



# Artificial Intelligence

Computer Science, CS541 - A

**Jonggi Hong**

# Announcements

- Team project

- Communicate with teammates.
  - Teammates will evaluate each other at the end of the semester.

- HW #2: Logic

- Due 11:59 pm, Tuesday, March 7.



# Recap

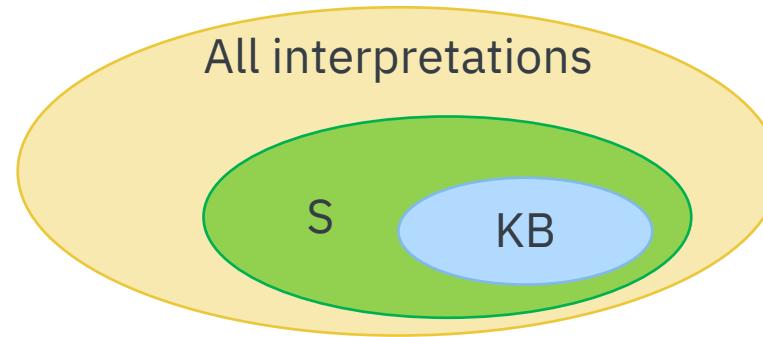
## Logic



# Entailment in First-Order Logic

KB entails S

- **For every interpretation I, if KB holds in I, then S holds in I.**



- Computing entailment is impossible in general because there are infinitely many possible interpretations.
- Even computing holds is impossible for interpretations with infinite universes.

# Proof and Entailment

- Entailment captures general notion of “follows from.”
- We can’t evaluate it directly by enumerating interpretations.
- So, we will do proofs.
- In FOL, if  $S$  is entailed by  $KB$ , then there is a finite proof of  $S$  from  $KB$ .

# First-Order Resolution

## 1. Convert to clausal form

$$\forall x. \neg P(x) \vee Q(x)$$

$$\frac{P(A)}{Q(A)}$$

$$\forall x. P(x) \rightarrow Q(x)$$

$$\frac{P(A)}{Q(A)}$$

## 2. Unification

Substitute A for x, still true

$$\neg P(A) \vee Q(A)$$

$$\frac{P(A)}{Q(A)}$$

## 3. Propositional resolution

# Converting to Clausal Form

## 1. Eliminate arrows

$$\alpha \leftrightarrow \beta \rightarrow (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$$

$$\alpha \rightarrow \beta \rightarrow \neg \alpha \vee \beta$$

## 2. Drive in negation

$$\neg(\alpha \vee \beta) \rightarrow \neg \alpha \wedge \neg \beta$$

$$\neg(\alpha \wedge \beta) \rightarrow \neg \alpha \vee \neg \beta$$

$$\neg \neg \alpha \rightarrow \alpha$$

$$\neg \forall x. \alpha \rightarrow \exists x. \neg \alpha$$

$$\neg \exists x. \alpha \rightarrow \forall x. \neg \alpha$$

## 3. Rename variables apart

$$\forall x. \exists y. (\neg P(x) \vee \exists x. Q(x,y)) \rightarrow \forall x_1. \exists y_2. (\neg P(x_1) \vee \exists x_3. Q(x_3,y_2))$$

# Converting to Clausal Form

## 4. Skolemize

Substitute a new name for each existential variable

$$\exists x. P(x) \rightarrow P(\text{Fred})$$

$$\exists xy. R(x,y) \rightarrow R(\text{Thing1}, \text{Thing2})$$

$$\exists x. P(x) \wedge Q(x) \rightarrow P(\text{Fleep}) \wedge Q(\text{Fleep})$$

$$(\exists x. P(x)) \wedge (\exists x. Q(x)) \rightarrow P(\text{Frog}) \wedge Q(\text{Grog})$$

$$\exists y. \forall x. \text{Loves}(x,y) \rightarrow \forall x. \text{Loves}(x, \text{Englebert})$$

Substitute a new function of all universal vars in outer scopes

$$\forall x. \exists y. \text{Loves}(x,y) \rightarrow \forall x. \text{Loves}(x, \text{Beloved}(x))$$

$$\forall x. \exists y. \forall z. \exists w. P(x, y, z) \wedge R(y, z, w) \rightarrow P(x, f(x), z) \wedge R(f(x), z, g(x, z))$$

# First-Order Resolution

## 1. Convert to Clausal Form

$$\forall x. \neg P(x) \vee Q(x)$$

$$\frac{P(A)}{Q(A)}$$

$$\frac{\forall x. P(x) \rightarrow Q(x)}{\frac{P(A)}{Q(A)}}$$

## 2. Unification

Substitute A for x, still true

$$\neg P(A) \vee Q(A)$$

$$\frac{P(A)}{Q(A)}$$

## 3. Propositional resolution

*The key is **finding the correct substitutions** for the variables.*

# Most General Unifier

$g$  is a **most general unifier (MGU)** of  $\omega_1$  and  $\omega_2$  iff for all unifiers  $s$ , there exists  $s'$  such that  $\omega_1 s = (\omega_1 g) s'$  and  $\omega_2 s = (\omega_2 g) s'$ .

$\omega_1$	$\omega_2$	MGU
$P(x)$	$P(A)$	$\{x/A\}$
$P(f(x), y, g(x))$	$P(f(x), x, g(x))$	$\{y/x\}$ or $\{x/y\}$
$P(f(x), y, g(y))$	$P(f(x), z, g(x))$	$\{y/x, z/x\}$
$P(g(f(v)), g(u))$	$P(x, x)$	$\{x/g(f(v)), u/f(v)\}$

# Curiosity Killed the Cat

1	D(Fido)	a				
2	O(J, Fido)	a				
3	$\neg D(y) \vee \neg O(x, y) \vee L(x)$	b				
4	$\neg L(x) \vee \neg A(y) \vee \neg K(x, y)$	c				
5	K(J, T) $\vee K(C, T)$	d				
6	C(T)	e				
7	$\neg C(x) \vee A(x)$	f				
8	$\neg K(C, T)$	Neg	12	$\neg L(J)$	10, 11	
9	K(J, T)	5, 8	13	$\neg D(y) \vee \neg O(J, y)$	3, 12 {x/J}	
10	A(T)	6, 7 {x/T}	14	$\neg D(Fido)$	13, 2 {y/Fido}	
11	$\neg L(J) \vee \neg A(T)$	4, 9 {x/J, y/T}	15	•	14, 1	

# Miscellaneous Logic Topics

- Factoring
- Green's trick
- Equality
- Completeness and decidability



# Bayesian Network

Chapter 13, 14



# Probabilistic Models

- Models describe how (a portion of) the world works.
- **Models are always simplifications**
  - May not account for every variable
  - May not account for all interactions between variables
  - “All models are wrong; but some are useful.”
    - George E. P. Box
- What do we do with probabilistic models?
  - We (or our agents) need to reason about unknown variables, given evidence.
  - Example: explanation (diagnostic reasoning)
  - Example: prediction (causal reasoning)
  - Example: value of information

# Independence

# Independence

- Two variables are **independent** if:  $\forall x, y : P(x, y) = P(x)P(y)$ 
  - This says that their joint distribution *factors* into a product two simpler distributions
  - Another form:
$$\forall x, y : P(x|y) = P(x)$$
  - We write:  $X \perp\!\!\!\perp Y$
- Independence is simplifying **modeling assumption**.
  - *Empirical* joint distributions: at best “close” to independent
  - What could we assume for {Weather, Traffic, Cavity, Toothache}?

# Example: Independence?

$P_1(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$P(T)$

T	P
hot	0.5
cold	0.5

$P_2(T, W)$

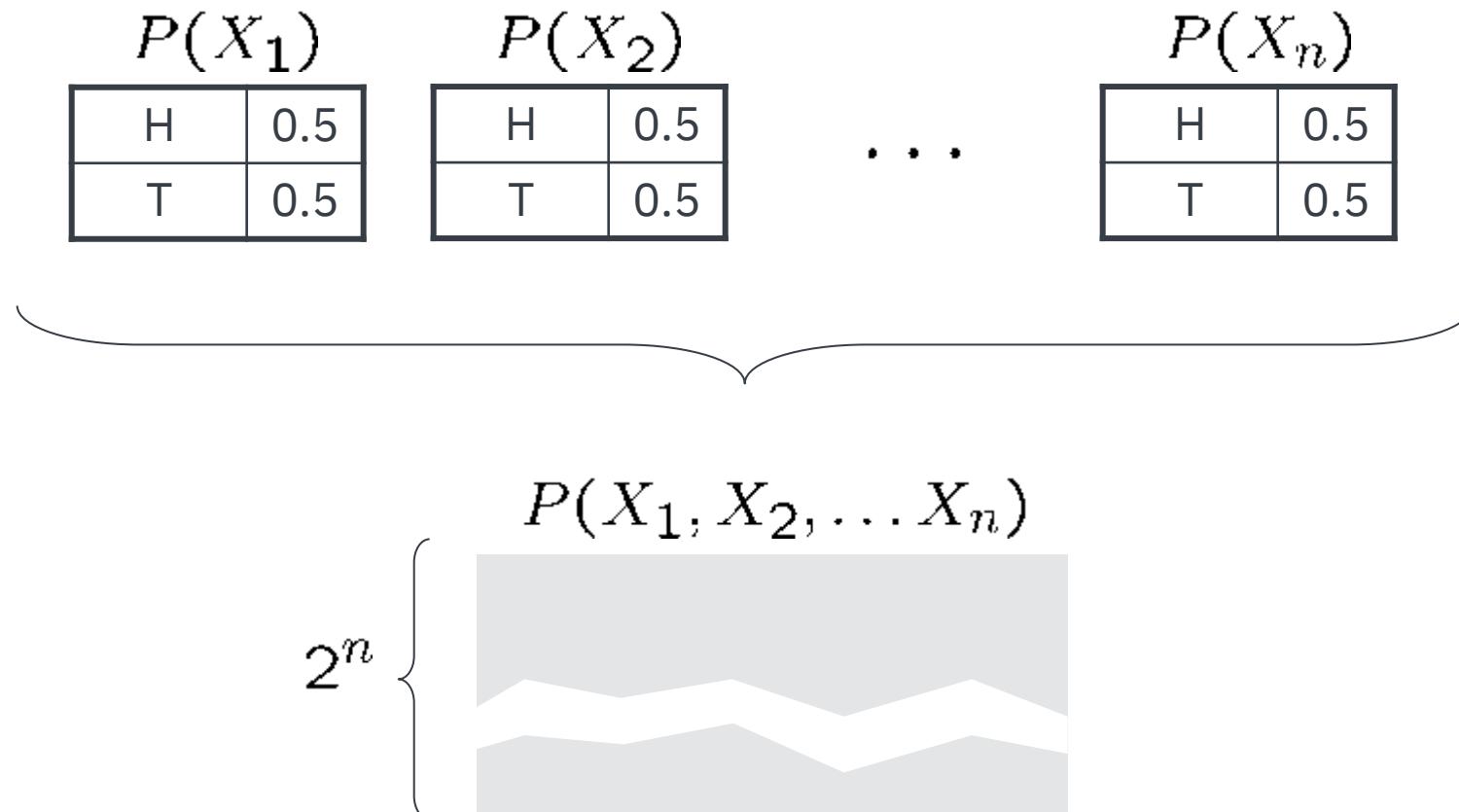
T	W	P
hot	sun	0.3
hot	rain	0.2
cold	sun	0.3
cold	rain	0.2

$P(W)$

W	P
sun	0.6
rain	0.4

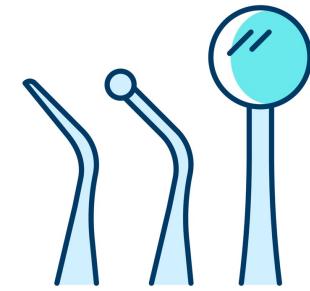
# Example: Independence

- N fair, independent coin flips:



# Conditional Independence

- $P(\text{Toothache}, \text{Cavity}, \text{Catch})$
- If I have a cavity, the probability that a probe catches it doesn't depend on whether I have a toothache:
  - $P(+\text{catch} | +\text{toothache}, +\text{cavity}) = P(+\text{catch} | +\text{cavity})$
- The same independence holds if I don't have a cavity:
  - $P(+\text{catch} | +\text{toothache}, -\text{cavity}) = P(+\text{catch} | -\text{cavity})$
- Catch is *conditionally independent* of Toothache given Cavity:
  - $P(\text{Catch} | \text{Toothache}, \text{Cavity}) = P(\text{Catch} | \text{Cavity})$
- Equivalent statements:
  - $P(\text{Toothache} | \text{Catch}, \text{Cavity}) = P(\text{Toothache} | \text{Cavity})$
  - $P(\text{Toothache}, \text{Catch} | \text{Cavity}) = P(\text{Toothache} | \text{Cavity}) P(\text{Catch} | \text{Cavity})$
  - One can be derived from the other easily.



