

# UNCERTAINTY

## AIMA2E CHAPTER 13

# Outline

- ◇ Uncertainty
- ◇ Probability
- ◇ Syntax and Semantics
- ◇ Inference
- ◇ Independence and Bayes' Rule

# Uncertainty

Let action  $A_t =$  leave for airport  $t$  minutes before flight

Will  $A_t$  get me there on time?

Problems:

- 1) partial observability (road state, other drivers' plans, etc.)
- 2) noisy sensors (KCBS traffic reports)
- 3) uncertainty in action outcomes (flat tire, etc.)
- 4) immense complexity of modelling and predicting traffic

Hence a purely logical approach either

1) risks falsehood: " $A_{25}$  will get me there on time"

or 2) leads to conclusions that are too weak for decision making:

" $A_{25}$  will get me there on time if there's no accident on the bridge and it doesn't rain and my tires remain intact etc etc."

( $A_{1440}$  might reasonably be said to get me there on time but I'd have to stay overnight in the airport ...)

# Methods for handling uncertainty

Default or nonmonotonic logic:

Assume my car does not have a flat tire

Assume  $A_{25}$  works unless contradicted by evidence

Issues: What assumptions are reasonable? How to handle contradiction?

Rules with fudge factors:

$A_{25} \mapsto_{0.3}$  get there on time

$Sprinkler \mapsto_{0.99} WetGrass$

$WetGrass \mapsto_{0.7} Rain$

Issues: Problems with combination, e.g., *Sprinkler causes Rain??*

Probability

Given the available evidence,

$A_{25}$  will get me there on time with probability 0.04

Mahaviracarya (9th C.), Cardano (1565) theory of gambling

(Fuzzy logic handles *degree of truth* NOT uncertainty e.g.,

$WetGrass$  is true to degree 0.2)

# Probability

Probabilistic assertions *summarize* effects of

**laziness**: failure to enumerate exceptions, qualifications, etc.

**ignorance**: lack of relevant facts, initial conditions, etc.

Subjective or Bayesian probability:

Probabilities relate propositions to one's own state of knowledge

$$\text{e.g., } P(A_{25} | \text{no reported accidents}) = 0.06$$

These are *not* claims of some **probabilistic tendency** in the current situation  
(but might be learned from past experience of similar situations)

Probabilities of propositions change with new evidence:

$$\text{e.g., } P(A_{25} | \text{no reported accidents, 5 a.m.}) = 0.15$$

(Analogous to logical entailment status  $KB \models \alpha$ , not truth.)

## Making decisions under uncertainty

Suppose I believe the following:

$$P(A_{25} \text{ gets me there on time} | \dots) = 0.04$$

$$P(A_{90} \text{ gets me there on time} | \dots) = 0.70$$

$$P(A_{120} \text{ gets me there on time} | \dots) = 0.95$$

$$P(A_{1440} \text{ gets me there on time} | \dots) = 0.9999$$

Which action to choose?

Depends on my **preferences** for missing flight vs. airport cuisine, etc.

**Utility theory** is used to represent and infer preferences

**Decision theory** = utility theory + probability theory

# Probability basics

Begin with a set  $\Omega$ —the *sample space*

e.g., 6 possible rolls of a die.

$\omega \in \Omega$  is a sample point/possible world/atomic event

A *probability space* or *probability model* is a sample space with an assignment  $P(\omega)$  for every  $\omega \in \Omega$  s.t.

$$0 \leq P(\omega) \leq 1$$

$$\sum_{\omega} P(\omega) = 1$$

e.g.,  $P(1) = P(2) = P(3) = P(4) = P(5) = P(6) = 1/6$ .

An *event*  $A$  is any subset of  $\Omega$

$$P(A) = \sum_{\{\omega \in A\}} P(\omega)$$

E.g.,  $P(\text{die roll} < 4) = 1/6 + 1/6 + 1/6 = 1/2$

## Random variables

A *random variable* is a function from sample points to some range, e.g., the reals or Booleans

e.g.,  $Odd(1) = true$ .

$P$  induces a *probability distribution* for any r.v.  $X$ :

$$P(X = x_i) = \sum_{\{\omega: X(\omega) = x_i\}} P(\omega)$$

e.g.,  $P(Odd = true) = 1/6 + 1/6 + 1/6 = 1/2$



# Propositions

Think of a proposition as the event (set of sample points) where the proposition is true

Given Boolean random variables  $A$  and  $B$ :

event  $a$  = set of sample points where  $A(\omega) = \text{true}$

event  $\neg a$  = set of sample points where  $A(\omega) = \text{false}$

event  $a \wedge b$  = points where  $A(\omega) = \text{true}$  and  $B(\omega) = \text{true}$

Often in AI applications, the sample points are *defined* by the values of a set of random variables, i.e., the sample space is the Cartesian product of the ranges of the variables

With Boolean variables, sample point = propositional logic model

e.g.,  $A = \text{true}$ ,  $B = \text{false}$ , or  $a \wedge \neg b$ .

Proposition = disjunction of atomic events in which it is true

e.g.,  $(a \vee b) \equiv (\neg a \wedge b) \vee (a \wedge \neg b) \vee (a \wedge b)$

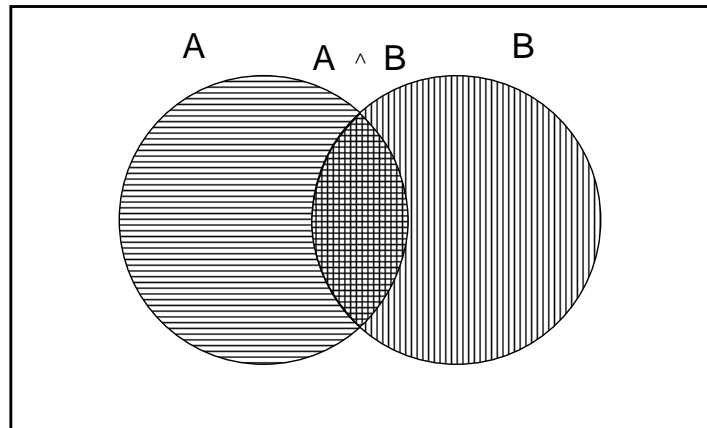
$\Rightarrow P(a \vee b) = P(\neg a \wedge b) + P(a \wedge \neg b) + P(a \wedge b)$

## Why use probability?

The definitions imply that certain logically related events must have related probabilities

$$\text{E.g., } P(a \vee b) = P(a) + P(b) - P(a \wedge b)$$

True



de Finetti (1931): an agent who bets according to probabilities that violate these axioms can be forced to bet so as to lose money regardless of outcome.

## Syntax for propositions

Propositional or Boolean random variables

e.g., *Cavity* (do I have a cavity?)

Discrete random variables (*finite* or *infinite*)

e.g., *Weather* is one of  $\langle \textit{sunny}, \textit{rain}, \textit{cloudy}, \textit{snow} \rangle$

*Weather = rain* is a proposition

Values must be exhaustive and mutually exclusive

Continuous random variables (*bounded* or *unbounded*)

e.g., *Temp* = 21.6; also allow, e.g., *Temp* < 22.0.

Arbitrary Boolean combinations of basic propositions

## Prior probability

Prior or unconditional probabilities of propositions

e.g.,  $P(\text{Cavity} = \text{true}) = 0.1$  and  $P(\text{Weather} = \text{sunny}) = 0.72$   
correspond to belief prior to arrival of any (new) evidence

Probability distribution gives values for all possible assignments:

$$\mathbf{P}(\text{Weather}) = \langle 0.72, 0.1, 0.08, 0.1 \rangle \text{ (normalized, i.e., sums to 1)}$$

Joint probability distribution for a set of r.v.s gives the probability of every atomic event on those r.v.s (i.e., every sample point)

$\mathbf{P}(\text{Weather}, \text{Cavity}) =$  a  $4 \times 2$  matrix of values:

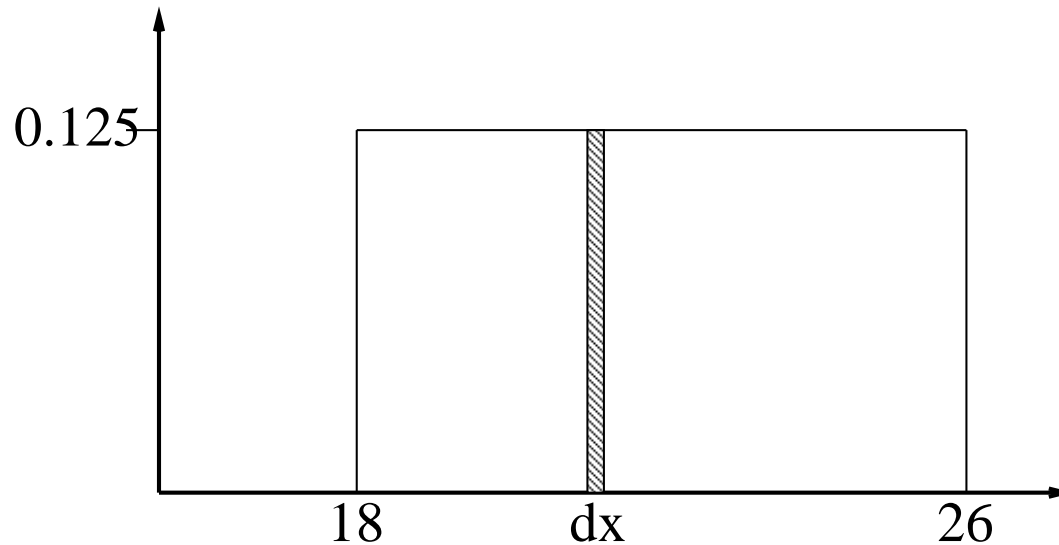
<i>Weather =</i>	<i>sunny</i>	<i>rain</i>	<i>cloudy</i>	<i>snow</i>
<i>Cavity = true</i>	0.144	0.02	0.016	0.02
<i>Cavity = false</i>	0.576	0.08	0.064	0.08

*Every question about a domain can be answered by the joint distribution because every event is a sum of sample points*

# Probability for continuous variables

Express distribution as a parameterized function of value:

$$P(X = x) = U[18, 26](x) = \text{uniform density between 18 and 26}$$



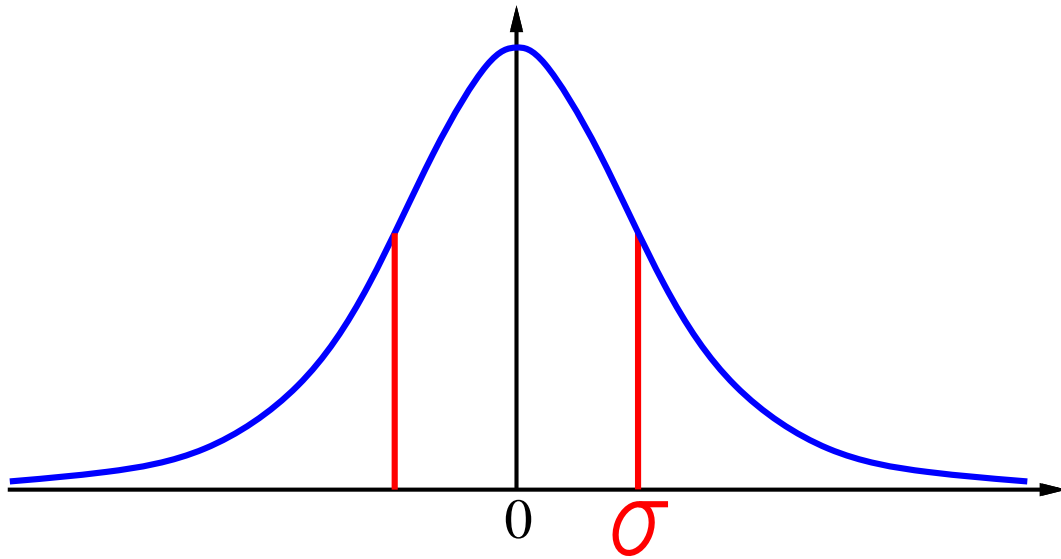
Here  $P$  is a *density*; integrates to 1.

$P(X = 20.5) = 0.125$  really means

$$\lim_{dx \rightarrow 0} P(20.5 \leq X \leq 20.5 + dx) / dx = 0.125$$

# Gaussian density

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$$



# Conditional probability

Conditional or posterior probabilities

e.g.,  $P(\text{cavity}|\text{toothache}) = 0.8$

i.e., given that *toothache* is all I know

NOT “if *toothache* then 80% chance of *cavity*”

(Notation for conditional distributions:

$\mathbf{P}(\text{Cavity}|\text{Toothache}) = 2\text{-element vector of 2-element vectors}$ )

If we know more, e.g., *cavity* is also given, then we have

$P(\text{cavity}|\text{toothache}, \text{cavity}) = 1$

Note: the less specific belief *remains valid* after more evidence arrives, but is not always *useful*

New evidence may be irrelevant, allowing simplification, e.g.,

$P(\text{cavity}|\text{toothache}, \text{49ersWin}) = P(\text{cavity}|\text{toothache}) = 0.8$

This kind of inference, sanctioned by domain knowledge, is crucial

# Conditional probability

Definition of conditional probability:

$$P(a|b) = \frac{P(a \wedge b)}{P(b)} \text{ if } P(b) \neq 0$$

**Product rule** gives an alternative formulation:

$$P(a \wedge b) = P(a|b)P(b) = P(b|a)P(a)$$

A general version holds for whole distributions, e.g.,

$$\mathbf{P}(\textit{Weather}, \textit{Cavity}) = \mathbf{P}(\textit{Weather}|\textit{Cavity})\mathbf{P}(\textit{Cavity})$$

(View as a  $4 \times 2$  set of equations, *not* matrix mult.)

**Chain rule** is derived by successive application of product rule:

$$\begin{aligned} \mathbf{P}(X_1, \dots, X_n) &= \mathbf{P}(X_1, \dots, X_{n-1}) \mathbf{P}(X_n|X_1, \dots, X_{n-1}) \\ &= \mathbf{P}(X_1, \dots, X_{n-2}) \mathbf{P}(X_{n-1}|X_1, \dots, X_{n-2}) \mathbf{P}(X_n|X_1, \dots, X_{n-1}) \\ &= \dots \\ &= \prod_{i=1}^n \mathbf{P}(X_i|X_1, \dots, X_{i-1}) \end{aligned}$$



## Inference by enumeration

Start with the joint distribution:

	<i>toothache</i>		$\neg$ <i>toothache</i>	
	<i>catch</i>	$\neg$ <i>catch</i>	<i>catch</i>	$\neg$ <i>catch</i>
<i>cavity</i>	<b>.108</b>	<b>.012</b>	<b>.072</b>	<b>.008</b>
$\neg$ <i>cavity</i>	<b>.016</b>	<b>.064</b>	<b>.144</b>	<b>.576</b>

For any proposition  $\phi$ , sum the atomic events where it is true:

$$P(\phi) = \sum_{\omega:\omega\models\phi} P(\omega)$$

## Inference by enumeration

Start with the joint distribution:

	<i>toothache</i>		$\neg$ <i>toothache</i>	
	<i>catch</i>	$\neg$ <i>catch</i>	<i>catch</i>	$\neg$ <i>catch</i>
<i>cavity</i>	<b>.108</b>	<b>.012</b>	<b>.072</b>	<b>.008</b>
$\neg$ <i>cavity</i>	<b>.016</b>	<b>.064</b>	<b>.144</b>	<b>.576</b>

For any proposition  $\phi$ , sum the atomic events where it is true:

$$P(\phi) = \sum_{\omega:\omega\models\phi} P(\omega)$$

$$P(\textit{toothache}) = 0.108 + 0.012 + 0.016 + 0.064 = 0.2$$

## Inference by enumeration

Start with the joint distribution:

	<i>toothache</i>		$\neg$ <i>toothache</i>	
	<i>catch</i>	$\neg$ <i>catch</i>	<i>catch</i>	$\neg$ <i>catch</i>
<i>cavity</i>	<b>.108</b>	<b>.012</b>	<b>.072</b>	<b>.008</b>
$\neg$ <i>cavity</i>	<b>.016</b>	<b>.064</b>	<b>.144</b>	<b>.576</b>

For any proposition  $\phi$ , sum the atomic events where it is true:

$$P(\phi) = \sum_{\omega:\omega\models\phi} P(\omega)$$

$$P(\text{cavity} \vee \text{toothache}) = 0.108 + 0.012 + 0.072 + 0.008 + 0.016 + 0.064 = 0.28$$

## Inference by enumeration

Start with the joint distribution:

	<i>toothache</i>		$\neg$ <i>toothache</i>	
	<i>catch</i>	$\neg$ <i>catch</i>	<i>catch</i>	$\neg$ <i>catch</i>
<i>cavity</i>	<b>.108</b>	<b>.012</b>	<b>.072</b>	<b>.008</b>
$\neg$ <i>cavity</i>	<b>.016</b>	<b>.064</b>	<b>.144</b>	<b>.576</b>

Can also compute conditional probabilities:

$$\begin{aligned} P(\neg \text{cavity} | \text{toothache}) &= \frac{P(\neg \text{cavity} \wedge \text{toothache})}{P(\text{toothache})} \\ &= \frac{0.016 + 0.064}{0.108 + 0.012 + 0.016 + 0.064} = 0.4 \end{aligned}$$

## Normalization

	<i>toothache</i>		$\neg$ <i>toothache</i>	
	<i>catch</i>	$\neg$ <i>catch</i>	<i>catch</i>	$\neg$ <i>catch</i>
<i>cavity</i>	.108	.012	.072	.008
$\neg$ <i>cavity</i>	.016	.064	.144	.576

**Cavity** is a random variable!

$P(\text{cavity})$  is the same as

$P(\text{Cavity}=\text{true})$

$P(\sim\text{cavity})$  is the same as

$P(\text{Cavity}=\text{false})$

$P(\text{Cavity})$  is the distribution over all values of Cavity, namely the pair

$\langle P(\text{cavity}), P(\sim\text{cavity}) \rangle$

Denominator can be viewed as a *normalization constant*  $\alpha$

$$\begin{aligned}
 P(\text{Cavity}|\text{toothache}) &= \alpha P(\text{Cavity}, \text{toothache}) \\
 &= \alpha [P(\text{Cavity}, \text{toothache}, \text{catch}) + P(\text{Cavity}, \text{toothache}, \neg\text{catch})] \\
 &= \alpha [\langle 0.108, 0.016 \rangle + \langle 0.012, 0.064 \rangle] \\
 &= \alpha \langle 0.12, 0.08 \rangle = \langle 0.6, 0.4 \rangle
 \end{aligned}$$

General idea: compute distribution on query variable  
by fixing **evidence variables** and summing over **hidden variables**

## Inference by enumeration, contd.

Typically, we are interested in

the posterior joint distribution of the **query variables**  $\mathbf{Y}$   
given specific values  $\mathbf{e}$  for the **evidence variables**  $\mathbf{E}$

Let the **hidden variables** be  $\mathbf{H} = \mathbf{X} - \mathbf{Y} - \mathbf{E}$

Then the required summation of joint entries is done by summing out the hidden variables:

$$\mathbf{P}(\mathbf{Y}|\mathbf{E} = \mathbf{e}) = \alpha \mathbf{P}(\mathbf{Y}, \mathbf{E} = \mathbf{e}) = \alpha \sum_{\mathbf{h}} \mathbf{P}(\mathbf{Y}, \mathbf{E} = \mathbf{e}, \mathbf{H} = \mathbf{h})$$

The terms in the summation are joint entries because  $\mathbf{Y}$ ,  $\mathbf{E}$ , and  $\mathbf{H}$  together exhaust the set of random variables

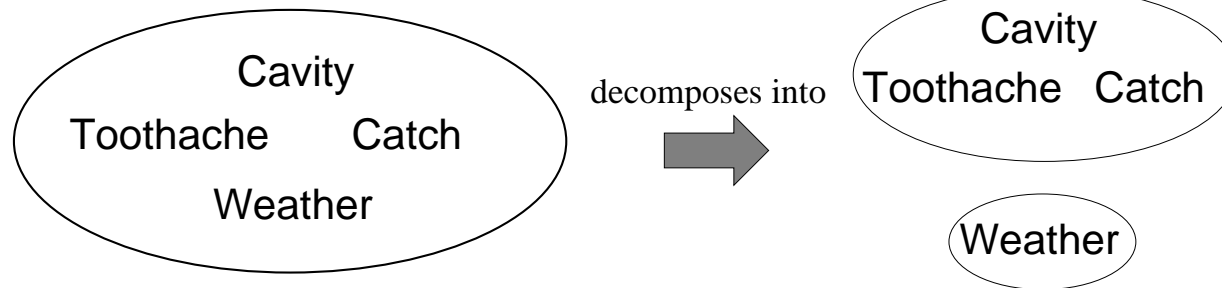
Obvious problems:

- 1) Worst-case time complexity  $O(d^n)$  where  $d$  is the largest arity
- 2) Space complexity  $O(d^n)$  to store the joint distribution
- 3) How to find the numbers for  $O(d^n)$  entries???

