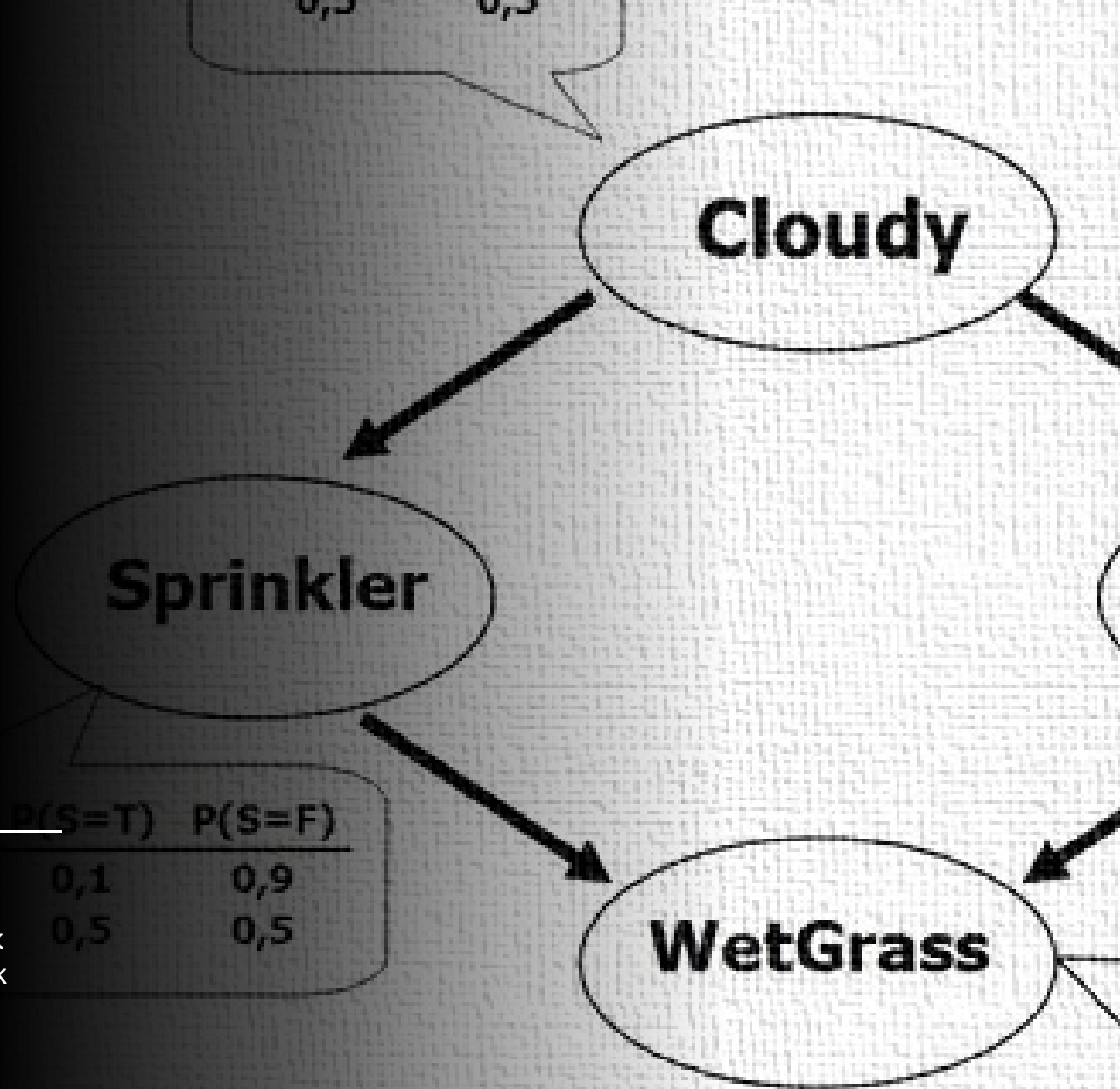


# CS 5/7320 Artificial Intelligence

## Probabilistic Reasoning AIMA Chapter 13

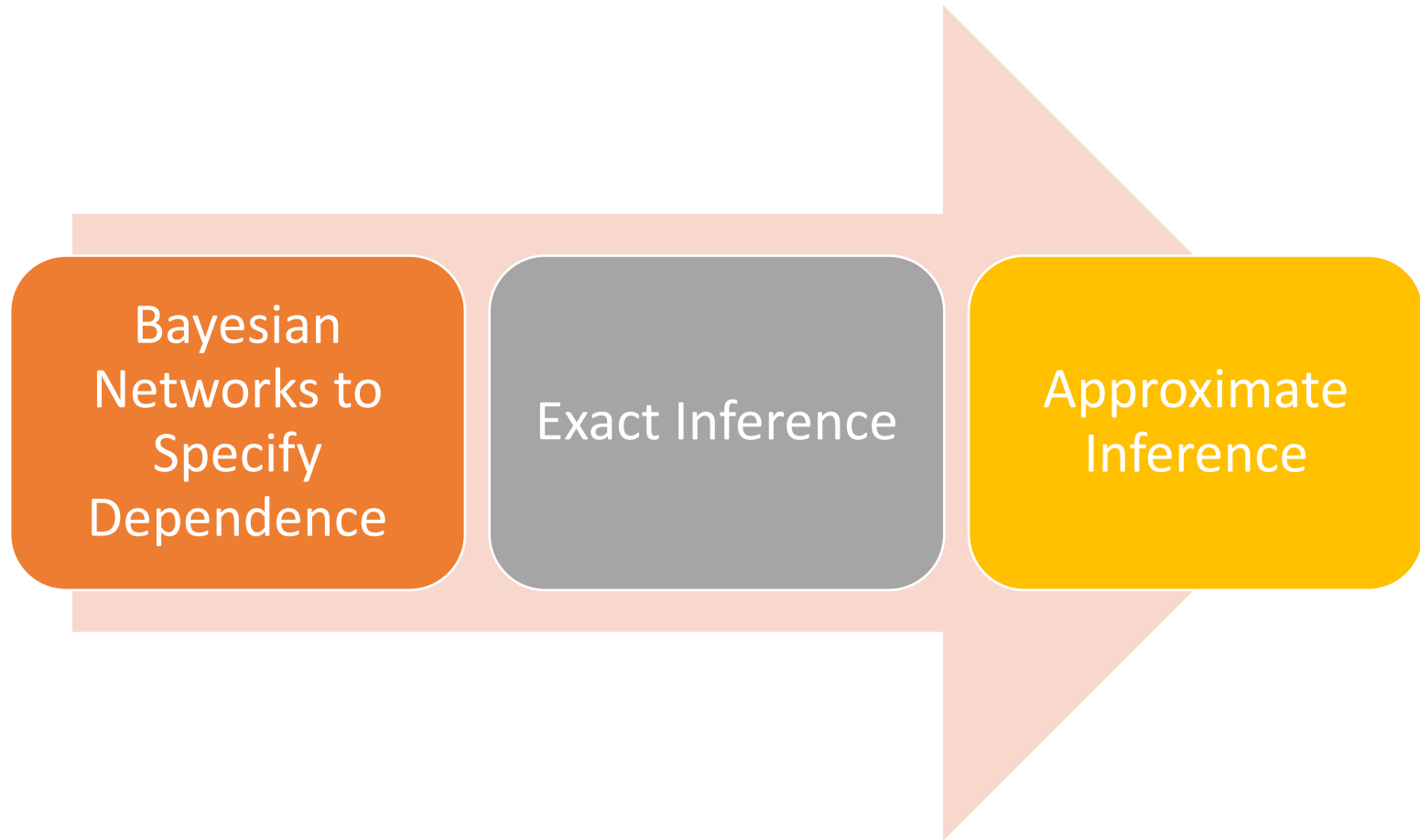
Slides by Michael Hahsler  
based on slides by Svetlana Lazepnik  
with figures from the AIMA textbook