# Conditional Independence

- Unconditional (absolute) independence very rare (why?)
- **Conditional independence** is our most basic and robust form of knowledge about uncertain environments.
- $X$  is conditionally independent of  $Y$  given  $Z$   
if and only if:

$$\forall x, y, z : P(x, y|z) = P(x|z)P(y|z)$$

or, equivalently, if and only if

$$\forall x, y, z : P(x|z, y) = P(x|z)$$

$$X \perp\!\!\!\perp Y | Z$$

# Conditional Independence

- What about this domain:

- Traffic
- Umbrella
- Raining

- What about this domain:

- Fire
- Smoke
- Alarm

# Conditional Independence and the Chain Rule

- Chain rule:

$$P(X_1, X_2, \dots, X_n) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\dots$$

- Trivial decomposition:

$$P(\text{Traffic, Rain, Umbrella}) =$$

$$P(\text{Rain})P(\text{Traffic}|\text{Rain})P(\text{Umbrella}|\text{Rain, Traffic})$$

- With assumption of conditional independence:

$$P(\text{Traffic, Rain, Umbrella}) =$$

$$P(\text{Rain})P(\text{Traffic}|\text{Rain})P(\text{Umbrella}|\text{Rain})$$

- Bayes' nets / graphical models help us express conditional independence assumptions

# Ghostbusters Chain Rule

- Each sensor depends only on where the ghost is.
- That means, the two sensors are conditionally independent, given the ghost position
- T: Top square is red  
B: Bottom square is red  
G: Ghost is in the top
- Givens:  
 $P(+g) = 0.5$   
 $P(-g) = 0.5$   
 $P(+t | +g) = 0.8$   
 $P(+t | -g) = 0.4$   
 $P(+b | +g) = 0.4$   
 $P(+b | -g) = 0.8$



$$P(T,B,G) = P(G) P(T|G) P(B|G)$$

T	B	G	$P(T,B,G)$
+t	+b	+g	0.16
+t	+b	-g	0.16
+t	-b	+g	0.24
+t	-b	-g	0.04
-t	+b	+g	0.04
-t	+b	-g	0.24
-t	-b	+g	0.06
-t	-b	-g	0.06

# Bayesian Network (Bayes' Net)

# Bayes' Nets: Big Picture

- Two problems with using full joint distribution tables as our probabilistic models:
  - Unless there are only a few variables, the joint is WAY too big to represent explicitly
  - Hard to learn (estimate) anything empirically about more than a few variables at a time
- **Bayesian network:** a technique for describing complex joint distributions (models) using simple, local distributions (conditional probabilities)
  - More properly called **graphical models**
  - We describe how variables locally interact
  - Local interactions chain together to give global, indirect interactions

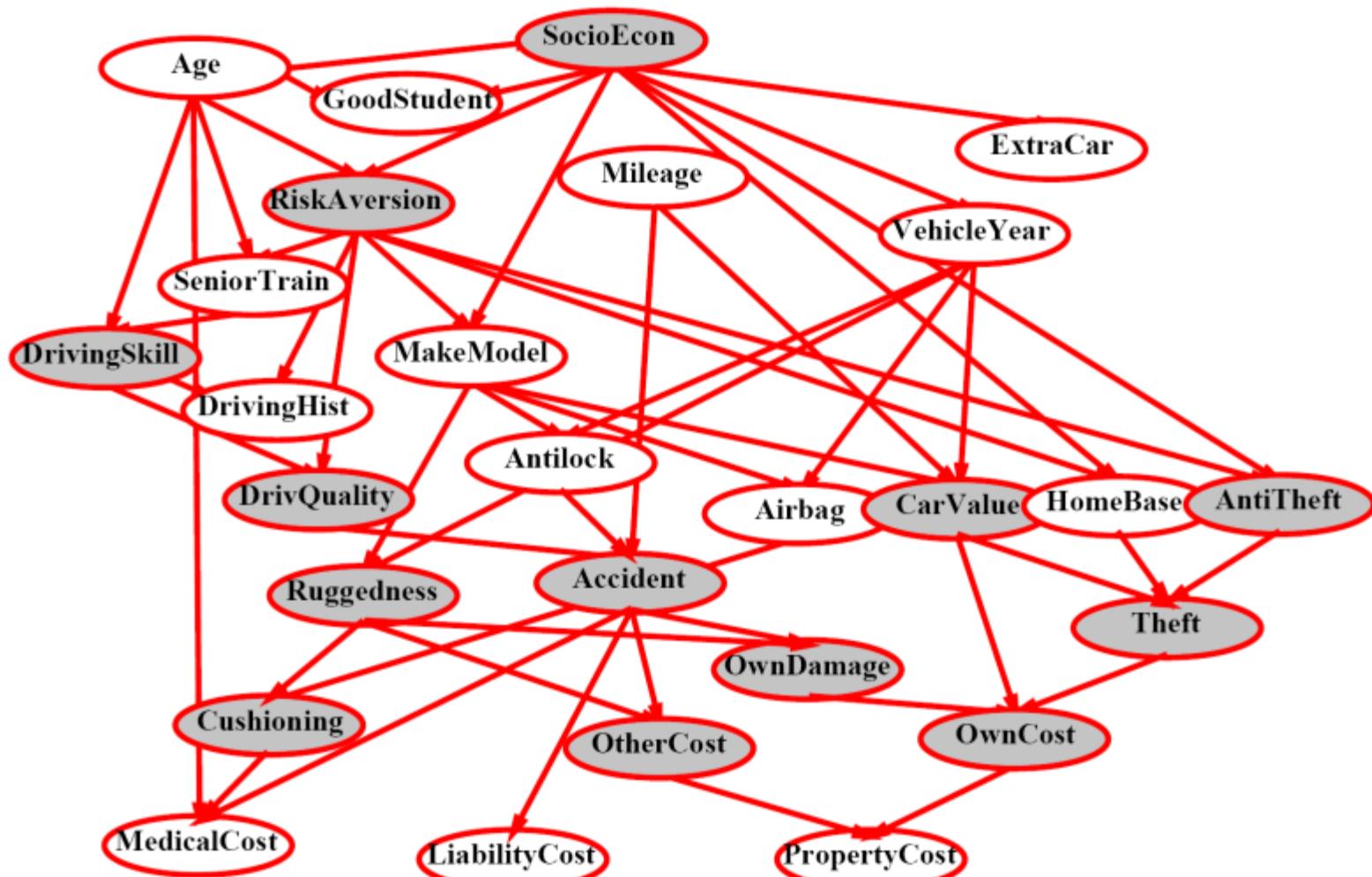
# Bayes' Nets

- Representation
- Conditional Independences
- Probabilistic Inference
  - Enumeration (exact, exponential complexity)
  - Variable elimination (exact, worst-case exponential complexity, often better)
  - Probabilistic inference is NP-complete
  - Sampling (approximate)
- Learning Bayes' Nets from Data

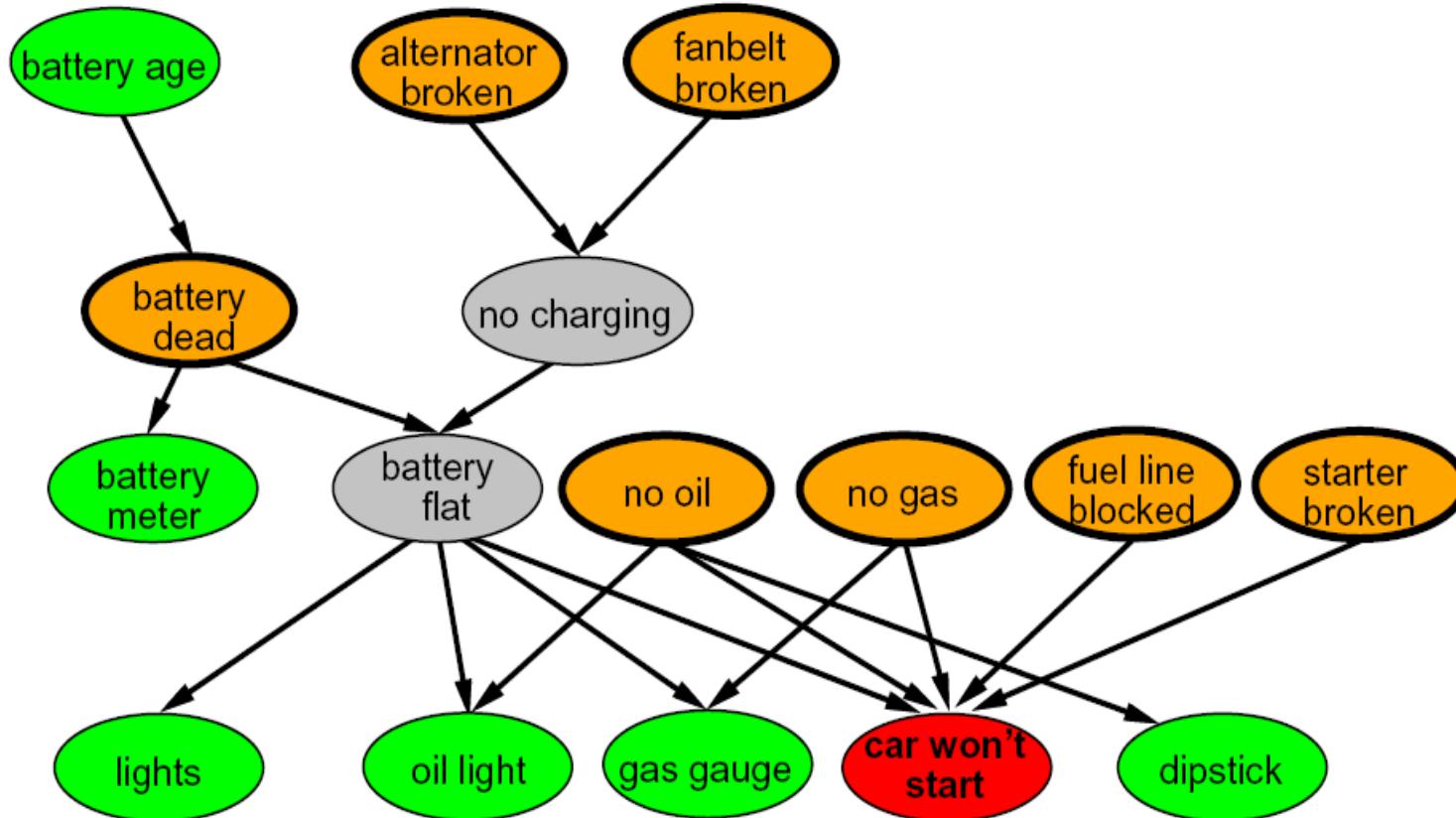
# Bayes' Nets

- Representation
- Conditional Independences
- Probabilistic Inference
  - Enumeration (exact, exponential complexity)
  - Variable elimination (exact, worst-case exponential complexity, often better)
  - Probabilistic inference is NP-complete
  - Sampling (approximate)
- Learning Bayes' Nets from Data

# Example Bayesian Network: Insurance

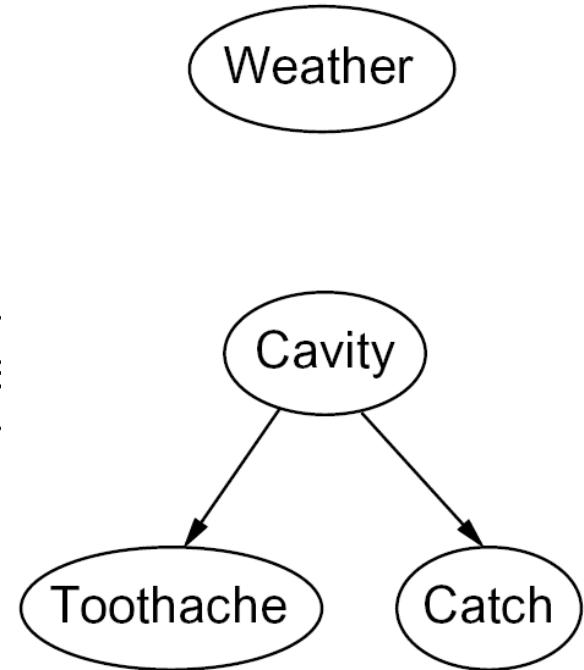


# Example Bayesian Network: Car



# Graphical Model Notation

- Nodes: variables (with domains)
  - Can be assigned (observed) or unassigned (unobserved)
- Arcs: interactions
  - Similar to CSP constraints
  - Indicate “direct influence” between variables
  - Formally: encode conditional independence (more later)
- For now: imagine that arrows mean direct causation (in general, they don’t!)



# Example: Coin Flips

- N independent coin flips



- No interactions between variables: **absolute independence**

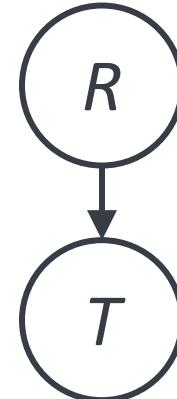
# Example: Traffic

- Variables:
  - R: It rains
  - T: There is traffic

Model 1: independence



Model 2: rain causes traffic



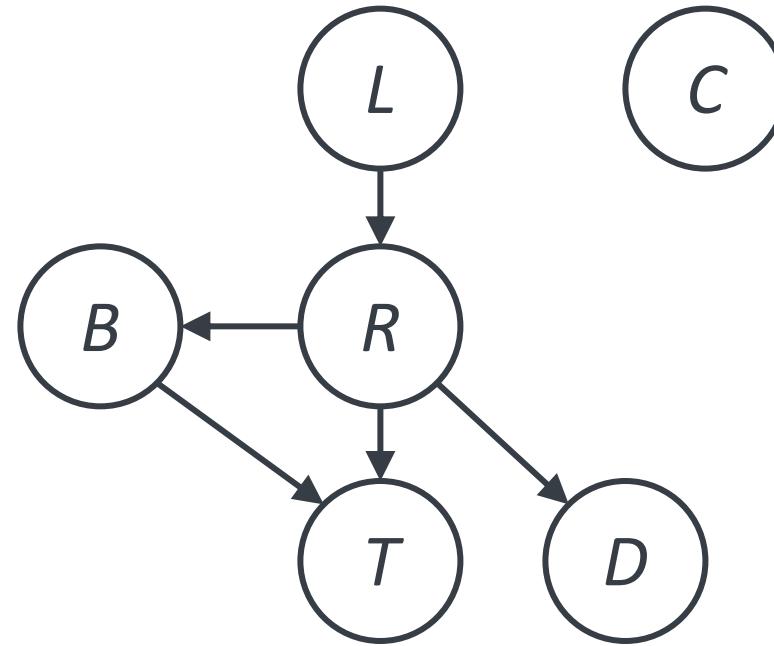
- Why is an agent using model 2 better?

