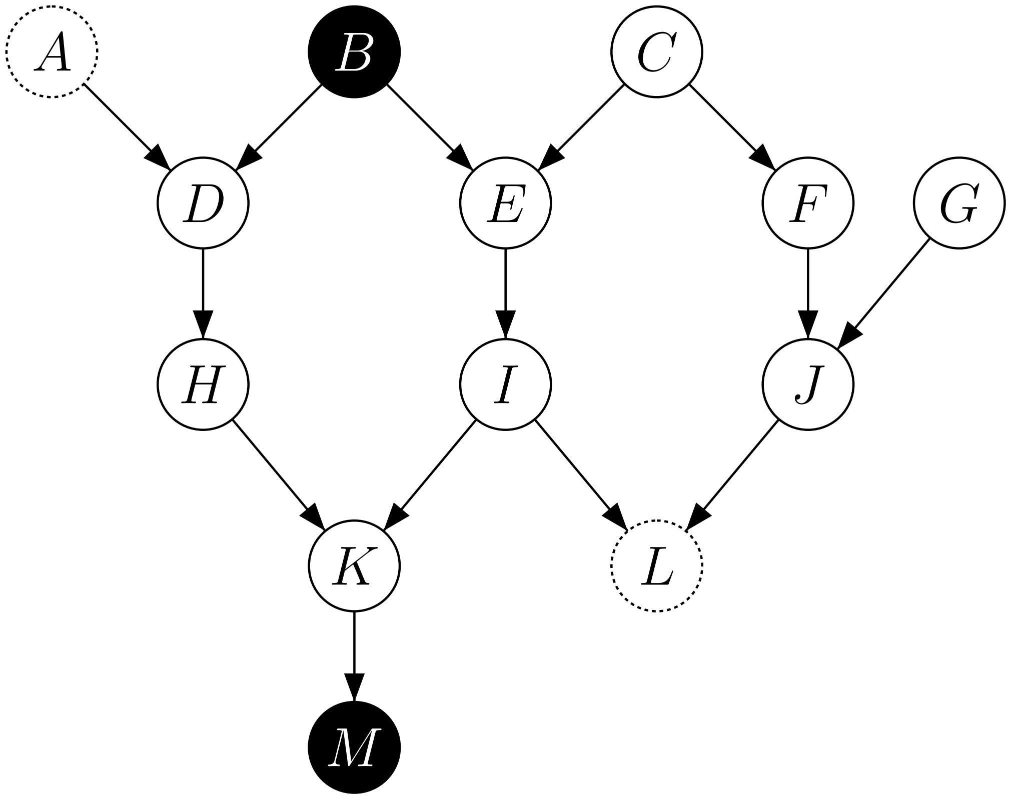
Example (cont.)

- Answer: X_2 and X_3 are d-separated via $\{X_1\}$. Therefore X_2 and X_3 become conditionally independent given X_1.

$$S = \{X_1, X_4\} \quad \Rightarrow \quad X_2 \text{ and } X_3 \text{ are d-separated by } S$$

$$S = \{X_1, X_6\} \quad \Rightarrow \quad X_2 \text{ and } X_3 \text{ are } \textit{not} \text{ d-separated by } S$$

Another Example



Are A and L conditionally independent given $\{B, M\}$?

Algebraic structure of CI statements

Question: Is it possible to use a formal scheme to infer new conditional independence (CI) statements from a set of initial CIs?

Repetition

Let $(\Omega, \mathcal{E}, P)$ be a probability space and W, X, Y, Z disjoint subsets of variables. If X and Y are conditionally independent given Z we write:

$$X \perp\!\!\!\perp_P Y \mid Z$$

Often, the following (equivalent) notation is used:

$$I_P(X \mid Z \mid Y) \quad \text{or} \quad I_P(X, Y \mid Z)$$

If the underlying space is known the index P is omitted.

(Semi-)Graphoid-Axioms

Let $(\Omega, \mathcal{E}, P)$ be a probability space and W, X, Y and Z four disjoint subsets of random variables (over Ω). Then the propositions

a) Symmetry: $(X \perp\!\!\!\perp_P Y \mid Z) \;\Rightarrow\; (Y \perp\!\!\!\perp_P X \mid Z)$

b) Decomposition: $(W \cup X \perp\!\!\!\perp_P Y \mid Z) \;\Rightarrow\; (W \perp\!\!\!\perp_P Y \mid Z) \wedge (X \perp\!\!\!\perp_P Y \mid Z)$

c) Weak Union: $(W \cup X \perp\!\!\!\perp_P Y \mid Z) \;\Rightarrow\; (X \perp\!\!\!\perp_P Y \mid Z \cup W)$

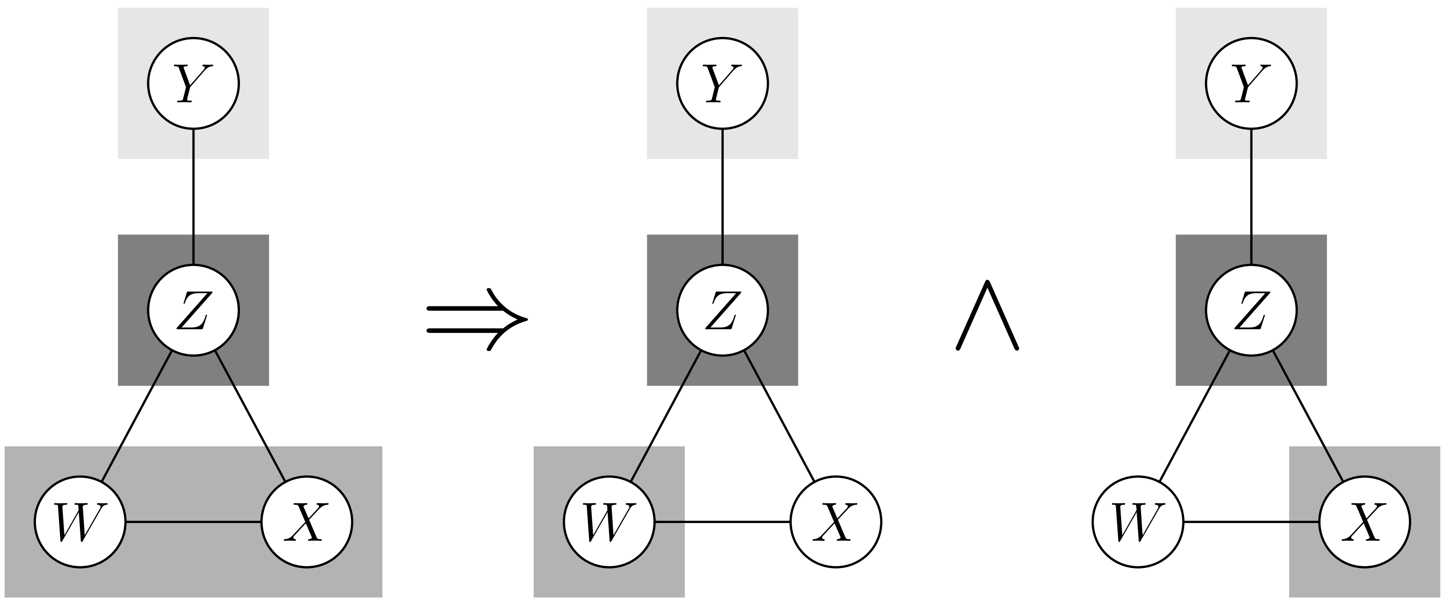
d) Contraction: $(X \perp\!\!\!\perp_P Y \mid Z \cup W) \wedge (W \perp\!\!\!\perp_P Y \mid Z) \;\Rightarrow\; (W \cup X \perp\!\!\!\perp_P Y \mid Z)$

are called the **Semi-Graphoid Axioms**. The above propositions and

e) Intersection: $(W \perp\!\!\!\perp_P Y \mid Z \cup X) \wedge (X \perp\!\!\!\perp_P Y \mid Z \cup W) \;\Rightarrow\; (W \cup X \perp\!\!\!\perp_P Y \mid Z)$

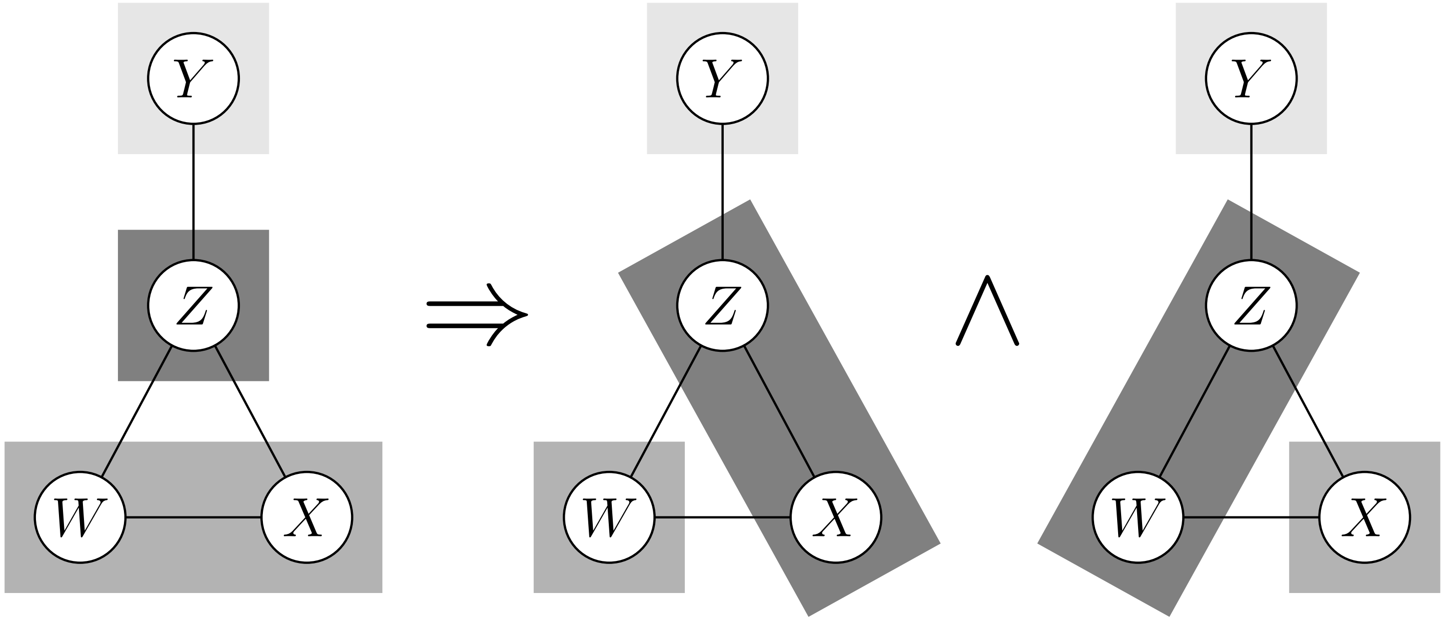
are called the **Graphoid Axioms**.

Decomposition



Drawings adapted from [Castillo *et al.* 1997].

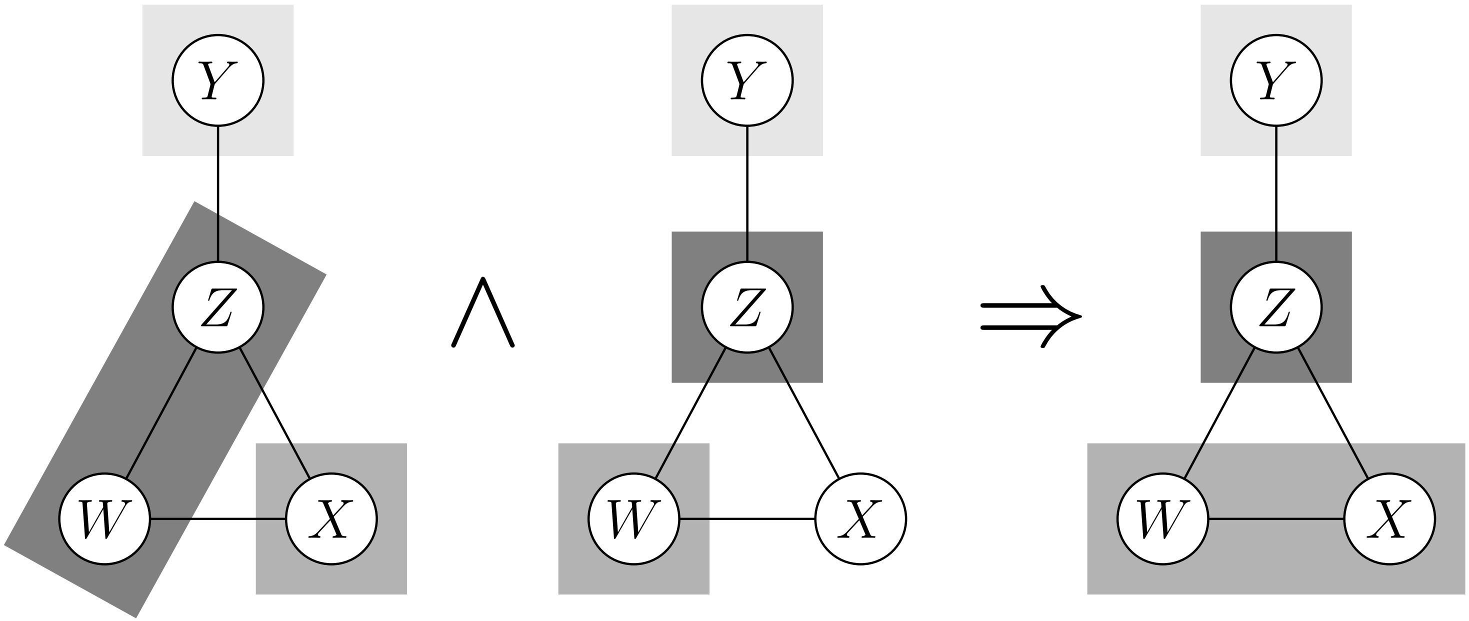
Weak Union



Learning irrelevant information W cannot render irrelevant information X relevant.

Drawings adapted from [Castillo *et al.* 1997].

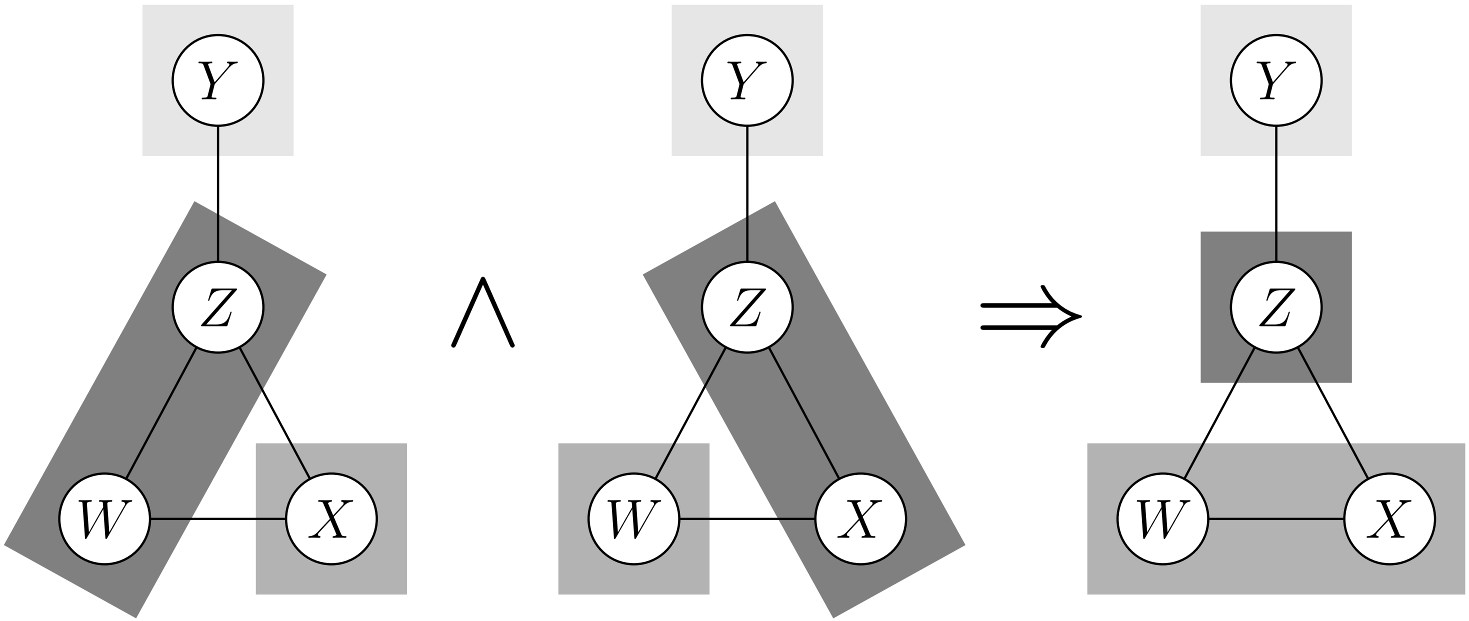
Contraction



If X is irrelevant (to Y) after having learnt some irrelevant information W, then X must have been irrelevant before.

Drawings adapted from [Castillo *et al.* 1997].

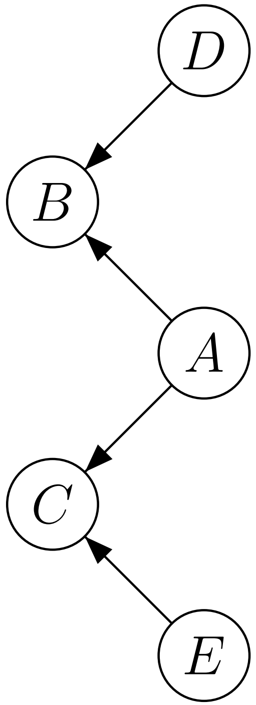
Intersection



Unless W affects Y when X is known or X affects Y when W is known, neither X nor W nor their combination can affect Y.

Drawings adapted from [Castillo *et al.* 1997].

Example



Proposition: $B \perp\!\!\!\perp C \mid A$

$$
\begin{aligned}
\text{Proof:}\quad & D \perp\!\!\!\perp A, C \mid \emptyset \quad \wedge \quad B \perp\!\!\!\perp C \mid A, D \\
\overset{\text{w. union}}{\Longrightarrow}\quad & D \perp\!\!\!\perp C \mid A \quad \wedge \quad B \perp\!\!\!\perp C \mid A, D \\
\overset{\text{symm.}}{\Longleftrightarrow}\quad & C \perp\!\!\!\perp D \mid A \quad \wedge \quad C \perp\!\!\!\perp B \mid A, D \\
\overset{\text{contr.}}{\Longrightarrow}\quad & C \perp\!\!\!\perp B, D \mid A \\
\overset{\text{decomp.}}{\Longrightarrow}\quad & C \perp\!\!\!\perp B \mid A \\
\overset{\text{symm.}}{\Longleftrightarrow}\quad & B \perp\!\!\!\perp C \mid A
\end{aligned}
$$

Conditional (In)Dependence Graphs

Definition: Let $(\cdot \perp\!\!\!\perp_\delta \cdot \mid \cdot)$ be a three-place relation representing the set of conditional independence statements that hold in a given distribution δ over a set U of attributes. An undirected graph $G = (U, E)$ over U is called a **conditional dependence graph** or a **dependence map** w. r. t. δ, iff for all disjoint subsets $X, Y, Z \subseteq U$ of attributes

$$X \perp\!\!\!\perp_\delta Y \mid Z \;\Rightarrow\; \langle X \mid Z \mid Y \rangle_G,$$

i. e., if G captures by u-separation all (conditional) independences that hold in δ and thus represents only valid (conditional) dependences. Similarly, G is called a **conditional independence graph** or an **independence map** w. r. t. δ, iff for all disjoint subsets $X, Y, Z \subseteq U$ of attributes

$$\langle X \mid Z \mid Y \rangle_G \;\Rightarrow\; X \perp\!\!\!\perp_\delta Y \mid Z,$$

i. e., if G captures by u-separation only (conditional) independences that are valid in δ. G is said to be a **perfect map** of the conditional (in)dependences in δ, if it is both a dependence map and an independence map.

Markov Properties of Undirected Graphs

Definition: An undirected graph $G = (U, E)$ over a set U of attributes is said to have (w.r.t. a distribution δ) the

pairwise Markov property,

iff in δ any pair of attributes which are nonadjacent in the graph are conditionally independent given all remaining attributes, i.e., iff

$$\forall A, B \in U, A \neq B : \quad (A, B) \notin E \;\Rightarrow\; A \perp\!\!\!\perp_\delta B \mid U - \{A, B\},$$

local Markov property,

iff in δ any attribute is conditionally independent of all remaining attributes given its neighbors, i.e., iff

$$\forall A \in U : \quad A \perp\!\!\!\perp_\delta U - \mathrm{closure}(A) \mid \mathrm{boundary}(A),$$

global Markov property,

iff in δ any two sets of attributes which are u-separated by a third are conditionally independent given the attributes in the third set, i.e., iff

$$\forall X, Y, Z \subseteq U : \quad \langle X \mid Z \mid Y \rangle_G \;\Rightarrow\; X \perp\!\!\!\perp_\delta Y \mid Z.$$

Markov Properties of Directed Acyclic Graphs

Definition: A directed acyclic graph $\vec{G} = (U, \vec{E})$ over a set U of attributes is said to have (w.r.t. a distribution δ) the

pairwise Markov property,

iff in δ any attribute is conditionally independent of any non-descendant not among its parents given all remaining non-descendants, i.e., iff

$$\forall A, B \in U: \; B \in \text{non-descs}(A) - \text{parents}(A) \;\Rightarrow\; A \perp\!\!\!\perp_\delta B \mid \text{non-descs}(A) - \{B\},$$

local Markov property,

iff in δ any attribute is conditionally independent of all remaining non-descendants given its parents, i.e., iff

$$\forall A \in U: \quad A \perp\!\!\!\perp_\delta \text{non-descs}(A) - \text{parents}(A) \mid \text{parents}(A),$$

global Markov property,

iff in δ any two sets of attributes which are d-separated by a third are conditionally independent given the attributes in the third set, i.e., iff

$$\forall X, Y, Z \subseteq U: \quad \langle X \mid Z \mid Y \rangle_{\vec{G}} \;\Rightarrow\; X \perp\!\!\!\perp_\delta Y \mid Z.$$

Propagation in Belief Networks

Objective

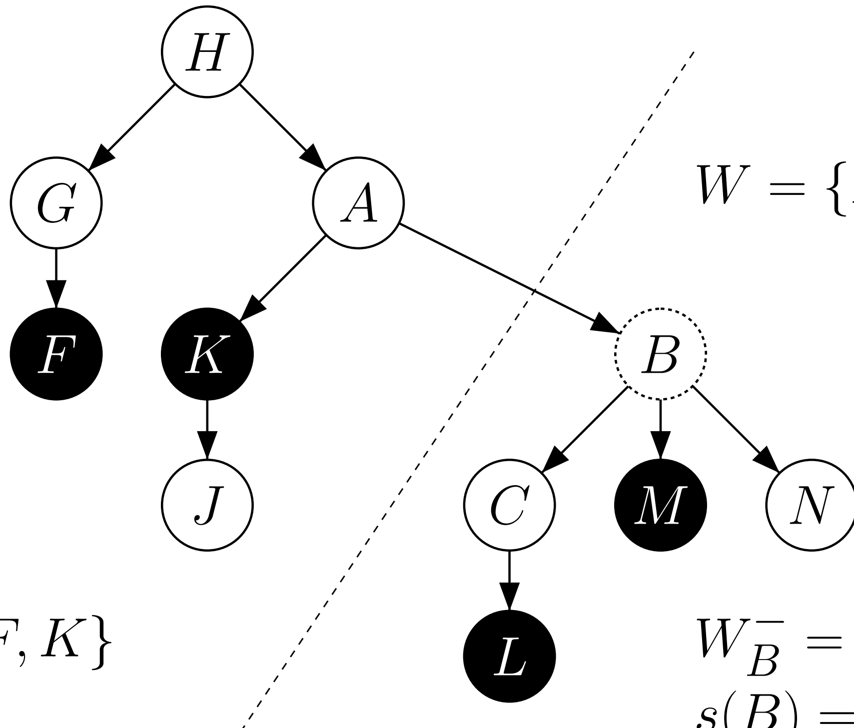
- **Given:** Belief network (V, E, P) with tree structure and $P(V) > 0$.
  Set $W \subseteq V$ of instantiated variables where
  a priori knowledge $W \neq \emptyset$ is allowed

- **Desired:** $P(B \mid W)$ for all $B \in V$

- **Notation:**

W_B^-	subset of those variables of W that belong to the subtree of (V, E) that has root B
W_B^+	$= W \backslash W_B^-$
$s(B)$	set of direct successors of B
Ω_B	domain of B
b^*	value that B is instantiated with

Example



$W = \{F, K, L, M\}$

$W_B^+ = \{F, K\}$

$W_B^- = \{L, M\}$

$s(B) = \{C, M, N\}$

Decomposition in the Tree

$$
\begin{aligned}
P(B = b \mid W) &= P(b \mid W_B^- \cup W_B^+) \qquad \text{with } B \notin W \\
&= \frac{P(W_B^- \cup W_B^+ \cup \{b\})}{P(W_B^- \cup W_B^+)} \\
&= \frac{P(W_B^- \cup W_B^+ \mid b) P(b)}{P(W_B^- \cup W_B^+)} \\
&= \frac{P(W_B^- \mid b) P(W_B^+ \mid b) P(b)}{P(W_B^- \cup W_B^+)} \\
&= \beta_{B,W} \quad \underbrace{P(W_B^- \mid b)}_{\text{Evidence from ``below''}} \quad \underbrace{P(b \mid W_B^+)}_{\text{Evidence from ``above''}}
\end{aligned}
$$

Since we ignore the constant $\beta_{B,W}$ for the derivations below, the following designations are used instead of $P(\cdot)$:

π-values and λ-values

Let $B \in V$ be a variable and $b \in \Omega_B$ a value of its domain. We define the π- and λ-values as follows:

$$\lambda(b) = \begin{cases} P(W_B^- \mid b) & \text{if } B \notin W \\ \quad 1 & \text{if } B \in W \wedge b^* = b \\ \quad 0 & \text{if } B \in W \wedge b^* \neq b \end{cases}$$

$$\pi(b) = P(b \mid W_B^+)$$

π- and λ-Values

$$\lambda(b) = \prod_{C \in s(B)} P(W_C^- \mid b) \qquad \text{if } B \in W$$

$$\lambda(b) = 1 \qquad \text{if } B \text{ leaf in } (V, E)$$

$$\pi(b) = P(b) \qquad \text{if } B \text{ root in } (V, E)$$

$$P(b \mid W) = \alpha_{B,W} \cdot \lambda(b) \cdot \pi(b)$$

λ-message

Let $B \in V$ be an attribute and $C \in s(B)$ its direct children with the respective domains $\mathrm{dom}(B) = \{B_1, \ldots, b_i, \ldots, b_k\}$ and $\mathrm{dom}(C) = \{c_1, \ldots, c_j, \ldots, c_m\}$.

$$\lambda_{C \to B}(b_i) \overset{\text{Def}}{=} \sum_{j=1}^{m} P(c_j \mid b_i) \cdot \lambda(c_j), \qquad i = 1, \ldots, k$$

The vector

$$\vec{\lambda}_{C \to B} \overset{\text{Def}}{=} \Big(\lambda_{C \to B}(b_i)\Big)_{i=1}^{k}$$

is called λ-*message* from C to B.

λ-Message

Let $B \in V$ an attribute an $b \in \mathrm{dom}(B)$ a value of its domain.
Then

$$\lambda(b) = \begin{cases} \rho_{B,W} \cdot \prod\limits_{C \in s(B)} \lambda_C(b) & \text{if } B \notin W \\ 1 & \text{if } B \in W \wedge b = b^* \\ 0 & \text{if } B \in W \wedge b \neq b^* \end{cases}$$

with $\rho_{B,W}$ being a positive constant.

π-message

Let $B \in V$ be a non-root node in (V, E) and $A \in V$ its parent with domain $\mathrm{dom}(A) = \{a_1, \ldots, a_j, \ldots, a_m\}$.

$$j = 1, \ldots, m:$$
$$\pi_{A \to B}(a_j) \stackrel{\text{Def}}{=} \begin{cases} \pi(a_j) \cdot \prod\limits_{C \in s(A) \setminus \{B\}} \lambda_C(a_j) & \text{if } A \notin W \\ 1 & \text{if } A \in W \wedge a = a^* \\ 0 & \text{if } A \in W \wedge a \neq a^* \end{cases}$$

The vector

$$\vec{\pi}_{A \to B} \stackrel{\text{Def}}{=} \Big(\pi_{A \to B}(a_j)\Big)_{j=1}^{m}$$

is called *π-message* from A to B.

π-Message

Let $B \in V$ be a non-root node in (V, E) and A the parent node of B.
Further let $b \in \mathrm{dom}(B)$ be a value of B's domain.

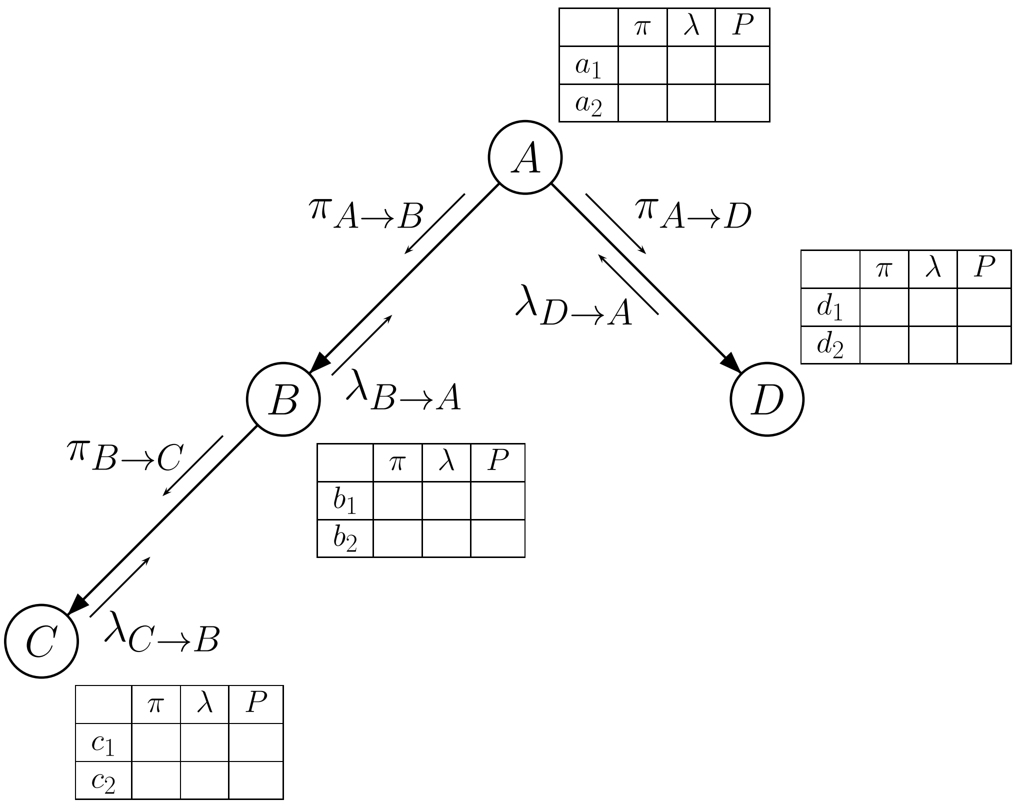
$$\pi(b) \;=\; \mu_{B,W} \;\cdot \sum_{a \in \mathrm{dom}(A)} P(b \mid a) \cdot \pi_{A \to B}(a)$$

Let $A \notin W$ a non-instantiated attribute and $P(V) > 0$.

$$\begin{aligned}
\pi_{A \to B}(a_j) \;&=\; \pi(a_j) \;\cdot \prod_{C \in s(A) \setminus \{B\}} \lambda_{C \to A}(a_j) \\
&=\; \tau_{B,W} \cdot \frac{P(a_j \mid W)}{\lambda_{B \to A}(a_j)}
\end{aligned}$$

Propagation in Belief Trees

Belief Tree:



Parameters:

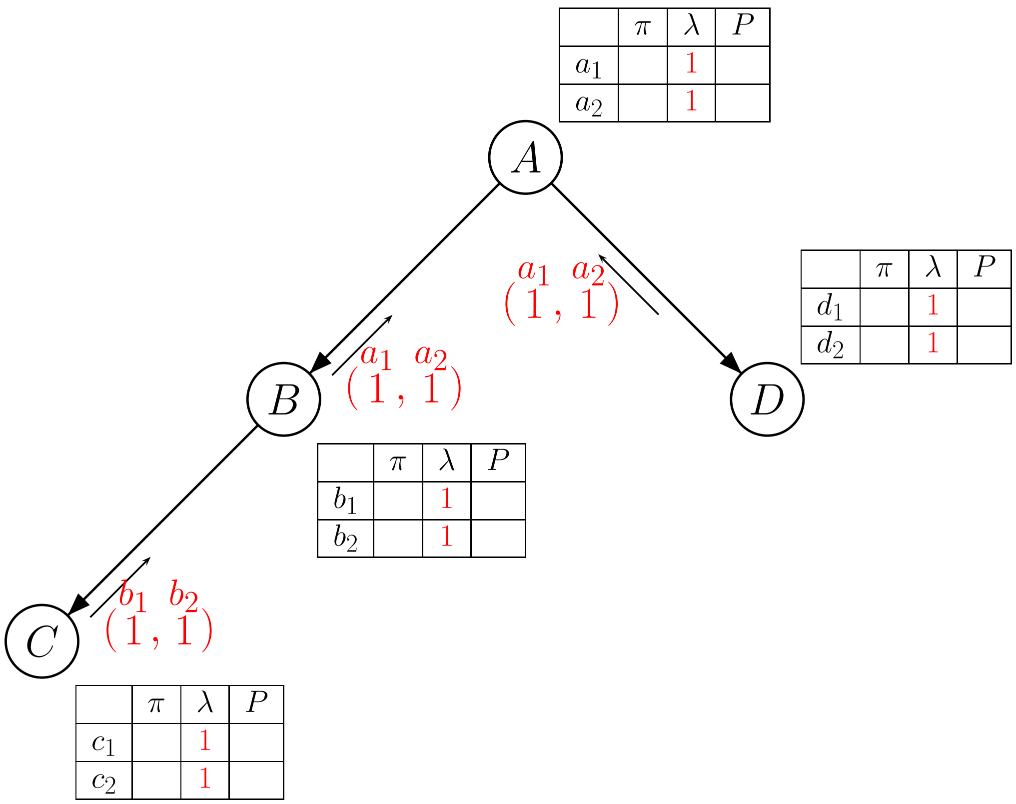
$$P(a_1) = 0.1 \quad P(b_1 \mid a_1) = 0.7$$
$$P(b_1 \mid a_2) = 0.2$$
$$P(d_1 \mid a_1) = 0.8 \quad P(c_1 \mid b_1) = 0.4$$
$$P(d_1 \mid a_2) = 0.4 \quad P(c_1 \mid b_2) = 0.001$$

Desired:

$$\forall X \in \{A, B, C, D\}: \ P(X \mid \emptyset) = ?$$

Propagation in Belief Trees (2)

Belief Tree:

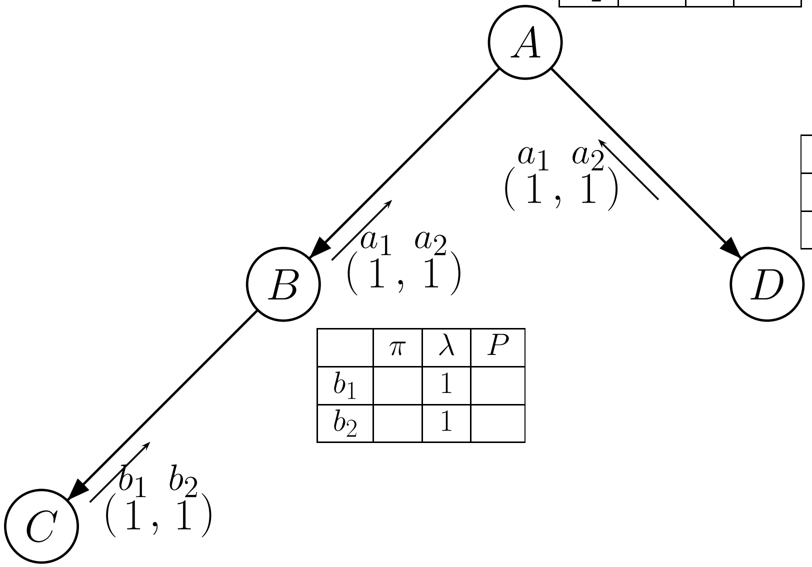


Initialization Phase:

- Set all λ-messages and λ-values to 1.

Propagation in Belief Trees (3)

Belief Tree:



	π	λ	P
a_1	0.1	1	0.1
a_2	0.9	1	0.9

	π	λ	P
d_1		1	
d_2		1	

	π	λ	P
b_1		1	
b_2		1	

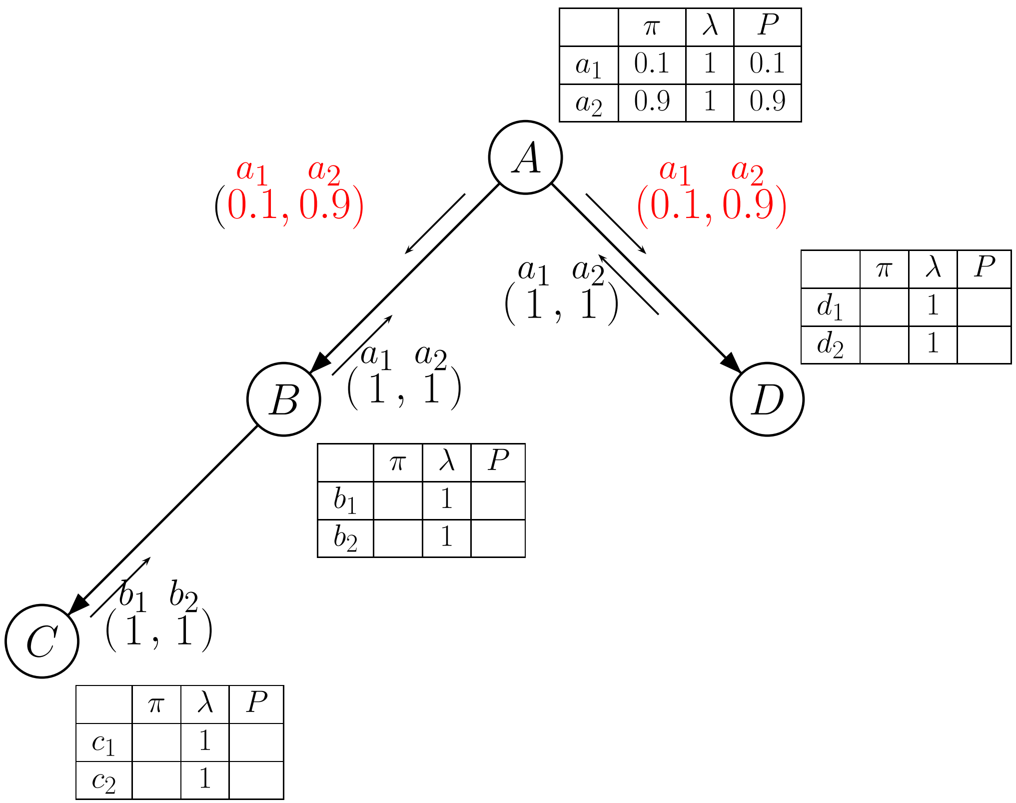
	π	λ	P
c_1		1	
c_2		1	

Initialization Phase:

- Set all λ-messages and λ-values to 1.
- $\pi(a_1) = P(a_1)$ and $\pi(a_2) = P(a_2)$

Propagation in Belief Trees (4)

Belief Tree:

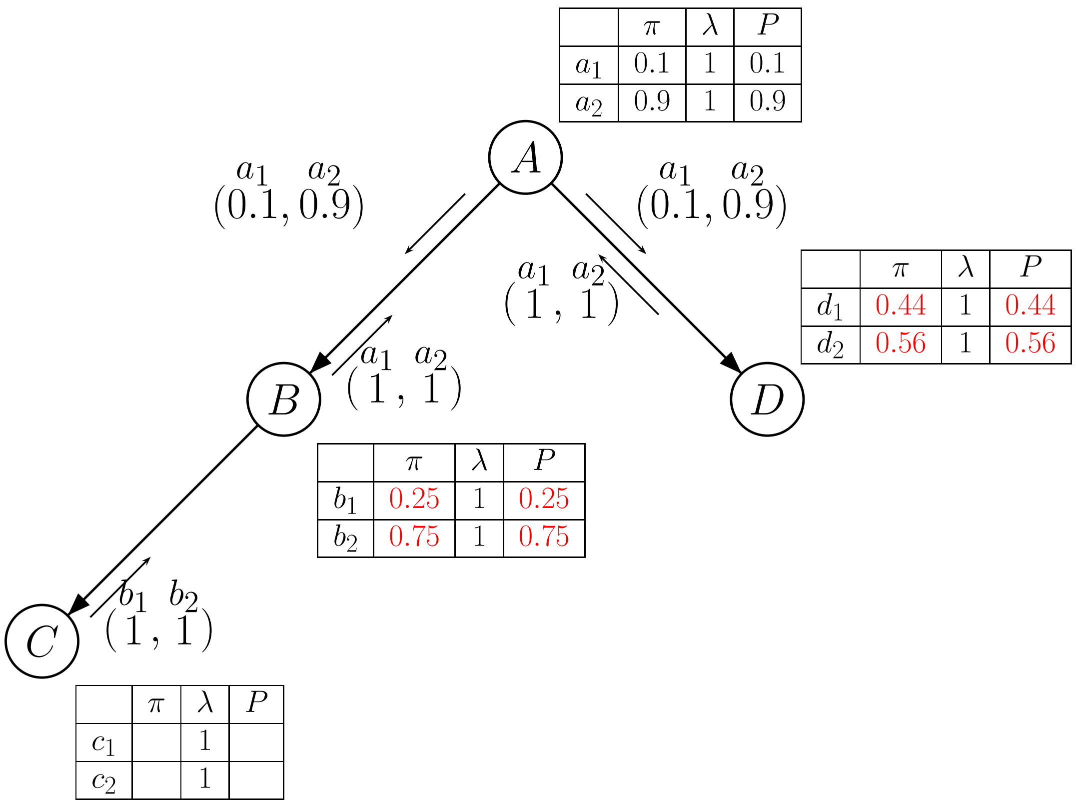


Initialization Phase:

- Set all λ-messages and λ-values to 1.
- $\pi(a_1) = P(a_1)$ and $\pi(a_2) = P(a_2)$.
- A sends π-messages to B and D.

Propagation in Belief Trees (5)

Belief Tree:



	π	λ	P
a_1	0.1	1	0.1
a_2	0.9	1	0.9

	π	λ	P
b_1	0.25	1	0.25
b_2	0.75	1	0.75

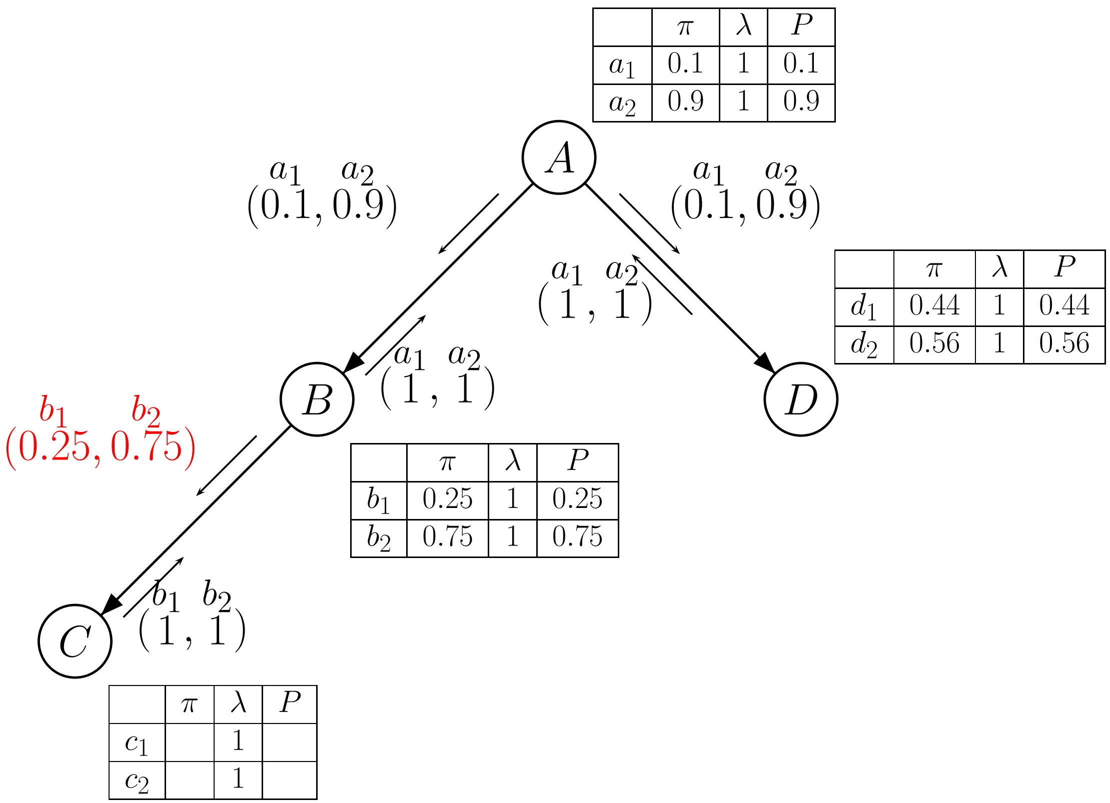
	π	λ	P
c_1		1	
c_2		1	

	π	λ	P
d_1	0.44	1	0.44
d_2	0.56	1	0.56

Initialization Phase:

- Set all λ-messages and λ-values to 1.
- $\pi(a_1) = P(a_1)$ and $\pi(a_2) = P(a_2)$.
- A sends π-messages to B and D.
- B and D update their π-values.

Belief Tree:

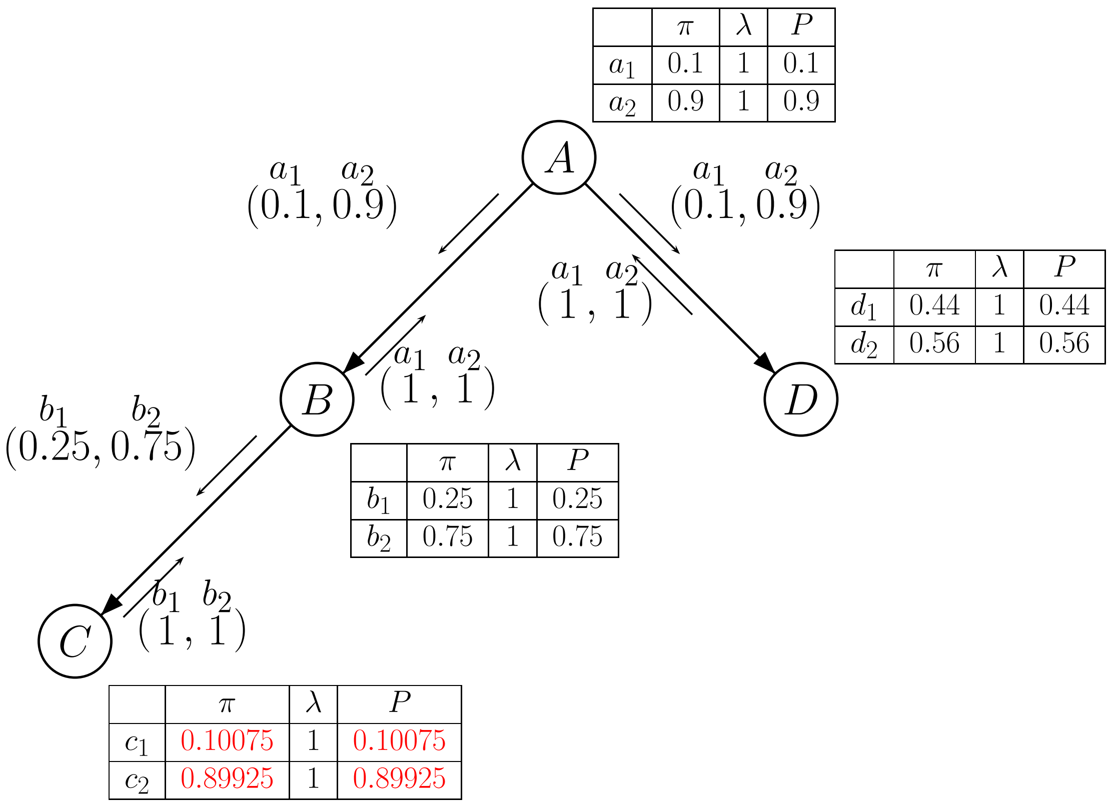


Initialization Phase:

- Set all λ-messages and λ-values to 1.
- $\pi(a_1) = P(a_1)$ and $\pi(a_2) = P(a_2)$.
- A sends π-messages to B and D.
- B and D update their π-values.
- B sends π-message to C.

Propagation in Belief Trees (7)

Belief Tree:



	π	λ	P
a_1	0.1	1	0.1
a_2	0.9	1	0.9

	π	λ	P
d_1	0.44	1	0.44
d_2	0.56	1	0.56

	π	λ	P
b_1	0.25	1	0.25
b_2	0.75	1	0.75

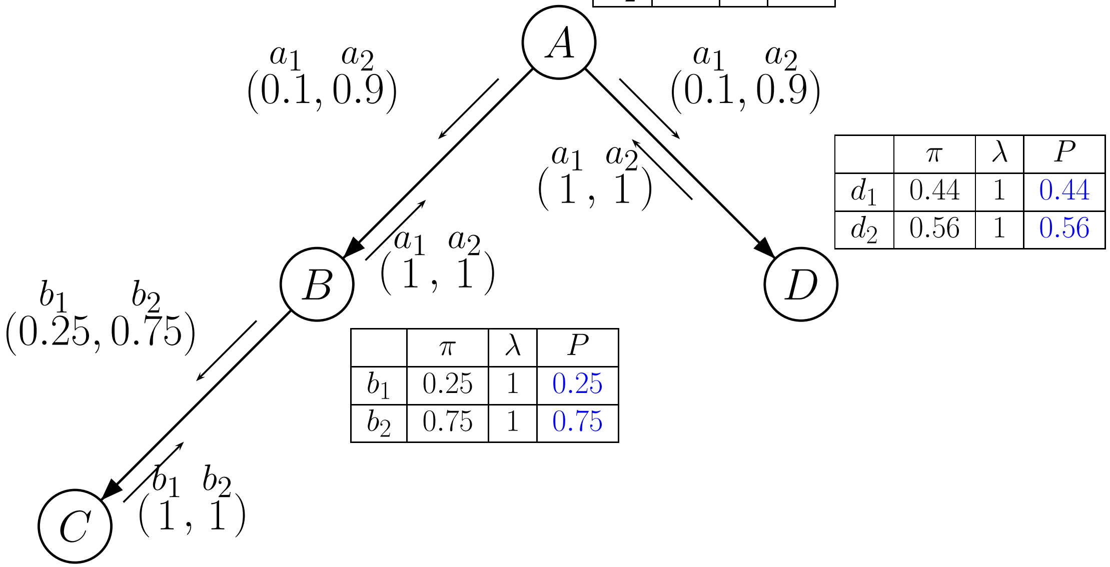
	π	λ	P
c_1	0.10075	1	0.10075
c_2	0.89925	1	0.89925

Initialization Phase:

- Set all λ-messages and λ-values to 1.
- $\pi(a_1) = P(a_1)$ and $\pi(a_2) = P(a_2)$.
- A sends π-messages to B and D.
- B and D update their π-values.
- B sends π-message to C.
- C updates it π-value.

Propagation in Belief Trees (8)

Belief Tree:



	π	λ	P
a_1	0.1	1	0.1
a_2	0.9	1	0.9

	π	λ	P
d_1	0.44	1	0.44
d_2	0.56	1	0.56

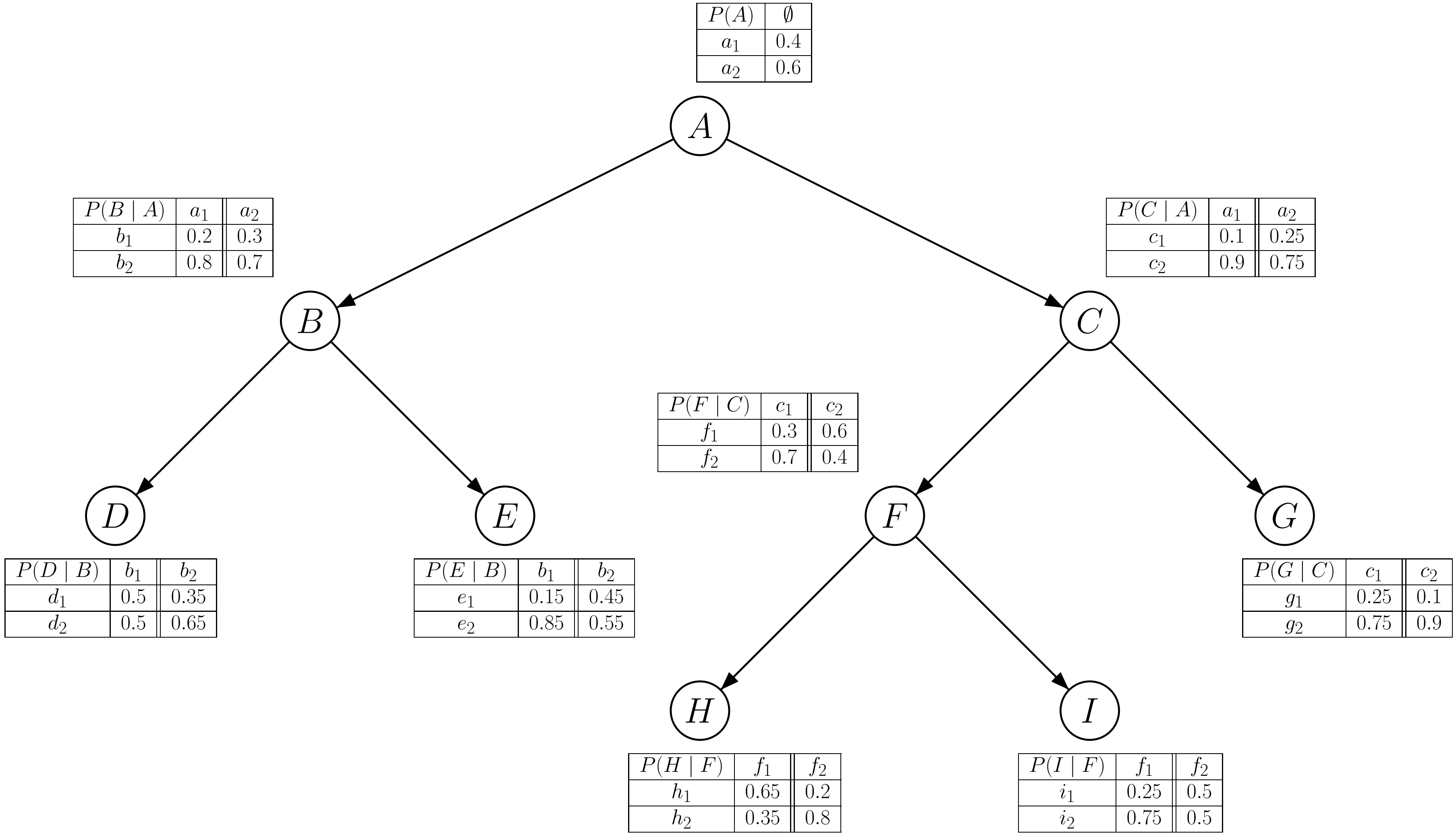
	π	λ	P
b_1	0.25	1	0.25
b_2	0.75	1	0.75

	π	λ	P
c_1	0.10075	1	0.10075
c_2	0.89925	1	0.89925

Initialization Phase:

- Set all λ-messages and λ-values to 1.
- $\pi(a_1) = P(a_1)$ and $\pi(a_2) = P(a_2)$.
- A sends π-messages to B and D.
- B and D update their π-values.
- B sends π-message to C.
- C updates it π-value.
- Initialization finished.

Larger Network (1): Parameters



$P(A)$	$\emptyset$
a_1	0.4
a_2	0.6

$P(B \mid A)$	a_1	a_2
b_1	0.2	0.3
b_2	0.8	0.7

$P(C \mid A)$	a_1	a_2
c_1	0.1	0.25
c_2	0.9	0.75

$P(F \mid C)$	c_1	c_2
f_1	0.3	0.6
f_2	0.7	0.4

$P(D \mid B)$	b_1	b_2
d_1	0.5	0.35
d_2	0.5	0.65

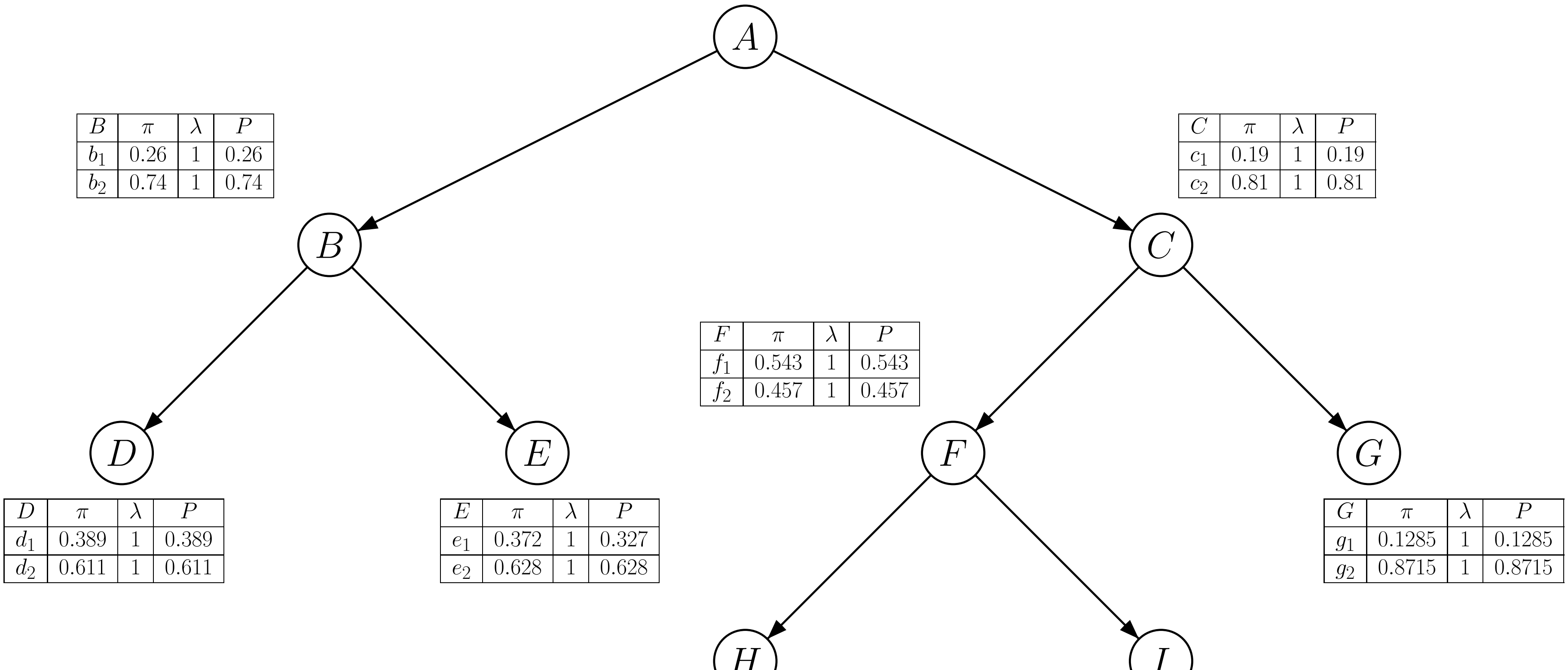
$P(E \mid B)$	b_1	b_2
e_1	0.15	0.45
e_2	0.85	0.55

$P(G \mid C)$	c_1	c_2
g_1	0.25	0.1
g_2	0.75	0.9

$P(H \mid F)$	f_1	f_2
h_1	0.65	0.2
h_2	0.35	0.8

$P(I \mid F)$	f_1	f_2
i_1	0.25	0.5
i_2	0.75	0.5

Larger Network (2): After Initialization



A	π	λ	P
a_1	0.4	1	0.4
a_2	0.6	1	0.6

B	π	λ	P
b_1	0.26	1	0.26
b_2	0.74	1	0.74

C	π	λ	P
c_1	0.19	1	0.19
c_2	0.81	1	0.81

D	π	λ	P
d_1	0.389	1	0.389
d_2	0.611	1	0.611

E	π	λ	P
e_1	0.372	1	0.327
e_2	0.628	1	0.628

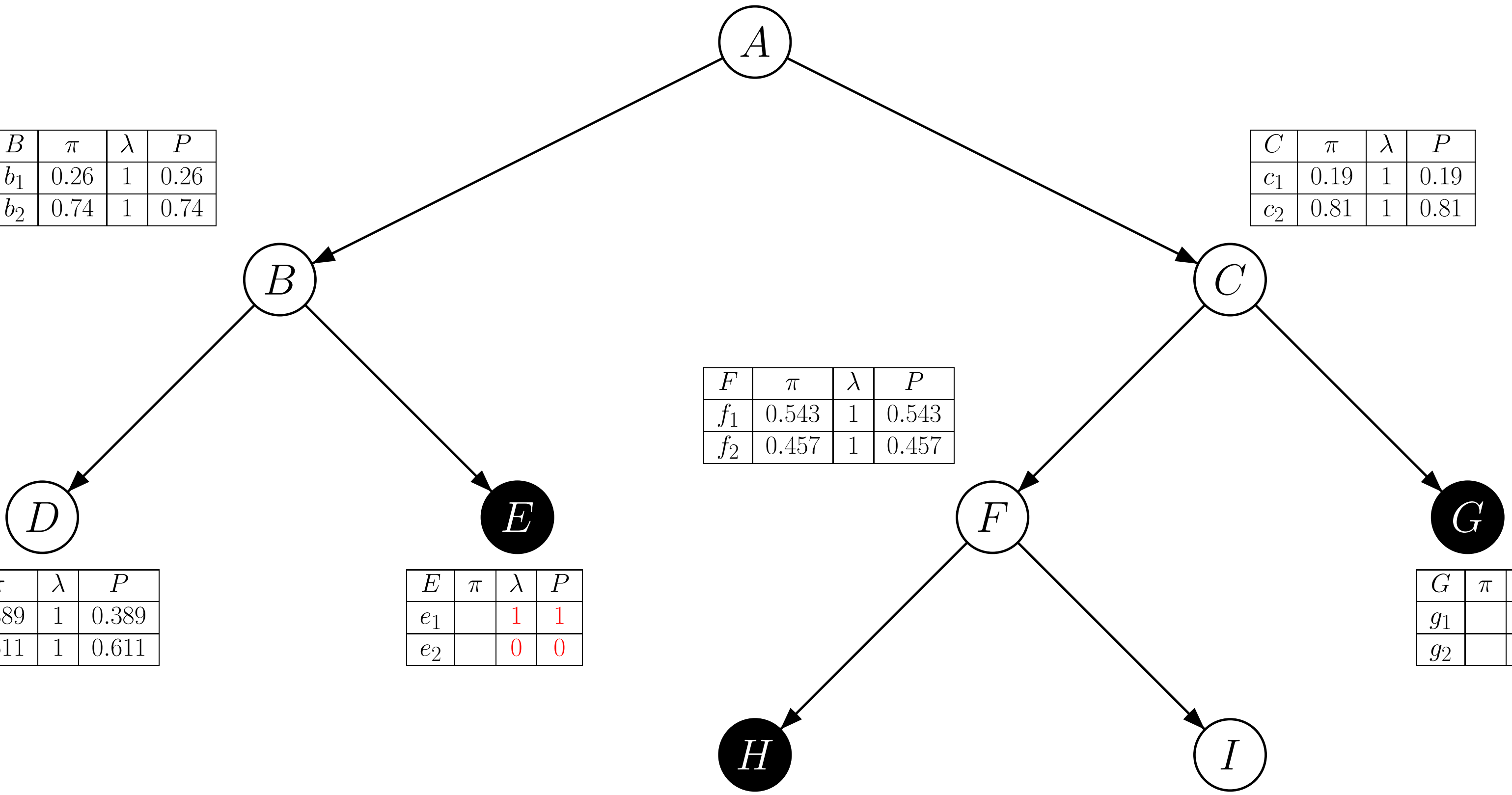
F	π	λ	P
f_1	0.543	1	0.543
f_2	0.457	1	0.457