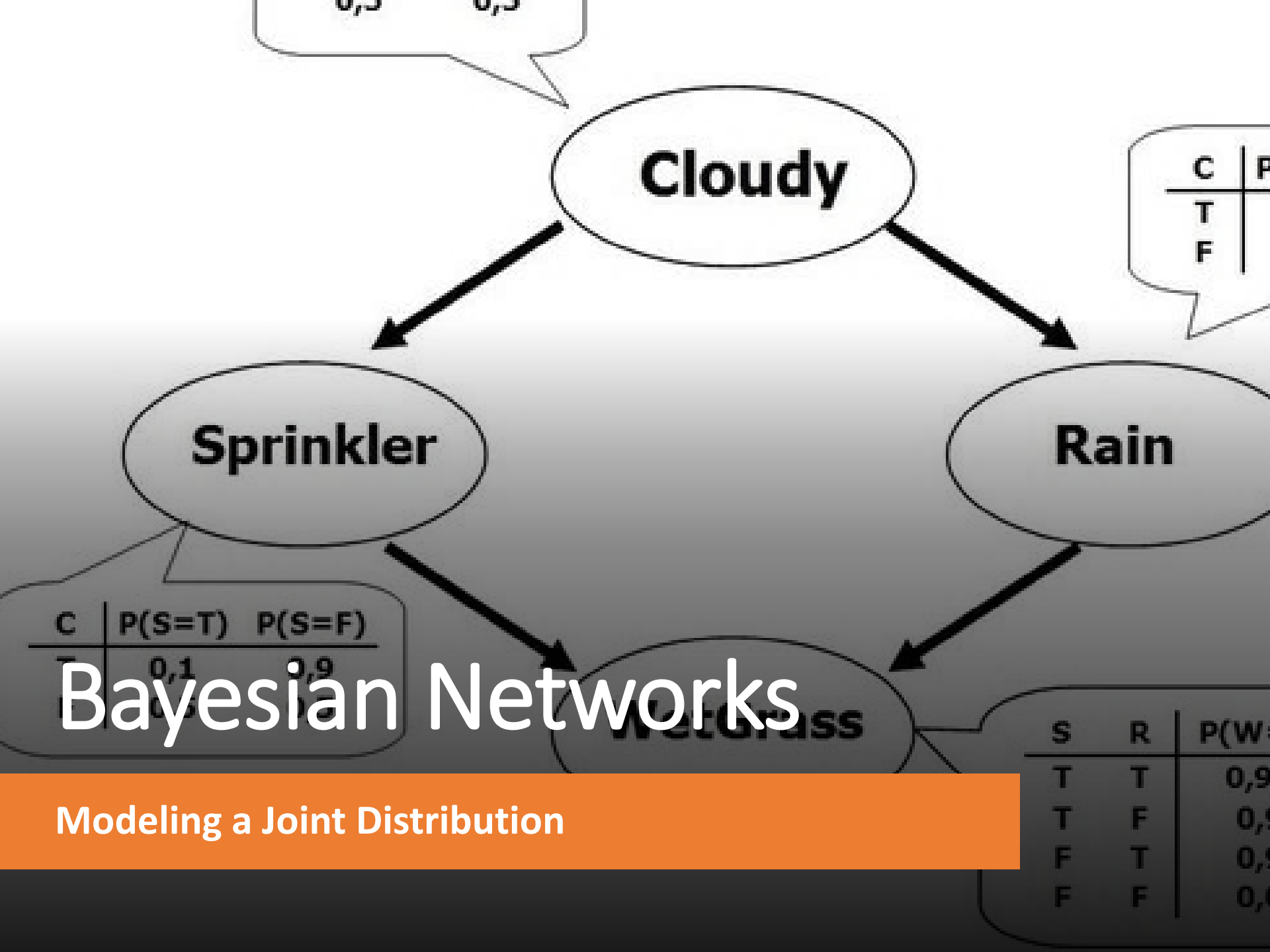


# Probability Theory Recap

- Notation: Prob. of an event  $P(X = x) = P(x)$   
Prob. distribution  $\mathbf{P}(X) = \langle P(X = x_1), P(X = x_2), \dots, P(X = x_n) \rangle$
- Product rule  $P(x, y) = P(x|y)P(y)$
- Chain rule 
$$\begin{aligned}\mathbf{P}(X_1, X_2, \dots, X_n) &= \mathbf{P}(X_1)\mathbf{P}(X_2|X_1)\mathbf{P}(X_3|X_1, X_2) \dots \\ &= \prod_{i=1}^n \mathbf{P}(X_i|X_1, \dots, X_{i-1})\end{aligned}$$
- Conditional probability  $P(x|y) = \frac{P(x,y)}{P(y)} = \alpha P(x, y)$
- Independence
  - $X \perp\!\!\!\perp Y$ :  $X, Y$  are independent (written as  $X \perp\!\!\!\perp Y$ ) if and only if:  
$$\forall x, y: P(x, y) = P(x)P(y)$$
  - $X \perp\!\!\!\perp Y|Z$ :  $X$  and  $Y$  are conditionally independent given  $Z$  if and only if:  
$$\forall x, y, z: P(x, y|z) = P(x|z)P(y|z)$$