# Independence

$A$  and  $B$  are independent iff

$$\mathbf{P}(A|B) = \mathbf{P}(A) \quad \text{or} \quad \mathbf{P}(B|A) = \mathbf{P}(B) \quad \text{or} \quad \mathbf{P}(A, B) = \mathbf{P}(A)\mathbf{P}(B)$$



$$\begin{aligned} &\mathbf{P}(\textit{Toothache}, \textit{Catch}, \textit{Cavity}, \textit{Weather}) \\ &= \mathbf{P}(\textit{Toothache}, \textit{Catch}, \textit{Cavity})\mathbf{P}(\textit{Weather}) \end{aligned}$$

32 entries reduced to 12; for  $n$  independent biased coins,  $2^n \rightarrow n$

Absolute independence powerful but rare

Dentistry is a large field with hundreds of variables, none of which are independent. What to do?

## Conditional independence

$\mathbf{P}(\textit{Toothache}, \textit{Cavity}, \textit{Catch})$  has  $2^3 - 1 = 7$  independent entries

If I have a cavity, the probability that the probe catches in it doesn't depend on whether I have a toothache:

$$(1) P(\textit{catch}|\textit{toothache}, \textit{cavity}) = P(\textit{catch}|\textit{cavity})$$

The same independence holds if I haven't got a cavity:

$$(2) P(\textit{catch}|\textit{toothache}, \neg\textit{cavity}) = P(\textit{catch}|\neg\textit{cavity})$$

*Catch* is **conditionally independent** of *Toothache* given *Cavity*:

$$\mathbf{P}(\textit{Catch}|\textit{Toothache}, \textit{Cavity}) = \mathbf{P}(\textit{Catch}|\textit{Cavity})$$

Equivalent statements:

$$\mathbf{P}(\textit{Toothache}|\textit{Catch}, \textit{Cavity}) = \mathbf{P}(\textit{Toothache}|\textit{Cavity})$$

$$\mathbf{P}(\textit{Toothache}, \textit{Catch}|\textit{Cavity}) = \mathbf{P}(\textit{Toothache}|\textit{Cavity})\mathbf{P}(\textit{Catch}|\textit{Cavity})$$



## Conditional independence contd.

Write out full joint distribution using chain rule:

$$\begin{aligned} & \mathbf{P}(Toothache, Catch, Cavity) \\ &= \mathbf{P}(Toothache|Catch, Cavity)\mathbf{P}(Catch, Cavity) \\ &= \mathbf{P}(Toothache|Catch, Cavity)\mathbf{P}(Catch|Cavity)\mathbf{P}(Cavity) \\ &= \mathbf{P}(Toothache|Cavity)\mathbf{P}(Catch|Cavity)\mathbf{P}(Cavity) \end{aligned}$$

I.e.,  $2 + 2 + 1 = 5$  independent numbers (equations 1 and 2 remove 2)

*In most cases, the use of conditional independence reduces the size of the representation of the joint distribution from exponential in  $n$  to linear in  $n$ .*

*Conditional independence is our most basic and robust form of knowledge about uncertain environments.*

## Bayes' Rule

Product rule  $P(a \wedge b) = P(a|b)P(b) = P(b|a)P(a)$

$$\Rightarrow \text{Bayes' rule } P(a|b) = \frac{P(b|a)P(a)}{P(b)}$$

or in distribution form

$$\mathbf{P}(Y|X) = \frac{\mathbf{P}(X|Y)\mathbf{P}(Y)}{\mathbf{P}(X)} = \alpha\mathbf{P}(X|Y)\mathbf{P}(Y)$$

Useful for assessing **diagnostic** probability from **causal** probability:

$$P(Cause|Effect) = \frac{P(Effect|Cause)P(Cause)}{P(Effect)}$$

E.g., let  $M$  be meningitis,  $S$  be stiff neck:

$$P(m|s) = \frac{P(s|m)P(m)}{P(s)} = \frac{0.8 \times 0.0001}{0.1} = 0.0008$$

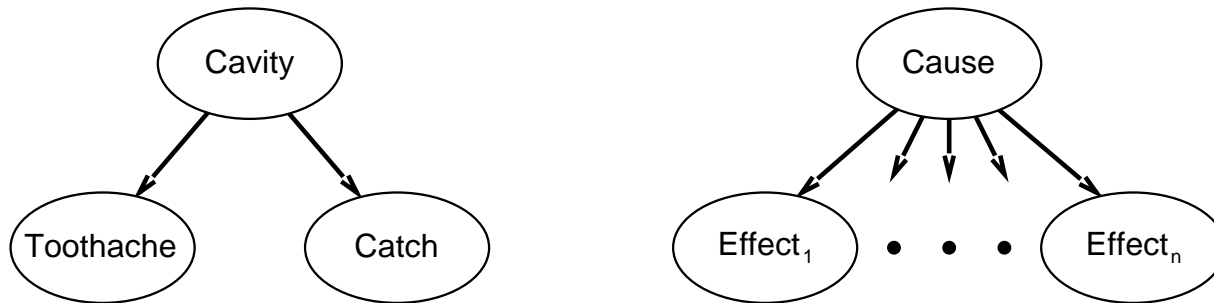
Note: posterior probability of meningitis still very small!

# Bayes' Rule and conditional independence

$$\begin{aligned} & \mathbf{P}(Cavity|toothache \wedge catch) \\ &= \alpha \mathbf{P}(toothache \wedge catch|Cavity)\mathbf{P}(Cavity) \\ &= \alpha \mathbf{P}(toothache|Cavity)\mathbf{P}(catch|Cavity)\mathbf{P}(Cavity) \end{aligned}$$

This is an example of a *naive Bayes* model:

$$\mathbf{P}(Cause, Effect_1, \dots, Effect_n) = \mathbf{P}(Cause) \prod_i \mathbf{P}(Effect_i|Cause)$$



Total number of parameters is *linear* in  $n$

## Summary

Probability is a rigorous formalism for uncertain knowledge

Joint probability distribution specifies probability of every atomic event

Queries can be answered by summing over atomic events

For nontrivial domains, we must find a way to reduce the joint size

Independence and conditional independence provide the tools

# BAYESIAN NETWORKS

AIMA2E CHAPTER 14.1–3

# Outline

- ◇ Syntax
- ◇ Semantics
- ◇ Parameterized distributions

# Bayesian networks

A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions

Syntax:

- a set of nodes, one per variable

- a directed, acyclic graph (link  $\approx$  “directly influences”)

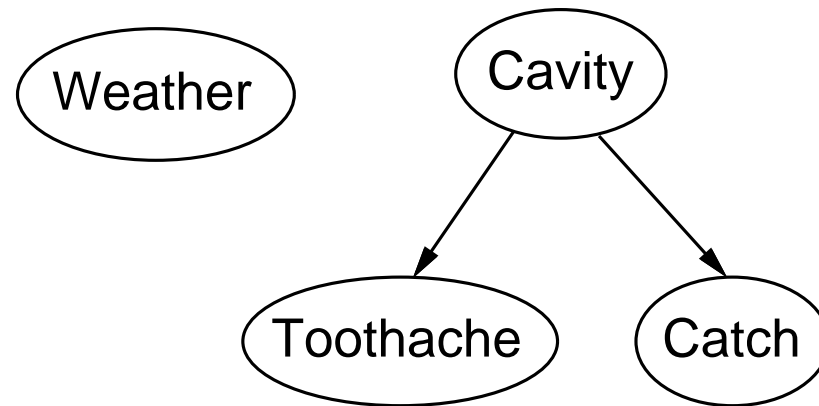
- a conditional distribution for each node given its parents:

$$P(X_i | Parents(X_i))$$

In the simplest case, conditional distribution represented as a **conditional probability table** (CPT) giving the distribution over  $X_i$  for each combination of parent values

## Example

Topology of network encodes conditional independence assertions:



*Weather* is independent of the other variables

*Toothache* and *Catch* are conditionally independent given *Cavity*



## Example

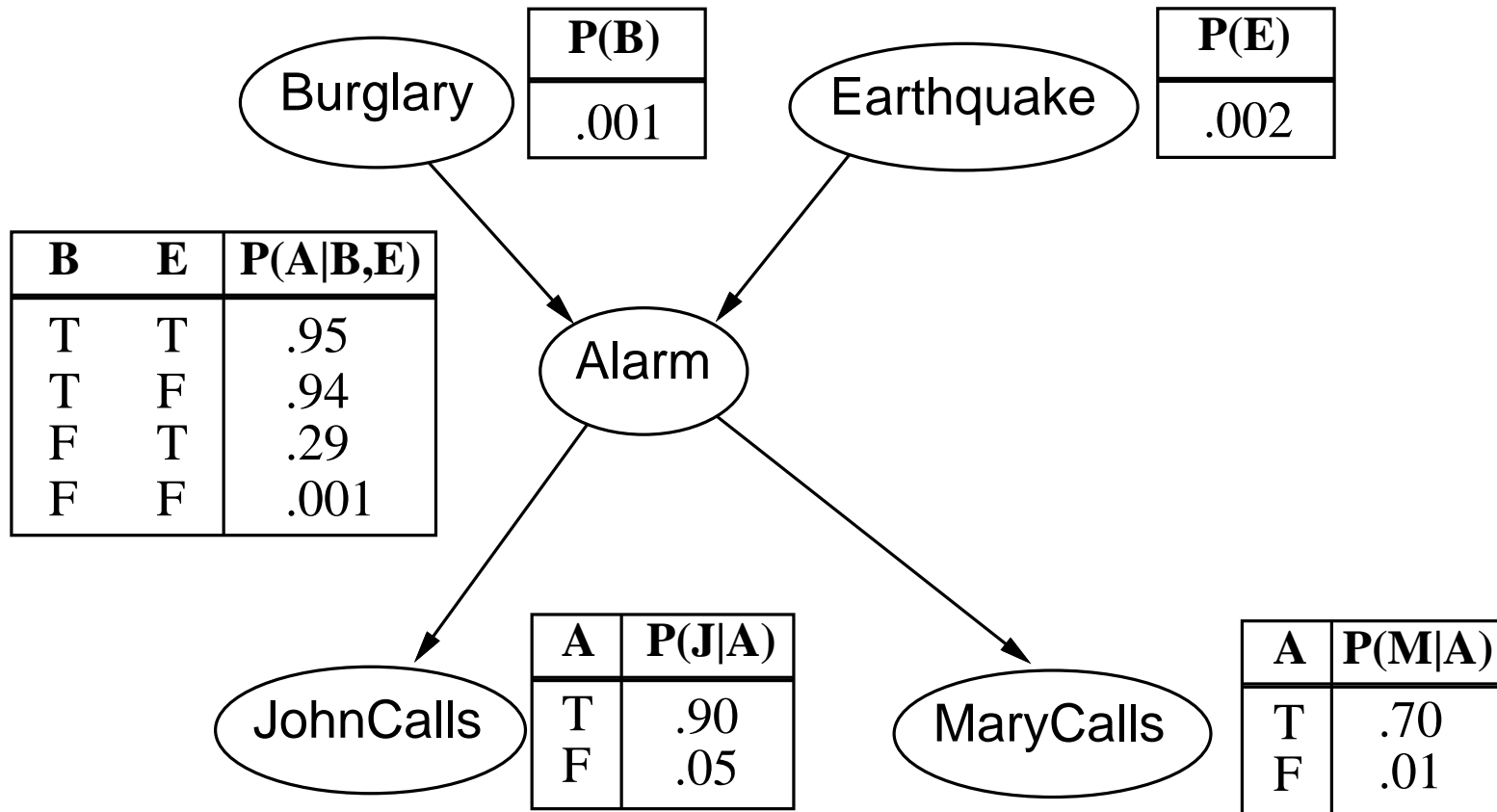
I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

Variables: *Burglar*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*

Network topology reflects "causal" knowledge:

- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call

## Example contd.



## Compactness

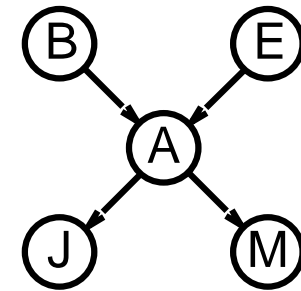
A CPT for Boolean  $X_i$  with  $k$  Boolean parents has  $2^k$  rows for the combinations of parent values

Each row requires one number  $p$  for  $X_i = \text{true}$  (the number for  $X_i = \text{false}$  is just  $1 - p$ )

If each variable has no more than  $k$  parents, the complete network requires  $O(n \cdot 2^k)$  numbers

I.e., grows linearly with  $n$ , vs.  $O(2^n)$  for the full joint distribution

For burglary net,  $1 + 1 + 4 + 2 + 2 = 10$  numbers (vs.  $2^5 - 1 = 31$ )



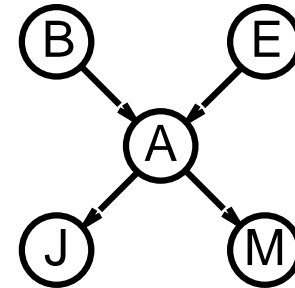
## Global semantics

Global semantics defines the full joint distribution as the product of the local conditional distributions:

$$\mathbf{P}(X_1, \dots, X_n) = \prod_{i=1}^n \mathbf{P}(X_i | \text{Parents}(X_i))$$

e.g.,  $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

=



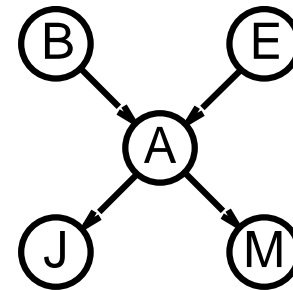
## Global semantics

“Global” semantics defines the full joint distribution as the product of the local conditional distributions:

$$\mathbf{P}(X_1, \dots, X_n) = \prod_{i=1}^n \mathbf{P}(X_i | \text{Parents}(X_i))$$

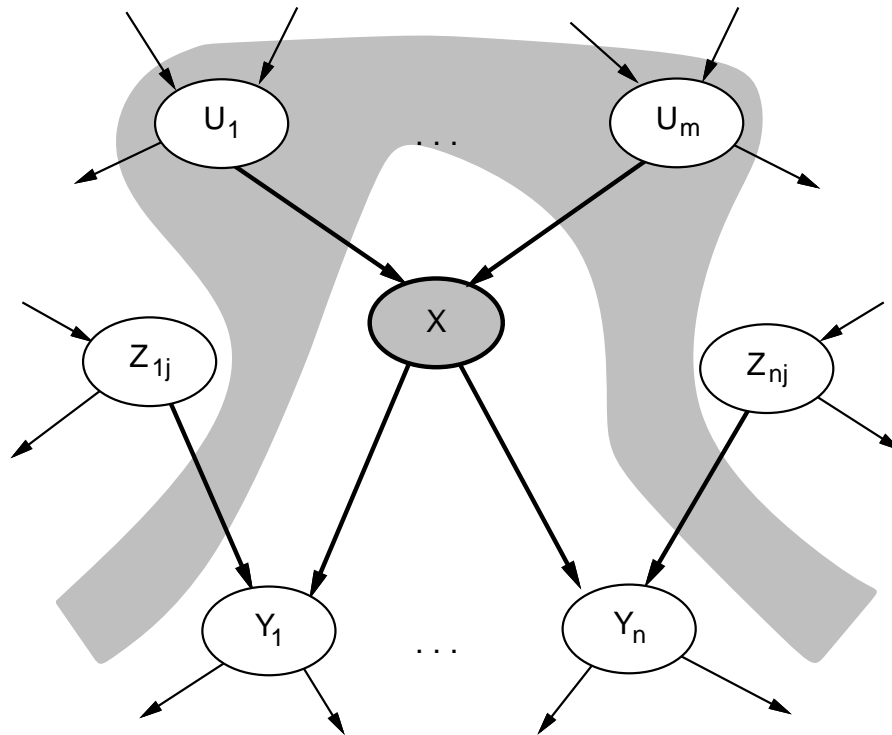
e.g.,  $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

$$= P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e)$$



# Local semantics

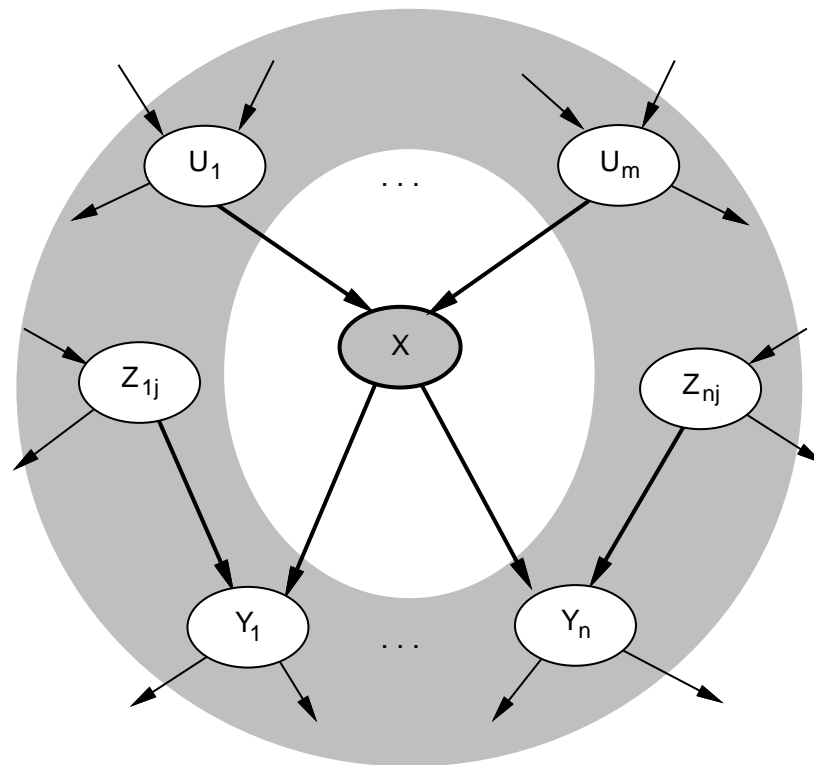
Local semantics: each node is conditionally independent of its nondescendants given its parents