G	π	λ	P
g_1	0.1285	1	0.1285
g_2	0.8715	1	0.8715

H	π	λ	P
h_1	0.4444	1	0.4444
h_2	0.5556	1	0.5556

I	π	λ	P
i_1	0.3643	1	0.3643
i_2	0.6357	1	0.6357

Larger Network (3): Set Evidence e_1, g_1, h_1



A	π	λ	P
a_1	0.4	1	0.4
a_2	0.6	1	0.6

B	π	λ	P
b_1	0.26	1	0.26
b_2	0.74	1	0.74

C	π	λ	P
c_1	0.19	1	0.19
c_2	0.81	1	0.81

F	π	λ	P
f_1	0.543	1	0.543
f_2	0.457	1	0.457

D	π	λ	P
d_1	0.389	1	0.389
d_2	0.611	1	0.611

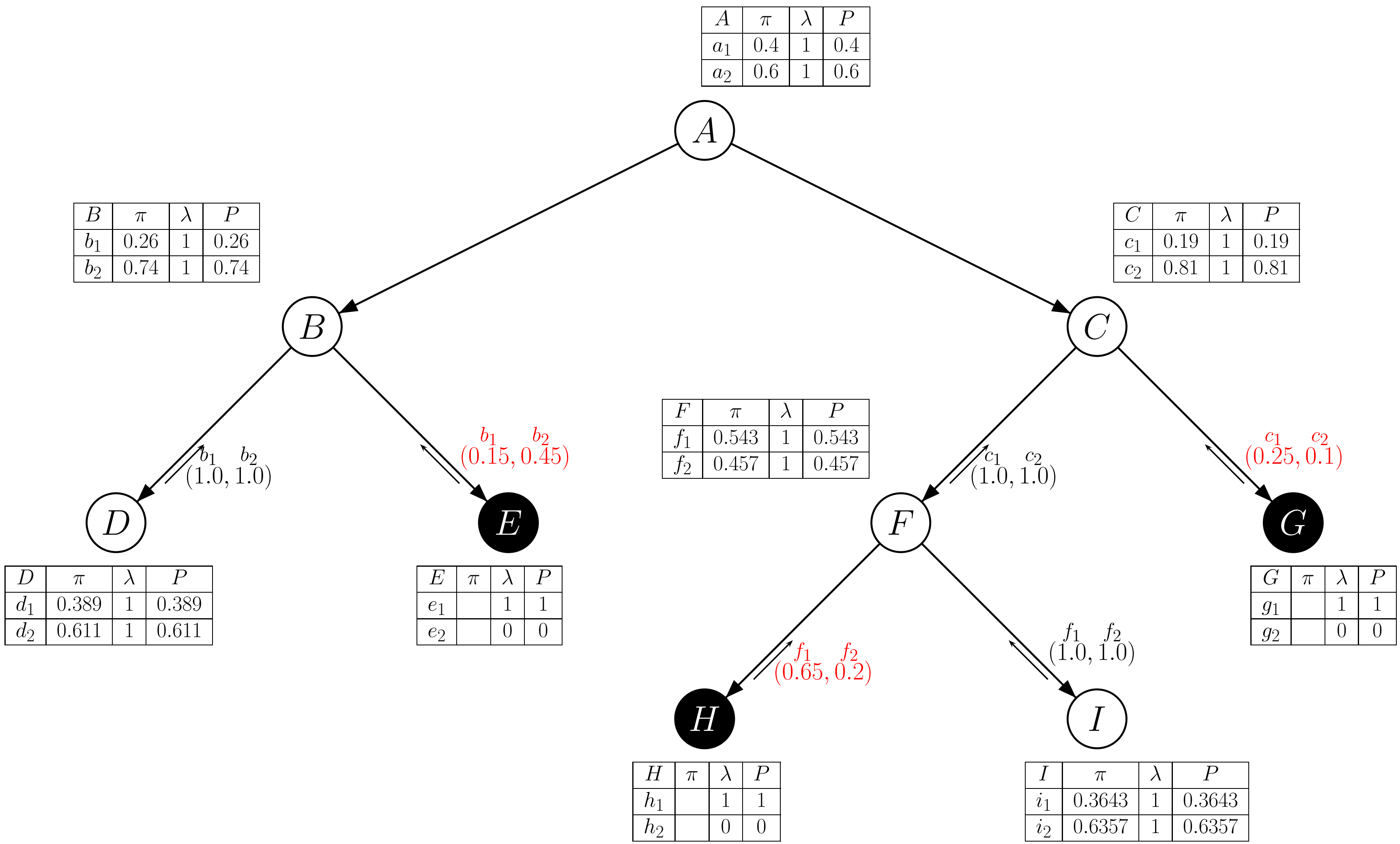
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3643	1	0.3643
i_2	0.6357	1	0.6357

Larger Network (4): Propagate Evidence



A	π	λ	P
a_1	0.4	1	0.4
a_2	0.6	1	0.6

B	π	λ	P
b_1	0.26	1	0.26
b_2	0.74	1	0.74

C	π	λ	P
c_1	0.19	1	0.19
c_2	0.81	1	0.81

F	π	λ	P
f_1	0.543	1	0.543
f_2	0.457	1	0.457

D	π	λ	P
d_1	0.389	1	0.389
d_2	0.611	1	0.611

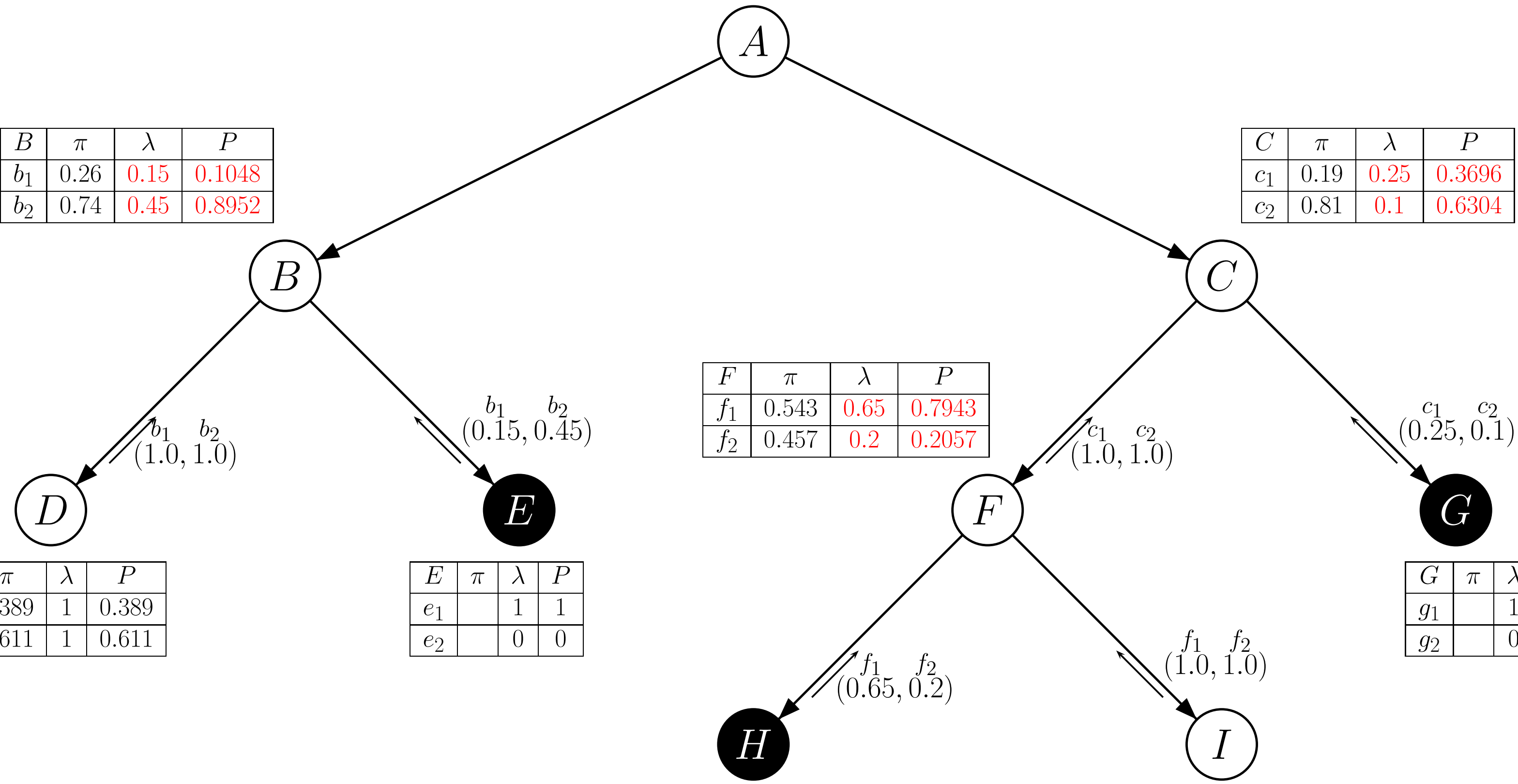
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3643	1	0.3643
i_2	0.6357	1	0.6357

Larger Network (5): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	1	0.4
a_2	0.6	1	0.6

B	π	λ	P
b_1	0.26	0.15	0.1048
b_2	0.74	0.45	0.8952

C	π	λ	P
c_1	0.19	0.25	0.3696
c_2	0.81	0.1	0.6304

F	π	λ	P
f_1	0.543	0.65	0.7943
f_2	0.457	0.2	0.2057

D	π	λ	P
d_1	0.389	1	0.389
d_2	0.611	1	0.611

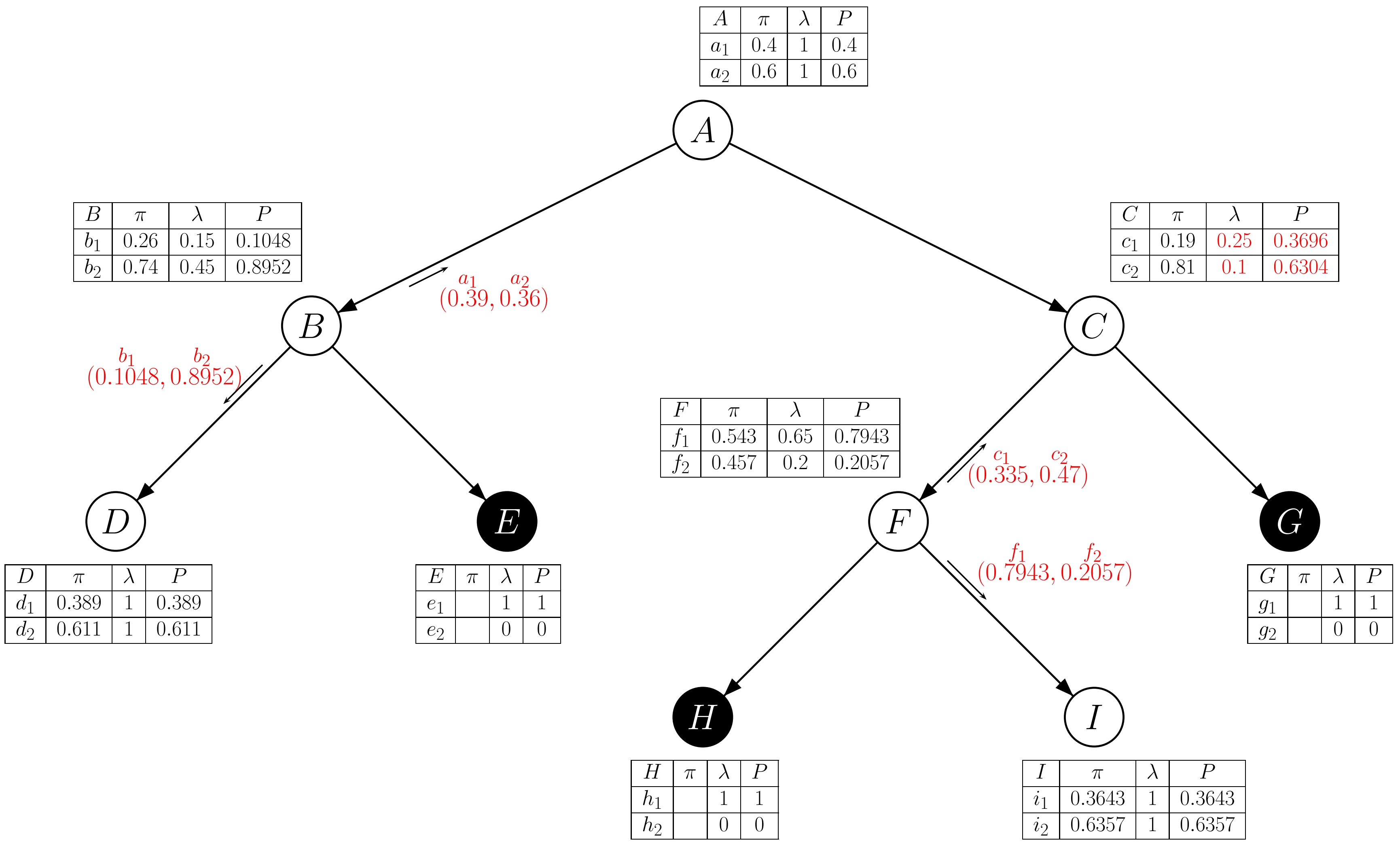
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3643	1	0.3643
i_2	0.6357	1	0.6357

Larger Network (6): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	1	0.4
a_2	0.6	1	0.6

B	π	λ	P
b_1	0.26	0.15	0.1048
b_2	0.74	0.45	0.8952

C	π	λ	P
c_1	0.19	0.25	0.3696
c_2	0.81	0.1	0.6304

F	π	λ	P
f_1	0.543	0.65	0.7943
f_2	0.457	0.2	0.2057

D	π	λ	P
d_1	0.389	1	0.389
d_2	0.611	1	0.611

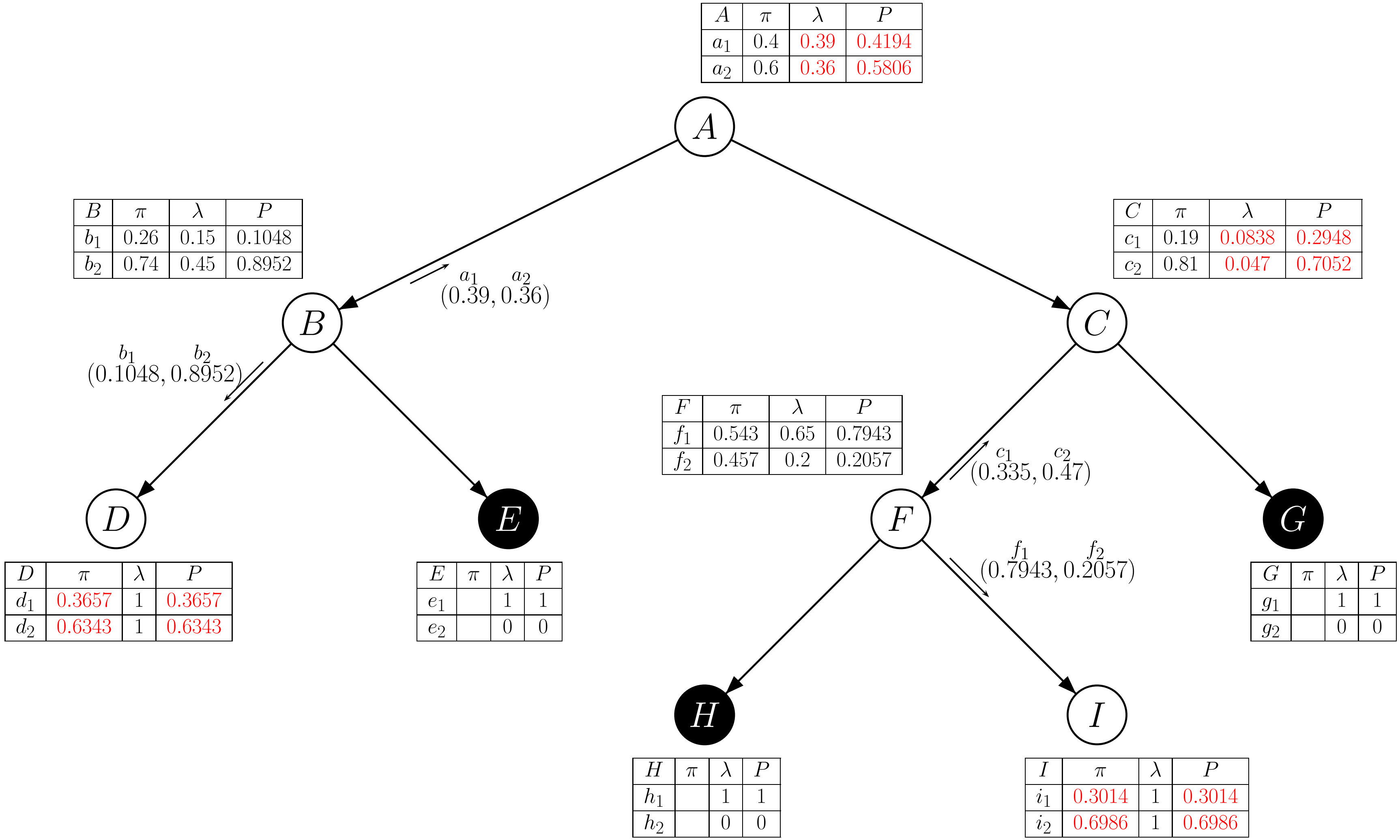
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3643	1	0.3643
i_2	0.6357	1	0.6357

Larger Network (7): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.39	0.4194
a_2	0.6	0.36	0.5806

B	π	λ	P
b_1	0.26	0.15	0.1048
b_2	0.74	0.45	0.8952

C	π	λ	P
c_1	0.19	0.0838	0.2948
c_2	0.81	0.047	0.7052

F	π	λ	P
f_1	0.543	0.65	0.7943
f_2	0.457	0.2	0.2057

D	π	λ	P
d_1	0.3657	1	0.3657
d_2	0.6343	1	0.6343

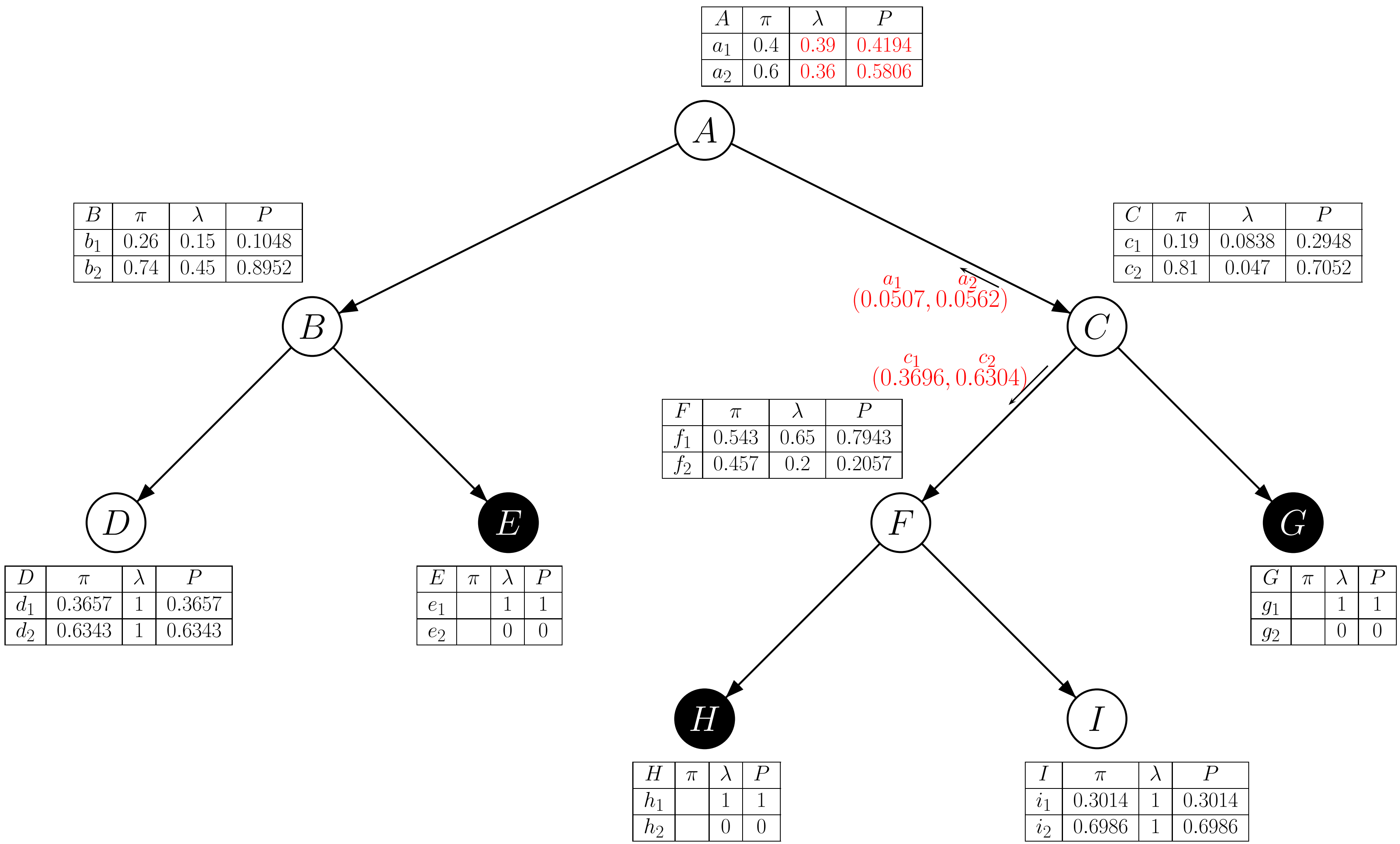
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3014	1	0.3014
i_2	0.6986	1	0.6986

Larger Network (8): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.39	0.4194
a_2	0.6	0.36	0.5806

B	π	λ	P
b_1	0.26	0.15	0.1048
b_2	0.74	0.45	0.8952

C	π	λ	P
c_1	0.19	0.0838	0.2948
c_2	0.81	0.047	0.7052

F	π	λ	P
f_1	0.543	0.65	0.7943
f_2	0.457	0.2	0.2057

D	π	λ	P
d_1	0.3657	1	0.3657
d_2	0.6343	1	0.6343

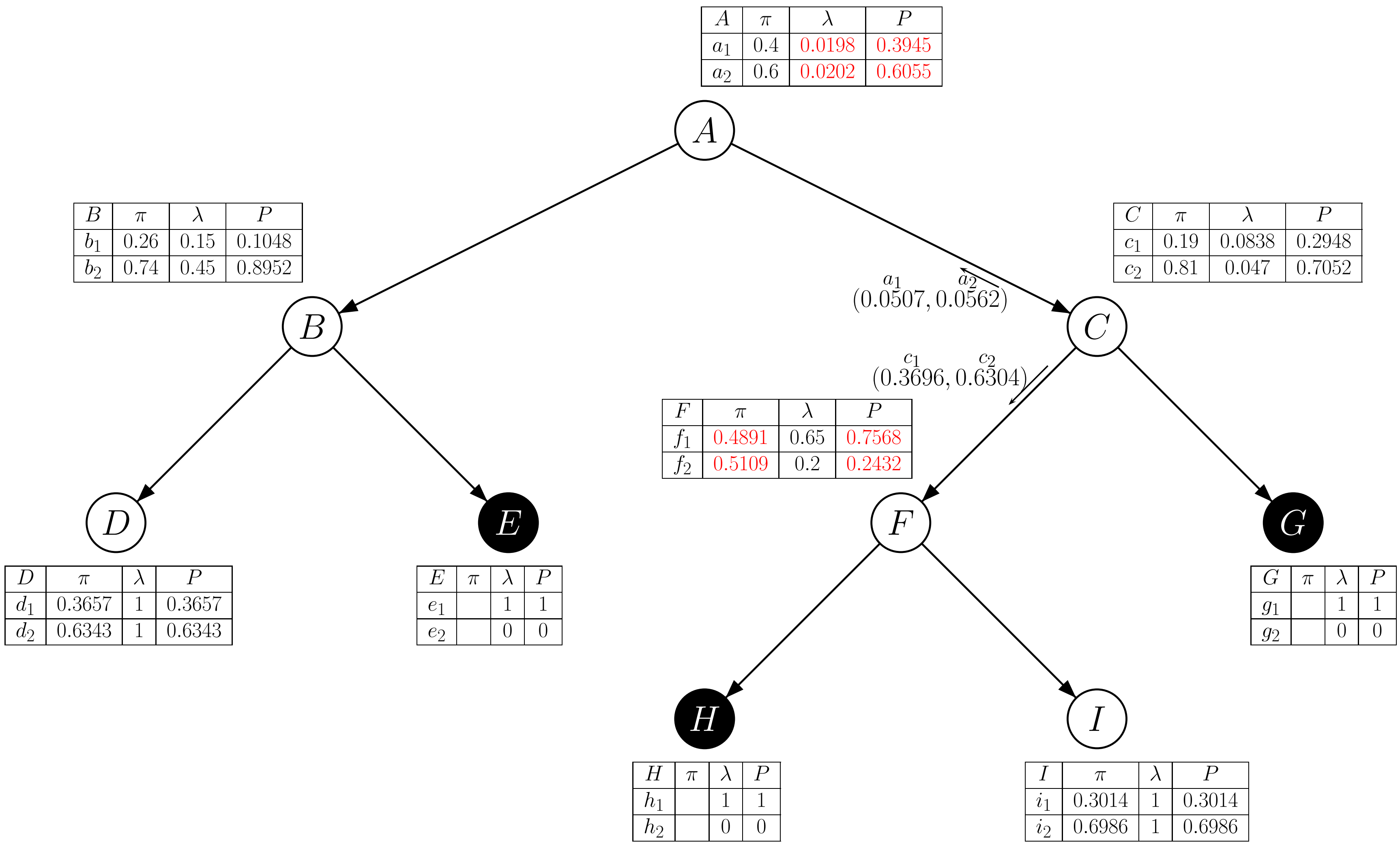
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3014	1	0.3014
i_2	0.6986	1	0.6986

Larger Network (9): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.26	0.15	0.1048
b_2	0.74	0.45	0.8952

C	π	λ	P
c_1	0.19	0.0838	0.2948
c_2	0.81	0.047	0.7052

F	π	λ	P
f_1	0.4891	0.65	0.7568
f_2	0.5109	0.2	0.2432

D	π	λ	P
d_1	0.3657	1	0.3657
d_2	0.6343	1	0.6343

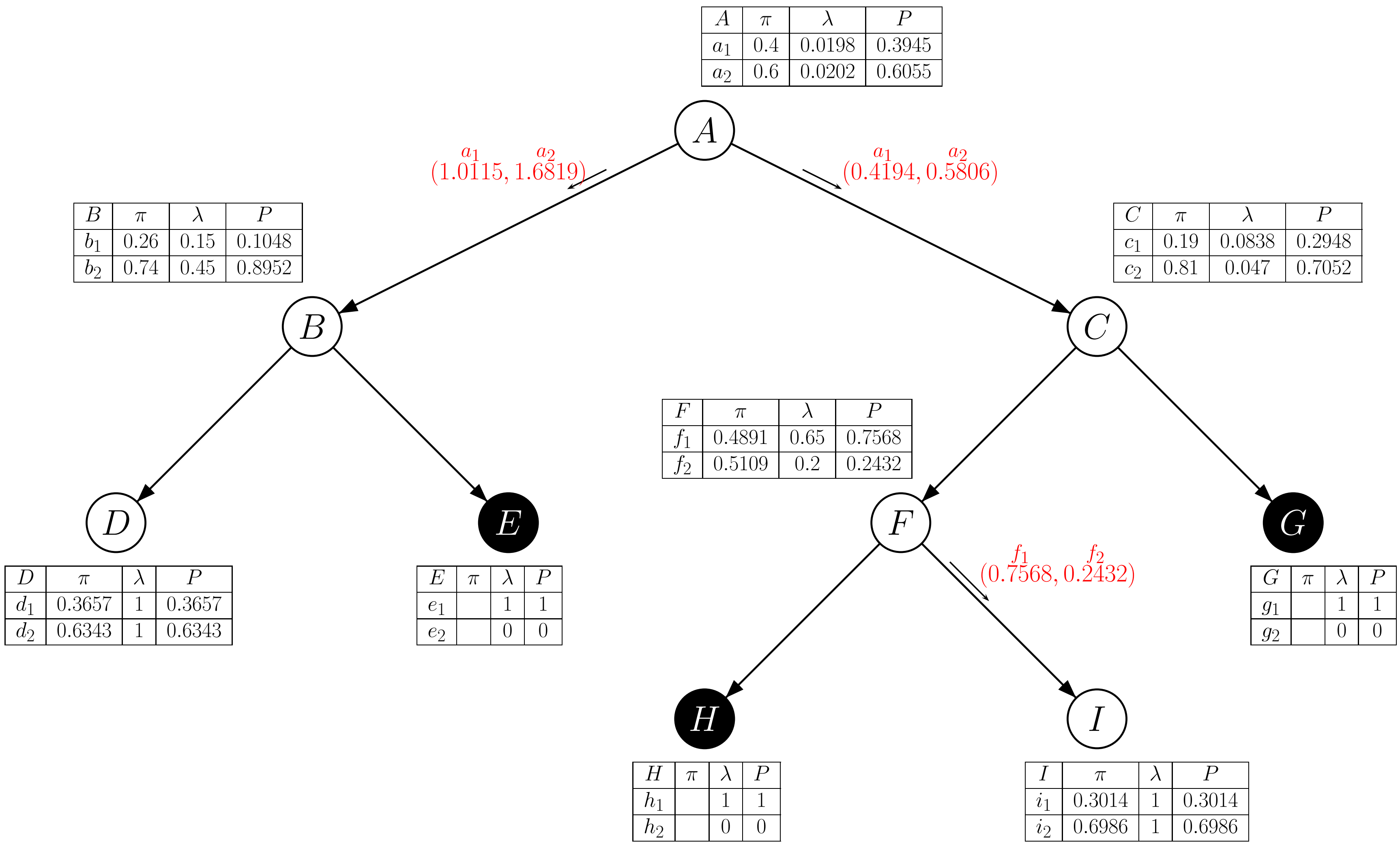
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3014	1	0.3014
i_2	0.6986	1	0.6986

Larger Network (10): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.26	0.15	0.1048
b_2	0.74	0.45	0.8952

C	π	λ	P
c_1	0.19	0.0838	0.2948
c_2	0.81	0.047	0.7052

F	π	λ	P
f_1	0.4891	0.65	0.7568
f_2	0.5109	0.2	0.2432

D	π	λ	P
d_1	0.3657	1	0.3657
d_2	0.6343	1	0.6343

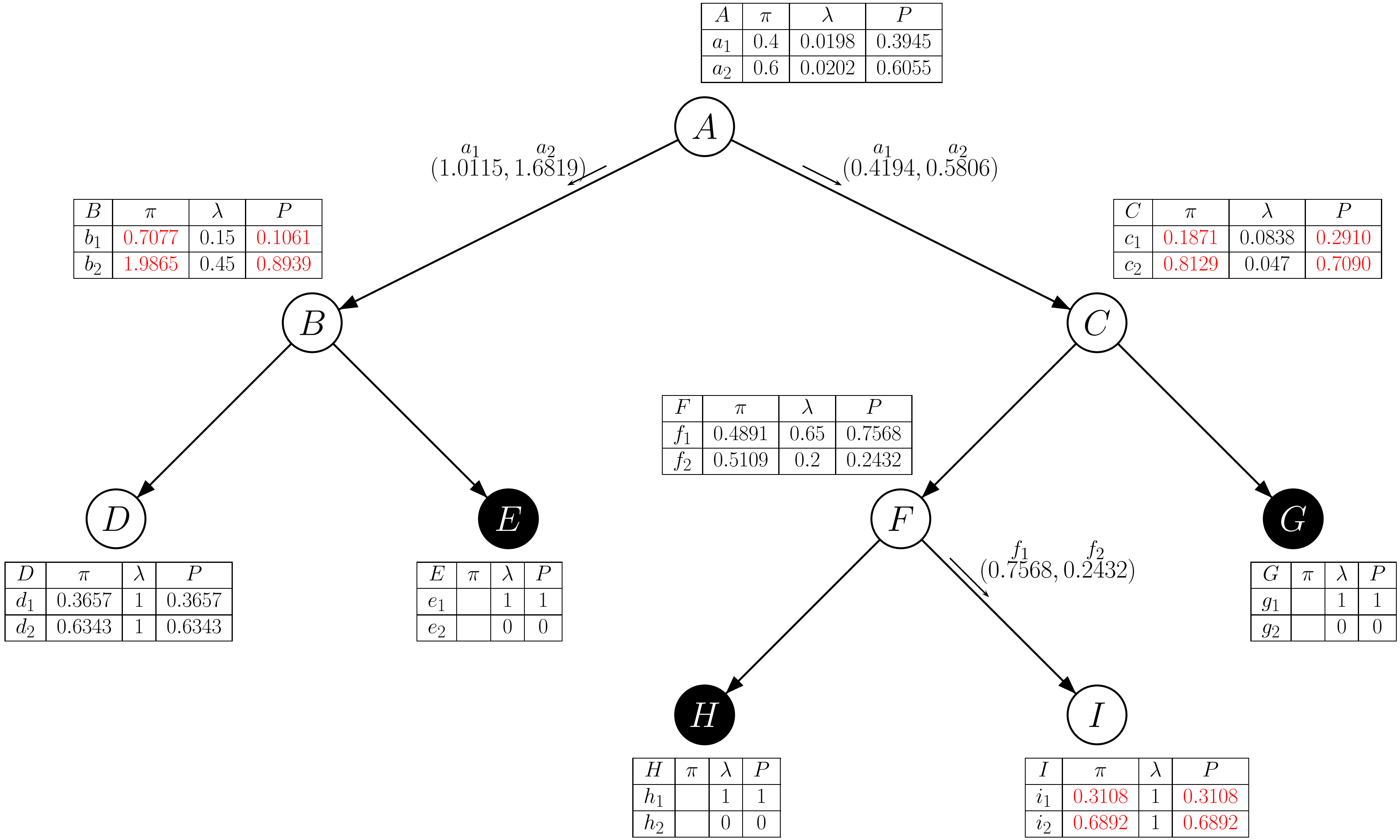
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3014	1	0.3014
i_2	0.6986	1	0.6986

Larger Network (11): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.7077	0.15	0.1061
b_2	1.9865	0.45	0.8939

C	π	λ	P
c_1	0.1871	0.0838	0.2910
c_2	0.8129	0.047	0.7090

F	π	λ	P
f_1	0.4891	0.65	0.7568
f_2	0.5109	0.2	0.2432

D	π	λ	P
d_1	0.3657	1	0.3657
d_2	0.6343	1	0.6343

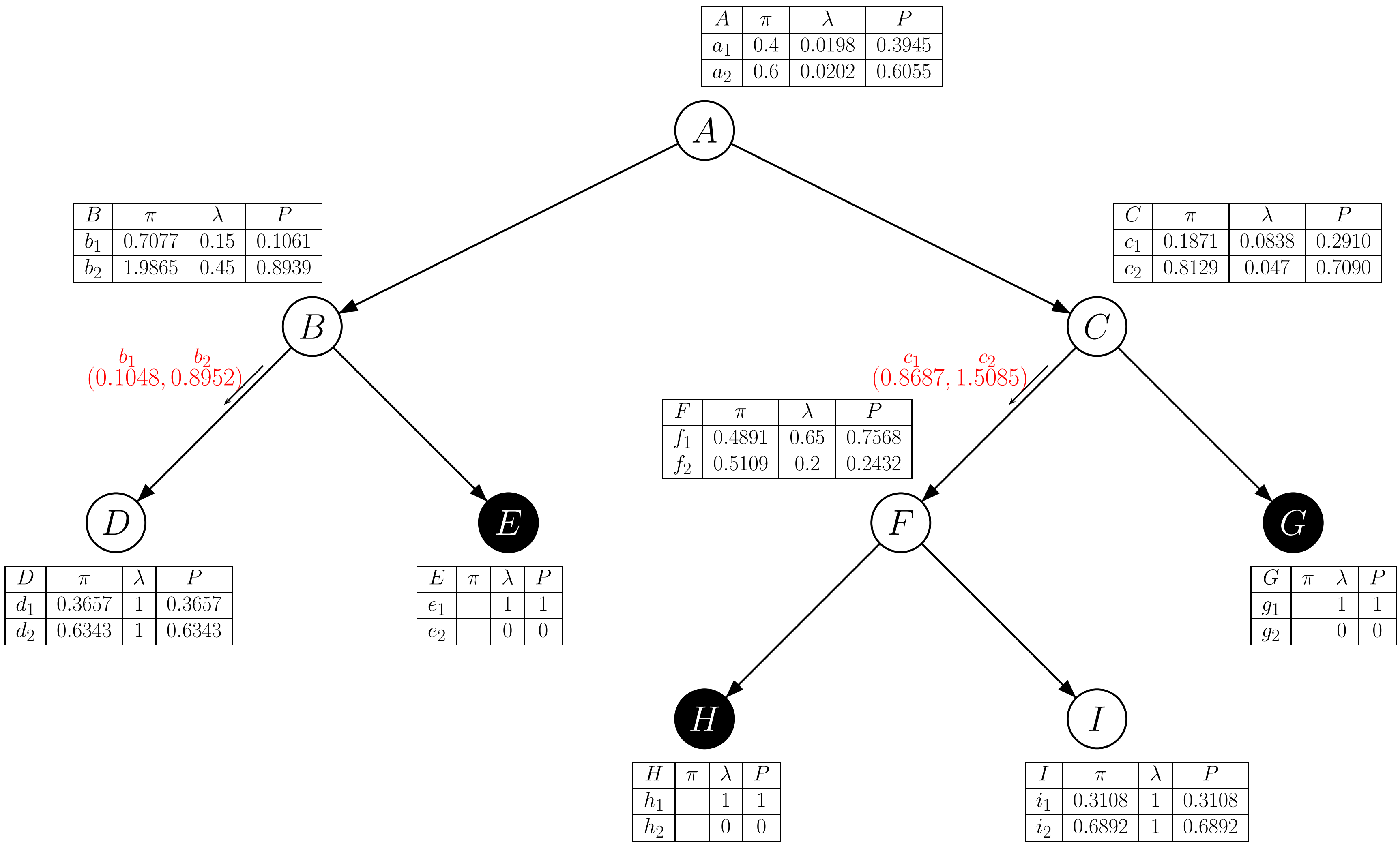
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3108	1	0.3108
i_2	0.6892	1	0.6892

Larger Network (12): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.7077	0.15	0.1061
b_2	1.9865	0.45	0.8939

C	π	λ	P
c_1	0.1871	0.0838	0.2910
c_2	0.8129	0.047	0.7090

F	π	λ	P
f_1	0.4891	0.65	0.7568
f_2	0.5109	0.2	0.2432

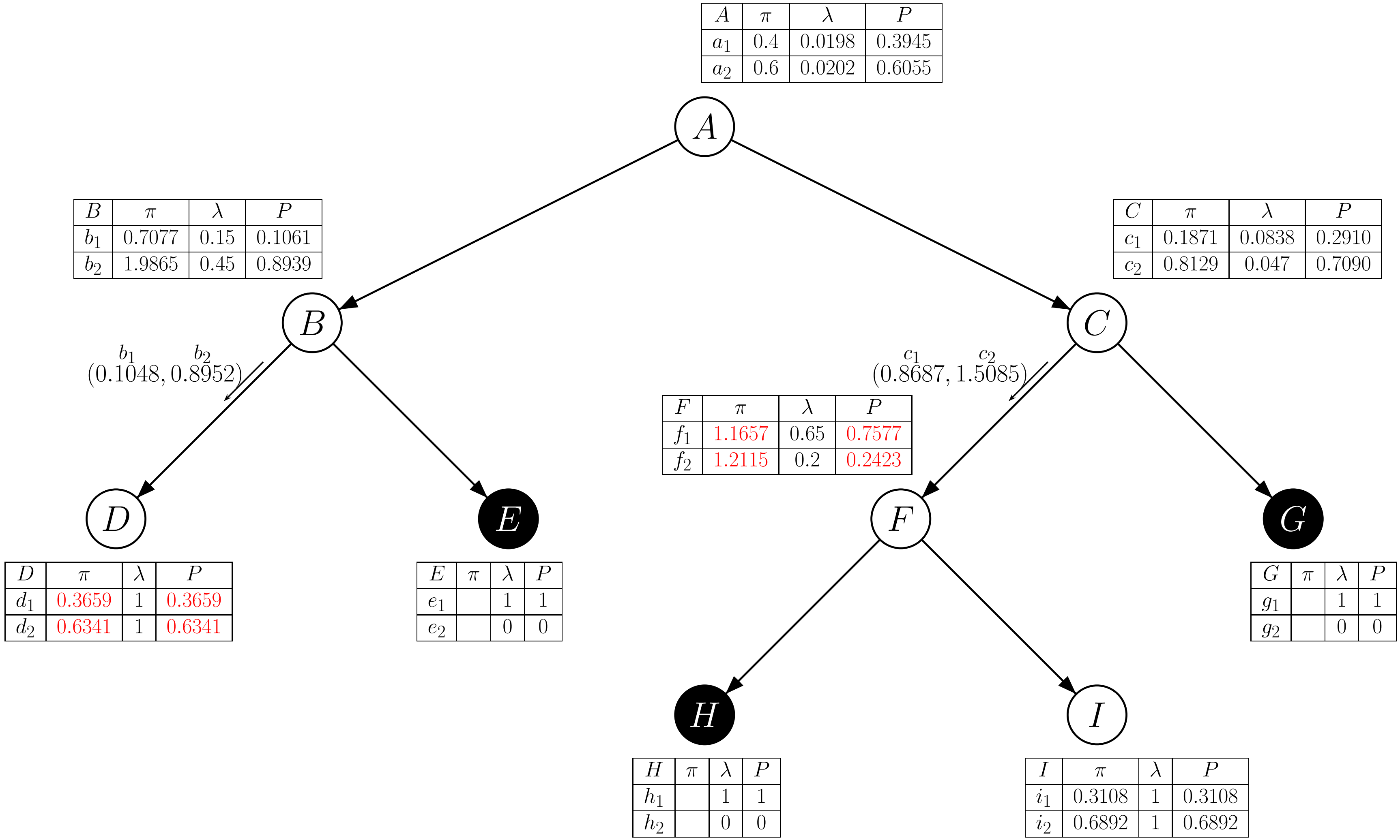
D	π	λ	P
d_1	0.3657	1	0.3657
d_2	0.6343	1	0.6343

E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3108	1	0.3108
i_2	0.6892	1	0.6892



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.7077	0.15	0.1061
b_2	1.9865	0.45	0.8939

C	π	λ	P
c_1	0.1871	0.0838	0.2910
c_2	0.8129	0.047	0.7090

F	π	λ	P
f_1	1.1657	0.65	0.7577
f_2	1.2115	0.2	0.2423

D	π	λ	P
d_1	0.3659	1	0.3659
d_2	0.6341	1	0.6341

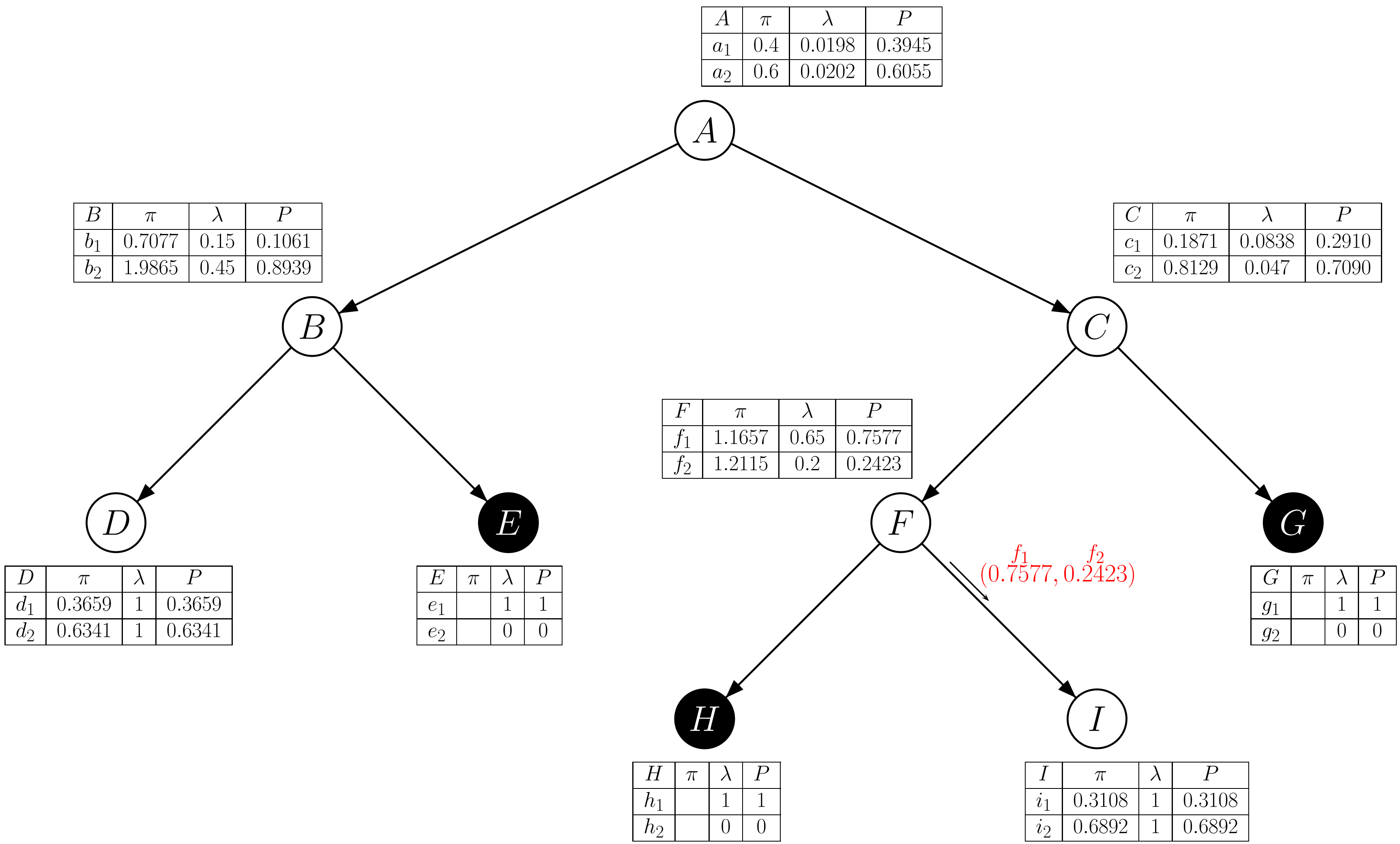
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3108	1	0.3108
i_2	0.6892	1	0.6892

Larger Network (14): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.7077	0.15	0.1061
b_2	1.9865	0.45	0.8939

C	π	λ	P
c_1	0.1871	0.0838	0.2910
c_2	0.8129	0.047	0.7090

F	π	λ	P
f_1	1.1657	0.65	0.7577
f_2	1.2115	0.2	0.2423

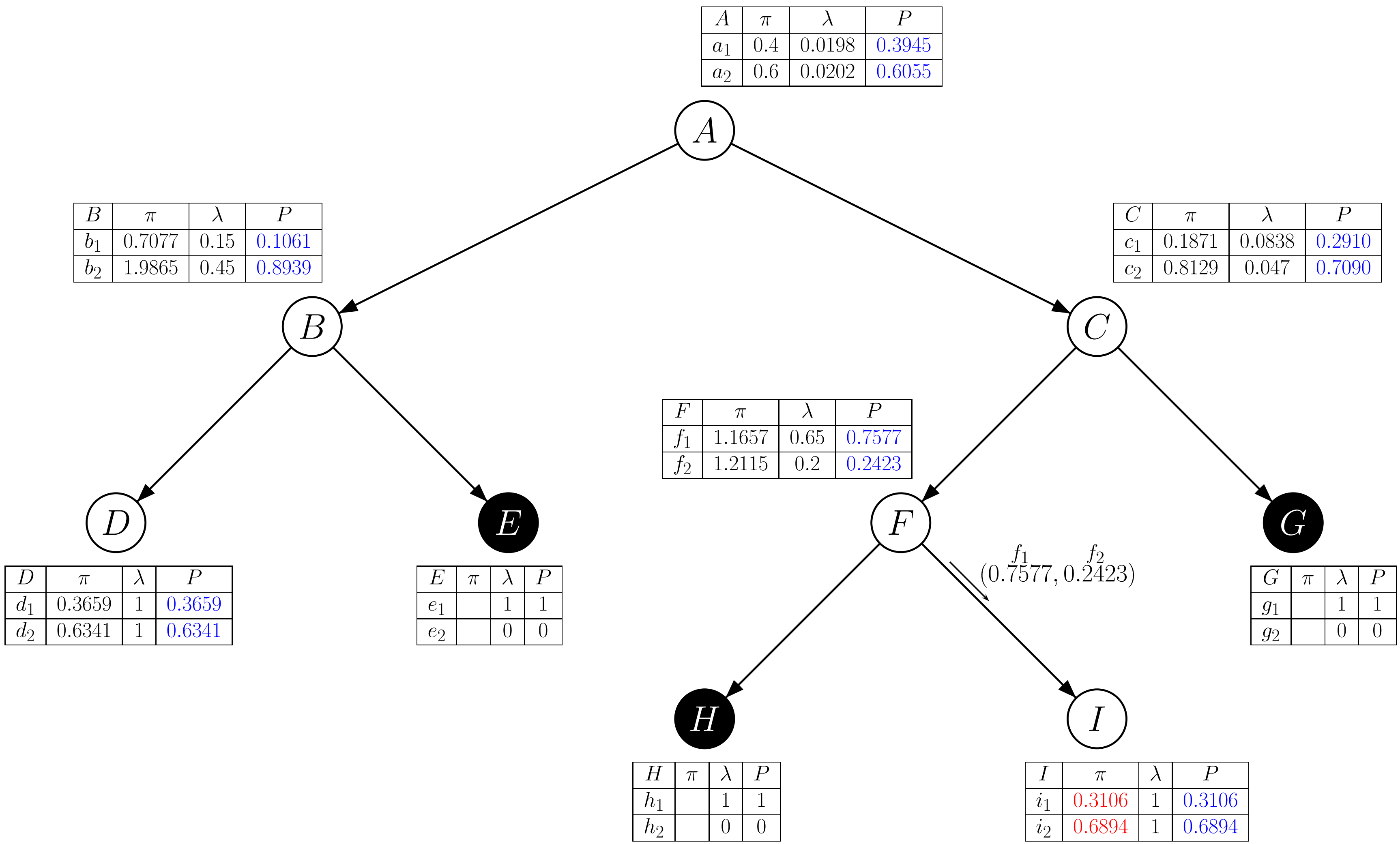
D	π	λ	P
d_1	0.3659	1	0.3659
d_2	0.6341	1	0.6341

E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3108	1	0.3108
i_2	0.6892	1	0.6892



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.7077	0.15	0.1061
b_2	1.9865	0.45	0.8939

C	π	λ	P
c_1	0.1871	0.0838	0.2910
c_2	0.8129	0.047	0.7090

F	π	λ	P
f_1	1.1657	0.65	0.7577
f_2	1.2115	0.2	0.2423

D	π	λ	P
d_1	0.3659	1	0.3659
d_2	0.6341	1	0.6341

E	π	λ	P
e_1		1	1
e_2		0	0

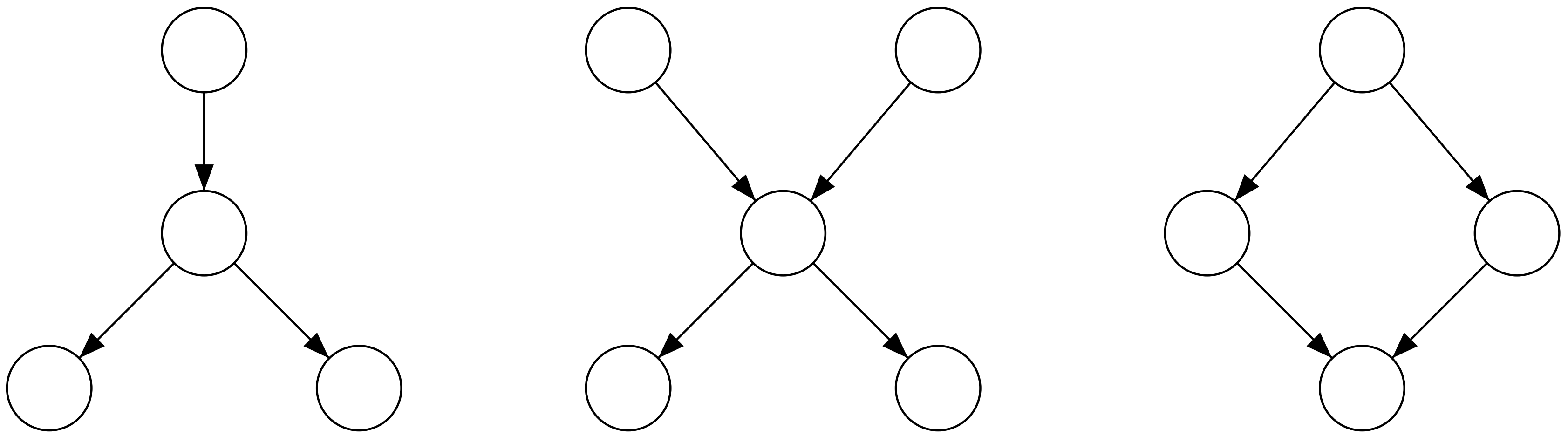
G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3106	1	0.3106
i_2	0.6894	1	0.6894

Propagation in Clique Trees

Problems



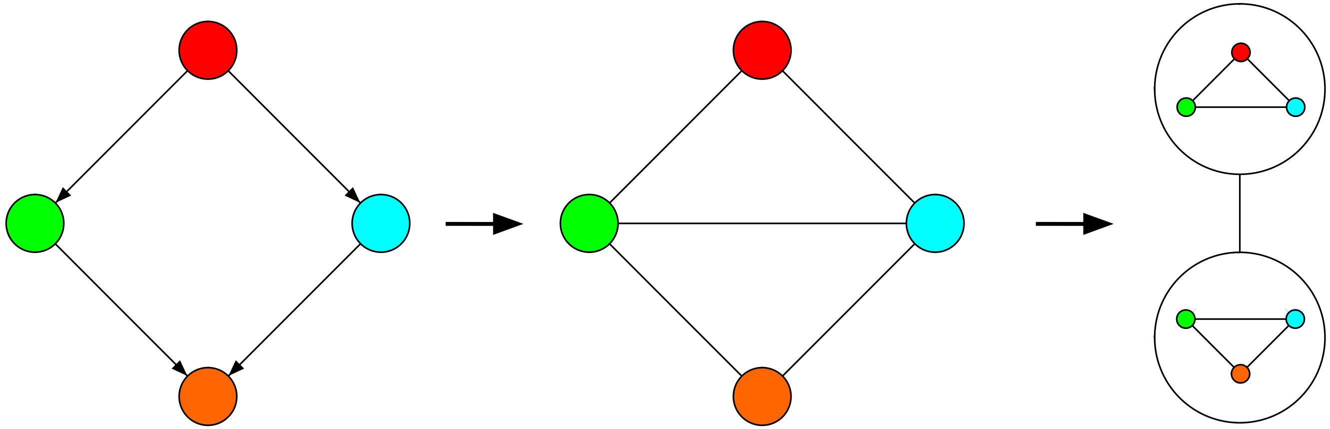
- The propagation algorithm as presented can only deal with *trees*.
- Can be extended to *polytrees* (i. e. singly connected graphs with multiple parents per node).
- However, it cannot handle networks that contain loops!

Main Objectives:

- Transform the cyclic directed graph into a secondary structure without cycles.
- Find a decomposition of the underlying joint distribution.

Task:

- Combine nodes of the original (primary) graph structure.
- These groups form the nodes of a secondary structure.
- Find a transformation that yields tree structure.



Secondary Structure:

- We will generate an undirected graph mimicking (some of) the conditional independence statements of the cyclic directed graph.
- Maximal cliques are identified and form the nodes of the secondary structure.
- Specify a so-called potential function for every clique such that the product of all potentials yields the initial joint distribution.
- In order to propagate evidence, create a **tree** from the clique nodes such that the following property is satisfied:

  If two cliques have some attributes in common, then these attributes have to be contained in every clique of the path connecting the two cliques. (called the **running intersection property**, **RIP**)

Justification:

- Tree: Unique path of evidence propagation.
- RIP: Update of an attribute reaches all cliques which contain it.

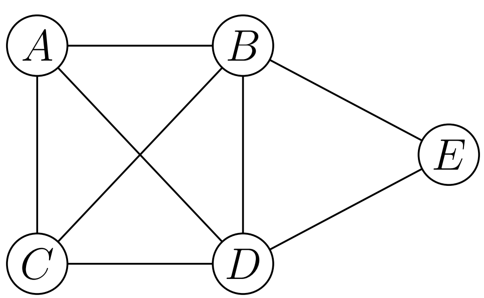
Complete Graph

An undirected Graph $G = (V, E)$ is called *complete*, if every pair of (distinct) nodes is connected by an edge.

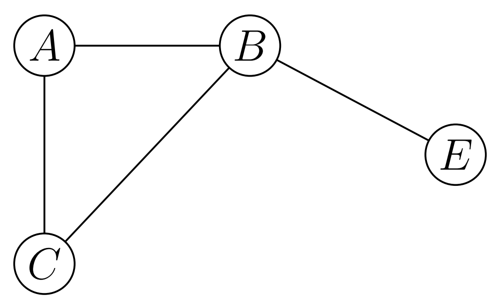
Induced Subgraph

Let $G = (V, E)$ be an undirected graph and $W \subseteq V$ a selection of nodes. Then, $G_W = (W, E_W)$ is called the *subgraph of G induced by W* with E_W being

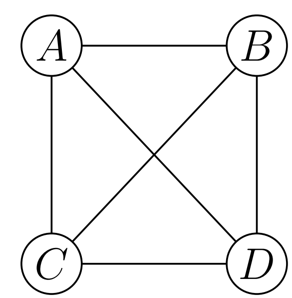
$$E_W = \{(u, v) \in E \mid u, v \in W\}.$$



Incomplete graph



Subgraph (W, E_W)
with $W = \{A, B, C, E\}$



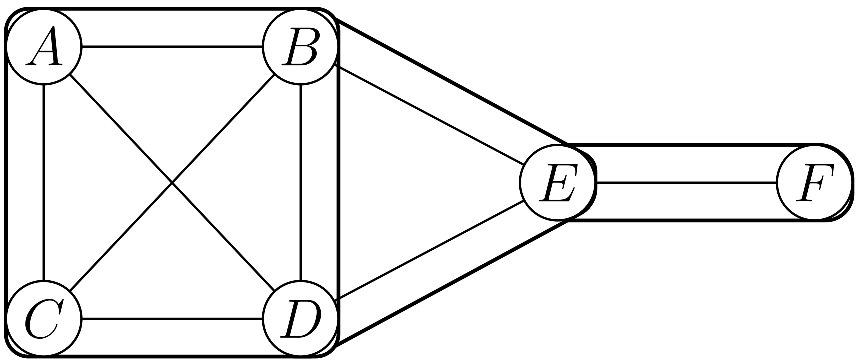
Complete (sub)graph

Complete Set, Clique

Let $G = (V, E)$ be an undirected graph. A set $W \subseteq V$ is called *complete* iff it induces a complete subgraph. It is further called a *clique*, iff W is maximal, i. e. it is not possible to add a node to W without violating the completeness condition.

a) W is complete $\Leftrightarrow$ W induces a complete subgraph

b) W is a clique $\Leftrightarrow$ W is complete and maximal



3 cliques

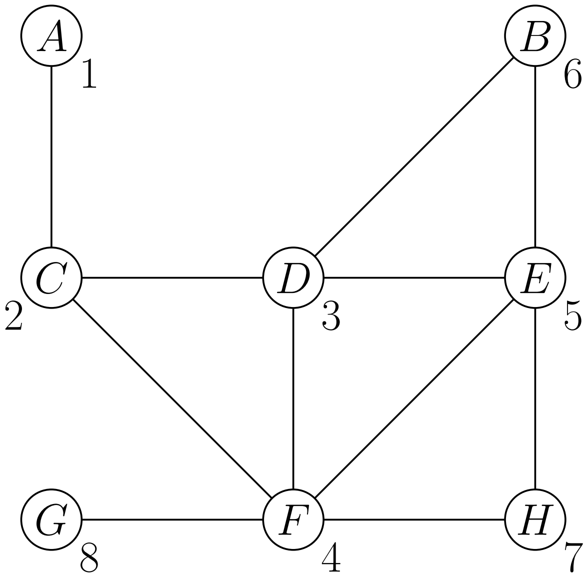
$$C_1 = \{A, B, C, D\}$$
$$C_2 = \{B, D, E\}$$
$$C_3 = \{E, F\}$$

Perfect Ordering

Let $G = (V, E)$ be an undirected graph with n nodes and $\alpha = \langle v_1, \ldots, v_n \rangle$ a total ordering on V. Then, α is called *perfect*, if the following sets

$$\mathrm{adj}(v_i) \cap \{v_1, \ldots, v_{i-1}\} \qquad i = 1, \ldots, n$$

are complete, where $\mathrm{adj}(v_i) = \{w \mid (v_i, w) \in E\}$ returns the adjacent nodes of v_i.



$\alpha = \langle A,\ C,\ D,\ F,\ E,\ B,\ H,\ G \rangle$

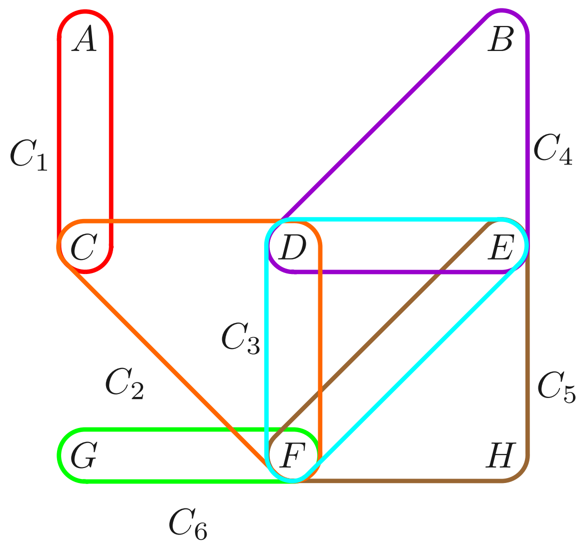
i	$\mathrm{adj}(v_i)$	$\mathrm{adj}(v_i) \cap \{v_1, \ldots, v_{i-1}\}$		
1	$\{C\}$	$\{C\} \cap \emptyset$	$= \emptyset$	complete
2	$\{A, D, F\}$	$\{A\} \cap \{A, D, F\}$	$= \{A\}$	complete
3	$\{C, B, E, F\}$	$\{A, C\} \cap \{C, B, E, F\}$	$= \{C\}$	complete
4	$\{G, C, D, E, H\}$	$\{A, C, D\} \cap \{G, C, D, E, H\}$	$= \{C, D\}$	complete
5	$\{B, D, F, H\}$	$\{A, C, D, F\} \cap \{B, D, F, H\}$	$= \{D, F\}$	complete
6	$\{D, E\}$	$\{A, C, D, F, E\} \cap \{D, E\}$	$= \{D, E\}$	complete
7	$\{F, E\}$	$\{A, C, D, F, E, B\} \cap \{F, E\}$	$= \{F, E\}$	complete
8	$\{F\}$	$\{A, C, D, F, E, B, H\} \cap \{F\}$	$= \{F\}$	complete

α is a perfect ordering

Running Intersection Property

Let $G = (V, E)$ be an undirected graph with p cliques. An ordering of these cliques has the *running intersection property (RIP)*, if for every $j > 1$ there exists an $i < j$ such that:

$$C_j \;\cap\; \left(C_1 \cup \cdots \cup C_{j-1}\right) \;\subseteq\; C_i$$

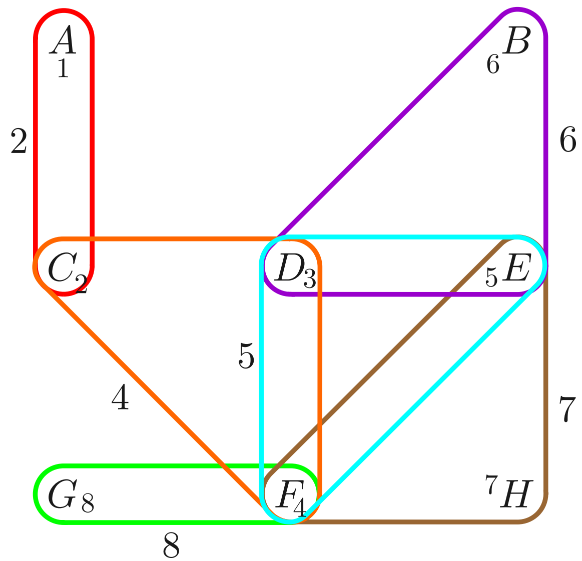


$\xi = \langle C_1,\ C_2,\ C_3,\ C_4,\ C_5,\ C_6 \rangle$

j				i
2	$C_2 \cap C_1$	$= \{C\}$	$\subseteq C_1$	1
3	$C_3 \cap (C_1 \cup C_2)$	$= \{D, F\}$	$\subseteq C_2$	2
4	$C_4 \cap (C_1 \cup C_2 \cup C_3)$	$= \{D, E\}$	$\subseteq C_3$	3
5	$C_5 \cap (C_1 \cup C_2 \cup C_3 \cup C_4)$	$= \{E, F\}$	$\subseteq C_3$	3
6	$C_6 \cap (C_1 \cup C_2 \cup C_3 \cup C_4 \cup C_5)$	$= \{F\}$	$\subseteq C_5$	5

ξ has running intersection property

If a node ordering α of an undirected graph $G = (V, E)$ is perfect and the cliques of G are ordered according to the highest rank (w. r. t. α) of the containing nodes, then this clique ordering has RIP.

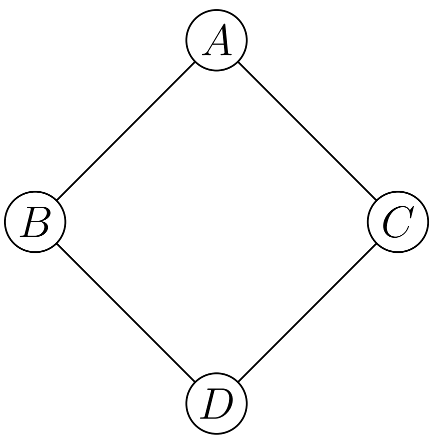


Clique	Rank		
$\{A, C\}$	$\max\{\alpha(A), \alpha(C)\}$	$= 2$	$\rightarrow C_1$
$\{C, D, F\}$	$\max\{\alpha(C), \alpha(D), \alpha(F)\}$	$= 4$	$\rightarrow C_2$
$\{D, E, F\}$	$\max\{\alpha(D), \alpha(E), \alpha(F)\}$	$= 5$	$\rightarrow C_3$
$\{B, D, E\}$	$\max\{\alpha(B), \alpha(D), \alpha(E)\}$	$= 6$	$\rightarrow C_4$
$\{F, E, H\}$	$\max\{\alpha(F), \alpha(E), \alpha(H)\}$	$= 7$	$\rightarrow C_5$
$\{F, G\}$	$\max\{\alpha(F), \alpha(G)\}$	$= 8$	$\rightarrow C_6$

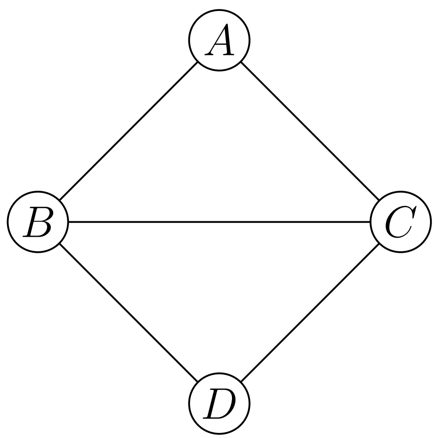
How to get a perfect ordering?

Triangulated Graph

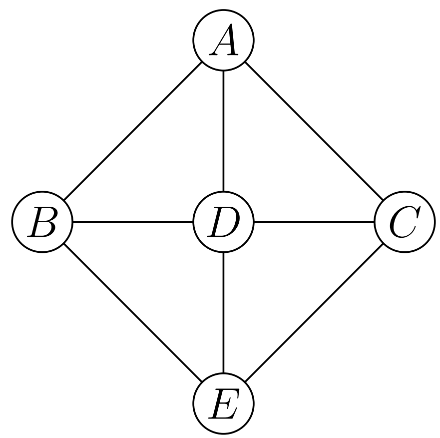
An undirected graph is called *triangulated* if every simple loop (i. e. path with identical start and end node but with any other node occurring at most once) of length greater 3 has a chord.