# Contents

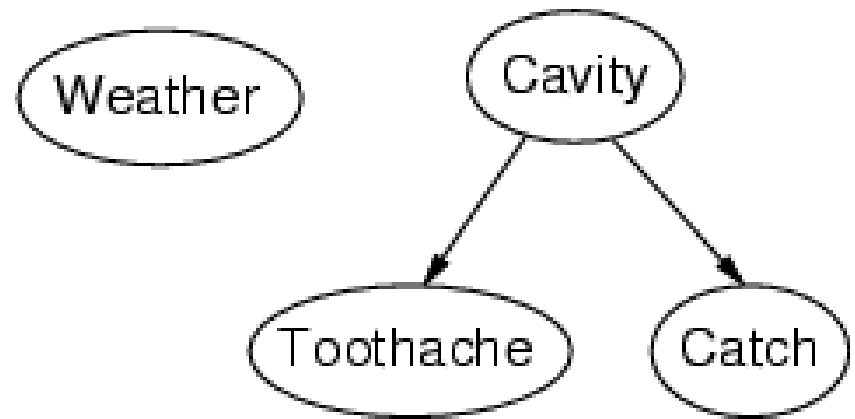




# Bayesian Networks

Modeling a Joint Distribution

# Bayesian Networks (aka Belief Networks)



A type of graphical model.



A way to specify dependence between random variables.



A compact specification of a full joint distributions.

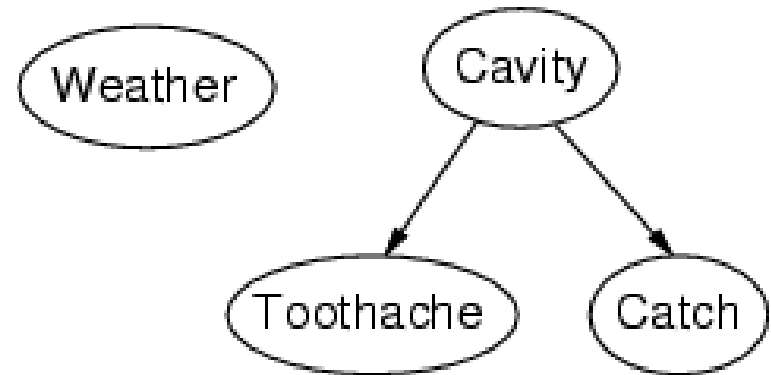


A general and important model to reason with uncertainty in AI.

# Structure of Bayesian Networks

## Nodes: Random variables

- Can be assigned (observed) or unassigned (unobserved)



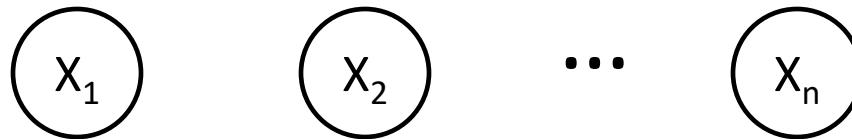
## Arcs: Dependencies

- An arrow from one variable to another indicates direct influence.
- Show independence
  - *Weather* is independent of the other variables (no connection).
  - *Toothache* and *Catch* are conditionally independent given *Cavity* (directed arc).
- Must form a directed *acyclic* graph (DAG)

A network with all random variables assigned represents a **state of the system**.

# Example: N independent coin flips

**Complete independence:** no interactions between coin flips



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

Joint probability  
distribution

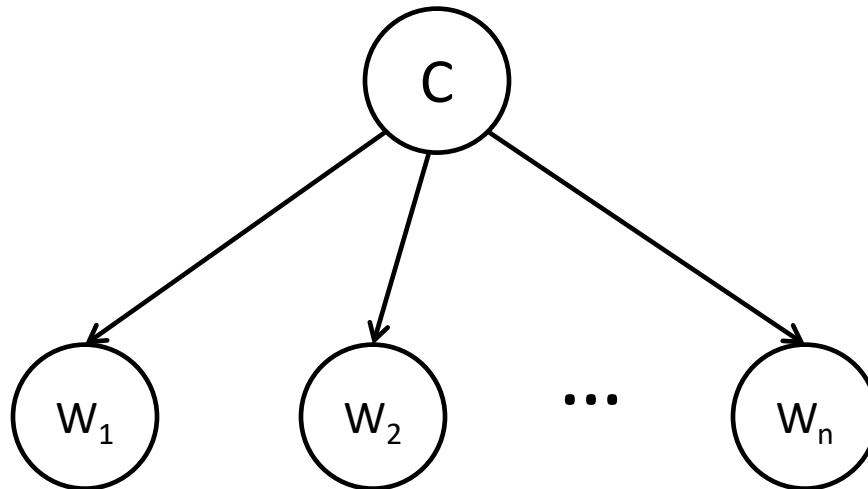
Marginal probability  
distributions

# Example: Naïve Bayes spam filter

Random variables:

- $C$ : message class (spam or not spam)
- $W_1, \dots, W_n$ : presence or absence of words comprising the message

**Words depend on the class, but they are modeled conditional independent of each other given the class (= no direct connection between words).**