Theorem: Local semantics  $\Leftrightarrow$  global semantics

# Markov blanket

Each node is conditionally independent of all others given its  
**Markov blanket**: parents + children + children's parents



# Constructing Bayesian networks

Need a method such that a series of locally testable assertions of conditional independence guarantees the required global semantics

1. Choose an ordering of variables  $X_1, \dots, X_n$
2. For  $i = 1$  to  $n$   
add  $X_i$  to the network  
select parents from  $X_1, \dots, X_{i-1}$  such that
$$\mathbf{P}(X_i | \text{Parents}(X_i)) = \mathbf{P}(X_i | X_1, \dots, X_{i-1})$$

This choice of parents guarantees the global semantics:

$$\begin{aligned} \mathbf{P}(X_1, \dots, X_n) &= \prod_{i=1}^n \mathbf{P}(X_i | X_1, \dots, X_{i-1}) \quad (\text{chain rule}) \\ &= \prod_{i=1}^n \mathbf{P}(X_i | \text{Parents}(X_i)) \quad (\text{by construction}) \end{aligned}$$



## Example

Suppose we choose the ordering  $M, J, A, B, E$

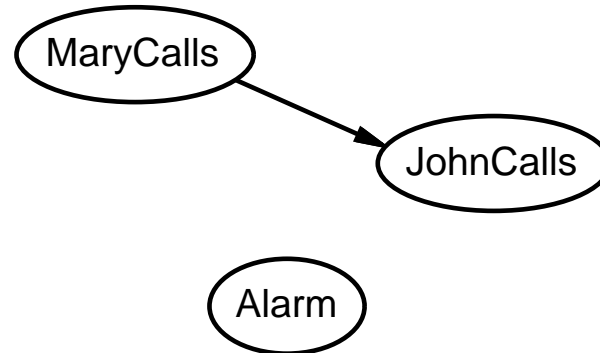
MaryCalls

JohnCalls

$$P(J|M) = P(J)?$$

## Example

Suppose we choose the ordering  $M, J, A, B, E$

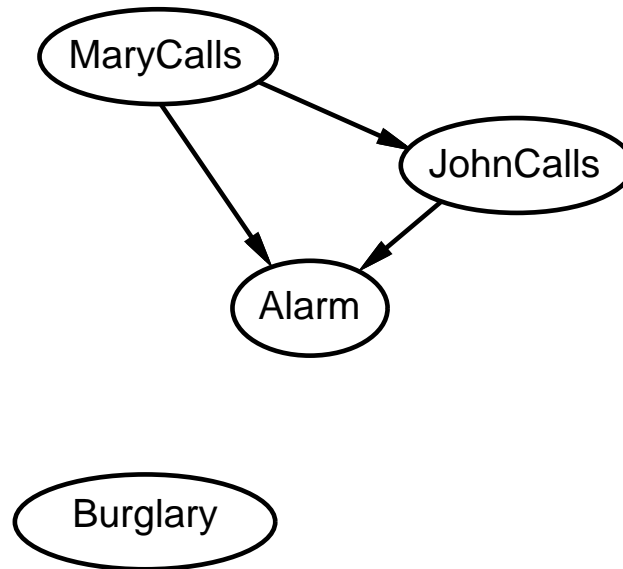


$P(J|M) = P(J)$ ? No

$P(A|J, M) = P(A|J)$ ?  $P(A|J, M) = P(A)$ ?

## Example

Suppose we choose the ordering  $M, J, A, B, E$



$P(J|M) = P(J)$ ? No

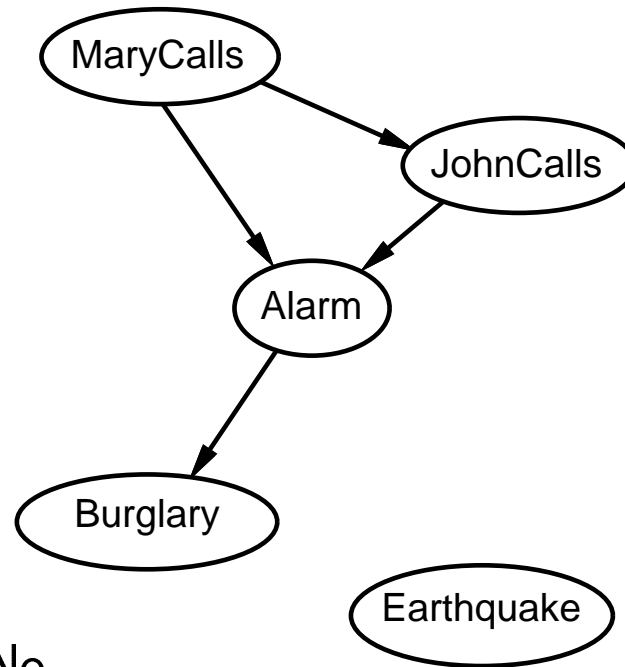
$P(A|J, M) = P(A|J)$ ?  $P(A|J, M) = P(A)$ ? No

$P(B|A, J, M) = P(B|A)$ ?

$P(B|A, J, M) = P(B)$ ?

## Example

Suppose we choose the ordering  $M, J, A, B, E$



$P(J|M) = P(J)$ ? No

$P(A|J, M) = P(A|J)$ ?  $P(A|J, M) = P(A)$ ? No

$P(B|A, J, M) = P(B|A)$ ? Yes

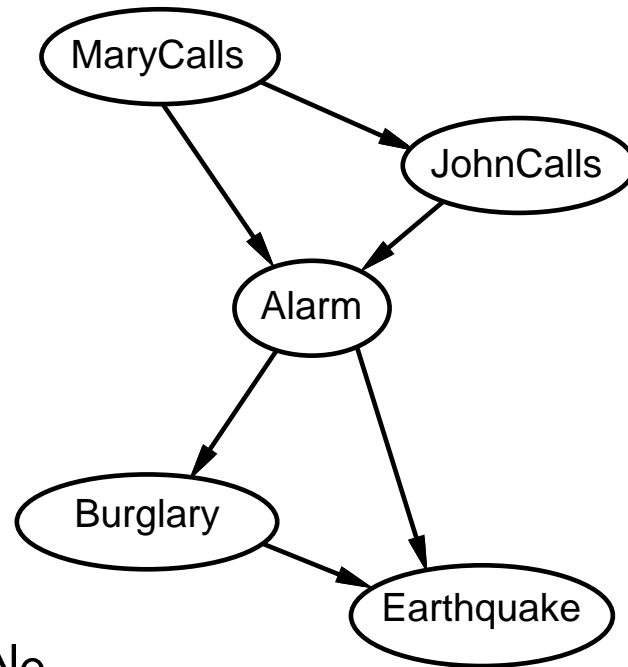
$P(B|A, J, M) = P(B)$ ? No

$P(E|B, A, J, M) = P(E|A)$ ?

$P(E|B, A, J, M) = P(E|A, B)$ ?

## Example

Suppose we choose the ordering  $M, J, A, B, E$



$P(J|M) = P(J)$ ? No

$P(A|J, M) = P(A|J)$ ?  $P(A|J, M) = P(A)$ ? No

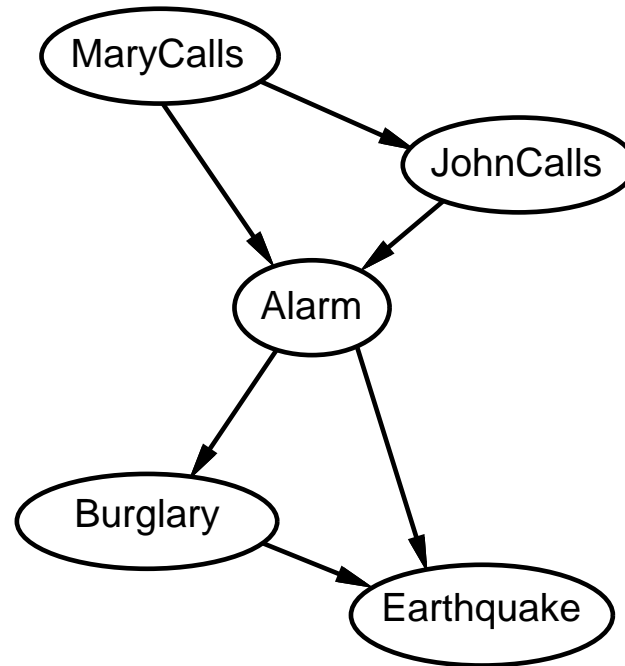
$P(B|A, J, M) = P(B|A)$ ? Yes

$P(B|A, J, M) = P(B)$ ? No

$P(E|B, A, J, M) = P(E|A)$ ? No

$P(E|B, A, J, M) = P(E|A, B)$ ? Yes

## Example contd.



Deciding conditional independence is hard in noncausal directions

(Causal models and conditional independence seem hardwired for humans!)

Assessing conditional probabilities is hard in noncausal directions

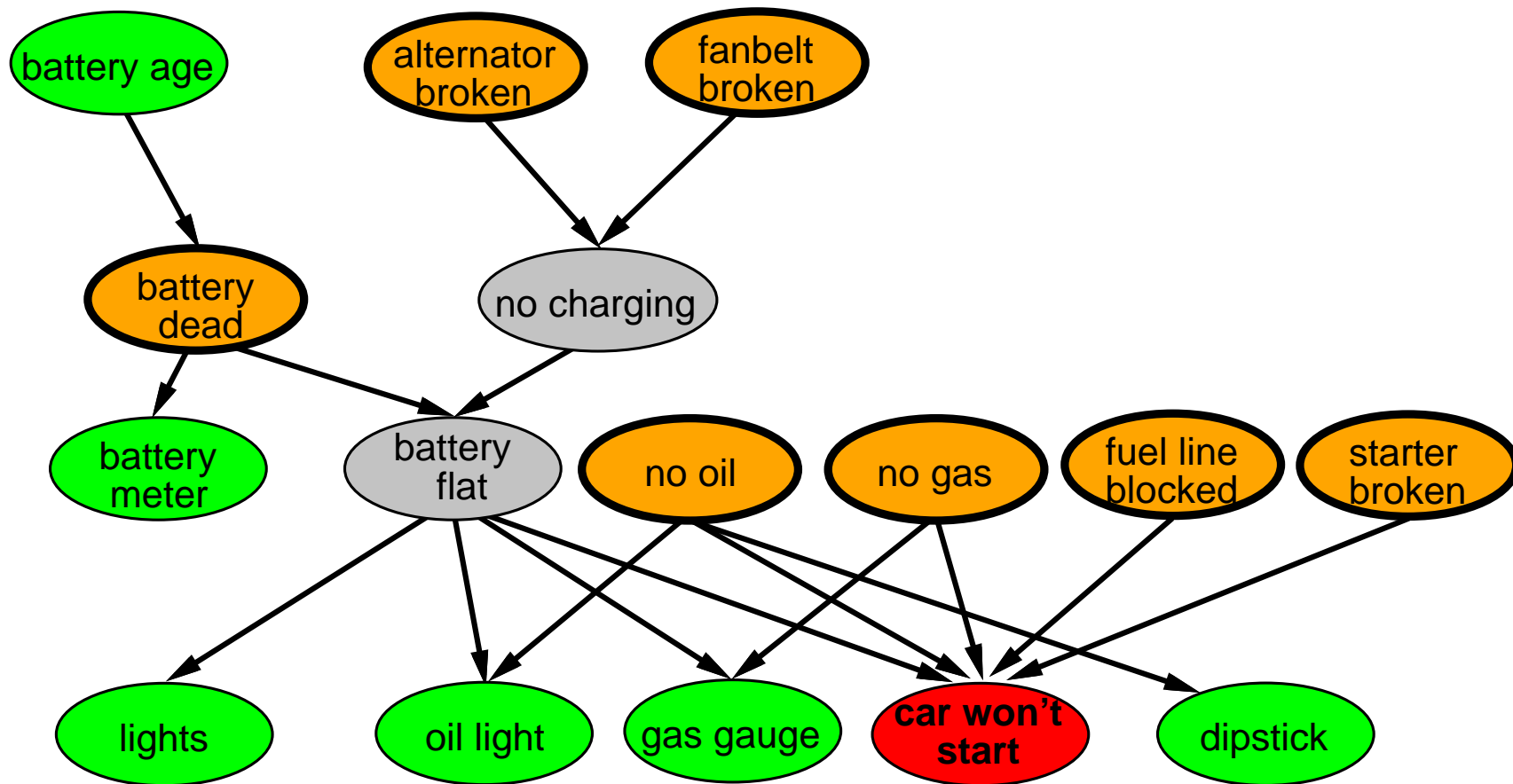
Network is less compact:  $1 + 2 + 4 + 2 + 4 = 13$  numbers needed

## Example: Car diagnosis

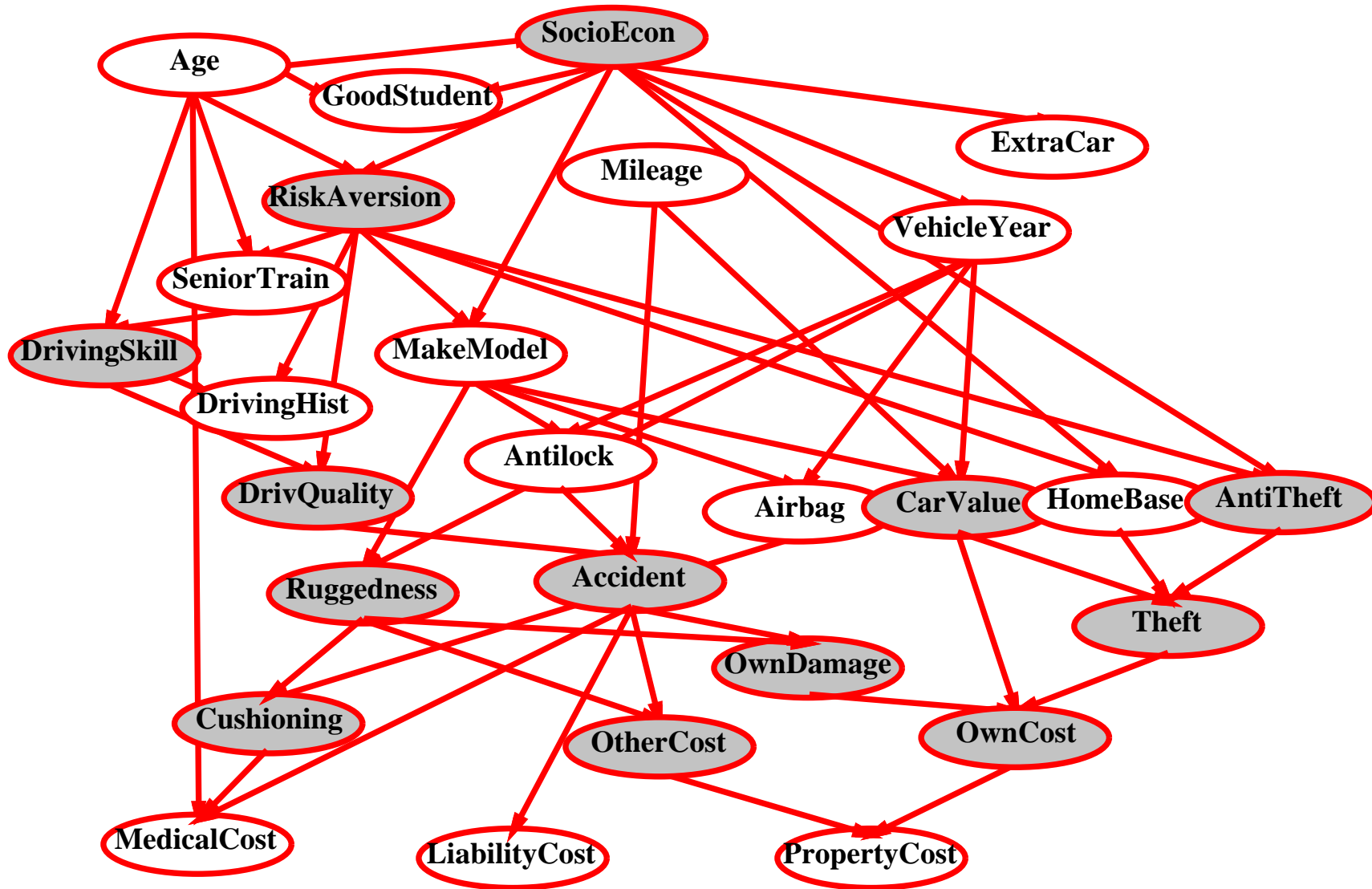
Initial evidence: car won't start

Testable variables (green), "broken, so fix it" variables (orange)