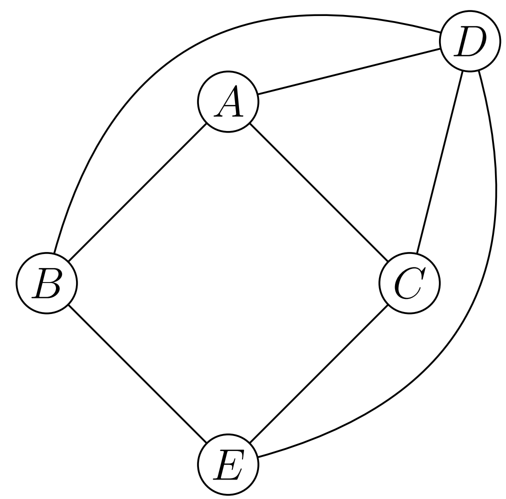
not triangulated



triangulated



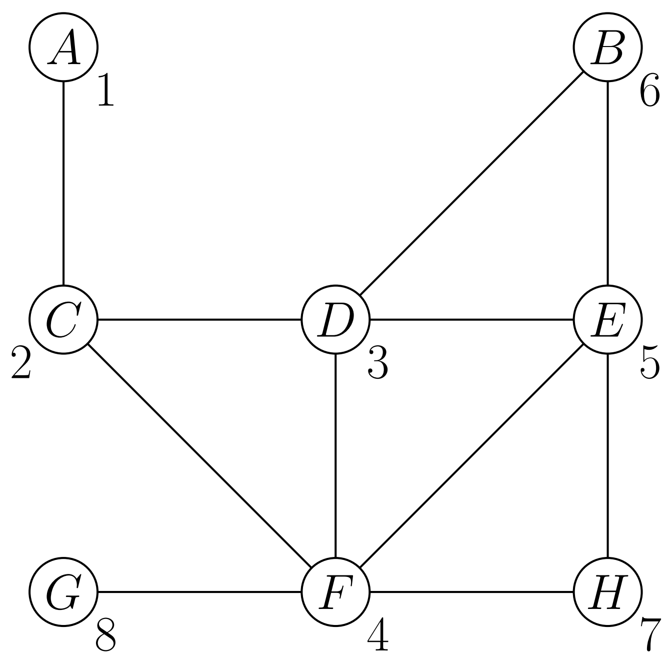
not triangulated



no chord for $\langle A, B, E, C\rangle$

Maximum Cardinality Search

Let $G = (V, E)$ be an undirected graph. An ordering according *maximum cardinality search (MCS)* is obtained by first assigning 1 to an arbitray node. If n numbers are assigned the node that is connected to most of the nodes already numbered gets assigned number $n + 1$.



3 can be assigned to D or F

6 can be assigned to H or B

Triangulated Graphs (3)

An undirected graph is triangulated iff the ordering obtained by MCS is perfect.

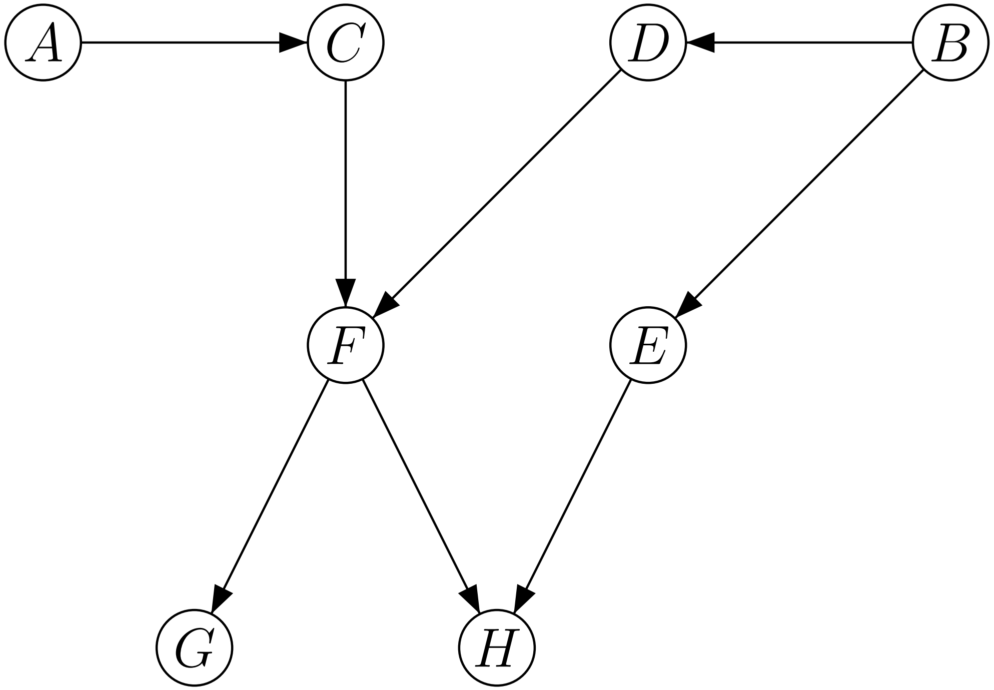
To check whether a graph is triangulated is efficient to implement. The optimization problem that is related to the triangulation task is NP-hard. However, there are good heuristics.

Moral Graph (Repetition)

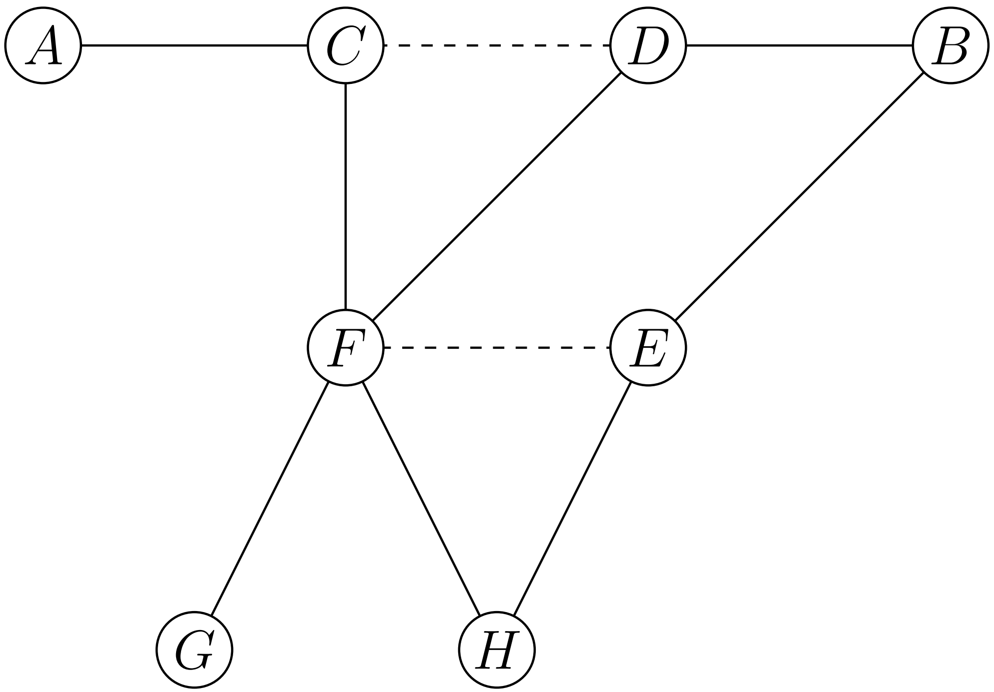
Let $G = (V, E)$ be a directed acyclic graph. If $u, w \in W$ are parents of $v \in V$ connect u and w with an (arbitrarily oriented) edge. After the removal of all edge directions the resulting graph $G_m = (V, E')$ is called the *moral graph* of G.

Join-Tree Construction (1)

Given directed graph.

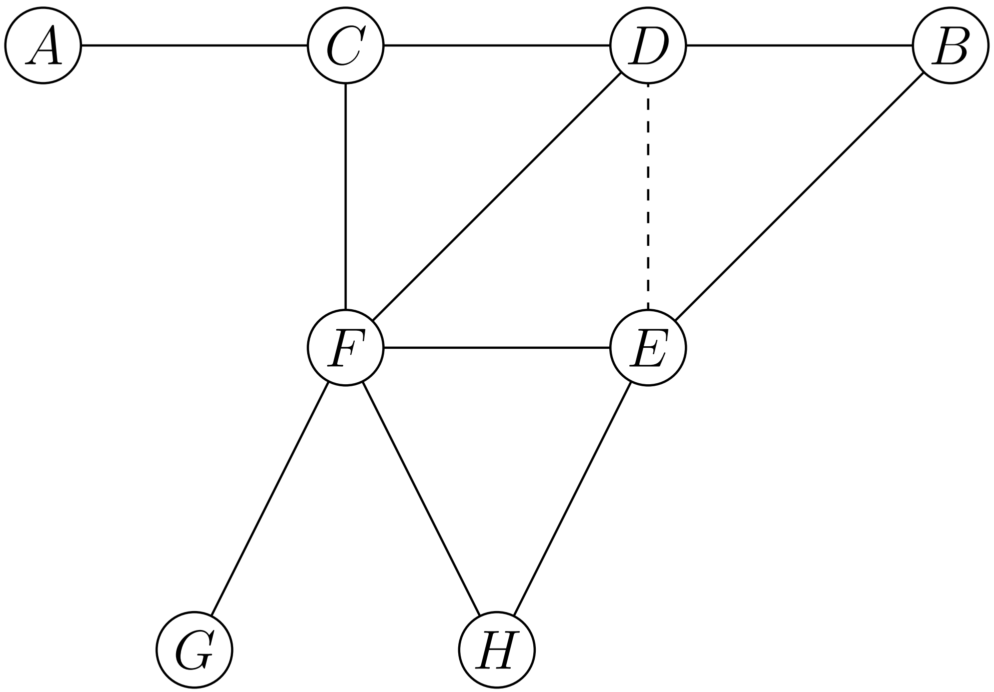


Join-Tree Construction (2)



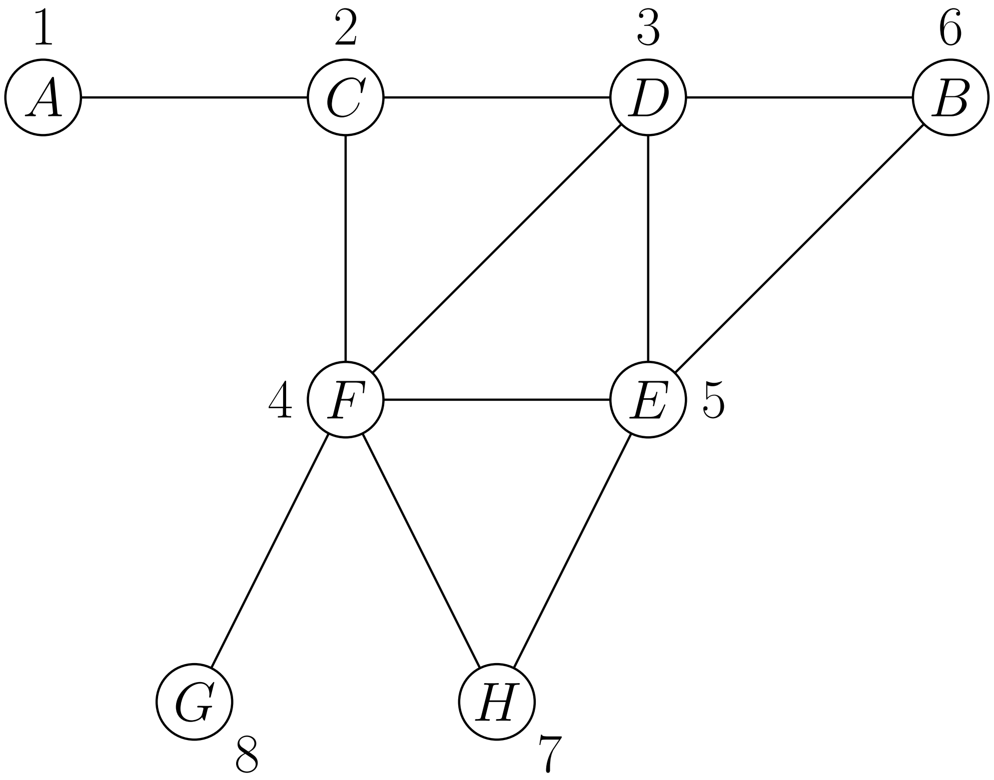
- Moral graph

Join-Tree Construction (3)



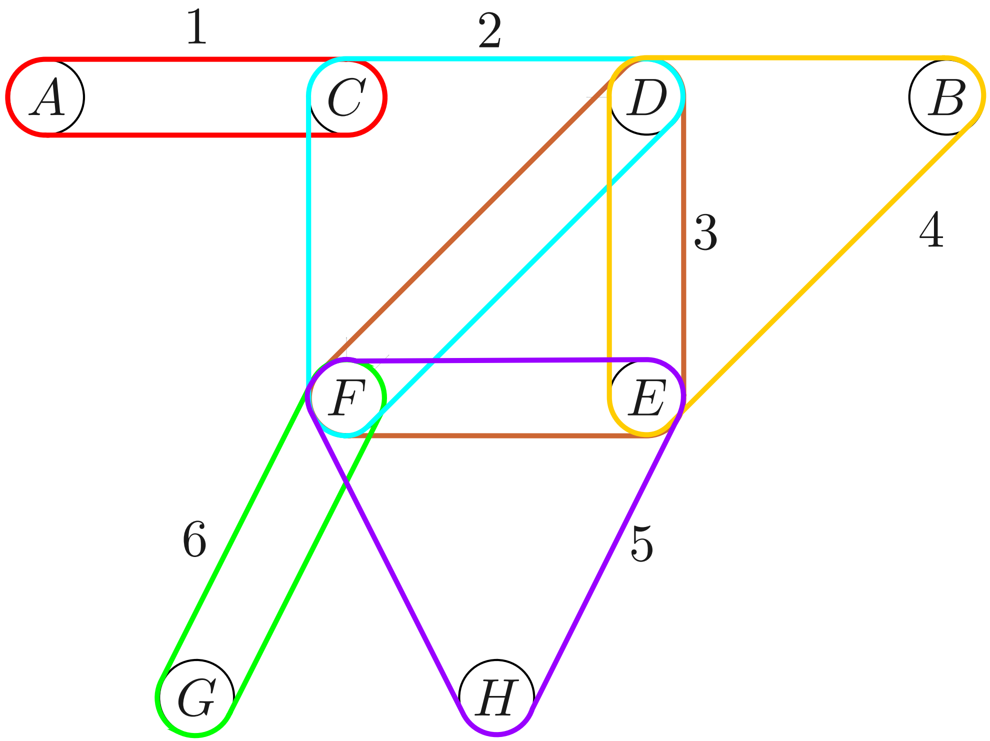
- Moral graph
- Triangulated graph

Join-Tree Construction (4)



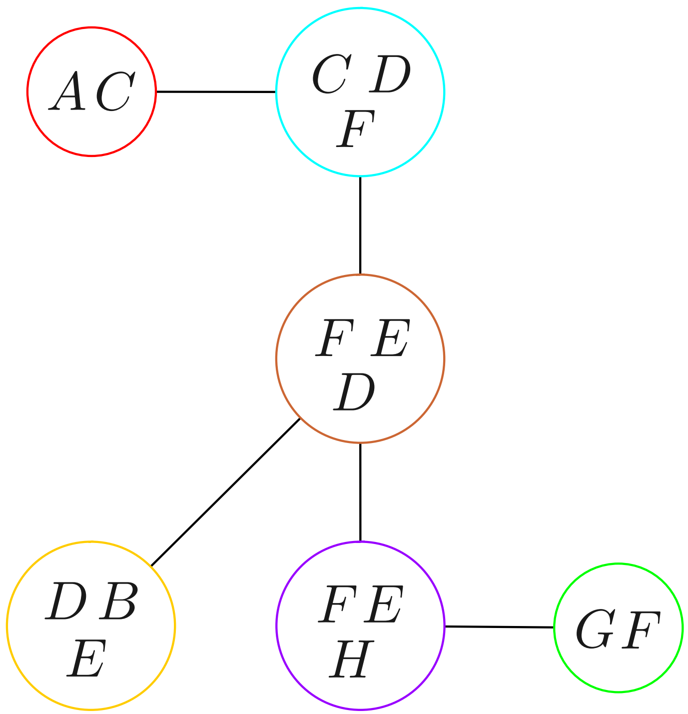
- Moral graph
- Triangulated graph
- MCS yields perfect ordering

Join-Tree Construction (5)



- Moral graph
- Triangulated graph
- MCS yields perfect ordering
- Clique order has RIP

Join-Tree Construction (6)



- Moral graph
- Triangulated graph
- MCS yields perfect ordering
- Clique order has RIP
- Form a join-tree

  Two cliques can be connected if they have a non-empty intersection. The generation of the tree follows the RIP. In case of a tie, connect cliques with the largest intersection. (e. g. DBE—FED instead of DBE—CFD) Break remaining ties arbitrarily.

Example: Expert Knowledge

- **Qualitative knowledge:**

  Metastatic cancer is a possible cause of brain tumor, and is also an explanation for increased total serum calcium. In turn, either of these could explain a patient falling into a coma. Severe headache is also possibly associated with a brain tumor.

- **Special case:**

  The patient has heavy headache.

- **Query:**

  Will the patient fall into coma?

Example: Choice of State Space

Attribute		Possible Values	
A	metastatic cancer	$\mathrm{dom}(A) = \{a_1, a_2\}$	$\cdot_1 = \text{existing}$
B	increased total serum calcium	$\mathrm{dom}(B) = \{b_1, b_2\}$	$\cdot_2 = \text{notexisting}$
C	brain tumor	$\mathrm{dom}(C) = \{c_1, c_2\}$	
D	coma	$\mathrm{dom}(D) = \{d_1, d_2\}$	
E	severe headache	$\mathrm{dom}(E) = \{e_1, e_2\}$	

Exhaustive state space:

$$\Omega = \mathrm{dom}(A) \times \mathrm{dom}(B) \times \mathrm{dom}(C) \times \mathrm{dom}(D) \times \mathrm{dom}(E)$$

Marginal and conditional probabilities have to be specified!

Example: Qualitative Knowledge

$$\left.\begin{aligned} P(e_1 \mid c_1) &= 0.8 \\ P(e_1 \mid c_2) &= 0.6 \end{aligned}\right\}\ \text{headaches common, but more common if tumor present}$$

$$\left.\begin{aligned} P(d_1 \mid b_1, c_1) &= 0.8 \\ P(d_1 \mid b_1, c_2) &= 0.8 \\ P(d_1 \mid b_2, c_1) &= 0.8 \\ P(d_1 \mid b_2, c_2) &= 0.05 \end{aligned}\right\}\ \text{coma rare but common, if either cause is present}$$

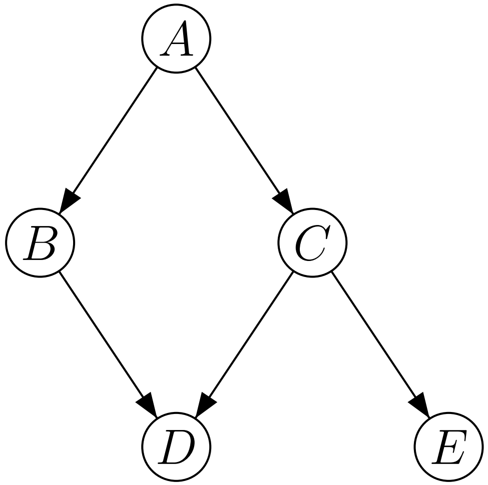
$$\left.\begin{aligned} P(b_1 \mid a_1) &= 0.8 \\ P(b_1 \mid a_2) &= 0.2 \end{aligned}\right\}\ \begin{array}{c}\text{increased calcium uncommon,} \\ \text{but common consequence of metastases}\end{array}$$

$$\left.\begin{aligned} P(c_1 \mid a_1) &= 0.2 \\ P(c_1 \mid a_2) &= 0.05 \end{aligned}\right\}\ \text{brain tumor rare, and uncommon consequence of metastases}$$

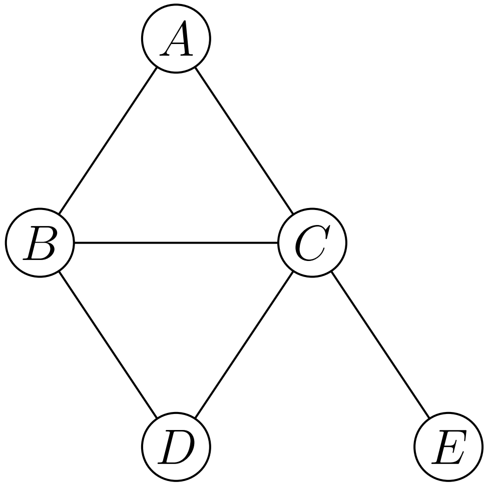
$$P(a_1) = 0.2 \quad \}\ \text{incidence of metastatic cancer in relevant clinic}$$

Propagation on Cliques (1)

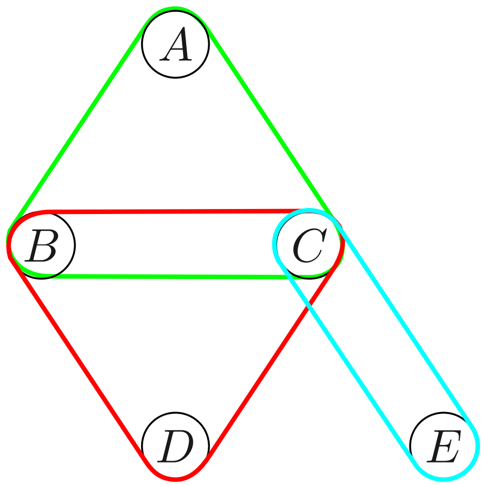
Example: Metastatic Cancer



Dependencies



Moralization/Triangulation



MCS, hyper graph

Clique tree with separator sets

Propagation on Cliques (3)

Quantitative knowledge:

(a,b,c)	$P(a,b,c)$
a_1, b_1, c_1	0.032
a_2, b_1, c_1	0.008
$\vdots$	$\vdots$
a_2, b_2, c_2	0.608

(b,c,d)	$P(b,c,d)$
b_1, c_1, d_1	0.032
b_2, c_1, d_1	0.032
$\vdots$	$\vdots$
b_2, c_2, d_2	0.608

(c,e)	$P(c,e)$
c_1, e_1	0.064
c_2, e_1	0.552
c_1, e_2	0.016
c_2, e_2	0.368

Potential representation:

$$\begin{aligned} P(A,B,C,D,E,) &= P(A \mid \emptyset)P(B \mid A)P(C \mid A)P(B \mid BC)P(E \mid C) \\ &= \frac{P(A,B,C)P(B,C,D),P(C,E)}{P(BC)P(C)} \end{aligned}$$

Propagation on Cliques (4)

Propagation:

- $P(d_1) = 0.32$, evidence $E = e_1$, desired: $P^*(\ldots) = P(\cdot \mid \{e_1\})$

$$P(c \mid e_1) = P(c \mid e_1)$$ conditional marginal distribution

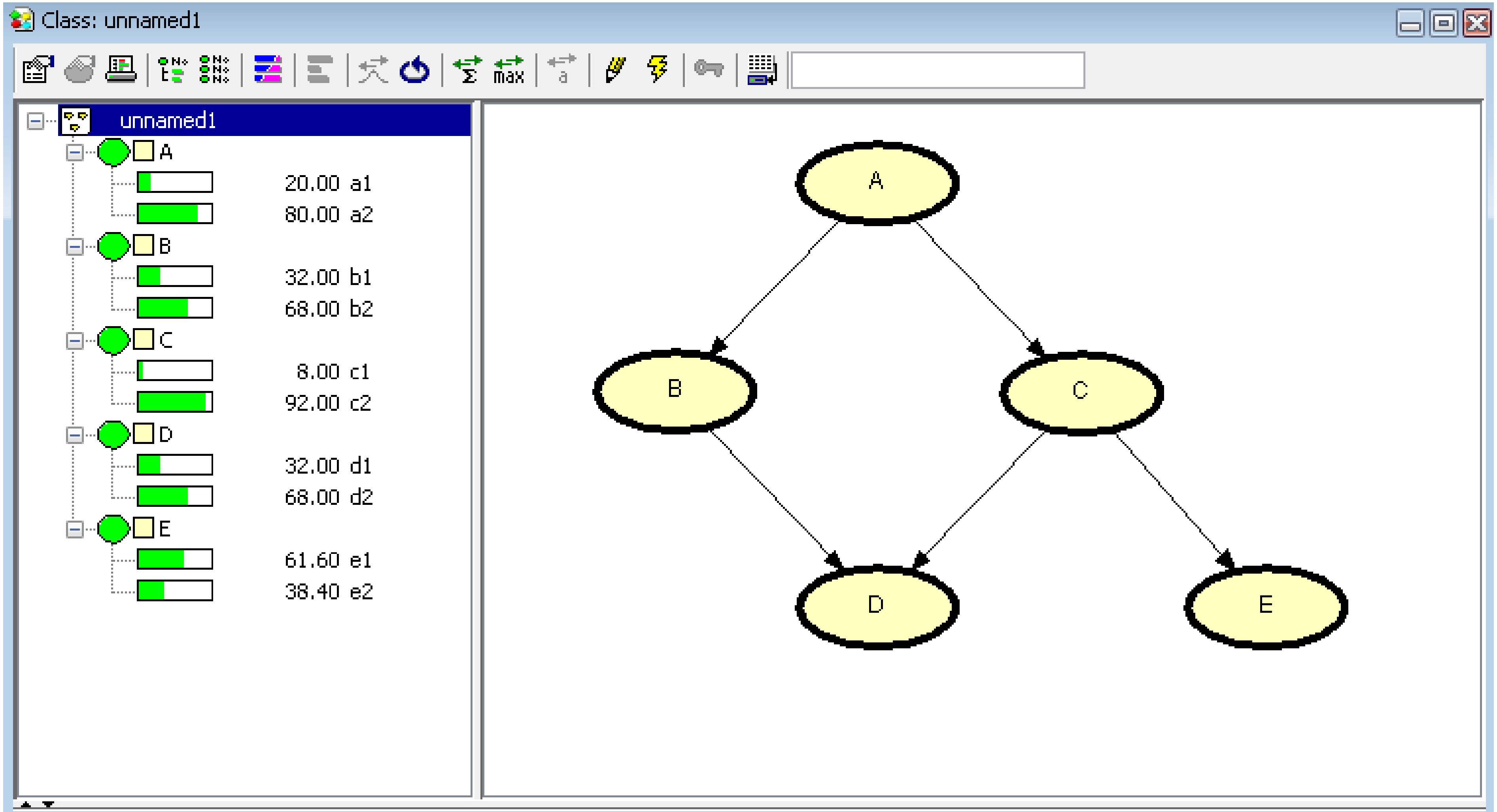
$$P(b, c, d \mid e_1) = \frac{P(b, c, d)}{P(c)} P(c \mid e_1)$$ multipl./division with separation prob.

$P(b, c)$, $P(b, c \mid e_1)$ calculate marginal distributions

$$P(a, b, c \mid e_1) = \frac{P(a, b, c)}{P(b, c)} P(b, c \mid e_1)$$ multipl./division with separation prob.

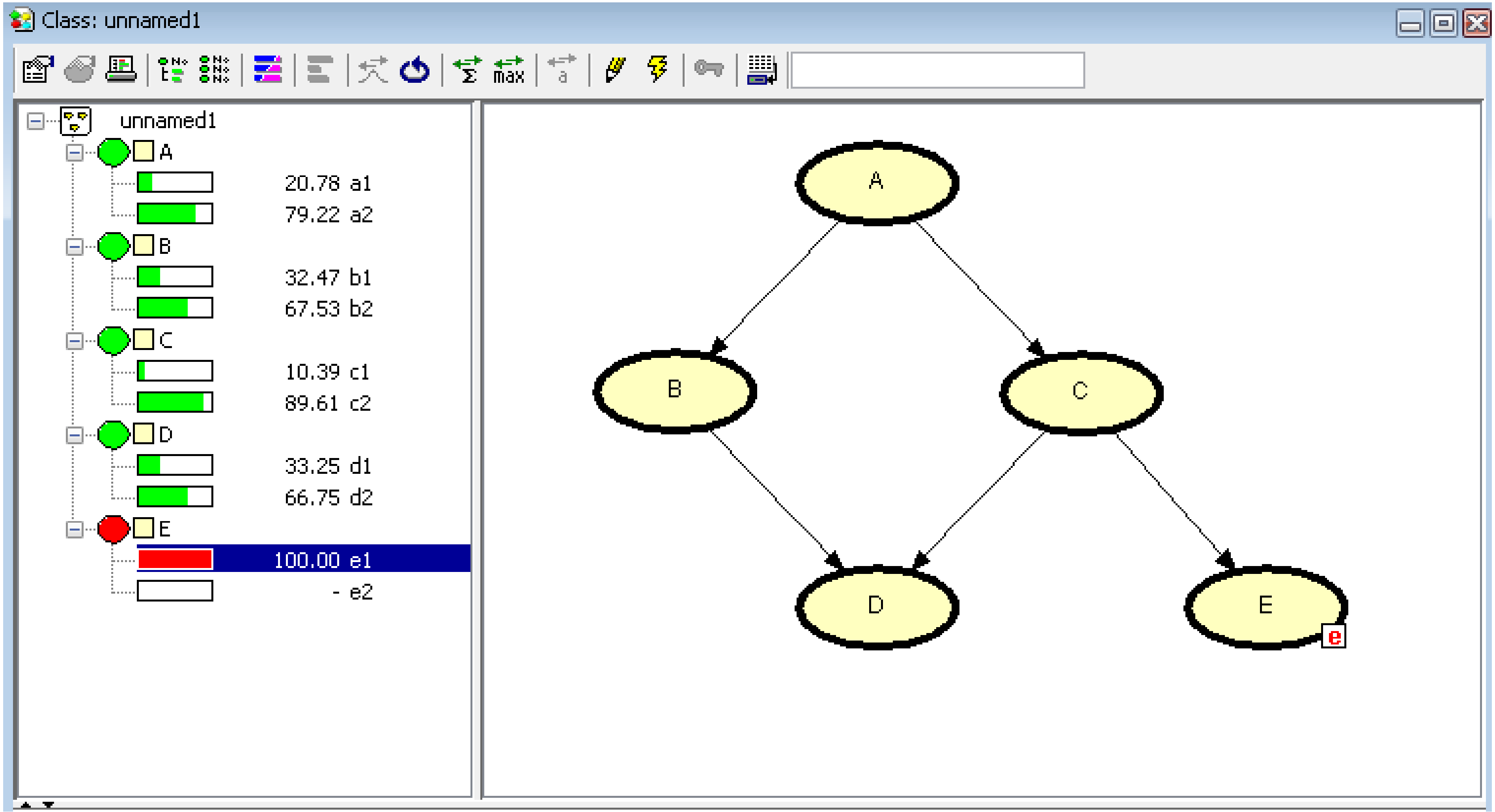
$$P(d_1 \mid e_1) = P(d_1 \mid e_1) = 0.33$$

Propagation on Cliques (5)



Marginal distributions in the HUGIN tool.

Propagation on Cliques (6)



Conditional marginal distributions with evidence $E = e_1$

Potential Representation

Let $V = \{X_j\}$ be a set of random variables $X_j : \Omega \to \mathrm{dom}(X_j)$ and P the joint distribution over V. Further, let

$$\{W_i \mid W_i \subseteq V, 1 \leq i \leq p\}$$

a family of subsets of V with associated functions

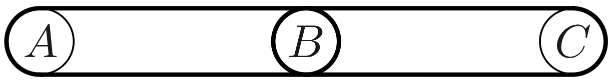
$$\psi_i : \bigtimes_{X_j \in W_i} \mathrm{dom}(X_j) \to \mathbb{R}$$

It is said that $P(V)$ *factorizes* according $\Big(\{W_1, \ldots, W_p\}, \{\psi_1, \ldots, \psi_p\}\Big)$ if $P(V)$ can be written as:

$$P(v) = k \cdot \prod_{i=1}^{p} \psi_i(w_i)$$

where $k \in \mathbb{R}$, w_i is a realization of W_i that meets the values of v.

Example



$V = \{A, B, C\},\ W_1 = \{A, B\},\ W_2 = \{B, C\}$

$\mathrm{dom}(A) = \{a_1, a_2\}$
$\mathrm{dom}(B) = \{b_1, b_2\}$
$\mathrm{dom}(C) = \{c_1, c_2\}$

$P(a, b, c) = \frac{1}{8}$

$\psi_1 : \{a_1, a_2\} \times \{b_1, b_2\} \to \mathbb{R}$
$\psi_2 : \{b_1, b_2\} \times \{c_1, c_2\} \to \mathbb{R}$

$\psi_1(a, b) = \frac{1}{4}$
$\psi_2(b, c) = \frac{1}{2}$

$(\{W_1, W_2\}, \{\psi_1, \psi_2\})$ is a potential representation of P.

Factorization of a Belief Network

Let (V, E, P) be an belief network and $\{C_1, \ldots, C_p\}$ the cliques of the join tree. For every node $v \in V$ choose a clique C such that v and all of its parents are contained in C, i. e. $\{v\} \cup c(v) \subseteq C$. The chosen clique is designated as $f(v)$.

To arrive at a factorization $(\{C_1, \ldots, C_p\}, \{\psi_1, \ldots, \psi_p\})$ of P the factor potentials are:

$$\psi_i(c_i) = \prod_{v:f(v)=C_i} P(v \mid c(v))$$

Separator Sets and Residual Sets

Let $\{C_1, \ldots, C_p\}$ be a set of cliques w. r. t. V. The sets

$$S_i = C_i \cap (C_1 \cup \cdots \cup C_{i-1}), \qquad i = 1, \ldots, p, \qquad S_1 = \emptyset$$

are called *separator sets* with their corresponding *residual sets*

$$R_i = C_i \backslash S_i$$

Decomposition w. r. t. a Join-Tree

- Given a clique ordering $\{C_1, \ldots, C_p\}$ that satisfies the RIP, we can easily conclude the following separation statements:

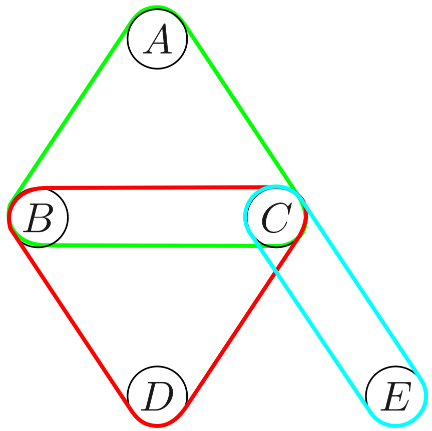
$$R_i \perp\!\!\!\perp (C_1 \cup \cdots \cup C_{i-1}) \backslash S_i \mid S_i \qquad \text{for } i > 1$$

- Hence, we can formulate the following factorization:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{p} P(R_i \mid S_i),$$

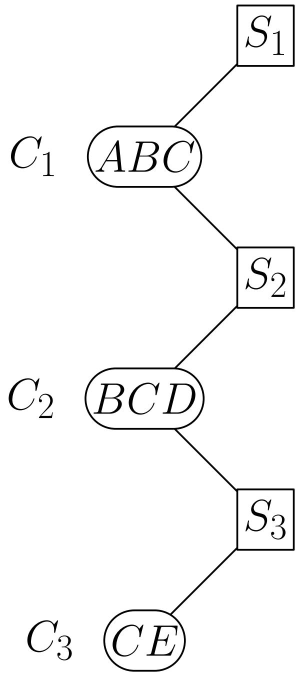
which also gives us a representation in terms of conditional probabilities (as for directed graphs before).

Example



$$S_1 = \emptyset \qquad R_1 = \{A, B, C\}$$
$$S_2 = \{B, C\} \qquad R_2 = \{D\}$$
$$S_3 = \{C\} \qquad R_3 = \{E\}$$

$$f(A) = C_1$$
$$f(B) = C_1$$
$$f(C) = C_1$$
$$f(D) = C_2$$
$$f(E) = C_3$$

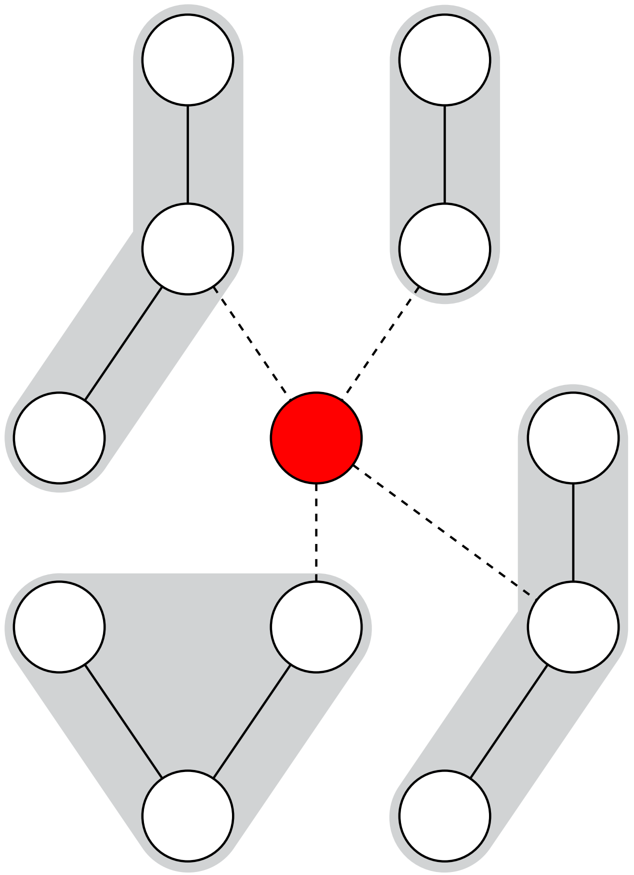


$$\psi_1(C_1) = P(A, B, C \mid \emptyset) = P(A) \cdot P(C \mid A) \cdot P(B|A)$$
$$\psi_2(C_2) = P(D \mid B, C)$$
$$\psi_3(C_3) = P(E \mid C)$$

Propagation is accomplished by sending messages across the cliques in the tree. The emerging potentials are maintained by each clique.

Main Idea

- Incorporate evidence into the clique potentials.
- Since we are dealing with a tree structure, exploit the fact that a clique "separates" all its neighboring cliques (and their respective subtrees) from each other.
- Apply a message passing scheme to inform neighboring cliques about evidence.
- Since we do not have edge directions, we will only need one type of message.
- After having updated all cliques' potentials, we marginalize (and normalize) to get the probabilities of single attributes.



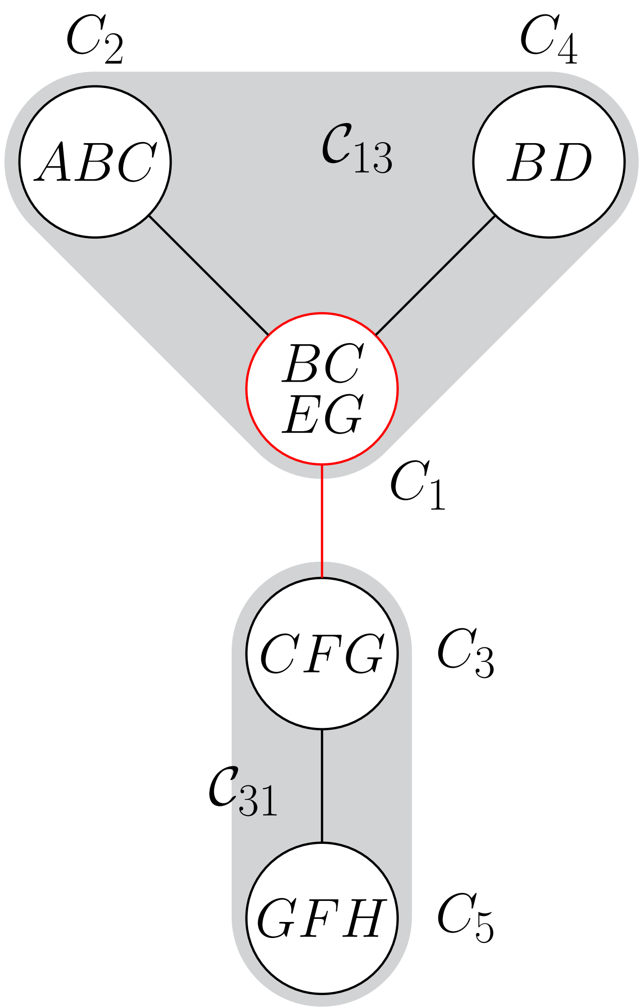
Incorporating Evidence

- Every clique C_i maintains a potential function ψ_i.
- If for an attribute E some evidence e becomes known, we alter all potential functions of cliques containing E as follows:

$$\psi_i^*(c_i) = \begin{cases} 0, & \text{if a value in } c_i \text{ is inconsistent with } e \\ \psi_i(c_i), & \text{otherwise} \end{cases}$$

- All other potential functions are unchanged.

Notation and Nomenclature

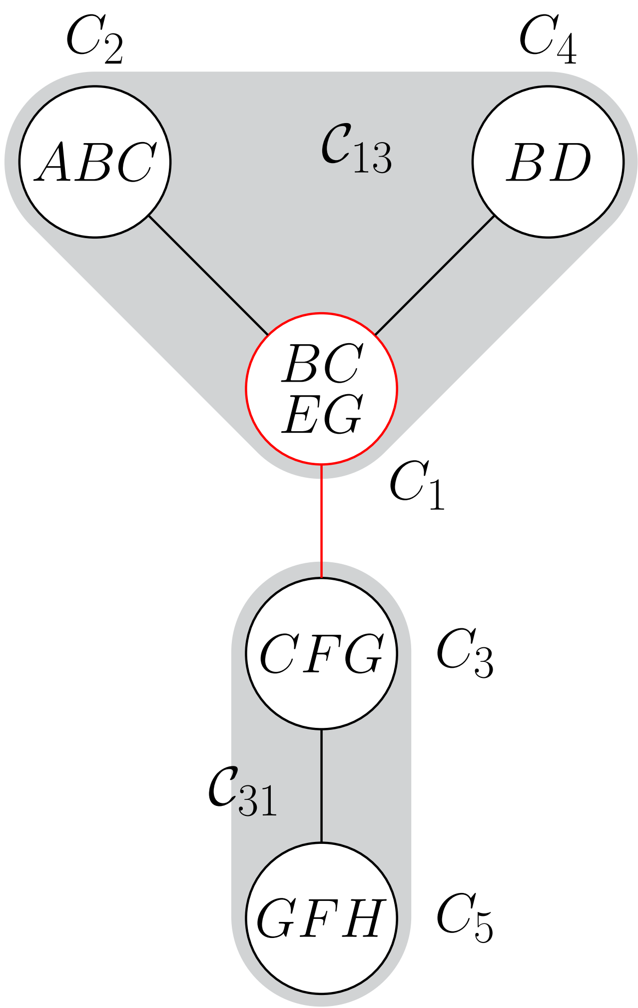


In general:

- Clique C_i has q neighboring cliques $B_1, \ldots, B_q$.
- $\mathcal{C}_{ij}$ is the set of cliques in the subtree containing C_i after dropping the link to B_j.
- X_{ij} is the set of attributes in the cliques of $\mathcal{C}_{ij}$.
- $V = X_{ij} \cup X_{ji}$ (complementary sets)
- $S_{ij} = S_{ji} = C_i \cap C_j$ (not shown here)
- $R_{ij} = X_{ij} \setminus S_{ij}$ (not shown here)

Here:

- Neighbors of C_1: $\{C_2, C_4, C_3\}$, $\mathcal{C}_{13} = \{C_1, C_2, C_4\}$
- $X_{13} = \{A, B, C, D, E, G\}$, $S_{13} = \{C, G\}$
- $V = X_{13} \cup X_{31} = \{A, B, C, D, E, F, G, H\}$
- $R_{13} = \{A, B, D, E\}$, $R_{31} = \{F, H\}$

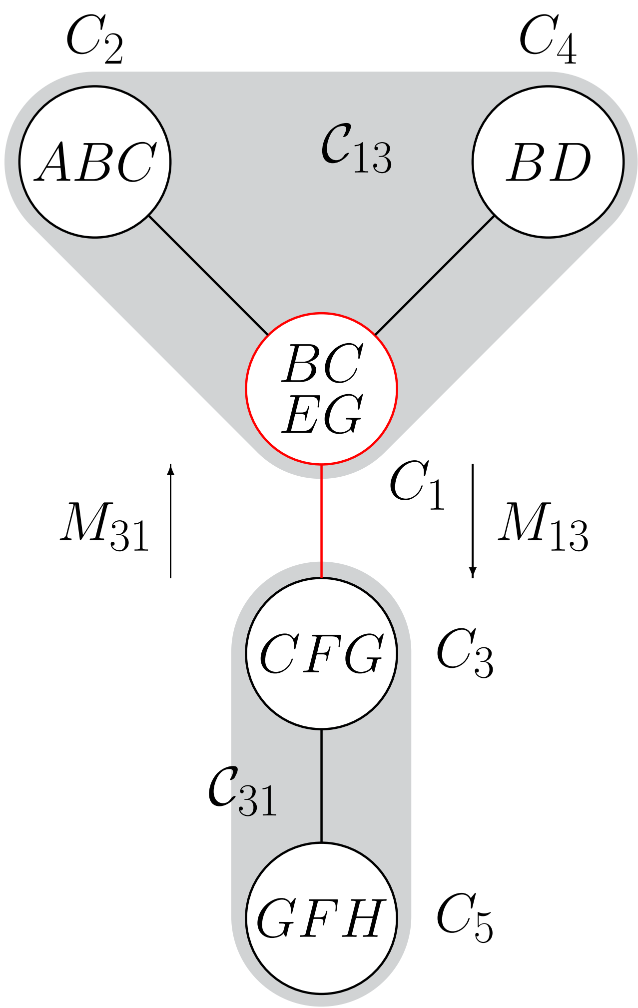


Task: Calculate $P(s_{ij})$:

$$\begin{aligned} V \setminus S_{ij} &= (X_{ij} \cup X_{ji}) \setminus S_{ij} \\ &= (X_{ij} \setminus S_{ij}) \ \cup \ (X_{ji} \setminus S_{ij}) \\ &= R_{ij} \ \cup \ R_{ji} \end{aligned}$$

$$\begin{aligned} V \setminus S_{13} &= (X_{13} \cup X_{31}) \setminus S_{13} \\ &= R_{13} \ \cup \ R_{31} \\ V \setminus \{C, G\} &= \{A, B, D, E\} \ \cup \ \{F, H\} \\ &= \{A, B, D, E, F, H\} \end{aligned}$$

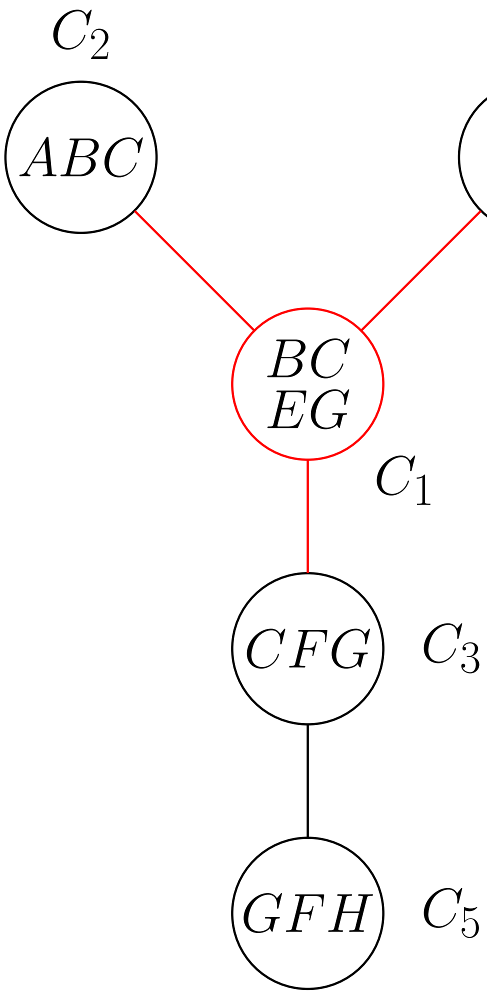
Note: R_{ij} is the set of attributes that are in C_i's subtree but not in B_j's. Therefore, R_{ij} and R_{ji} are always **disjoint**.



Task: Calculate $P(s_{ij})$:

$$
\begin{aligned}
P(s_{ij}) &= \sum_{v \setminus s_{ij}} \prod_{k=1}^{m} \psi_k(c_k) \\
&\overset{\text{last slide}}{=} \sum_{r_{ij} \cup r_{ji}} \prod_{k=1}^{m} \psi_k(c_k) \\
&\overset{\text{sum rule}}{=} \left(\sum_{r_{ij}} \prod_{c_k \in \mathcal{C}_{ij}} \psi_k(c_k) \right) \cdot \left(\sum_{r_{ji}} \prod_{c_k \in \mathcal{C}_{ji}} \psi_k(c_k) \right) \\
&= M_{ij}(s_{ij}) \cdot M_{ji}(s_{ij})
\end{aligned}
$$

M_{ij} is the message sent from C_i to neighbor B_j and vice versa.

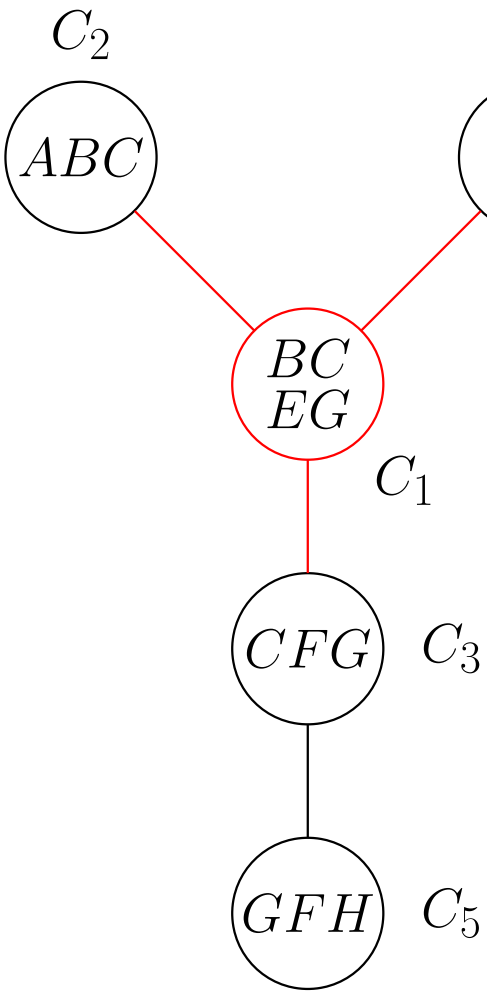


Task: Calculate $P(c_i)$:

$$
\begin{aligned}
V \setminus C_i &= \left(\bigcup_{k=1}^{q} X_{ki} \right) \setminus C_i \\
&= \bigcup_{k=1}^{q} \left(X_{ki} \setminus C_i \right) \\
&= \bigcup_{k=1}^{q} R_{ki}
\end{aligned}
$$

Example:

$$
\begin{aligned}
V \setminus C_1 &= R_{21} \cup R_{41} \cup R_{31} \\
\{A, D, F, H\} &= \{A\} \cup \{D\} \cup \{F, H\}
\end{aligned}
$$



Task: Calculate $P(c_i)$:

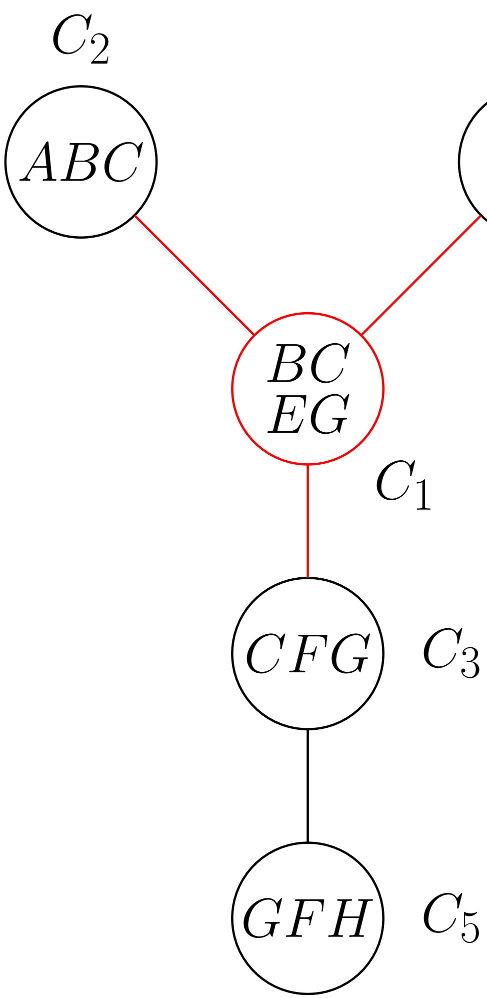
$$P(c_i) = \underbrace{\sum_{v \backslash c_i}}_{\text{Marginalization}} \underbrace{\prod_{j=1}^{m} \psi_j(c_j)}_{\text{Decomposition}}$$

$$= \psi_i(c_i) \sum_{v \backslash c_i} \prod_{i \neq j} \psi_j(c_j)$$

$$= \psi_i(c_i) \sum_{r_{1i} \cup \cdots \cup r_{qi}} \prod_{i \neq j} \psi_j(c_j)$$

$$= \psi_i(c_i) \underbrace{\left(\sum_{r_{1i}} \prod_{c_k \in \mathcal{C}_{1i}} \psi_k(c_k) \right)}_{M_{1i}(s_{ij})} \cdots \underbrace{\left(\sum_{r_{qi}} \prod_{c_k \in \mathcal{C}_{qi}} \psi_k(c_k) \right)}_{M_{qi}(s_{ij})}$$

$$= \psi_i(c_i) \prod_{j=1}^{q} M_{ji}(s_{ij})$$



Example: $P(c_1)$:

$$P(c_1) = \psi_1(c_1)M_{21}(s_{12})M_{41}(s_{14})M_{31}(s_{13})$$

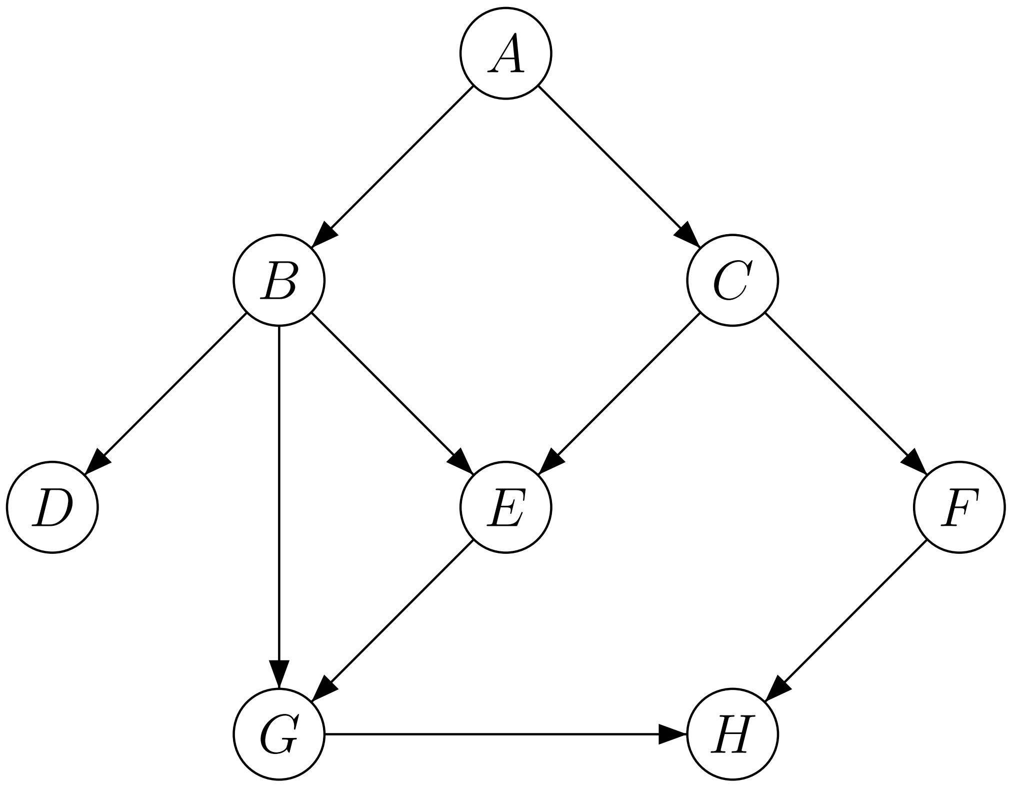
$M_{ij}(s_{ij})$ can be simplified further (without proof):

$$\begin{aligned} M_{ij}(s_{ij}) &= \sum_{r_{ij}} \prod_{c_k \in \mathcal{C}_{ij}} \psi_k(c_k) \\ &= \sum_{c_i \setminus s_{ij}} \psi_i(c_i) \prod_{k \neq j} M_{ki}(s_{ki}) \end{aligned}$$

Final Algorithm

- **Input:** Join tree $(\mathcal{C}, \Psi)$ over set of variables V and evidence $E = e$.
- **Output:** The a-posteriori probability $P(x_i \,|\, e)$ for every non-evidential X_i.
- **Initialization:** Incorporate evidence $E = e$ into potential functions.
- **Iterations:**
  1. For every clique C_i do: For every neighbor B_j of C_i do: If C_i has received all messages from the *other* neighbors, calculate and send $M_{ij}(s_{ij})$ to B_j.
  2. Repeat step 1 until no message is calculated.
  3. Calculate the joint probability distribution for every clique:
  $$P(c_i) \;\propto\; \psi_i(c_i) \prod_{j=1}^{q} M_{ji}(s_{ij})$$
  4. For every $X \in V$ calculate the a-posteriori probability:
  $$P(x_i \mid e) \;=\; \sum_{c_k \setminus x_i} P(c_k)$$
  where C_k is the smallest clique containing X_i.

Example: Putting it together

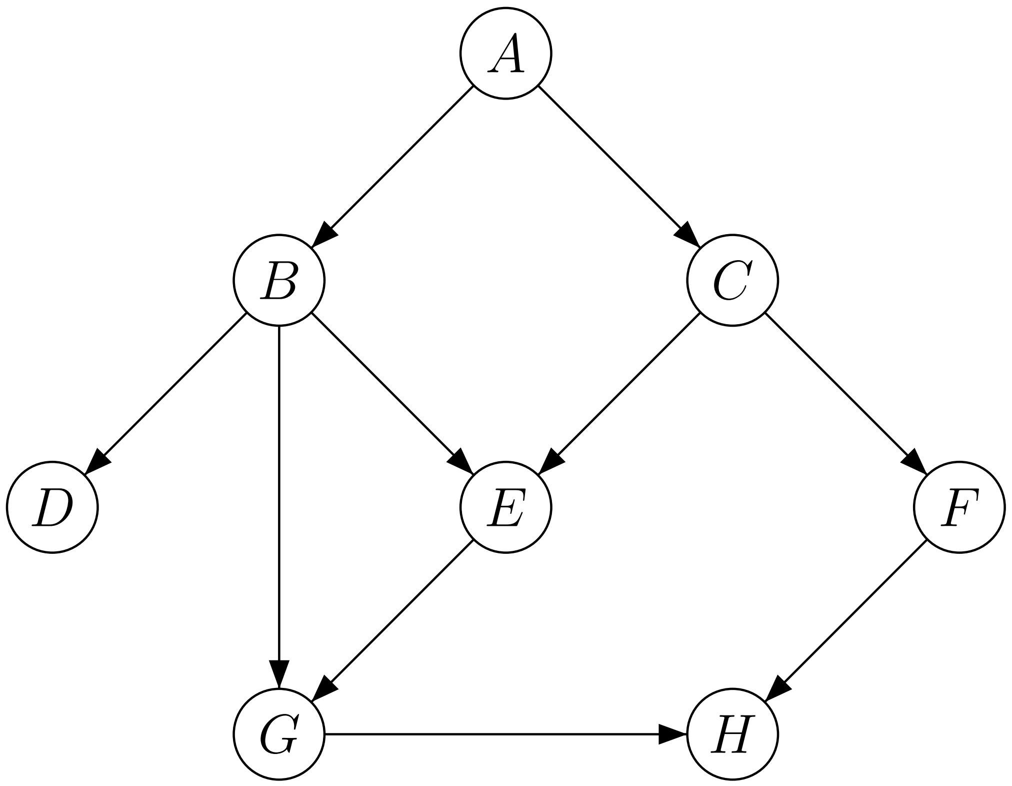


Goals: Find the marginal distributions and update them when evidence $H = h_1$ becomes known.

Steps:

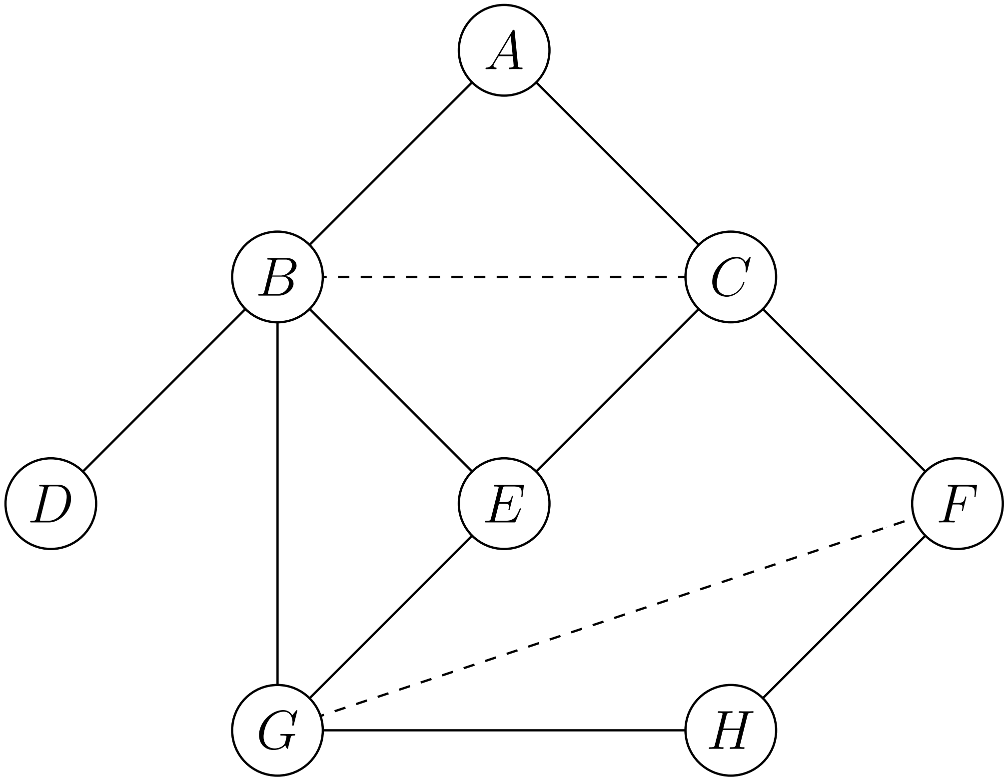
1. Transform network into join-tree.
2. Specify factor potentials.
3. Propagate "zero" evidence to obtain the marginals before evidence is present.
4. Update factor potentials w. r. t. the evidence and do another propagation run.