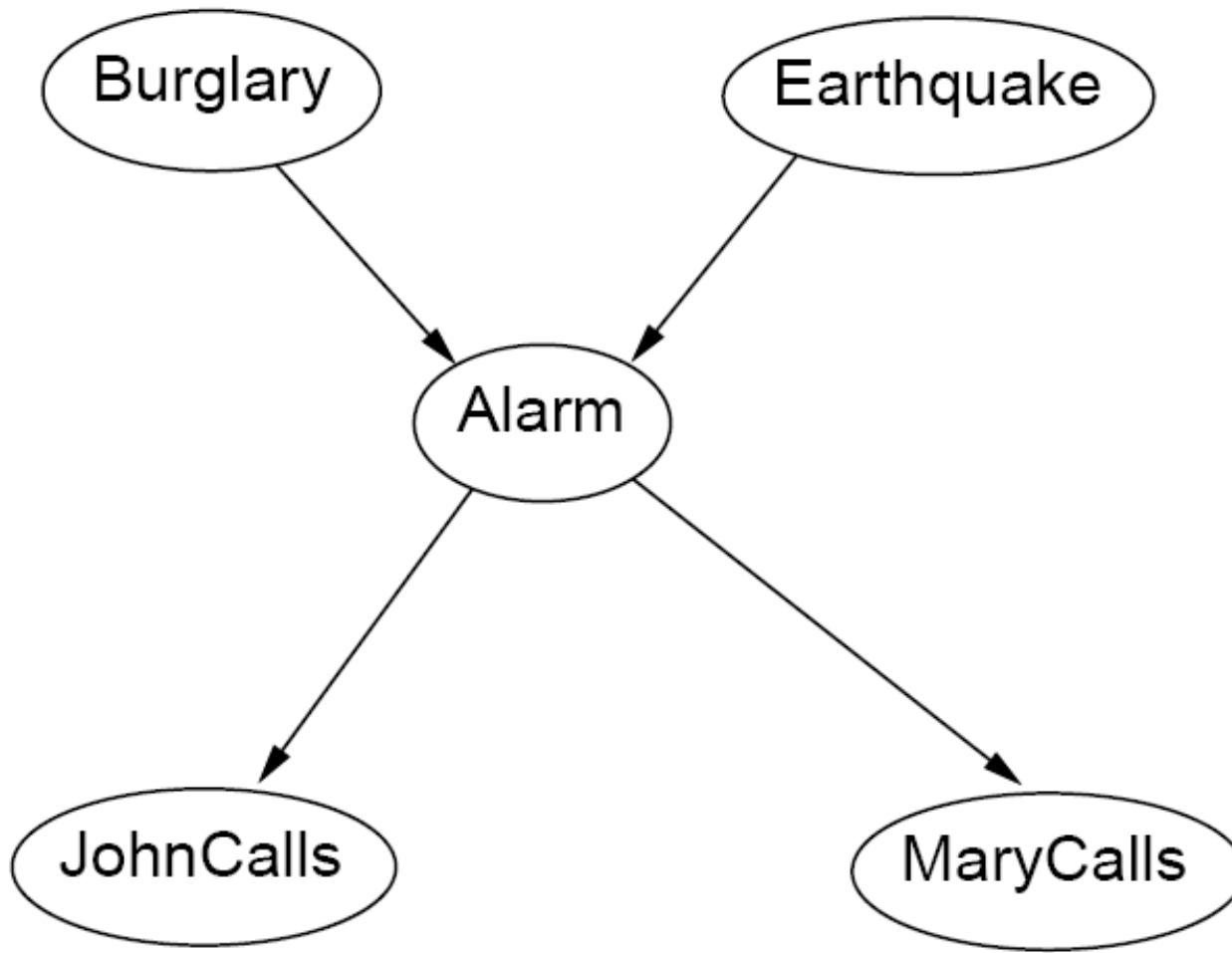
$$P(W_1, W_2, \dots, W_n | C) = P(W_1 | C) P(W_2 | C) \dots P(W_n | C)$$



# Example: Burglar Alarm

- **Description:** I have a burglar alarm that is sometimes set off by minor earthquakes. My two neighbors, John and Mary, promised to call me at work if they hear the alarm
- Example inference task: suppose Mary calls and John doesn't call. What is the probability of a burglary?
- What are the random variables?
  - Burglary, Earthquake, Alarm, JohnCalls, MaryCalls
- What are the direct influence relationships?
  - A burglar can set off the alarm
  - An earthquake can set off the alarm
  - The alarm can cause Mary to call
  - The alarm can cause John to call