Hidden variables (gray) ensure sparse structure, reduce parameters



# Example: Car insurance





## Compact conditional distributions

CPT grows exponentially with no. of parents

CPT becomes infinite with continuous-valued parent or child

Solution: **canonical** distributions that are defined compactly

**Deterministic** nodes are the simplest case:

$$X = f(\text{Parents}(X)) \text{ for some function } f$$

E.g., Boolean functions

$$\text{NorthAmerican} \Leftrightarrow \text{Canadian} \vee \text{US} \vee \text{Mexican}$$

E.g., numerical relationships among continuous variables

$$\frac{\partial \text{Level}}{\partial t} = \text{inflow} + \text{precipitation} - \text{outflow} - \text{evaporation}$$

## Compact conditional distributions contd.

Noisy-OR distributions model multiple noninteracting causes

- 1) Parents  $U_1 \dots U_k$  include all causes (can add **leak node**)
- 2) Independent failure probability  $q_i$  for each cause alone

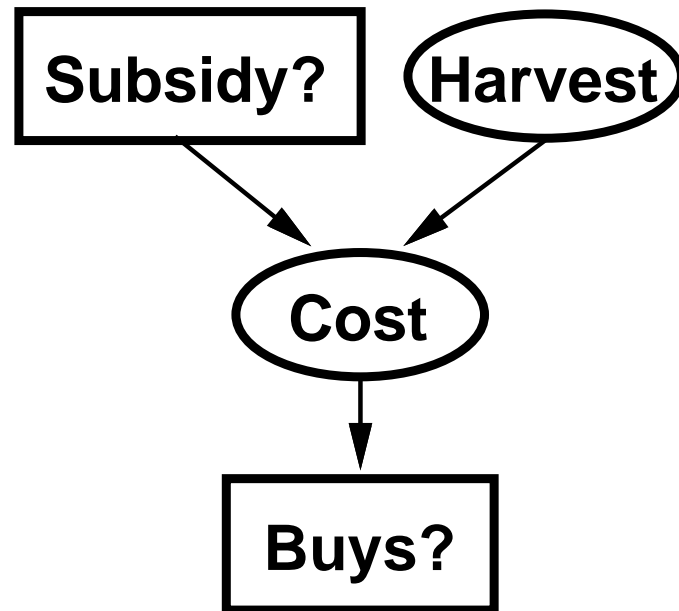
$$\Rightarrow P(X|U_1 \dots U_j, \neg U_{j+1} \dots \neg U_k) = 1 - \prod_{i=1}^j q_i$$

<i>Cold</i>	<i>Flu</i>	<i>Malaria</i>	$P(\text{Fever})$	$P(\neg \text{Fever})$
F	F	F	<b>0.0</b>	1.0
F	F	T	0.9	<b>0.1</b>
F	T	F	0.8	<b>0.2</b>
F	T	T	0.98	$0.02 = 0.2 \times 0.1$
T	F	F	0.4	<b>0.6</b>
T	F	T	0.94	$0.06 = 0.6 \times 0.1$
T	T	F	0.88	$0.12 = 0.6 \times 0.2$
T	T	T	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

Number of parameters **linear** in number of parents

## Hybrid (discrete+continuous) networks

Discrete (*Subsidy?* and *Buys?*); continuous (*Harvest* and *Cost*)



Option 1: discretization—possibly large errors, large CPTs

Option 2: finitely parameterized canonical families

- 1) Continuous variable, discrete+continuous parents (e.g., *Cost*)
- 2) Discrete variable, continuous parents (e.g., *Buys?*)

## Continuous child variables

Need one **conditional density** function for child variable given continuous parents, for each possible assignment to discrete parents

Most common is the **linear Gaussian** model, e.g.,:

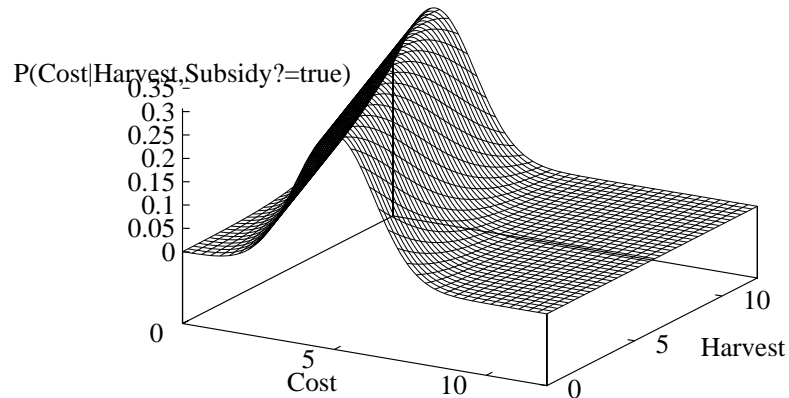
$$\begin{aligned} P(\text{Cost} = c | \text{Harvest} = h, \text{Subsidy?} = \text{true}) \\ &= N(a_t h + b_t, \sigma_t)(c) \\ &= \frac{1}{\sigma_t \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{c - (a_t h + b_t)}{\sigma_t}\right)^2\right) \end{aligned}$$

Mean *Cost* varies linearly with *Harvest*, variance is fixed

Linear variation is unreasonable over the full range

but works OK if the **likely** range of *Harvest* is narrow

# Continuous child variables



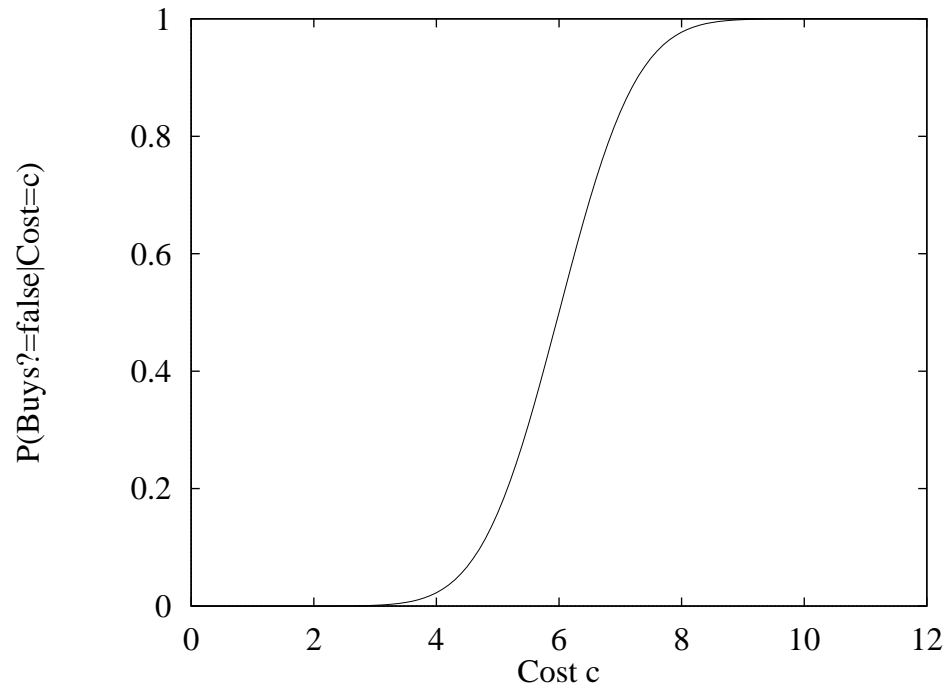
All-continuous network with LG distributions

⇒ full joint distribution is a multivariate Gaussian

Discrete+continuous LG network is a **conditional Gaussian** network i.e., a multivariate Gaussian over all continuous variables for each combination of discrete variable values

## Discrete variable w/ continuous parents

Probability of *Buys?* given *Cost* should be a “soft” threshold:



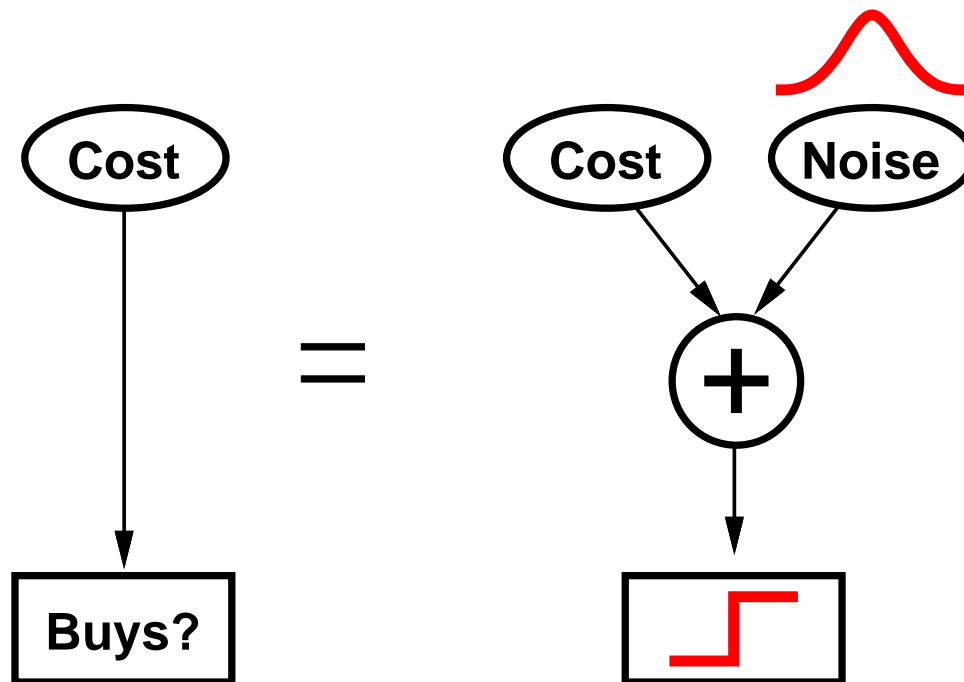
Probit distribution uses integral of Gaussian:

$$\Phi(x) = \int_{-\infty}^x N(0, 1)(x)dx$$

$$P(\text{Buys?} = \text{true} \mid \text{Cost} = c) = \Phi((-c + \mu)/\sigma)$$

# Why the probit?

1. It's sort of the right shape
2. Can view as hard threshold whose location is subject to noise



## Summary

Bayes nets provide a natural representation for (causally induced) conditional independence

Topology + CPTs = compact representation of joint distribution

Generally easy for (non)experts to construct

Canonical distributions (e.g., noisy-OR) = compact representation of CPTs

Continuous variables  $\Rightarrow$  parameterized distributions (e.g., linear Gaussian)



# INFERENCE IN BAYESIAN NETWORKS

AIMA2E CHAPTER 14.4–5

## Outline

- ◇ Exact inference by enumeration
- ◇ Exact inference by variable elimination
- ◇ Approximate inference by stochastic simulation
- ◇ Approximate inference by Markov chain Monte Carlo

## Inference tasks

Simple queries: compute posterior marginal  $\mathbf{P}(X_i|\mathbf{E} = \mathbf{e})$

e.g.,  $P(\text{NoGas}|\text{Gauge} = \text{empty}, \text{Lights} = \text{on}, \text{Starts} = \text{false})$

Conjunctive queries:  $\mathbf{P}(X_i, X_j|\mathbf{E} = \mathbf{e}) = \mathbf{P}(X_i|\mathbf{E} = \mathbf{e})\mathbf{P}(X_j|X_i, \mathbf{E} = \mathbf{e})$

Optimal decisions: decision networks include utility information;  
probabilistic inference required for  $P(\text{outcome}|\text{action}, \text{evidence})$

Value of information: which evidence to seek next?

Sensitivity analysis: which probability values are most critical?

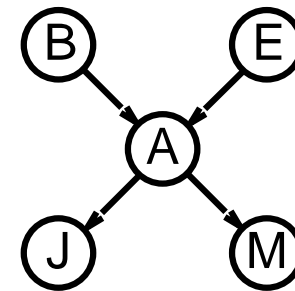
Explanation: why do I need a new starter motor?

## Inference by enumeration

Slightly intelligent way to sum out variables from the joint without actually constructing its explicit representation

Simple query on the burglary network:

$$\begin{aligned} & \mathbf{P}(B|j, m) \\ &= \mathbf{P}(B, j, m) / P(j, m) \\ &= \alpha \mathbf{P}(B, j, m) \\ &= \alpha \sum_e \sum_a \mathbf{P}(B, e, a, j, m) \end{aligned}$$



Rewrite full joint entries using product of CPT entries:

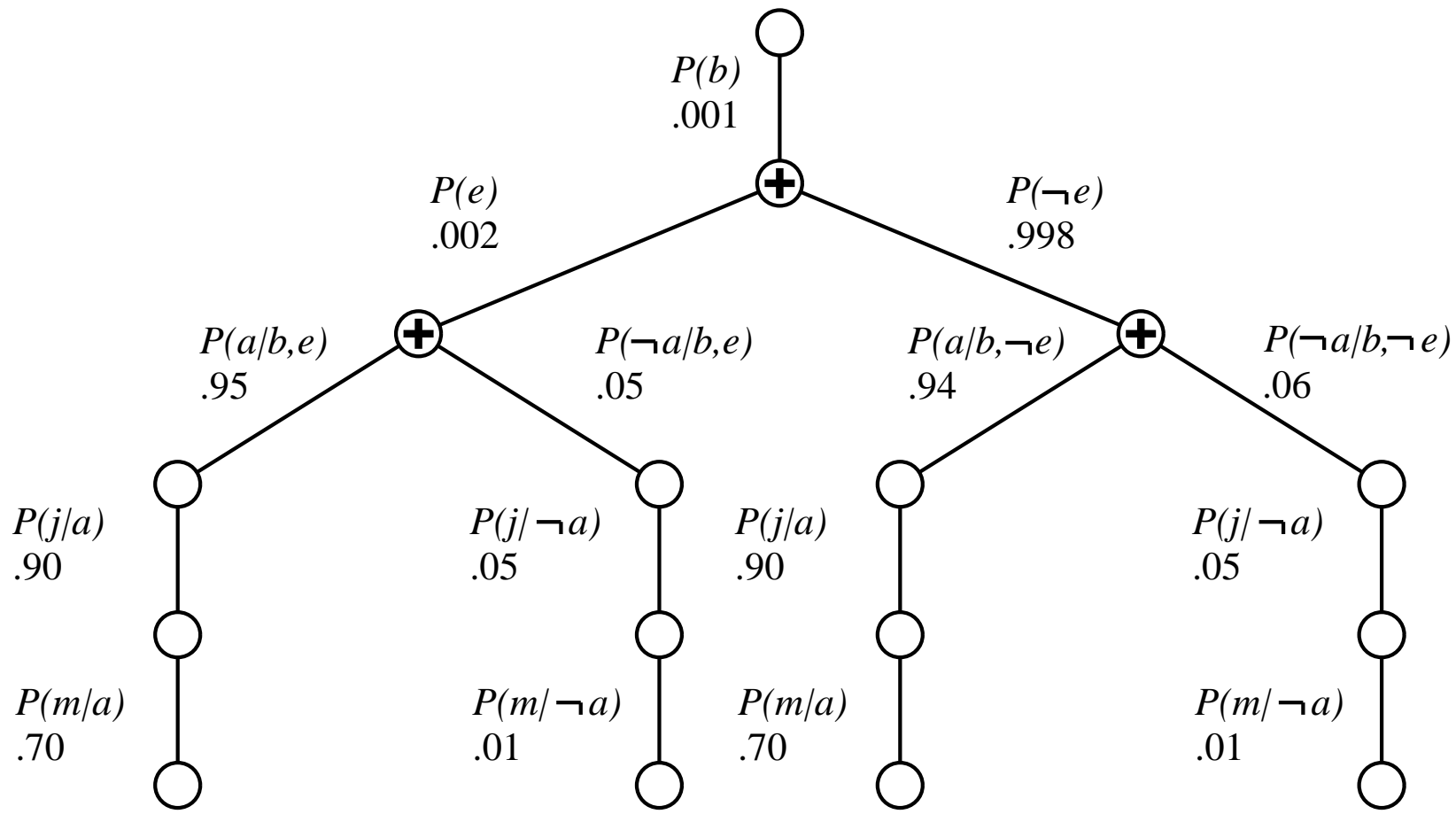
$$\begin{aligned} & \mathbf{P}(B|j, m) \\ &= \alpha \sum_e \sum_a \mathbf{P}(B) P(e) \mathbf{P}(a|B, e) P(j|a) P(m|a) \\ &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) P(j|a) P(m|a) \end{aligned}$$

Recursive depth-first enumeration:  $O(n)$  space,  $O(d^n)$  time

# Evaluation tree

Enumeration is inefficient: repeated computation

e.g., computes  $P(j|a)P(m|a)$  for each value of  $e$



## Inference by variable elimination

Variable elimination: carry out summations right-to-left, storing intermediate results (**factors**) to avoid recomputation

$$\begin{aligned} \mathbf{P}(B|j, m) &= \alpha \underbrace{\mathbf{P}(B)}_B \underbrace{\sum_e P(e)}_E \underbrace{\sum_a \mathbf{P}(a|B, e)}_A \underbrace{P(j|a)}_J \underbrace{P(m|a)}_M \\ &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) P(j|a) f_M(a) \\ &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) f_J(a) f_M(a) \\ &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a f_A(a, b, e) f_J(a) f_M(a) \\ &= \alpha \mathbf{P}(B) \sum_e P(e) f_{\bar{A}JM}(b, e) \text{ (sum out } A) \\ &= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E) \\ &= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b) \end{aligned}$$

# Inference by variable elimination

Variable elimination: carry out summations right-to-left, storing intermediate results (**factors**) to avoid recomputation

$$\begin{aligned}
 \mathbf{P}(B|j, m) &= \alpha \underbrace{\mathbf{P}(B)}_B \underbrace{\sum_e P(e)}_E \underbrace{\sum_a \mathbf{P}(a|B, e)}_A \underbrace{P(j|a)}_J \underbrace{P(m|a)}_M \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) P(j|a) \underbrace{f_M(a)} \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) f_J(a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a f_A(a, b, e) f_J(a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) f_{\bar{A}JM}(b, e) \text{ (sum out } A) \\
 &= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E) \\
 &= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
 \end{aligned}$$

vector  
 $P(m|a)$   
 $P(m|\sim a)$

# Inference by variable elimination

Variable elimination: carry out summations right-to-left, storing intermediate results (**factors**) to avoid recomputation