Example: Step 1: Find a Join-Tree



Join-Tree creation:

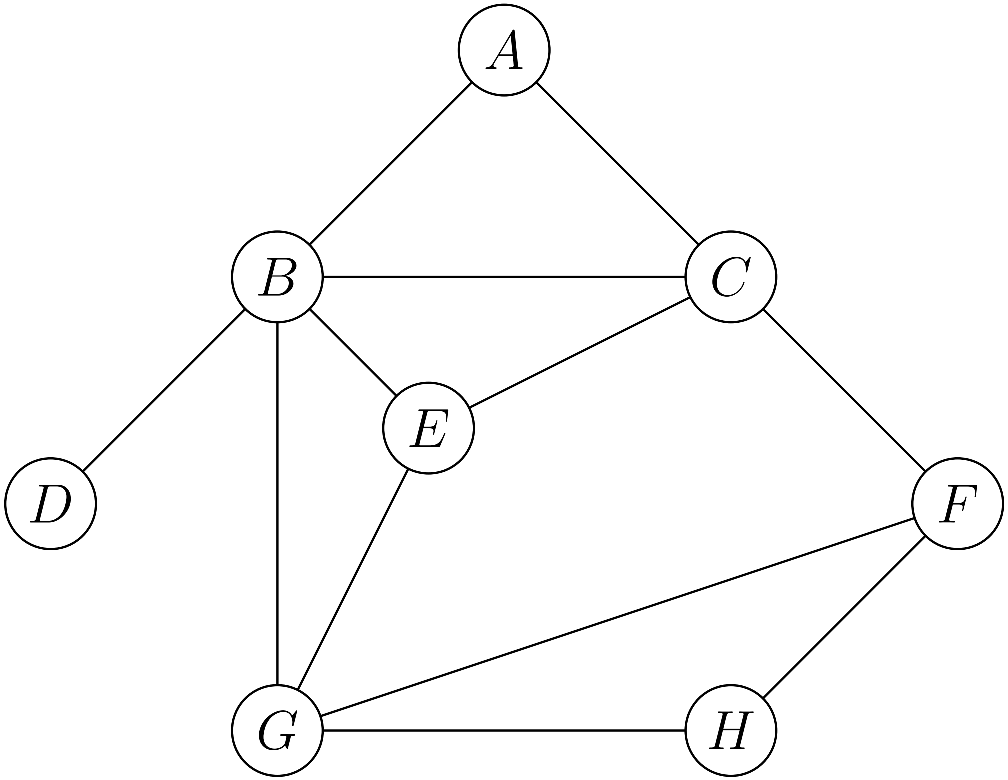
Example: Step 1: Find a Join-Tree



Join-Tree creation:

1. Moralize the graph.

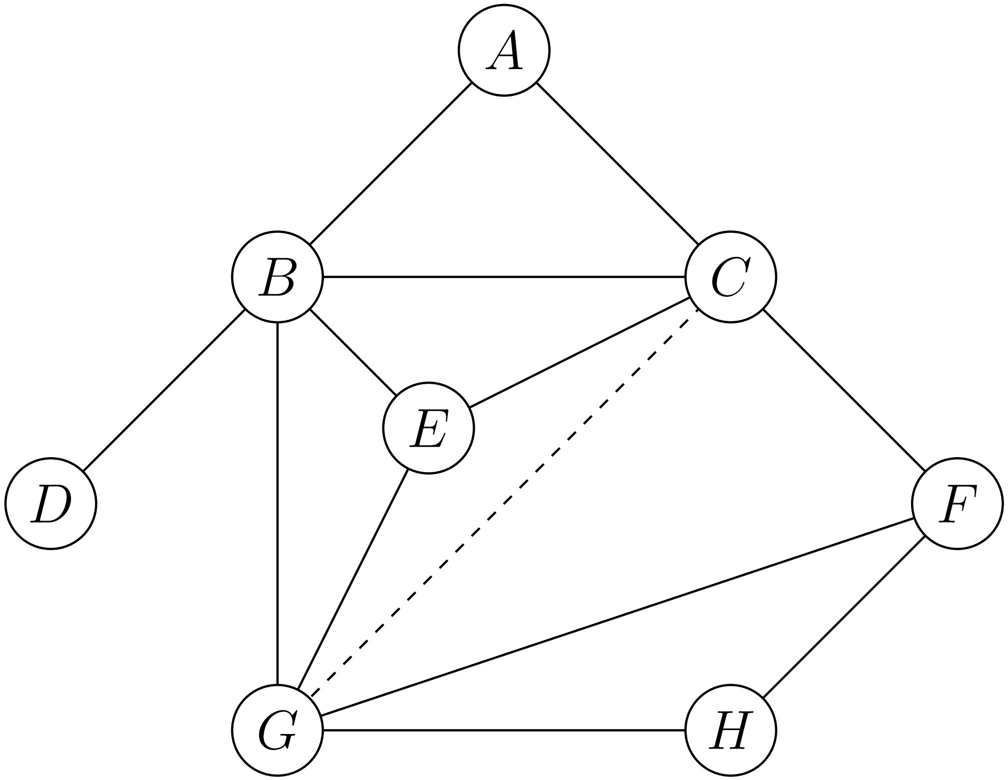
Example: Step 1: Find a Join-Tree



Join-Tree creation:

1. Moralize the graph.
2. Not yet triangulated.

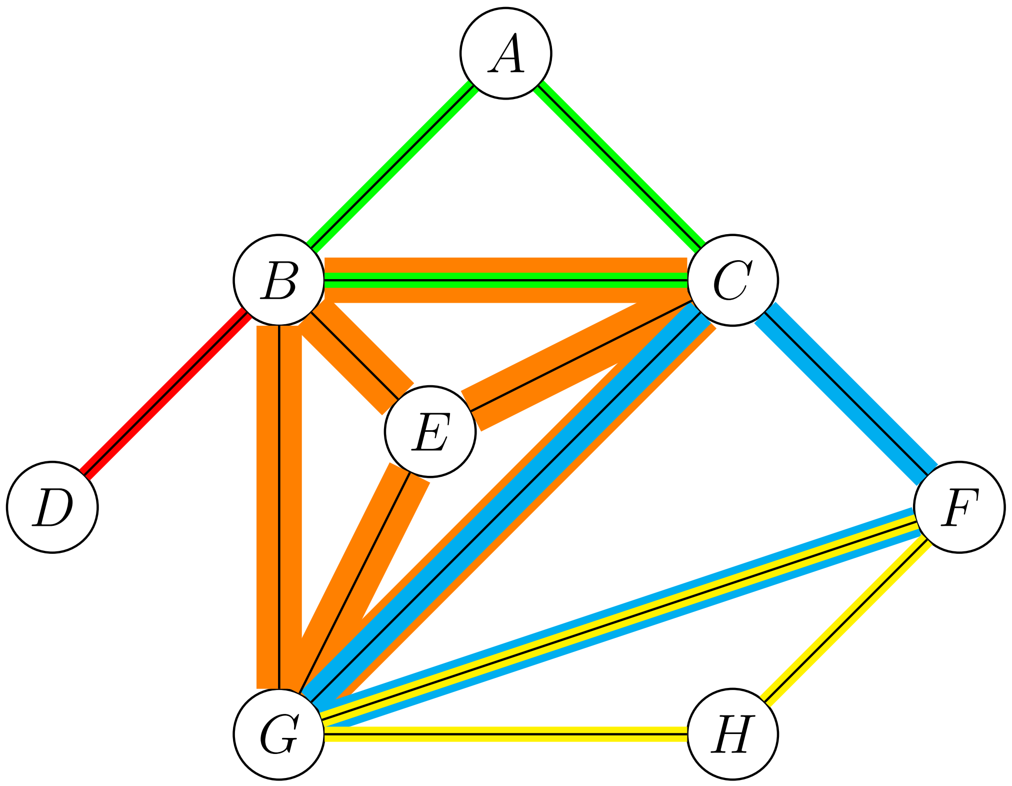
Example: Step 1: Find a Join-Tree



Join-Tree creation:

1. Moralize the graph.
2. Triangulate the graph.

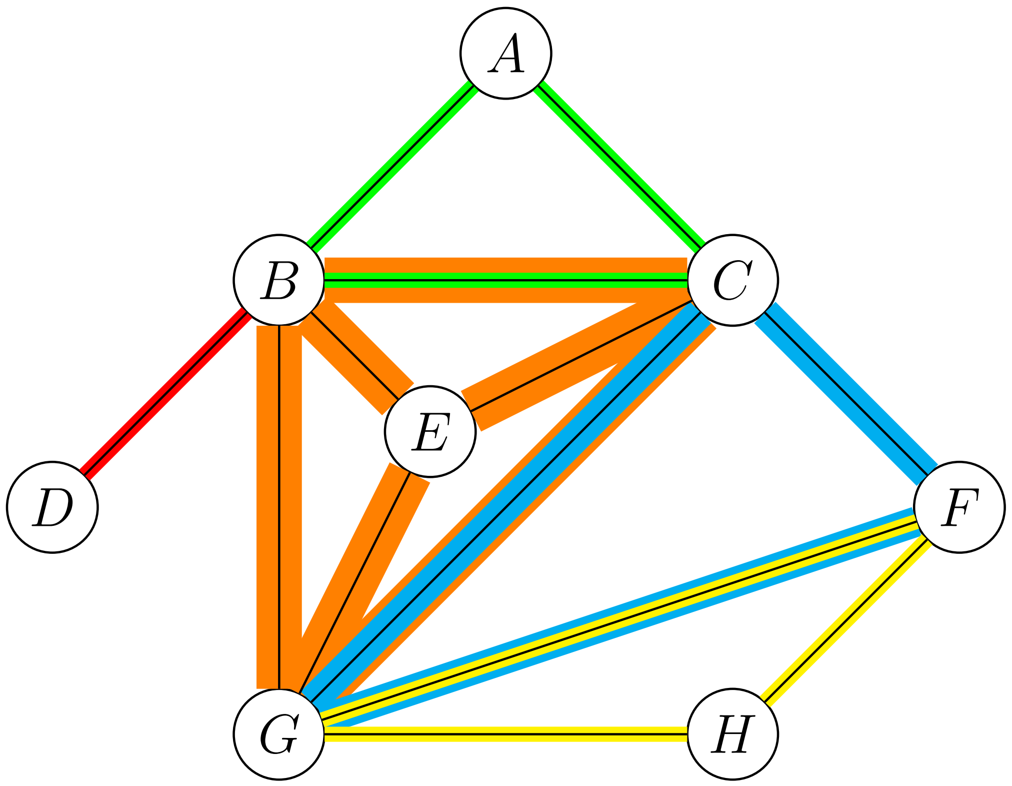
Example: Step 1: Find a Join-Tree



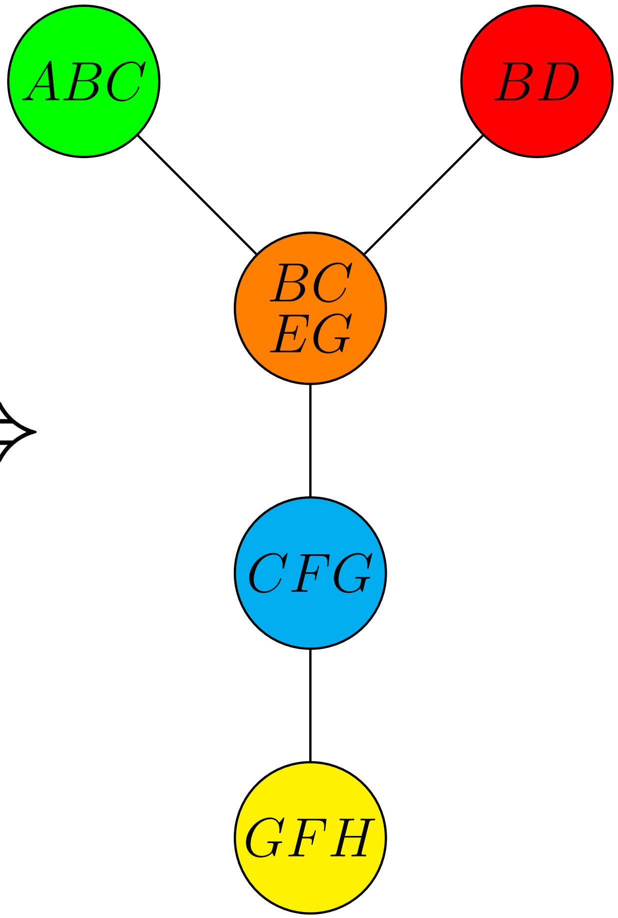
Join-Tree creation:

1. Moralize the graph.
2. Triangulate the graph.
3. Identify the maximal cliques.

Example: Step 1: Find a Join-Tree

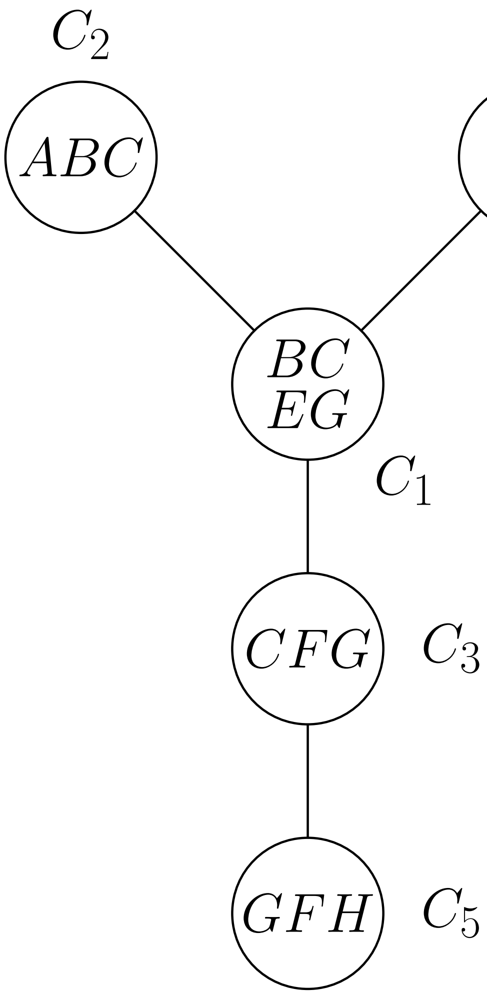


Example Bayesian network



One of the join trees

Example: Step 2: Specify the Factor Potentials



Decomposition of $P(A, B, C, D, E, F, G, H)$:

$$\begin{aligned}
P(a, b, c, d, e, f, g, h) &= \prod_{i=1}^{5} \Psi_i(c_i) \\
&= \Psi_1(b, c, e, g) \cdot \Psi_2(a, b, c) \\
&\quad \cdot \Psi_3(c, f, g) \cdot \Psi_4(b, d) \\
&\quad\quad \cdot \Psi_5(g, f, h)
\end{aligned}$$

Where to get the factor potentials from?

Example: Step 2: Specify the Factor Potentials

As long as the factor potentials multiply together as on the previous slide, we are free to choose them.

- **Option 1:** A factor potential of clique C_i is the product of all conditional probabilities of all node families properly contained in C_i:

$$\Psi_i(c_i) = 1 \cdot \prod_{\substack{\{X_i\} \cup Y_i \subseteq C_i \,\wedge \\ \text{parents}(X_i) = Y_i}} P(x_i \mid y_i)$$

  The 1 stresses that if no node family satisfies the product condition, we assign a constant 1 to the potential.

- **Option 2:** Choose potentials from the decomposition formula:

$$P\left(\bigcup_{i=1}^{n} C_i\right) = \frac{\prod_{i=1}^{n} P(C_i)}{\prod_{j=1}^{m} P(S_j)}$$

Example: Step 2: Specify the Factor Potentials

- **Option 1:** Factor potentials according to the conditional distributions of the node families of the underlying Bayesian network:

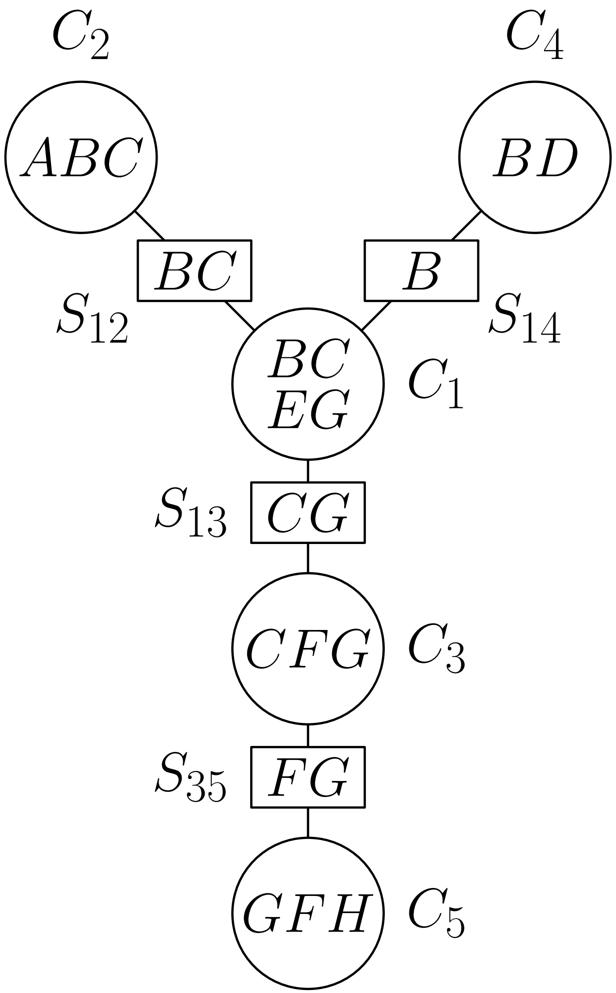
$$
\begin{aligned}
\Psi_1(b,c,e,g) &= P(e \mid b,c) \cdot P(g \mid e,b) \\
\Psi_2(a,b,c) &= P(b \mid a) \cdot P(c \mid a) \cdot P(a) \\
\Psi_3(c,f,g) &= P(f \mid c) \\
\Psi_4(b,d) &= P(d \mid b) \\
\Psi_5(g,f,h) &= P(h \mid g,f)
\end{aligned}
$$

  (This assignment of factor potentials is used in this example.)

- **Option 2:** Factor potentials chosen from the join-tree decomposition:

$$
\begin{aligned}
\Psi_1(b,c,e,g) &= P(b,e \mid c,g) \\
\Psi_2(a,b,c) &= P(a \mid b,c) \\
\Psi_3(c,f,g) &= P(c \mid f,g) \\
\Psi_4(b,d) &= P(d \mid b) \\
\Psi_5(g,f,h) &= P(h,g,f)
\end{aligned}
$$

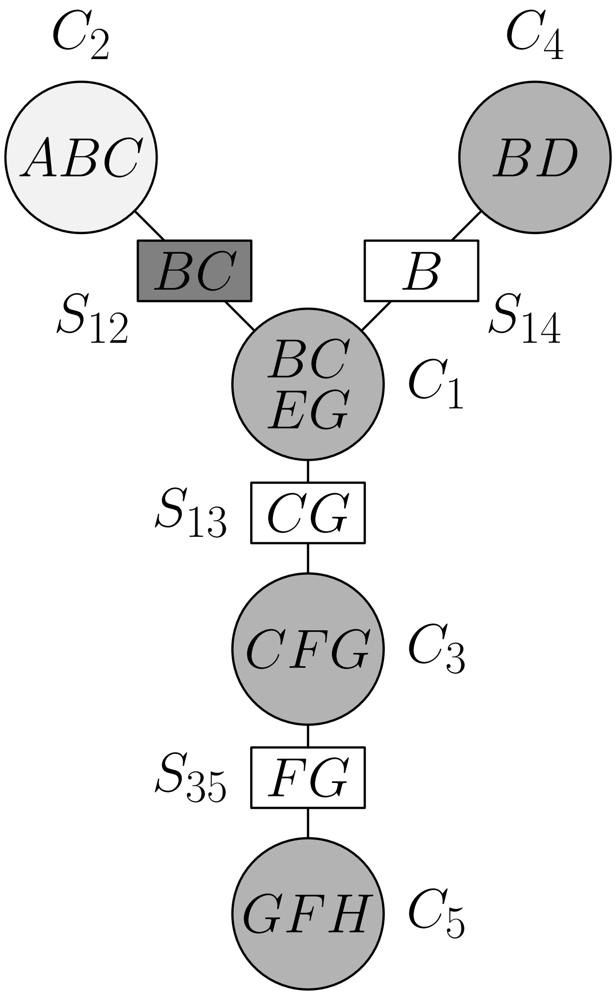
Example: Closer Look on Option 2: Separation in a Join-Tree



Encoded independence statements:

Given any separator, the variables in the cliques on one side become independent of the variables in the cliques on the other side.

Example: Closer Look on Option 2: Separation in a Join-Tree

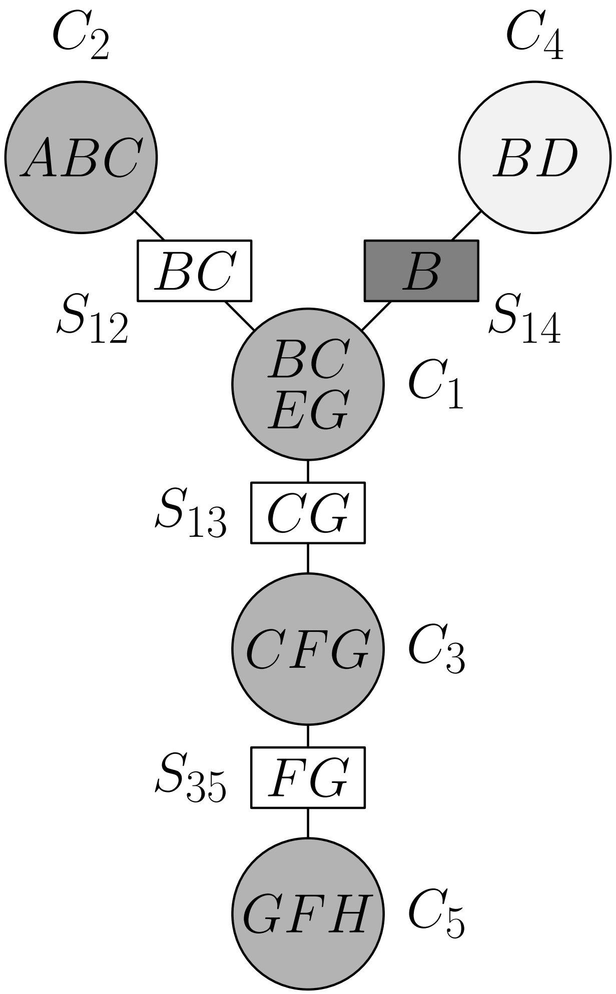


Encoded independence statements:

Given any separator, the variables in the cliques on one side become independent of the variables in the cliques on the other side.

$$A \perp\!\!\!\perp D, E, F, G, H \mid B, C$$

Example: Closer Look on Option 2: Separation in a Join-Tree

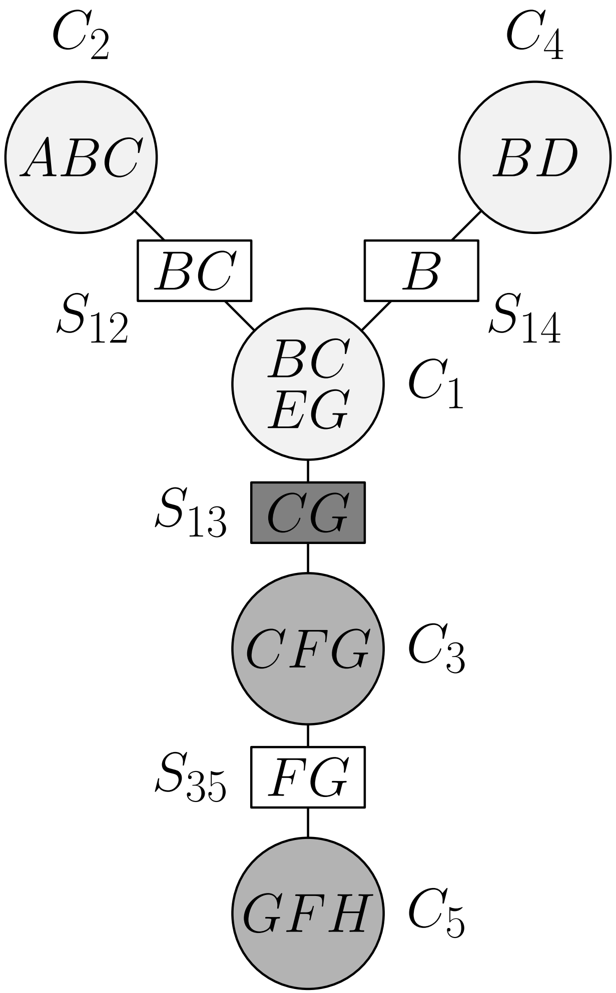


Encoded independence statements:

Given any separator, the variables in the cliques on one side become independent of the variables in the cliques on the other side.

$$A \perp\!\!\!\perp D, E, F, G, H \mid B, C$$
$$D \perp\!\!\!\perp A, C, E, F, G, H \mid B$$

Example: Closer Look on Option 2: Separation in a Join-Tree

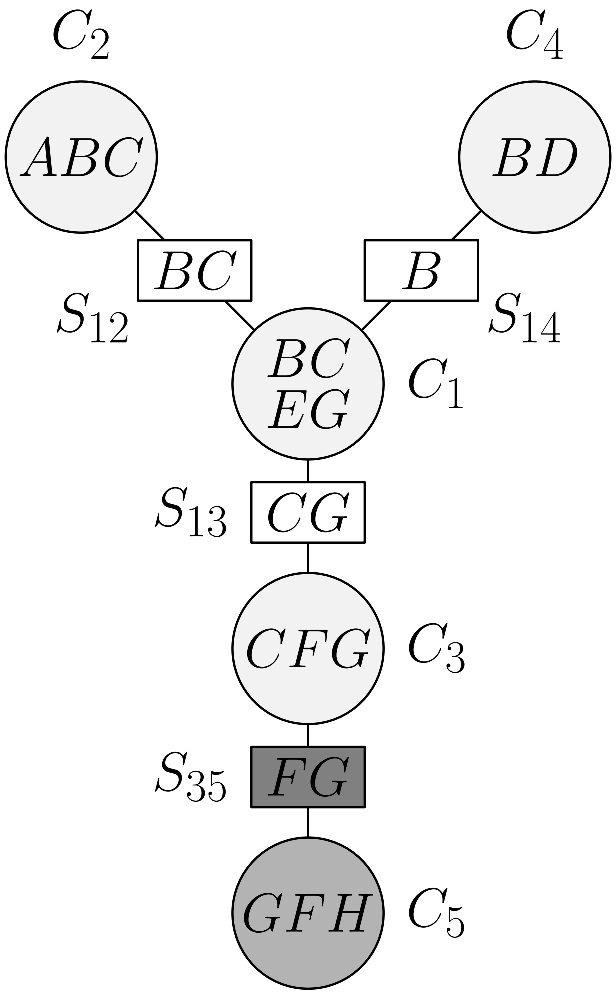


Encoded independence statements:

Given any separator, the variables in the cliques on one side become independent of the variables in the cliques on the other side.

$$A \perp\!\!\!\perp D, E, F, G, H \mid B, C$$
$$D \perp\!\!\!\perp A, C, E, F, G, H \mid B$$
$$A, B, E, D \perp\!\!\!\perp F, H \mid G, C$$

Example: Closer Look on Option 2: Separation in a Join-Tree



Encoded independence statements:

Given any separator, the variables in the cliques on one side become independent of the variables in the cliques on the other side.

$$
\begin{aligned}
A &\perp\!\!\!\perp D, E, F, G, H \mid B, C \\
D &\perp\!\!\!\perp A, C, E, F, G, H \mid B \\
A, B, E, D &\perp\!\!\!\perp F, H \mid G, C \\
H &\perp\!\!\!\perp A, B, C, D, E \mid F, G
\end{aligned}
$$

Example: Closer Look on Option 2: Decomposition

The four separation statements translate into the following independence statements:

$$
\begin{aligned}
A \perp\!\!\!\perp D, E, F, G, H \mid B, C \quad &\Leftrightarrow P(A \mid B, C, D, E, F, G, H) &= P(A \mid B, C) \\
D \perp\!\!\!\perp A, C, E, F, G, H \mid B \quad &\Rightarrow P(D \mid B, C, E, F, G, H) &= P(D \mid B) \\
A, B, E, D \perp\!\!\!\perp F, H \mid G, C \quad &\Rightarrow P(B, E \mid G, C, F, H) &= P(B, E \mid G, C) \\
H \perp\!\!\!\perp A, B, C, D, E \mid F, G \quad &\Rightarrow P(C \mid F, G, H) &= P(C \mid F, G)
\end{aligned}
$$

According to the chain rule we always have the following relation:

$$
\begin{aligned}
P(A, B, C, D, E, F, G, H) = \; & P(A \mid B, C, D, E, F, G, H) \cdot \\
& P(D \mid B, C, E, F, G, H) \cdot \\
& P(B, E \mid C, F, G, H) \cdot \\
& P(C \mid F, G, H) \cdot \\
& P(F, G, H)
\end{aligned}
$$

Example: Closer Look on Option 2: Decomposition

The four separation statements translate into the following independence statements:

$$
\begin{array}{rlr}
A \perp\!\!\!\perp D, E, F, G, H \mid B, C & \Leftrightarrow P(A \mid B, C, D, E, F, G, H) & = P(A \mid B, C) \\
D \perp\!\!\!\perp A, C, E, F, G, H \mid B & \Rightarrow P(D \mid B, C, E, F, G, H) & = P(D \mid B) \\
A, B, E, D \perp\!\!\!\perp F, H \mid G, C & \Rightarrow P(B, E \mid G, C, F, H) & = P(B, E \mid G, C) \\
H \perp\!\!\!\perp A, B, C, D, E \mid F, G & \Rightarrow P(C \mid F, G, H) & = P(C \mid F, G)
\end{array}
$$

Exploiting the above independencies yields:

$$
\begin{aligned}
P(A, B, C, D, E, F, G, H) = \; & P(A \mid B, C) \cdot \\
& P(D \mid B) \cdot \\
& P(B, E \mid C, G) \cdot \\
& P(C \mid F, G) \cdot \\
& P(F, G, H)
\end{aligned}
$$

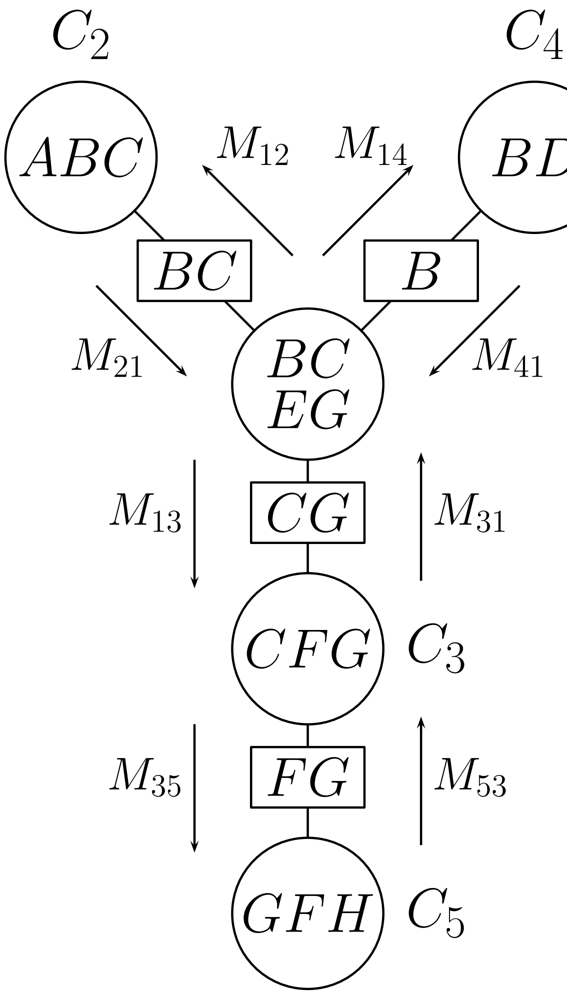
The four separation statements translate into the following independence statements:

$$
\begin{array}{rll}
A \perp\!\!\!\perp D, E, F, G, H \mid B, C & \Leftrightarrow P(A \mid B, C, D, E, F, G, H) & = P(A \mid B, C) \\
D \perp\!\!\!\perp A, C, E, F, G, H \mid B & \Rightarrow P(D \mid B, C, E, F, G, H) & = P(D \mid B) \\
A, B, E, D \perp\!\!\!\perp F, H \mid G, C & \Rightarrow P(B, E \mid G, C, F, H) & = P(B, E \mid G, C) \\
H \perp\!\!\!\perp A, B, C, D, E \mid F, G & \Rightarrow P(C \mid F, G, H) & = P(C \mid F, G)
\end{array}
$$

Getting rid of the conditions results in the final decomposition equation:

$$
\begin{aligned}
P(A, B, C, D, E, F, G, H) &= P(A \mid B, C) P(D \mid B) P(B, E \mid C, G) P(C \mid F, G) P(F, G, H) \\
&= \frac{P(A, B, C) P(D, B) P(B, E, C, G) P(C, F, G) P(F, G, H)}{P(B, C) P(B) P(C, G) P(F, G)} \\
&= \frac{P(C_1) P(C_2) P(C_3) P(C_4) P(C_5)}{P(S_{12}) P(S_{14}) P(S_{13}) P(S_{35})}
\end{aligned}
$$

Example: Step 3: Messages to be sent for Propagation



- According to the join-tree propagation algorithm, the probability distributions of all clique instantiations c_i is calculated as follows:

$$P(c_i) \;\propto\; \Psi_i(c_i) \prod_{j=1}^{q} M_{ji}(s_{ij})$$

- Spelt out for our example, we get:

$$\begin{aligned}
P(c_1) &= P(b,c,e,g) &= \Psi_1(b,c,e,g) &\cdot M_{21}(b,c) \cdot M_{31}(c,g) \cdot M_{41}(b) \\
P(c_2) &= P(a,b,c) &\propto \Psi_2(a,b,c) &\cdot M_{12}(b,c) \\
P(c_3) &= P(c,f,g) &\propto \Psi_3(c,f,g) &\cdot M_{13}(c,g) \cdot M_{53}(f,g) \\
P(c_4) &= P(b,d) &\propto \Psi_4(b,d) &\cdot M_{14}(b) \\
P(c_5) &= P(f,g,h) &\propto \Psi_5(f,g,h) &\cdot M_{35}(f,g)
\end{aligned}$$

- The $\propto$-symbol indicates that the right-hand side may not add up to one. In that case we just normalize.

Example: Step 3: Message Computation Order

- The structure of the join-tree imposes a partial ordering according to which the messages need to be computed:

$$M_{41}(b) = \sum_{d} \Psi_4(b, d)$$

$$M_{53}(f, g) = \sum_{h} \Psi_5(f, g, h)$$

$$M_{21}(b, c) = \sum_{a} \Psi_2(a, b, c)$$

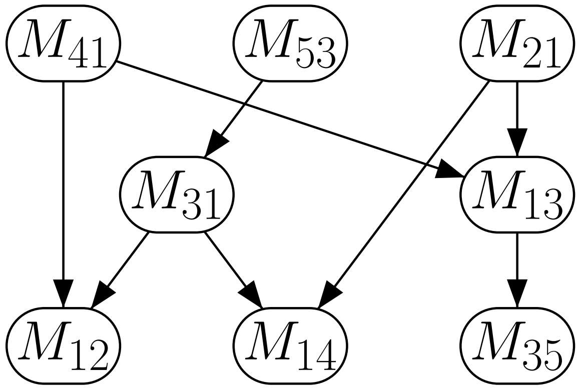
$$M_{31}(c, g) = \sum_{f} \Psi_3(c, f, g) M_{53}(f, g)$$

$$M_{13}(c, g) = \sum_{b,e} \Psi_1(b, c, e, g) M_{21}(b, c) M_{41}(b)$$

$$M_{12}(b, c) = \sum_{e,g} \Psi_2(b, c, e, g) M_{31}(c, g) M_{41}(b)$$

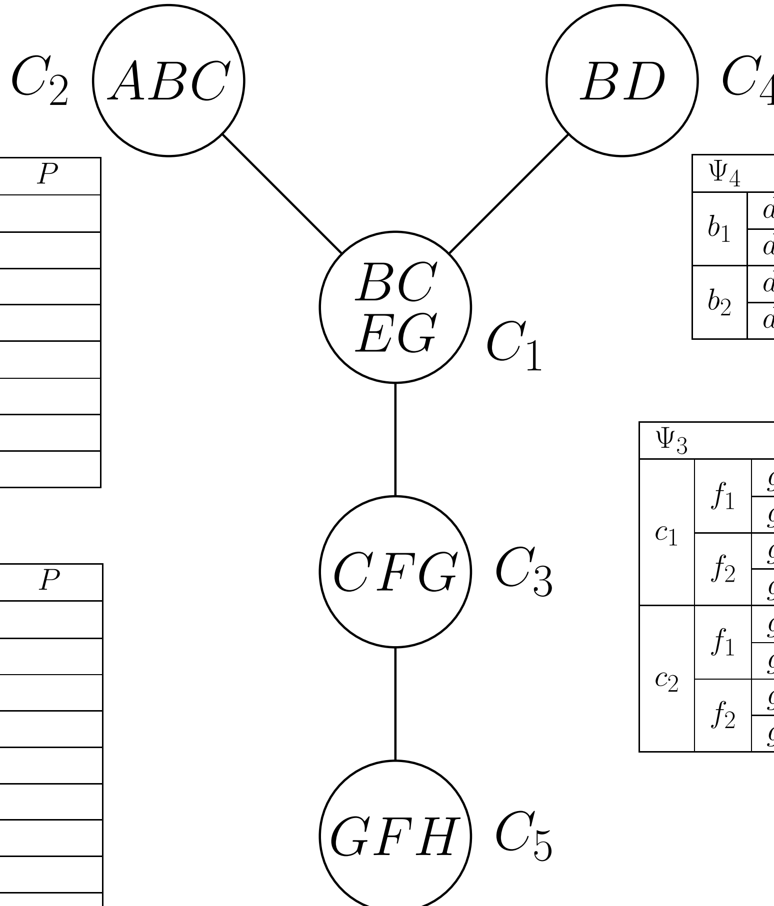
$$M_{14}(b) = \sum_{c,e,g} \Psi_1(b, c, e, g) M_{21}(b, c) M_{31}(c, g)$$

$$M_{35}(f, g) = \sum_{c} \Psi_3(c, f, g) M_{13}(c, g)$$



Arrows represent is-needed-for relations. Messages on the same level can be computed in any order. Messages are computed level-wise from top to bottom.

Example: Step 3: Initialization (Potential Layouts)



Ψ_2				P
a_1	b_1	c_1		
		c_2		
	b_2	c_1		
		c_2		
a_2	b_1	c_1		
		c_2		
	b_2	c_1		
		c_2		

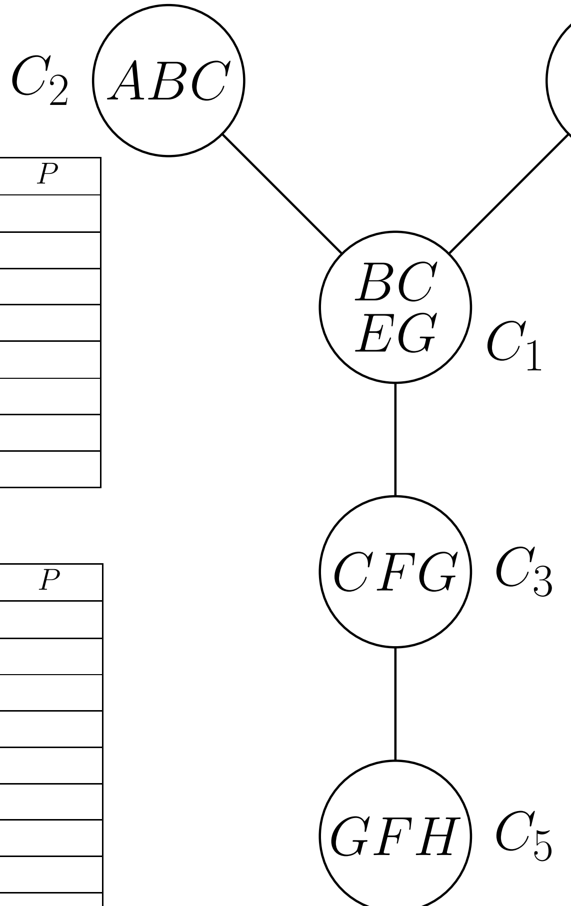
Ψ_4			P
b_1	d_1		
	d_2		
b_2	d_1		
	d_2		

Ψ_1					P
b_1	c_1	e_1	g_1		
			g_2		
		e_2	g_1		
			g_2		
	c_2	e_1	g_1		
			g_2		
		e_2	g_1		
			g_2		
b_2	c_1	e_1	g_1		
			g_2		
		e_2	g_1		
			g_2		
	c_2	e_1	g_1		
			g_2		
		e_2	g_1		
			g_2		

Ψ_3				P
c_1	f_1	g_1		
		g_2		
	f_2	g_1		
		g_2		
c_2	f_1	g_1		
		g_2		
	f_2	g_1		
		g_2		

Ψ_5				P
f_1	g_1	h_1		
		h_2		
	g_2	h_1		
		h_2		
f_2	g_1	h_1		
		h_2		
	g_2	h_1		
		h_2		

Example: Step 3: Initialization (Potential Values)



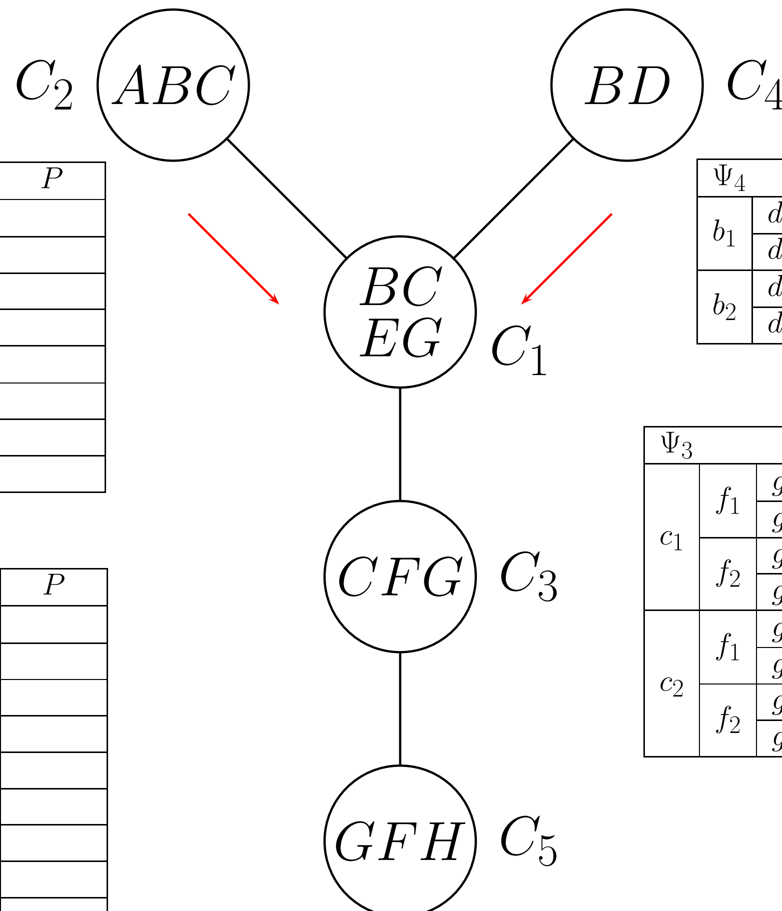
Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0.8	
	g_2	h_1	0.5	
		h_2	0.5	
f_2	g_1	h_1	0.4	
		h_2	0.6	
	g_2	h_1	0.7	
		h_2	0.3	



$$M_{21} = \begin{array}{c} \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{array} \\ \left(\begin{array}{cccc} 0.06, & 0.10, & 0.40, & 0.44 \end{array} \right) \end{array}$$

$$M_{41} = \begin{array}{c} \begin{array}{cc} b_1 & b_2 \end{array} \\ \left(\begin{array}{cc} 1, & 1 \end{array} \right) \end{array}$$

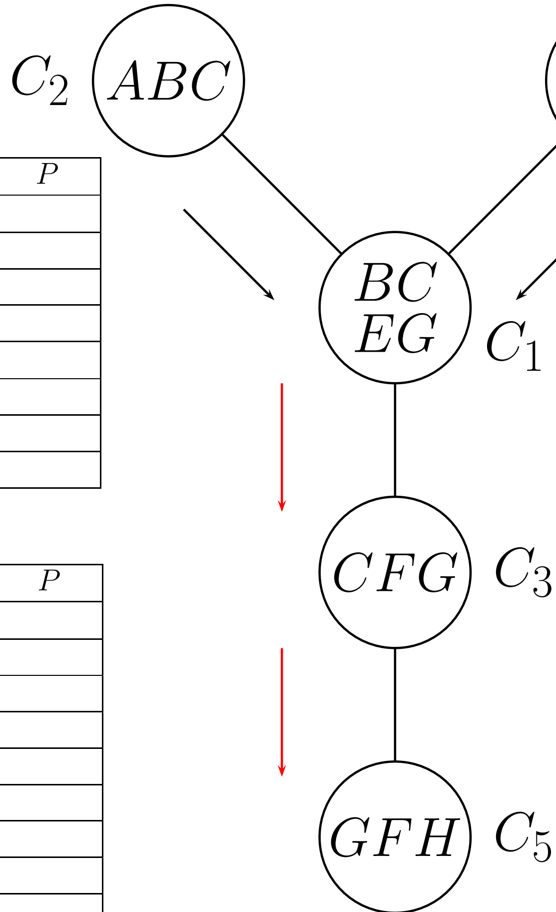
Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0.8	
	g_2	h_1	0.5	
		h_2	0.5	
f_2	g_1	h_1	0.4	
		h_2	0.6	
	g_2	h_1	0.7	
		h_2	0.3	



$$M_{21} = \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \\ \end{array}$$
$$M_{21} = (0.06, 0.10, 0.40, 0.44)$$

$$M_{41} = \begin{array}{cc} b_1 & b_2 \\ \end{array}$$
$$M_{41} = (1, 1)$$

$$M_{13} = \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \\ \end{array}$$
$$M_{13} = (0.254, 0.206, 0.290, 0.250)$$

$$M_{35} = \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \\ \end{array}$$
$$M_{35} = (0.14, 0.12, 0.40, 0.33)$$

Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

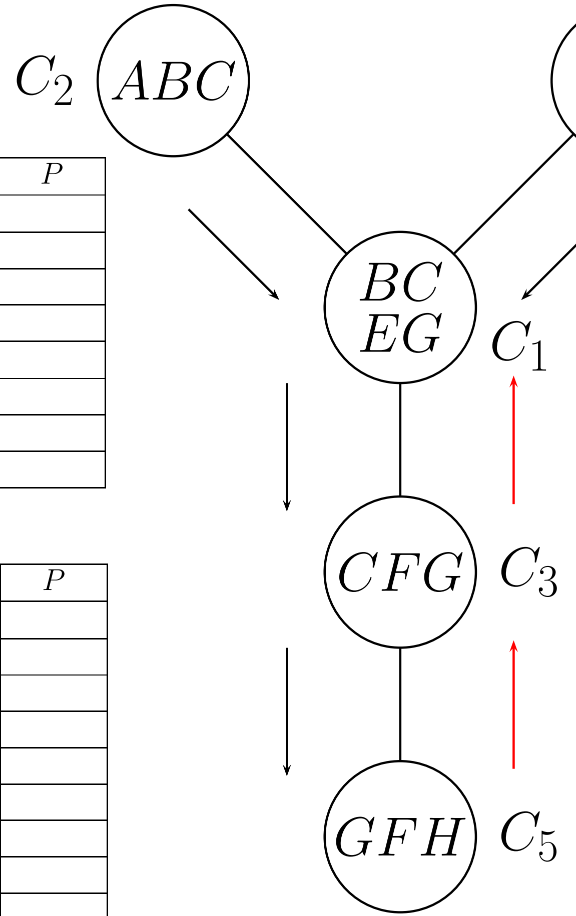
Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0.8	
	g_2	h_1	0.5	
		h_2	0.5	
f_2	g_1	h_1	0.4	
		h_2	0.6	
	g_2	h_1	0.7	
		h_2	0.3	

Example: Step 3: Initialization (Sending Messages)



Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0.8	
	g_2	h_1	0.5	
		h_2	0.5	
f_2	g_1	h_1	0.4	
		h_2	0.6	
	g_2	h_1	0.7	
		h_2	0.3	

$$M_{21} = \begin{array}{c} \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{array} \\ (0.06,\ 0.10,\ 0.40,\ 0.44) \end{array}$$

$$M_{41} = \begin{array}{c} \begin{array}{cc} b_1 & b_2 \end{array} \\ (1,\ 1) \end{array}$$

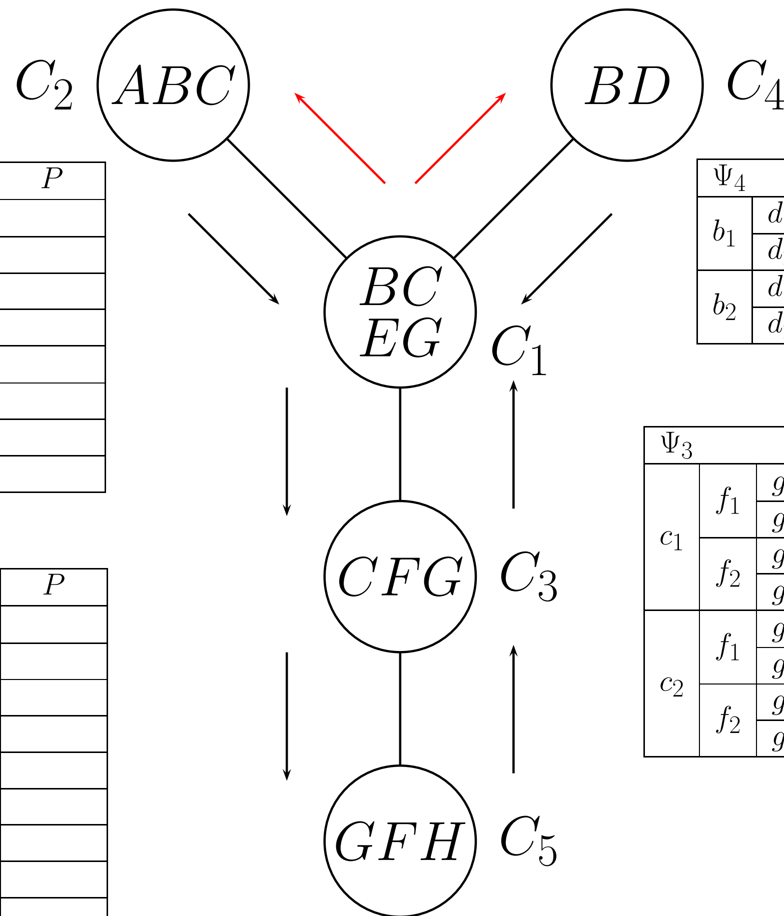
$$M_{13} = \begin{array}{c} \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{array} \\ (0.254,\ 0.206,\ 0.290,\ 0.250) \end{array}$$

$$M_{35} = \begin{array}{c} \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \\ (0.14,\ 0.12,\ 0.40,\ 0.33) \end{array}$$

$$M_{53} = \begin{array}{c} \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \\ (1,\ 1,\ 1,\ 1) \end{array}$$

$$M_{31} = \begin{array}{c} \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{array} \\ (1,\ 1,\ 1,\ 1) \end{array}$$

Example: Step 3: Initialization (Sending Messages)



Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0.8	
	g_2	h_1	0.5	
		h_2	0.5	
f_2	g_1	h_1	0.4	
		h_2	0.6	
	g_2	h_1	0.7	
		h_2	0.3	

$$M_{21} = \begin{array}{c} \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{array} \\ \left(\begin{array}{cccc} 0.06, & 0.10, & 0.40, & 0.44 \end{array}\right) \end{array}$$

$$M_{41} = \begin{array}{c} \begin{array}{cc} b_1 & b_2 \end{array} \\ \left(\begin{array}{cc} 1, & 1 \end{array}\right) \end{array}$$

$$M_{13} = \begin{array}{c} \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 0.254, & 0.206, & 0.290, & 0.250 \end{array}\right) \end{array}$$

$$M_{35} = \begin{array}{c} \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 0.14, & 0.12, & 0.40, & 0.33 \end{array}\right) \end{array}$$

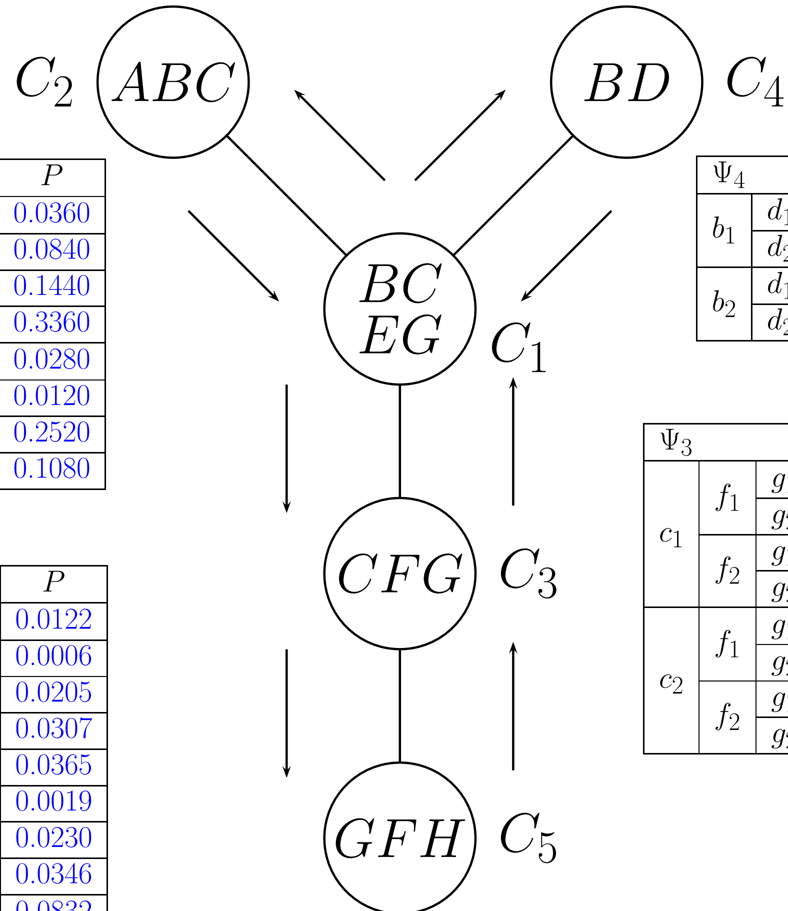
$$M_{53} = \begin{array}{c} \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 1, & 1, & 1, & 1 \end{array}\right) \end{array}$$

$$M_{31} = \begin{array}{c} \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 1, & 1, & 1, & 1 \end{array}\right) \end{array}$$

$$M_{12} = \begin{array}{c} \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{array} \\ \left(\begin{array}{cccc} 1, & 1, & 1, & 1 \end{array}\right) \end{array}$$

$$M_{14} = \begin{array}{c} \begin{array}{cc} b_1 & b_2 \end{array} \\ \left(\begin{array}{cc} 0.16, & 0.84 \end{array}\right) \end{array}$$

Example: Step 3: Initialization Complete



Ψ_2				P
a_1	b_1	c_1	0.036	0.0360
		c_2	0.084	0.0840
	b_2	c_1	0.144	0.1440
		c_2	0.336	0.3360
a_2	b_1	c_1	0.028	0.0280
		c_2	0.012	0.0120
	b_2	c_1	0.252	0.2520
		c_2	0.108	0.1080

Ψ_4			P
b_1	d_1	0.4	0.0640
	d_2	0.6	0.0960
b_2	d_1	0.7	0.5880
	d_2	0.3	0.2520

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	0.0122
			g_2	0.010	0.0006
		e_2	g_1	0.320	0.0205
			g_2	0.480	0.0307
	c_2	e_1	g_1	0.380	0.0365
			g_2	0.020	0.0019
		e_2	g_1	0.240	0.0230
			g_2	0.360	0.0346
b_2	c_1	e_1	g_1	0.210	0.0832
			g_2	0.090	0.0356
		e_2	g_1	0.350	0.1386
			g_2	0.350	0.1386
	c_2	e_1	g_1	0.070	0.0311
			g_2	0.030	0.0133
		e_2	g_1	0.450	0.1998
			g_2	0.450	0.1998

Ψ_3				P
c_1	f_1	g_1	0.1	0.0254
		g_2	0.1	0.0206
	f_2	g_1	0.9	0.2290
		g_2	0.9	0.1850
c_2	f_1	g_1	0.4	0.1162
		g_2	0.4	0.0998
	f_2	g_1	0.6	0.1742
		g_2	0.6	0.1498

Ψ_5				P
f_1	g_1	h_1	0.2	0.0283
		h_2	0.8	0.1133
	g_2	h_1	0.5	0.0602
		h_2	0.5	0.0602
f_2	g_1	h_1	0.4	0.1613
		h_2	0.6	0.2419
	g_2	h_1	0.7	0.2344
		h_2	0.3	0.1004

$$M_{21} = \left(\overset{b_1,c_1}{0.06}, \overset{b_1,c_2}{0.10}, \overset{b_2,c_1}{0.40}, \overset{b_2,c_2}{0.44}\right)$$

$$M_{41} = \left(\overset{b_1}{1}, \overset{b_2}{1}\right)$$

$$M_{13} = \left(\overset{c_1,g_1}{0.254}, \overset{c_1,g_2}{0.206}, \overset{c_2,g_1}{0.290}, \overset{c_2,g_2}{0.250}\right)$$

$$M_{35} = \left(\overset{f_1,g_1}{0.14}, \overset{f_1,g_2}{0.12}, \overset{f_2,g_1}{0.40}, \overset{f_2,g_2}{0.33}\right)$$

$$M_{53} = \left(\overset{f_1,g_1}{1}, \overset{f_1,g_2}{1}, \overset{f_2,g_1}{1}, \overset{f_2,g_2}{1}\right)$$

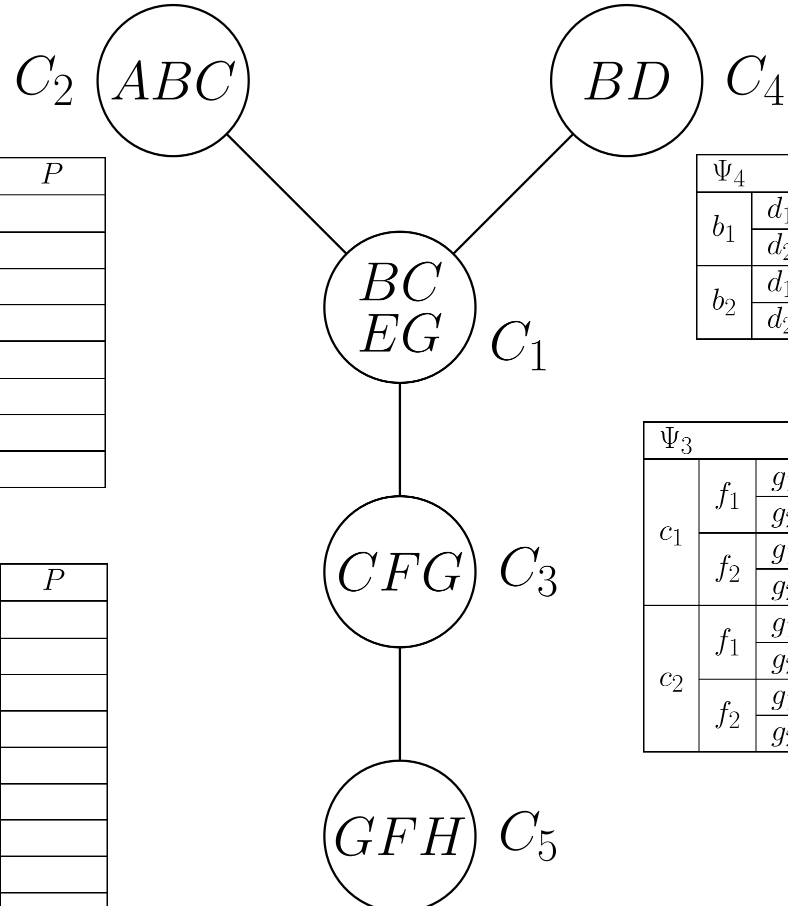
$$M_{31} = \left(\overset{c_1,g_1}{1}, \overset{c_1,g_2}{1}, \overset{c_2,g_1}{1}, \overset{c_2,g_2}{1}\right)$$

$$M_{12} = \left(\overset{b_1,c_1}{1}, \overset{b_1,c_2}{1}, \overset{b_2,c_1}{1}, \overset{b_2,c_2}{1}\right)$$

$$M_{14} = \left(\overset{b_1}{0.16}, \overset{b_2}{0.84}\right)$$

P	A	B	C	D	E	F	G	H
$\cdot_1$	**0.6000**	**0.1600**	**0.4600**	**0.6520**	**0.2144**	**0.2620**	**0.5448**	**0.4842**
$\cdot_2$	**0.4000**	**0.8400**	**0.4500**	**0.3480**	**0.7856**	**0.7380**	**0.4552**	**0.5158**

Example: Step 4: Evidence $H = h_1$ (Altering Potentials)



Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

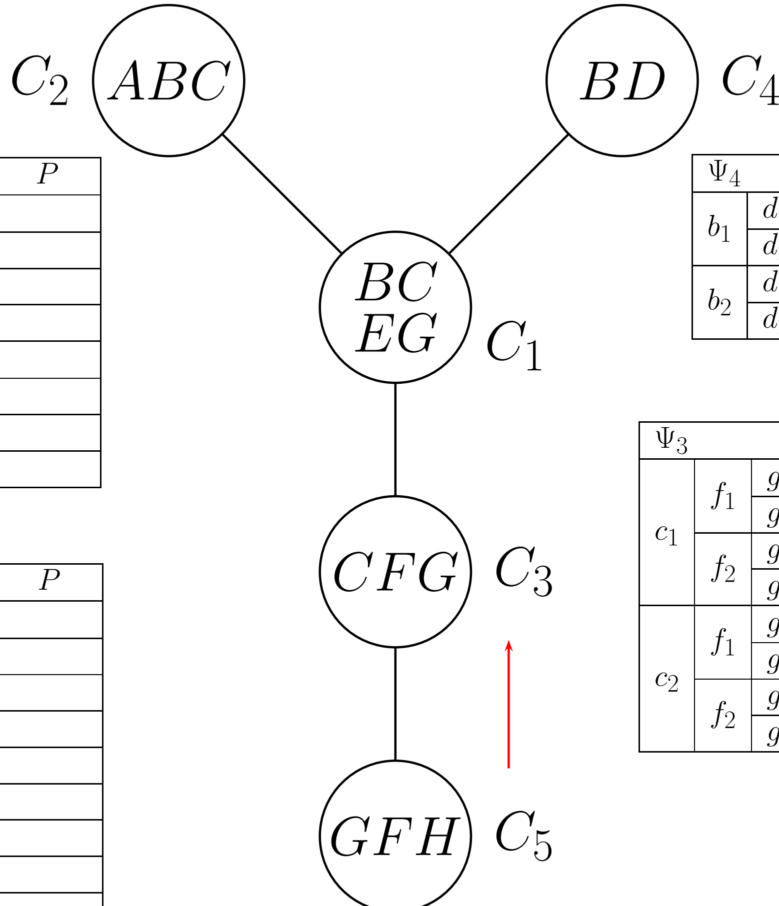
Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	**0**	
	g_2	h_1	0.5	
		h_2	**0**	
f_2	g_1	h_1	0.4	
		h_2	**0**	
	g_2	h_1	0.7	
		h_2	**0**	

Example: Step 4: Evidence $H = h_1$ (Sending Messages)

$$M_{53} = \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 0.2\,, & 0.5\,, & 0.4\,, & 0.7 \end{array} \right)$$



Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

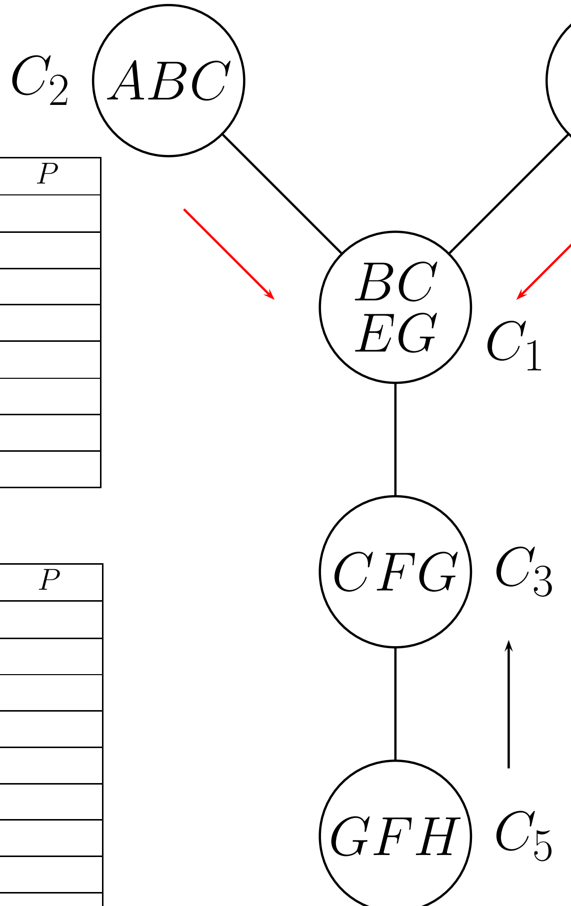
Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0	
	g_2	h_1	0.5	
		h_2	0	
f_2	g_1	h_1	0.4	
		h_2	0	
	g_2	h_1	0.7	
		h_2	0	

Example: Step 4: Evidence $H = h_1$ (Sending Messages)



$$M_{53} = \begin{array}{c} \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 0.2\,, & 0.5\,, & 0.4\,, & 0.7 \end{array}\right) \end{array}$$

$$M_{21} = \begin{array}{c} \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{array} \\ \left(\begin{array}{cccc} 0.06, & 0.10, & 0.40, & 0.44 \end{array}\right) \end{array}$$

$$M_{41} = \begin{array}{c} \begin{array}{cc} b_1 & b_2 \end{array} \\ \left(\begin{array}{cc} 1, & 1 \end{array}\right) \end{array}$$

Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

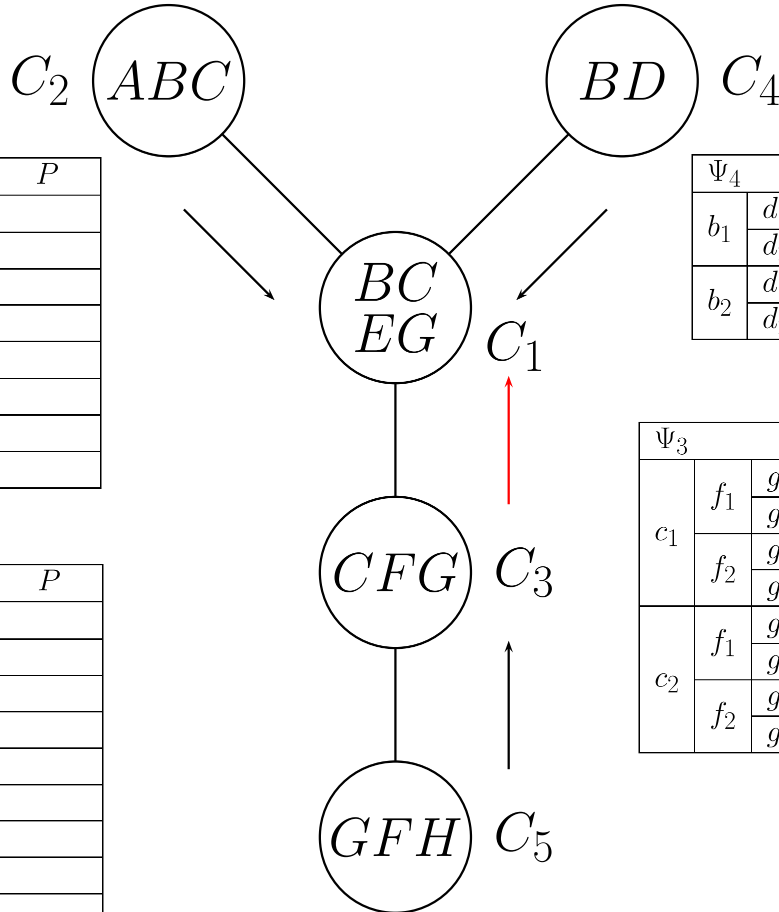
Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0	
	g_2	h_1	0.5	
		h_2	0	
f_2	g_1	h_1	0.4	
		h_2	0	
	g_2	h_1	0.7	
		h_2	0	

Example: Step 4: Evidence $H = h_1$ (Sending Messages)



Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0	
	g_2	h_1	0.5	
		h_2	0	
f_2	g_1	h_1	0.4	
		h_2	0	
	g_2	h_1	0.7	
		h_2	0	