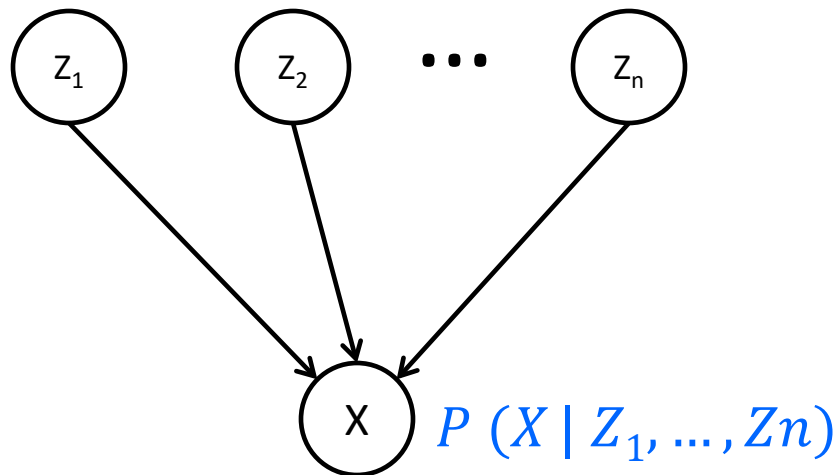
## Example: Burglar Alarm



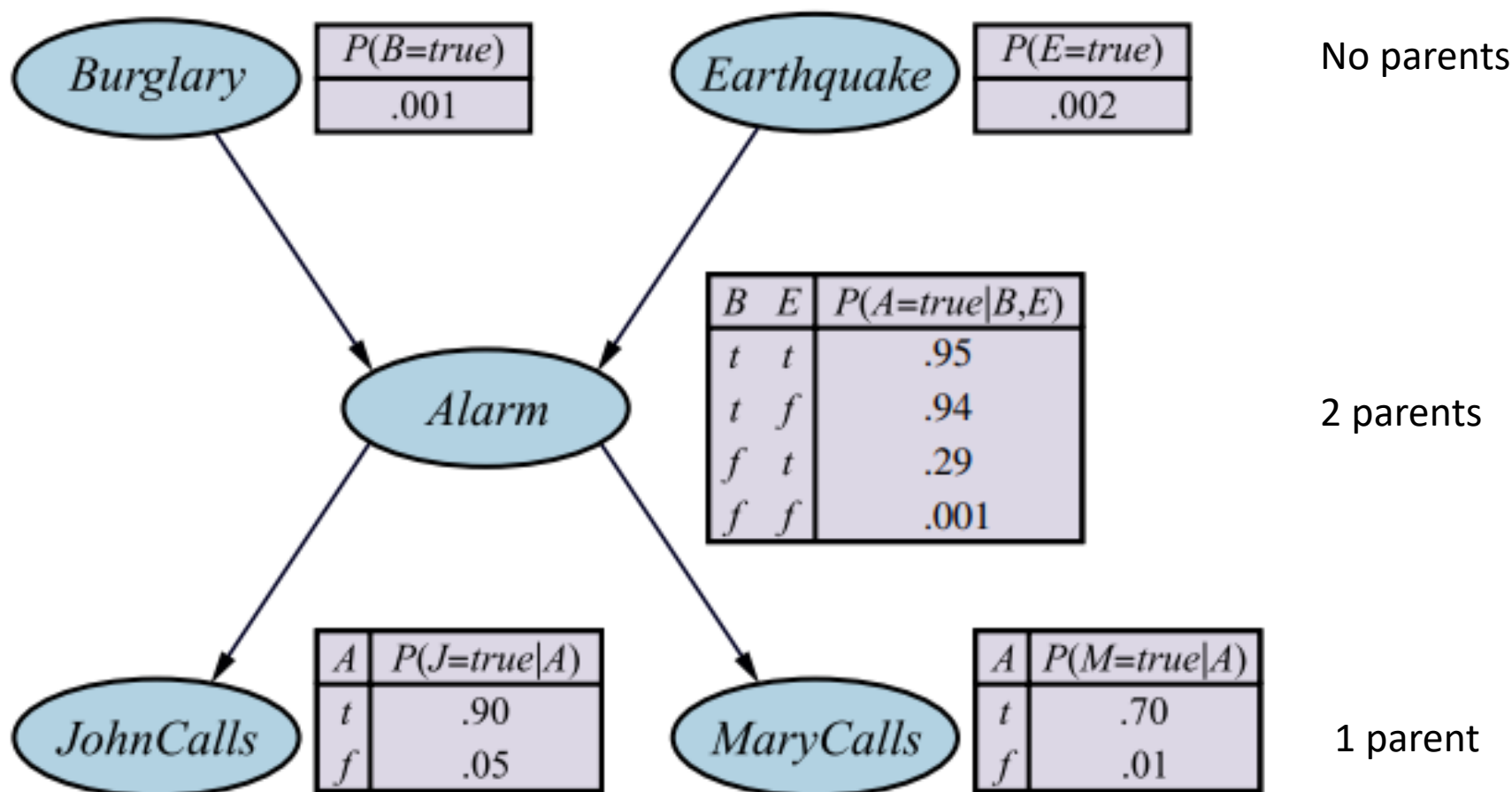
**What are the model parameters?**

# Parameters: Conditional probability tables

To specify the full joint distribution, we need to specify a *conditional* distribution for each node given its parents as a conditional probability table (CPT):  $P(X | \text{Parents}(X))$



## Example: Burglar Alarm with CPTs



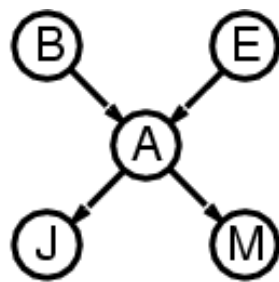
# The joint probability distribution

- For each node  $X_i$ , we know  $P(X_i \mid \text{Parents}(X_i))$
- How do we get the full joint distribution  $P(X_1, \dots, X_n)$ ?

- Using chain rule:

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

- Example:

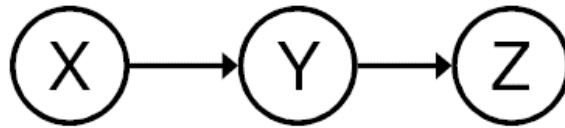


Construct  
following  
arrows

$$P(J, M, A, B, E) = P(B) P(E) P(A \mid B, E) P(J \mid A) P(M \mid A)$$

# Dependence

- Example: *causal chain*



X: Low pressure

Y: Rain

Z: Traffic

- Are X and Z independent?

$$P(X, Y, Z) = P(X)P(Y|X)P(Z|Y)$$

Conditioning

$$P(X, Z) = \sum_y P(X)P(y|X)P(Z|y)$$

Marginalize  
over Y

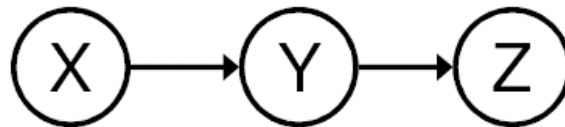
$$= P(X) \sum_y P(Z|y)P(y|X) \neq P(X)P(Z)$$



**X and Z are not independent!**

# Conditional independence

- Example: *causal chain*



X: Low pressure

Y: Rain

Z: Traffic

- Is Z independent of X given Y?

$$P(X, Z|Y) = \frac{P(X, Y, Z)}{P(Y)} = \frac{P(X)P(Y|X)P(Z|Y)}{P(Y)}$$

Conditioning

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

Bayes' rule

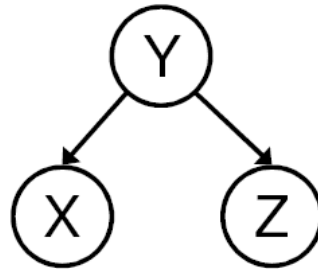
$$= P(X|Y)P(Z|Y) = \text{Definition of conditional independence}$$



X and Z are conditionally independent given Y

# Conditional independence

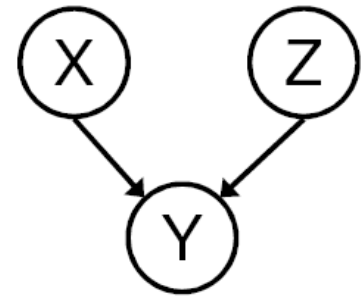
- Common cause



Y: Project due  
X: Newsgroup busy  
Z: Lab full

- Are X and Z independent?
  - No
- Are they conditionally independent given Y?
  - Yes