# Example: Traffic II

- Let's build a causal graphical model!

- Variables

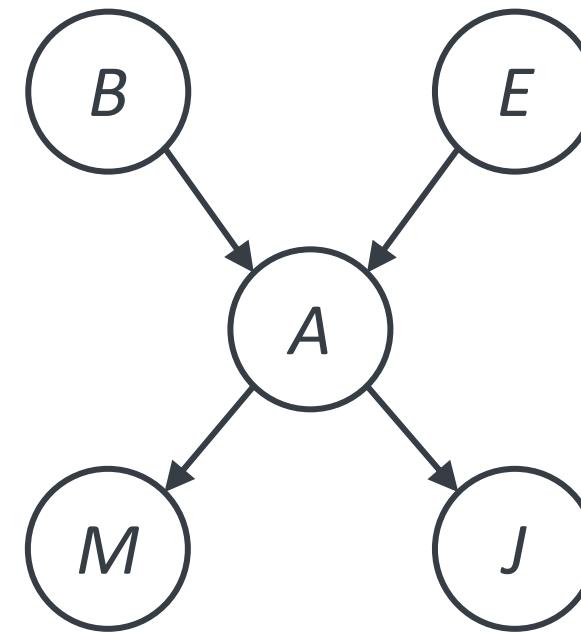
- T: Traffic
- R: It rains
- L: Low pressure
- D: Roof drips
- B: Ballgame
- C: Cavity



# Example: Alarm Network

## ■ Variables

- B: Burglary
- A: Alarm goes off
- M: Mary calls
- J: John calls
- E: Earthquake!



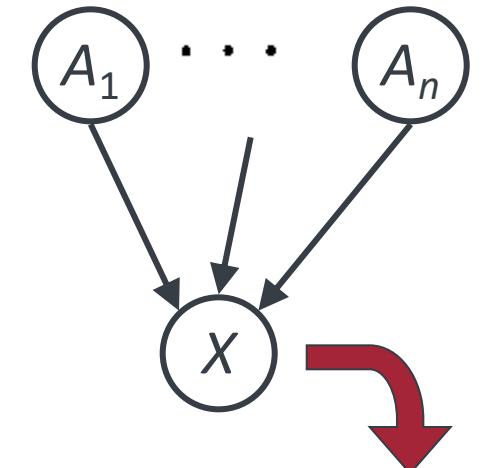
# Bayesian Network Semantics

# Bayesian Network Semantics

- A set of nodes, one per variable  $X$
- A directed, acyclic graph
- A conditional distribution for each node
  - A collection of distributions over  $X$ , one for each combination of parents' values

$$P(X|a_1 \dots a_n)$$

- CPT: conditional probability table
- Description of a noisy “causal” process



$$P(X|A_1 \dots A_n)$$

A Bayes net = Topology (graph) + Local Conditional Probabilities

# Probabilities in BNs

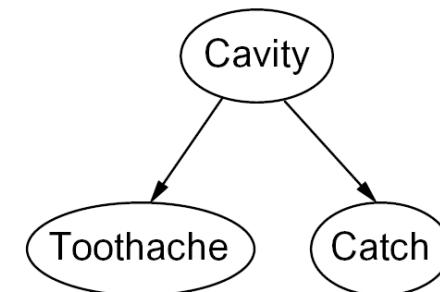
- Bayes' nets **implicitly** encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$$

- Example:

$P(+\text{cavity}, +\text{catch}, -\text{toothache})$

$= P(+\text{catch})P(+\text{catch}|\text{+cavity})P(-\text{toothache}|\text{+cavity})$



# Probabilities in BNs

- Why are we guaranteed that this equation results in a proper joint distribution?

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$$

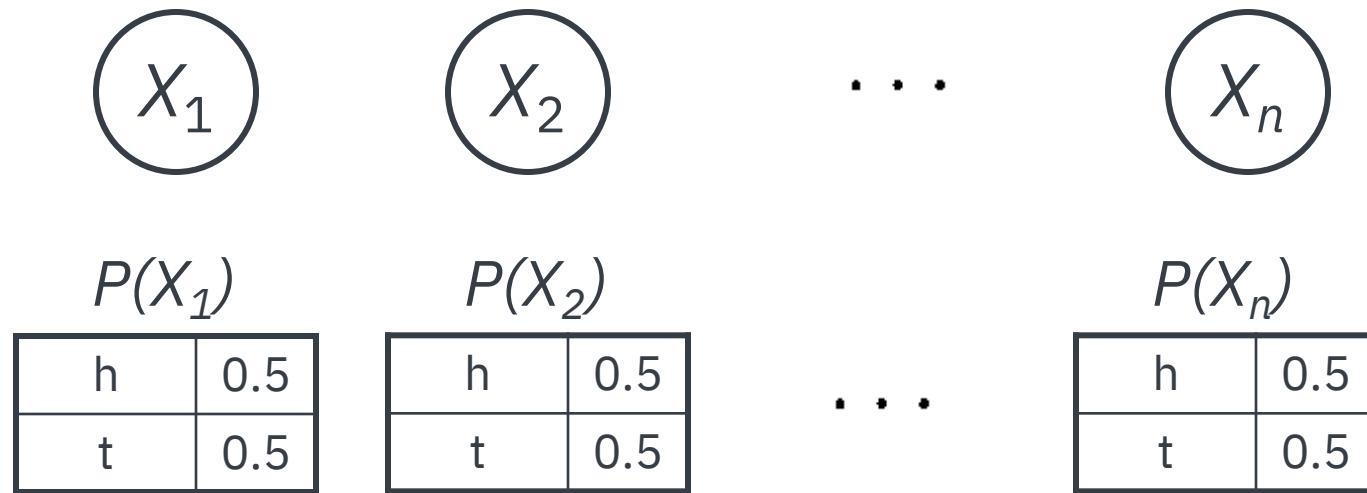
- Chain rule (valid for all distributions):  $P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | x_1 \dots x_{i-1})$

- Assume conditional independences:  $P(x_i | x_1, \dots, x_{i-1}) = P(x_i | \text{parents}(X_i))$

→ Consequence:  $P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$

- Not every BN can represent every joint distribution.
  - The topology enforces certain conditional independencies.

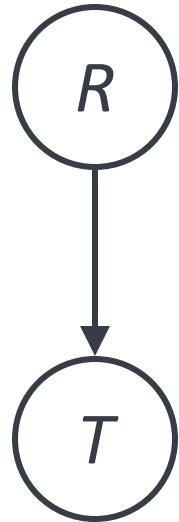
# Example: Coin Flips



$$P(h, h, t, h) = P(X_1=h)P(X_2=h)P(X_3=t)P(X_4=h)$$

*Only distributions whose variables are absolutely independent can be represented by a Bayesian network with no arcs.*

# Example: Traffic



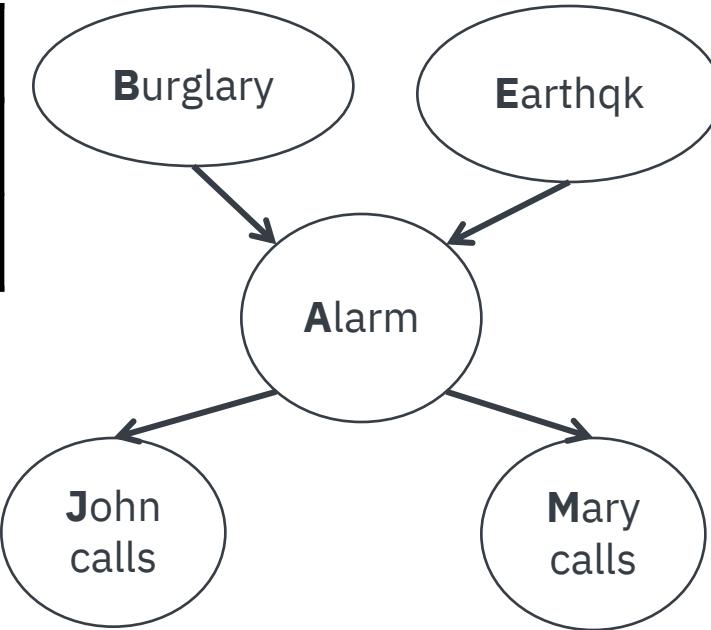
P(R)	
+r	1/4
-r	3/4

P(T R)	
+r	+t 3/4 -t 1/4
-r	+t 1/2 -t 1/2

$$\begin{aligned} P(+r, -t) &= P(+r)P(-t|r) \\ &= 1/4 * 1/4 \end{aligned}$$

# Example: Alarm Network

B	P(B)
+b	0.001
-b	0.999



E	P(E)
+e	0.002
-e	0.998

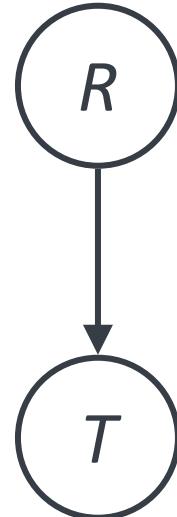
A	J	P(J A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

B	E	A	P(A B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

# Example: Traffic

- Causal direction



$P(R)$

+r	1/4
-r	3/4

$P(T|R)$

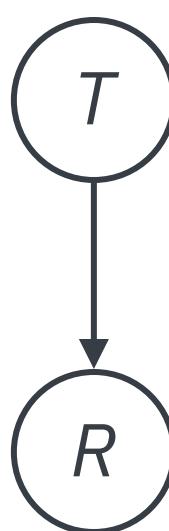
+r	+t	3/4
	-t	1/4
-r	+t	1/2
	-t	1/2

$P(T, R)$

+r	+t	3/16
+r	-t	1/16
-r	+t	6/16
-r	-t	6/16

# Example: Reverse Traffic

- Reverse causality?



$P(T)$

+t	9/16
-t	7/16

$P(R|T)$

+t	+r	1/3
	-r	2/3
-t	+r	1/7
	-r	6/7

$P(T, R)$

+r	+t	3/16
+r	-t	1/16
-r	+t	6/16
-r	-t	6/16

# Size of a Bayes' Net

- How big is a joint distribution over N Boolean variables?

$$2^N$$

- How big is an N-node net if nodes have up to k parents?

$$O(N * 2^{k+1})$$

- Both give you the power to calculate.

$$P(X_1, X_2, \dots, X_n)$$

- BNs: Huge space savings!
- Also easier to elicit local CPTs
- Also faster to answer queries (coming)

# Causality?

- When Bayes' nets reflect the true causal patterns:
  - Often simpler (nodes have fewer parents)
  - Often easier to think about
  - Often easier to elicit from experts
- BNs need not actually be causal
  - Sometimes no causal net exists over the domain (especially if variables are missing)
  - E.g., consider the variables *Traffic* and *Drips*
  - End up with arrows that reflect correlation, not causation
- What do the arrows really mean?
  - Topology may happen to encode causal structure.
  - **Topology really encodes conditional independence.**

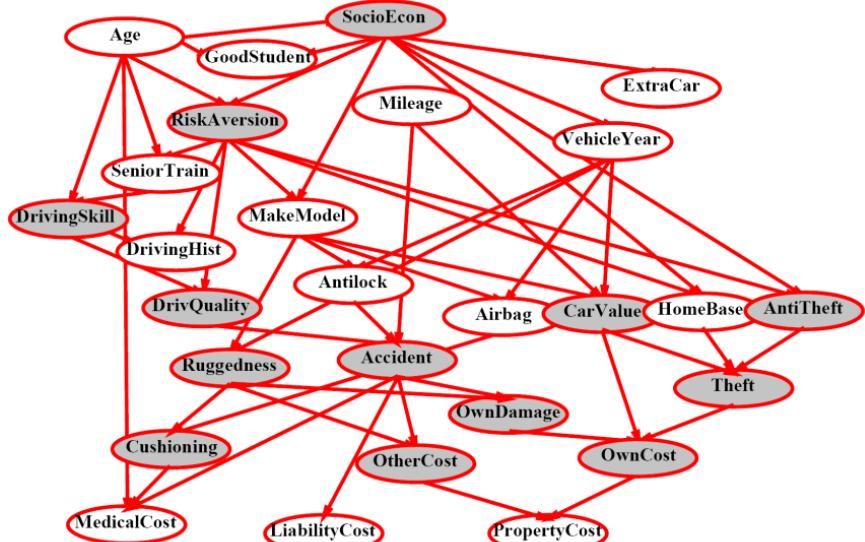
$$P(x_i|x_1, \dots, x_{i-1}) = P(x_i|\text{parents}(X_i))$$

# Bayes' Nets

- A Bayes' net is an efficient encoding of a probabilistic model of a domain.