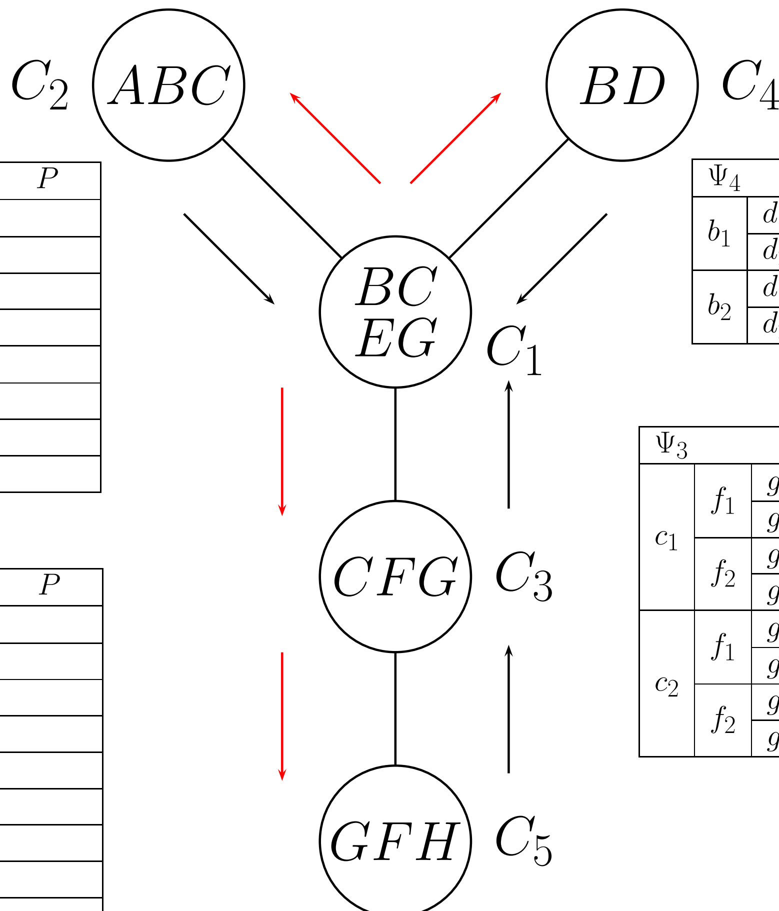
$$M_{53} = \begin{array}{c} \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 0.2\,, & 0.5\,, & 0.4\,, & 0.7 \end{array}\right) \end{array}$$

$$M_{21} = \begin{array}{c} \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{array} \\ \left(\begin{array}{cccc} 0.06, & 0.10, & 0.40, & 0.44 \end{array}\right) \end{array}$$

$$M_{41} = \begin{array}{c} \begin{array}{cc} b_1 & b_2 \end{array} \\ \left(\begin{array}{cc} 1\,, & 1 \end{array}\right) \end{array}$$

$$M_{31} = \begin{array}{c} \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 0.38, & 0.68, & 0.32, & 0.62 \end{array}\right) \end{array}$$

Example: Step 4: Evidence $H = h_1$ (Sending Messages)



Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0	
	g_2	h_1	0.5	
		h_2	0	
f_2	g_1	h_1	0.4	
		h_2	0	
	g_2	h_1	0.7	
		h_2	0	

$$M_{53} = \left(\begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \\ 0.2, & 0.5, & 0.4, & 0.7 \end{array}\right)$$

$$M_{21} = \left(\begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \\ 0.06, & 0.10, & 0.40, & 0.44 \end{array}\right)$$

$$M_{41} = \left(\begin{array}{cc} b_1 & b_2 \\ 1, & 1 \end{array}\right)$$

$$M_{31} = \left(\begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \\ 0.38, & 0.68, & 0.32, & 0.62 \end{array}\right)$$

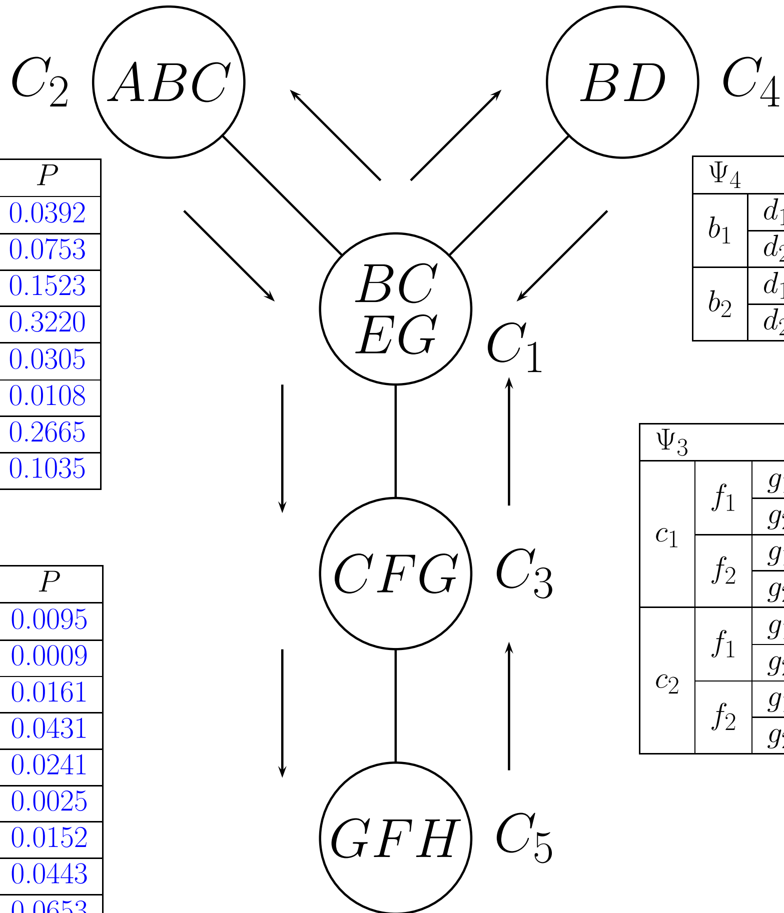
$$M_{12} = \left(\begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \\ 0.527, & 0.434, & 0.512, & 0.464 \end{array}\right)$$

$$M_{14} = \left(\begin{array}{cc} b_1 & b_2 \\ 0.075, & 0.409 \end{array}\right)$$

$$M_{13} = \left(\begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \\ 0.254, & 0.206, & 0.290, & 0.250 \end{array}\right)$$

$$M_{35} = \left(\begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \\ 0.14, & 0.12, & 0.40, & 0.33 \end{array}\right)$$

Example: Step 4: Evidence $H = h_1$ Incorporated



Ψ_2				P
a_1	b_1	c_1	0.036	0.0392
		c_2	0.084	0.0753
	b_2	c_1	0.144	0.1523
		c_2	0.336	0.3220
a_2	b_1	c_1	0.028	0.0305
		c_2	0.012	0.0108
	b_2	c_1	0.252	0.2665
		c_2	0.108	0.1035

Ψ_4			P
b_1	d_1	0.4	0.0623
	d_2	0.6	0.0934
b_2	d_1	0.7	0.5910
	d_2	0.3	0.2533

Ψ_3				P
c_1	f_1	g_1	0.1	0.0105
		g_2	0.1	0.0212
	f_2	g_1	0.9	0.1892
		g_2	0.9	0.2675
c_2	f_1	g_1	0.4	0.0480
		g_2	0.4	0.1031
	f_2	g_1	0.6	0.1440
		g_2	0.6	0.2165

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	0.0095
			g_2	0.010	0.0009
		e_2	g_1	0.320	0.0161
			g_2	0.480	0.0431
	c_2	e_1	g_1	0.380	0.0241
			g_2	0.020	0.0025
		e_2	g_1	0.240	0.0152
			g_2	0.360	0.0443
b_2	c_1	e_1	g_1	0.210	0.0653
			g_2	0.090	0.0501
		e_2	g_1	0.350	0.1088
			g_2	0.350	0.1947
	c_2	e_1	g_1	0.070	0.0205
			g_2	0.030	0.0171
		e_2	g_1	0.450	0.1321
			g_2	0.450	0.2559

Ψ_5				P
f_1	g_1	h_1	0.2	0.0585
		h_2	0	0
	g_2	h_1	0.5	0.1243
		h_2	0	0
f_2	g_1	h_1	0.4	0.3331
		h_2	0	0
	g_2	h_1	0.7	0.4841
		h_2	0	0

$$M_{53} = \begin{array}{c} \begin{matrix} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{matrix} \\ \begin{pmatrix} 0.2, & 0.5, & 0.4, & 0.7 \end{pmatrix} \end{array}$$

$$M_{21} = \begin{array}{c} \begin{matrix} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{matrix} \\ \begin{pmatrix} 0.06, & 0.10, & 0.40, & 0.44 \end{pmatrix} \end{array}$$

$$M_{41} = \begin{array}{c} \begin{matrix} b_1 & b_2 \end{matrix} \\ \begin{pmatrix} 1, & 1 \end{pmatrix} \end{array}$$

$$M_{31} = \begin{array}{c} \begin{matrix} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{matrix} \\ \begin{pmatrix} 0.38, & 0.68, & 0.32, & 0.62 \end{pmatrix} \end{array}$$

$$M_{12} = \begin{array}{c} \begin{matrix} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{matrix} \\ \begin{pmatrix} 0.527, & 0.434, & 0.512, & 0.464 \end{pmatrix} \end{array}$$

$$M_{14} = \begin{array}{c} \begin{matrix} b_1 & b_2 \end{matrix} \\ \begin{pmatrix} 0.075, & 0.409 \end{pmatrix} \end{array}$$

$$M_{13} = \begin{array}{c} \begin{matrix} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{matrix} \\ \begin{pmatrix} 0.254, & 0.206, & 0.290, & 0.250 \end{pmatrix} \end{array}$$

$$M_{35} = \begin{array}{c} \begin{matrix} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{matrix} \\ \begin{pmatrix} 0.14, & 0.12, & 0.40, & 0.33 \end{pmatrix} \end{array}$$

P	A	B	C	D	E	F	G	H
$\cdot_1$	**0.5888**	**0.1557**	**0.4884**	**0.6533**	**0.1899**	**0.1828**	**0.3916**	**1.0000**
$\cdot_2$	**0.4112**	**0.8443**	**0.5116**	**0.3467**	**0.8101**	**0.8172**	**0.6084**	**0.0000**

Manual creation of a reasoning system based on a graphical model:

causal model of given domain

↓ **heuristics!**

conditional independence graph

↓ formally provable

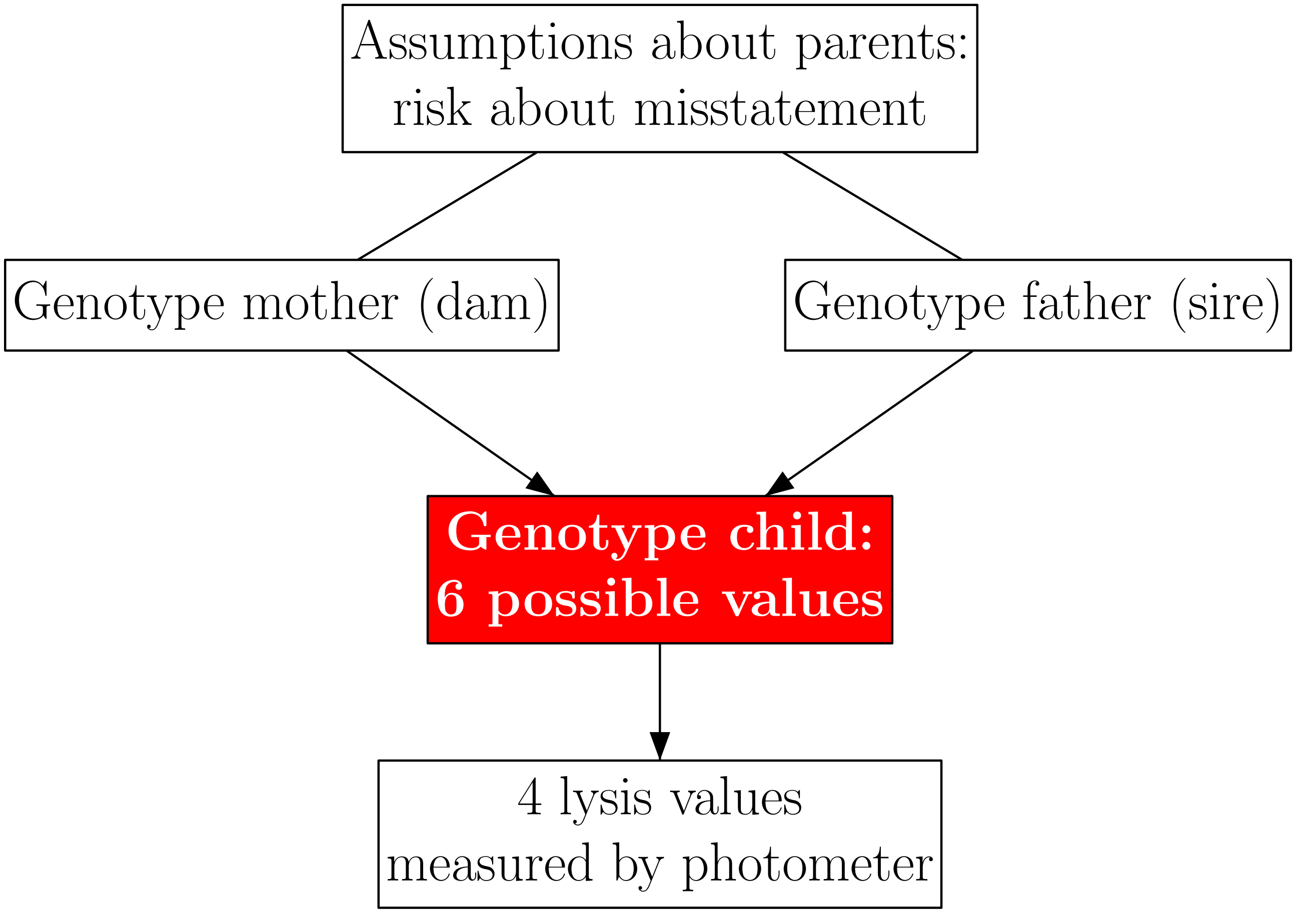
decomposition of the distribution

↓ formally provable

evidence propagation scheme

- Problem: strong assumptions about the statistical effects of causal relations.
- Nevertheless this approach often yields usable graphical models.

Example 1: Genotype Determination of Danish Jersey Cattle

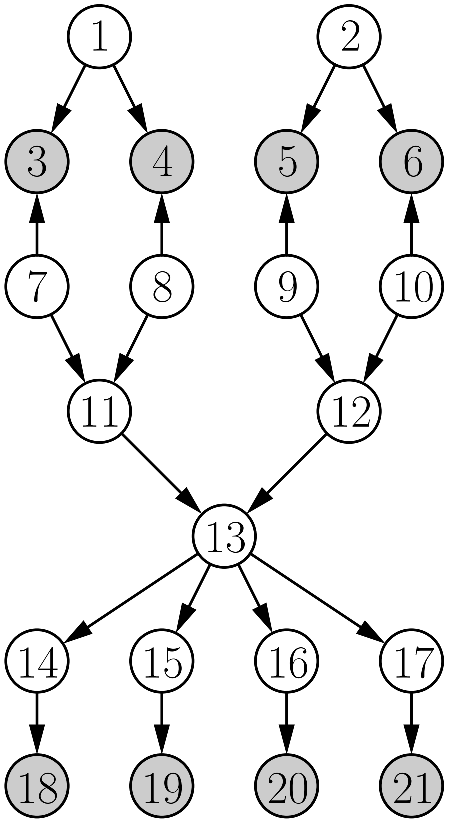


Reliability of databases

Inheritance rules

Blood group determination

Danish Jersey Cattle Blood Type Determination



21 attributes:
1 – dam correct?
2 – sire correct?
3 – stated dam ph.gr. 1
4 – stated dam ph.gr. 2
5 – stated sire ph.gr. 1
6 – stated sire ph.gr. 2
7 – true dam ph.gr. 1
8 – true dam ph.gr. 2
9 – true sire ph.gr. 1
10 – true sire ph.gr. 2
11 – offspring ph.gr. 1
12 – offspring ph.gr. 2
13 – offspring genotype
14 – factor 40
15 – factor 41
16 – factor 42
17 – factor 43
18 – lysis 40
19 – lysis 41
20 – lysis 42
21 – lysis 43

The grey nodes correspond to observable attributes.

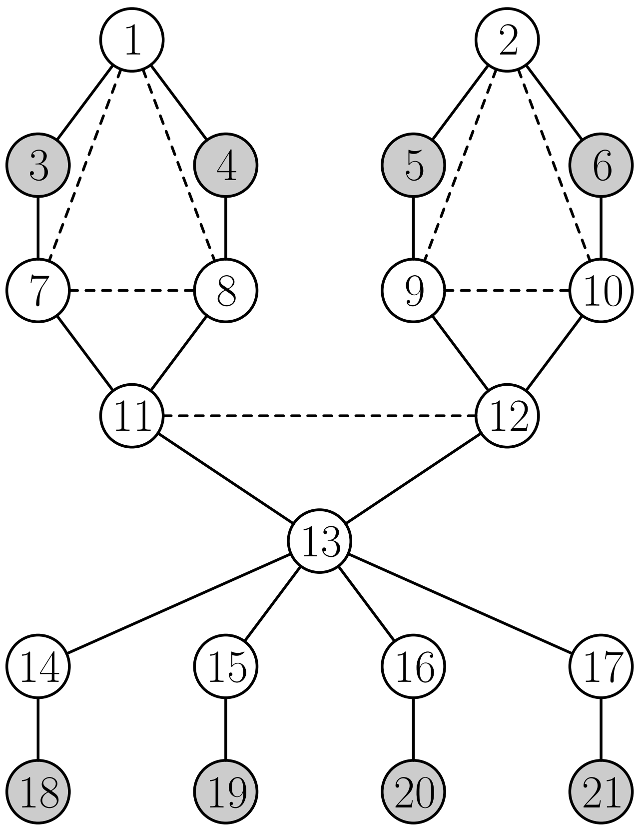
- This graph was specified by human domain experts, based on knowledge about (causal) dependences of the variables.

Example 1: Genotype Determination of Danish Jersey Cattle

- Full 21-dimensional domain has $2^6 \cdot 3^{10} \cdot 6 \cdot 8^4 = 92\,876\,046\,336$ possible states.
- Bayesian network requires only 306 conditional probabilities.
- Example of a conditional probability table (attributes 2, 9, and 5):

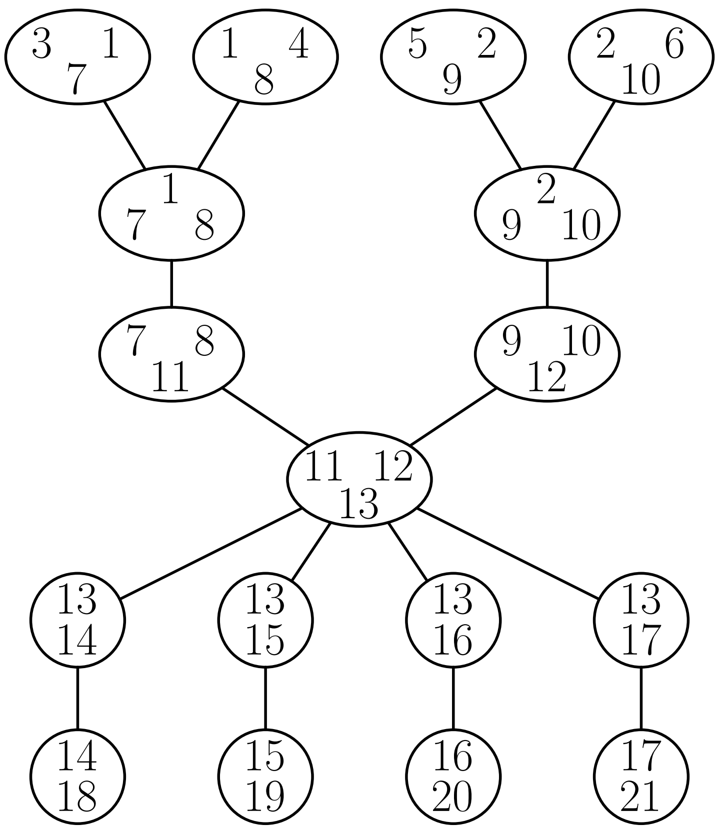
sire correct	true sire phenogroup 1	stated sire phenogroup 1 F1	V1	V2
yes	F1	1	0	0
yes	V1	0	1	0
yes	V2	0	0	1
no	F1	0.58	0.10	0.32
no	V1	0.58	0.10	0.32
no	V2	0.58	0.10	0.32

- The probabilities are acquired from human domain experts or estimated from historical data.



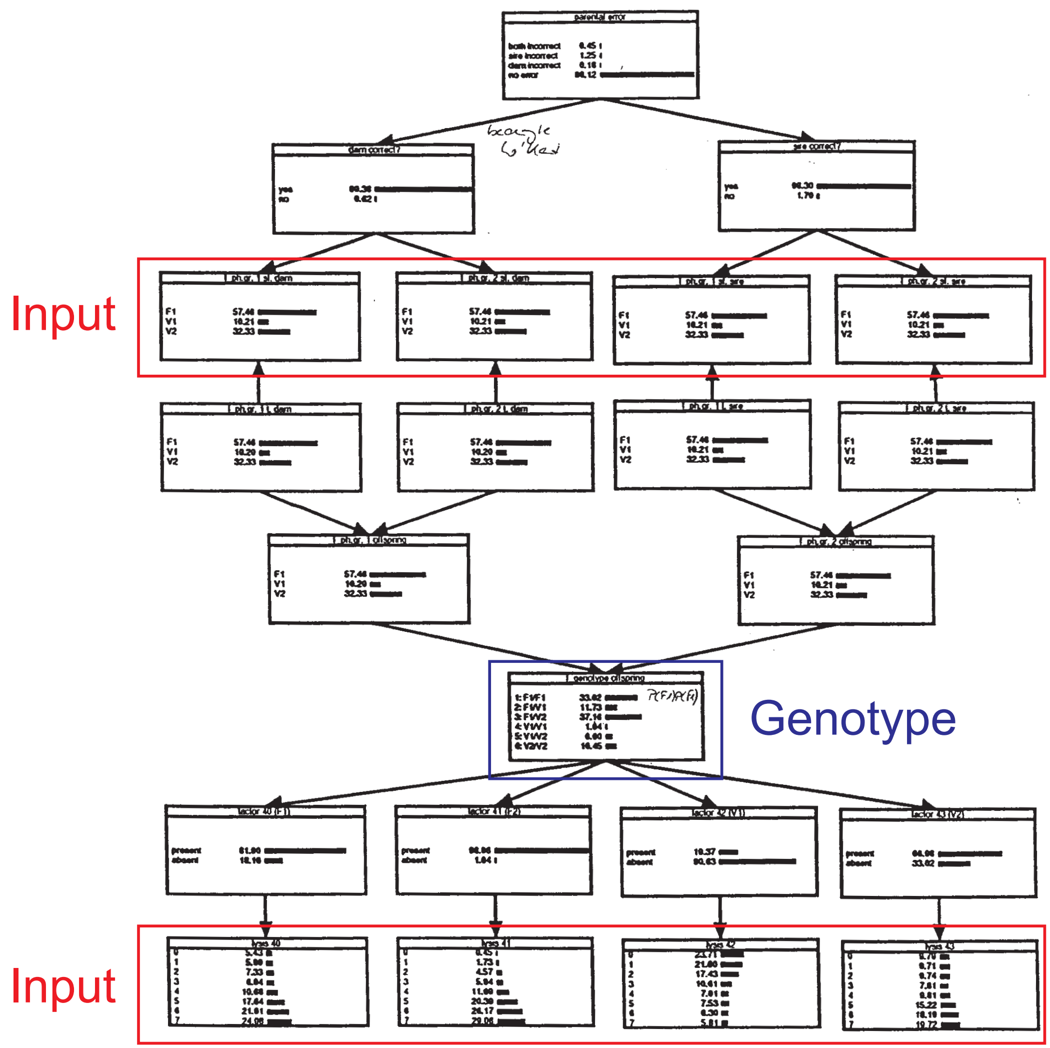
moral graph

(already triangulated)

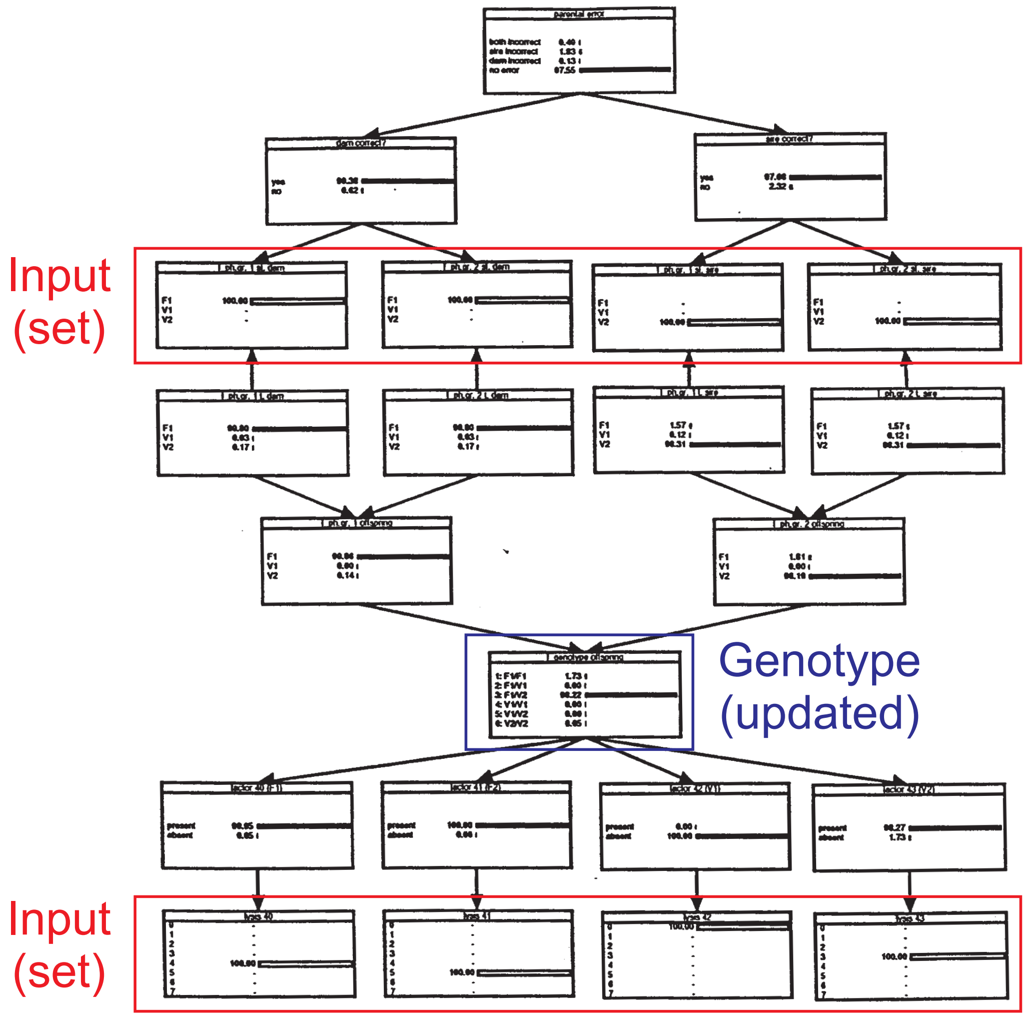


join tree

Marginal distributions before setting evidence:



Conditional distributions given evidence in the input variables:



Strategy of the VW Group

Marketing strategy	Vehicle specification by clients	Bestsellers defined by manufacturer
Complexity	Huge number of variants	Small number of variants

Vehicle specification

Equipment	fastback	2,8 l, 150 kW	Type Alpha	4	leather	...
Group	car body type	engine	radio	doors	seat cover	...

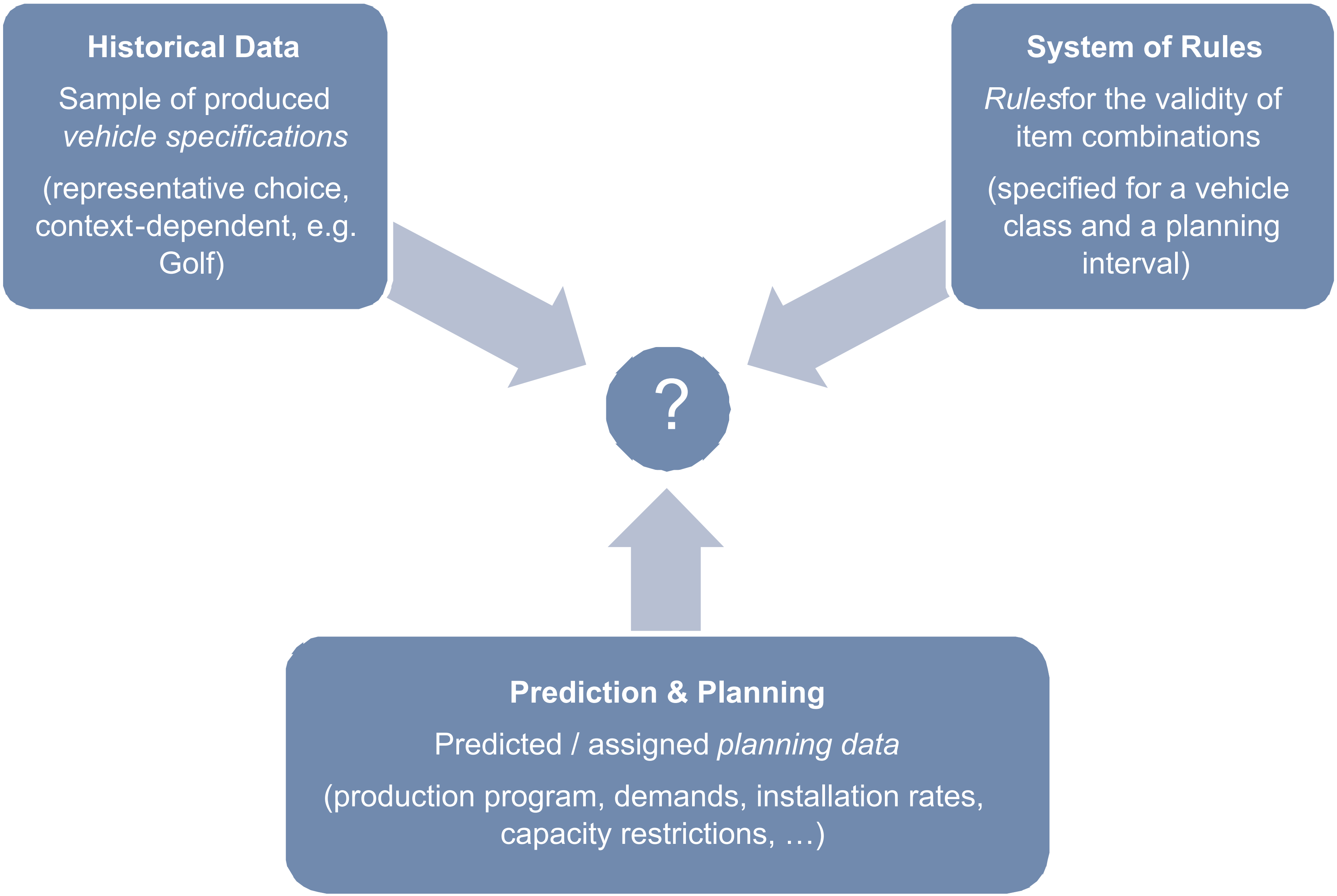
Example 2: Model "Golf"

- Approx. 200 equipment groups
- 2 to 50 items per group
- Therefore more than 2^{200} possible vehicle specifications
- Choice of valid specifications is constrained by a rule system (10000 technical rules, plus marketing and production rules)

Example of technical rules:

- **If** Engine=e_1 **then** Transmission=t_3
- **If** Engine=e_4 and Heating=h_2 **then** Generator $\in \{g_3, g_4, g_5\}$

Problem Representation



Complexity of the Planning Problem

Equipment table

	Engine	Transmission	Heating	Generator	$\cdots$
1	e_1	t_3	h_1	g_1	$\cdots$
2	e_2	t_4	h_3	g_5	$\cdots$
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
100000	e_7	t_1	h_3	g_2	$\cdots$

Installation rates

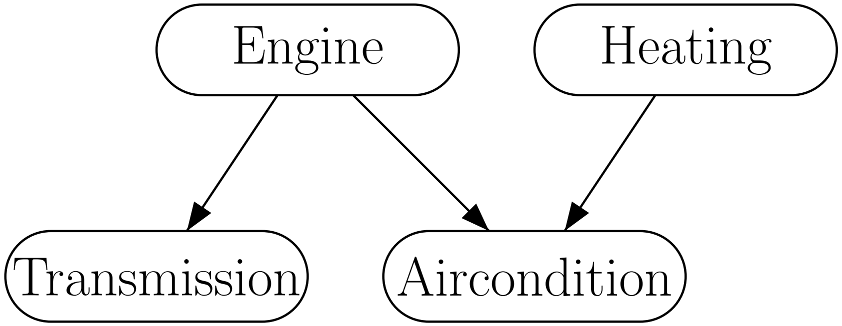
Engine	Transmission	Heating	Generator	$\cdots$	Rate
e_1	t_1	h_1	g_1	$\cdots$	0.0000012
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Result is a 200-dimensional, finite probability space

- $P(\text{Engine} = e_1, \text{Transmission} = t_3) = \,?$
- $P(\text{Heating} = h_1 \mid \text{Generator} = g_3) = \,?$

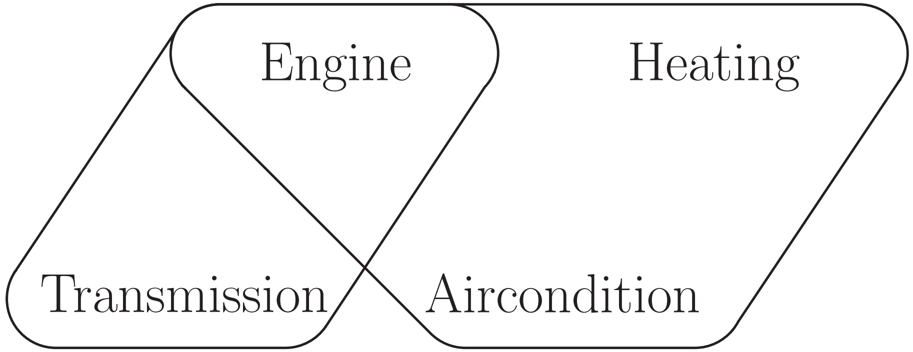
Problem of complexity!

Solution: Decomposition into Subspaces



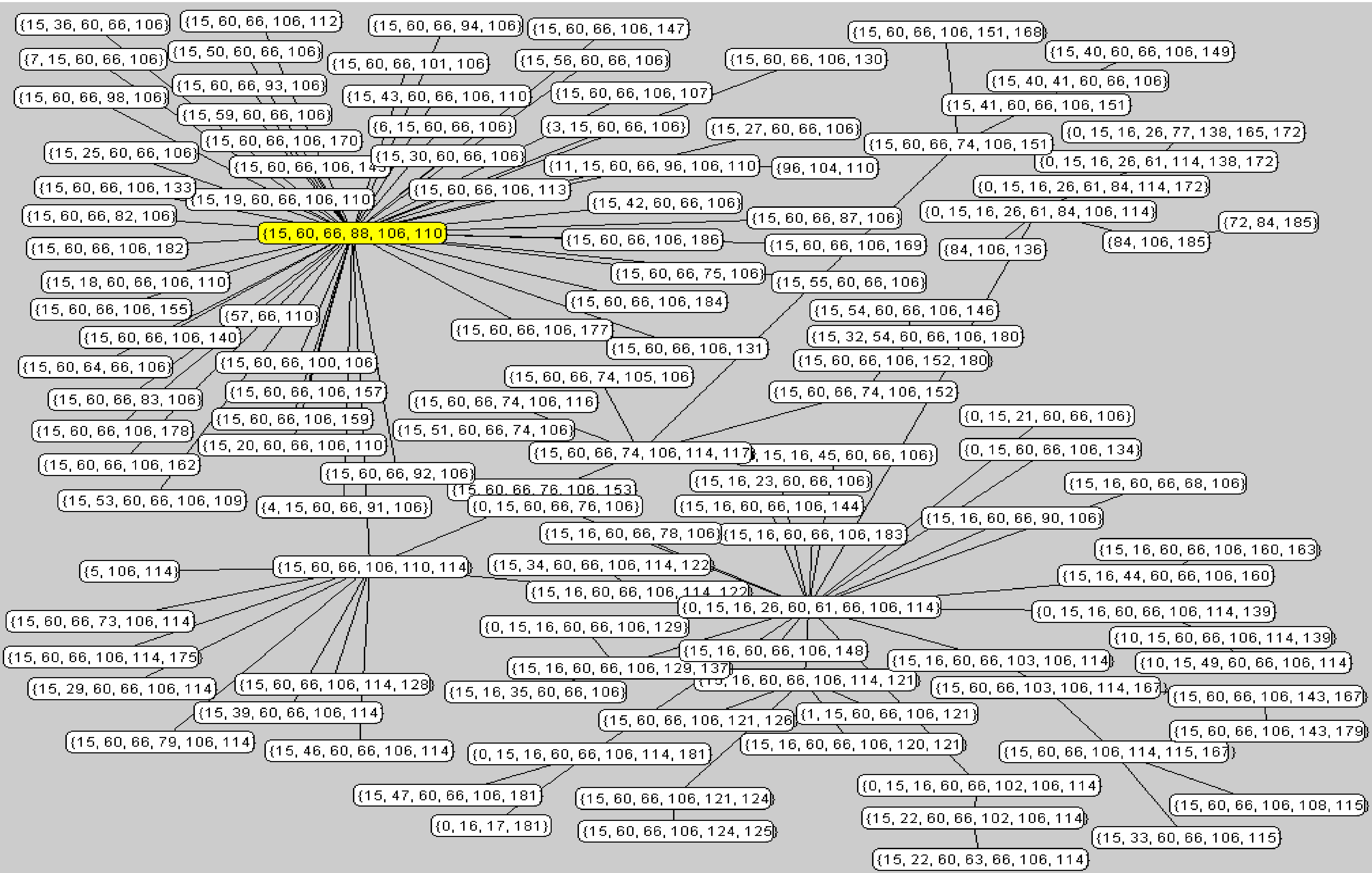
Bayesian Network

$$
\begin{aligned}
P(E,H,T,A) &= P(A \mid E,H,T) \cdot P(T \mid E,H) \cdot P(E \mid H) \cdot P(H) \\
&\overset{\text{here}}{=} P(A \mid E,H) \cdot P(T \mid E) \cdot P(E) \cdot P(H)
\end{aligned}
$$



Hypergraph Decomposition

Clique Tree of the VW Bora

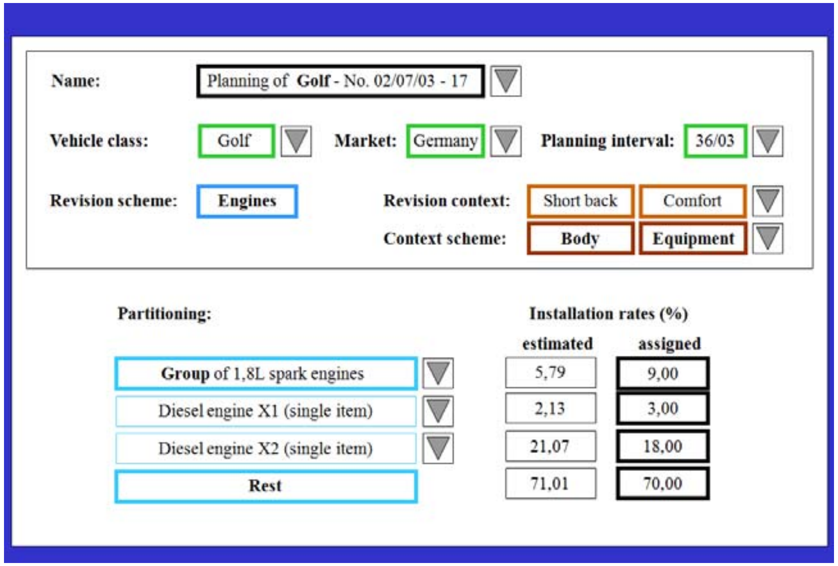


Typical Planning Operation: Focusing

- **Application:**
  - **Compute item demand**
    Calculation of installation rates of equipment combinations
  - **Simulation**
    Analyze customer requirements (e. g. of persons having ordered a navigation system for a VW Polo)

- **Input:** Equipment combinations

- **Operation:** Compute
  - the conditional network distribution and
  - the probabilities of the specified equipment combinations.



Implementation and Deployment

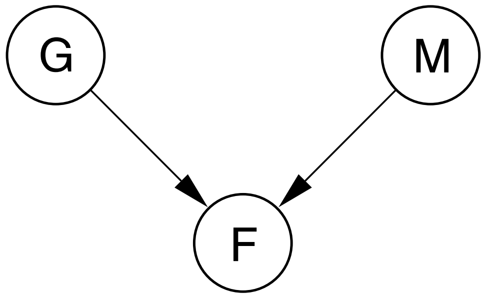
- Projec leader: Jörg Gebhardt
- Client server system
- Server on 6–8 maschines
- Quadcore platform
- Terabyte hard drive
- Java, Linux, Oracle
- WebSphere application server
- Software used daily worldwide
- 15 developers
- 4000 Bayesian networks are currently used

Learning Graphical Models

Prerequisites: Structure vs. Parameters

$A_1 = \mathsf{G}$	$Q_{11} = \phi$
$a_{11} = \mathsf{g}$	
$a_{12} = \overline{\mathsf{g}}$	

$A_2 = \mathsf{M}$	$Q_{21} = \phi$
$a_{12} = \mathsf{m}$	
$a_{22} = \overline{\mathsf{m}}$	



$A_3 = \mathsf{F}$	$Q_{31} = (\mathsf{g}, \mathsf{m})$	$Q_{32} = (\mathsf{g}, \overline{\mathsf{m}})$	$Q_{33} = (\overline{\mathsf{g}}, \mathsf{m})$	$Q_{34} = (\overline{\mathsf{g}}, \overline{\mathsf{m}})$
$a_{31} = \mathsf{f}$				
$a_{32} = \overline{\mathsf{f}}$				

- $V = \{\mathsf{G}, \mathsf{M}, \mathsf{F}\}$
- $\mathrm{dom}(\mathsf{G}) = \{\mathsf{g}, \overline{\mathsf{g}}\}$
- $\mathrm{dom}(\mathsf{M}) = \{\mathsf{m}, \overline{\mathsf{m}}\}$
- $\mathrm{dom}(\mathsf{F}) = \{\mathsf{f}, \overline{\mathsf{f}}\}$

- The potential tables' layout is determined by the graph structure.
- The parameters (i. e. the table entries) can be easily estimated from the database, e. g.:

$$\hat{P}(\mathsf{f} \mid \mathsf{g}, \mathsf{m}) = \frac{\#(\mathsf{F} = \mathsf{f}, \mathsf{G} = \mathsf{g}, \mathsf{M} = \mathsf{m})}{\#(\mathsf{G} = \mathsf{g}, \mathsf{M} = \mathsf{m})}$$

Prerequisites: Likelihood of a Database

Flu G Malaria M Fever F	$\overline{\mathsf{g}}$ $\overline{\mathsf{m}}$ $\overline{\mathsf{f}}$	$\overline{\mathsf{g}}$ $\overline{\mathsf{m}}$ f	$\overline{\mathsf{g}}$ m $\overline{\mathsf{f}}$	$\overline{\mathsf{g}}$ m f	g $\overline{\mathsf{m}}$ $\overline{\mathsf{f}}$	g $\overline{\mathsf{m}}$ f	g m $\overline{\mathsf{f}}$	g m f
#	34	6	2	8	16	24	0	10

Database D with 100 entries for 3 attributes.

$$P(D \mid G) = \prod_{h=1}^{100} P(c_h \mid G)$$

$$= \underbrace{\overbrace{P(\mathsf{g},\mathsf{m},\mathsf{f})}^{\text{Case 1}} \cdot \cdots \cdot \overbrace{P(\mathsf{g},\mathsf{m},\mathsf{f})}^{\text{Case 10}}}_{\text{10 times}} \quad \cdots \quad \underbrace{\overbrace{P(\overline{\mathsf{g}},\mathsf{m},\mathsf{f})}^{\text{Case 51}} \cdot \cdots \cdot \overbrace{P(\overline{\mathsf{g}},\mathsf{m},\mathsf{f})}^{\text{Case 58}}}_{\text{8 times}} \quad \cdots \quad \underbrace{\overbrace{P(\overline{\mathsf{g}},\overline{\mathsf{m}},\overline{\mathsf{f}})}^{\text{Case 67}} \cdot \cdots \cdot \overbrace{P(\overline{\mathsf{g}},\overline{\mathsf{m}},\overline{\mathsf{f}})}^{\text{Case 100}}}_{\text{34 times}}$$

$$= P(\mathsf{g},\mathsf{m},\mathsf{f})^{10} \quad \cdots \quad P(\overline{\mathsf{g}},\mathsf{m},\mathsf{f})^{8} \quad \cdots \quad P(\overline{\mathsf{g}},\overline{\mathsf{m}},\overline{\mathsf{f}})^{34}$$

$$= P(\mathsf{f} \mid \mathsf{g},\mathsf{m})^{10} P(\mathsf{g})^{10} P(\mathsf{m})^{10} \quad \cdots \quad P(\mathsf{f} \mid \overline{\mathsf{g}},\mathsf{m})^{8} P(\overline{\mathsf{g}})^{8} P(\mathsf{m})^{8} \quad \cdots \quad P(\overline{\mathsf{f}} \mid \overline{\mathsf{g}},\overline{\mathsf{m}})^{34} P(\overline{\mathsf{g}})^{34} P(\overline{\mathsf{m}})^{34}$$

Prerequisites: Likelihood of a Database (2)

$$
\begin{aligned}
P(D \mid G) \quad &= \quad \prod_{h=1}^{100} P(c_h \mid G) \\
= P(\mathsf{f} \mid \mathsf{g}, \mathsf{m})^{10} P(\bar{\mathsf{f}} \mid \mathsf{g}, \mathsf{m})^{0} P(\mathsf{f} \mid \mathsf{g}, \overline{\mathsf{m}})^{24} P(\bar{\mathsf{f}} \mid \mathsf{g}, \overline{\mathsf{m}})^{16} \\
\cdot P(\mathsf{f} \mid \overline{\mathsf{g}}, \mathsf{m})^{8} P(\bar{\mathsf{f}} \mid \overline{\mathsf{g}}, \mathsf{m})^{2} P(\mathsf{f} \mid \overline{\mathsf{g}}, \overline{\mathsf{m}})^{6} P(\bar{\mathsf{f}} \mid \overline{\mathsf{g}}, \overline{\mathsf{m}})^{34} \\
\cdot P(\mathsf{g})^{50} P(\overline{\mathsf{g}})^{50} P(\mathsf{m})^{20} P(\overline{\mathsf{m}})^{80}
\end{aligned}
$$

The last equation shows the principle of reordering the factors:

- First, we sort by attributes (here: F, G then M).
- Within the same attributes, factors are grouped by the parent attributes' values combinations (here: for F: (g, m), $(\mathsf{g}, \overline{\mathsf{m}})$, $(\overline{\mathsf{g}}, \mathsf{m})$ and $(\overline{\mathsf{g}}, \overline{\mathsf{m}})$).
- Finally, it is sorted by attribute values (here: for F: first f, then $\bar{\mathsf{f}}$).

General likelihood of a database D given a DAG G:

$$P(D \mid G) \quad = \quad \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}}$$

General potential table:

A_i	Q_{i1}	$\cdots$	Q_{ij}	$\cdots$	Q_{iq_i}
a_{i1}	θ_{i11}	$\cdots$	θ_{ij1}	$\cdots$	θ_{iq_i1}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_{ik}	θ_{i1k}	$\cdots$	θ_{ijk}	$\cdots$	θ_{iq_ik}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_{ir_i}	θ_{i1r_i}	$\cdots$	θ_{ijr_i}	$\cdots$	$\theta_{iq_ir_i}$

$$P(A_i = a_{ik} \mid \text{parents}(A_i) = Q_{ij}) = \theta_{ijk}$$

$$\sum_{k=1}^{r_i} \theta_{ijk} = 1$$

(A) Test whether a distribution is decomposable w. r. t. a given graph.

This is the most direct approach. It is not bound to a graphical representation, but can also be carried out w.r.t. other representations of the set of subspaces to be used to compute the (candidate) decomposition of the given distribution.

(B) Find a suitable graph by measuring the strength of dependences.

This is a heuristic, but often highly successful approach, which is based on the frequently valid assumption that in a conditional independence graph an attribute is more strongly dependent on adjacent attributes than on attributes that are not directly connected to them.

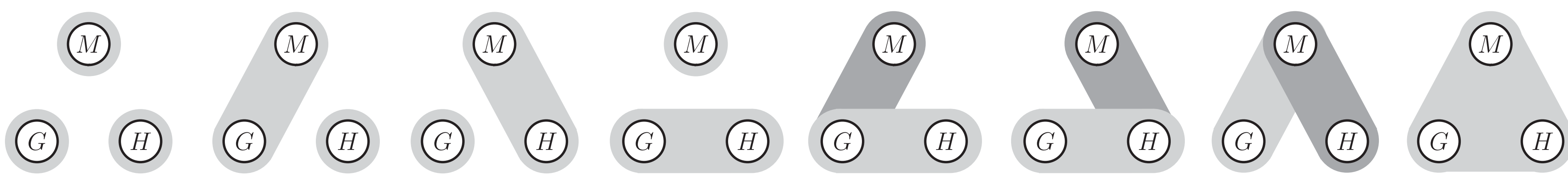
(C) Find an independence map by conditional independence tests.

This approach exploits the theorems that connect conditional independence graphs and graphs that represent decompositions. It has the advantage that a single conditional independence test, if it fails, can exclude several candidate graphs. However, wrong test results can thus have severe consequences.

Evaluation Measures and Search Methods

- All learning algorithms for graphical models consist of an **evaluation measure** or **scoring function** and a (heuristic) **search method**, e. g.
  - conditional independence search
  - greedy search (spanning tree or K2 algorithm)
  - guided random search (simulated annealing, genetic algorithms)

- An exhaustive search over all graphs is too expensive:
  - $2^{\binom{n}{2}}$ possible undirected graphs for n attributes.
  - $f(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i)$ possible directed acyclic graphs.



8 possible undirected graphs with 3 nodes

Evaluation Measures / Scoring Functions

Relational Networks

- Hartley Information Gain
- Conditional Hartley Information Gain

Probabilistic Networks

- χ^2-Measure
- Mutual Information / Cross Entropy / Information Gain
- (Symmetric) Information Gain Ratio
- (Symmetric/Modified) Gini Index
- Bayesian Measures (K2 metric, BDeu metric)
- Measures based on the Minimum Description Length Principle
- Other measures that are known from Decision Tree Induction

(A) Test whether a distribution is decomposable w. r. t. a given graph.

This is the most direct approach. It is not bound to a graphical representation, but can also be carried out w.r.t. other representations of the set of subspaces to be used to compute the (candidate) decomposition of the given distribution.

(B) Find a suitable graph by measuring the strength of dependences.

This is a heuristic, but often highly successful approach, which is based on the frequently valid assumption that in a conditional independence graph an attribute is more strongly dependent on adjacent attributes than on attributes that are not directly connected to them.

(C) Find an independence map by conditional independence tests.

This approach exploits the theorems that connect conditional independence graphs and graphs that represent decompositions. It has the advantage that a single conditional independence test, if it fails, can exclude several candidate graphs. However, wrong test results can thus have severe consequences.

Testing for Decomposability: Comparing Relations

- In order to evaluate a graph structure, we need a measure that compares the actual relation to the relation represented by the graph.

- For arbitrary R, E_1, and E_2 it is

$$R(E_1 \cap E_2) \leq \min\{R(E_1), R(E_2)\}.$$

- This relation entails that for any family $\mathcal{M}$ of subsets of U it is always:

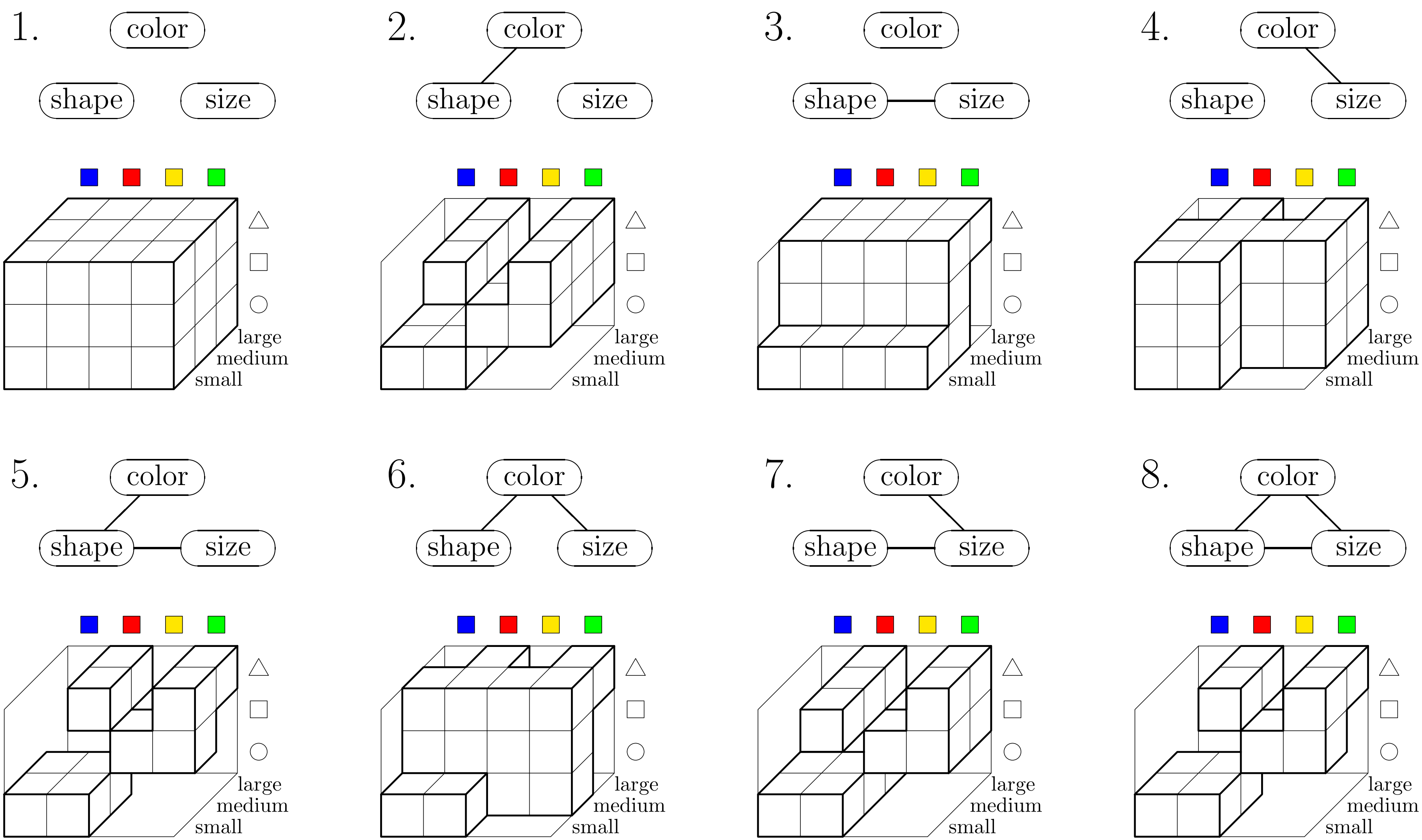
$$\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) :$$
$$r_U\left(\bigwedge_{A_i \in U} A_i = a_i\right) \leq \min_{M \in \mathcal{M}} \left\{ r_M\left(\bigwedge_{A_i \in M} A_i = a_i\right)\right\}.$$

- Therefore: Measure the quality of a family $\mathcal{M}$ as:

$$\sum_{a_1 \in \mathrm{dom}(A_1)} \cdots \sum_{a_n \in \mathrm{dom}(A_n)} \left(\min_{M \in \mathcal{M}} \left\{ r_M\left(\bigwedge_{A_i \in M} A_i = a_i\right)\right\} - r_U\left(\bigwedge_{A_i \in U} A_i = a_i\right)\right)$$

Intuitively: **Count the number of additional tuples.**

Direct Test for Decomposability: Relational



Comparing Probability Distributions

Definition: Let P_1 and P_2 be two strictly positive probability distributions on the same set $\mathcal{E}$ of events. Then

$$I_{\mathrm{KLdiv}}(P_1, P_2) = \sum_{F \in \mathcal{E}} P_1(F) \log_2 \frac{P_1(F)}{P_2(F)}$$

is called the **Kullback-Leibler information divergence** of P_1 and P_2.

- The Kullback-Leibler information divergence is non-negative.
- It is zero if and only if $P_1 \equiv P_2$.
- Therefore it is plausible that this measure can be used to assess the quality of the approximation of a given multi-dimensional distribution P_1 by the distribution P_2 that is represented by a given graph:

  The smaller the value of this measure, the better the approximation.

Excursus: Shannon Entropy

Let X be a random variable with domain $\mathrm{dom}(X) = \{x_1, \ldots, x_n\}$. Then,

$$H^{(\mathrm{Shannon})}(X) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i)$$

is called the **Shannon entropy** of (the probability distribution of) X,
where $0 \cdot \log_2 0 = 0$ is assumed.

Intuitively: **Expected number of yes/no questions that have to be asked in order to determine the obtaining value of X.**

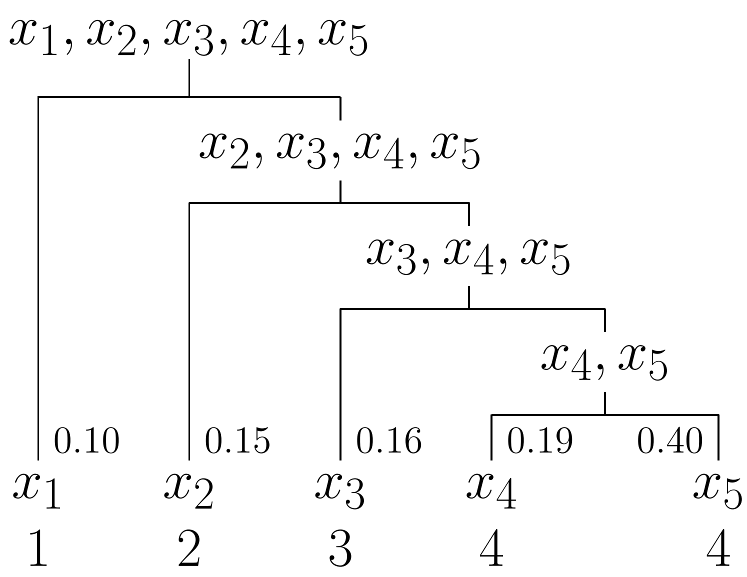
- Suppose there is an oracle, which knows the obtaining value, but responds only if the question can be answered with "yes" or "no".
- A better question scheme than asking for one alternative after the other can easily be found: Divide the set into two subsets of about equal size.
- Ask for containment in an arbitrarily chosen subset.
- Apply this scheme recursively $\rightarrow$ number of questions bounded by $\lceil \log_2 n \rceil$.

Question/Coding Schemes

$$P(x_1) = 0.10, \quad P(x_2) = 0.15, \quad P(x_3) = 0.16, \quad P(x_4) = 0.19, \quad P(x_5) = 0.40$$

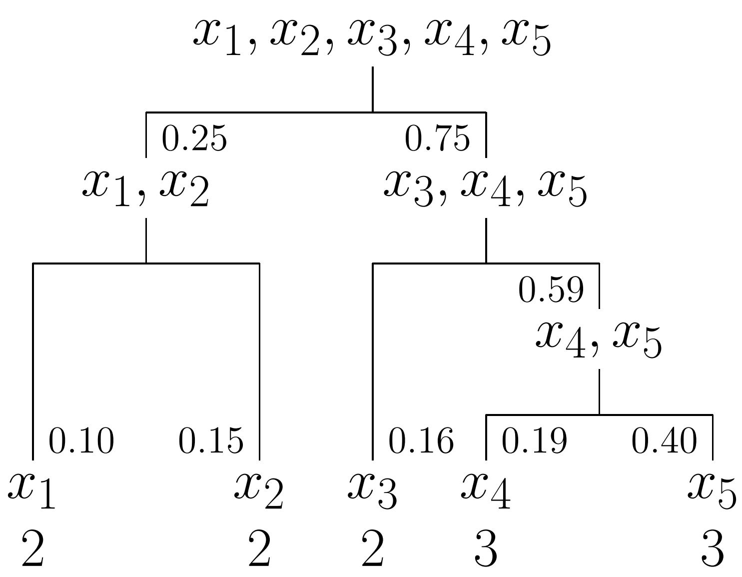
Shannon entropy: $\quad -\sum_i P(x_i) \log_2 P(x_i) = 2.15$ bit/symbol

Linear Traversal



Code length: 3.24 bit/symbol
Code efficiency: 0.664

Equal Size Subsets



Code length: 2.59 bit/symbol
Code efficiency: 0.830

Question/Coding Schemes

- Splitting into subsets of about equal size can lead to a bad arrangement of the alternatives into subsets → high expected number of questions.

- Good question schemes take the probability of the alternatives into account.

- **Shannon-Fano Coding** (1948)
  - Build the question/coding scheme top-down.
  - Sort the alternatives w.r.t. their probabilities.
  - Split the set so that the subsets have about equal *probability*
    (splits must respect the probability order of the alternatives).

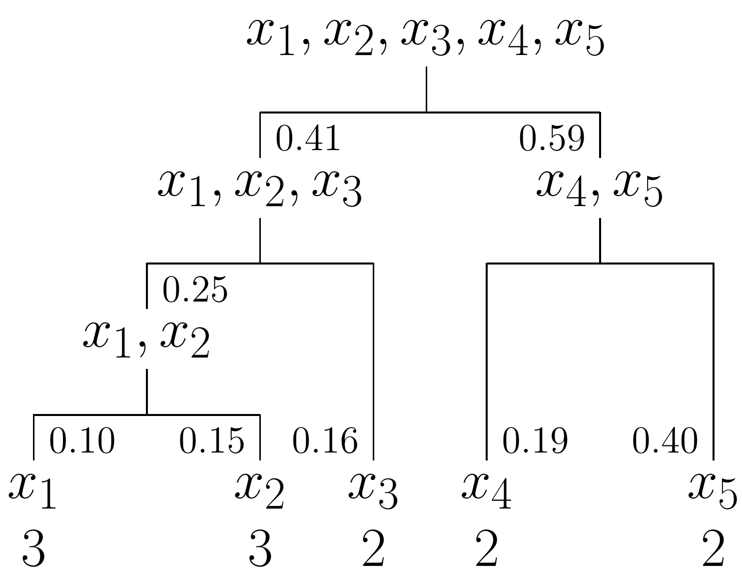
- **Huffman Coding** (1952)
  - Build the question/coding scheme bottom-up.
  - Start with one element sets.
  - Always combine those two sets that have the smallest probabilities.

Question/Coding Schemes

$$P(x_1) = 0.10, \quad P(x_2) = 0.15, \quad P(x_3) = 0.16, \quad P(x_4) = 0.19, \quad P(x_5) = 0.40$$

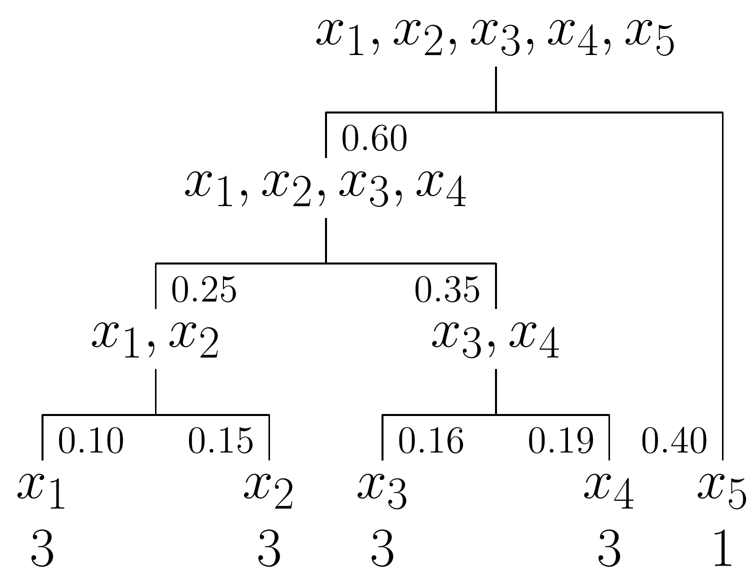
Shannon entropy: $\quad -\sum_i P(x_i) \log_2 P(x_i) = 2.15$ bit/symbol

Shannon–Fano Coding (1948)



Code length: 2.25 bit/symbol
Code efficiency: 0.955

Huffman Coding (1952)



Code length: 2.20 bit/symbol
Code efficiency: 0.977

Question/Coding Schemes

- It can be shown that Huffman coding is optimal if we have to determine the obtaining alternative in a single instance.
  (No question/coding scheme has a smaller expected number of questions.)
- Only if the obtaining alternative has to be determined in a sequence of (independent) situations, this scheme can be improved upon.
- Idea: Process the sequence not instance by instance, but combine two, three or more consecutive instances and ask directly for the obtaining combination of alternatives.
- Although this enlarges the question/coding scheme, the expected number of questions per identification is reduced (because each interrogation identifies the obtaining alternative for several situations).
- However, the expected number of questions per identification cannot be made arbitrarily small. Shannon showed that there is a lower bound, namely the Shannon entropy.

Interpretation of Shannon Entropy

$$P(x_1) = \tfrac{1}{2}, \quad P(x_2) = \tfrac{1}{4}, \quad P(x_3) = \tfrac{1}{8}, \quad P(x_4) = \tfrac{1}{16}, \quad P(x_5) = \tfrac{1}{16}$$

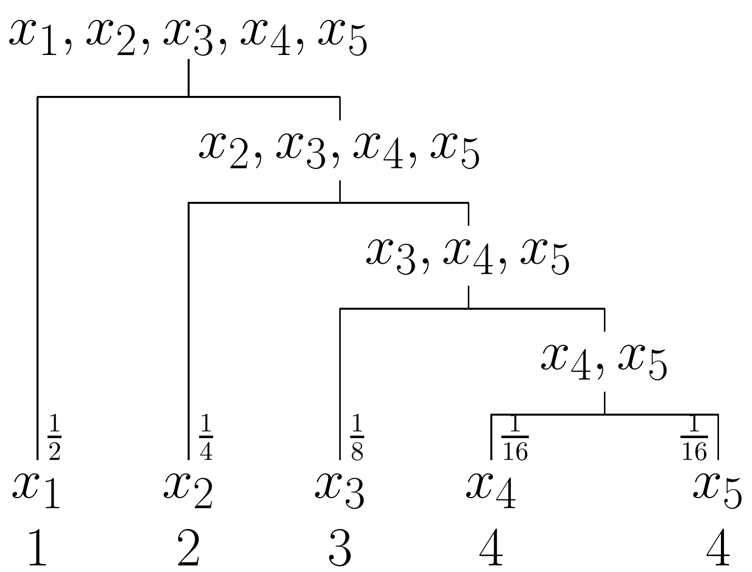
Shannon entropy: $\quad -\sum_i P(x_i) \log_2 P(x_i) = 1.875$ bit/symbol

If the probability distribution allows for a perfect Huffman code (code efficiency 1), the Shannon entropy can easily be interpreted as follows:

$$-\sum_i P(x_i) \log_2 P(x_i)$$
$$= \sum_i \underbrace{P(x_i)}_{\substack{\text{occurrence}\\ \text{probability}}} \cdot \underbrace{\log_2 \frac{1}{P(x_i)}}_{\substack{\text{path length}\\ \text{in tree}}}.$$

In other words, it is the expected number of needed yes/no questions.