- Common effect

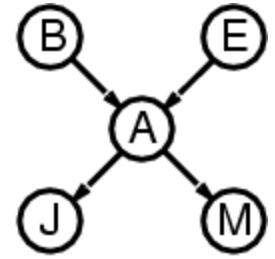


X: Raining  
Z: Ballgame  
Y: Traffic

- Are X and Z independent?
  - Yes
- Are they conditionally independent given Y?
  - No



# Compactness

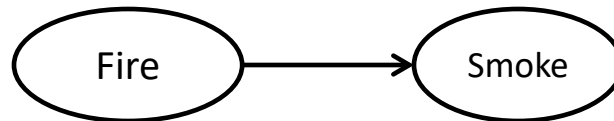


- Suppose we have a Boolean variable  $X_i$  with  $k$  Boolean parents. How many rows does its conditional probability table have?
  - $2^k$  rows for all the combinations of parent values, each row requires one number  $p$  for  $X_i = \text{true}$
- If each variable has no more than  $k$  parents, how many numbers does the complete network require?
  - $O(n \cdot 2^k)$  numbers – vs.  $O(2^n)$  for the full joint distribution
  - This reduces the complexity from exponential to linear in  $n$ !
- Example: How many nodes for the burglary network?  
 $1 + 1 + 4 + 2 + 2 = 10$  numbers  
(vs. specification of the complete joint probability  $2^5 - 1 = 31$ )

# Constructing Bayesian networks

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
$$P(X_i \mid \text{Parents}(X_i)) = P(X_i \mid X_1, \dots, X_{i-1})$$
that is, add an connection only from nodes it directly depends on.

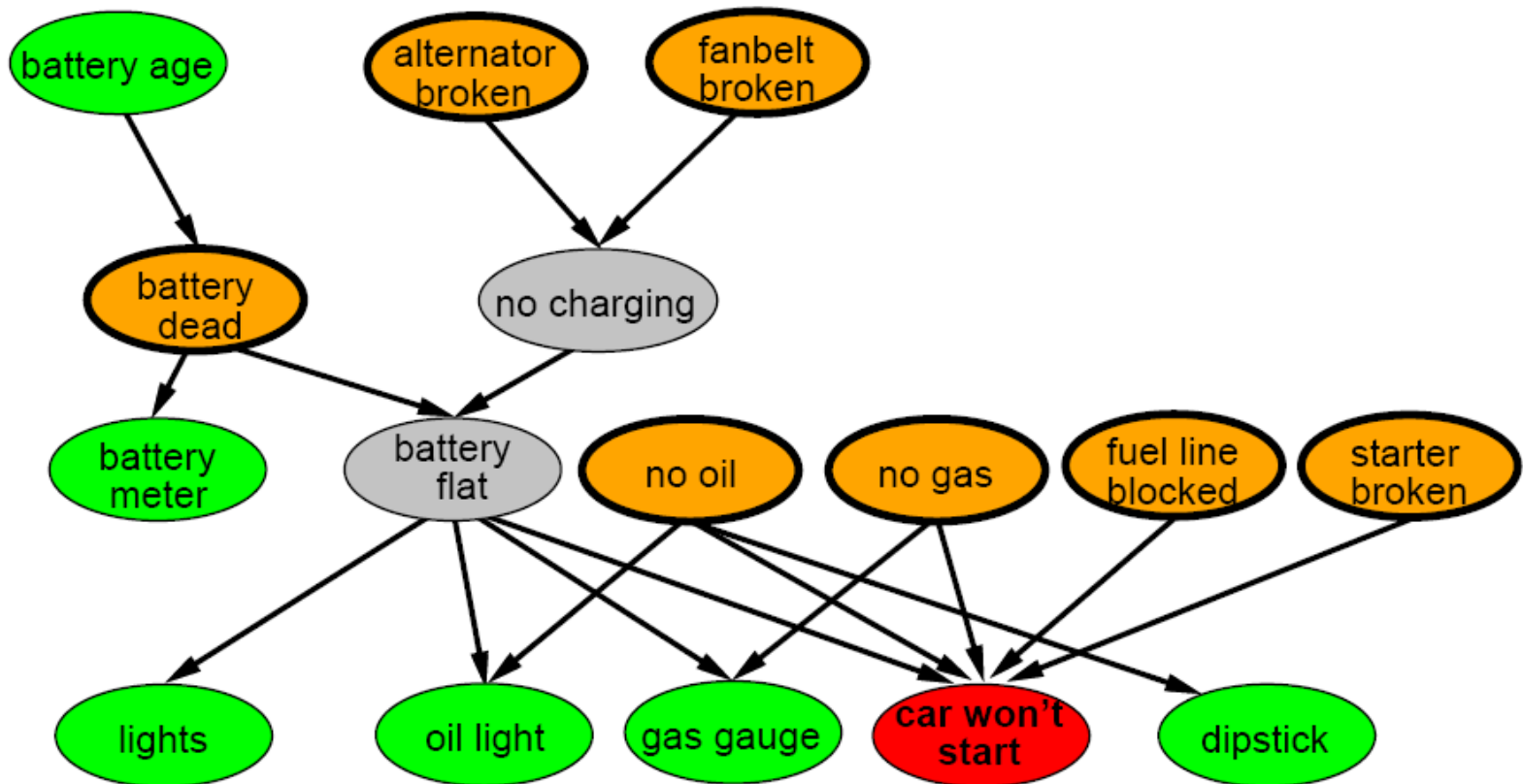
**Note:** There are many ways to order the variables. Networks are typically constructed by domain experts with causality in mind. E.g., Fire causes Smoke:



The resulting network is sparse and conditional probabilities are easier to judge because they represent causal relationships.