- Questions we can ask:

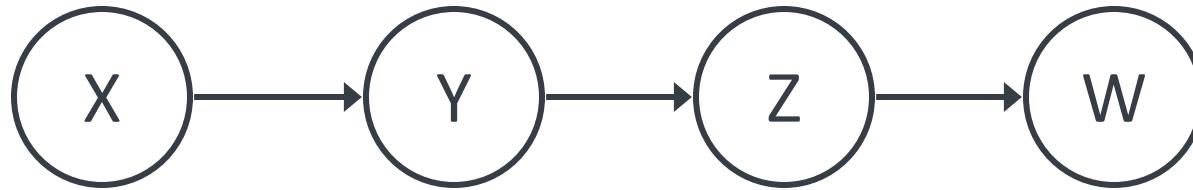
- Representation: given a BN graph, what kinds of distributions can it encode?
- Inference: given a fixed BN, what is  $P(X | e)$ ?
- Modeling: what BN is most appropriate for a given domain?



# Bayes' Nets

- Representation
- Conditional Independences
- Probabilistic Inference
  - Enumeration (exact, exponential complexity)
  - Variable elimination (exact, worst-case exponential complexity, often better)
  - Probabilistic inference is NP-complete
  - Sampling (approximate)
- Learning Bayes' Nets from Data

# Example



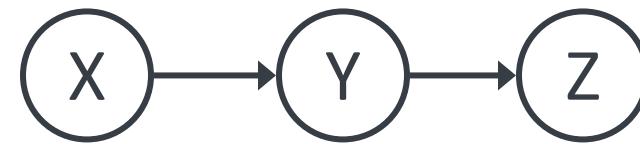
- Conditional independence assumptions directly from simplifications in chain rule:
- Additional implied conditional independence assumptions?
  - $P(W|X, Y) = P(W|Y)?$

$$\begin{aligned}\frac{P(w,x,y)}{P(x,y)} &= \sum_z \frac{P(x) P(y|x) P(z|y) P(w|z)}{P(x) P(y|x)} \\ &= \sum_z P(z|y) P(w|z) \\ &= \sum_z P(z|y) P(w|z, y) \\ &= \sum_z P(z, w|y) = P(w|y)\end{aligned}$$

# Independence in a BN

- Important question about a BN:

- Are two nodes independent given certain evidence?
- If yes, can prove using algebra (tedious in general)
- If no, can prove with a counter example
- Example:



- Question: are X and Z necessarily independent?
  - Answer: no. (Example: low pressure causes rain, which causes traffic.)
  - X can influence Z, Z can influence X (via Y)
  - Addendum: they *could* be independent: how?

# D-separation

# D-separation: Outline

- Study independence properties for triples
- Analyze complex cases in terms of member triples
- D-separation: a condition / algorithm for answering such queries

# Causal Chains

- This configuration is a “causal chain.”



X: Low pressure

Y: Rain

Z: Traffic

$$P(x, y, z) = P(x)P(y|x)P(z|y)$$

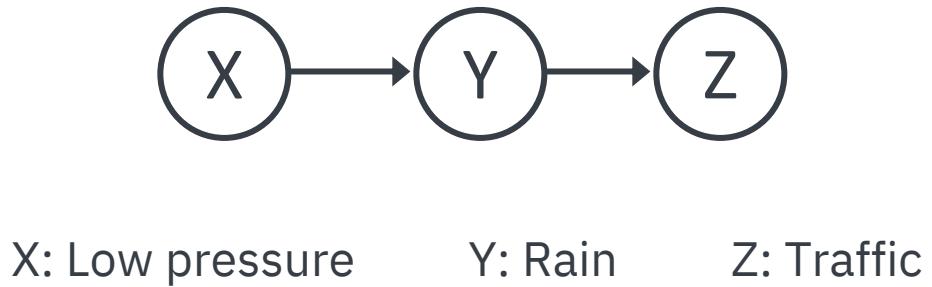
P(X)

P(Z)

- Guaranteed X independent of Z ? *No!*
  - One example set of CPTs for which X is not independent of Z is sufficient to show this independence is not guaranteed.
- Example:
  - Low pressure causes rain causes traffic, high pressure causes no rain causes no traffic
  - In numbers: $P(+y | +x) = 1, P(-y | -x) = 1,$  $P(+z | +y) = 1, P(-z | -y) = 1$

# Causal Chains

- This configuration is a “causal chain.”



$$P(x, y, z) = P(x)P(y|x)P(z|y)$$

- Guaranteed X independent of Z given Y?

$$\begin{aligned} P(z|x, y) &= \frac{P(x, y, z)}{P(x, y)} \\ &= \frac{P(x)P(y|x)P(z|y)}{P(x)P(y|x)} \end{aligned}$$

$$= P(z|y)$$

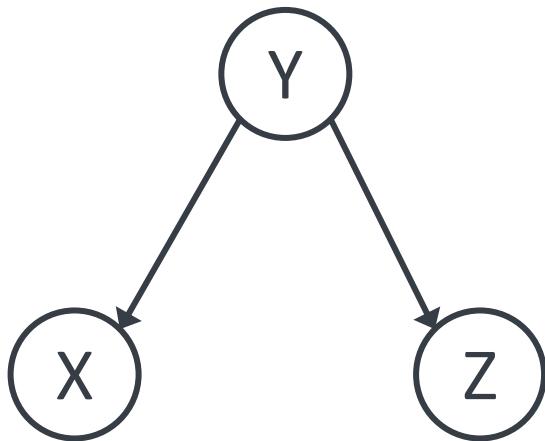
Yes!

- Evidence along the chain “blocks” the influence.

# Common Cause

- This configuration is a “common cause”

Y: Project due



X: Forums busy

Z: Lab full

$$P(x, y, z) = P(y)P(x|y)P(z|y)$$

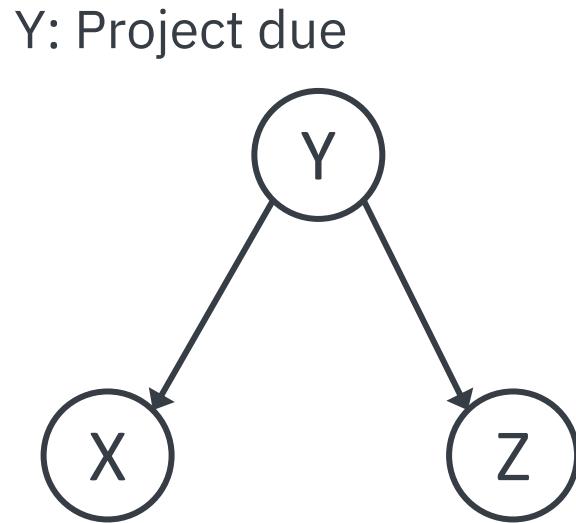
P(X)

P(Z)

- Guaranteed X independent of Z ? *No!*
  - One example set of CPTs for which X is not independent of Z is sufficient to show this independence is not guaranteed.
- Example:
  - Project due causes both forums busy and lab full
- In numbers:
$$P(+x | +y) = 1, P(-x | -y) = 1,$$
$$P(+z | +y) = 1, P(-z | -y) = 1$$

# Common Cause

- This configuration is a “common cause”
- Guaranteed X and Z independent given Y?



$$P(X|Y) \quad P(Z|Y)$$

$$\begin{aligned} P(z|x,y) &= \frac{P(x,y,z)}{P(x,y)} \\ &= \frac{P(y)P(x|y)P(z|y)}{P(y)P(x|y)} \\ &= P(z|y) \end{aligned}$$

X: Forums busy

Z: Lab full

Yes!

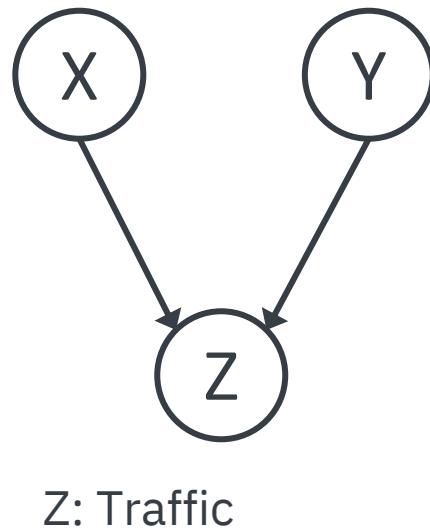
$$P(x,y,z) = P(y)P(x|y)P(z|y)$$

- Observing the cause blocks influence between effects.

# Common Effect

- Last configuration: two causes of one effect (v-structures)

X: Raining      Y: Ballgame



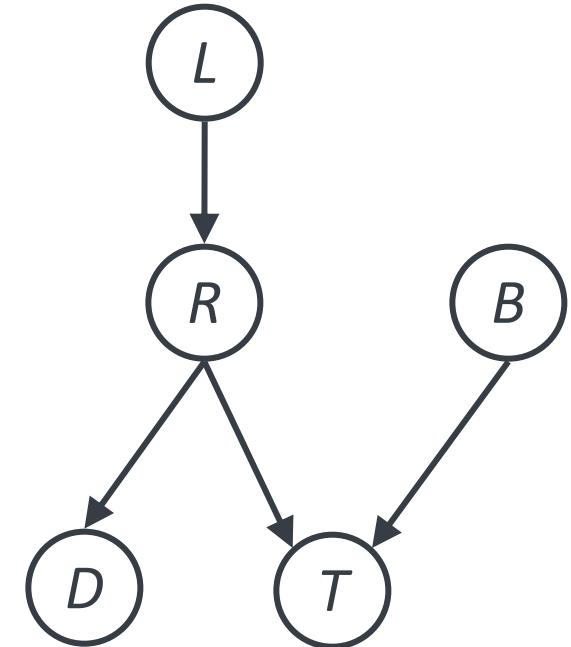
- Are X and Y independent?
  - Yes: the ballgame and the rain cause traffic, but they are not correlated.
  - Still need to prove they must be (try it!)
- Are X and Y independent given Z?
  - No: seeing traffic puts the rain and the ballgame in competition as explanation.
- This is backwards from the other cases.
  - Observing an effect activates influence between possible causes.

# The General Case

- General question: in a given BN, are two variables independent (given evidence)?
- Solution: analyze the graph
- Any complex example can be broken into repetitions of the three canonical cases.

# Reachability

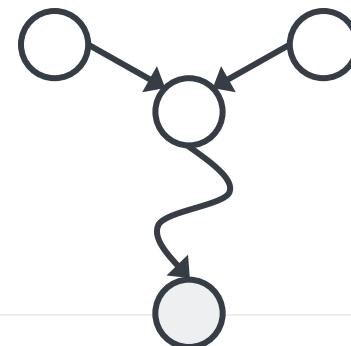
- Recipe: shade evidence nodes, look for paths in the resulting graph
- Attempt 1: if two nodes are connected by an undirected path not blocked by a shaded node, they are conditionally independent
- Almost works, but not quite
  - Where does it break?
  - Answer: the v-structure at T doesn't count as a link in a path unless “active”



# Active / Inactive Paths

- Question: Are X and Y conditionally independent given evidence variables  $\{Z\}$ ?
  - Yes, if X and Y “d-separated” by Z
  - Consider all (undirected) paths from X to Y
  - No active paths = independence!
- A path is active if each triple is active:
  - Causal chain  $A \rightarrow B \rightarrow C$  where B is unobserved (either direction)
  - Common cause  $A \leftarrow B \rightarrow C$  where B is unobserved
  - Common effect (aka v-structure)  
 $A \rightarrow B \leftarrow C$  where B or one of its descendants is observed
- All it takes to block a path is a single inactive segment.