Perfect Question Scheme



Code length: 1.875 bit/symbol
Code efficiency: 1

Reference to Kullback-Leibler Information Divergence

Information Content

The information content of an event $F \in \mathcal{E}$ that occurs with probability $P(F)$ is defined as

$$\mathrm{Inf}_P(F) = -\log_2 P(F).$$

Intention:

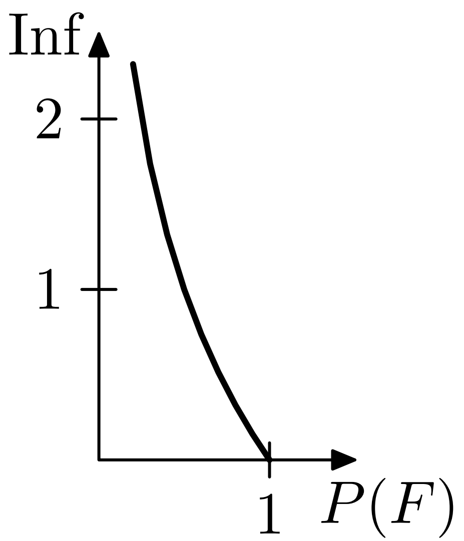
- Neglect all subjective references to F and let the information content be determined by $P(F)$ only.
- The information of a certain message ($P(\Omega) = 1$) is zero.
- The less frequent a message occurs (i. e., the less probable it is), the more interesting is the fact of its occurrence:

$$P(F_1) < P(F_2) \quad \Rightarrow \quad \mathrm{Inf}_P(F_1) > \mathrm{Inf}_P(F_2)$$

- We only use one bit to encode the occurrence of a message with probability $\frac{1}{2}$.

Excursus: Information Content

The function Inf fulfills all these requirements:



- The expected value (w. r. t. to a probability distribution P_1) of Inf_{P_2} can be written as follows:

$$E_{P_1}(\mathrm{Inf}_{P_2}) = -\sum_{F \in \mathcal{E}} P_1(F) \cdot \log_2 P_2(F)$$

- $H^{(\text{Shannon})}(P)$ is the expected value (in bits) of the information content that is related to the occurrence of the events $F \in \mathcal{E}$:

$$H(P) = E_P(\mathrm{Inf}_P)$$

$$H^{(\text{Shannon})}(P) = \sum_{F \in \mathcal{E}} \underbrace{P(F)}_{\text{Probability of } F} \cdot \underbrace{(-\log_2 P(F))}_{\text{Information content of } F}$$

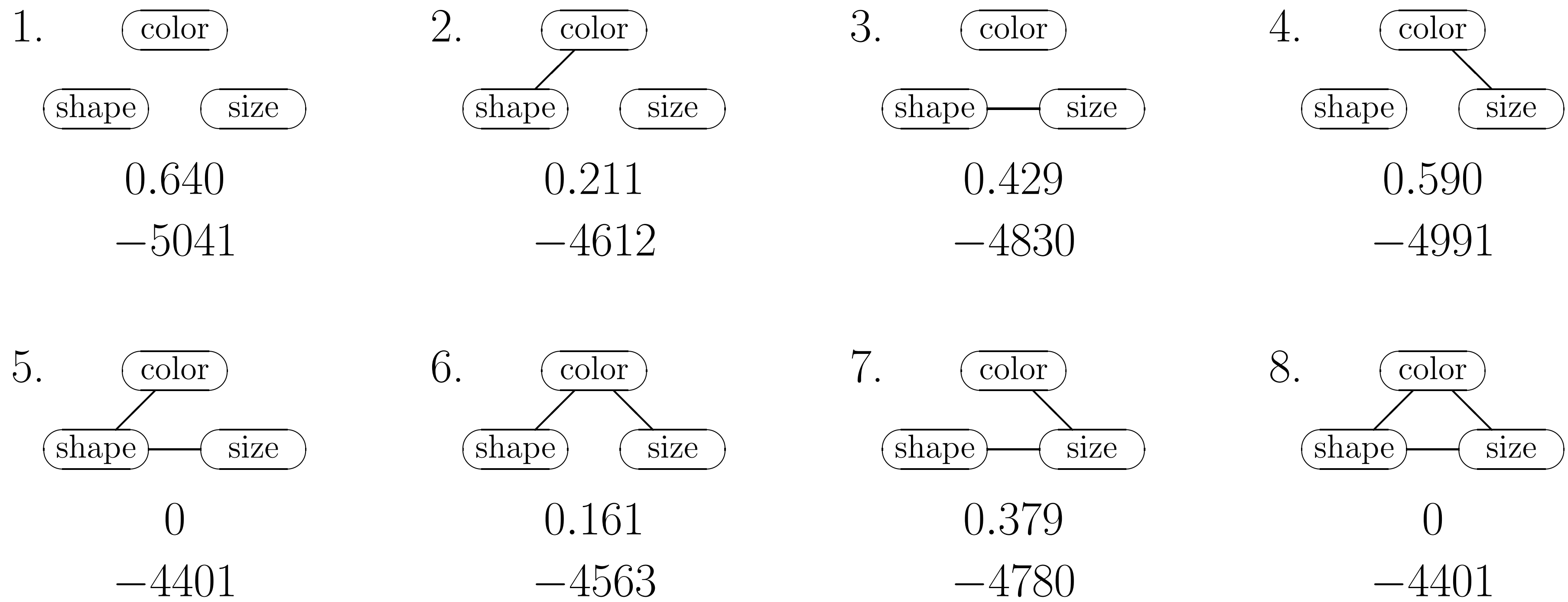
Excursus: Approximation Measure

- Let P^* be a hypothetical probability distribution and P a (given or known) probability distribution that acts as a reference.

- We can compare both P^* and P by computing the **difference of the expected information contents**:

$$
\begin{aligned}
E_P(\mathrm{Inf}_{P^*}) - E_P(\mathrm{Inf}_P) &= -\sum_{F \in \mathcal{E}} P(F) \log_2 P^*(F) + \sum_{F \in \mathcal{E}} P(F) \log_2 P(F) \\
&= \sum_{F \in \mathcal{E}} \Big(P(F) \log_2 P(F) - P(F) \log_2 P^*(F) \Big) \\
&= \sum_{F \in \mathcal{E}} P(F) \Big(\log_2 P(F) - \log_2 P^*(F) \Big) \\
I_{\mathrm{KLdiv}}(P, P^*) &= \sum_{F \in \mathcal{E}} P(F) \log_2 \frac{P(F)}{P^*(F)}
\end{aligned}
$$

Direct Test for Decomposability: Probabilistic



Upper numbers: The Kullback-Leibler information divergence of the original distribution and its approximation.

Lower numbers: The binary logarithms of the probability of an example database (log-likelihood of data).

Learning the Structure of Graphical Models from Data

(A) Test whether a distribution is decomposable w. r. t. a given graph.

This is the most direct approach. It is not bound to a graphical representation, but can also be carried out w.r.t. other representations of the set of subspaces to be used to compute the (candidate) decomposition of the given distribution.

(B) Find a suitable graph by measuring the strength of dependences.

This is a heuristic, but often highly successful approach, which is based on the frequently valid assumption that in a conditional independence graph an attribute is more strongly dependent on adjacent attributes than on attributes that are not directly connected to them.

(C) Find an independence map by conditional independence tests.

This approach exploits the theorems that connect conditional independence graphs and graphs that represent decompositions. It has the advantage that a single conditional independence test, if it fails, can exclude several candidate graphs. However, wrong test results can thus have severe consequences.

Strength of Marginal Dependences: Relational

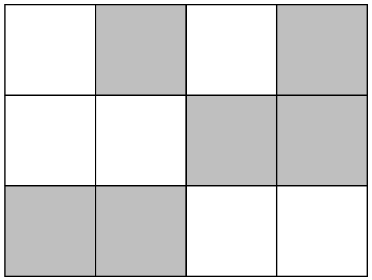
- Learning a relational network consists in finding those subspace, for which the intersection of the cylindrical extensions of the projections to these subspaces approximates best the set of possible world states, i. e. contains as few additional tuples as possible.

- Since computing explicitly the intersection of the cylindrical extensions of the projections and comparing it to the original relation is too expensive, local evaluation functions are used, for instance:

subspace	color × shape	shape × size	size × color
possible combinations	12	9	12
occurring combinations	6	5	8
relative number	50%	56%	67%

- The relational network can be obtained by interpreting the relative numbers as edge weights and constructing the minimum weight spanning tree.

Strength of Marginal Dependences: Relational



Hartley information needed to determine

coordinates:	$\log_2 4 + \log_2 3 = \log_2 12$	≈ 3.58
coordinate pair:	$\log_2 6$	≈ 2.58
gain:	$\log_2 12 - \log_2 6 = \log_2 2 = 1$	

Definition: Let A and B be two attributes and R a discrete possibility measure with $\exists a \in \mathrm{dom}(A) : \exists b \in \mathrm{dom}(B) : R(A = a, B = b) = 1$. Then

$$\begin{aligned}
I_{\text{gain}}^{(\text{Hartley})}(A, B) &= \log_2 \left(\sum_{a \in \mathrm{dom}(A)} R(A = a) \right) + \log_2 \left(\sum_{b \in \mathrm{dom}(B)} R(B = b) \right) \\
&\quad - \log_2 \left(\sum_{a \in \mathrm{dom}(A)} \sum_{b \in \mathrm{dom}(B)} R(A = a, B = b) \right) \\
&= \log_2 \frac{\left(\sum_{a \in \mathrm{dom}(A)} R(A = a) \right) \cdot \left(\sum_{b \in \mathrm{dom}(B)} R(B = b) \right)}{\sum_{a \in \mathrm{dom}(A)} \sum_{b \in \mathrm{dom}(B)} R(A = a, B = b)},
\end{aligned}$$

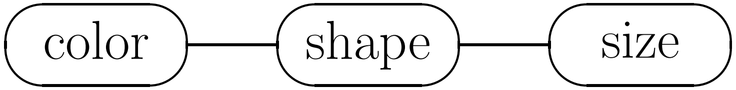
is called the **Hartley information gain** of A and B w.r.t. R.

- **Intuitive interpretation of Hartley information gain:**
  The binary logarithm measures the number of questions to find the obtaining value with a scheme like a binary search. Thus Hartley information gain measures the reduction in the number of necessary questions.

- Results for the simple example:

$$
\begin{aligned}
I_{\text{gain}}^{(\text{Hartley})}(\text{color}, \text{shape}) &= 1.00 \text{ bit} \\
I_{\text{gain}}^{(\text{Hartley})}(\text{shape}, \text{size}) &\approx 0.86 \text{ bit} \\
I_{\text{gain}}^{(\text{Hartley})}(\text{color}, \text{size}) &\approx 0.58 \text{ bit}
\end{aligned}
$$

- Applying the Kruskal algorithm yields as a learning result:



As we know, this graph describes indeed a decomposition of the relation.

Mutual Information / Cross Entropy / Information Gain

Based on Shannon Entropy $H = -\sum_{i=1}^{n} p_i \log_2 p_i$ (Shannon 1948)

$$
\begin{aligned}
I_{\text{gain}}(A, B) &= \overbrace{-\sum_{\forall a} P(a) \log_2 P(a)}^{H(A)} - \overbrace{\sum_{\forall b} P(b) \left(-\sum_{\forall a} P(a|b) \log_2 P(a|b) \right)}^{H(A \mid B)}
\end{aligned}
$$

$H(A)$	Entropy of the distribution on attribute A
$H(A\|B)$	*Expected entropy* of the distribution on attribute A if the value of attribute B becomes known
$H(A) - H(A\|B)$	Expected reduction in entropy or *information gain*

Strength of Marginal Dependences: Probabilistic

$$\begin{aligned}
I_{\text{gain}}(A,B) &= -\sum_{\forall a} P(a)\log_2 P(a) - \sum_{\forall b} P(b)\left(-\sum_{\forall a} P(a|b)\log_2 P(a|b)\right) \\
&= -\sum_{\forall a}\sum_{\forall b} P(a,b)\log_2 P(a) + \sum_{\forall b}\sum_{\forall a} P(a|b)P(b)\log_2 P(a|b) \\
&= \quad \sum_{\forall a}\sum_{\forall b} P(a,b)\Big(\log_2 \frac{P(a,b)}{P(b)} - \log_2 P(a)\Big) \\
&= \quad \sum_{\forall a}\sum_{\forall b} P(a,b)\log_2 \frac{P(a,b)}{P(a)P(b)}
\end{aligned}$$

The information gain equals the Kullback-Leibler information divergence between the actual distribution $P(A,B)$ and a hypothetical distribution P^* in which A and B are marginal independent:

$$P^*(A,B) = P(A)\cdot P(B)$$

$$I_{\text{gain}}(A,B) = I_{\text{KLdiv}}(P,P^*)$$

Information Gain: Simple Example

projection to subspace	product of marginals	information gain

Projection to subspace:

	🟦	🟥	🟨	🟩
△	40	180	20	160
□	12	6	120	102
○	168	144	30	18

Product of marginals:

	🟦	🟥	🟨	🟩
△	88	132	68	112
□	53	79	41	67
○	79	119	61	101

Information gain: 0.429 bit

Projection to subspace:

	s	m	l
△	20	180	200
□	40	160	40
○	180	120	60

Product of marginals:

	s	m	l
△	96	184	120
□	58	110	72
○	86	166	108

Information gain: 0.211 bit

Projection to subspace:

	🟦	🟥	🟨	🟩
large	50	115	35	100
medium	82	133	99	146
small	88	82	36	34

Product of marginals:

	🟦	🟥	🟨	🟩
large	66	99	51	84
medium	101	152	78	129
small	53	79	41	67

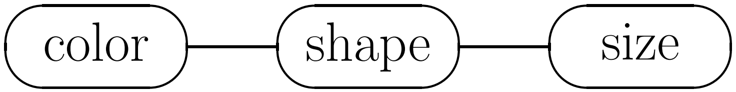
Information gain: 0.050 bit

Strength of Marginal Dependences: Simple Example

- Results for the simple example:

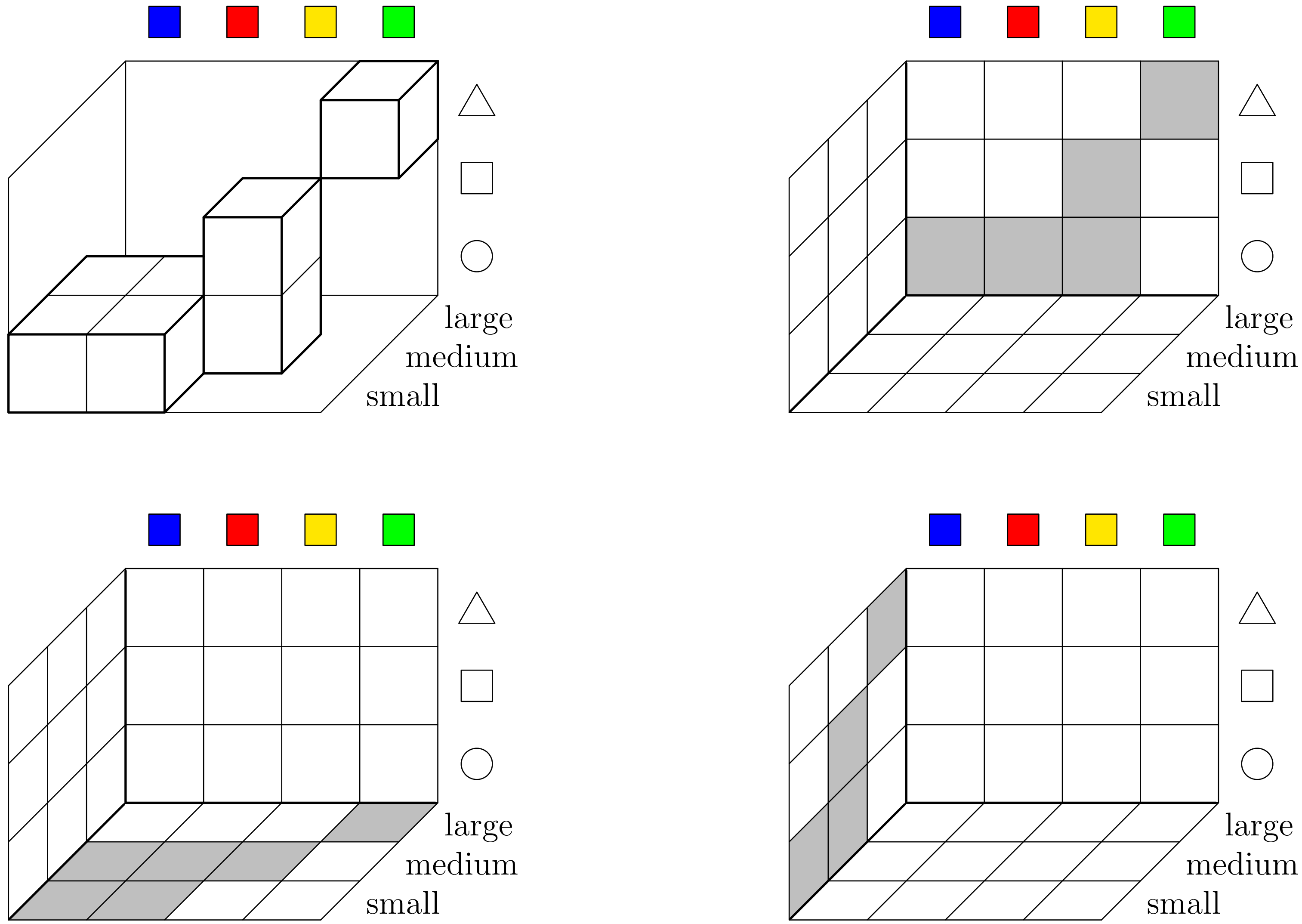
$$
\begin{aligned}
I_{\text{gain}}(\text{color}, \text{shape}) &= 0.429 \text{ bit} \\
I_{\text{gain}}(\text{shape}, \text{size}) &= 0.211 \text{ bit} \\
I_{\text{gain}}(\text{color}, \text{size}) &= 0.050 \text{ bit}
\end{aligned}
$$

- Applying the Kruskal algorithm yields as a learning result:

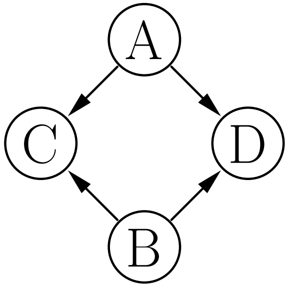


- It can be shown that this approach always yields the best possible spanning tree w.r.t. Kullback-Leibler information divergence (Chow and Liu 1968).
- In an extended form this also holds for certain classes of graphs (for example, tree-augmented naive Bayes classifiers).
- For more complex graphs, the best graph need not be found (there are counterexamples, see below).

Strength of Marginal Dependences: Drawbacks



Strength of Marginal Dependences: Drawbacks



p_A	a_1	a_2
	0.5	0.5

p_B	b_1	b_2
	0.5	0.5

$p_{C\|AB}$	a_1b_1	a_1b_2	a_2b_1	a_2b_2
c_1	0.9	0.3	0.3	0.5
c_2	0.1	0.7	0.7	0.5

$p_{D\|AB}$	a_1b_1	a_1b_2	a_2b_1	a_2b_2
d_1	0.9	0.3	0.3	0.5
d_2	0.1	0.7	0.7	0.5

p_{AD}	a_1	a_2
d_1	0.3	0.2
d_2	0.2	0.3

p_{BD}	b_1	b_2
d_1	0.3	0.2
d_2	0.2	0.3

p_{CD}	c_1	c_2
d_1	0.31	0.19
d_2	0.19	0.31

- Greedy parent selection can lead to suboptimal results if there is more than one path connecting two attributes.
- Here: the edge $C \to D$ is selected first.

- **Optimum Weight Spanning Tree Construction**
  - Compute an evaluation measure on all possible edges (two-dimensional subspaces).
  - Use the Kruskal algorithm to determine an optimum weight spanning tree.

- **Greedy Parent Selection** (for directed graphs)
  - Define a topological order of the attributes (to restrict the search space).
  - Compute an evaluation measure on all single attribute hyperedges.
  - For each preceding attribute (w.r.t. the topological order): add it as a candidate parent to the hyperedge and compute the evaluation measure again.
  - Greedily select a parent according to the evaluation measure.
  - Repeat the previous two steps until no improvement results from them.

K2 Algorithm

- Idea: Compute the probability of a directed graph $\vec{G}$ given the database D (Bayesian approach by [Cooper and Herskovits 1992])

$$\begin{aligned}\vec{G}_{\text{opt}} &= \arg\max_{\vec{G}} P(\vec{G} \mid D) = \arg\max_{\vec{G}} \frac{P(\vec{G}, D)}{P(D)} \\ &= \arg\max_{\vec{G}} P(\vec{G}, D)\end{aligned}$$

$\Rightarrow$ Find an equation for $P(\vec{G}, D)$.

- In order to compare two graphs, it is sufficient to compute the **Bayes factor**

$$\frac{P(\vec{G}_1 \mid D)}{P(\vec{G}_2 \mid D)} = \frac{P(\vec{G}_1, D)}{P(\vec{G}_2, D)}$$

In both ways one can avoid computing the probability $P(D)$.
Assuming equal probability of all graphs simplifies further.

Model Averaging

We first consider $P(\vec{G}, D)$ to be the marginalization of $P(\vec{G}, \Theta, D)$ over all possible parameters Θ.

$$
\begin{aligned}
P(\vec{G}, D) &= \int_\Theta P(\vec{G}, \Theta, D)\,\mathrm{d}\Theta \\
&= \int_\Theta P(D \mid \vec{G}, \Theta)\, P(\vec{G}, \Theta)\,\mathrm{d}\Theta \\
&= \int_\Theta P(D \mid \vec{G}, \Theta)\, f(\Theta \mid \vec{G}) P(\vec{G})\,\mathrm{d}\Theta \\
&= \underbrace{P(\vec{G})}_{\text{A priori prob.}} \int_\Theta \underbrace{P(D \mid \vec{G}, \Theta)}_{\text{Likelihood of } D} \underbrace{f(\Theta \mid \vec{G})}_{\text{Parameter densities}} \,\mathrm{d}\Theta
\end{aligned}
$$

K2 Algorithm

- The a priori distribution $P(\vec{G})$ can be used to bias the evaluation measure towards user-specific network structures.

- Substitute the likelihood $P(D \mid \vec{G}, \Theta)$ for its specific form:

$$P(\vec{G}, D) \;=\; P(\vec{G}) \int_\Theta \underbrace{\left[\prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}}\right]}_{P(D|\vec{G},\Theta)} f(\Theta \mid \vec{G})\, \mathrm{d}\Theta$$

- See slide 300 for the derivation of the likelihood term.

K2 Algorithm

- The parameter densities $f(\Theta \mid \vec{G})$ describe the probabilities of the parameters given a network structure.

- They are densities of second order (distribution over distributions)

- For fixed i and j, a vector $(\theta_{ij1}, \ldots, \theta_{ijr_i})$ represents a probability distribution, namely the j-th column of the i-th potential table.

- Assuming mutual independence between the potential tables, we arrive for $f(\Theta \mid \vec{G})$ at the following:

$$f(\Theta \mid \vec{G}) \quad = \quad \prod_{i=1}^{n} \prod_{j=1}^{q_i} f(\theta_{ij1}, \ldots, \theta_{ijr_i})$$

K2 Algorithm

- Thus, we can further concretize the equation for $P(\vec{G}, D)$:

$$\begin{aligned} P(\vec{G}, D) &= P(\vec{G}) \underbrace{\int \cdots \int}_{\theta_{ijk}} \left[\prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}} \right] \cdot \left[\prod_{i=1}^{n} \prod_{j=1}^{q_i} f(\theta_{ij1}, \ldots, \theta_{ijr_i}) \right] \mathrm{d}\theta_{111}, \ldots, \mathrm{d}\theta_{nq_nr_n} \\ &= P(\vec{G}) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \underbrace{\int \cdots \int}_{\theta_{ijk}} \left[\prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}} \right] \cdot f(\theta_{ij1}, \ldots, \theta_{ijr_i}) \, \mathrm{d}\theta_{ij1}, \ldots, \mathrm{d}\theta_{ijr_i} \end{aligned}$$

K2 Algorithm

- A last assumption: For fixed i and j the density $f(\theta_{ij1}, \ldots, \theta_{ijr_i})$ is uniform:

$$f(\theta_{ij1}, \ldots, \theta_{ijr_i}) = (r_i - 1)!$$

- It simplifies $P(\vec{G}, D)$ further:

$$\begin{aligned} P(\vec{G}, D) &= P(\vec{G}) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \int_{\theta_{ijk}} \cdots \int \left[\prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}} \right] \cdot (r_i - 1)!\, \mathrm{d}\theta_{ij1}, \ldots, \mathrm{d}\theta_{ijr_i} \\ &= P(\vec{G}) \prod_{i=1}^{n} \prod_{j=1}^{q_i} (r_i - 1)! \qquad \underbrace{\int_{\theta_{ijk}} \cdots \int \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}}\, \mathrm{d}\theta_{ij1}, \ldots, \mathrm{d}\theta_{ijr_i}}_{\text{Dirichlet's integral} = \frac{\prod_{k=1}^{r_i} \alpha_{ijk}!}{(\sum_{k=1}^{r_i} \alpha_{ijk} + r_i - 1)!}} \end{aligned}$$

K2 Algorithm

- We finally arrive at an expression for $P(\vec{G}, D)$:

$$P(\vec{G}, D) \quad = \quad \mathrm{K2}(\vec{G} \mid D) \quad = \quad P(\vec{G}) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \left[\frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}! \right]$$

n	number of attributes describing the domain under consideration
r_i	number of values of the i-th attribute A_i, i. e., $r_i = \lvert\mathrm{dom}(A_i)\rvert$
q_i	number of instantiations of the parents of the i-th attribute in $\vec{G}$, i. e., $q_i = \prod_{A_j \in \mathrm{parents}(A_i)} r_i = \prod_{A_j \in \mathrm{parents}(A_i)} \lvert\mathrm{dom}(A_i)\rvert$
α_{ijk}	number of sample cases in which the i-th attribute has its k-th value and its parents in $\vec{G}$ have their j-th instantiation
N_{ij}	$= \sum_{k=1}^{r_i} \alpha_{ijk}$

Properties of the K2 Metric

- **Global** — Refers to the outer product: The total value of the K2 metric is the product over all K2 values of attribute families.
- **Local** — The likelihood equation assumes that given a parents instantiation, the probabilities for the respective child attribute values are mutual independent. This is reflected in the product over all q_i different parent attributes' value combinations of attribute A_i.

We exploit the global property to write the K2 metric as follows:

$$\mathrm{K2}(\vec{G} \mid D) \quad = \quad P(\vec{G}) \prod_{i=1}^{n} \mathrm{K2}_{\mathrm{local}}(A_i \mid D)$$

with

$$\mathrm{K2}_{\mathrm{local}}(A_i \mid D) \quad = \quad \prod_{j=1}^{q_i} \left[\frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}! \right]$$

K2 Algorithm

Prerequisites:

- Choose a topological order on the attributes $(A_1, \ldots, A_n)$
- Start out with a network that consists of n isolated nodes.
- Let ζ_i be the quality of the i-th attribute given the (tentative) set of parent attributes M:

$$\zeta_i(M) = \mathrm{K2}_{\mathrm{local}}(A_i \mid D) \quad \text{with} \quad \mathrm{parents}(A_i) = M$$

K2 Algorithm

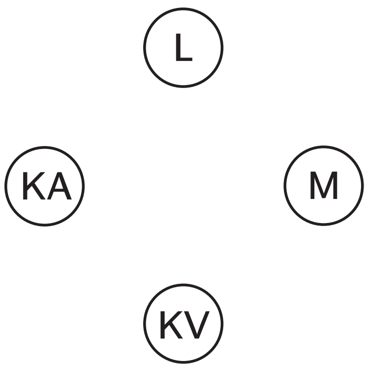
Execution:

1. Determine for the parentless node A_i the quality measure $\zeta_i(\emptyset)$
2. Evaluate for every predecessor $\{A_1, \ldots, A_{i-1}\}$ whether inserted as parent of A_i, the quality measure would increase. Let Y be the node that yields the highest quality (increase):
$$Y = \arg\max_{1 \leq l \leq i-1} \zeta_i(\{A_l\})$$
This best quality measure be $\zeta = \zeta_i(\{Y\})$.
3. If ζ is better than $\zeta_i(\emptyset)$, Y is inserted permanently as a parent node: $\text{parents}(A_i) = \text{parents}(A_i) \cup \{Y\}$
4. Repeat steps 2 and 3 to increase the parent set until no quality increase can be achieved or no nodes are left or a predefined maximum number of parent nodes per node is reached.

K2 Algorithm

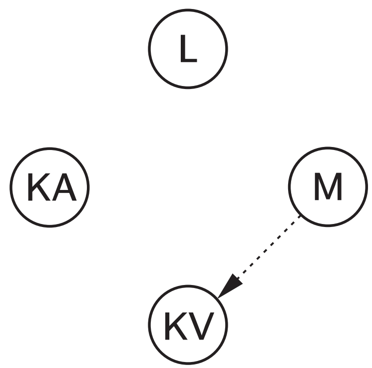
for $i \leftarrow 1 \ldots n$ **do** // Initialization
  $\text{parents}(A_i) \leftarrow \emptyset$
end for
for $i \leftarrow n, \ldots, 1$ **do** // Iteration
  repeat
    Select $Y \in \{A_1, \ldots, A_{i-1}\} \setminus \text{parents}(A_i)$,
      which maximizes $\zeta = \zeta_i(\text{parents}(A_i) \cup \{Y\})$
    $\delta \leftarrow \zeta - \zeta_i(\text{parents}(A_i))$
    if $\delta > 0$ **then**
      $\text{parents}(A_i) \leftarrow \text{parents}(A_i) \cup \{Y\}$
    end if
  until $\delta \leq 0$ or $\text{parents}(A_i) = \{A_1, \ldots, A_{i-1}\}$ or $|\text{parents}(A_i)| = n_{\max}$
end for

Demo of K2 Algorithm



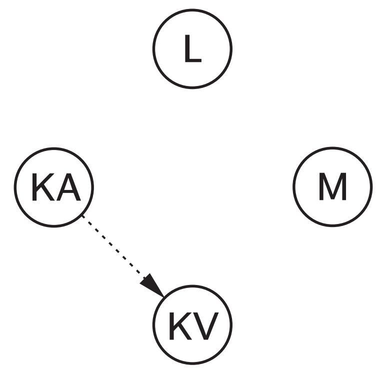
-43942.99

Step 1 – Edgeless graph



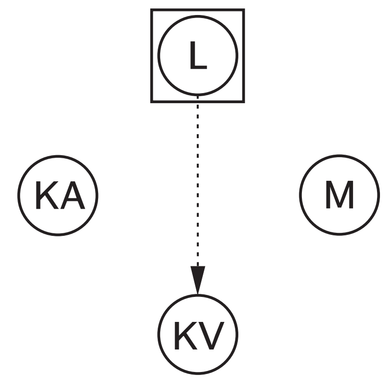
-43950.53

Step 2 – Insert M temporarily.



-43949.30

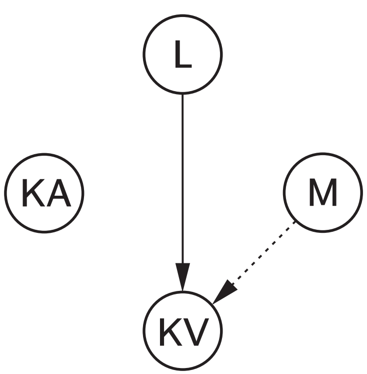
Step 3 – Insert KA temporarily.



-43942.48

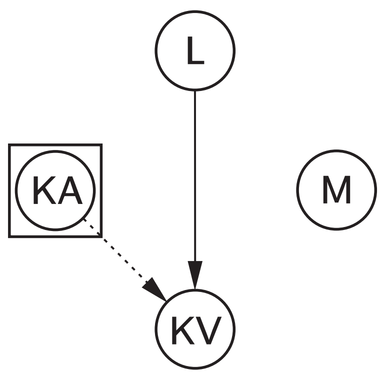
Step 4 – Node L maximizes K2 value and thus is added permantently.

Demo of K2 Algorithm



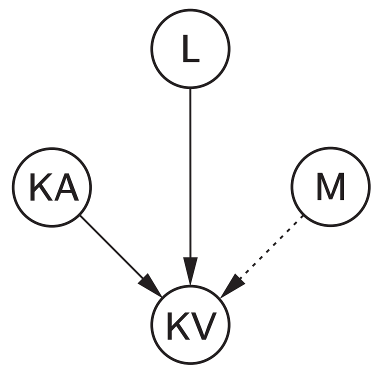
-43964.12

Step 5 – Insert M temporarily.



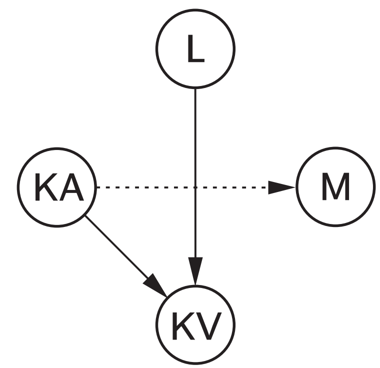
-43933.71

Step 6 – KA is added as second parent node of KV.



-43995.02

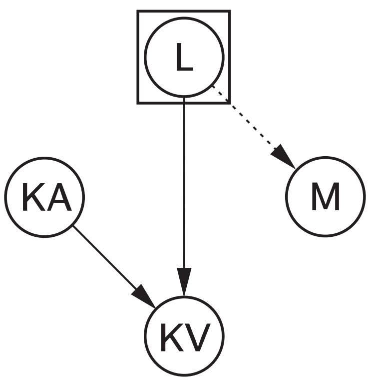
Step 7 – M does not increase the quality of the network if insertes as third parent node.



-43492.08

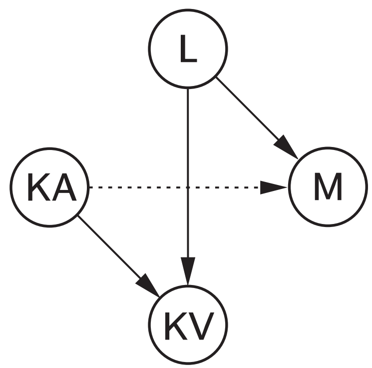
Step 8 – Insert KA temporarily.

Demo of K2 Algorithm



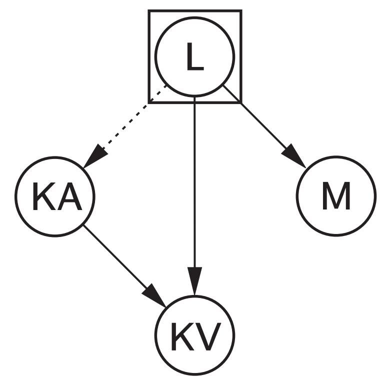
-40823.76

Step 9 – Node L becomes perent node of M.



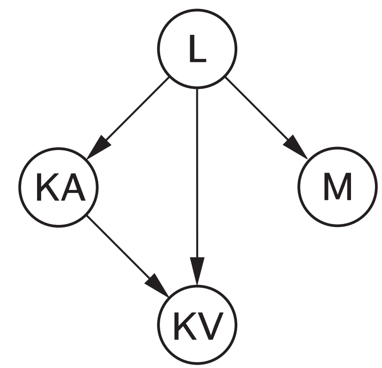
-40903.39

Step 10 – Adding KA does not increase overall network quaility.



-39190.67

Step 11 – Node L becomes parent node of KA.



Result

(A) Test whether a distribution is decomposable w. r. t. a given graph.

This is the most direct approach. It is not bound to a graphical representation, but can also be carried out w.r.t. other representations of the set of subspaces to be used to compute the (candidate) decomposition of the given distribution.

(B) Find a suitable graph by measuring the strength of dependences.

This is a heuristic, but often highly successful approach, which is based on the frequently valid assumption that in a conditional independence graph an attribute is more strongly dependent on adjacent attributes than on attributes that are not directly connected to them.

(C) Find an independence map by conditional independence tests.

This approach exploits the theorems that connect conditional independence graphs and graphs that represent decompositions. It has the advantage that a single conditional independence test, if it fails, can exclude several candidate graphs. However, wrong test results can thus have severe consequences.

Structure Learning with Conditional Independence Tests

General Idea: Exploit the theorems that connect conditional independence graphs and graphs that represent decompositions.

In other words: we want a graph describing a decomposition,
but we search for a conditional independence graph.

This approach has the advantage that a single conditional independence test, if it fails, can exclude several candidate graphs.

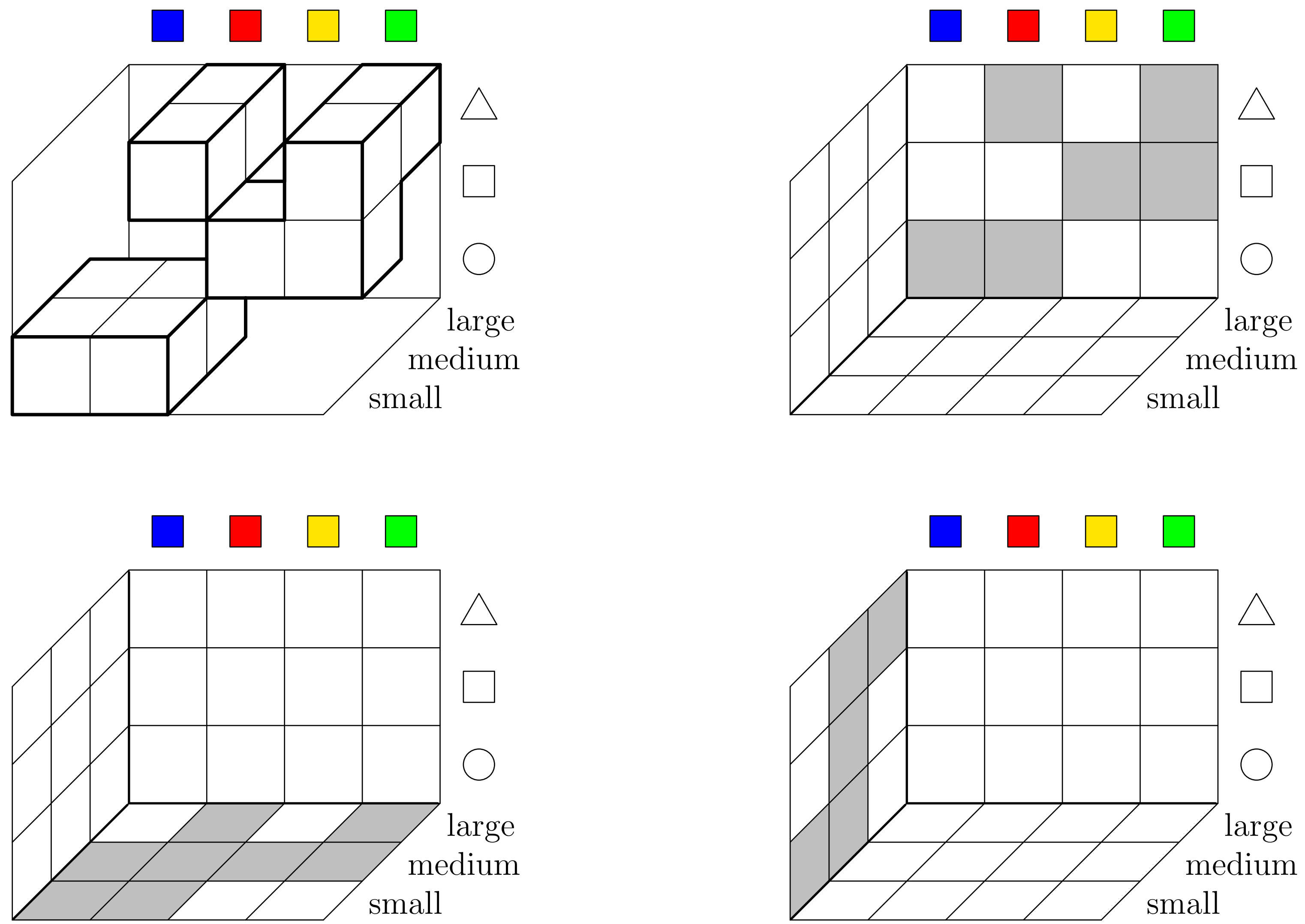
Assumptions:

- *Faithfulness:* The domain under consideration can be accurately described with a graphical model (more precisely: there exists a perfect map).

- *Reliability of Tests:* The result of all conditional independence tests coincides with the actual situation in the underlying distribution.

- Other assumptions that are specific to individual algorithms.

Conditional Independence Tests: Relational



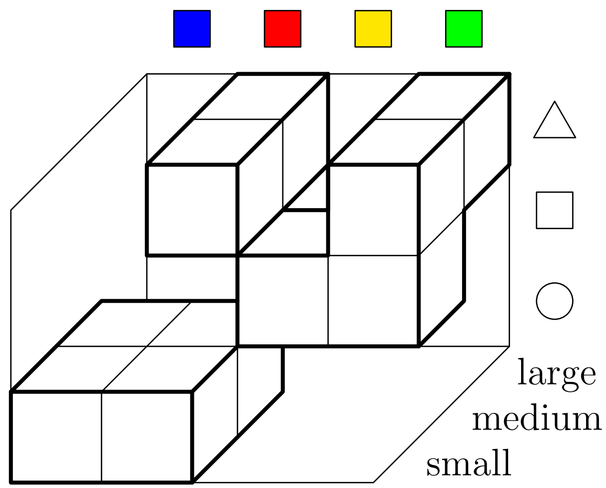
Conditional Independence Tests: Relational

- The Hartley information gain can be used directly to test for (approximate) **marginal independence**.

attributes	relative number of possible value combinations	Hartley information gain
color, shape	$\frac{6}{3\cdot 4} = \frac{1}{2} = 50\%$	$\log_2 3 + \log_2 4 - \log_2 6 = 1$
color, size	$\frac{8}{3\cdot 4} = \frac{2}{3} \approx 67\%$	$\log_2 3 + \log_2 4 - \log_2 8 \approx 0.58$
shape, size	$\frac{5}{3\cdot 3} = \frac{5}{9} \approx 56\%$	$\log_2 3 + \log_2 3 - \log_2 5 \approx 0.85$

- In order to test for (approximate) **conditional independence**:
  - Compute the Hartley information gain for each possible instantiation of the conditioning attributes.
  - Aggregate the result over all possible instantiations, for instance, by simply averaging them.

Conditional Independence Tests: Simple Example



color	Hartley information gain	
🟦	$\log_2 1 + \log_2 2 - \log_2 2 = 0$	
🟥	$\log_2 2 + \log_2 3 - \log_2 4 \approx 0.58$	
🟨	$\log_2 1 + \log_2 1 - \log_2 1 = 0$	
🟩	$\log_2 2 + \log_2 2 - \log_2 2 = 1$	
	average:	≈ 0.40

shape	Hartley information gain	
△	$\log_2 2 + \log_2 2 - \log_2 4 = 0$	
□	$\log_2 2 + \log_2 1 - \log_2 2 = 0$	
○	$\log_2 2 + \log_2 2 - \log_2 4 = 0$	
	average:	$= 0$

size	Hartley information gain	
large	$\log_2 2 + \log_2 1 - \log_2 2 = 0$	
medium	$\log_2 4 + \log_2 3 - \log_2 6 = 1$	
small	$\log_2 2 + \log_2 1 - \log_2 2 = 0$	
	average:	≈ 0.33

Conditional Independence Tests: Simple Example

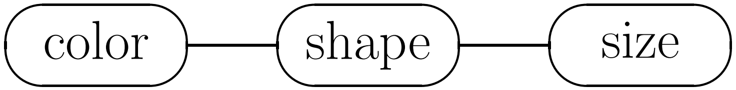
- The Shannon information gain can be used directly to test for (approximate) **marginal independence**.

- Conditional independence tests may be carried out by summing the information gain for all instantiations of the conditioning variables:

$$
\begin{aligned}
&I_{\text{gain}}(A, B \mid C) \\
&\quad = \sum_{c \in \text{dom}(C)} P(c) \sum_{a \in \text{dom}(A)} \sum_{b \in \text{dom}(B)} P(a, b \mid c) \, \log_2 \frac{P(a, b \mid c)}{P(a \mid c) \, P(b \mid c)},
\end{aligned}
$$

  where $P(c)$ is an abbreviation of $P(C = c)$ etc.

- Since $I_{\text{gain}}(\text{color}, \text{size} \mid \text{shape}) = 0$ indicates the only conditional independence, we get the following learning result:



Conditional Independence Tests: General Algorithm

Algorithm: (conditional independence graph construction)

1. For each pair of attributes A and B, search for a set $S_{AB} \subseteq U \backslash \{A, B\}$ such that $A \perp\!\!\!\perp B \mid S_{AB}$ holds in $\widehat{P}$, i.e., A and B are independent in $\widehat{P}$ conditioned on S_{AB}. If there is no such S_{AB}, connect the attributes by an undirected edge.

2. For each pair of non-adjacent variables A and B with a common neighbour C (i.e., C is adjacent to A as well as to B), check whether $C \in S_{AB}$.
   - If it is, continue.
   - If it is not, add arrow heads pointing to C, i.e., $A \rightarrow C \leftarrow B$.

3. Recursively direct all undirected edges according to the rules:
   - If for two adjacent variables A and B there is a strictly directed path from A to B not including $A \rightarrow B$, then direct the edge towards B.
   - If there are three variables A, B, and C with A and B not adjacent, $B - C$, and $A \rightarrow C$, then direct the edge $C \rightarrow B$.

Conditional Independence Tests: Simple Example

Suppose that the following conditional independence statements hold:

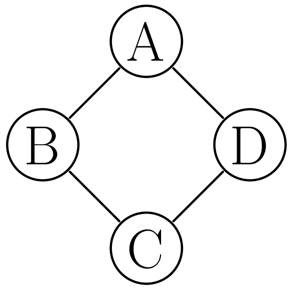
$$\begin{array}{ll} A \perp\!\!\!\perp_{\hat{P}} B \mid \emptyset & \qquad B \perp\!\!\!\perp_{\hat{P}} A \mid \emptyset \\ A \perp\!\!\!\perp_{\hat{P}} D \mid C & \qquad D \perp\!\!\!\perp_{\hat{P}} A \mid C \\ B \perp\!\!\!\perp_{\hat{P}} D \mid C & \qquad D \perp\!\!\!\perp_{\hat{P}} B \mid C \end{array}$$

All other possible conditional independence statements that can be formed with the attributes A, B, C, and D (with single attributes on the left) do not hold.

- **Step 1:** Since there is no set rendering A and C, B and C and C and D independent, the edges $A - C$, $B - C$, and $C - D$ are inserted.
- **Step 2:** Since C is a common neighbor of A and B and we have $A \perp\!\!\!\perp_{\hat{P}} B \mid \emptyset$, but $A \not\perp\!\!\!\perp_{\hat{P}} B \mid C$, the first two edges must be directed $A \rightarrow C \leftarrow B$.
- **Step 3:** Since A and D are not adjacent, $C - D$ and $A \rightarrow C$, the edge $C - D$ must be directed $C \rightarrow D$.
  (Otherwise step 2 would have already fixed the orientation $C \leftarrow D$.)

Conditional Independence Tests: Drawbacks

- The conditional independence graph construction algorithm presupposes that there is a **perfect map**. If there is no perfect map, the result may be invalid.



p_{ABCD}		$A = a_1$, $B = b_1$	$A = a_1$, $B = b_2$	$A = a_2$, $B = b_1$	$A = a_2$, $B = b_2$
$C = c_1$	$D = d_1$	$^1/_{47}$	$^1/_{47}$	$^1/_{47}$	$^2/_{47}$
	$D = d_2$	$^1/_{47}$	$^1/_{47}$	$^2/_{47}$	$^4/_{47}$
$C = c_2$	$D = d_1$	$^1/_{47}$	$^2/_{47}$	$^1/_{47}$	$^4/_{47}$
	$D = d_2$	$^2/_{47}$	$^4/_{47}$	$^4/_{47}$	$^{16}/_{47}$

- **Independence tests of high order**, i. e., with a large number of conditions, may be necessary.

- There are approaches to mitigate these drawbacks.
(For example, the order is restricted and all tests of higher order are assumed to fail, if all tests of lower order failed.)

The Cheng–Bell–Liu Algorithm

- **Drafting:** Build a so-called Chow–Liu tree as an initial graphical model.
  - Evaluate all attribute pairs (candidate edges) with information gain.
  - Discard edges with evaluation below independence threshold ($\sim$0.1 bits).
  - Build optimum (maximum) weight spanning tree.
- **Thickening:** Add necessary edges.
  - Traverse remaining candidate edges in the order of decreasing evaluation.
  - Test for conditional independence in order to determine whether an edge is needed in the graphical model.
  - Use local Markov property to select a condition set: an attribute is conditionally independent of all non-descendants given its parents.
  - Since the graph is undirected in this step, the set of adjacent nodes is reduced iteratively and greedily in order to remove possible children.

The Cheng–Bell–Liu Algorithm (continued)

- **Thinning:** Remove superfluous edges.
  - In the thickening phase a conditional independence test may have failed, because the graph was still too sparse.
  - Traverse all edges that have been added to the current graphical model and test for conditional independence.
  - Remove unnecessary edges.
    (two phases/approaches: heuristic test/strict test)
- **Orienting:** Direct the edges of the graphical model.
  - Identify the v-structures (converging directed edges).
    (Markov equivalence: same skeleton and same set of v-structures.)
  - Traverse all pairs of attributes with common neighbors and check which common neighbors are in the (maximally) reduced set of conditions.
  - Direct remaining edges by extending chains and avoiding cycles.

Learning Undirected Graphical Models Directly

- **Drafting:** Build a Chow–Liu tree as an initial graphical model
  - Evaluate all attribute pairs (candidate edges) with specificity gain.
  - Discard edges with evaluation below independence threshold ($\sim$0.015).
  - Build optimum (maximum) weight spanning tree.
- **Thickening:** Add necessary edges.
  - Traverse remaining candidate edges in the order of decreasing evaluation.
  - Test for conditional independence in order to determine whether an edge is needed in the graphical model.
  - Use local Markov property to select a condition set: an attribute is conditionally independent of any non-neighbor given its neighbors.
  - Since the graphical model to be learned is undirected, *no (iterative) reduction of the condition set is needed* (decisive difference to Cheng–Bell–Liu Algorithm).

Learning Undirected Graphical Models Directly

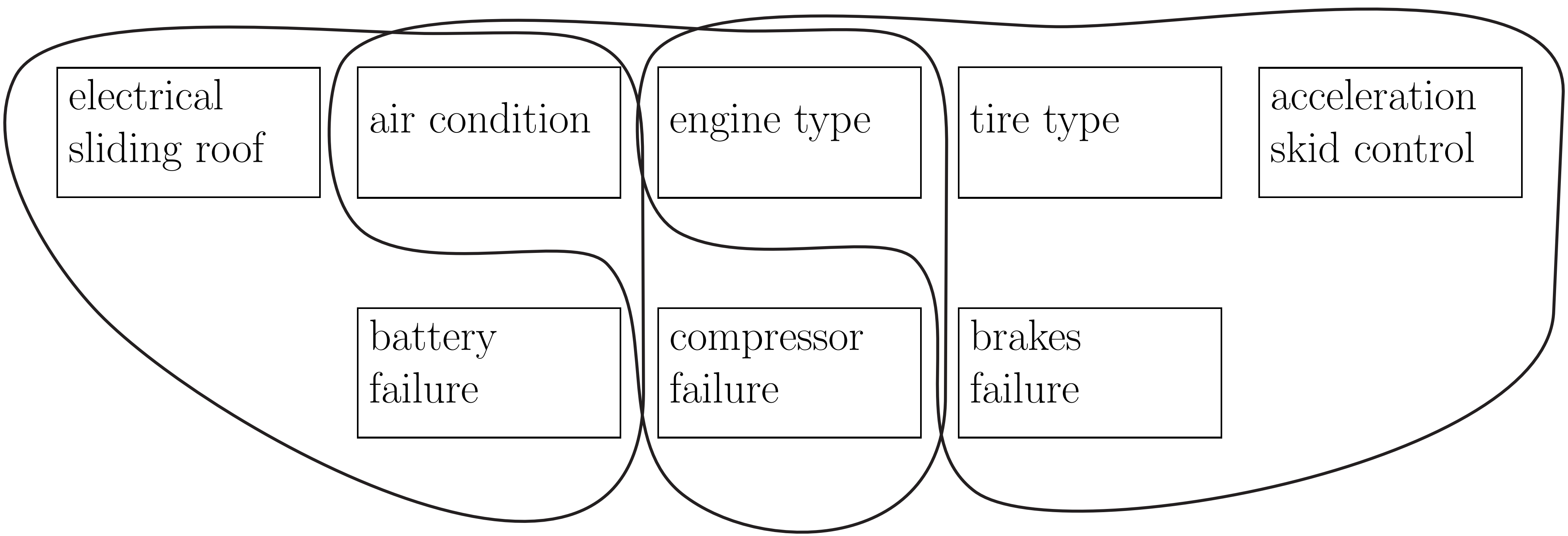
- **Moralizing:** Take care of possible v-structures.
  - If one assumes a perfect undirected map, this step is unnecessary. However, v-structures are too common and cannot be represented without loss in an undirected graphical model.
  - Possible v-structures can be taken care of by connecting the parents.
  - Traverse all edges with an evaluation below the independence threshold that have a common neighbor in the graph.
  - Add edge if conditional independence given the neighbors does not hold.
- **Thinning:** Remove superfluous edges.
  - In the thickening phase a conditional independence test may have failed, because the graph was still too sparse.
  - Traverse all edges that have been added to the current graphical model and test for conditional independence.

Application at Daimler AG

- **Improving the Product Quality by Detecting Weaknesses**
  - Learn a decision tree or inference network for vehicle properties and failures.
  - Look for suspicious conditional failure rates.
  - Find causes of these suspicious rates.
  - Optimize design of vehicle.
- **Improve the Error Diagnosis in Service Garages**
  - Learn a decision tree or inference network for vehicle properties and failures.
  - Record new faults.
  - Test for most probable errors.

Analysis of the Daimler Database

- Database: approx. 18500 vehicles with more than 100 attributes
- Analysis of dependencies between **specific equipment** and **failure**.
- Results are used as a starting point for technical investigation.



Fictitious example: There are significantly more battery failures, if an aircondition and an electrical sliding roof are installed.

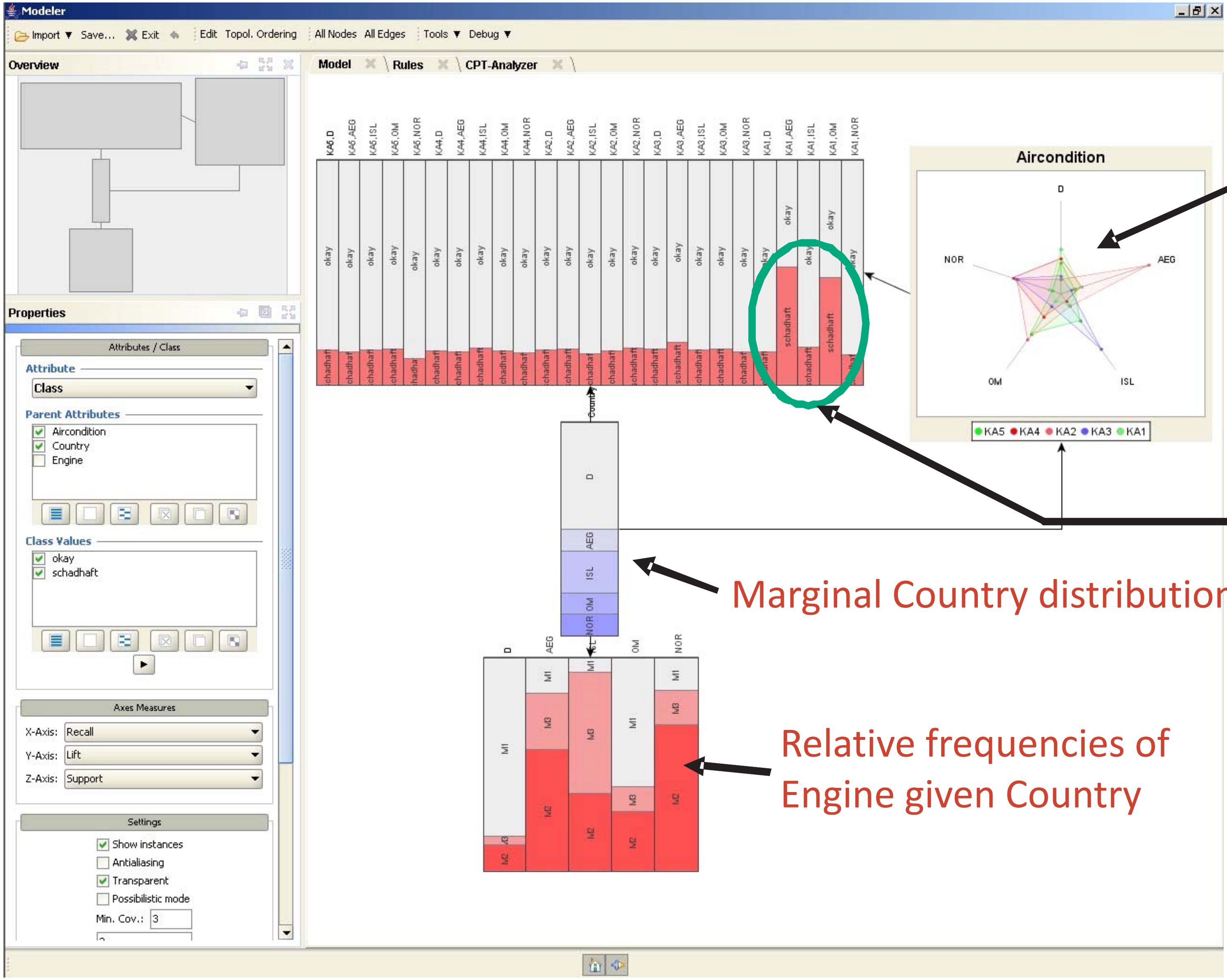
Example Network

Influence of specific equipment on battery failure:

(fictitious) battery failure rate		Aircondition	
		with	without
elec. sliding roof	with	8%	3%
	without	3%	2%

- Significant deviation from independent distribution.
- Hint for possible causes.
- Here: Larger battery might be required if both aircondition *and* electrical sliding roof are installed.

Explorative Data Analysis

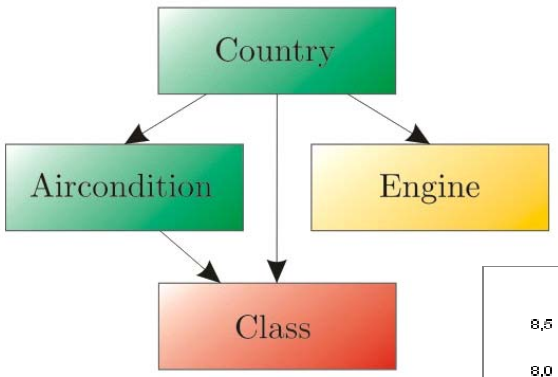


Ratio between (marginal) Aircondition sale rate and sale rate given the Country

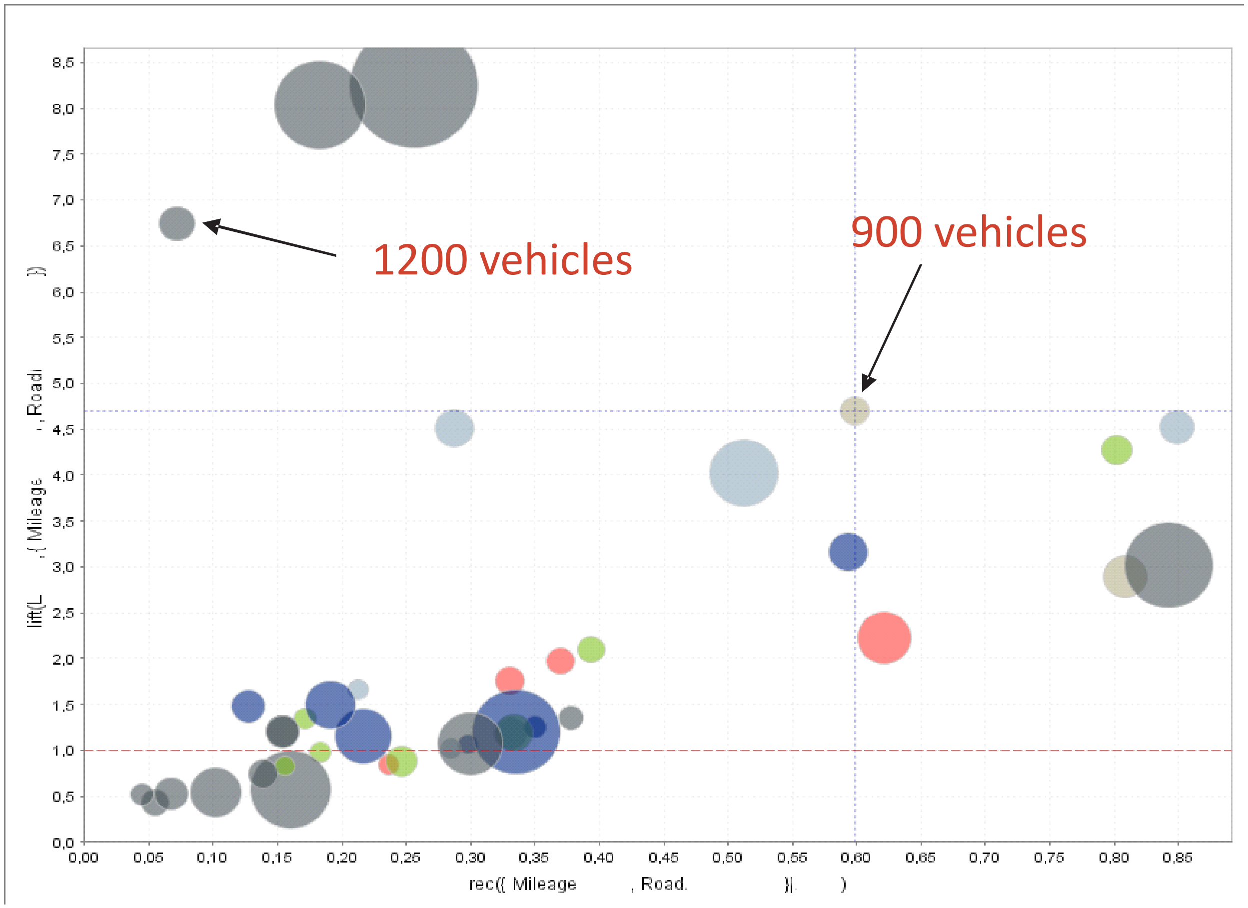
Airconditions of type 1 fail much more often in Egypt and Oman

Marginal Country distribution

Relative frequencies of Engine given Country



Subnet of the entire network



Partition of the entire vehicle set according to the network's potential tables.

Decision Graphs / Influence Diagrams

Preference Orderings

- A *preference ordering* $\succeq$ is a ranking of all possible states of affairs (worlds) S
  - these could be outcomes of actions, truth assts, states in a search problem, etc.
  - $s \succeq t$: means that state s is *at least as good as* t
  - $s \succ t$: means that state s *is strictly preferred to* t
- We insist that $\succeq$ is
  - reflexive: i.e., s $\succeq$ s for all states s
  - transitive: i.e., if s $\succeq$ t and t $\succeq$ w, then s $\succeq$ w
  - connected: for all states s,t, either s $\succeq$ t or t $\succeq$ s

Why Impose These Conditions?