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

$$\begin{aligned}
 &= \alpha \underbrace{\mathbf{P}(B)}_B \underbrace{\sum_e P(e)}_E \underbrace{\sum_a \mathbf{P}(a|B, e)}_A \underbrace{P(j|a)}_J \underbrace{P(m|a)}_M \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) P(j|a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) f_J(a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a f_A(a, b, e) f_J(a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) f_{\bar{A}JM}(b, e) \text{ (sum out } A) \\
 &= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E) \\
 &= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
 \end{aligned}$$

vector  
 $P(j|a)$   
 $P(j|\sim a)$



# Inference by variable elimination

Variable elimination: carry out summations right-to-left, storing intermediate results (**factors**) to avoid recomputation

$$\begin{aligned}
 \mathbf{P}(B|j, m) &= \alpha \underbrace{\mathbf{P}(B)}_B \underbrace{\sum_e P(e)}_E \underbrace{\sum_a \mathbf{P}(a|B, e)}_A \underbrace{P(j|a)}_J \underbrace{P(m|a)}_M \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) P(j|a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) f_J(a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \underbrace{\sum_a f_A(a, b, e)}_{\text{circled}} f_J(a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) f_{\bar{A}JM}(b, e) \text{ (sum out } A) \\
 &= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E) \\
 &= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
 \end{aligned}$$

2x2x2 array

$P(a|b, e)$     $P(a|\sim b, e)$

$P(a|b, \sim e)$     $P(a|\sim b, \sim e)$

-----  
 $P(\sim a|b, e)$     $P(\sim a|\sim b, e)$

$P(\sim a|b, \sim e)$     $P(\sim a|\sim b, \sim e)$

# Inference by variable elimination

Variable elimination: carry out summations right-to-left, storing intermediate results (**factors**) to avoid recomputation

$$\begin{aligned}
 \mathbf{P}(B|j, m) &= \alpha \underbrace{\mathbf{P}(B)}_B \underbrace{\sum_e P(e)}_E \underbrace{\sum_a \mathbf{P}(a|B, e)}_A \underbrace{P(j|a)}_J \underbrace{P(m|a)}_M \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) P(j|a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) f_J(a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a f_A(a, b, e) f_J(a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) f_{\bar{A}JM}(b, e) \text{ (sum out } A) \\
 &= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E) \\
 &= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
 \end{aligned}$$

2x2 array

$P(m, j b, e)$	$P(m, j \sim b, e)$
$P(m, j b, \sim e)$	$P(m, j \sim b, \sim e)$

## Inference by variable elimination

Variable elimination: carry out summations right-to-left, storing intermediate results (**factors**) to avoid recomputation

$$\begin{aligned}
 \mathbf{P}(B|j, m) &= \alpha \underbrace{\mathbf{P}(B)}_B \underbrace{\sum_e P(e)}_E \underbrace{\sum_a \mathbf{P}(a|B, e)}_A \underbrace{P(j|a)}_J \underbrace{P(m|a)}_M \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) P(j|a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) f_J(a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a f_A(a, b, e) f_J(a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) f_{\bar{A}JM}(b, e) \text{ (sum out } A) \\
 &= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E) \\
 &= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
 \end{aligned}$$

vector  
 $P(m, j|b)$   
 $P(m, j|\sim b)$

## Inference by variable elimination

Variable elimination: carry out summations right-to-left, storing intermediate results (**factors**) to avoid recomputation

$$\begin{aligned}
 \mathbf{P}(B|j, m) &= \alpha \underbrace{\mathbf{P}(B)}_B \underbrace{\sum_e P(e)}_E \underbrace{\sum_a \mathbf{P}(a|B, e)}_A \underbrace{P(j|a)}_J \underbrace{P(m|a)}_M \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) P(j|a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) f_J(a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a f_A(a, b, e) f_J(a) f_M(a) \\
 &= \alpha \mathbf{P}(B) \sum_e P(e) f_{\bar{A}JM}(b, e) \text{ (sum out } A) \\
 &= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E) \\
 &= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
 \end{aligned}$$

vector  
 $P(m, j|b)P(b)$   
 $P(m, j|\sim b)P(\sim b)$

Note use of normalization constant instead of calculating denominator in Bayes law

## Variable elimination: Basic operations

Summing out a variable from a product of factors:

move any constant factors outside the summation

add up submatrices in pointwise product of remaining factors

$$\sum_x f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_i \sum_x f_{i+1} \times \cdots \times f_k = f_1 \times \cdots \times f_i \times f_{\bar{X}}$$

assuming  $f_1, \dots, f_i$  do not depend on  $X$

Pointwise product of factors  $f_1$  and  $f_2$ :

$$\begin{aligned} f_1(x_1, \dots, x_j, y_1, \dots, y_k) \times f_2(y_1, \dots, y_k, z_1, \dots, z_l) \\ = f(x_1, \dots, x_j, y_1, \dots, y_k, z_1, \dots, z_l) \end{aligned}$$

E.g.,  $f_1(a, b) \times f_2(b, c) = f(a, b, c)$

## Variable elimination algorithm

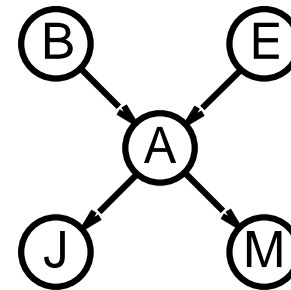
```
function ELIMINATION-ASK( $X, e, bn$ ) returns a distribution over  $X$   
inputs:  $X$ , the query variable  
          $e$ , evidence specified as an event  
          $bn$ , a belief network specifying joint distribution  $\mathbf{P}(X_1, \dots, X_n)$   
  
 $factors \leftarrow []$ ;  $vars \leftarrow \text{REVERSE}(\text{VARS}[bn])$   
for each  $var$  in  $vars$  do  
     $factors \leftarrow [\text{MAKE-FACTOR}(var, e) | factors]$   
    if  $var$  is a hidden variable then  $factors \leftarrow \text{SUM-OUT}(var, factors)$   
return  $\text{NORMALIZE}(\text{POINTWISE-PRODUCT}(factors))$ 
```

## Irrelevant variables

Consider the query  $P(\text{JohnCalls} | \text{Burglary} = \text{true})$

$$P(J|b) = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) P(J|a) \sum_m P(m|a)$$

Sum over  $m$  is identically 1;  $M$  is **irrelevant** to the query



Thm 1:  $Y$  is irrelevant unless  $Y \in \text{Ancestors}(\{X\} \cup \mathbf{E})$

Here,  $X = \text{JohnCalls}$ ,  $\mathbf{E} = \{\text{Burglary}\}$ , and  
 $\text{Ancestors}(\{X\} \cup \mathbf{E}) = \{\text{Alarm}, \text{Earthquake}\}$   
so  $M$  is irrelevant

(Compare this to backward chaining from the query in Horn clause KBs)

# Complexity of exact inference

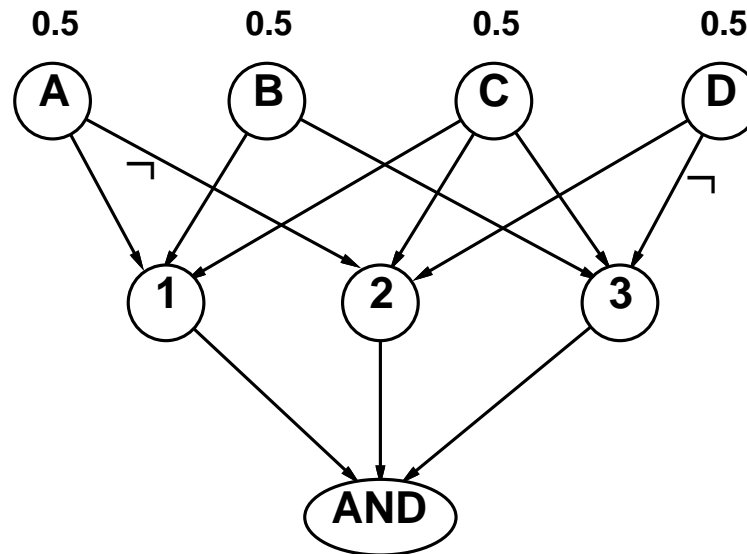
Singly connected networks (or **polytrees**):

- any two nodes are connected by at most one (undirected) path
- time and space cost of variable elimination are  $O(d^k n)$

Multiply connected networks:

- can reduce 3SAT to exact inference  $\Rightarrow$  NP-hard
- equivalent to **counting** 3SAT models  $\Rightarrow$  #P-complete

1.  $A \vee B \vee C$
2.  $C \vee D \vee \neg A$
3.  $B \vee C \vee \neg D$





# Inference by stochastic simulation

Basic idea:

- 1) Draw  $N$  samples from a sampling distribution  $S$
- 2) Compute an approximate posterior probability  $\hat{P}$
- 3) Show this converges to the true probability  $P$



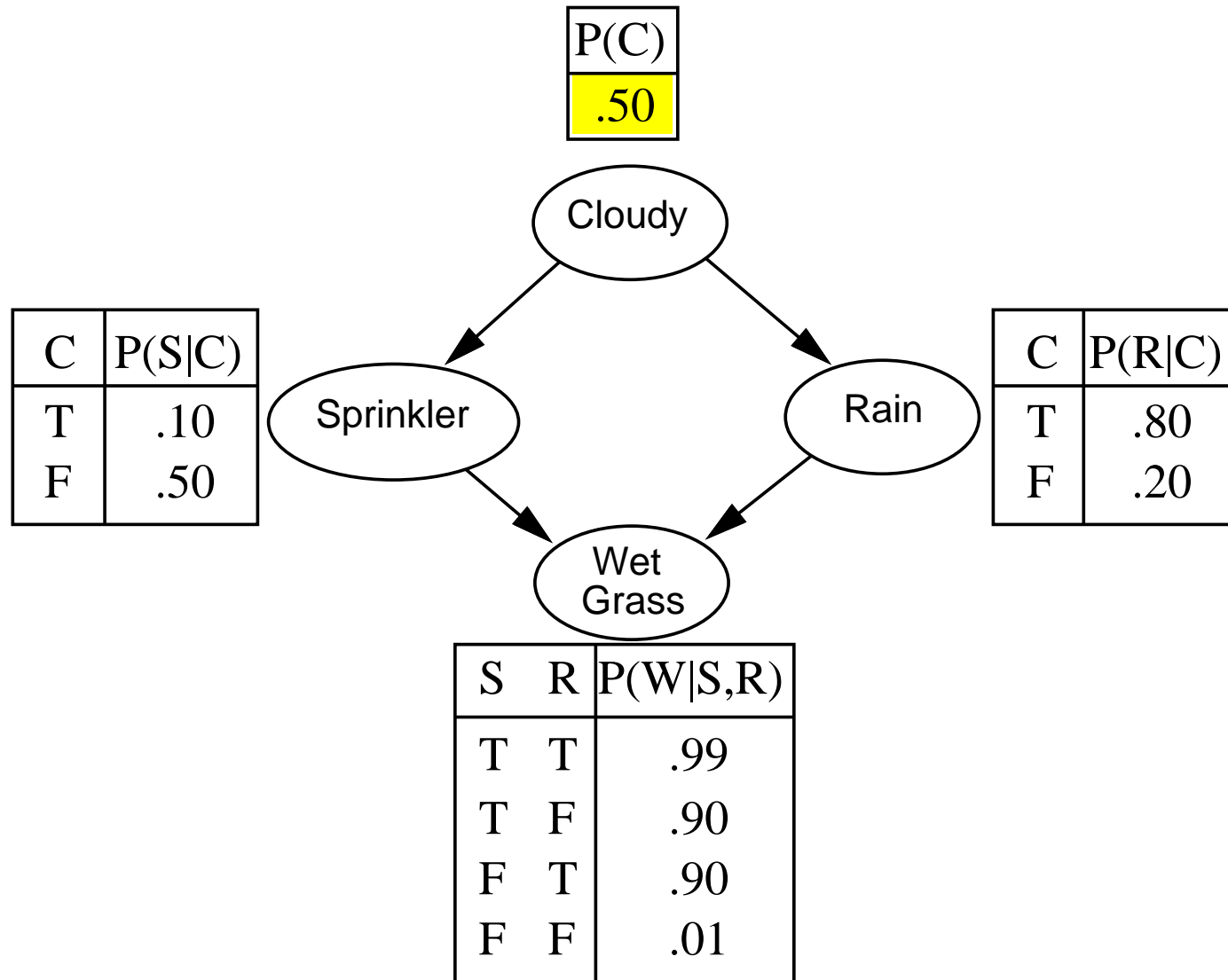
Outline:

- Sampling from an empty network
- Rejection sampling: reject samples disagreeing with evidence
- Likelihood weighting: use evidence to weight samples
- Markov chain Monte Carlo (MCMC): sample from a stochastic process whose stationary distribution is the true posterior

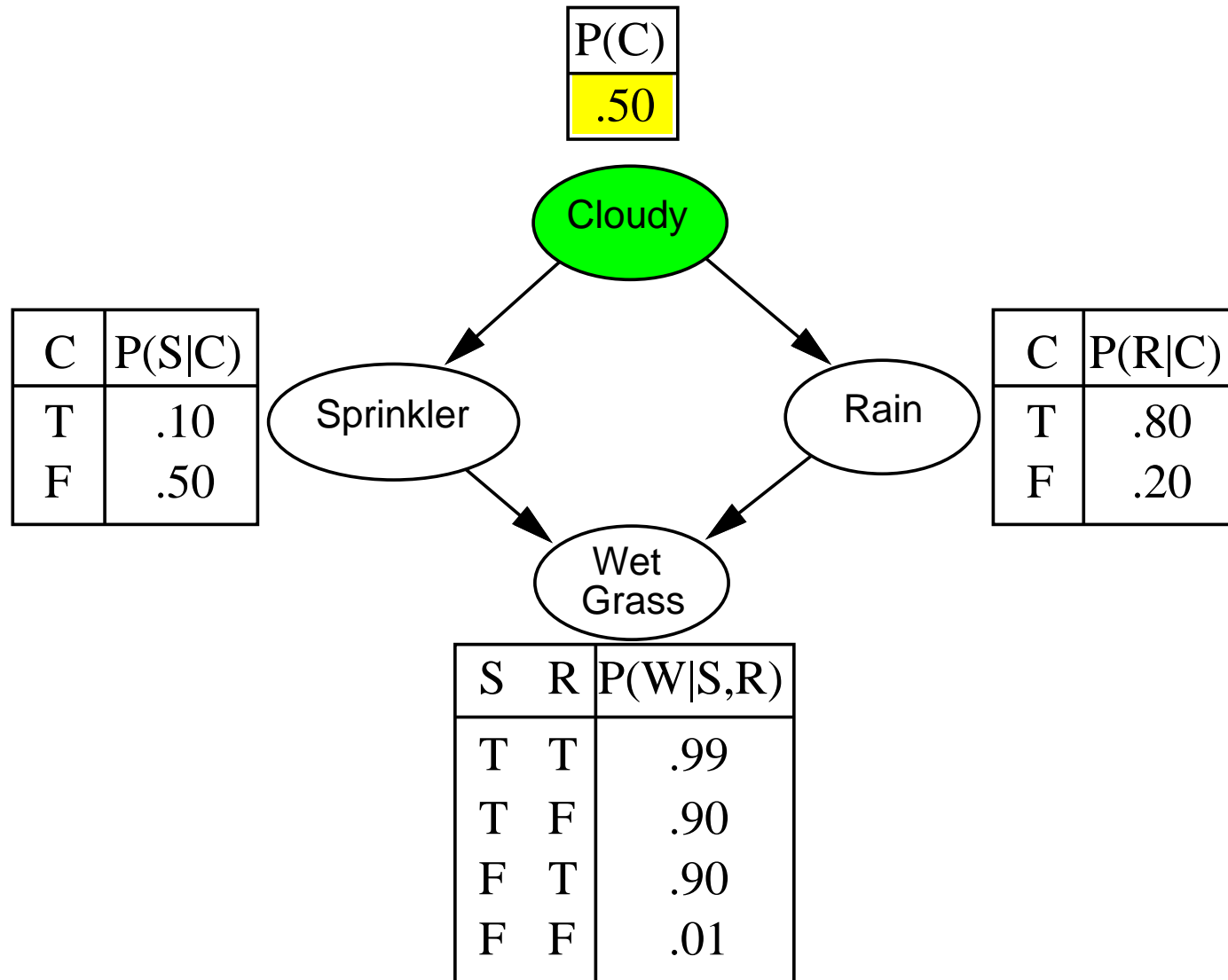
## Sampling from an empty network

```
function PRIOR-SAMPLE(bn) returns an event sampled from bn  
inputs: bn, a belief network specifying joint distribution  $\mathbf{P}(X_1, \dots, X_n)$   
x  $\leftarrow$  an event with n elements  
for i = 1 to n do  
     $x_i \leftarrow$  a random sample from  $\mathbf{P}(X_i \mid \text{Parents}(X_i))$   
return x
```