Active Triples



Inactive Triples



# D-Separation

- **Query:**  $X_i \perp\!\!\!\perp X_j | \{X_{k_1}, \dots, X_{k_n}\}$  ?
- Check all (undirected!) paths between  $X_i$  and  $X_j$ 
  - If one or more active, then independence not guaranteed.

$$X_i \not\perp\!\!\!\perp X_j | \{X_{k_1}, \dots, X_{k_n}\}$$

- Otherwise (*i.e.*, if all paths are inactive), then independence is guaranteed.

$$X_i \perp\!\!\!\perp X_j | \{X_{k_1}, \dots, X_{k_n}\}$$

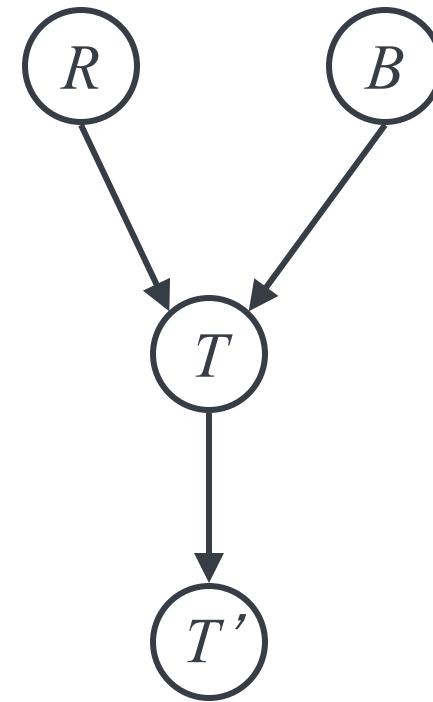
# Example

$R \perp\!\!\!\perp B$

*Yes*

$R \perp\!\!\!\perp B | T$

$R \perp\!\!\!\perp B | T'$



# Example

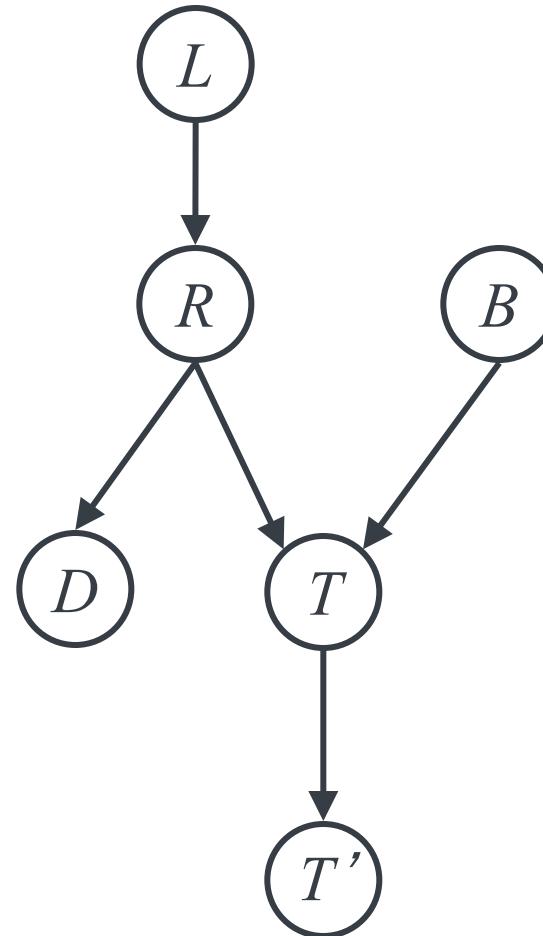
$L \perp\!\!\!\perp T' | T$       Yes

$L \perp\!\!\!\perp B$       Yes

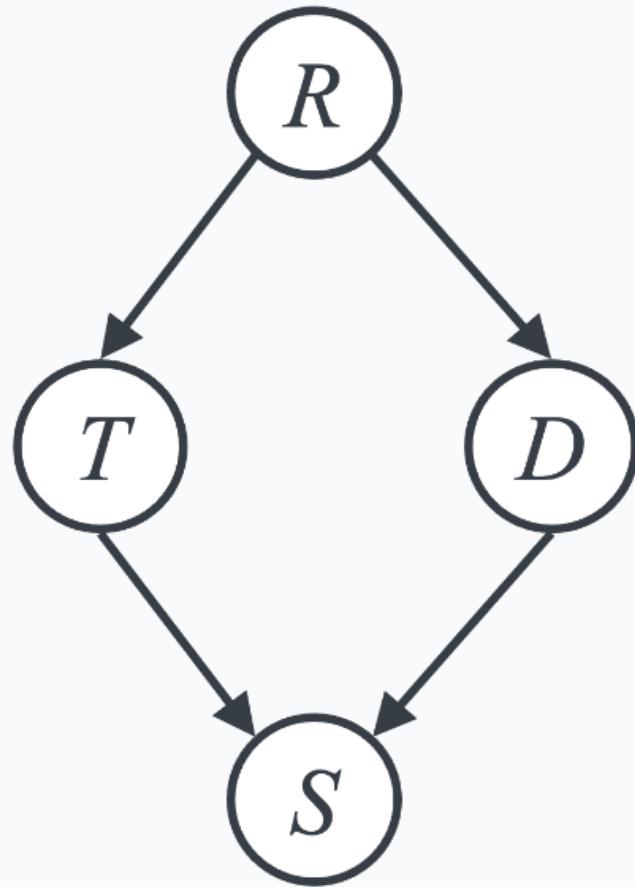
$L \perp\!\!\!\perp B | T$

$L \perp\!\!\!\perp B | T'$

$L \perp\!\!\!\perp B | T, R$       Yes



# Choose a pair with independence.



- $T \perp\!\!\!\perp D$
- $T \perp\!\!\!\perp D | R$
- $T \perp\!\!\!\perp D | R, S$
- None of the above

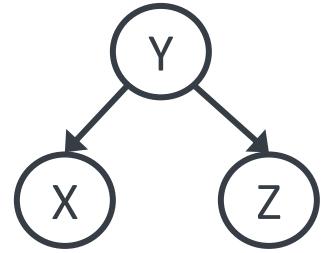
# Structure Implications

- Given a Bayes net structure, can run d-separation algorithm to build a complete list of conditional independences that are necessarily true of the form

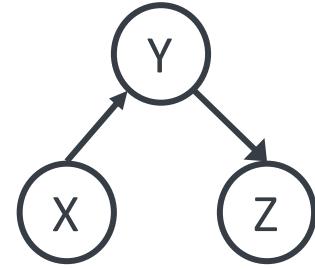
$$X_i \perp\!\!\!\perp X_j | \{X_{k_1}, \dots, X_{k_n}\}$$

- This list determines the set of probability distributions that can be represented.

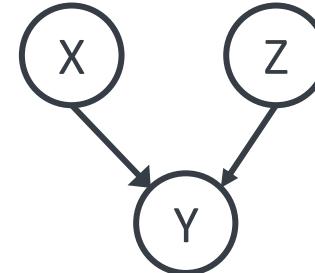
# Computing All Independences



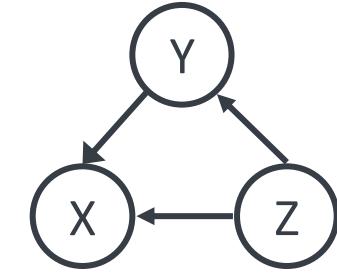
$\{X \perp\!\!\!\perp Z \mid Y\}$



$\{X \perp\!\!\!\perp Z \mid Y\}$



$\{X \perp\!\!\!\perp Z\}$

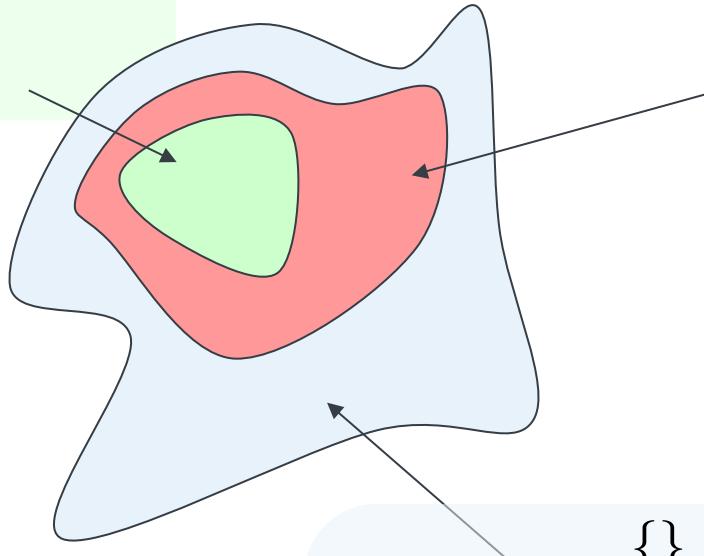
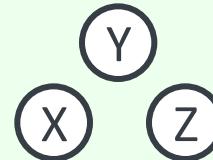


$\{\}$

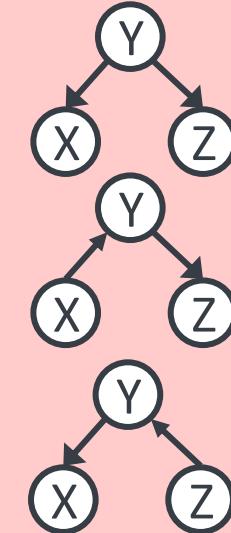
# Topology Limits Distributions

- Given some graph topology  $G$ , only certain joint distributions can be encoded.
- The graph structure guarantees certain (conditional) independences.
- (There might be more independence)
- Adding arcs increases the set of distributions, but has several costs.
- Full conditioning can encode any distribution.

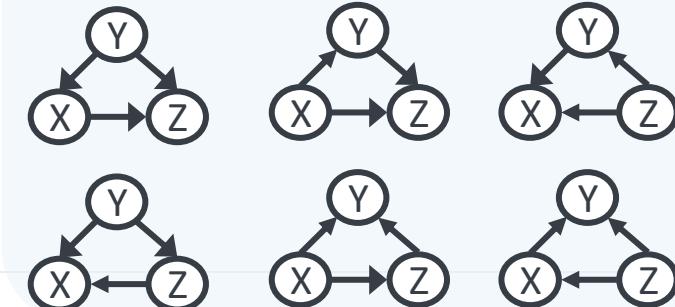
$$\{X \perp\!\!\!\perp Y, X \perp\!\!\!\perp Z, Y \perp\!\!\!\perp Z, \\ X \perp\!\!\!\perp Z \mid Y, X \perp\!\!\!\perp Y \mid Z, Y \perp\!\!\!\perp Z \mid X\}$$



$$\{X \perp\!\!\!\perp Z \mid Y\}$$



{}



# Bayes Nets Representation Summary

- Bayes nets compactly encode joint distributions.
- Guaranteed independencies of distributions can be deduced from BN graph structure.
- D-separation gives precise conditional independence guarantees from graph alone.
- A Bayes' net's joint distribution may have further (conditional) independence that is not detectable until you inspect its specific distribution.



STEVENS  
INSTITUTE OF TECHNOLOGY  
1870

15 min. break

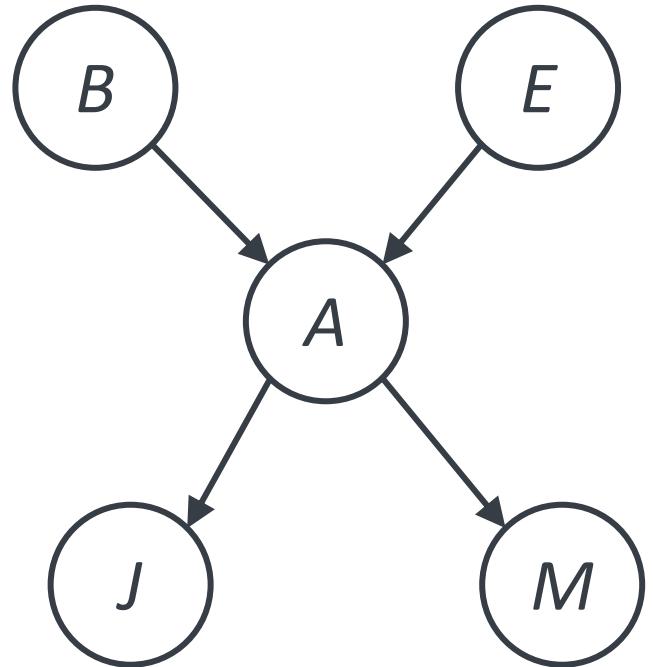


# Bayes' Nets

- Representation
- Conditional Independences
- Probabilistic Inference
  - Enumeration (exact, exponential complexity)
  - Variable elimination (exact, worst-case exponential complexity, often better)
  - Probabilistic inference is NP-complete
  - Sampling (approximate)
- Learning Bayes' Nets from Data

# Inference

- What is  $P(B|+j, +m)$ ?



# Inference

- Inference: calculating some useful quantity from a joint probability distribution
- Examples
  - Posterior priority  $P(Q|E_1 = e_1, \dots, E_k = e_k)$
  - Most likely explanation  $\operatorname{argmax}_q P(Q = q|E_1 = e_1 \dots)$

# Inference by Enumeration

- General case:

- Evidence variables:  $E_1 \dots E_k = e_1 \dots e_k$
  - Query\* variable:  $Q$
  - Hidden variables:  $H_1 \dots H_r$
- All variables*

- We want:  
*\* Works fine with multiple query variables, too*

$$P(Q|e_1 \dots e_k)$$

- Step 1: Select the entries consistent with the evidence

- Step 2: Sum out H to get joint of Query and evidence

- Step 3: Normalize

$$\times \frac{1}{Z}$$

$$Z = \sum_q P(Q, e_1 \dots e_k)$$

$$P(Q|e_1 \dots e_k) = \frac{1}{Z} P(Q, e_1 \dots e_k)$$

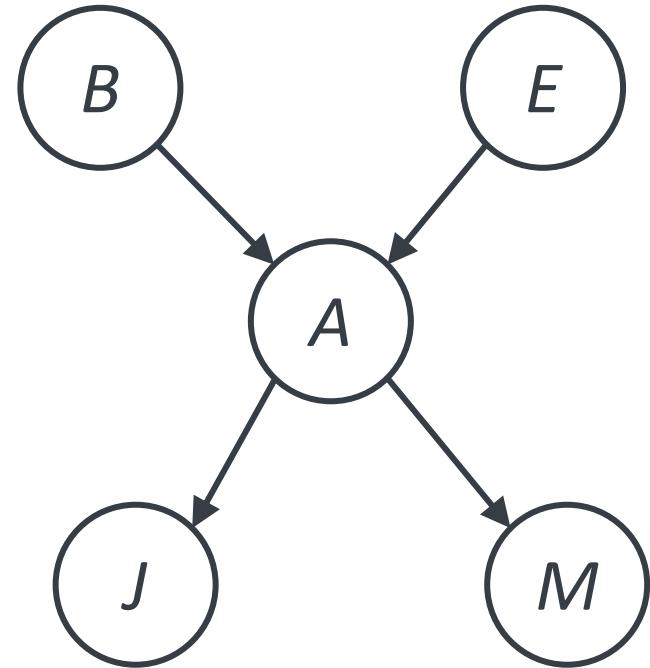
$$P(Q, e_1 \dots e_k) = \sum_{h_1 \dots h_r} \underbrace{P(Q, h_1 \dots h_r, e_1 \dots e_k)}_{X_1, X_2, \dots, X_n}$$