- Structure of preference ordering imposes certain "rationality requirements" (it is a weak ordering)
- E.g., why transitivity?
  - Suppose you (strictly) prefer coffee to tea, tea to OJ, OJ to coffee
  - If you prefer X to Y, you will trade me Y plus $1 for X
  - I can construct a "money pump" and extract arbitrary amounts of money from you

Utilities

- Rather than just ranking outcomes, we must quantify our degree of preference
  - e.g., how much more important is *chc* than $\sim$*mess*
- A *utility function* $U : S \to \mathbb{R}$ associates a realvalued *utility* with each outcome.
  - $U(s)$ measures your *degree* of preference for s
- Note: U induces a preference ordering $\succeq_U$ over S defined as: s $\succeq_U$ t iff $U(s) \geq U(t)$
  - obviously $\succeq_U$ will be reflexive, transitive, connected

Expected Utility

- Under conditions of uncertainty, each decision d induces a distribution Pr_d over possible outcomes
  - $Pr_d(s)$ is probability of outcome s under decision d

- The *expected utility* of decision d is defined

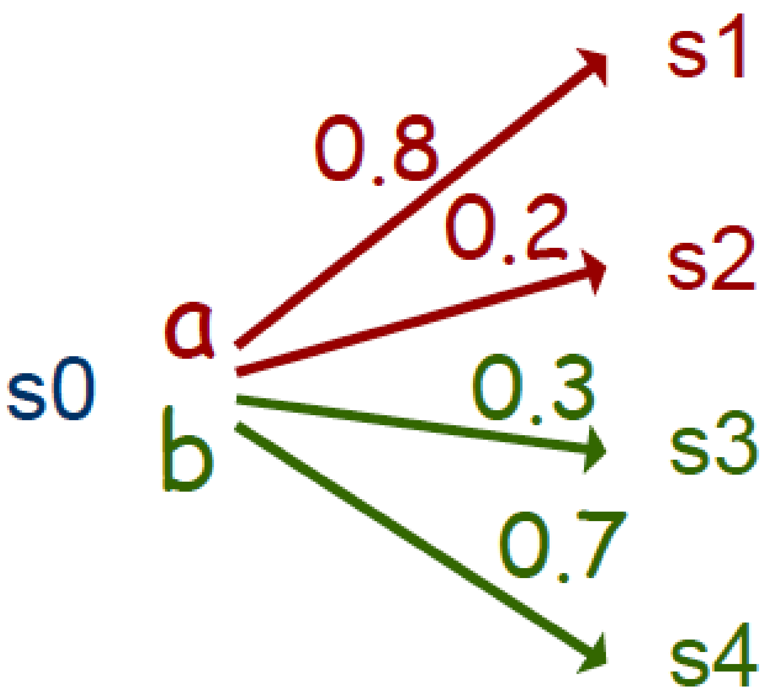
- The *principle of maximum expected utility (MEU)* states that the optimal decision under conditions of uncertainty is that with the greatest expected utility.

$$EU(d) = \sum_{s \in S} Pr_d(s) U(s)$$

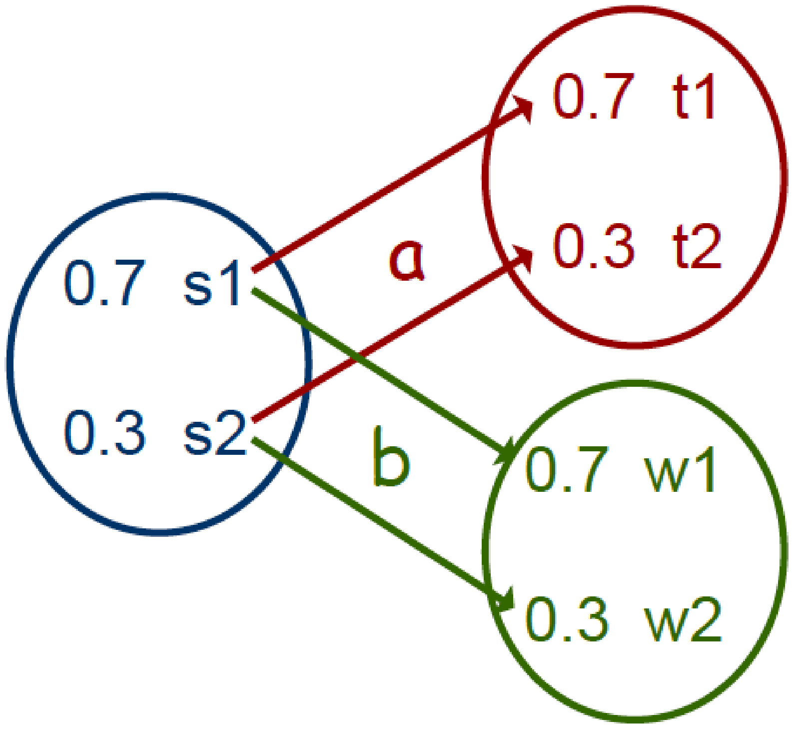
Decision Problems: Uncertainty

- A *decision problem under uncertainty* is:
  - a set of *decisions* D
  - a set of *outcomes* or states S
  - an *outcome function* $Pr : D \to \Delta(S)$
    - $\Delta(S)$ is the set of distributions over S (e.g., Prd)
  - a *utility function* U over S
- A solution to a decision problem under uncertainty is any $d^* \in D$ such that $EU(d^*) \succeq EU(d)$ for all $d \in D$
- Again, for single-shot problems, this is trivial

- Note that this viewpoint accounts for both:
  - uncertainty in action outcomes
  - uncertainty in state of knowledge
  - any combination of the two



Stochastic actions



Uncertain knowledge

Expected Utility: Notes

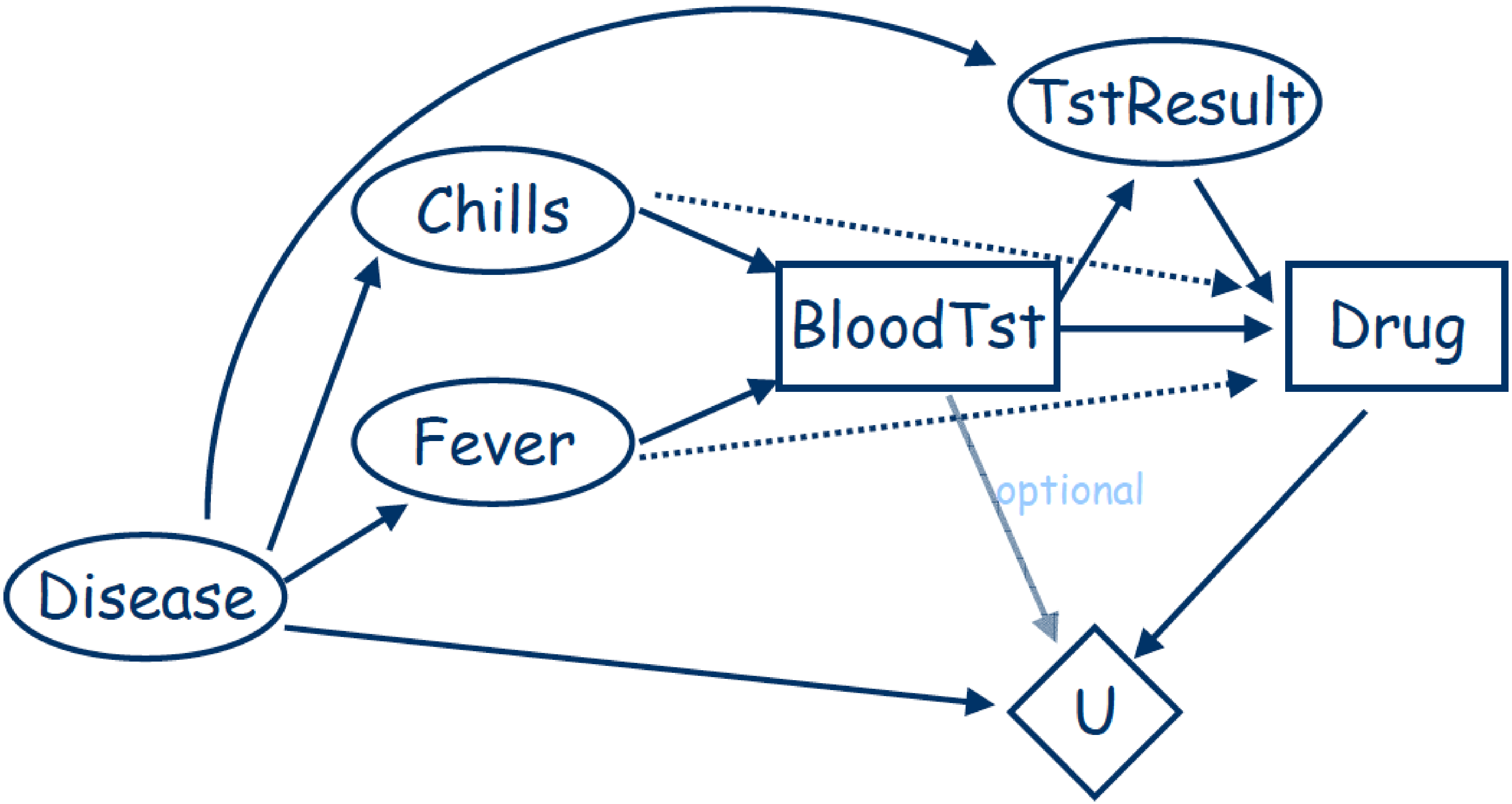
- Why MEU? Where do utilities come from?
  - underlying foundations of utility theory tightly couple utility with action/choice
  - a utility function can be determined by asking someone about their preferences for actions in specific scenarios (or "lotteries" over outcomes)
- Utility functions needn't be unique
  - if I multiply U by a positive constant, all decisions have same relative utility
  - if I add a constant to U, same thing
  - *U is unique up to positive affine transformation*

So What are the Complications?

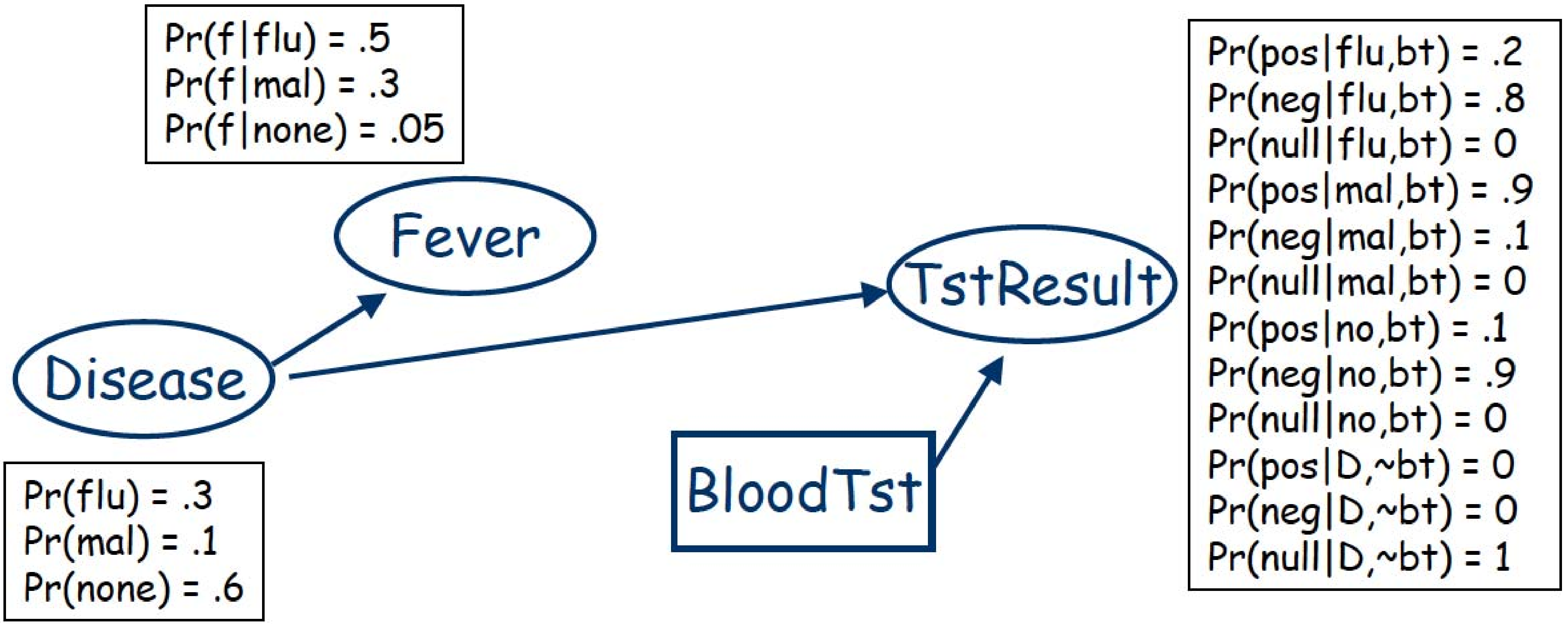
- Outcome space is large
  - like all of our problems, states spaces can be huge
  - don't want to spell out distributions like Pr_d explicitly
  - Solution: Bayes nets (or related: *influence diagrams*)
- Decision space is large
  - usually our decisions are not one-shot actions
  - rather they involve sequential choices (like plans)
  - if we treat each plan as a distinct decision, decision space is too large to handle directly
  - Soln: use dynamic programming methods to *construct* optimal plans (actually generalizations of plans, called policies. . . like in game trees)

So What are the Complications?

- *Decision networks* (more commonly known as *influence diagrams*) provide a way of representing sequential decision problems
  - basic idea: represent the variables in the problem as you would in a BN
  - add decision variables – variables that you "control"
  - add utility variables – how good different states are

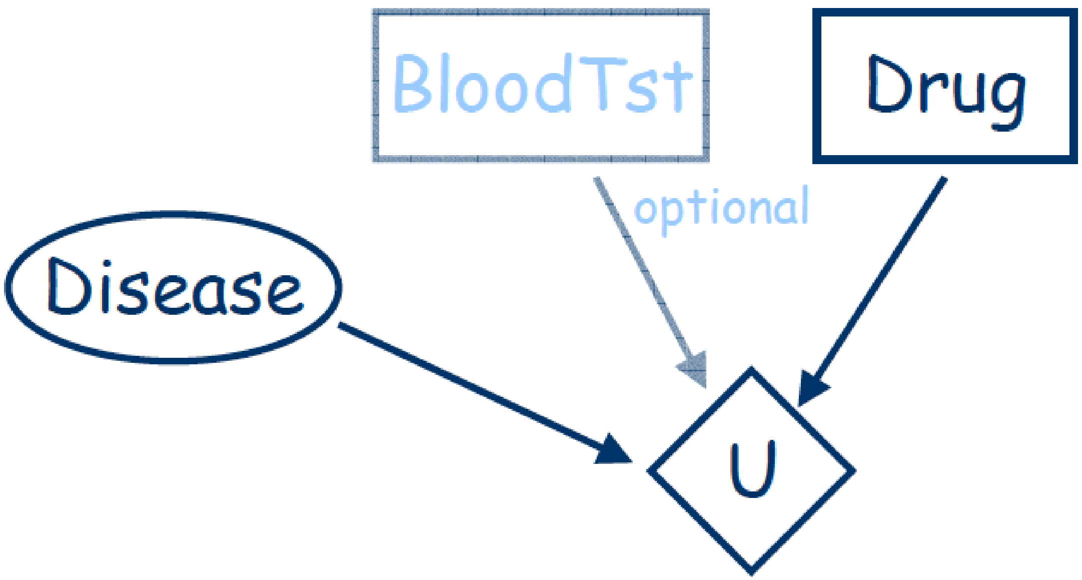

TstResult
Chills
BloodTst
Drug
Fever
optional
Disease
U

- **Chance nodes**
  - random variables, denoted by circles
  - as in a BN, probabilistic dependence on parents



- **Decision nodes**
  - variables decision maker sets, denoted by squares
  - parents reflect *information available* at time decision is to be made
- In example decision node: the actual values of Ch and Fev will be observed before the decision to take test must be made
  - agent can make different decisions for each instantiation of parents (i.e., policies)

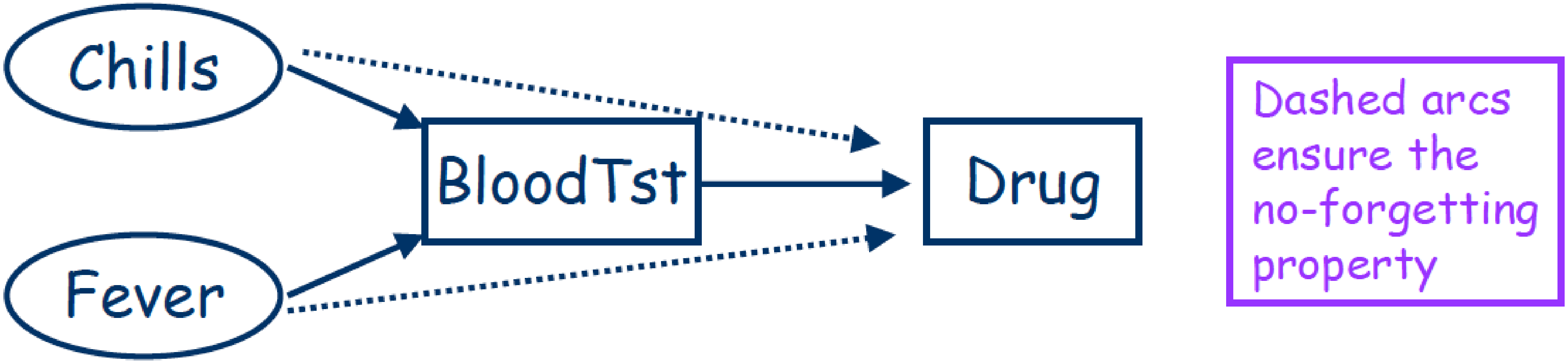
- **Value node**
  - specifies utility of a state, denoted by a diamond
  - utility depends *only on state of parents* of value node
  - generally: only one value node in a decision network
- Utility depends only on disease and drug



U(fludrug, flu) = 20
U(fludrug, mal) = -300
U(fludrug, none) = -5
U(maldrug, flu) = -30
U(maldrug, mal) = 10
U(maldrug, none) = -20
U(no drug, flu) = -10
U(no drug, mal) = -285
U(no drug, none) = 30

Decision Networks: Assumptions

- Decision nodes are totally ordered
  - decision variables $D_1, D_2, \ldots, D_n$
  - decisions are made in sequence
  - e.g., BloodTst (yes,no) decided before Drug (fd,md,no)
- *No-forgetting property*
  - any information available when decision D_i is made is available when decision D_j is made (for $i < j$)
  - thus all parents of D_i are parents of D_j



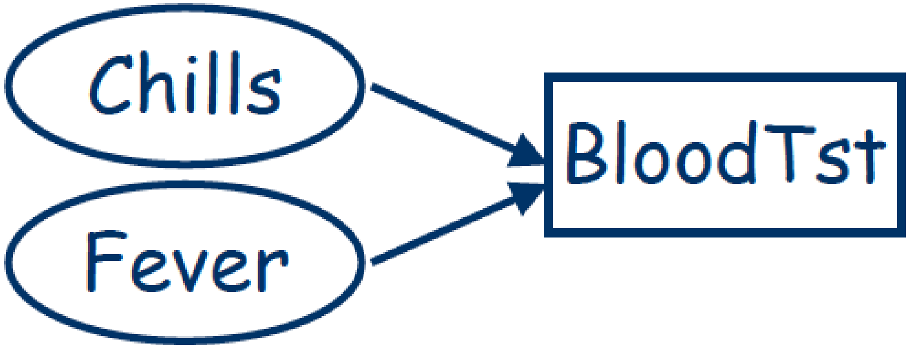
Policies

- Let $Par(D_i)$ be the parents of decision node D_i
  - $Dom(Par(Di))$ is the set of assignments to parents

- A policy δ is a set of mappings δ_i, one for each decision node D_i
  - $\delta_i : Dom(Par(D_i)) \to (D_i)$
  - δ_i associates a decision with each parent asst for D_i

- For example, a policy for BT might be:

$$\delta_{BT}(c, f) = bt$$
$$\delta_{BT}(c, \sim f) =\sim bt$$
$$\delta_{BT}(\sim c, f) = bt$$
$$\delta_{BT}(\sim c, \sim f) =\sim bt$$



Value of a Policy

- Value of a policy δ is the expected utility given that decision nodes are executed according to δ

- Given associates $\boldsymbol{x}$ to the set $\boldsymbol{X}$ of all chance variables, let $\delta(\boldsymbol{x})$ denote the asst to decision variables dictated by δ
  - e.g., asst to D_1 determined by it's parents' asst in $\boldsymbol{x}$
  - e.g., asst to D_2 determined by it's parents' asst in $\boldsymbol{x}$ along with whatever was assigned to D1
  - etc.

- Value of δ:

$$EU(\delta) = \sum_{\boldsymbol{X}} P(\boldsymbol{X}, \delta(\boldsymbol{X}) U(\boldsymbol{X}, \delta(\boldsymbol{X}))$$

Optimal Policies

- An *optimal policy* is a policy δ^* such that $EU(\delta^*) \geq EU(\delta)$ for all policies δ
- We can use the dynamic programming principle yet again to avoid enumerating all policies
- We can also use the structure of the decision network to use variable elimination to aid in the computation

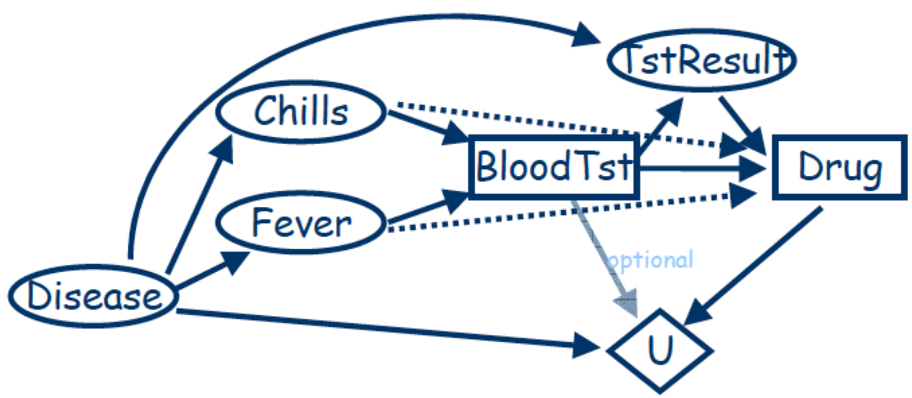
Computing the Best Policy

- We can work backwards as follows

- First compute optimal policy for Drug (last dec'n)
  - for each asst to parents (C,F,BT,TR) and for each decision value (D = md,fd,none), *compute the expected value* of choosing that value of D

  - set policy choice for each value of parents to be the value of D that has max value

  - eg: $\delta_D(c, f, bt, pos) = md$

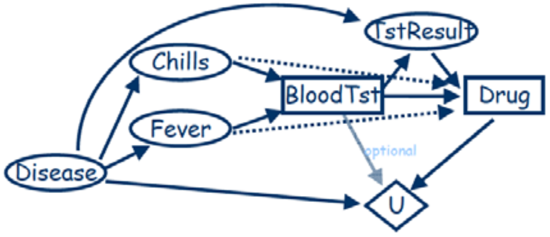


Computing the Best Policy

- Next compute policy for BT given policy $\delta_D(C, F, BT, TR)$ just determined for Drug
  - since $\delta_D(C, F, BT, TR)$ is fixed, we can treat Drug as a normal random variable with deterministic probabilities
  - i.e., for any instantiation of parents, value of Drug is fixed by policy δ_D
  - this means we can solve for optimal policy for BT just as before
  - only uninstantiated vars are random vars (once we fix *its* parents)

Computing the Best Policy

- How do we compute these expected values?
  - suppose we have asst $< c, f, bt, pos >$ to parents of $Drug$
  - we want to compute EU of deciding to set $Drug = md$
  - we can run variable elimination!
- Treat C, F, BT, TR, Dr as evidence
  - this reduces factors (e.g., U restricted to bt, md: depends on Dis)
  - eliminate remaining variables (e.g., only Disease left)
  - left with factor: $U() = \sum_{Dis} P(Dis|c, f, bt, pos, md)U(Dis)$
- We now know EU of doing $Dr = md$ when c, f, bt, pos true
- Can do same for fd, no to decide which is best

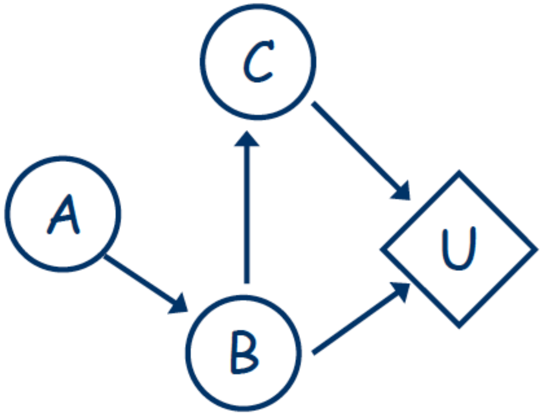


Computing Expected Utilities

- The preceding illustrates a general phenomenon
  - computing expected utilities with BNs is quite easy
  - utility nodes are just factors that can be dealt with using variable elimination

$$\begin{aligned} EU &= \sum_{A,B,C} P(A,B,C)U(B,C) \\ &= \sum_{A,B,C} P(C|B)P(B|A)P(A)U(B,C) \end{aligned}$$

- Just eliminate variables in the usual way



Optimizing Policies: Key Points

- If a decision node D has no decisions that follow it, we can find its policy by instantiating each of its parents and computing the expected utility of each decision for each parent instantiation
  - no-forgetting means that all other decisions are instantiated (they must be parents)
  - its easy to compute the expected utility using VE
  - the number of computations is quite large: we run expected utility calculations (VE) for each parent instantiation together with each possible decision D might allow
  - policy: choose max decision for each parent instant'n

Optimizing Policies: Key Points

- When a decision D node is optimized, it can be treated as a random variable
  - for each instantiation of its parents we now know what value the decision should take
  - just treat policy as a new CPT: for a given parent instantiation $\boldsymbol{x}$, D gets $\delta(\boldsymbol{x})$ with probability 1 (all other decisions get probability zero)
- If we optimize from last decision to first, at each point we can optimize a specific decision by (a bunch of) simple VE calculations
  - it's successor decisions (optimized) are just normal nodes in the BNs (with CPTs)

Decision Network Notes

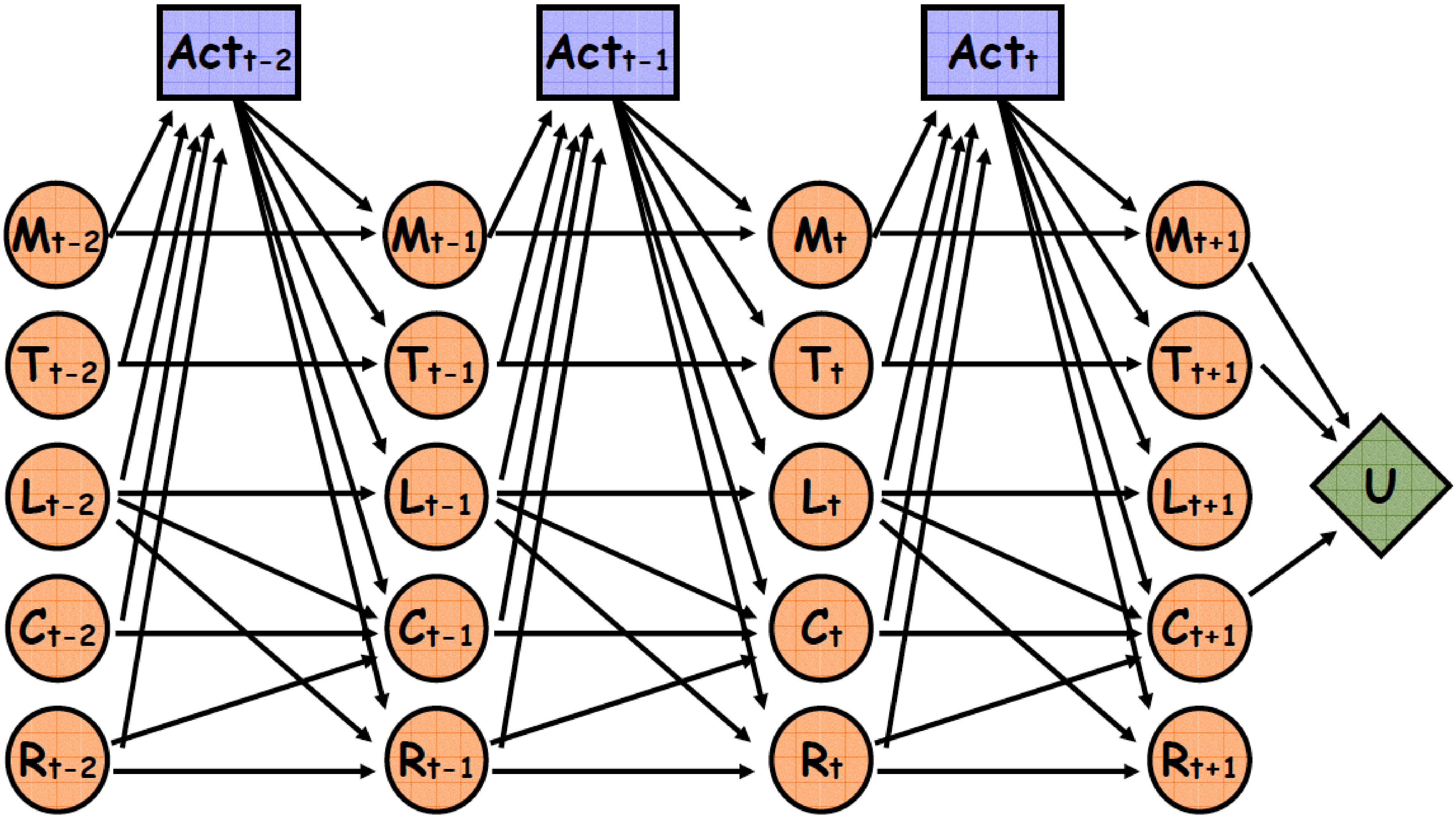
- Decision networks commonly used by decision analysts to help structure decision problems

- Much work put into computationally effective techniques to solve these
  - common trick: replace the decision nodes with random variables at outset and solve a plain Bayes net (a subtle but useful transformation)

- Complexity much greater than BN inference
  - we need to solve a number of BN inference problems

  - one BN problem for each setting of decision node parents and decision node value

DBN-Decision Nets for Planning



Decision Network Notes

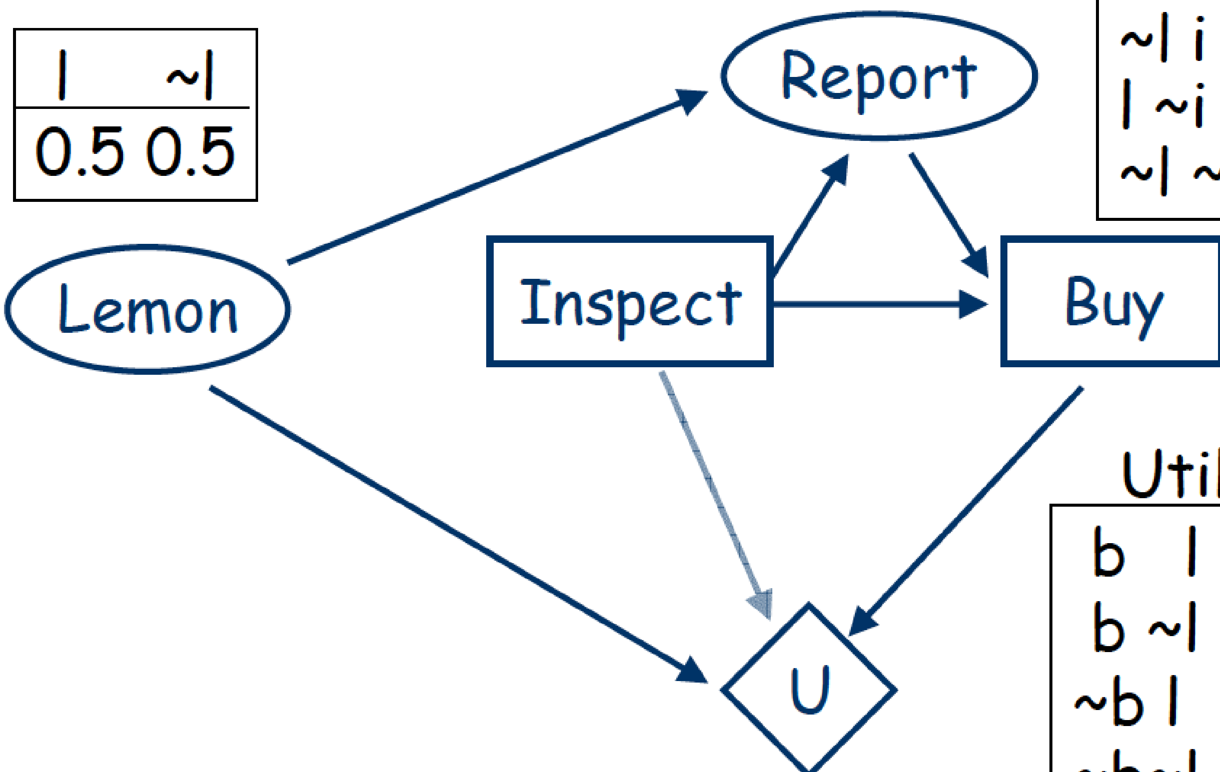
- In example on previous slide:
  - we assume the state (of the variables at any stage) is fully observable
    * hence all time t vars point to time t decision
  - this means the state at time t d-separates the decision at time t-1 from the decision at time t-2
  - so we ignore "no-forgetting" arcs between decisions
    * once you *know* the state at time t, what you *did* at time t-1 to get there is irrelevant to the decision at time t-1
- If the state were not fully observable, we could not ignore the "no-forgetting" arcs

A Detailed Decision Net Example

- Setting: you want to buy a used car, but there's a good chance it is a "lemon" (i.e., prone to breakdown). Before deciding to buy it, you can take it to a mechanic for inspection. S/he will give you a report on the car, labelling it either "good" or "bad". A good report is positively correlated with the car being sound, while a bad report is positively correlated with the car being a lemon.

- The report costs \$50 however. So you could risk it, and buy the car without the report.

- Owning a sound car is better than having no car, which is better than owning a lemon.



l	~l
0.5	0.5

Rep: good,bad,none

	g	b	n
l i	0.2	0.8	0
~l i	0.9	0.1	0
l ~i	0	0	1
~l ~i	0	0	1

Utility

b l	-600
b ~l	1000
~b l	-300
~b~l	-300

-50 if inspect

Evaluate Last Decision: Buy (1)

- $EU(B|I,R) = \sum_L P(L|I,R,B)U(L,B)$

- $I = i, R = g$:

$$\begin{aligned}
EU(buy) &= P(l|i,g)U(l,buy) + P(\sim l|i,g)U(\sim l,buy) - 50 \\
&= .18 \cdot -600 + .82 \cdot 1000 - 50 = 662 \\
EU(\sim buy) &= P(l|i,g)U(l,\sim buy) + P(\sim l|i,g)U(\sim l,\sim buy) - 50 \\
&= -300 - 50 = -350 (-300 \text{ indep. of lemon})
\end{aligned}$$

- So optimal $\delta_{Buy}(i,g) = buy$

Evaluate Last Decision: Buy (2)

- $I = i, R = b$:

$$
\begin{aligned}
EU(buy) &= P(l|i,b)U(l,buy) + P(\sim l|i,b)U(\sim l,buy) - 50 \\
&= .89 \cdot -600 + .11 \cdot 1000 - 50 = -474 \\
EU(\sim buy) &= P(l|i,b)U(l,\sim buy) + P(\sim l|i,b)U(\sim l,\sim buy) - 50 \\
&= -300 - 50 = -350 (-300 \text{ indep. of lemon})
\end{aligned}
$$

- So optimal $\delta_{Buy}(i,b) = \sim buy$

Evaluate Last Decision: Buy (3)

- $I = \sim i, R = g$ (note: no inspection cost subtracted):

$$\begin{aligned} EU(buy) &= P(l| \sim i, g)U(l, buy) + P(\sim l| \sim i, g)U(\sim l, buy) \\ &= .5 \cdot -600 + .5 \cdot 1000 = 200 \\ EU(\sim buy) &= P(l| \sim i, g)U(l, \sim buy) + P(\sim l| \sim i, g)U(\sim l, \sim buy) - 50 \\ &= -300 - 50 = -350 (-300 \text{ indep. of lemon}) \end{aligned}$$

- So optimal $\delta_{Buy}(\sim i, g) = \sim buy$

- So optimal policy for Buy is:
  - $\delta_{Buy}(i, g) = buy; \delta_{Buy}(i, b) = \sim buy; \delta_{Buy}(\sim i, n) = buy$

- Note: we don't bother computing policy for $(i, \sim n)$, $(\sim i, g)$, or $(\sim i, b)$, since these occur with probability 0

Evaluate First Decision: Inspect

- $EU(I) = \sum_{L,R} P(L,R|I)U(L, \boldsymbol{\delta_{Buy}(I,R)})$
  - where $P(R,L|I) = P(R|L,I)P(L|I)$

$$
\begin{aligned}
EU(i) &= .1 \cdot -600 + .4 \cdot -300 + .45 \cdot 1000 + .05 \cdot -300 - 50 \\
&= 237.5 - 50 = 187.5 \\
EU(\sim i) &= P(l|\sim i, n)U(l, buy) + P(\sim l|\sim i, n)U(\sim l, buy) \\
&= .5 \cdot -600 + .5 \cdot 1000 = 200
\end{aligned}
$$

- So optimal $\delta_{Inspect}(\sim i) = buy$

	$P(R,L\|I)$	$\boldsymbol{\delta_{Buy}}$	$U(L, \boldsymbol{\delta_{Buy}})$
g, l	0.1	buy	$-600 - 50 = -650$
$g, \sim l$	0.45	buy	$1000 - 50 = 950$
b, l	0.4	$\sim buy$	$-300 - 50 = -350$
$b, \sim l$	0.05	$\sim buy$	$-300 - 50 = -350$

Value of Information

- So optimal policy is: don't inspect, buy the car
  - EU = 200
  - Notice that the EU of inspecting the car, then buying it iff you get a good report, is 237.5 less the cost of the inspection (50). So inspection not worth the improvement in EU.
  - But suppose inspection cost \$25: then it would be worth it ($EU = 237.5 - 25 = 212.5 > EU(\sim i)$)
  - The *expected value of information* associated with inspection is 37.5 (it improves expected utility by this amount ignoring cost of inspection). How? Gives opportunity to change decision ($\sim buy$ if bad).
  - You should be willing to pay up to \$37.5 for the report

Influence Diagrams

Up to now, we used Bayesian networks for

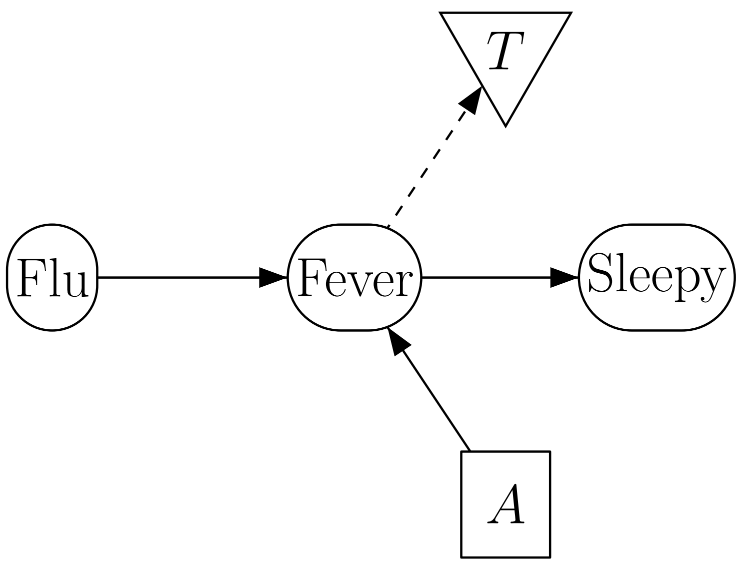
- modeling (in)dependence relations between random/chance variables
- quantifying the strength of these relations by assigning (conditional) probabilities
- update these probabilities after evidence observations

However, in practical, this is only a part of a more complex task: **decision making under uncertainty**.

If a set of actions solves a problem, we have to choose one particular action based on predefined criteria, e. g. costs and/or gains.

Therefore, we will now augment the current framework with special nodes that serve these purposes.

Example: Observations and Actions



$T \ldots$ Temperature
$A \ldots$ Aspirine

- Rectangular nodes: intervening actions/decisions
- Triangular nodes: test actions/observations
- Observations may change probabilities of nodes that are causes:

  Observing $T = 37°C$ decreases probability of Fever and Flu (and, of course, Sleepy).

- The impact of intervening actions can only follow the direction of the (causal) edges:

  Taking Aspirine (A) decreases the probability of Fever and Sleepy and may result in an alike observation for T. However, it cannot change the state for Flu since Aspirine only eases the pain and does not kill viruses.

Example: Utilities

Mildew Fungus Infestation (dt. Mehltau-Befall)

Before the harvest, a farmer checks the state of his crop and decides whether to apply a fungi treatment or not.

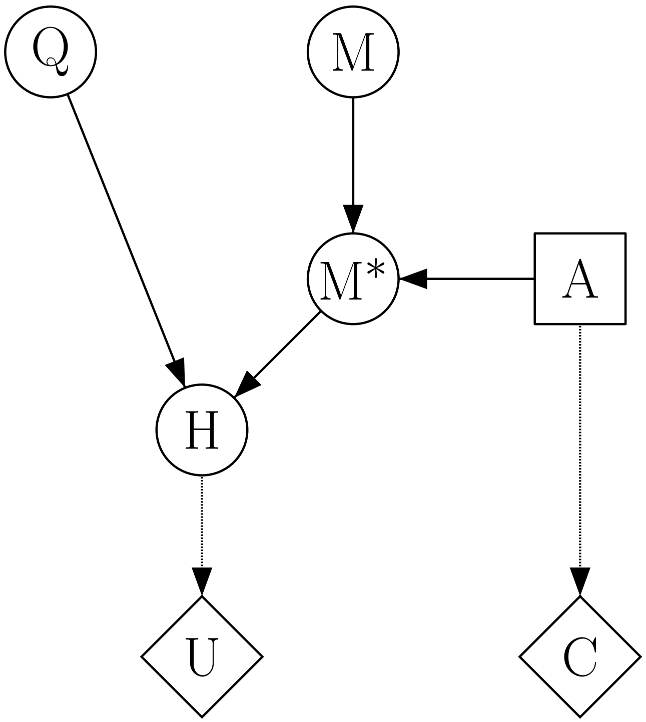
- Q — Quality of the crop
- M — Mildew infestation severity
- H — Harvest quality
- A — Action to be taken
- M* — Mildew infestation after action A
- U — Utility function of the harvest (i. e. the benefit)
- C — Utility functon of the action (i. e. the treatment costs)

$\longrightarrow$ edges leading to chance nodes

$\dashrightarrow$ edges leading to decision nodes

$\cdots\rightarrow$ edges leading to utility nodes

Example: Utilities (2)

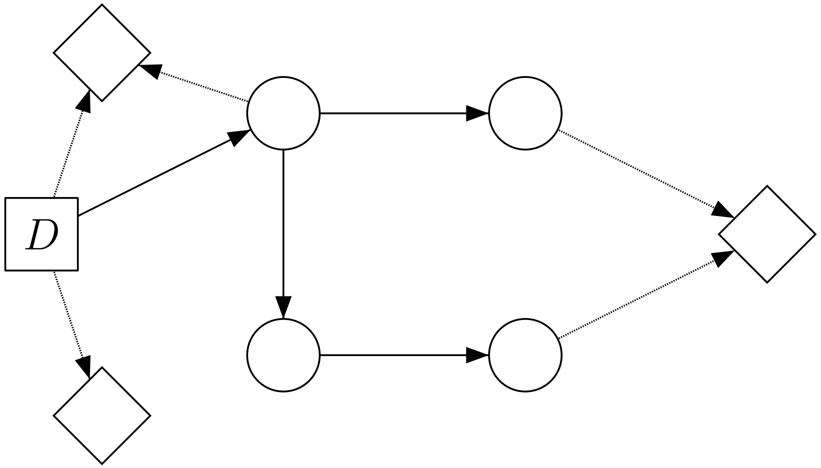


- Diamond-shaped nodes: utility functions (costs/benefits)
- Given the quality of the crops and the mildew state, which action maximizes the benefit?
- $C(\mathrm{A}) < 0$
- $U(\mathrm{H}) \geq 0$
- Expected total utility of action $\mathrm{A} = a$:

$$\mathrm{E}(U(a \mid q, m)) = C(a) + \sum_h U(h) \cdot P(h \mid a, q, m)$$

A single-action model consists of

- a Bayesian network representing the chance nodes
- one decision (action) node
- a set of utility nodes
- decision nodes can affect chance and utility nodes
- utility nodes can be affected by chance and decision nodes



Single-Action Models (2)

Given n utility nodes $U_1, \ldots, U_n$ and assuming they all depend on only one respective chance node X_i, the total expected utility given a decision $D = d$ and (chance node) evidence e is defined as:
vskip-2mm

$$\mathrm{E}(U(d \mid e)) = \sum_{i=1}^{n} \sum_{x \in \mathrm{dom}(X_i)} U_1(x_1) \cdot P(x_1 \mid d, e)$$

The optimal decision d^* is then chosen:

$$d^* = \underset{d \in \mathrm{dom}(D)}{\arg\max}\ \mathrm{E}(U(d \mid e))$$

Influence Diagrams

An influence diagram consists of a directed acyclic graph over chance nodes, decision nodes and utility nodes that obey the following structural properties:

- there is a directed path comprising all decision nodes
- utility nodes cannot have children
- decision and chance nodes are discrete
- utility nodes do not have states
- chance nodes are assigned potential tables given their parents (including decision nodes)
- each utility node U gets assigned a real-valued utility function over its parents

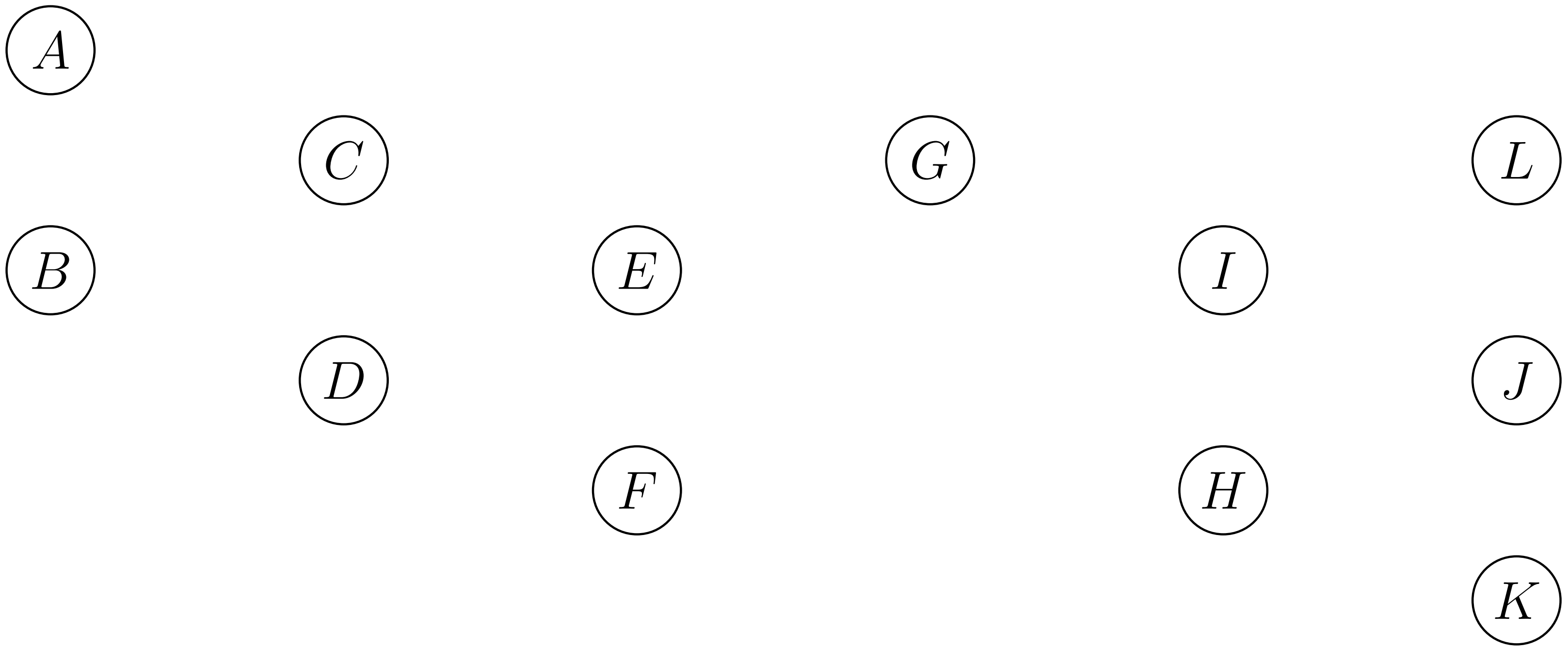
$$U : \bigtimes_{X \in \mathrm{parents}(U)} \mathrm{dom}(X) \rightarrow \mathbb{R}$$

Influence Diagrams (2)

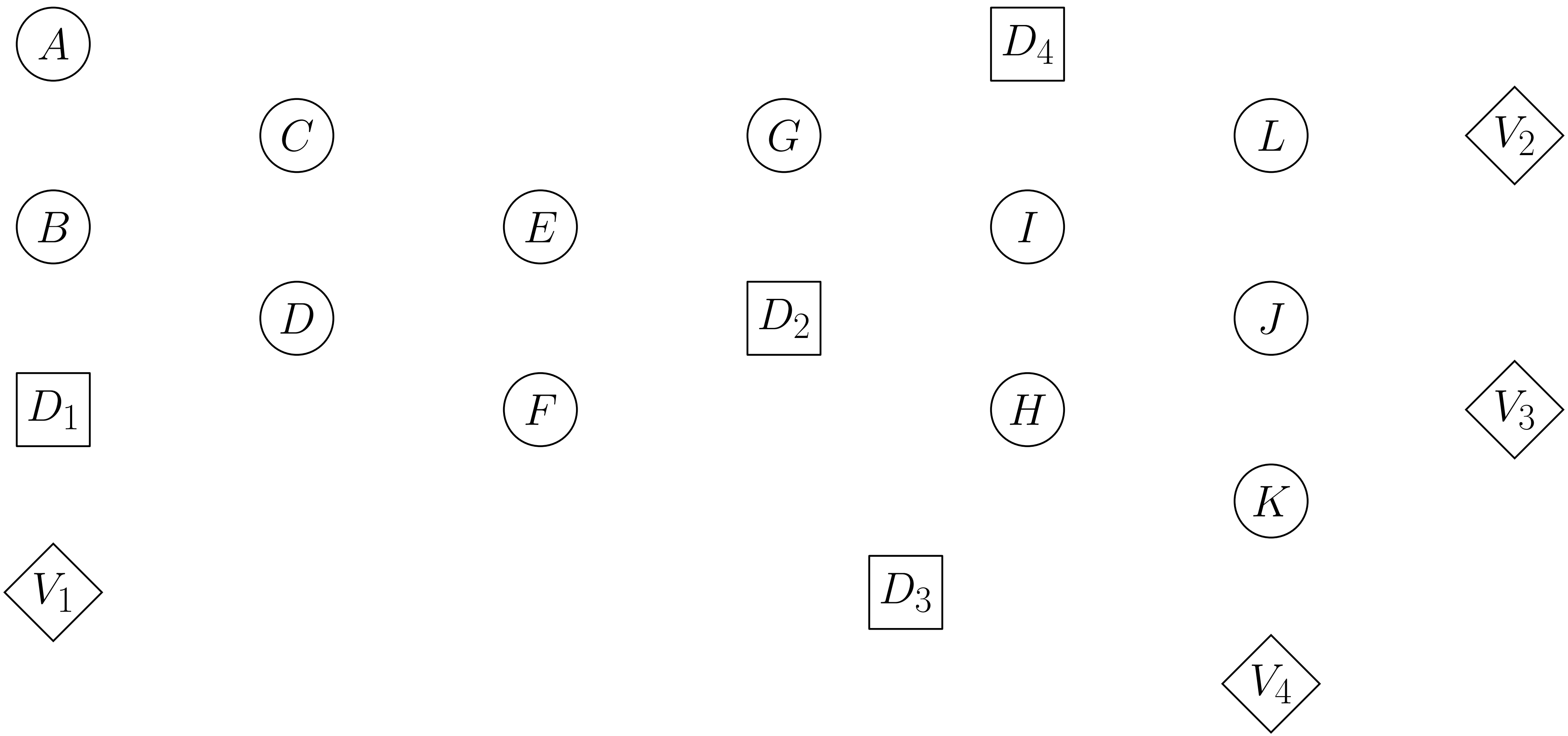
- Links into decision nodes carry no quantitative information, they only introduce a temporal ordering.
- The required path between the decision nodes induces a temporal partition of the chance nodes:

  If there are n decision nodes, then for $1 \leq i < n$ the set I_i represents all chance nodes that have to be observed after decision D_i but before decision D_{i+1}.
- I_0 is the set of chance nodes to be observed before any decision.
- I_n is the set of chance nodes that are not observed.

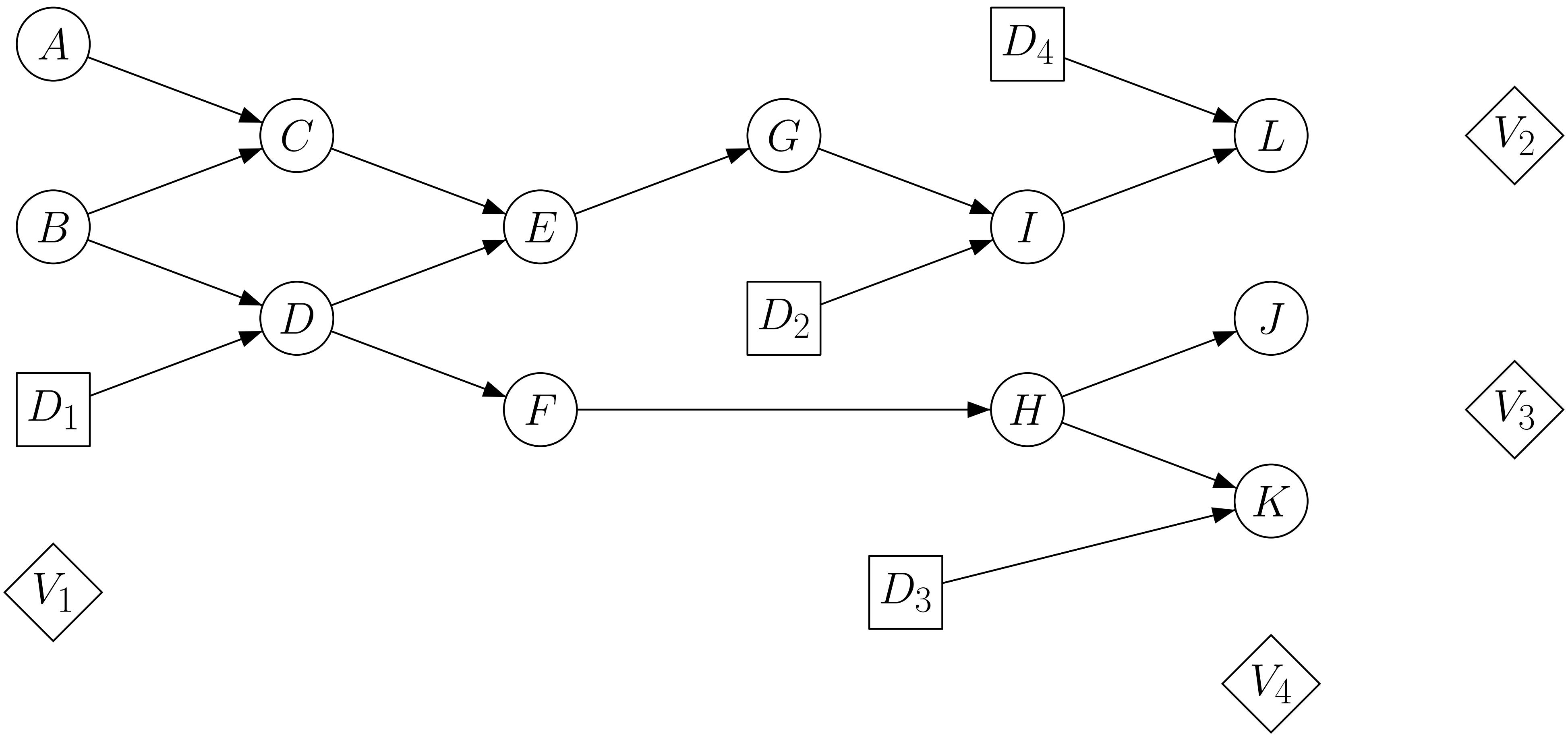
Influence Diagrams (3)



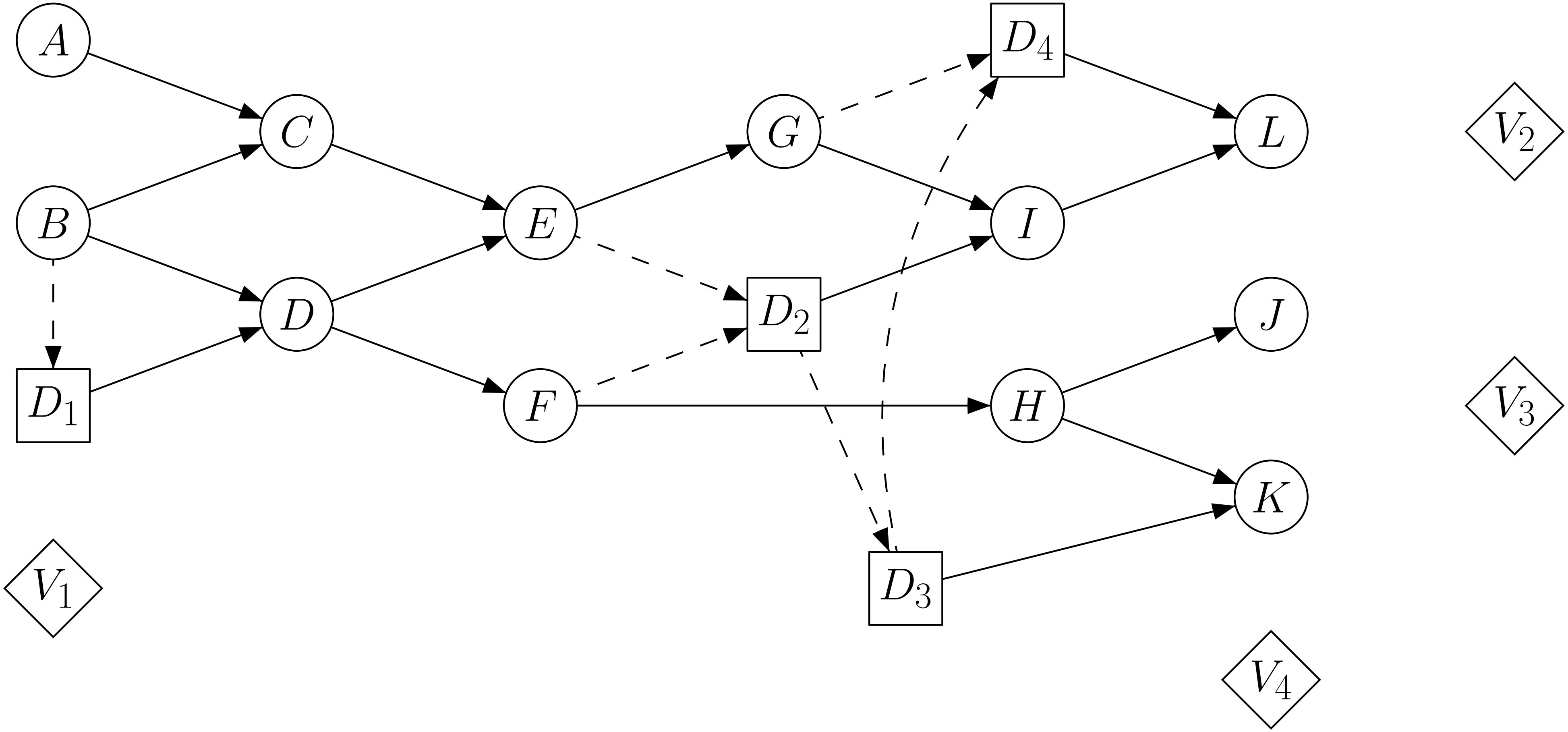
Influence Diagrams (3)



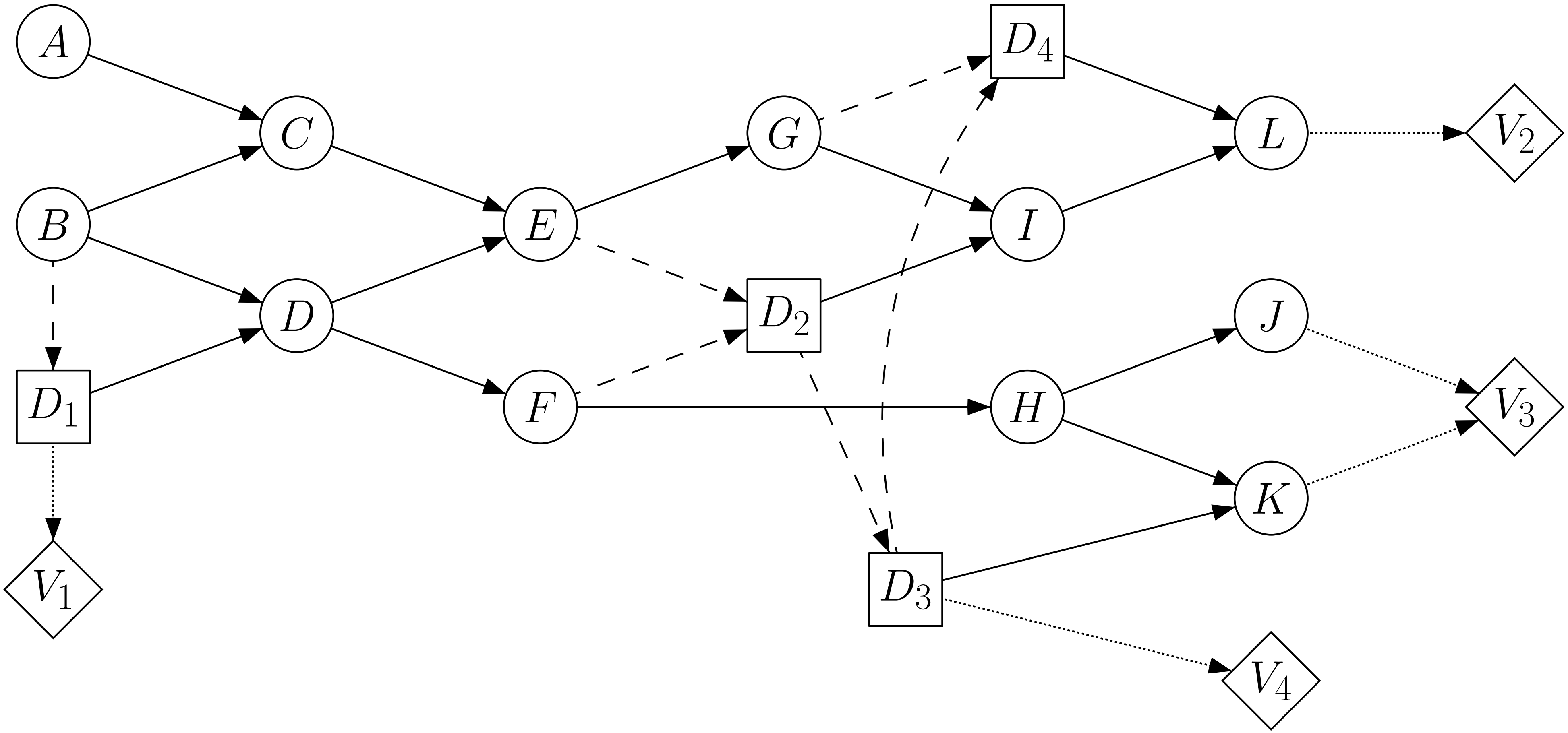
Influence Diagrams (3)

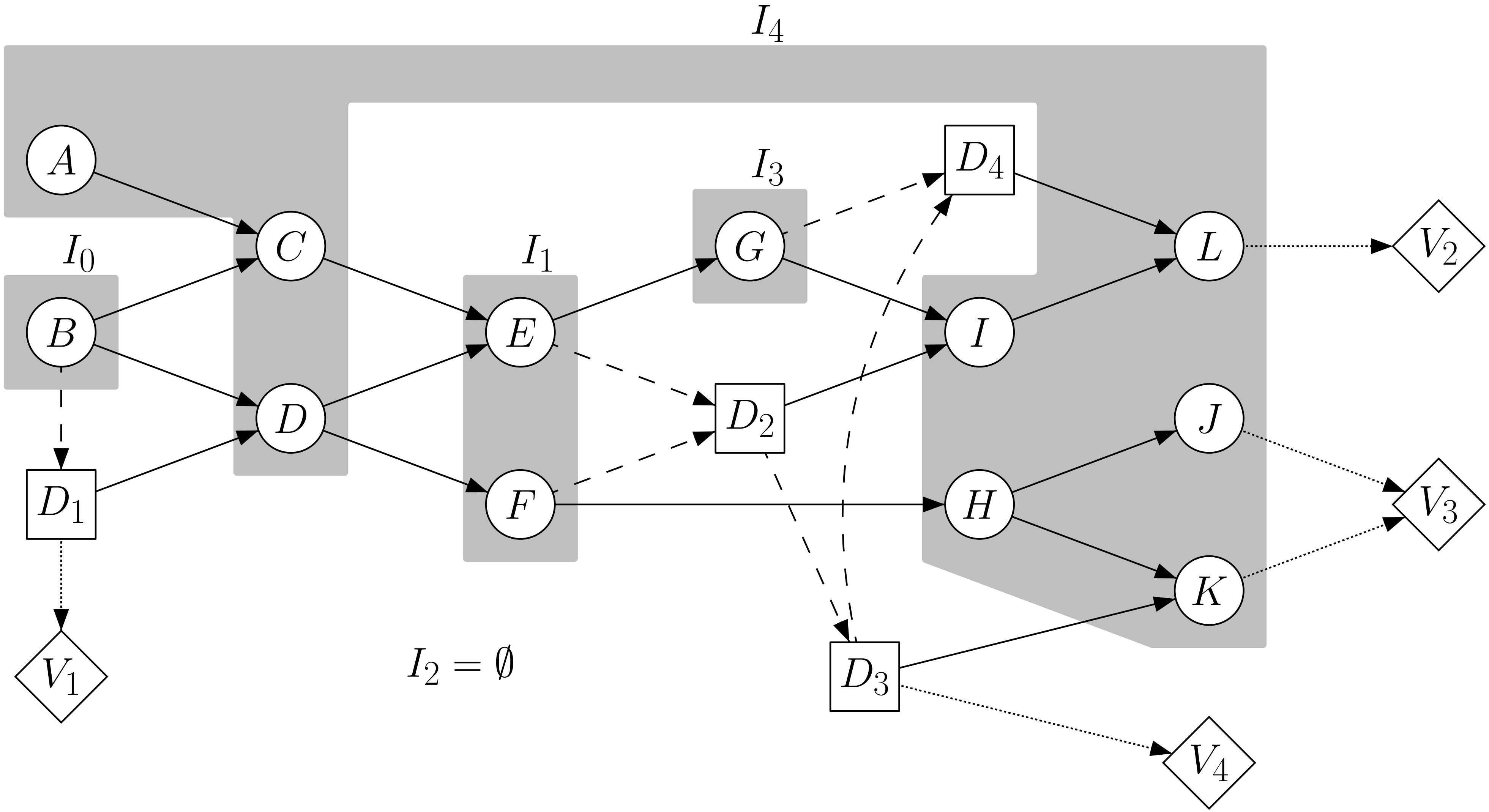


Influence Diagrams (3)



Influence Diagrams (3)

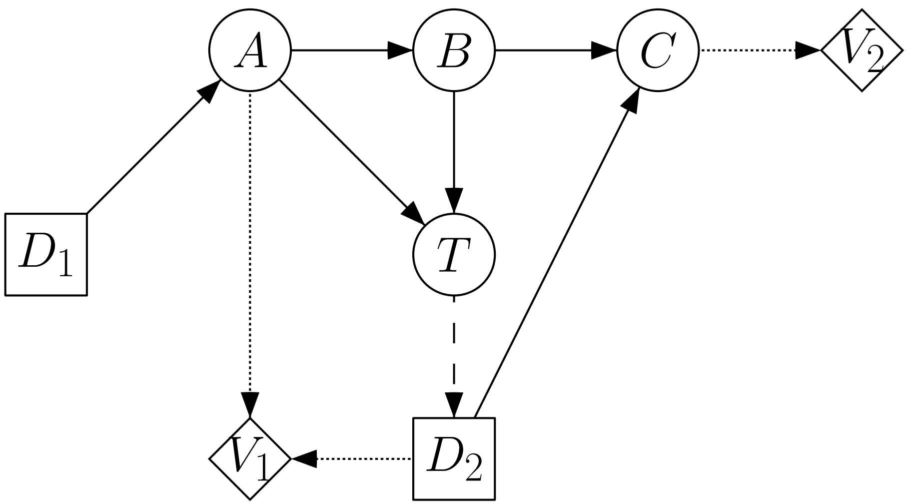



I_4
A
I_3
D_4
I_0
C
G
I_1
L
V_2
B
E
I
D
D_2
J
D_1
F
H
V_3
K
V_1
$I_2 = \emptyset$
D_3
V_4

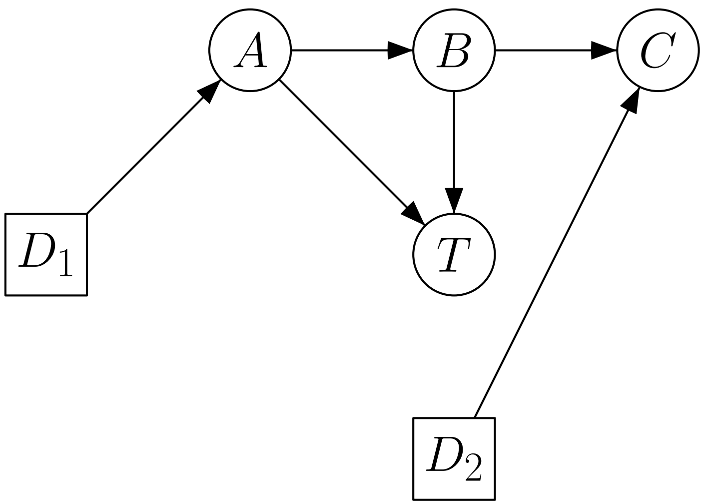
d-Separation in Influence Diagrams

To be able to use the d-separation, we need to preprocess the graphical structure of an influence diagram as follows:

- remove all utility nodes (and the edges towards them)
- remove edges that point to decision nodes



$\Rightarrow$



For example: $C \perp\!\!\!\perp T \mid B$ or $\{A, T\} \perp\!\!\!\perp D_2 \mid \emptyset$.