# A more realistic Bayes Network: Car diagnosis

- **Initial observation:** car won't start
- **Green:** testable evidence
- **Orange:** "broken, so fix it" nodes
- **Gray:** "hidden variables" to ensure sparse structure, reduce parameters

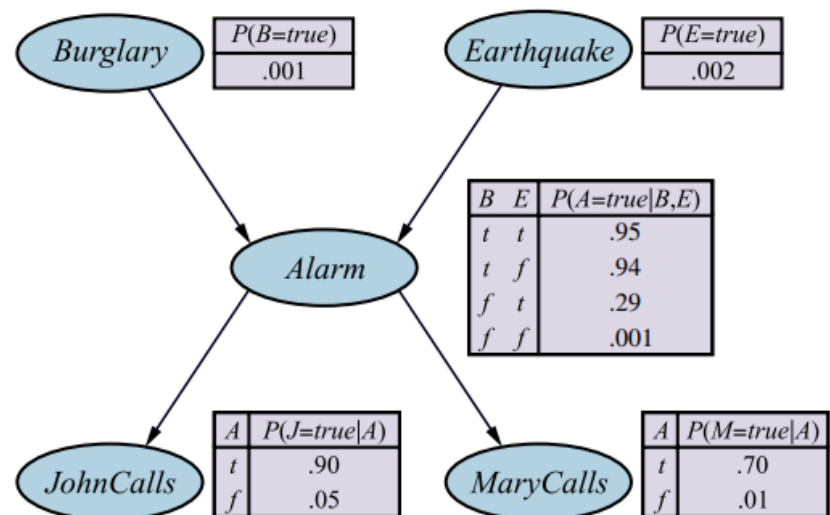


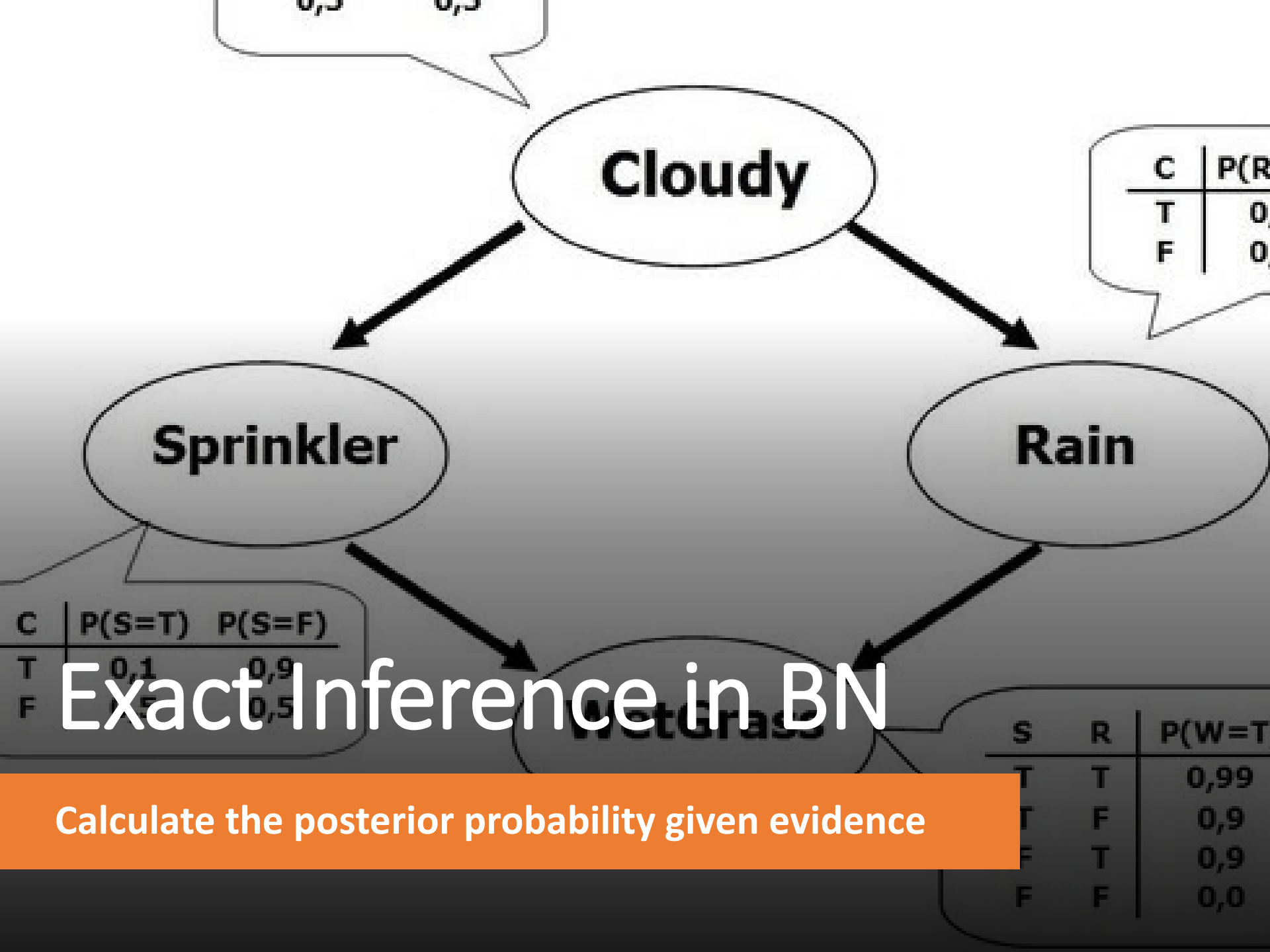
# Summary

- Bayesian networks provide a natural representation for joint probabilities used to calculate conditional probabilities used in inference.
- Conditional independence (induced by causality) reduces the number of needed parameters.

$P(B, E, A, J, M)$  is defined by

- Representation
  - Topology
  - Conditional probability tables
  - Generally easy for domain experts to construct





# Exact Inference in BN

Calculate the posterior probability given evidence

# Exact Inference

## Goal

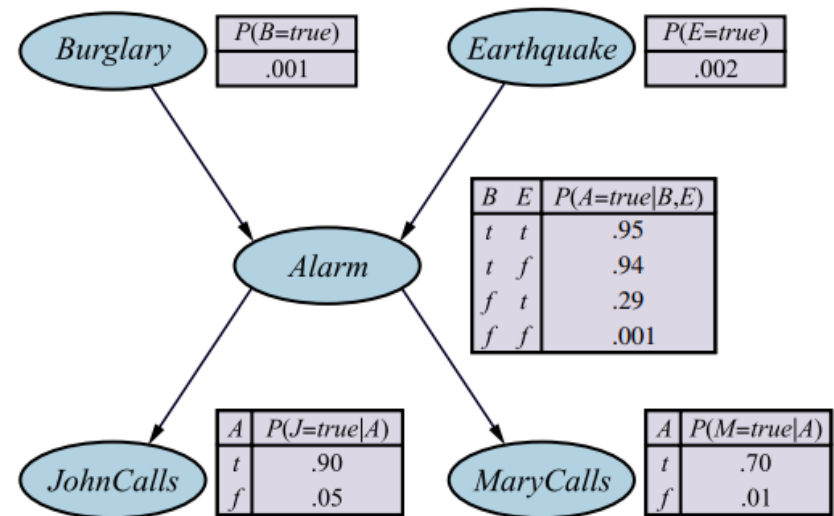
- Query variables:  $X$
- Evidence (observed) variables:  $E = e$
- Set of unobserved variables:  $Y$
- Calculate the probability of  $X$  given  $e$ .