# Inference by Enumeration in Bayes' Net

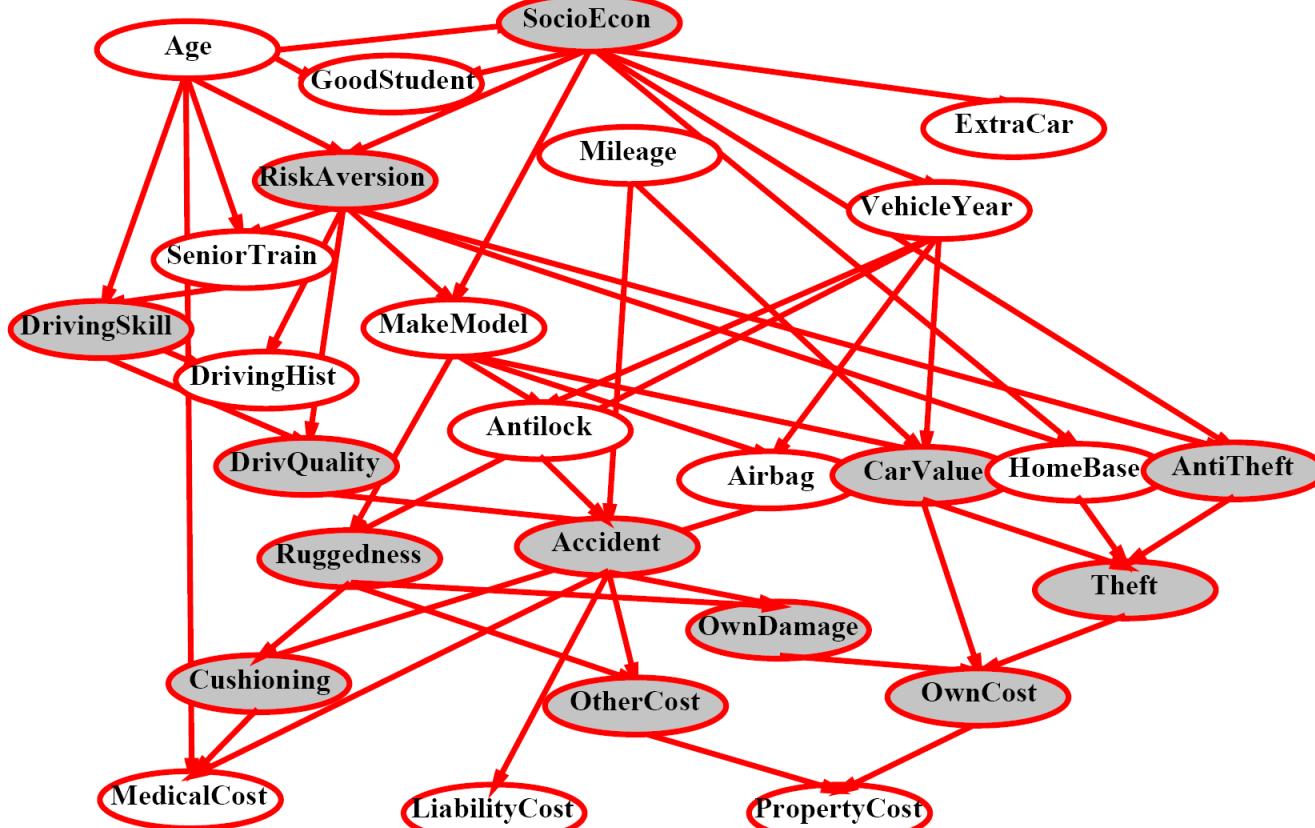
- Given unlimited time, inference in BNs is easy
- Reminder of inference by enumeration by example:

$$\begin{aligned} P(B \mid +j, +m) &\propto_B P(B, +j, +m) \\ &= \sum_{e,a} P(B, e, a, +j, +m) \\ &= \sum_{e,a} P(B)P(e)P(a|B, e)P(+j|a)P(+m|a) \end{aligned}$$

$$\begin{aligned} &= P(B)P(+e)P(+a|B, +e)P(+j|+a)P(+m|+a) + P(B)P(+e)P(-a|B, +e)P(+j|-a)P(+m|-a) \\ &\quad P(B)P(-e)P(+a|B, -e)P(+j|+a)P(+m|+a) + P(B)P(-e)P(-a|B, -e)P(+j|-a)P(+m|-a) \end{aligned}$$



# Inference by Enumeration?



$$P(\text{Antilock} | \text{observed variables}) = ?$$

# Inference by Enumeration vs. Variable Elimination

- Why is inference by enumeration so slow?
  - You join up the whole joint distribution before you sum out the hidden variables.
- Idea: interleave joining and marginalizing!
  - Called “Variable Elimination”
  - Still NP-hard, but usually much faster than inference by enumeration.

# Factor Zoo

# Factor Zoo I

- Joint distribution:  $P(X, Y)$ 
  - Entries  $P(x, y)$  for all  $x, y$
  - Sums to 1
- Selected joint:  $P(x, Y)$ 
  - A slice of the joint distribution
  - Entries  $P(x, y)$  for fixed  $x$ , all  $y$
  - Sums to  $P(x)$
- Number of capitals = dimensionality of the table

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$P(\text{cold}, W)$

T	W	P
cold	sun	0.2
cold	rain	0.3

# Factor Zoo II

- Single conditional:  $P(Y | x)$ 
  - Entries  $P(y | x)$  for fixed  $x$ , all  $y$
  - Sums to 1
- Family of conditionals:  $P(X | Y)$ 
  - Multiple conditionals
  - Entries  $P(x | y)$  for all  $x, y$
  - Sums to  $|Y|$

$P(W | cold)$

T	W	P
cold	sun	0.4
cold	rain	0.6

$P(W | T)$

T	W	P
hot	sun	0.8
hot	rain	0.2
cold	sun	0.4
cold	rain	0.6

$P(W | hot)$

$P(W | cold)$

# Factor Zoo III

- Specified family:  $P(y|X)$

- Entries  $P(y|x)$  for fixed  $y$ , but for all  $x$
- Sums to  $P(y)$

$P(rain|T)$

T	W	P
hot	rain	0.2
cold	rain	0.6

$$\begin{aligned} & P(rain|hot) \\ & P(rain|cold) \end{aligned}$$

# Factor Zoo Summary

- In general, when we write  $P(Y_1 \dots Y_N | X_1 \dots X_M)$ 
  - It is a “factor,” a multi-dimensional array
  - Its values are  $P(y_1 \dots y_N | x_1 \dots x_M)$
  - Any assigned (=lower-case) X or Y is a dimension missing (selected) from the array

# Variable Elimination (VE)

# Inference by Enumeration: Traffic Domain

- Random Variables
  - R: Raining
  - T: Traffic
  - L: Late for class!

$$P(L) = ?$$

$$= \sum_{r,t} P(r,t,L)$$

$$= \sum_{r,t} P(r)P(t|r)P(L|t)$$



$$P(R)$$

+r	0.1
-r	0.9

$$P(T|R)$$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$$P(L|T)$$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

# Inference by Enumeration: Procedural Outline

- Track objects called **factors**.
- Initial factors are local CPTs (one per node)

$P(R)$	
+r	0.1
-r	0.9

$P(T R)$		
+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L T)$		
+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- Any known values are selected
  - E.g. if we know  $L = +\ell$ , the initial factors are

$P(R)$	
+r	0.1
-r	0.9

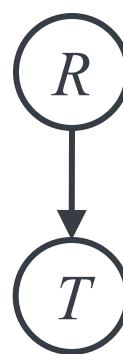
$P(T R)$		
+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(+\ell T)$		
+t	+l	0.3
-t	+l	0.1

- Procedure: **Join** all factors, then eliminate all hidden variables

# Operation 1: Join Factors

- First basic operation: **joining factors**
- Combining factors:
  - Just like a database join
  - Get all factors over the joining variable
  - Build a new factor over the union of the variables involved
- Example: Join on R
  - Computation for each entry: pointwise products


$$P(R) \times P(T|R)$$

+r	0.1
-r	0.9

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9



$$P(R, T)$$

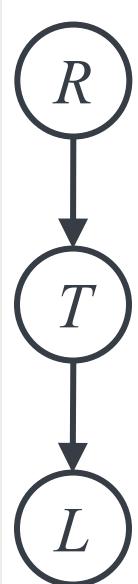
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81



$$\forall r, t : P(r, t) = P(r) \cdot P(t|r)$$

# Example: Multiple Joins

$P(R)$



+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join R

$P(R, T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

Join T

$R, T, L$

$P(R, T, L)$

+r	+t	+l	0.024
+r	+t	-l	0.056
+r	-t	+l	0.002
+r	-t	-l	0.018
-r	+t	+l	0.027
-r	+t	-l	0.063
-r	-t	+l	0.081
-r	-t	-l	0.729

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

# Operation 2: Eliminate

- Second basic operation: **marginalization**

- Take a factor and sum out a variable

- Shrinks a factor to a smaller one
  - A **projection** operation

- Example:

$$P(R, T)$$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

sum  $R$


$$P(T)$$

+t	0.17
-t	0.83

# Multiple Elimination

$P(R, T, L)$

$R, T, L$			
$+r$	$+t$	$+l$	0.024
$+r$	$+t$	$-l$	0.056
$+r$	$-t$	$+l$	0.002
$+r$	$-t$	$-l$	0.018
$-r$	$+t$	$+l$	0.027
$-r$	$+t$	$-l$	0.063
$-r$	$-t$	$+l$	0.081
$-r$	$-t$	$-l$	0.729

Sum  
out R



$T, L$

$P(T, L)$		
$+t$	$+l$	0.051
$+t$	$-l$	0.119
$-t$	$+l$	0.083
$-t$	$-l$	0.747

Sum  
out T



$L$

$P(L)$

$+l$	0.134
$-l$	0.886

# Variable Elimination

- Thus Far: Multiple Join, Multiple Eliminate
  - Inference by Enumeration
- Marginalizing Early
  - Variable Elimination

# Traffic Domain



$$P(L) = ?$$

- Inference by Enumeration

$$= \sum_t \sum_r P(L|t) P(r) P(t|r)$$

Join on r

Join on t

Eliminate r

Eliminate t

- Variable Elimination

$$= \sum_t P(L|t) \sum_r P(r) P(t|r)$$

Join on r

Eliminate r

Join on t

Eliminate t

# Marginalizing Early! (aka VE)

$P(R)$

+r	0.1
-r	0.9

Join R

$P(R, T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

Sum out R

$P(T)$

+t	0.17
-t	0.83

Join T

Sum out T



$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$R, T$

$L$

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

$T$

$L$

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

$T, L$

$P(T, L)$

+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

$L$

$P(L)$

+l	0.134
-l	0.866

# Evidence

- If evidence, start with factors with that evidence

- No evidence uses these initial factors:

$$P(R)$$

+r	0.1
-r	0.9

$$P(T|R)$$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$$P(L|T)$$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- Computing  $P(L|r)$ , the initial factors become:

$$P(+r)$$

+r	0.1
----	-----

$$P(T|+r)$$

+r	+t	0.8
+r	-t	0.2

$$P(L|T)$$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- We eliminate all vars other than query + evidence

# Evidence II

- Result will be a selected joint of query and evidence
  - E.g., for  $P(L | +r)$ , we would end up with:

$$P(+r, L)$$

+r	+l	0.026
+r	-l	0.074

Normalize

$$P(L | +r)$$



+l	0.26
-l	0.74

- To get our answer, just normalize this!
- That's it!