Chain Rule

The semantics of an influence diagram disallow some probabilities:

- $P(D)$ for a decision node D has no meaning
- $P(A \mid D)$ has no meaning unless a decision $d \in \mathrm{dom}(D)$ has been chosen

Given an influence diagram G with U_C being the set of chance nodes and U_D being the set of decision nodes, we can factorize P as follows:

$$P(U_C \mid U_D) = \prod_{X \in U_C} P(X \mid \mathrm{parents}(X))$$

- Given: an influence diagram
- Desired: a strategy which decision(s) to make

Policy

A *policy* for decision D_i is a mapping σ_i, which for any configuration of the past of D_i yields a decision for D_i, i. e.

$$\sigma_i(I_0, D_1, I_1, \ldots, D_{i-1}, I_{i-1}) \in \mathrm{dom}(D_i)$$

Strategy

A *strategy* for an influence diagram is a set of policies, one for each decision node.

Solution

A *solution* to an influence diagram is a strategy maximizing the expected utility.

Solutions to Influence Diagrams (2)

Assume, we are given an influence diagram G over $U = U_C \cup U_D$ and U_V.

- U_C ... set of chance nodes
- U_D ... set of decision nodes and
- $U_V = \{V_i\}$ ... set of utility nodes

Further, we know the following temporal order:

$$I_0 \prec D_1 \prec I_1 \prec \cdots \prec D_n \prec I_n$$

The total utility V be defined as the sum of all utility nodes: $V = \sum_i V_i$

Solutions to Influence Diagrams (3)

- An optimal policy for D_i is

$$\sigma_i(I_0, D_1, \ldots, I_{i-1}) = \arg\max_{d_i} \sum_{I_i} \max_{d_{i+1}} \cdots \max_{d_n} \sum_{I_n} P(U_C \mid U_D) \cdot V$$

  where $d_x \in \mathrm{dom}(D_x)$.

- The expected utility from following policy σ_i (and acting optimally in the future) is

$$\rho_i(I_0, D_1, \ldots, I_{i-1}) = \frac{\max_{d_i} \sum_{I_i} \max_{d_{i+1}} \cdots \max_{d_n} \sum_{I_n} P(U_C \mid U_D) \cdot V}{P(I_0, \ldots, I_{i-1} \mid D_1, \ldots, D_{i-1})}$$

  where $d_x \in \mathrm{dom}(D_x)$.

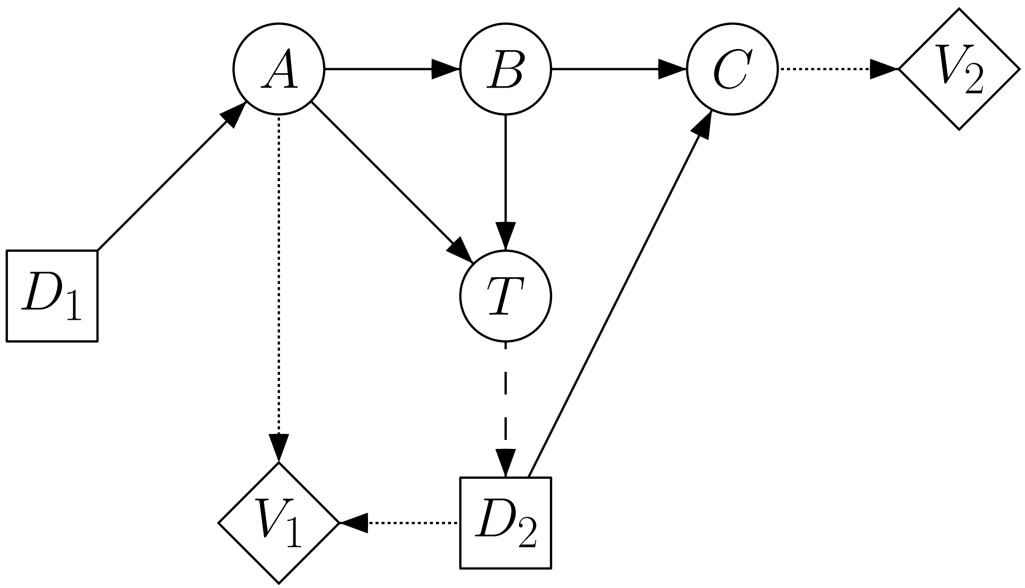
Solutions to Influence Diagrams (4)

- An optimal strategy yields the maximum expected utility of

$$\mathrm{MEU}(G) = \sum_{I_0} \max_{d_1} \sum_{I_1} \max_{d_2} \cdots \max_{d_n} \sum_{I_n} P(U_C \mid U_D) \cdot V$$

- $\sum_{I_i}$ means (sum-)marginalizing over all nodes in I_i
- $\max_{d_i}$ means taking the maximum over all $d_i \in \mathrm{dom}(D_i)$ and thus (max-)marginalizing over D_i
- Everytime I_i is marginalized out, the result is used to determine a policy for D_i.
- Marginalization in reverse temporal order
- $\Rightarrow$ use simplification techniques from the Bayesian network realm to simplify the joint probability distribution $P(U_C \mid U_D)$

Example



$P(A \mid D_1)$	$d_1^{(1)}$	$d_1^{(2)}$
y	0.2	0.8
n	0.8	0.2

$P(B \mid A)$	y	n
y	0.8	0.2
n	0.2	0.8

$P(T \mid A, B)$	y, y	y, n	n, y	n, n
y	0.9	0.5	0.5	0.1
n	0.1	0.5	0.5	0.9

$V_2(C)$	
y	10
n	0

$V_1(A, D_2)$	$d_2^{(1)}$	$d_2^{(2)}$
y	3	0
n	0	2

Utility functions

$P(C \mid B, D_2)$	y, $d_2^{(1)}$	y, $d_2^{(2)}$	n, $d_2^{(1)}$	n, $d_2^{(2)}$
y	0.9	0.5	0.5	0.9
n	0.1	0.5	0.5	0.1

Chance potentials

Example (2)

For D_2 we can read from the graph:

$$I_0 = \emptyset \qquad I_1 = \{T\} \qquad I_2 = \{A, B, C\}$$

Thus, σ_2 can be solved to the following strategy:

$\sigma_2(\emptyset, D_1, \{T\})$	$d_1^{(1)}$	$d_1^{(2)}$
y	$d_2^{(1)}$	$d_2^{(1)}$
n	$d_2^{(2)}$	$d_2^{(2)}$

$\rho_2(\emptyset, D_1, \{T\})$	$d_1^{(1)}$	$d_1^{(2)}$
y	9.51	11.29
n	10.34	8.97

Finally, $\sigma_1 = d_1^{(2)}$ and $\mathrm{MEU}(G) = 10.58$.

Frameworks of Imprecision and Uncertainty

Problems with Probability Theory

Representation of Ignorance (dt. Unwissen)

- We are given a die with faces $1, \ldots, 6$
  What is the certainty of showing up face i ?
  - Conduct a statistical survey (roll the die 10000 times) and estimate the relative frequency: $P(\{i\}) = \frac{1}{6}$
  - Use subjective probabilities (which is often the normal case): We do not know anything (especially and explicitly we do not have any reason to assign unequal probabilities), so the most plausible distribution is a uniform one.

  $\Rightarrow$ Problem: Uniform distribution because of ignorance or extensive statistical tests

- Experts analyze aircraft shapes: 3 aircraft types A, B, C
  "It is type A or B with 90% certainty. About C, I don't have any clue and I do not want to commit myself. No preferences for A or B."

$\Rightarrow$ Problem: Propositions hard to handle with Bayesian theory

Modeling Imprecise Data

"$A \subseteq X$ being an imprecise date" means: the true value x_0 lies in A but there are no preferences on A.

Ω	set of possible elementary events
$\Theta = \{\xi\}$	set of observers
$\lambda(\xi)$	importance of observer ξ

Some elementary event from Ω occurs and every observer $\xi \in O$ shall announce which elementary events she personally considers possible. This set is denoted by $\Gamma(\xi) \subseteq \Omega$. $\Gamma(\xi)$ is then an imprecise date.

$\lambda : 2^{\Theta} \to [0, 1]$	probability measure
	(interpreted as importance measure)
$(\Theta, 2^{\Theta}, \lambda)$	probability space
$\Gamma : \Theta \to 2^{\Omega}$	set-valued mapping

Imprecise Data (2)

Let $A \subseteq \Omega$:

a) $\Gamma^*(A) \stackrel{\text{Def}}{=} \{\xi \in \Theta \mid \Gamma(\xi) \cap A \neq \emptyset\}$

b) $\Gamma_*(A) \stackrel{\text{Def}}{=} \{\xi \in \Theta \mid \Gamma(\xi) \neq \emptyset \text{ and } \Gamma(\xi) \subseteq A\}$

Remarks:

a) If $\xi \in \Gamma^*(A)$, then it is *plausible* for ξ that the occurred elementary event lies in A.

b) If $\xi \in \Gamma_*(A)$, then it is *certain* for ξ that the event lies in A.

c) $\{\xi \mid \Gamma(\xi) \neq \emptyset\} = \Gamma^*(\Omega) = \Gamma_*(\Omega)$

Let $\lambda(\Gamma^*(\Omega)) > 0$. Then we call

$$P^*(A) = \frac{\lambda(\Gamma^*(A))}{\lambda(\Gamma^*(\Omega))} \quad \text{the upper, and} \quad P_*(A) = \frac{\lambda(\Gamma_*(A))}{\lambda(\Gamma_*(\Omega))} \quad \text{the lower}$$

probability w. r. t. λ and Γ.

Example

$$\Theta = \{a, b, c, d\} \qquad \lambda\colon\ a \mapsto {}^1\!/_6 \qquad \Gamma\colon\ a \mapsto \{1\}$$
$$\Omega = \{1, 2, 3\} \qquad b \mapsto {}^1\!/_6 \qquad b \mapsto \{2\}$$
$$\Gamma^*(\Omega) = \{a, b, d\} \qquad c \mapsto {}^2\!/_6 \qquad c \mapsto \emptyset$$
$$\lambda(\Gamma^*(\Omega)) = {}^4\!/_6 \qquad d \mapsto {}^2\!/_6 \qquad d \mapsto \{2, 3\}$$

A	$\Gamma^*(A)$	$\Gamma_*(A)$	$P^*(A)$	$P_*(A)$
$\emptyset$	$\emptyset$	$\emptyset$	0	0
$\{1\}$	$\{a\}$	$\{a\}$	$\frac{1}{4}$	$\frac{1}{4}$
$\{2\}$	$\{b, d\}$	$\{b\}$	$\frac{3}{4}$	$\frac{1}{4}$
$\{3\}$	$\{d\}$	$\emptyset$	$\frac{1}{2}$	0
$\{1, 2\}$	$\{a, b, d\}$	$\{a, b\}$	1	$\frac{1}{2}$
$\{1, 3\}$	$\{a, d\}$	$\{a\}$	$\frac{3}{4}$	$\frac{1}{4}$
$\{2, 3\}$	$\{b, d\}$	$\{b, d\}$	$\frac{3}{4}$	$\frac{3}{4}$
$\{1, 2, 3\}$	$\{a, b, d\}$	$\{a, b, d\}$	1	1

One can consider $P^*(A)$ and $P_*(A)$ as upper and lower probability bounds.

Imprecise Data (3)

Some properties of probability bounds:

a) $P^*\colon\ 2^\Omega \to [0, 1]$

b) $0 \leq P_* \leq P^* \leq 1, \qquad P_*(\emptyset) = P^*(\emptyset) = 0, \qquad P_*(\Omega) = P^*(\Omega) = 1$

c) $A \subseteq B \quad \Rightarrow \quad P^*(A) \leq P^*(B) \quad \text{and} \quad P_*(A) \leq P_*(B)$

d) $A \cap B = \emptyset \quad \not\Rightarrow \quad P^*(A) + P^*(B) = P^*(A \cup B)$

e) $P_*(A \cup B) \geq P_*(A) + P_*(B) - P_*(A \cap B)$

f) $P^*(A \cup B) \leq P^*(A) + P^*(B) - P^*(A \cap B)$

g) $P_*(A) = 1 - P^*(\Omega \backslash A)$

One can prove the following generalized equation:

$$P_*(\bigcup_{i=1}^{n} A_i) \quad \geq \sum_{\emptyset \neq I : I \subseteq \{1,\ldots,n\}} (-1)^{|I|+1} \cdot P_*(\bigcap_{i \in I} A_i)$$

These set functions also play an important role in theoretical physics (capacities, Choquet, 1955). Shafer did generalize these thoughts and developed a theory of belief functions.

Belief Revision

How is new knowledge incoporated?

Every observer announces the location of the ship in form of a subset of all possible ship locations. Given these set-valued mappings, we can derive upper and lower probabilities with the help of the observer importance measure. Let us assume the ship is certainly at sea.

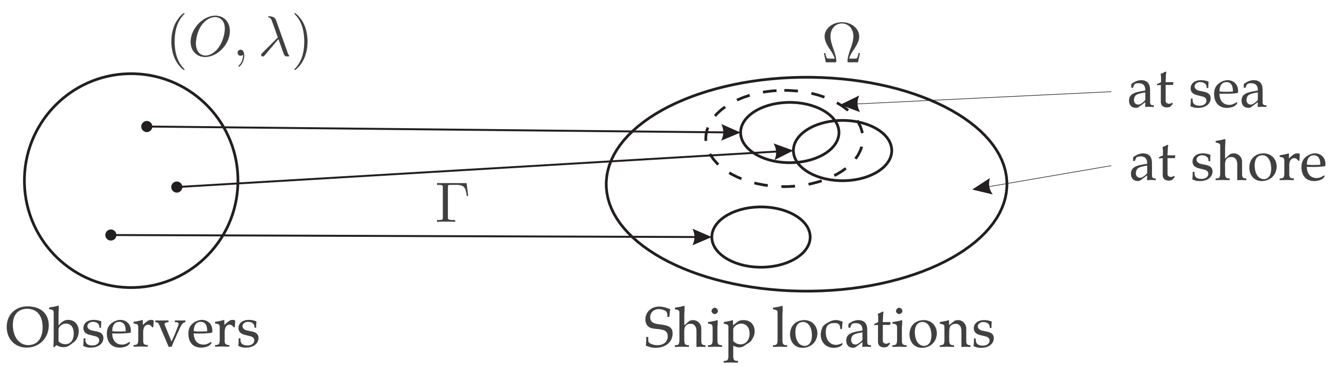
How do the upper/lower probabilities change?

Example

a) *Geometric Conditioning*
(observers that give partial or full wrong information are discarded)

$$P_*(A \mid B) = \frac{\lambda(\{\xi \in \Theta \mid \Gamma(\xi) \subseteq A \text{ and } \Gamma(\xi) \subseteq B\})}{\lambda(\{\xi \in \Theta \mid \Gamma(\xi) \subseteq B\})} = \frac{P_*(A \cap B)}{P_*(B)}$$

$$P^*(A \mid B) = \frac{\lambda(\{\xi \in \Theta \mid \Gamma(\xi) \subseteq B \text{ and } \Gamma(\xi) \cap A \neq \emptyset\})}{\lambda(\{\xi \in \Theta \mid \Gamma(\xi) \subseteq B\})} = \frac{P^*(A \cup \overline{B}) - P^*(\overline{B})}{1 - P^*(\overline{B})}$$

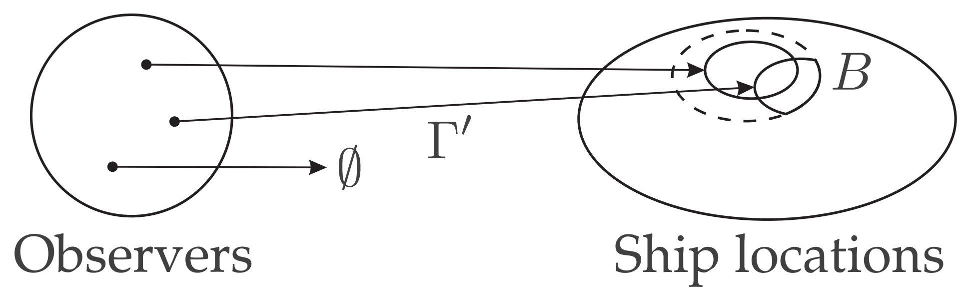


Belief Revision (2)

b) *Data Revision*
(the observed data is modified such that they fit the certain information)

$$(P_*)_B(A) = \frac{P_*(A \cup \overline{B}) - P_*(\overline{B})}{1 - P_*(B)}$$

$$(P^*)_B(A) = \frac{P^*(A \cap B)}{P^*(B)}$$



These two concepts have different semantics. There are several more belief revision concepts.

Combination of Random Sets

Let $(\Omega, 2^\Omega)$ be a space of events. Further be $(O_1, 2^{O_1}, \lambda_1)$ and $(O_2, 2^{O_2}, \lambda_2)$ spaces of independent observers.

We call $(O_1 \times O_2, \lambda_1 \cdot \lambda_2)$ the product space of observers and

$$\Gamma : O_1 \times O_2 \to 2^\Omega, \Gamma(x_1, x_2) = \Gamma_1(x_1) \cap \Gamma_2(x_2)$$

the combined observer function.

We obtain with

$$(P_L)_*(A) = \frac{(\lambda_1 \cdot \lambda_2)(\{(x_1, x_2) \mid \Gamma(x_1, x_2) \neq \emptyset \wedge \Gamma(x_1, x_2) \sqsubseteq A\})}{(\lambda_1 \cdot \lambda_2)(\{(x_1, x_2 \mid \Gamma(x_1, x_2) \neq \emptyset)\})}$$

the lower probability of A that respects both observations.

Example

$$\Omega = \{1,2,3\} \qquad \begin{aligned} \lambda_1\colon\ \{a\} &\mapsto {}^1\!/_3 \\ \{b\} &\mapsto {}^2\!/_3 \end{aligned} \qquad \begin{aligned} \lambda_2\colon\ \{c\} &\mapsto {}^1\!/_2 \\ \lambda_2\colon\ \{d\} &\mapsto {}^1\!/_2 \end{aligned}$$

$$\begin{aligned} O_1 &= \{a,b\} \\ O_2 &= \{c,d\} \end{aligned} \qquad \Gamma_1\colon \begin{array}{ccc} a & \mapsto & \{1,2\} \\ b & \mapsto & \{2,3\} \end{array} \qquad \Gamma_2\colon \begin{array}{ccc} c & \mapsto & \{1\} \\ d & \mapsto & \{2,3\} \end{array}$$

Combination:

$$O_1 \times O_2 \;=\; \{\overline{ac}, \overline{bc}, \overline{ad}, \overline{bd}\}$$

$$\begin{aligned} \lambda\colon\ \{\overline{ac}\} &\mapsto {}^1\!/_6 \\ \{\overline{ad}\} &\mapsto {}^1\!/_6 \\ \{\overline{bc}\} &\mapsto {}^2\!/_6 \\ \{\overline{bd}\} &\mapsto {}^2\!/_6 \end{aligned} \qquad \begin{aligned} \Gamma\colon\ \overline{ac} &\mapsto \{1\} \\ \overline{ad} &\mapsto \{2\} \\ \overline{bc} &\mapsto \emptyset \\ \overline{bd} &\mapsto \{2,3\} \end{aligned} \qquad \begin{aligned} \Gamma_*(\Omega) &= \{(x_1,x_2) \mid \Gamma(x_1,x_2) \neq \emptyset\} \\ &= \{\overline{ac}, \overline{ad}, \overline{bd}\} \\ \\ \lambda(\Gamma_*(\Omega)) &= {}^4\!/_6 \end{aligned}$$

Example (2)

A	$m_1(A)$	$(P_*)_{\Gamma_1}(A)$	$m_2(A)$	$(P_*)_{\Gamma_2}(A)$	$m(A)$	$(P_*)_{\Gamma}(A)$
$\emptyset$	0	0	0	0	0	0
$\{1\}$	0	0	$1/2$	$1/2$	$1/4 = \frac{1/6}{4/6}$	$1/4$
$\{2\}$	0	0	0	0	$1/4$	$1/4$
$\{3\}$	0	0	0	0	0	0
$\{1,2\}$	$1/3$	$1/3$	0	$1/2$	0	$1/2$
$\{1,3\}$	0	0	0	$1/2$	0	$1/4$
$\{2,3\}$	$2/3$	$2/3$	$1/2$	$1/2$	$1/2$	$3/4$
$\{1,2,3\}$	0	1	0	1	0	1

Imprecise Probabilities

Let x_0 be the true value but assume there is no information about $P(A)$ to decide whether $x_0 \in A$. There are only probability boundaries.

Let $\mathcal{L}$ be a set of probability measures. Then we call

$$(P_{\mathcal{L}})_* : 2^{\Omega} \to [0,1]\,, A \mapsto \inf\{P(A) \mid P \in \mathcal{L}\} \qquad \text{the lower and}$$

$$(P_{\mathcal{L}})^* : 2^{\Omega} \to [0,1]\,, A \mapsto \sup\{P(A) \mid P \in \mathcal{L}\} \qquad \text{the upper}$$

probability of A w. r. t. $\mathcal{L}$.

a) $(P_{\mathcal{L}})_*(\emptyset) = (P_{\mathcal{L}})^*(\emptyset) = 0; \quad (P_{\mathcal{L}})_*(\Omega) = (P_{\mathcal{L}})^*(\Omega) = 1$

b) $0 \leq (P_{\mathcal{L}})_*(A) \leq (P_{\mathcal{L}})^*(A) \leq 1$

c) $(P_{\mathcal{L}})^*(A) = 1 - (P_{\mathcal{L}})_*(\overline{A})$

d) $(P_{\mathcal{L}})_*(A) + (P_{\mathcal{L}})_*(B) \leq (P_{\mathcal{L}})_*(A \cup B)$

e) $(P_{\mathcal{L}})_*(A \cap B) + (P_{\mathcal{L}})_*(A \cup B) \not\geq (P_{\mathcal{L}})_*(A) + (P_{\mathcal{L}})_*(B)$

Belief Revision

Let $B \subseteq \Omega$ and $\mathcal{L}$ a class of probabilities. The we call

$$A \subseteq \Omega : (P_{\mathcal{L}})_*(A \mid B) = \inf\{P(A \mid B) \mid P \in \mathcal{L} \wedge P(B) > 0\} \qquad \text{the lower and}$$
$$A \subseteq \Omega : (P_{\mathcal{L}})^*(A \mid B) = \sup\{P(A \mid B) \mid P \in \mathcal{L} \wedge P(B) > 0\} \qquad \text{the upper}$$

conditional probability of A given B.

A class $\mathcal{L}$ of probability measures on $\Omega = \{\omega_1, \ldots, \omega_n\}$ is of type 1, iff there exist functions R_1 and R_2 from 2^Ω into $[0, 1]$ with:

$$\mathcal{L} = \{P \mid \forall A \subseteq \Omega : R_1(A) \leq P(A) \leq R_2(A)\}$$

Belief Revision (2)

Intuition: P is determined by $P(\{\omega_i\})$, $i = 1, \ldots, n$ which corresponds to a point in $\mathbb{R}^n$ with coordinates $\big(P(\{\omega_1\}), \ldots, P(\{\omega_n\})\big)$.

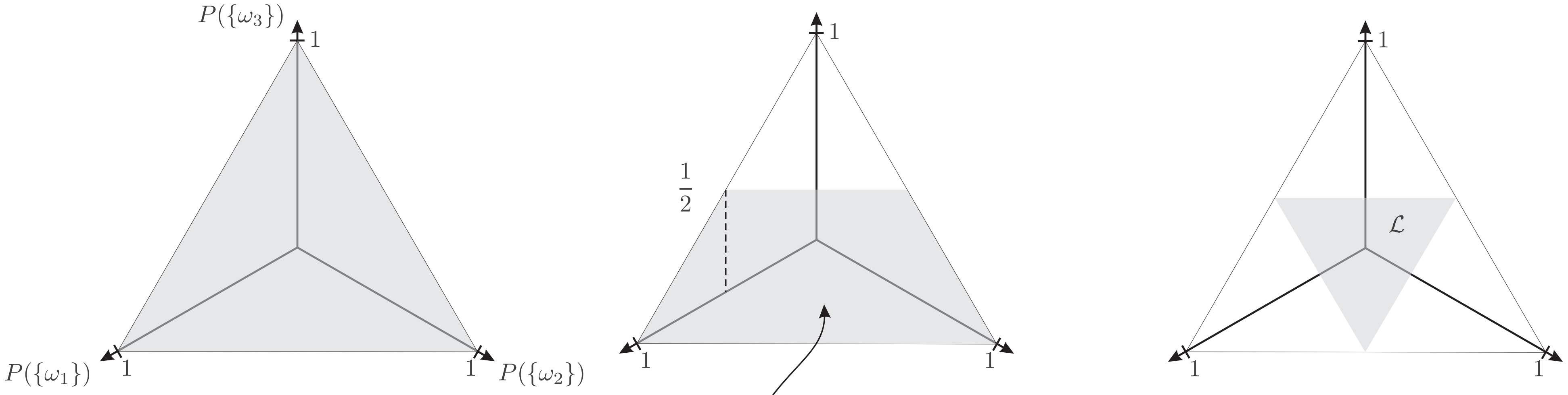
If $\mathcal{L}$ is type 1, it holds true that:

$$\mathcal{L} \Leftrightarrow \Big\{(r_1, \ldots, r_n) \in \mathbb{R}^n \mid \exists P\colon \forall A \subseteq \Omega\colon \\ (P_{\mathcal{L}})_*(A) \leq P(A) \leq (P_{\mathcal{L}})^*(A) \\ \text{and} \;\; r_i = P(\{\omega_i\}),\; i = 1, \ldots, n\Big\}$$

Example

$\Omega = \{\omega_1, \omega_2, \omega_3\}$

$\mathcal{L} = \{P \mid \frac{1}{2} \le P(\{\omega_1, \omega_2\}) \le 1, \quad \frac{1}{2} \le P(\{\omega_2, \omega_3\}) \le 1, \quad \frac{1}{2} \le P(\{\omega_1, \omega_3\}) \le 1\}$



general restriction:
$0 \le P(\{\omega_i\}) \le 1$
$P(\{\omega_1\}) + P(\{\omega_2\}) + P(\{\omega_3\}) = 1$

$\{P \mid \frac{1}{2} \le P(\{\omega_1, \omega_2\}) \le 1\}$

Let $A_1 = \{\omega_1, \omega_2\}$, $A_2 = \{\omega_2, \omega_3\}$, $A_3 = \{\omega_1, \omega_3\}$

$$P_*(A_1) + P_*(A_2) + P_*(A_3) - P_*(A_1 \cap A_2) - P_*(A_2 \cap A_3) - P_*(A_1 \cap A_3) + P_*(A_1 \cap A_2 \cap A_3)$$
$$= \frac{1}{2} + \frac{1}{2} + \frac{1}{2} - 0 - 0 - 0 + 0 = \frac{3}{2} > 1 = P(A_1 \cup A_2 \cup A_3)$$

Belief Revision (3)

If $\mathcal{L}$ is type 1 and $(P_\mathcal{L})^*(A \cup B) \geq (P_\mathcal{L})^*(A) + (P_\mathcal{L})^*(B) - (P_\mathcal{L})^*(A \cap B)$, then

$$(P_\mathcal{L})^*(A \mid B) = \frac{(P_\mathcal{L})^*(A \cap B)}{(P_\mathcal{L})^*(A \cap B) + (P_\mathcal{L})_*(B \cap \overline{A})}$$

and

$$(P_\mathcal{L})_*(A \mid B) = \frac{(P_\mathcal{L})_*(A \cap B)}{(P_\mathcal{L})_*(A \cap B) + (P_\mathcal{L})^*(B \cap \overline{A})}$$

Let $\mathcal{L}$ be a class of type 1. $\mathcal{L}$ is of type 2, iff

$$(P_\mathcal{L})_*(A_1 \cup \cdots \cup A_n) \geq \sum_{I:\emptyset \neq I \subseteq \{1,\ldots,n\}} (-1)^{|I|+1} \cdot (P_\mathcal{L})_*(\bigcap_{i \in I} A_i)$$

Fuzzy Sets

Classical description of concepts/properties:

Example: concept "two-digit number"

a) as a set: $\{10, 11, \ldots, 99\} = M$

b) as predicate $\text{two-digit}(x) = \begin{cases} \textsf{true} & \text{if} \quad 10 \leq x \leq 99 \\ \textsf{false} & \text{else} \end{cases}$

Connection between a) and b):

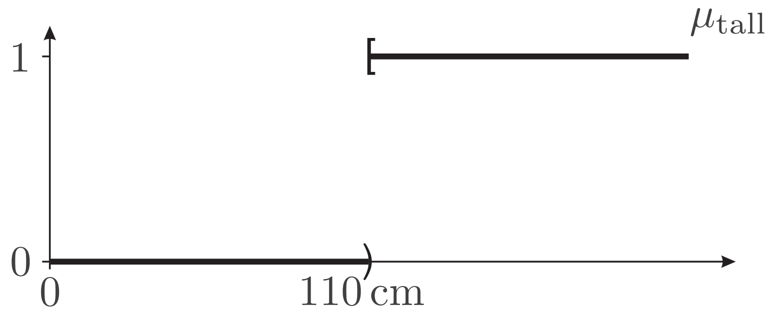
$$M = \{x \in \mathbb{N} \mid \text{two-digit}(x)\}; \qquad \text{two-digit}(x) \Leftrightarrow x \in M$$

Both concepts are not suited for defining concepts like:

- "large"
- "old"
- "heavy"

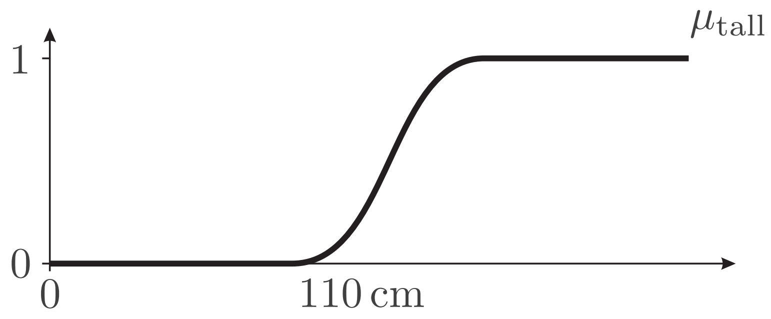
Example

“Set” of sizes (in cm) at which a child would be regarded “tall”.



characteristic function of the concept “tall”
($\widehat{=} \{x \mid x \geq 110\,\text{cm}\}$)

The saltus at 110 cm from 0 to 1 is not intuitive. Therefore:



membership degree function

Fuzzy Sets

A *fuzzy set* over a basic set X is a mapping

$$\mu_X : X \to [0, 1]$$

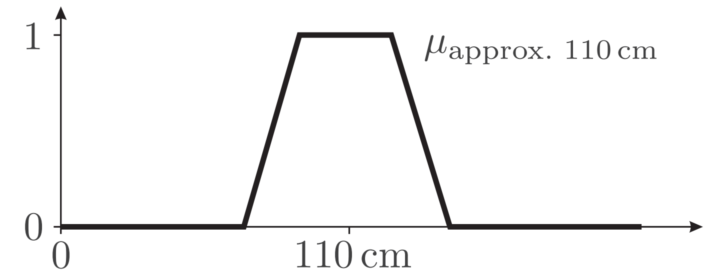
$\mu_X(x) \in [0, 1]$ is the degree of membership of x to the fuzzy set μ_X.

Combination of concepts like "tall", "approx. 110 cm", ...

a) The child is "tall" **and** "approx. 110 cm (tall)"

b) The child is "tall" **or** "approx. 110 cm (tall)"

c) The child is **not** "tall"



a)	$\widehat{=}$	Intersection:	classical:	$x \in A \cap B$	$\Leftrightarrow$	$x \in A \wedge x \in B$
b)	$\widehat{=}$	Union:	classical:	$x \in A \cup B$	$\Leftrightarrow$	$x \in A \vee x \in B$
c)	$\widehat{=}$	Complement:	classical:	$x \in \overline{A}$	$\Leftrightarrow$	$\neg(x \in A)$

Postulate:

$$\mu_{\text{tall}\wedge\text{approx. 110 cm}}(x) = \mu_{\text{tall}}(x) \top \mu_{\text{approx. 110 cm}}(x)$$

I. e., we need a mapping $\top : [0,1]^2 \to [0,1]$

Generalized Conjunction, t-Norm

A *t-norm* is a mapping $\top : [0,1]^2 \to [0,1]$ with

(T1) $\top(a,1) = a$

(T2) $a \leq a' \Rightarrow \top(a,b) \leq \top(a',b)$

(T3) $\top(a,b) = \top(b,a)$

(T4) $\top(\top(a,b),c) = \top(a,\top(b,c))$

Examples:

$$\min\{a,b\},\ a \cdot b,\ \max\{a+b-1,0\}$$

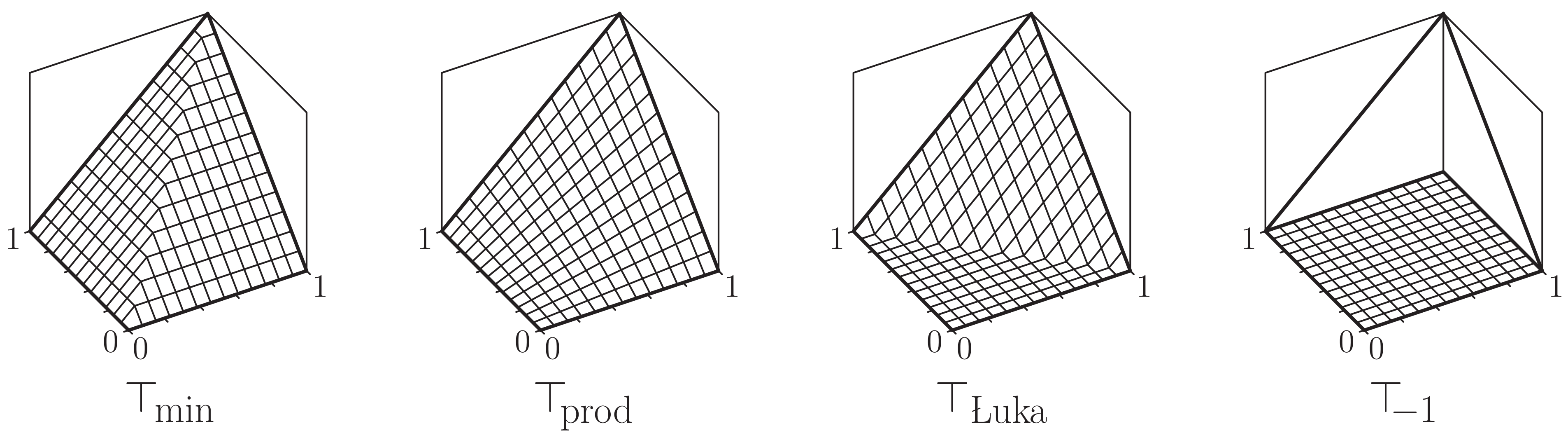
largest t-norm, the only idempotent t-norm (i. e., $\top(a,a) = a$)

$$0 \leq \top(0,0) \overset{(T2)}{\leq} \top(1,0) \overset{(T3)}{=} \top(0,1) \overset{(T1)}{=} 0; \quad \top(1,1) \overset{(T1)}{=} 1$$

Reasonable claim: $\mu_{\text{tall}}(x) \top \mu_{\text{tall}}(x) = \mu_{\text{tall}}(x) \ \Rightarrow\ \top$ idempotent

t-Norms / Fuzzy Conjunctions

standard conjunction:	$\top_{\text{min}}(a,b)$	$= \min\{a,b\}$
algebraic product:	$\top_{\text{prod}}(a,b)$	$= a \cdot b$
Łukasiewicz:	$\top_{\text{Łuka}}(a,b)$	$= \max\{0, a+b-1\}$
drastic product:	$\top_{-1}(a,b)$	$= \begin{cases} a, & \text{if } \ b=1, \\ b, & \text{if } \ a=1, \\ 0, & \text{otherwise.} \end{cases}$



Example

$X = \{c_1, c_2, c_3\}$	Set of computers
μ_{cheap}	Fuzzy set of cheap computers
μ_{fast}	Fuzzy set of fast computers
$\mu_{\text{goodvalue}}$	$\mu_{\text{cheap}} \top \mu_{\text{fast}}$

Computer	Price	Speed	μ_{cheap}	μ_{fast}	$\mu_{\text{goodvalue}}$ $(\top = \top_{\min})$	$(\top = \top_{\text{prod}})$
c_1	2000	20	1.0	0.4	0.4	0.40
c_2	2500	40	0.6	0.8	0.6	0.48
c_3	2500	50	0.6	0.9	0.6	0.54

Generalized Disjunction, t-Conorm

A *t-conorm* is a mapping $\perp : [0,1]^2 \to [0,1]$ with

(S1) $\perp(a, 0) = a$

(S2) $a \leq a' \Rightarrow \perp(a, b) \leq \perp(a', b)$

(S3) $\perp(a, b) = \perp(b, a)$

(S4) $\perp(\perp(a, b), c) = \perp(a, \perp(b, c))$

Examples:

$$\max\{a, b\},\ a + b - a \cdot b,\ \min\{a + b, 1\}$$

smallest t-conorm, the only idempotent t-conorm (i. e., $\perp(a, a) = a$)

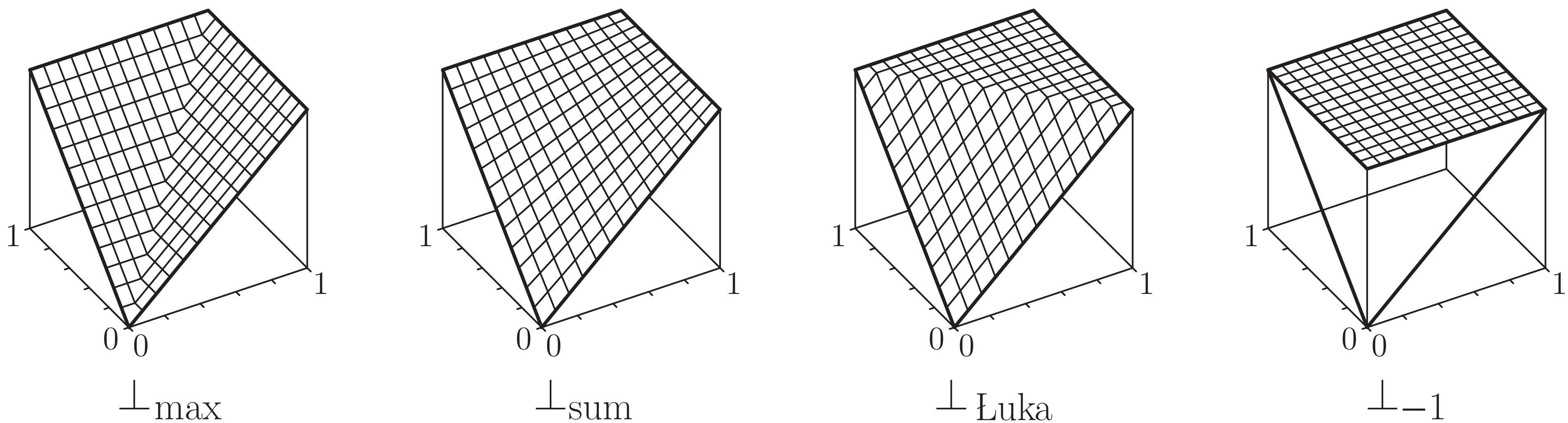
t-Conorms / Fuzzy Disjunctions

standard disjunction: $\bot_{\max}(a,b) = \max\{a,b\}$

algebraic sum: $\bot_{\text{sum}}(a,b) = a + b - a \cdot b$

Łukasiewicz: $\bot_{\text{Łuka}}(a,b) = \min\{1, a+b\}$

drastic sum:
$$\bot_{-1}(a,b) = \begin{cases} a, & \text{if } b = 0, \\ b, & \text{if } a = 0, \\ 1, & \text{otherwise.} \end{cases}$$



Generalized Negation

A *negation operator* is a mapping $\sim\colon [0,1] \to [0,1]$ with

(N1) $\sim 0 = 1$

(N2) $a \leq b \quad \Rightarrow \quad \sim b \leq \sim a$

(N3) $\sim(\sim a) = a$

From (N1) and (N3) follows: $\quad \sim 1 = 0$

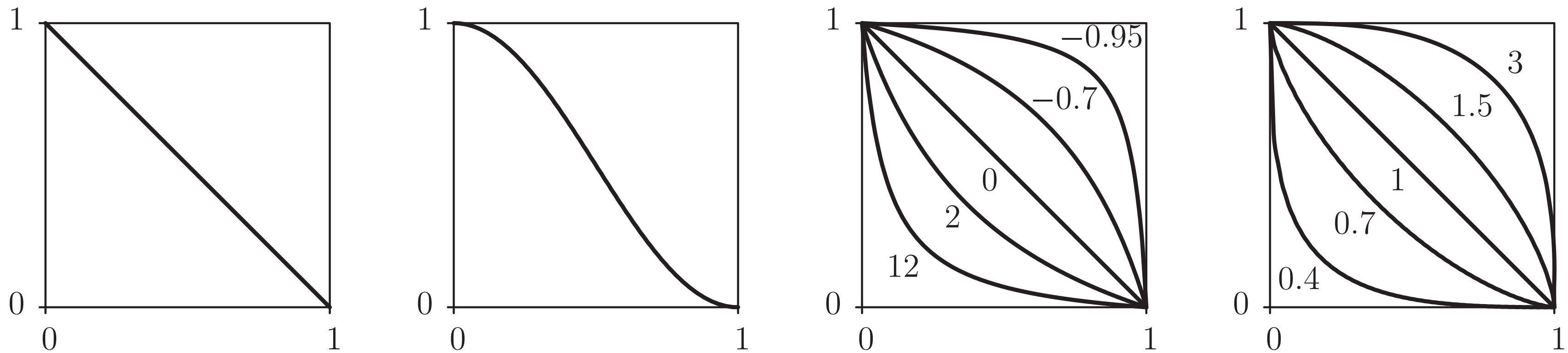
Relation between t-norms and t-conorms:

$\top$ t-norm $\Leftrightarrow$ $\bot_\sim$ t-conorm: $\bot_\sim(a,b) = \ \sim(\top(\sim a, \sim b))$ $\left(a \vee b \mathrel{\widehat{=}} \neg(\neg a \wedge \neg b)\right)$

$\bot$ t-conorm $\Leftrightarrow$ $\top_\sim$ t-norm: $\top_\sim(a,b) = \ \sim(\bot(\sim a, \sim b))$ $\left(a \wedge b \mathrel{\widehat{=}} \neg(\neg a \vee \neg b)\right)$

Fuzzy Negations

standard negation:	$\sim a$	$= 1 - a$
threshold negation:	$\sim(a;\theta)$	$= \begin{cases} 1, & \text{if } \; x \leq \theta, \\ 0, & \text{otherwise.} \end{cases}$
cosine negation:	$\sim a$	$= \frac{1}{2}(1 + \cos \pi a)$
Sugeno negation:	$\sim(a;\lambda)$	$= \dfrac{1 - a}{1 + \lambda a}$
Yager negation:	$\sim(a;\lambda)$	$= (1 - a^{\lambda})^{\frac{1}{\lambda}}$



standard cosine Sugeno Yager

Reasoning with Uncertainty Module (RUM)

Motivation:

$$\text{modus ponens (mp):} \quad \frac{A \to B, A}{B}, \qquad \text{modus tollens (mt):} \quad \frac{A \to B, \neg B}{\neg A}$$

Generalization of mp and mt on $[0,1]$-valued propositions, e. g.:

$$\mu_{\text{tall}}(x) \xrightarrow{0.8} \mu_{\text{heavy}}(x), \mu_{\text{tall}}(x) \geq 0.9 \quad \Rightarrow \quad \mu_{\text{heavy}} \geq \,?$$

Reasoning with Uncertainty Module (2)

Modus Ponens: $[\![\cdot]\!]$ fulfillment degree

- Given: $[\![A \to B]\!] \geq \gamma$; $[\![A]\!] \geq \alpha$
- Desired: $[\![B]\!] \geq \beta = \beta(\gamma, \alpha)$
- $[\![B]\!] \geq [\![A \wedge (A \to B)]\!] = \top([\![A]\!], [\![A \to B]\!]) \geq \top(\alpha, \gamma) = \beta$

Modus Tollens:

- Given: $[\![B]\!] \leq \beta$,$[\![A \to B]\!] \geq \gamma$
- Desired: $[\![A]\!] \leq \alpha = \alpha(\beta, \gamma)$
- $[\![\neg A]\!] \geq [\![\neg B \wedge (A \to B)]\!] = \top(\sim(/B/), [\![A \to B]\!]) \geq \top(\sim(\beta), \gamma)$

$\Rightarrow$ $[\![A]\!] = [\![\neg\neg A]\!] = \sim([\![\neg A]\!]) \leq \sim(\top(\sim(\beta), \gamma)) = \bot(\beta, \sim(\gamma))$

Possibility Theory

- The best-known calculus for handling uncertainty is, of course, **probability theory**. [Laplace 1812]
- An less well-known, but noteworthy alternative is **possibility theory**. [Dubois and Prade 1988]
- In the interpretation we consider here, possibility theory can handle **uncertain and imprecise information**, while probability theory, at least in its basic form, was only designed to handle *uncertain information.*
- Types of **imperfect information**:
  - **Imprecision:** disjunctive or set-valued information about the obtaining state, which is certain: the true state is contained in the disjunction or set.
  - **Uncertainty:** precise information about the obtaining state (single case), which is not certain: the true state may differ from the stated one.
  - **Vagueness:** meaning of the information is in doubt: the interpretation of the given statements about the obtaining state may depend on the user.

Possibility Theory: Axiomatic Approach

Definition: Let Ω be a (finite) sample space.
A **possibility measure** Π on Ω is a function $\Pi : 2^\Omega \to [0, 1]$ satisfying

1. $\Pi(\emptyset) = 0$ and
2. $\forall E_1, E_2 \subseteq \Omega : \Pi(E_1 \cup E_2) = \max\{\Pi(E_1), \Pi(E_2)\}$.

- Similar to Kolmogorov's axioms of probability theory.
- From the axioms follows $\Pi(E_1 \cap E_2) \leq \min\{\Pi(E_1), \Pi(E_2)\}$.
- Attributes are introduced as random variables (as in probability theory).
- $\Pi(A = a)$ is an abbreviation of $\Pi(\{\omega \in \Omega \mid A(\omega) = a\})$
- If an event E is possible without restriction, then $\Pi(E) = 1$.
  If an event E is impossible, then $\Pi(E) = 0$.

Interpretation of Degrees of Possibility [Gebhardt and Kruse 1993]

- Let Ω be the (nonempty) set of all possible states of the world, ω_0 the actual (but unknown) state.

- Let $C = \{c_1, \ldots, c_n\}$ be a set of contexts (observers, frame conditions etc.) and $(C, 2^C, P)$ a finite probability space (context weights).

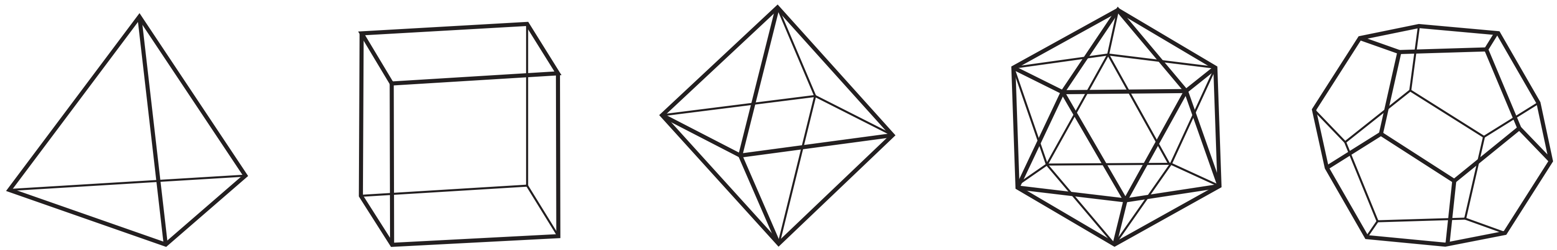
- Let $\Gamma : C \to 2^\Omega$ be a set-valued mapping, which assigns to each context the **most specific correct set-valued specification of** ω_0. The sets $\Gamma(c)$ are called the **focal sets** of Γ.

- Γ is a **random set** (i.e., a set-valued random variable) [Nguyen 1978]. The **basic possibility assignment** induced by Γ is the mapping

$$\begin{aligned} \pi : \Omega &\to [0, 1] \\ \pi(\omega) &\mapsto P(\{c \in C \mid \omega \in \Gamma(c)\}). \end{aligned}$$

Example: Dice and Shakers

shaker 1	shaker 2	shaker 3	shaker 4	shaker 5



tetrahedron	hexahedron	octahedron	icosahedron	dodecahedron
1 – 4	1 – 6	1 – 8	1 – 10	1 – 12

numbers	degree of possibility
1 – 4	$\frac{1}{5}+\frac{1}{5}+\frac{1}{5}+\frac{1}{5}+\frac{1}{5} = 1$
5 – 6	$\frac{1}{5}+\frac{1}{5}+\frac{1}{5}+\frac{1}{5} = \frac{4}{5}$
7 – 8	$\frac{1}{5}+\frac{1}{5}+\frac{1}{5} = \frac{3}{5}$
9 – 10	$\frac{1}{5}+\frac{1}{5} = \frac{2}{5}$
11 – 12	$\frac{1}{5} = \frac{1}{5}$

From the Context Model to Possibility Measures

Definition: Let $\Gamma : C \to 2^\Omega$ be a random set.
The **possibility measure** induced by Γ is the mapping

$$\begin{aligned} \Pi : 2^\Omega &\to [0,1], \\ E &\mapsto P(\{c \in C \mid E \cap \Gamma(c) \neq \emptyset\}). \end{aligned}$$

Problem: From the given interpretation it follows only:

$$\forall E \subseteq \Omega : \quad \max_{\omega \in E} \pi(\omega) \;\leq\; \Pi(E) \;\leq\; \min\left\{1, \sum_{\omega \in E} \pi(\omega)\right\}.$$

	1	2	3	4	5
$c_1 : \frac{1}{2}$			●		
$c_2 : \frac{1}{4}$		●	●	●	
$c_3 : \frac{1}{4}$	●	●	●	●	●
π	0	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{4}$

	1	2	3	4	5
$c_1 : \frac{1}{2}$			●		
$c_2 : \frac{1}{4}$	●	●			
$c_3 : \frac{1}{4}$				●	●
π	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$

From the Context Model to Possibility Measures (cont.)

Attempts to solve the indicated problem:

- Require the focal sets to be **consonant**:
  Definition: Let $\Gamma : C \to 2^\Omega$ be a random set with $C = \{c_1, \ldots, c_n\}$. The focal sets $\Gamma(c_i)$, $1 \leq i \leq n$, are called **consonant**, iff there exists a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_n}$, $1 \leq i_1, \ldots, i_n \leq n$, $\forall 1 \leq j < k \leq n : i_j \neq i_k$, so that

  $$\Gamma(c_{i_1}) \subseteq \Gamma(c_{i_2}) \subseteq \ldots \subseteq \Gamma(c_{i_n}).$$

  $\to$ mass assignment theory [Baldwin *et al.* 1995]

  Problem: The "voting model" is not sufficient to justify consonance.

- Use the lower bound as the "most pessimistic" choice. [Gebhardt 1997]

  Problem: Basic possibility assignments represent negative information, the lower bound is actually the *most optimistic* choice.

- Justify the lower bound from decision making purposes.

From the Context Model to Possibility Measures (cont.)

- Assume that in the end we have to decide on a single event.
- Each event is described by the values of a set of attributes.
- Then it can be useful to assign to a set of events the degree of possibility of the "most possible" event in the set.

Example:

$\sum$	36	18	18	28	
28	0	0	0	28	28
18	18	0	0	0	18
18	18	0	0	0	18
36	0	18	18	0	18
	18	18	18	28	max

0	40	0	40
40	0	0	40
0	0	20	20
40	40	20	max

Possibility Distributions

Definition: Let $X = \{A_1, \ldots, A_n\}$ be a set of attributes defined on a (finite) sample space Ω with respective domains $\mathrm{dom}(A_i)$, $i = 1, \ldots, n$. A **possibility distribution** π_X over X is the restriction of a possibility measure Π on Ω to the set of all events that can be defined by stating values for all attributes in X. That is, $\pi_X = \Pi|_{\mathcal{E}_X}$, where

$$\begin{aligned}
\mathcal{E}_X &= \left\{ E \in 2^\Omega \;\middle|\; \exists a_1 \in \mathrm{dom}(A_1) : \ldots \exists a_n \in \mathrm{dom}(A_n) : \right. \\
&\qquad\qquad \left. E \mathrel{\widehat{=}} \bigwedge_{A_j \in X} A_j = a_j \right\} \\
&= \left\{ E \in 2^\Omega \;\middle|\; \exists a_1 \in \mathrm{dom}(A_1) : \ldots \exists a_n \in \mathrm{dom}(A_n) : \right. \\
&\qquad\qquad \left. E = \left\{ \omega \in \Omega \;\middle|\; \bigwedge_{A_j \in X} A_j(\omega) = a_j \right\} \right\}.
\end{aligned}$$

- Corresponds to the notion of a probability distribution.
- Advantage of this formalization: No index transformation functions are needed for projections, there are just fewer terms in the conjunctions.

A Possibility Distribution

all numbers in parts per 1000

	🟦	🟥	🟨	🟩	
	80	90	70	70	

large: 70

	🟦	🟥	🟨	🟩	
△	40	70	10	70	80
□	20	10	20	20	70
○	30	30	20	10	90

medium: 80

	🟦	🟥	🟨	🟩
△	40	80	10	70
□	30	10	70	60
○	60	60	20	10

small: 90

	🟦	🟥	🟨	🟩
△	20	20	10	20
□	30	10	40	40
○	80	90	20	10

	🟦	🟥	🟨	🟩
△	40	80	10	70
□	30	10	70	60
○	80	90	20	10

	s	m	l
△	20	80	70
□	40	70	20
○	90	60	30

	🟦	🟥	🟨	🟩
large	40	70	20	70
medium	60	80	70	70
small	80	90	40	40

- The numbers state the degrees of possibility of the corresp. value combination.

Reasoning

all numbers in parts per 1000

0	0	0	70

large

△	0	0	0	70	70
□	0	0	0	20	60
○	0	0	0	10	10

large: 70

medium

△	0	0	0	70
□	0	0	0	60
○	0	0	0	10

medium: 70

small

△	0	0	0	20
□	0	0	0	40
○	0	0	0	10

small: 40

△	0	0	0	70
□	0	0	0	60
○	0	0	0	10

	s	m	l
△	20	70	70
□	40	60	20
○	10	10	10

large	0	0	0	70
medium	0	0	0	70
small	0	0	0	40

- Using the information that the given object is green.

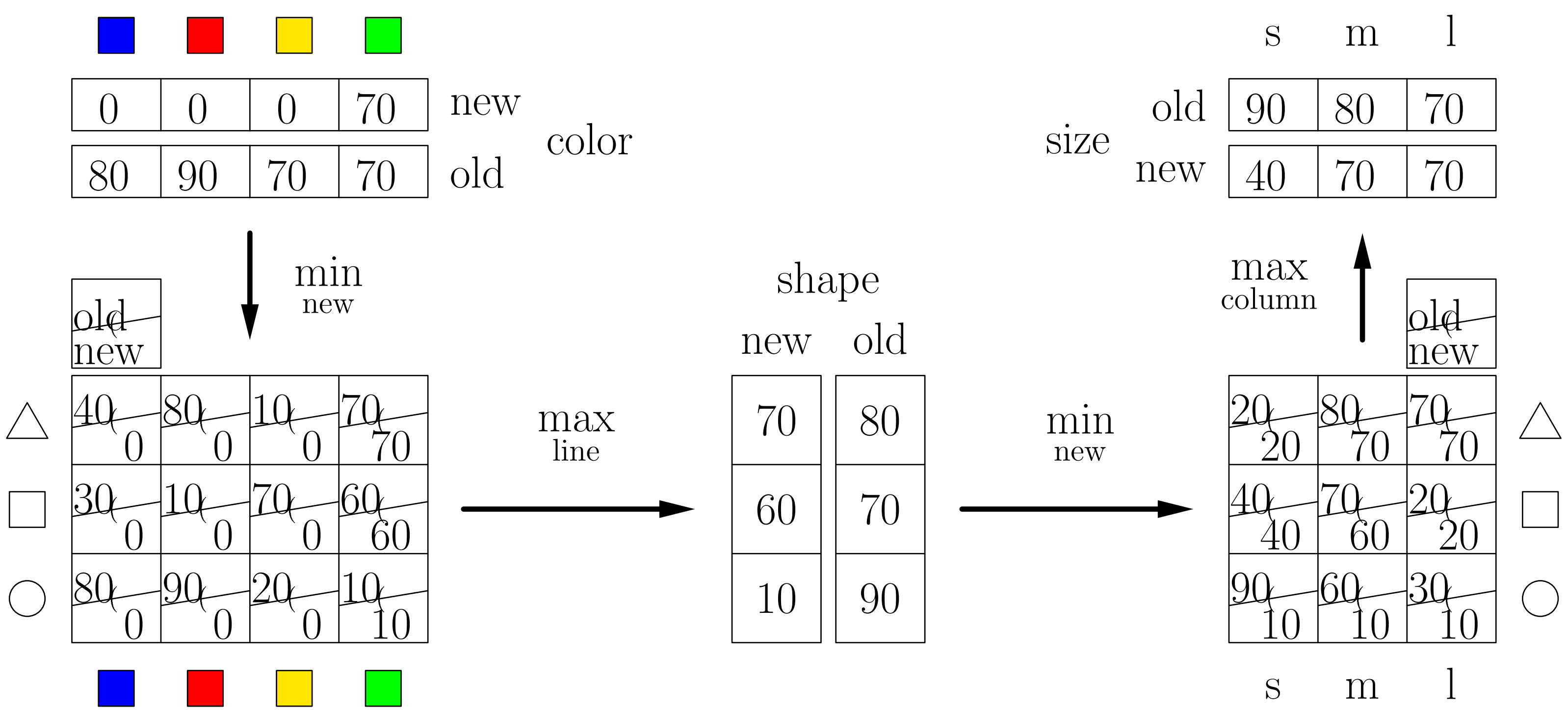
Possibilistic Decomposition

- As for relational and probabilistic networks, the three-dimensional possibility distribution can be decomposed into projections to subspaces, namely:
  - the maximum projection to the subspace color $\times$ shape and
  - the maximum projection to the subspace shape $\times$ size.
- It can be reconstructed using the following formula:

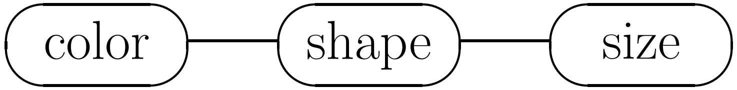
$$
\begin{aligned}
\forall i,j,k:\ \pi\left(a_i^{(\text{color})}, a_j^{(\text{shape})}, a_k^{(\text{size})}\right)
&= \min\left\{\pi\left(a_i^{(\text{color})}, a_j^{(\text{shape})}\right), \pi\left(a_j^{(\text{shape})}, a_k^{(\text{size})}\right)\right\} \\
&= \min\left\{\max_k \pi\left(a_i^{(\text{color})}, a_j^{(\text{shape})}, a_k^{(\text{size})}\right),\right. \\
&\qquad\quad \left.\max_i \pi\left(a_i^{(\text{color})}, a_j^{(\text{shape})}, a_k^{(\text{size})}\right)\right\}
\end{aligned}
$$

- Note the analogy to the probabilistic reconstruction formulas.

Again the same result can be obtained using only projections to subspaces (maximal degrees of possibility):



This justifies a graph representation:



Conditional Possibility and Independence

Definition: Let Ω be a (finite) sample space, Π a possibility measure on Ω, and $E_1, E_2 \subseteq \Omega$ events. Then

$$\Pi(E_1 \mid E_2) = \Pi(E_1 \cap E_2)$$

is called the **conditional possibility** of E_1 given E_2.

Definition: Let Ω be a (finite) sample space, Π a possibility measure on Ω, and A, B, and C attributes with respective domains $\mathrm{dom}(A)$, $\mathrm{dom}(B)$, and $\mathrm{dom}(C)$. A and B are called **conditionally possibilistically independent** given C, written $A \perp\!\!\!\perp_\Pi B \mid C$, iff

$$\begin{aligned} &\forall a \in \mathrm{dom}(A) : \forall b \in \mathrm{dom}(B) : \forall c \in \mathrm{dom}(C) : \\ &\qquad \Pi(A = a, B = b \mid C = c) = \min\{\Pi(A = a \mid C = c), \Pi(B = b \mid C = c)\}. \end{aligned}$$

- Similar to the corresponding notions of probability theory.

Possibilistic Evidence Propagation

$$
\begin{aligned}
&\pi(B = b \mid A = a_{\text{obs}}) \\
&= \pi\left(\bigvee_{a \in \text{dom}(A)} A = a, B = b, \bigvee_{c \in \text{dom}(C)} C = c \,\middle|\, A = a_{\text{obs}} \right) \\
&\overset{(1)}{=} \max_{a \in \text{dom}(A)} \{ \max_{c \in \text{dom}(C)} \{ \pi(A = a, B = b, C = c \mid A = a_{\text{obs}}) \} \} \\
&\overset{(2)}{=} \max_{a \in \text{dom}(A)} \{ \max_{c \in \text{dom}(C)} \{ \min\{ \pi(A = a, B = b, C = c), \pi(A = a \mid A = a_{\text{obs}}) \} \} \} \\
&\overset{(3)}{=} \max_{a \in \text{dom}(A)} \{ \max_{c \in \text{dom}(C)} \{ \min\{ \pi(A = a, B = b), \pi(B = b, C = c), \\
&\qquad\qquad \pi(A = a \mid A = a_{\text{obs}}) \} \} \} \\
&= \max_{a \in \text{dom}(A)} \{ \min\{ \pi(A = a, B = b), \pi(A = a \mid A = a_{\text{obs}}), \\
&\qquad \underbrace{\max_{c \in \text{dom}(C)} \{ \pi(B = b, C = c) \}}_{= \pi(B=b) \geq \pi(A=a, B=b)} \} \} \\
&= \max_{a \in \text{dom}(A)} \{ \min\{ \pi(A = a, B = b), \pi(A = a \mid A = a_{\text{obs}}) \} \}
\end{aligned}
$$

A:	color
B:	shape
C:	size

References

References

[Baldwin *et. al* 1995]
J.F. Baldwin, T.P. Martin, and B.W. Pilsworth.
FRIL — Fuzzy and Evidential Reasoning in Artificial Intelligence. Research Studies Press/J. Wiley & Sons, Taunton/Chichester, United Kingdom, 1995

[Borgelt *et al.* 2009]
Christian Borgelt, Matthias Steinbrecher and Rudolf Kruse.
Graphical Models — Representations for Learning, Reasoning and Data Mining, 2nd edition, John Wiley & Sons Ltd, United Kingdom, 2009

[Castillo *et al.* 1997]
Enrique Castillo, José Manuel Gutiérrez and Ali S. Hadi.
Expert Systems and Probabilistic Network Models, Springer Verlag, 1997

[Cooper and Herskovits 1992]
Gregory F. Cooper and Edward Herskovits.
A Bayesian Method for the Induction of Probabilistic Networks from Data, Machine Learning, Volume 9, pp. 309–347, 1992

References

[Dubois and Prade 1988]
D. Dubois and H. Prade.
Possibility Theory. Plenum Press, New York, NY, USA, 1988

[Gebhardt 1997]
J. Gebhardt.
Learning from Data: Possibilistic Graphical Models. Habilitation Thesis, University of Braunschweig, Germany, 1997

[Gebhardt and Kruse 1993]
J. Gebhardt and R. Kruse.
The Context Model — **An Integrating View of Vagueness and Uncertainty**. Int. Journal of Approximate Reasoning 9:283–314, North-Holland, Amsterdam, Netherlands, 1993

References

[Kruse *et. al* 1991]
Rudolf Kruse, Erhard Schwecke and Jochen Heinsohn.
Uncertainty and Vagueness in Knowledge Based Systems, Springer Verlag, 1991

[Laplace 1812]
P.S. Laplace
Théorie analytique des probabilités. Paris: Veuve Courcier, 1812

[Nguyen 1978]
H.T. Nguyen.
On Random Sets and Belief Functions. Journal of Mathematical Analysis and Applications 65:531–542. Academic Press, San Diego, CA, USA, 1978

Homepages

- Otto-von-Guericke-University of Magdeburg
  `http://www.uni-magdeburg.de/`

- School of Computer Science
  `http://www.cs.uni-magdeburg.de/`

- Computational Intelligence Group
  `http://fuzzy.cs.uni-magdeburg.de/`