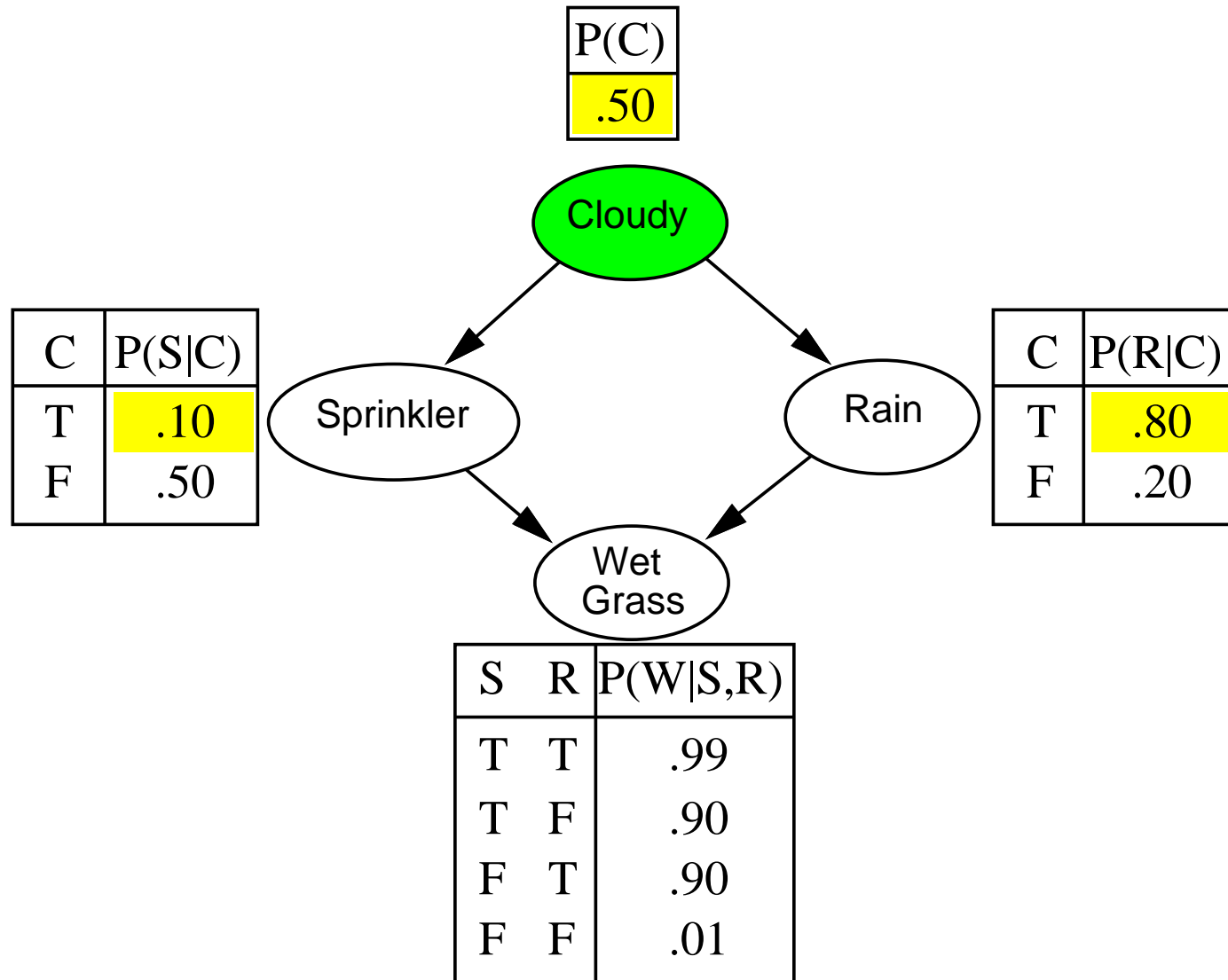
# Example



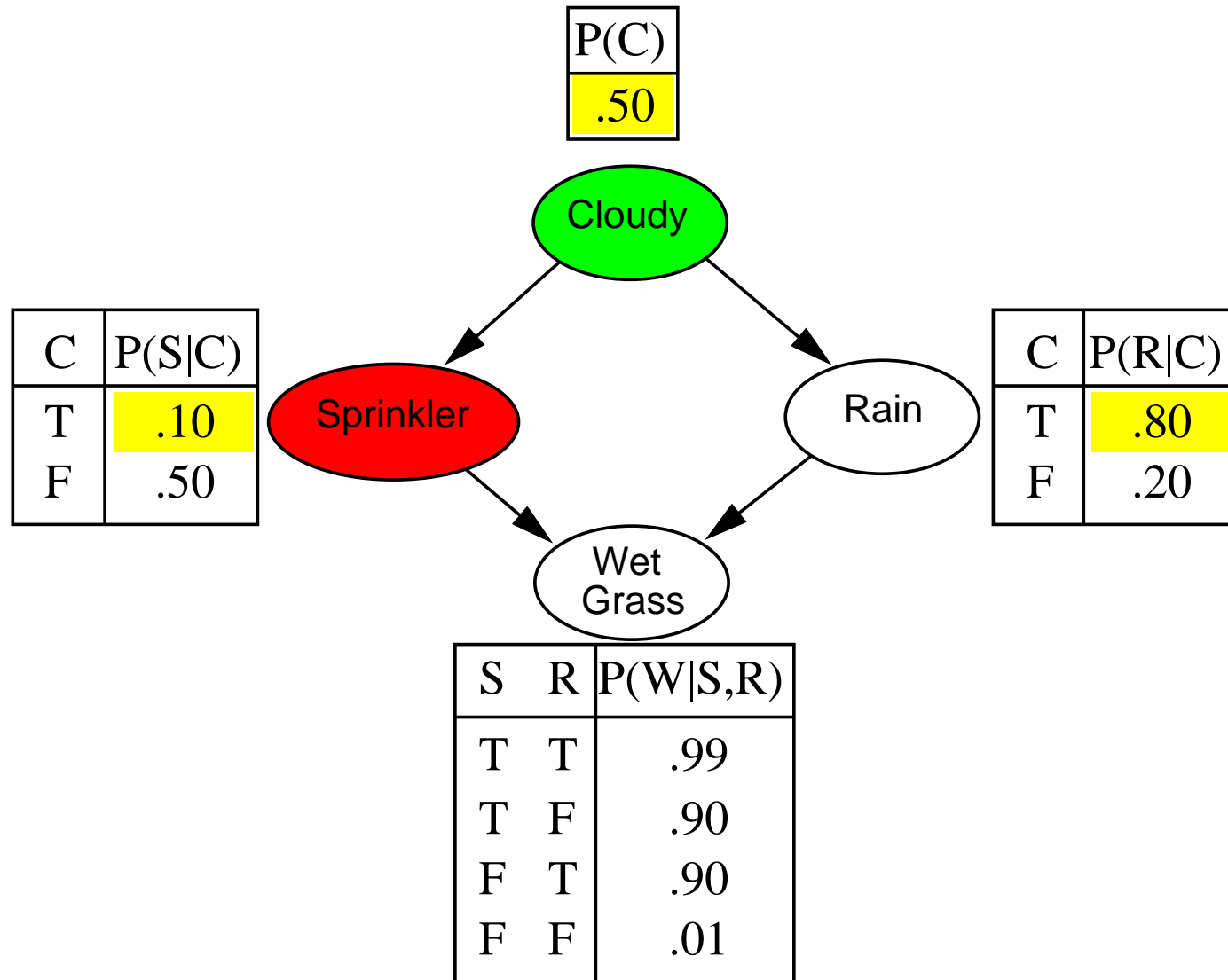
# Example



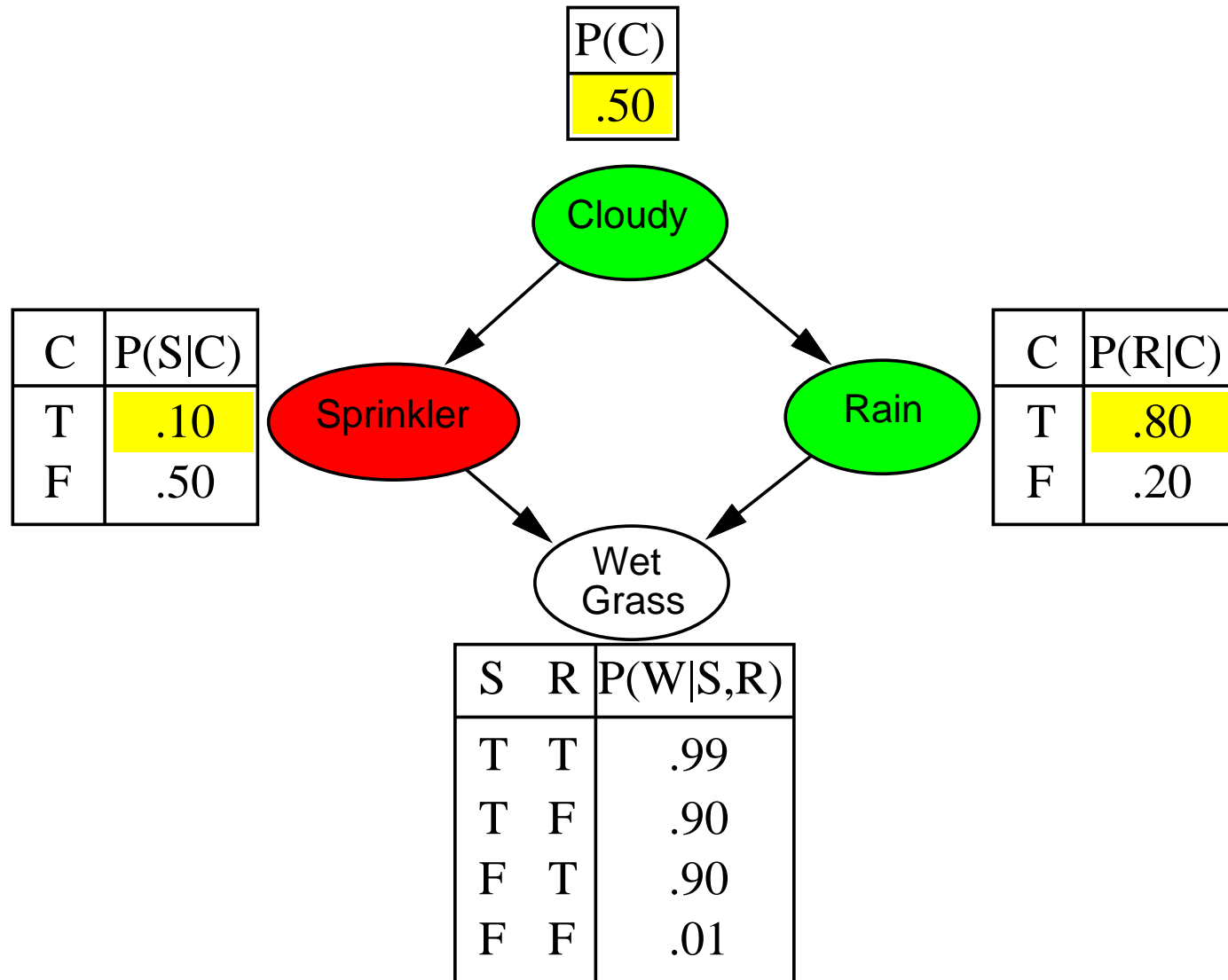
# Example



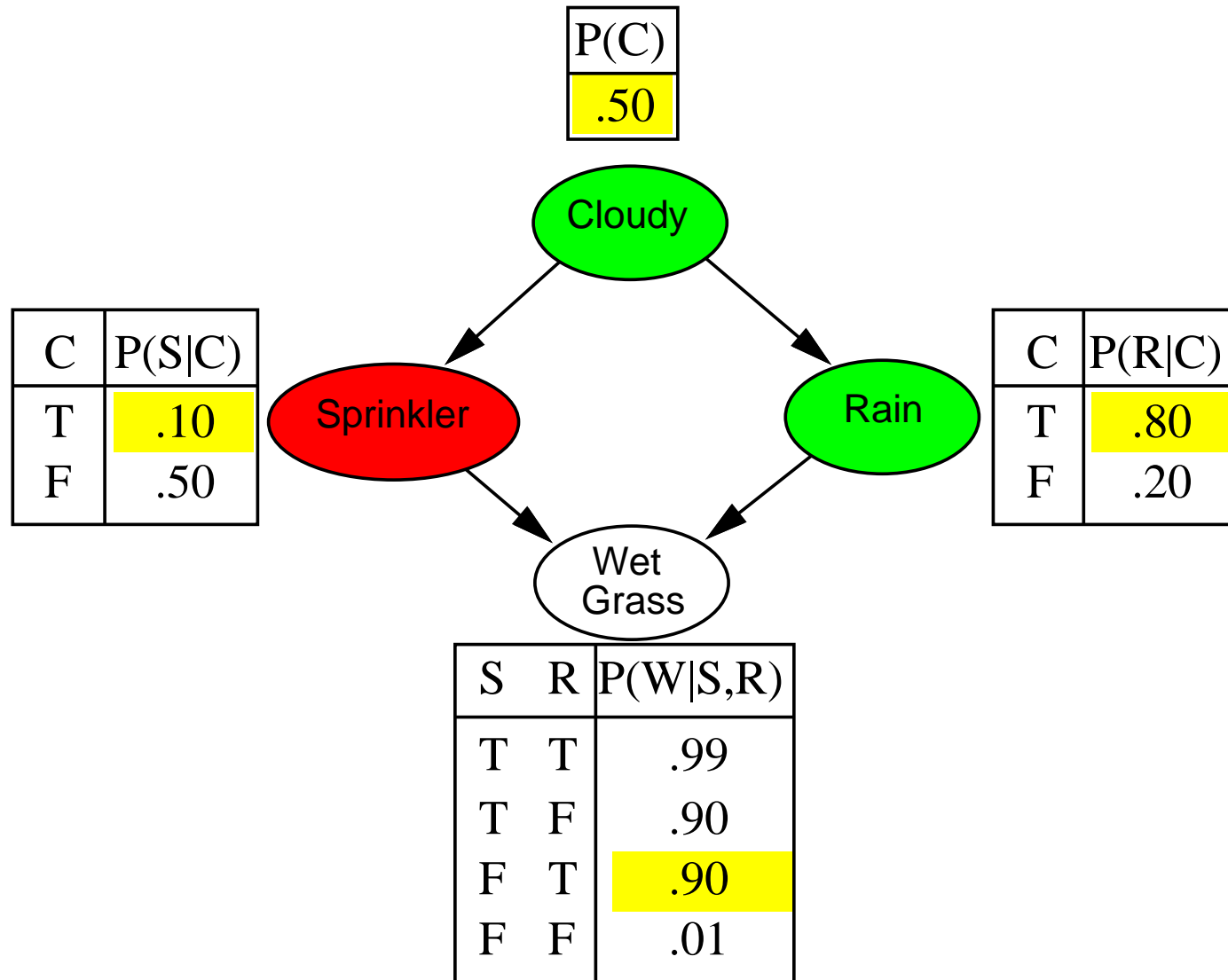
# Example



# Example

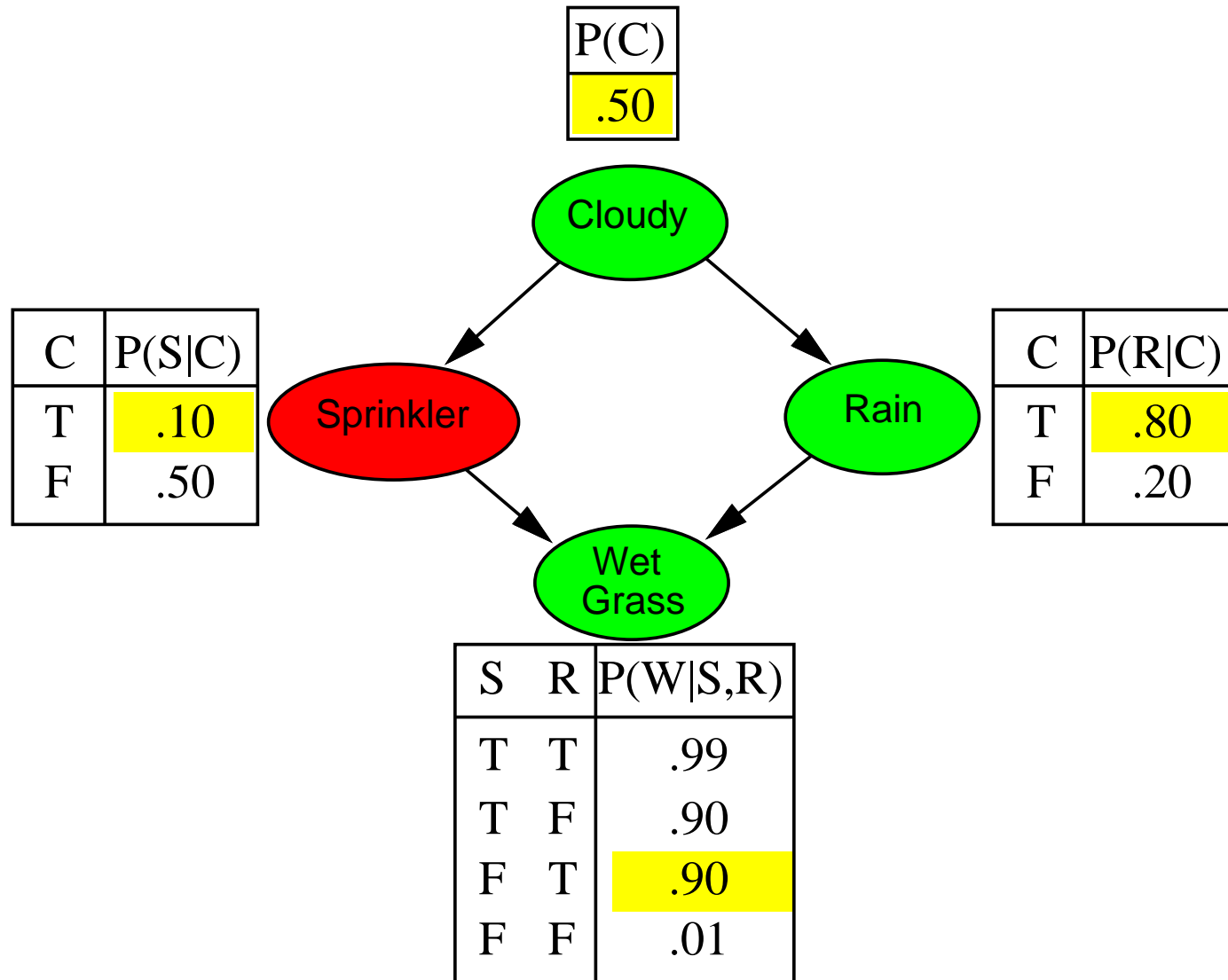


# Example





# Example



## Sampling from an empty network contd.

Probability that PRIORSAMPLE generates a particular event

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

i.e., the true prior probability

E.g.,  $S_{PS}(t, f, t, t) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324 = P(t, f, t, t)$

Let  $N_{PS}(x_1 \dots x_n)$  be the number of samples generated for event  $x_1, \dots, x_n$

Then we have

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

That is, estimates derived from PRIORSAMPLE are **consistent**

Shorthand:  $\hat{P}(x_1, \dots, x_n) \approx P(x_1 \dots x_n)$

## Rejection sampling

$\hat{\mathbf{P}}(X|\mathbf{e})$  estimated from samples agreeing with  $\mathbf{e}$

```
function REJECTION-SAMPLING( $X, \mathbf{e}, bn, N$ ) returns an estimate of  $P(X|\mathbf{e})$ 
  local variables:  $\mathbf{N}$ , a vector of counts over  $X$ , initially zero
  for  $j = 1$  to  $N$  do
     $\mathbf{x} \leftarrow \text{PRIOR-SAMPLE}(bn)$ 
    if  $\mathbf{x}$  is consistent with  $\mathbf{e}$  then
       $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$  where  $x$  is the value of  $X$  in  $\mathbf{x}$ 
  return NORMALIZE( $\mathbf{N}[X]$ )
```