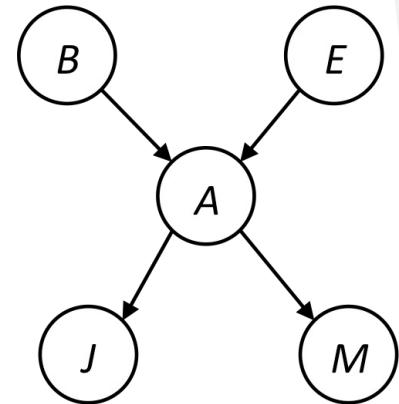
# General Variable Elimination

- Query:  $P(Q|E_1 = e_1, \dots, E_k = e_k)$
- Start with initial factors:
  - Local CPTs (but instantiated by evidence)
- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H
  - Join all factors mentioning H
  - Eliminate (sum out) H
- Join all remaining factors and normalize

# Example

$$P(B|j, m) \propto P(B, j, m)$$

$P(B)$	$P(E)$	$P(A B, E)$	$P(j A)$	$P(m A)$
--------	--------	-------------	----------	----------

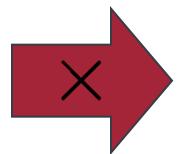


Choose A

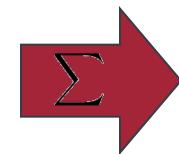
$$P(A|B, E)$$

$$P(j|A)$$

$$P(m|A)$$



$$P(j, m, A|B, E)$$



$$P(j, m|B, E)$$

$P(B)$	$P(E)$	$P(j, m B, E)$
--------	--------	----------------

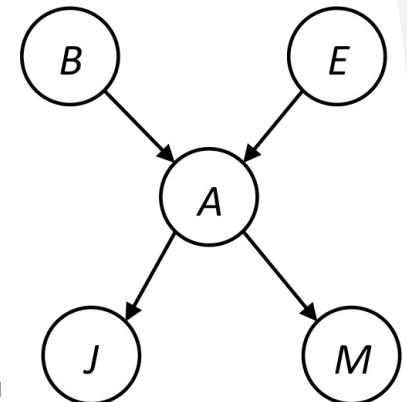
# Example

$$\boxed{P(B) \quad P(E) \quad P(j, m|B, E)}$$

Choose E

$$P(E) \quad \times \quad P(j, m, E|B) \quad \sum \quad P(j, m|B)$$

$P(j, m|B, E)$



$$\boxed{P(B) \quad P(j, m|B)}$$

Finish with B

$$P(B) \quad \times \quad P(j, m, B) \quad \xrightarrow{\text{Normalize}} \quad P(B|j, m)$$

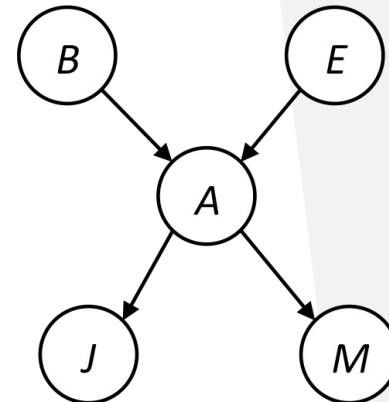
$P(j, m|B)$

# Same Example in Equations

$$P(B|j, m) \propto P(B, j, m)$$

$P(B)$	$P(E)$	$P(A B, E)$	$P(j A)$	$P(m A)$
--------	--------	-------------	----------	----------

$$\begin{aligned}
 P(B|j, m) &\propto P(B, j, m) && \text{marginal can be obtained from joint by summing out} \\
 &= \sum_{e,a} P(B, j, m, e, a) \\
 &= \sum_{e,a} P(B)P(e)P(a|B, e)P(j|a)P(m|a) && \text{use Bayes' net joint distribution expression} \\
 &= \sum_e P(B)P(e) \sum_a P(a|B, e)P(j|a)P(m|a) && \text{use } x^*(y+z) = xy + xz \\
 &= \sum_e P(B)P(e)f_1(B, e, j, m) && \text{joining on } a, \text{ and then summing out gives } f_1 \\
 &= P(B) \sum_e P(e)f_1(B, e, j, m) && \text{use } x^*(y+z) = xy + xz \\
 &= P(B)f_2(B, j, m) && \text{joining on } e, \text{ and then summing out gives } f_2
 \end{aligned}$$



# Another Variable Elimination Example

Query:  $P(X_3|Y_1 = y_1, Y_2 = y_2, Y_3 = y_3)$

Start by inserting evidence, which gives the following initial factors:

$$p(Z)p(X_1|Z)p(X_2|Z)p(X_3|Z)p(y_1|X_1)p(y_2|X_2)p(y_3|X_3)$$

Eliminate  $X_1$ , this introduces the factor  $f_1(Z, y_1) = \sum_{x_1} p(x_1|Z)p(y_1|x_1)$ , and we are left with:

$$p(Z)f_1(Z, y_1)p(X_2|Z)p(X_3|Z)p(y_2|X_2)p(y_3|X_3)$$

Eliminate  $X_2$ , this introduces the factor  $f_2(Z, y_2) = \sum_{x_2} p(x_2|Z)p(y_2|x_2)$ , and we are left with:

$$p(Z)f_1(Z, y_1)f_2(Z, y_2)p(X_3|Z)p(y_3|X_3)$$

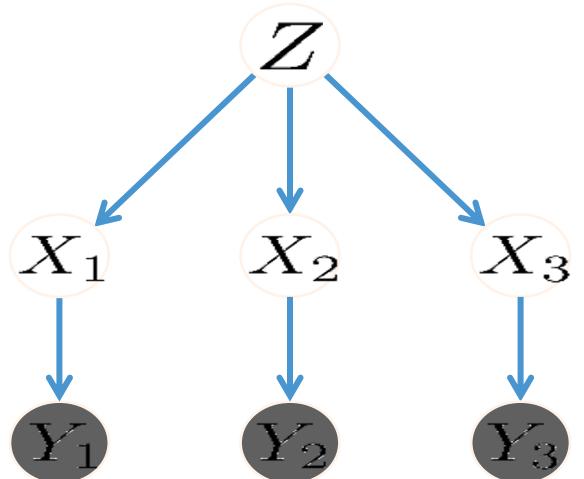
Eliminate  $Z$ , this introduces the factor  $f_3(y_1, y_2, X_3) = \sum_z p(z)f_1(z, y_1)f_2(z, y_2)p(X_3|z)$ , and we are left:

$$p(y_3|X_3), f_3(y_1, y_2, X_3)$$

No hidden variables left. Join the remaining factors to get:

$$f_4(y_1, y_2, y_3, X_3) = P(y_3|X_3)f_3(y_1, y_2, X_3).$$

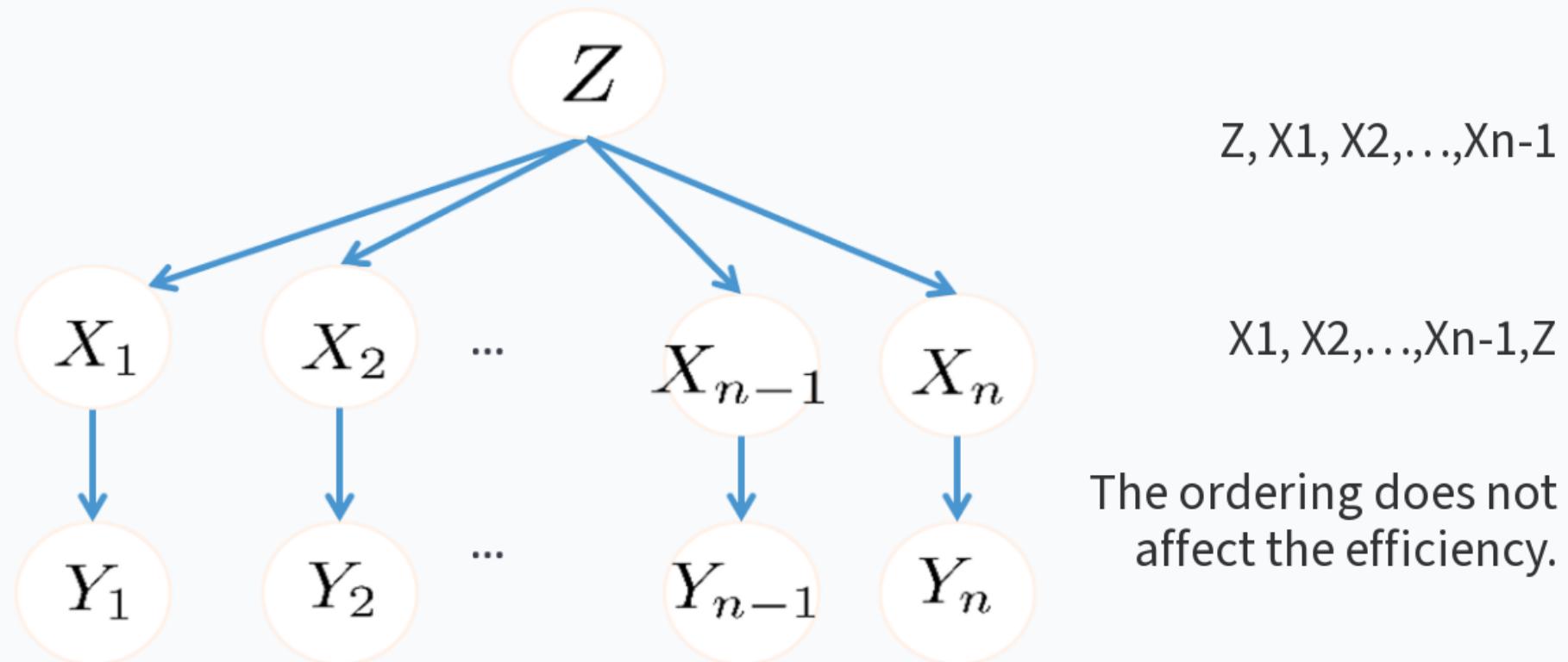
Normalizing over  $X_3$  gives  $P(X_3|y_1, y_2, y_3)$ .



**Computational complexity critically depends on the largest factor being generated in this process.**

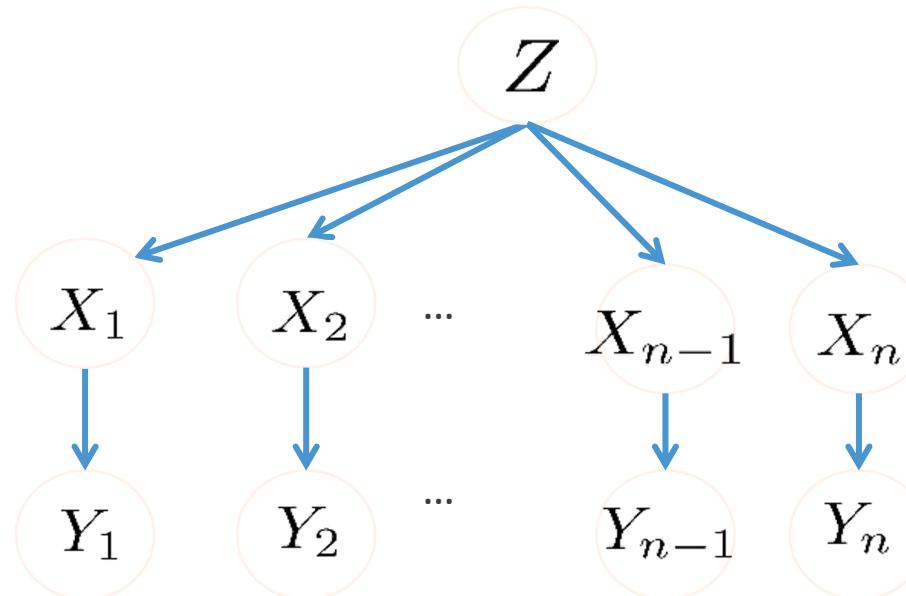
- Size of factor = number of entries in table.
- In example above (assuming binary) all factors generated are of size 2 --- as they all only have one.

# For the query $P(X_n|y_1, \dots, y_n)$ , which ordering is more efficient?



# Variable Elimination Ordering

- For the query  $P(X_n|y_1, \dots, y_n)$  work through the following two different orderings as done in previous slide:  $Z, X_1, \dots, X_{n-1}$  and  $X_1, \dots, X_{n-1}, Z$ . What is the size of the maximum factor generated for each of the orderings?



- Answer:  $2^{n+1}$  versus  $2^2$  (assuming binary)
- In general: the ordering can greatly affect efficiency.

# VE: Computational and Space Complexity

- The computational and space complexity of variable elimination is determined by the largest factor.
- The elimination ordering can greatly affect the size of the largest factor.
  - E.g., previous slide's example  $2^{n+1}$  vs.  $2^2$
- Does there always exist an ordering that only results in small factors?
  - **No!**

# Worst Case Complexity?

- CSP:

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee \neg x_4) \wedge (x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5) \wedge (x_2 \vee x_5 \vee x_7) \wedge (x_4 \vee x_5 \vee x_6) \wedge (\neg x_5 \vee x_6 \vee \neg x_7) \wedge (\neg x_5 \vee \neg x_6 \vee x_7)$$

$$P(X_i = 0) = P(X_i = 1) = 0.5$$

$$Y_1 = X_1 \vee X_2 \vee \neg X_3$$

$$\dots Y_8 = \neg X_5 \vee X_6 \vee X_7$$

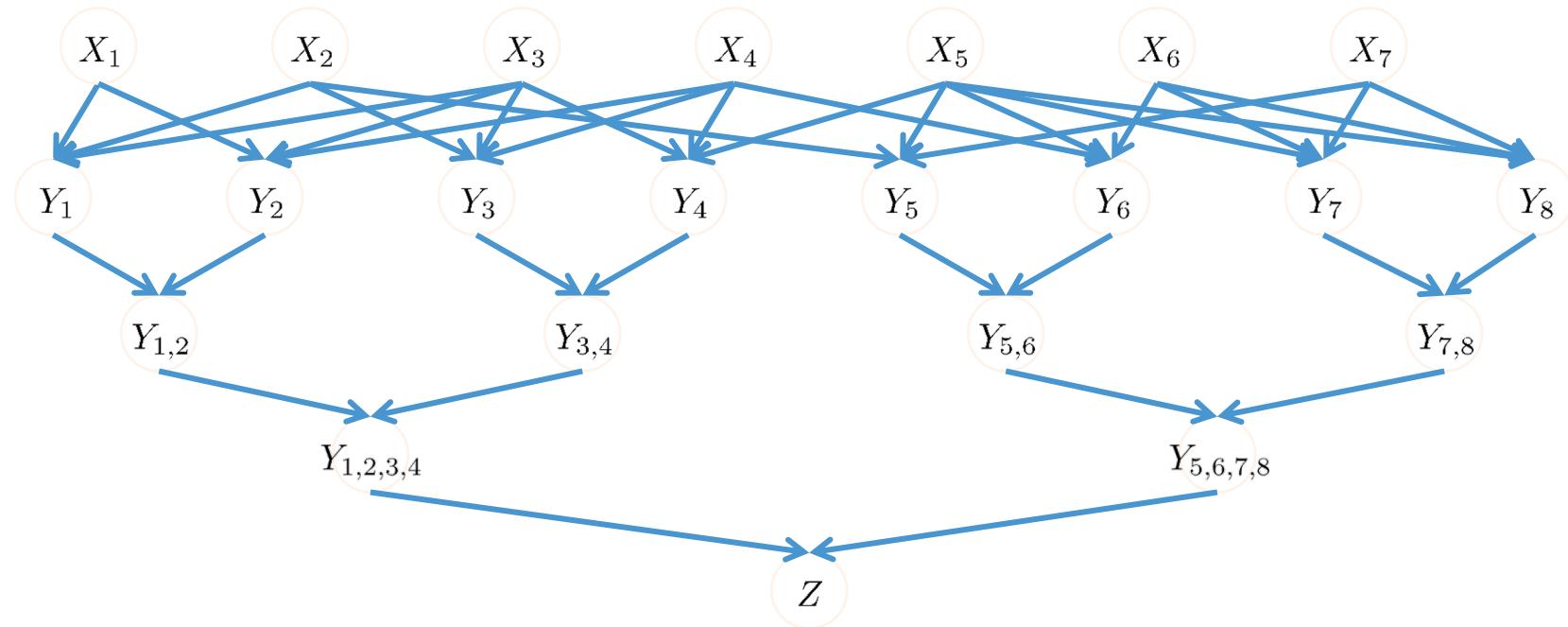
$$Y_{1,2} = Y_1 \wedge Y_2$$

$$\dots Y_{7,8} = Y_7 \wedge Y_8$$

$$Y_{1,2,3,4} = Y_{1,2} \wedge Y_{3,4}$$

$$Y_{5,6,7,8} = Y_{5,6} \wedge Y_{7,8}$$

$$Z = Y_{1,2,3,4} \wedge Y_{5,6,7,8}$$

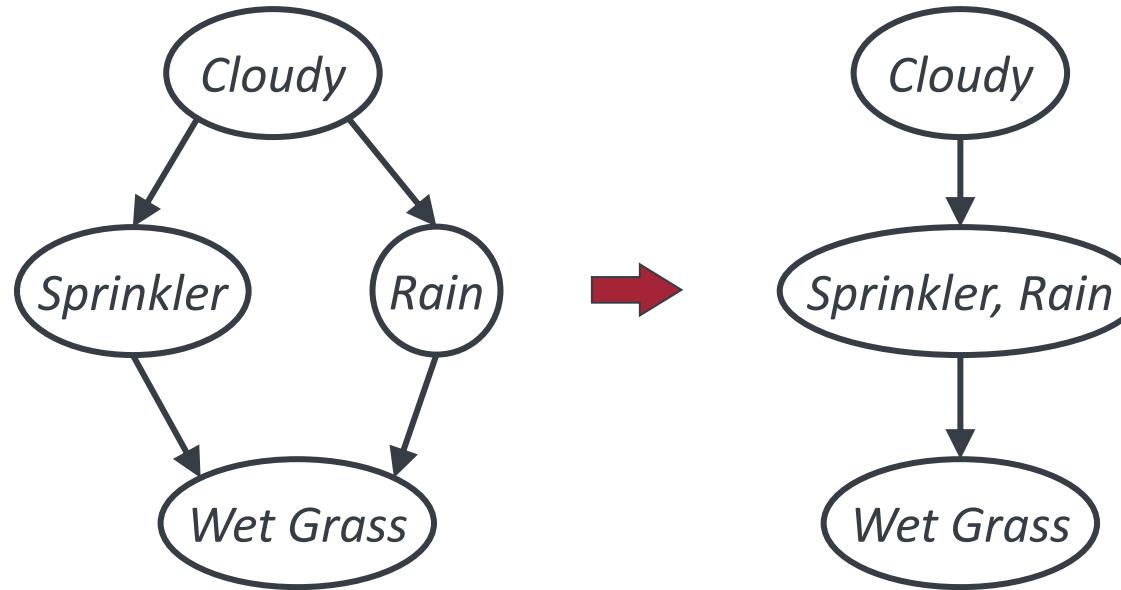


- If we can answer  $P(z)$  equal to zero or not, we answered whether the 3-SAT problem has a solution.
- Hence inference in Bayes' nets is NP-hard. No known efficient probabilistic inference in general.

# Polytrees

- A polytree is a directed graph with no undirected cycles.
- For poly-trees you can always find an ordering that is efficient.
  - The time and space complexity of exact inference in polytrees is linear in the size of the network.
- Cut-set conditioning for Bayes' net inference
  - Choose set of variables such that if removed only a polytree remains
  - Exercise: Think about how the specifics would work out!

# Polytrees



- The graph can be converted to a polytree by merging the variables.
  - The size of the CPTs increases.
- It is always possible to build a join tree but may require exponentially many combinations. of values.

# Bayes' Nets

- Representation
- Conditional Independences
- Probabilistic Inference
  - Enumeration (exact, exponential complexity)
  - Variable elimination (exact, worst-case exponential complexity, often better)
  - Probabilistic inference is NP-complete
  - Sampling (approximate)
- Learning Bayes' Nets from Data

# Approximate Inference: Sampling

# Sampling

- Sampling is a lot like repeated simulation.
  - Predicting the weather, basketball games, ...
- Basic idea
  - Draw N samples from a sampling distribution S.
  - Compute an approximate posterior probability.
  - Show this converges to the true probability P.
- Why sample?
  - Learning: get samples from a distribution you don't know
  - Inference: getting a sample is faster than computing the right answer (e.g., with variable elimination)

# Sampling

## ■ Sampling from given distribution

- Step 1: Get sample  $u$  from uniform distribution over  $[0, 1]$ 
  - E.g., `random()` in python
- Step 2: Convert this sample  $u$  into an outcome for the given distribution by having each outcome associated with a sub-interval of  $[0,1)$  with sub-interval size equal to probability of the outcome

## ■ Example

C	P(C)
red	0.6
green	0.1
blue	0.3

$0 \leq u < 0.6, \rightarrow C = \text{red}$   
 $0.6 \leq u < 0.7, \rightarrow C = \text{green}$   
 $0.7 \leq u < 1, \rightarrow C = \text{blue}$

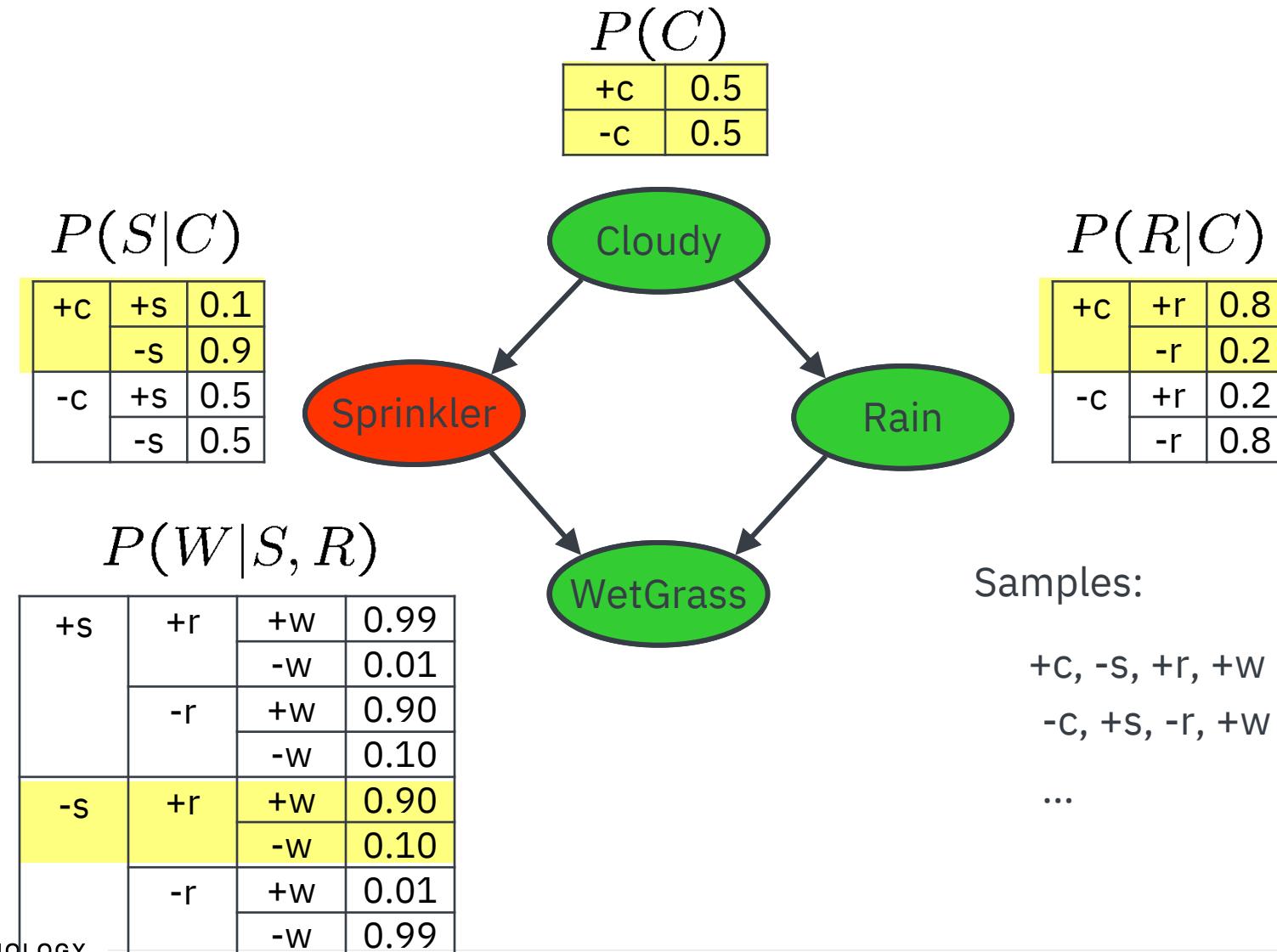
- If `random()` returns  $u = 0.83$ , then our sample is  $C = \text{blue}$
- E.g., after sampling 8 times:
  - 5 red, 2 blue, 1 green

# Sampling in Bayes' Nets

- Prior Sampling
- Rejection Sampling
- Likelihood Weighting
- Gibbs Sampling
- ...

# Prior Sampling

# Prior Sampling



# Prior Sampling

- For  $i=1, 2, \dots, n$ 
  - Sample  $x_i$  from  $P(X_i | \text{Parents}(X_i))$
- Return  $(x_1, x_2, \dots, x_n)$

# Prior Sampling

- This process generates samples with probability:

$$S_{PS}(x_1 \dots x_n) = \prod_{i=1}^n P(x_i | \text{Parents}(X_i)) = P(x_1 \dots x_n)$$

...i.e., the BN's joint probability

- Let the number of samples of an event be

$$N_{PS}(x_1 \dots x_n)$$

- Then

$$\begin{aligned}\lim_{N \rightarrow \infty} \hat{P}(x_1, \dots, x_n) &= \lim_{N \rightarrow \infty} N_{PS}(x_1, \dots, x_n)/N \\ &= S_{PS}(x_1, \dots, x_n) \\ &= P(x_1 \dots x_n)\end{aligned}$$

- I.e., the sampling procedure is **consistent**.

# Example

- We will get a bunch of samples from the BN:

+c, -s, +r, +w

+c, +s, +r, +w

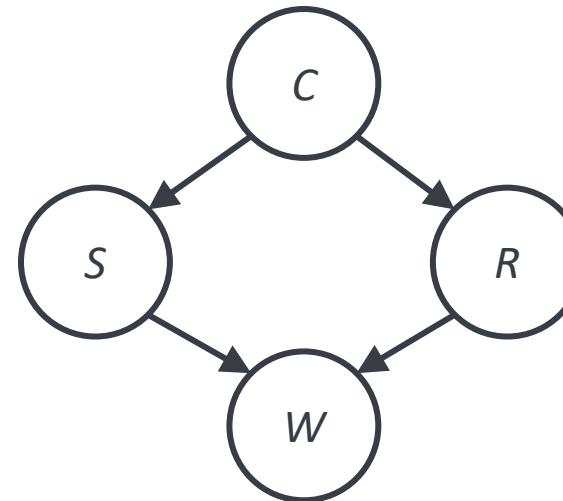
-c, +s, +r, -w

+c, -s, +r, +w

-c, -s, -r, +w

- If we want to know  $P(W)$

- We have counts  $\langle +w:4, -w:1 \rangle$
- Normalize to get  $P(W) = \langle +w:0.8, -w:0.2 \rangle$
- This will get closer to the true distribution with more samples.
- Can estimate anything else, too.
- What about  $P(C| +w)$ ?  $P(C| +r, +w)$ ?  $P(C| -r, -w)$ ?
- Fast: can use fewer samples if less time (what's the drawback?)





# Questions?

## Acknowledgement

Fahiem Bacchus, University of Toronto  
Dan Klein, UC Berkeley  
Kate Larson, University of Waterloo