E.g., estimate  $\mathbf{P}(Rain|Sprinkler = true)$  using 100 samples

27 samples have  $Sprinkler = true$

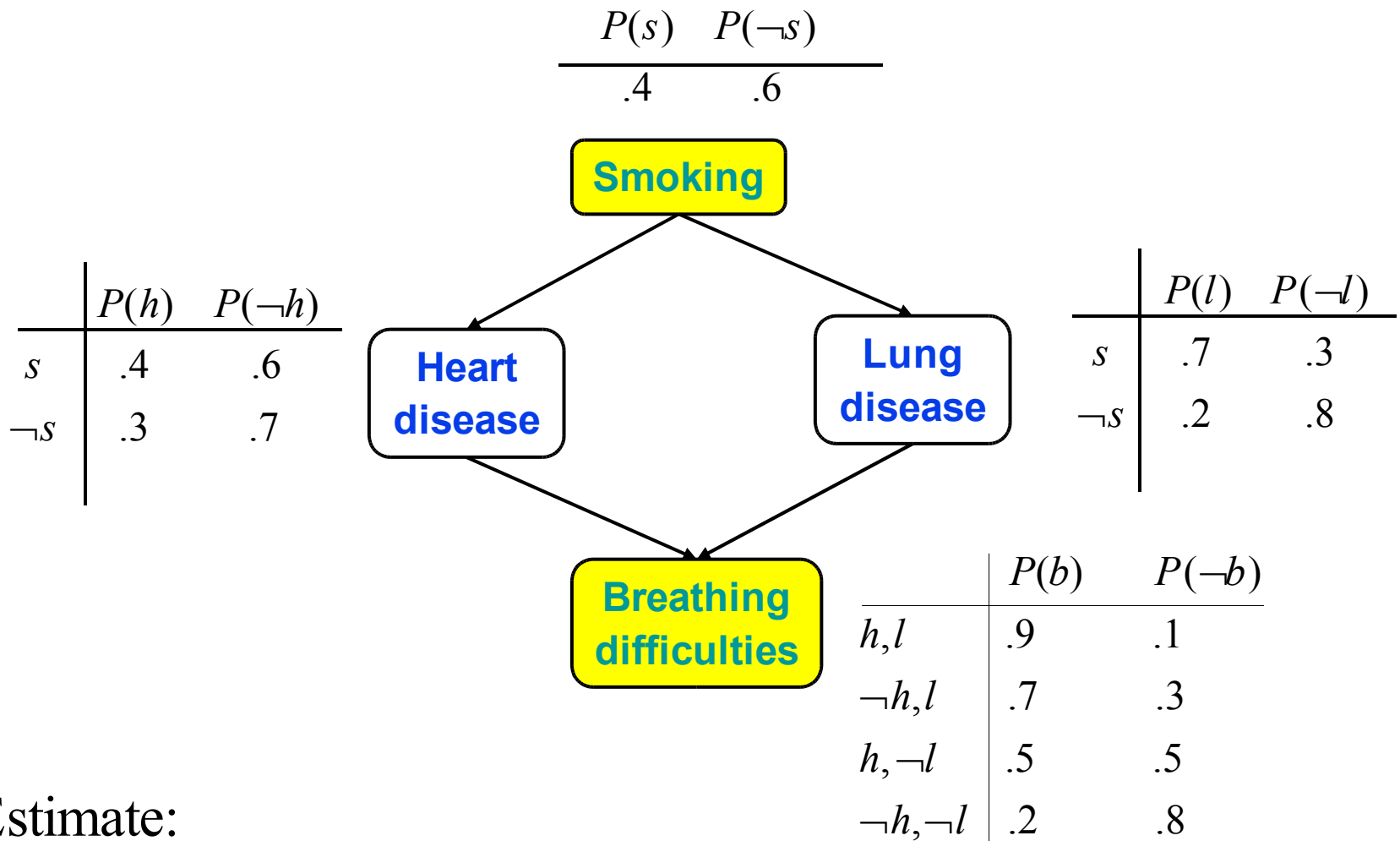
Of these, 8 have  $Rain = true$  and 19 have  $Rain = false$ .

$\hat{\mathbf{P}}(Rain|Sprinkler = true) = \text{NORMALIZE}(\langle 8, 19 \rangle) = \langle 0.296, 0.704 \rangle$

Similar to a basic real-world empirical estimation procedure

# Exercise: Inference by Stochastic Sampling

- Write down
  - last 4 digits of your SS#
  - last 4 digits of your home phone number
  - your favorite 4 digit lucky number
- Use each digit as a random biased coin
  - digit < Prob. is “yes”
  - $P(\text{foo})=0.5$  then  $\text{foo}=\text{true}$  for 0, 1, 2, 3, 4
  - $P(\text{foo})=0.3$  then  $\text{foo}=\text{true}$  for 0, 1, 2



Estimate:

$$P(H \mid S = \text{true}, B = \text{true})$$

by rejection sampling

## Analysis of rejection sampling

$$\begin{aligned}\hat{\mathbf{P}}(X|\mathbf{e}) &= \alpha \mathbf{N}_{PS}(X, \mathbf{e}) && \text{(algorithm defn.)} \\ &= \mathbf{N}_{PS}(X, \mathbf{e}) / N_{PS}(\mathbf{e}) && \text{(normalized by } N_{PS}(\mathbf{e})\text{)} \\ &\approx \mathbf{P}(X, \mathbf{e}) / P(\mathbf{e}) && \text{(property of PRIORSAMPLE)} \\ &= \mathbf{P}(X|\mathbf{e}) && \text{(defn. of conditional probability)}\end{aligned}$$

Hence rejection sampling returns consistent posterior estimates

Problem: hopelessly expensive if  $P(\mathbf{e})$  is small

$P(\mathbf{e})$  drops off exponentially with number of evidence variables!

## Likelihood weighting

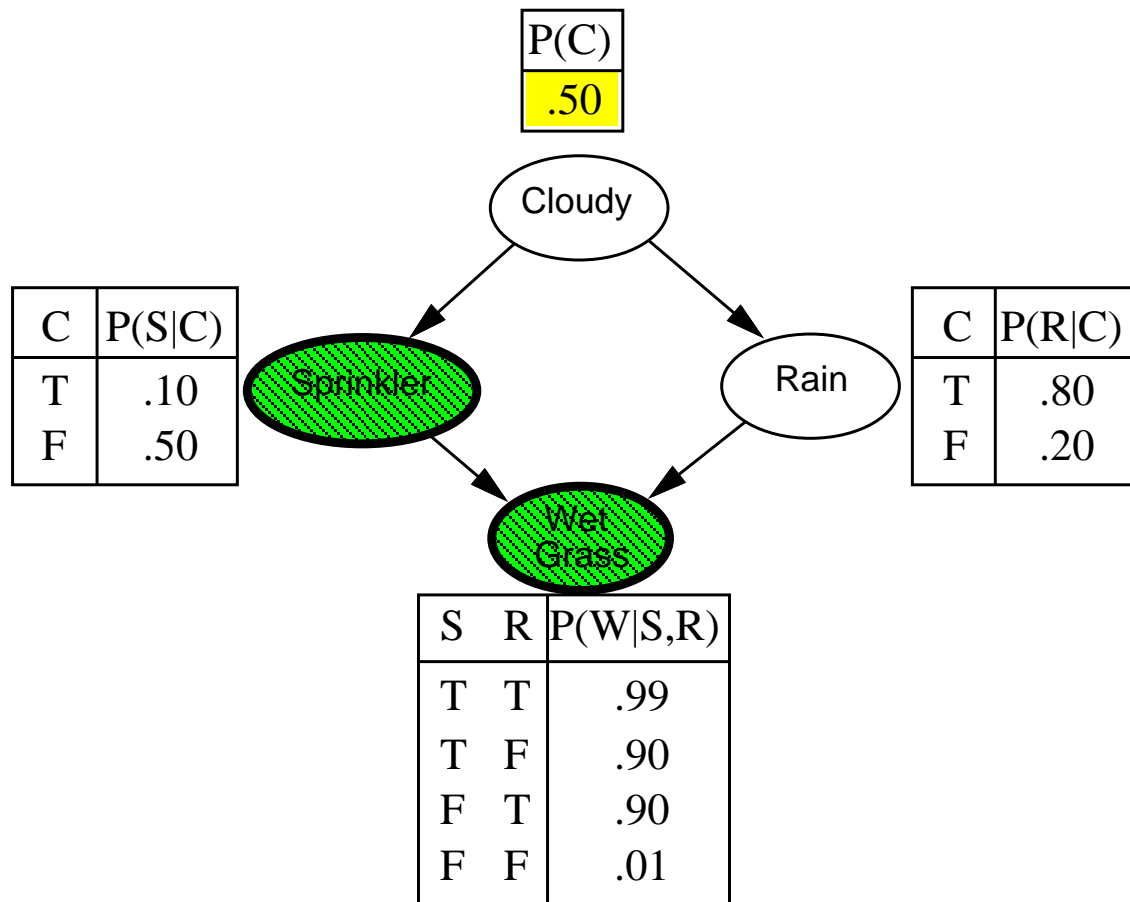
Idea: fix evidence variables, sample only nonevidence variables, and weight each sample by the likelihood it accords the evidence

```
function LIKELIHOOD-WEIGHTING( $X, e, bn, N$ ) returns an estimate of  $P(X|e)$   
  local variables:  $W$ , a vector of weighted counts over  $X$ , initially zero  
  for  $j = 1$  to  $N$  do  
     $x, w \leftarrow$  WEIGHTED-SAMPLE( $bn$ )  
     $W[x] \leftarrow W[x] + w$  where  $x$  is the value of  $X$  in  $x$   
  return NORMALIZE( $W[X]$ )
```

---

```
function WEIGHTED-SAMPLE( $bn, e$ ) returns an event and a weight  
   $x \leftarrow$  an event with  $n$  elements;  $w \leftarrow 1$   
  for  $i = 1$  to  $n$  do  
    if  $X_i$  has a value  $x_i$  in  $e$   
      then  $w \leftarrow w \times P(X_i = x_i \mid Parents(X_i))$   
      else  $x_i \leftarrow$  a random sample from  $P(X_i \mid Parents(X_i))$   
  return  $x, w$ 
```

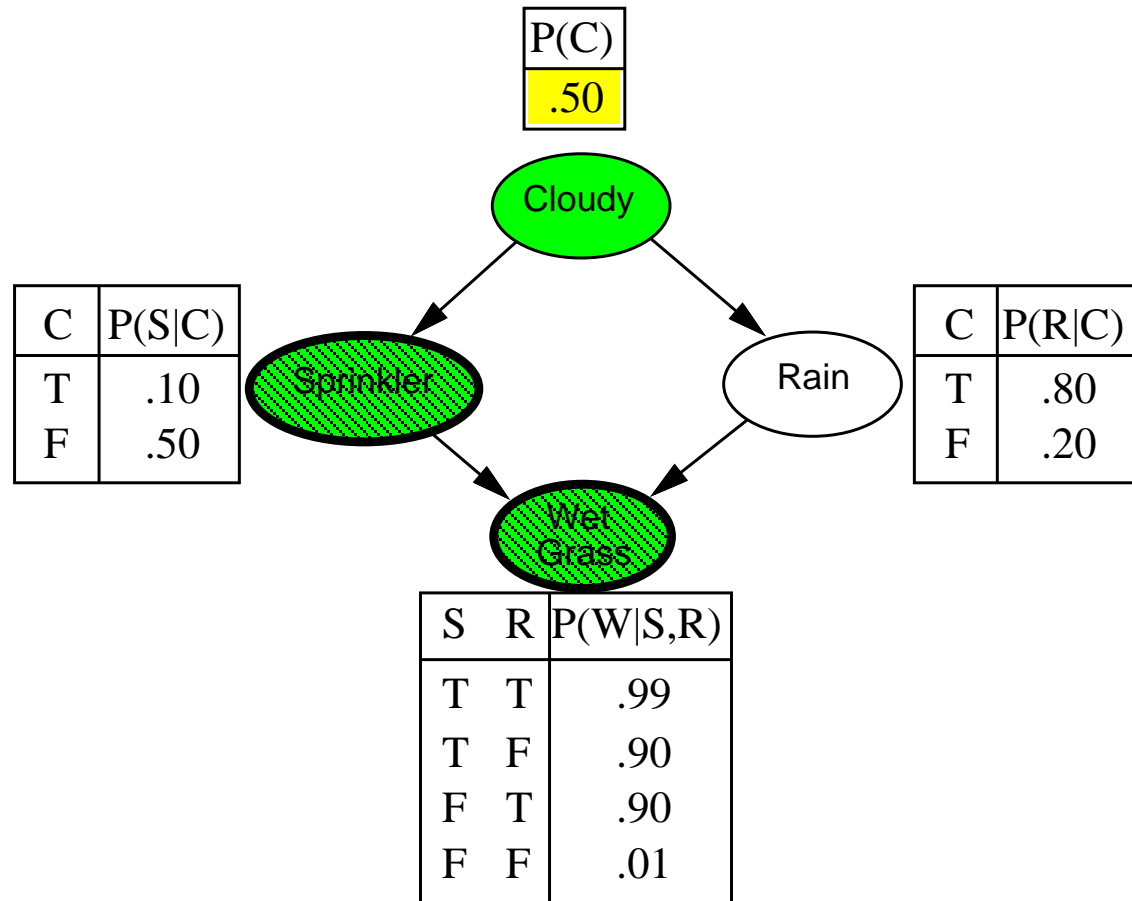
# Likelihood weighting example



$w = 1.0$

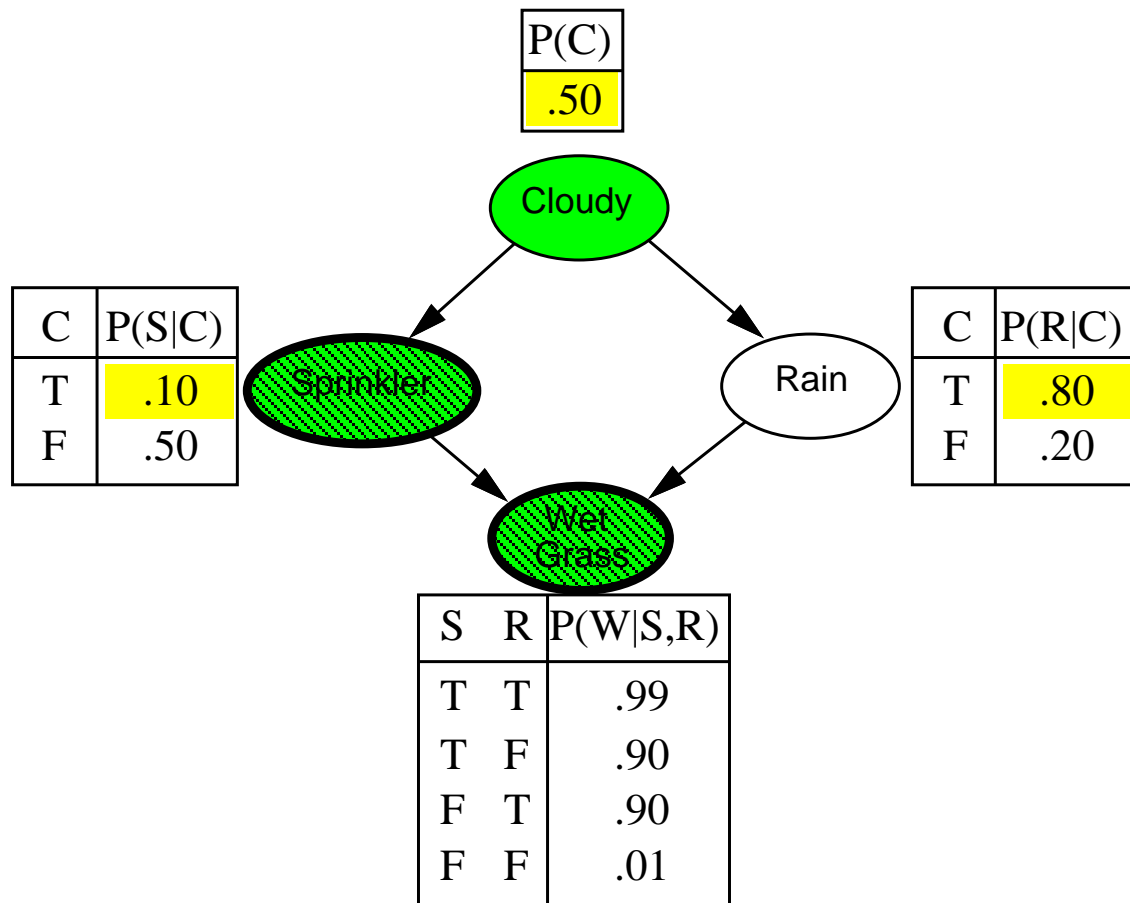


# Likelihood weighting example



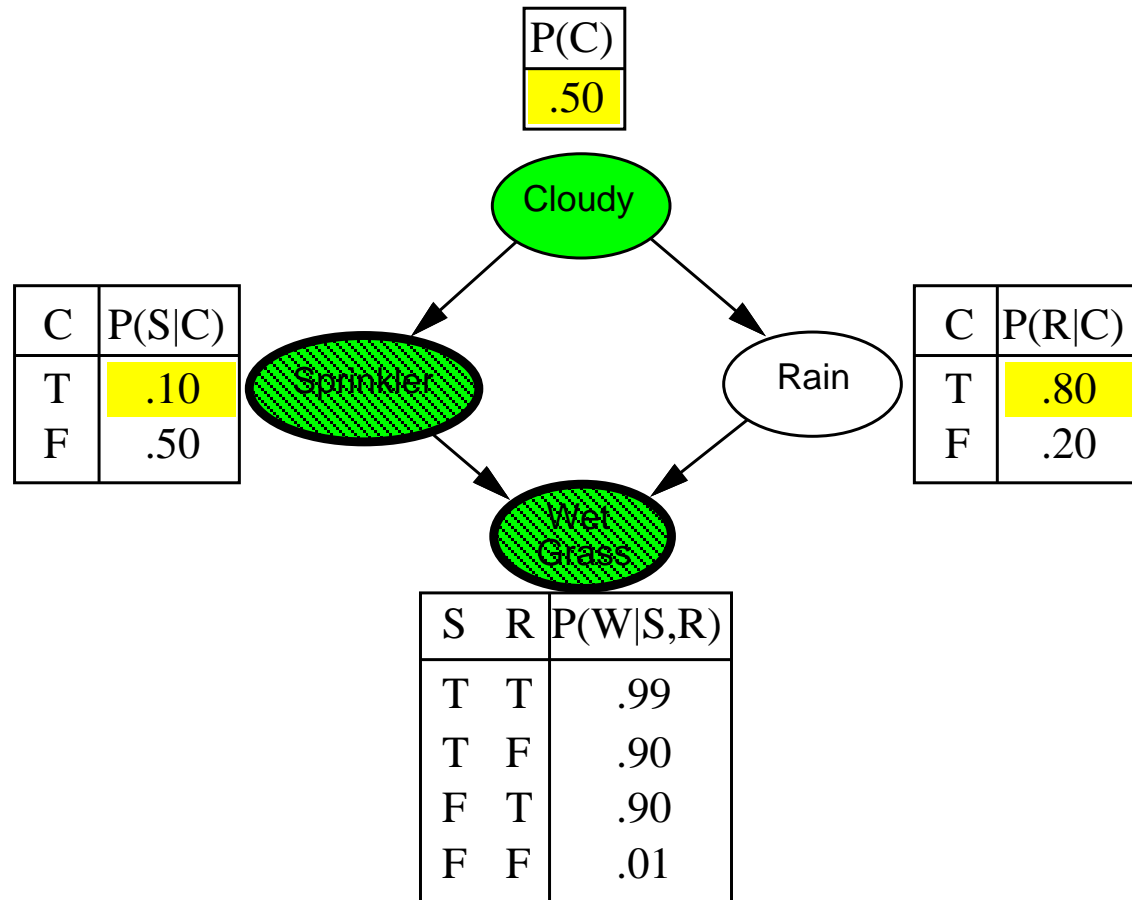
$w = 1.0$

# Likelihood weighting example



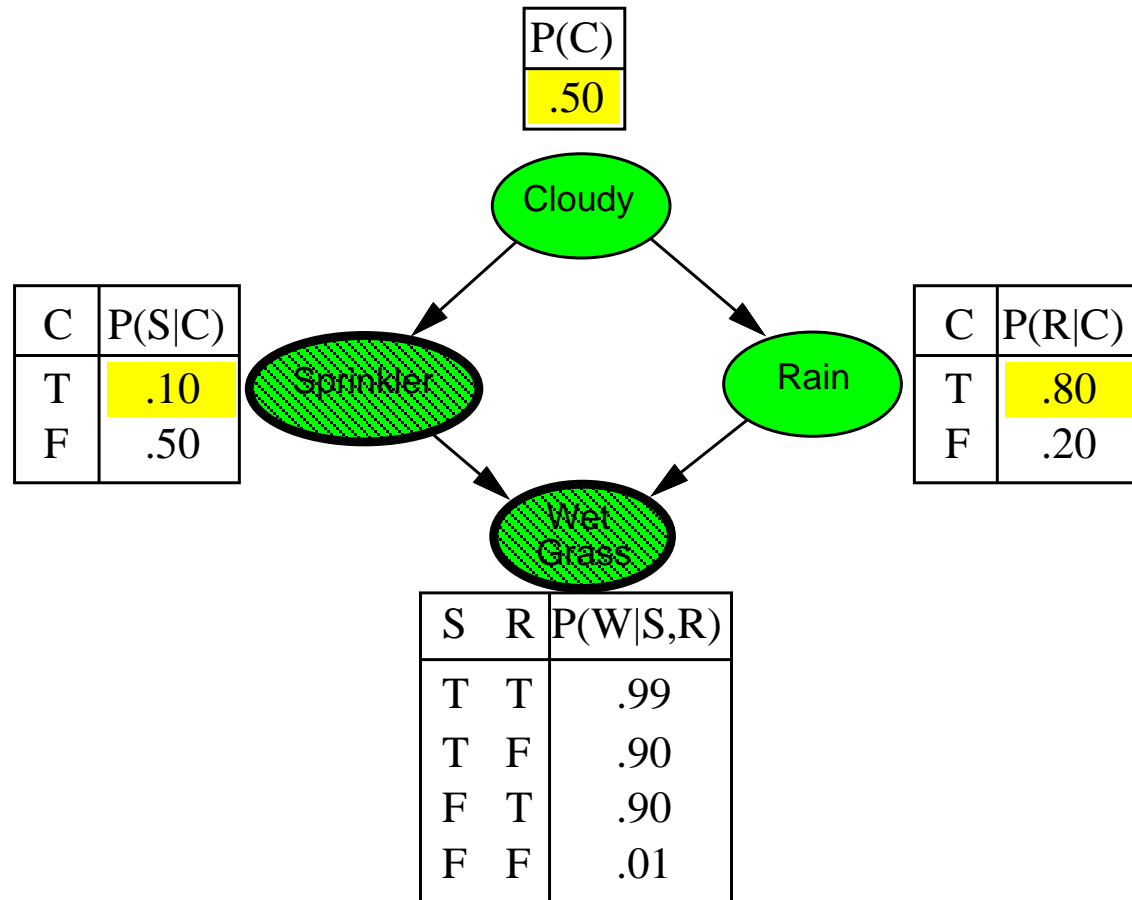
$w = 1.0$

# Likelihood weighting example



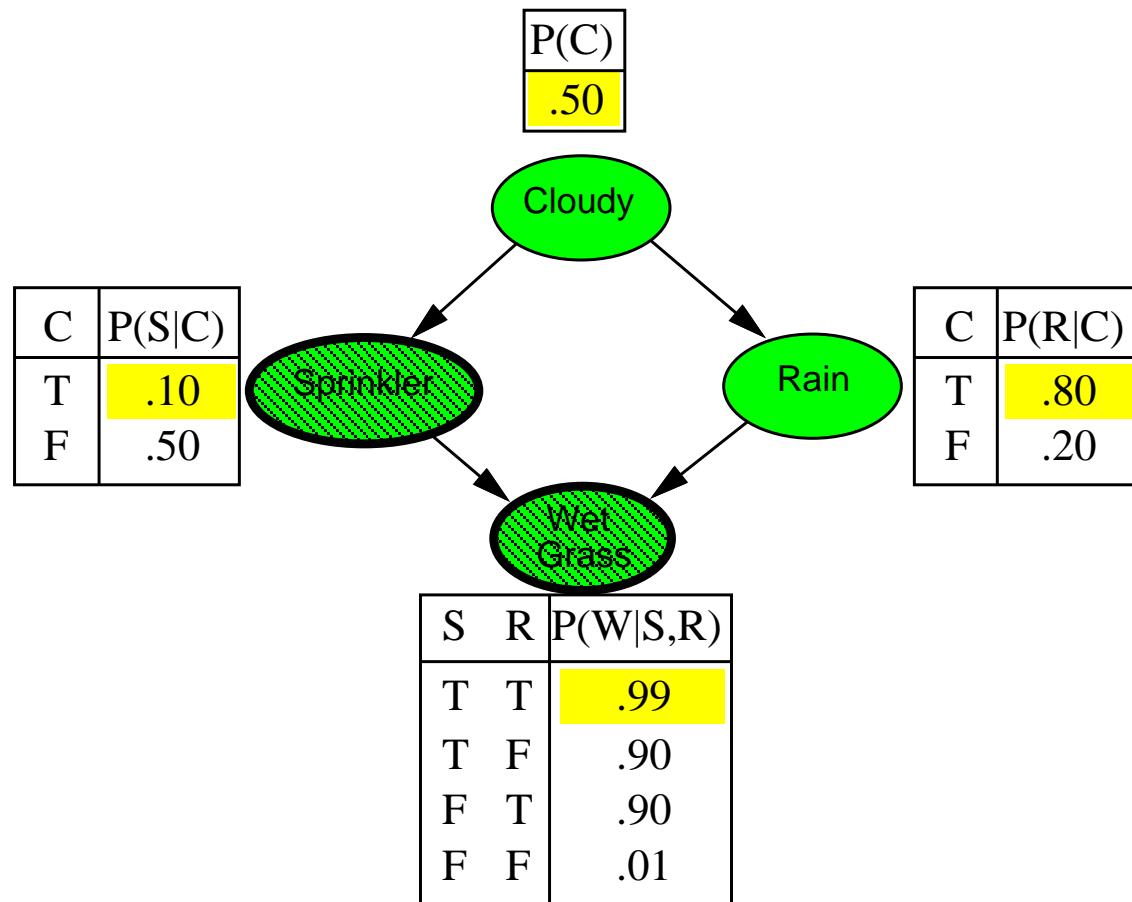
$$w = 1.0 \times 0.1$$

# Likelihood weighting example



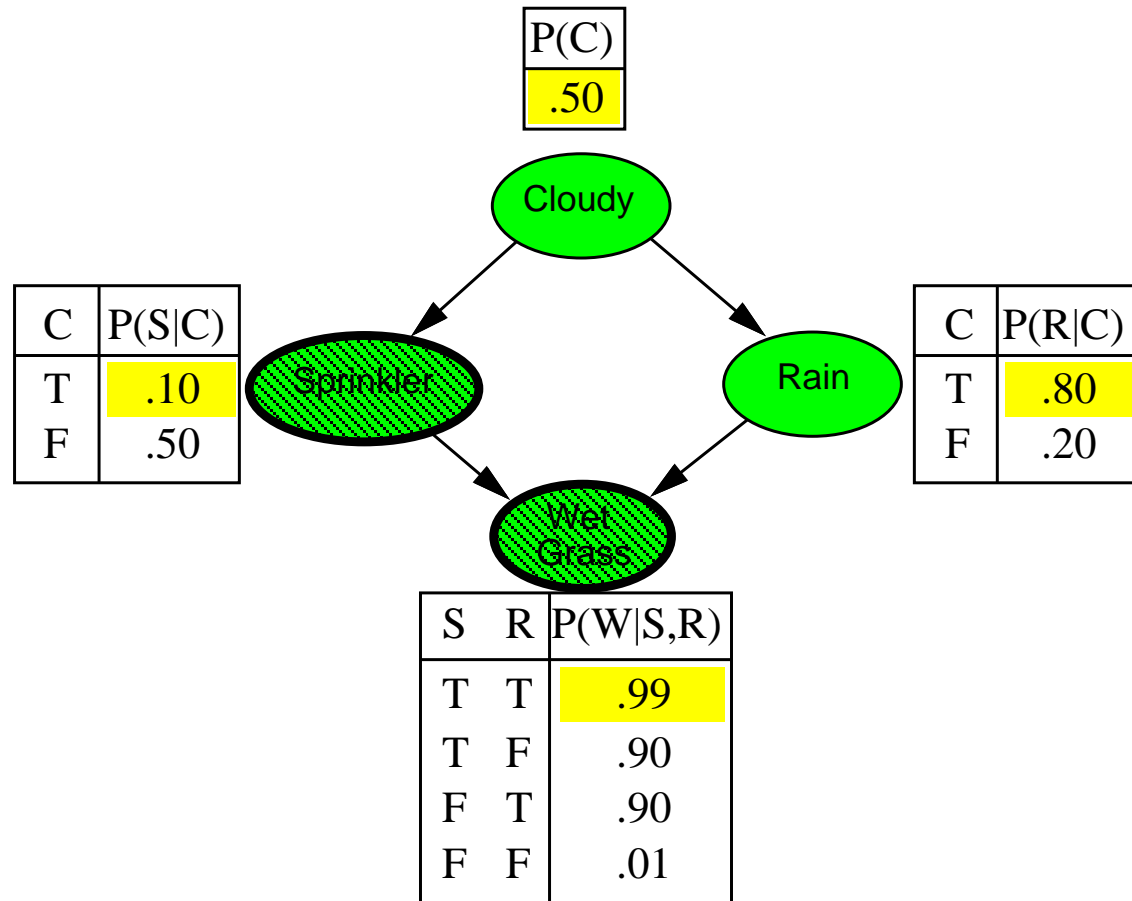
$$w = 1.0 \times 0.1$$

# Likelihood weighting example



$$w = 1.0 \times 0.1$$

# Likelihood weighting example



$$w = 1.0 \times 0.1 \times 0.99 = 0.099$$

# Likelihood weighting analysis

Sampling probability for WEIGHTEDSAMPLE is

$$S_{WS}(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^l P(z_i | Parents(Z_i))$$

Note: pays attention to evidence in **ancestors** only

⇒ somewhere “in between” prior and posterior distribution

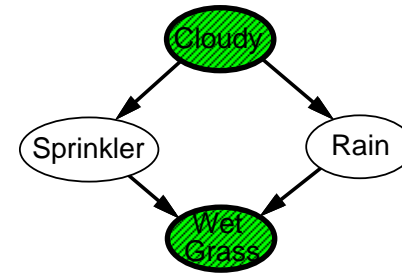
Weight for a given sample  $\mathbf{z}, \mathbf{e}$  is

$$w(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^m P(e_i | Parents(E_i))$$

Weighted sampling probability is

$$\begin{aligned} & S_{WS}(\mathbf{z}, \mathbf{e}) w(\mathbf{z}, \mathbf{e}) \\ &= \prod_{i=1}^l P(z_i | Parents(Z_i)) \prod_{i=1}^m P(e_i | Parents(E_i)) \\ &= P(\mathbf{z}, \mathbf{e}) \text{ (by standard global semantics of network)} \end{aligned}$$

Hence likelihood weighting returns consistent estimates but performance still degrades with many evidence variables because a few samples have nearly all the total weight



## Approximate inference using MCMC

“State” of network = current assignment to all variables.

Generate next state by sampling one variable given Markov blanket  
Sample each variable in turn, keeping evidence fixed

```
function MCMC-ASK( $X, e, bn, N$ ) returns an estimate of  $P(X|e)$ 
  local variables:  $\mathbf{N}[X]$ , a vector of counts over  $X$ , initially zero
                   $\mathbf{Z}$ , the nonevidence variables in  $bn$ 
                   $\mathbf{x}$ , the current state of the network, initially copied from  $e$ 

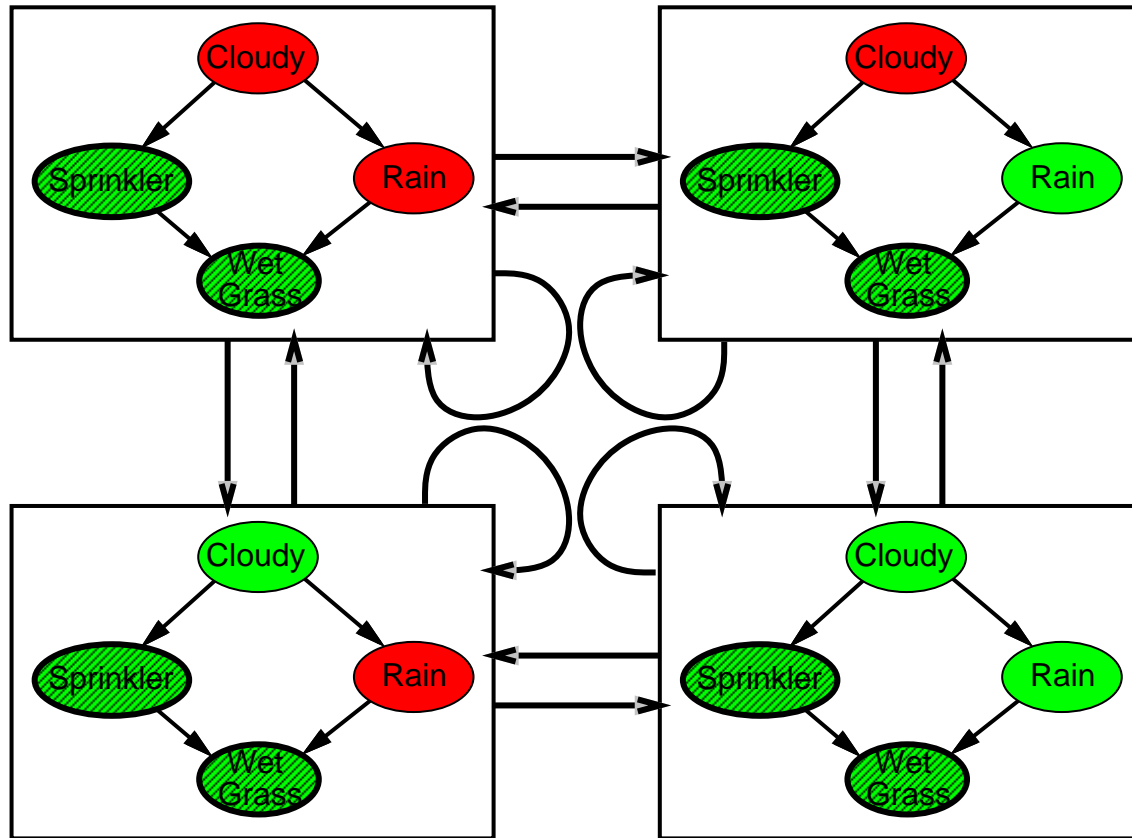
  initialize  $\mathbf{x}$  with random values for the variables in  $\mathbf{Y}$ 
  for  $j = 1$  to  $N$  do
     $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$  where  $x$  is the value of  $X$  in  $\mathbf{x}$ 
    for each  $Z_i$  in  $\mathbf{Z}$  do
      sample the value of  $Z_i$  in  $\mathbf{x}$  from  $\mathbf{P}(Z_i|MB(Z_i))$  given the values of
       $MB(Z_i)$  in  $\mathbf{x}$ 
  return NORMALIZE( $\mathbf{N}[X]$ )
```

Can also choose a variable to sample at random each time



# The Markov chain

With *Sprinkler = true*, *WetGrass = true*, there are four states:



Wander about for a while, average what you see

## MCMC example contd.

Estimate  $\mathbf{P}(Rain|Sprinkler = true, WetGrass = true)$

Sample *Cloudy* or *Rain* given its Markov blanket, repeat.  
Count number of times *Rain* is true and false in the samples.

E.g., visit 100 states

31 have *Rain = true*, 69 have *Rain = false*

$$\hat{\mathbf{P}}(Rain|Sprinkler = true, WetGrass = true) \\ = \text{NORMALIZE}(\langle 31, 69 \rangle) = \langle 0.31, 0.69 \rangle$$

Theorem: chain approaches **stationary distribution**:

long-run fraction of time spent in each state is exactly  
proportional to its posterior probability

## Summary

Exact inference by variable elimination:

- polytime on polytrees, NP-hard on general graphs
- space = time, very sensitive to topology

Approximate inference by LW, MCMC:

- LW does poorly when there is lots of (downstream) evidence
- LW, MCMC generally insensitive to topology
- Convergence can be very slow with probabilities close to 1 or 0
- Can handle arbitrary combinations of discrete and continuous variables