If we know the full joint distribution  $P(X, E, Y)$ , we can infer  $X$  by:

$$P(X|E = e) = \frac{P(X, e)}{P(e)} \propto \sum_y P(X, e, y)$$

Sum over values of unobservable variables = marginalizing them out.

# Exact inference: Example



Assume we can observe being called and the two variables have the values  $j$  and  $m$ . We want to know the probability of a burglary.

**Query:**  $P(B | j, m)$  with unobservable variables: Earthquake, Alarm

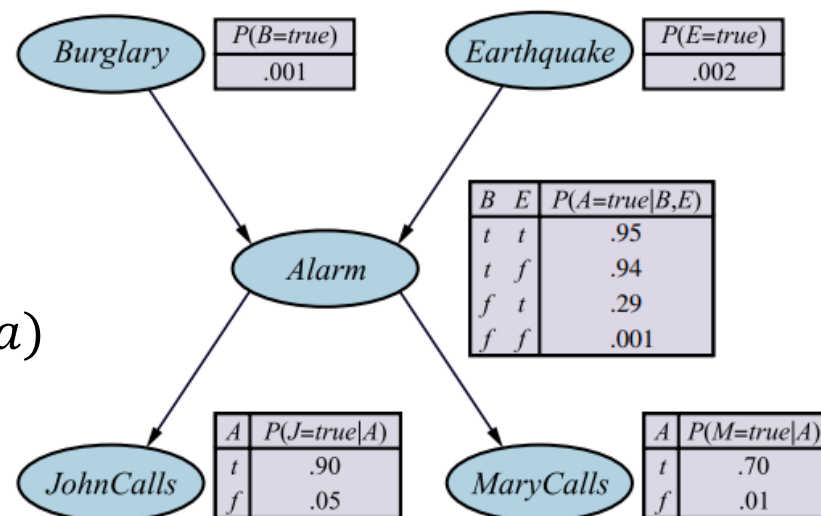
$$\begin{aligned}
 P(b|j, m) &= \frac{P(b, j, m)}{P(j, m)} \propto \sum_e \sum_a P(b, e, a, j, m) \\
 &= \sum_e \sum_a P(b)P(e)P(a|b, e)P(j|a)P(m|a) \\
 &= P(b) \sum_e P(e) \sum_a P(a|b, e) P(j|a)P(m|a)
 \end{aligned}$$

Full joint probability and marginalize over E and A

# Exact inference: Example

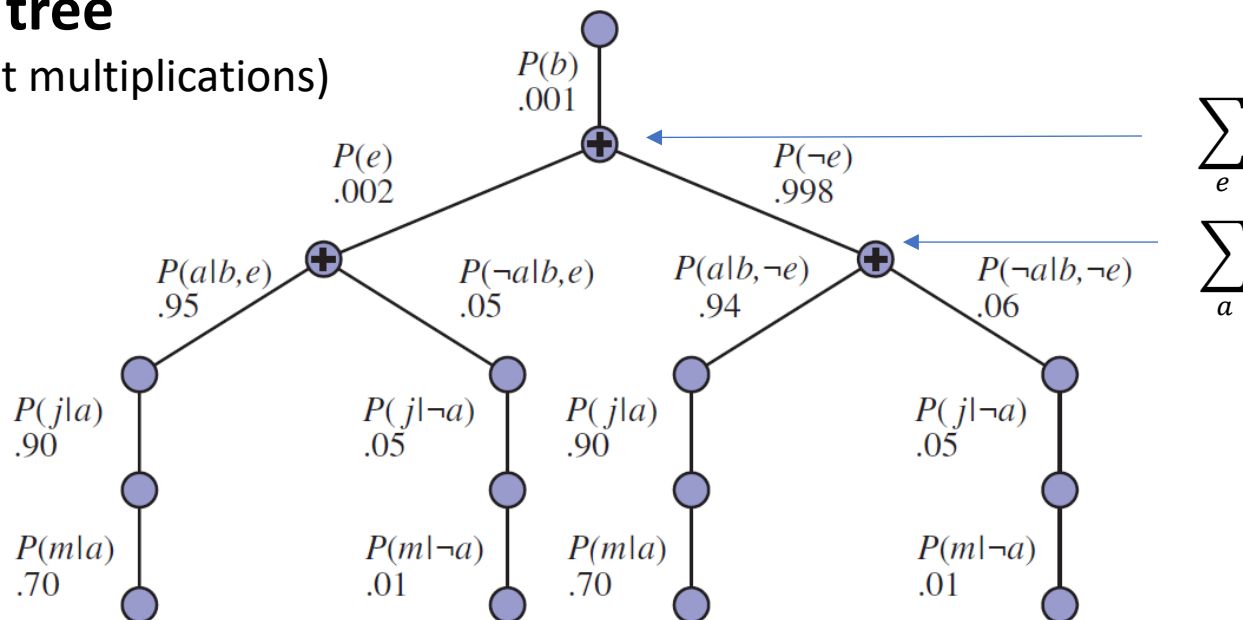
$$P(b|j, m)$$

$$\propto P(b) \sum_e P(e) \sum_a P(a|b, e) P(j|a) P(m|a)$$



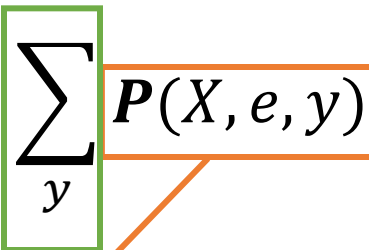
## Evaluation tree

(lines represent multiplications)





# Issues with Exact Inference in AI

$$P(X|E = e) = \frac{P(X, e)}{P(e)} \propto \sum_y P(X, e, y)$$


## Problems

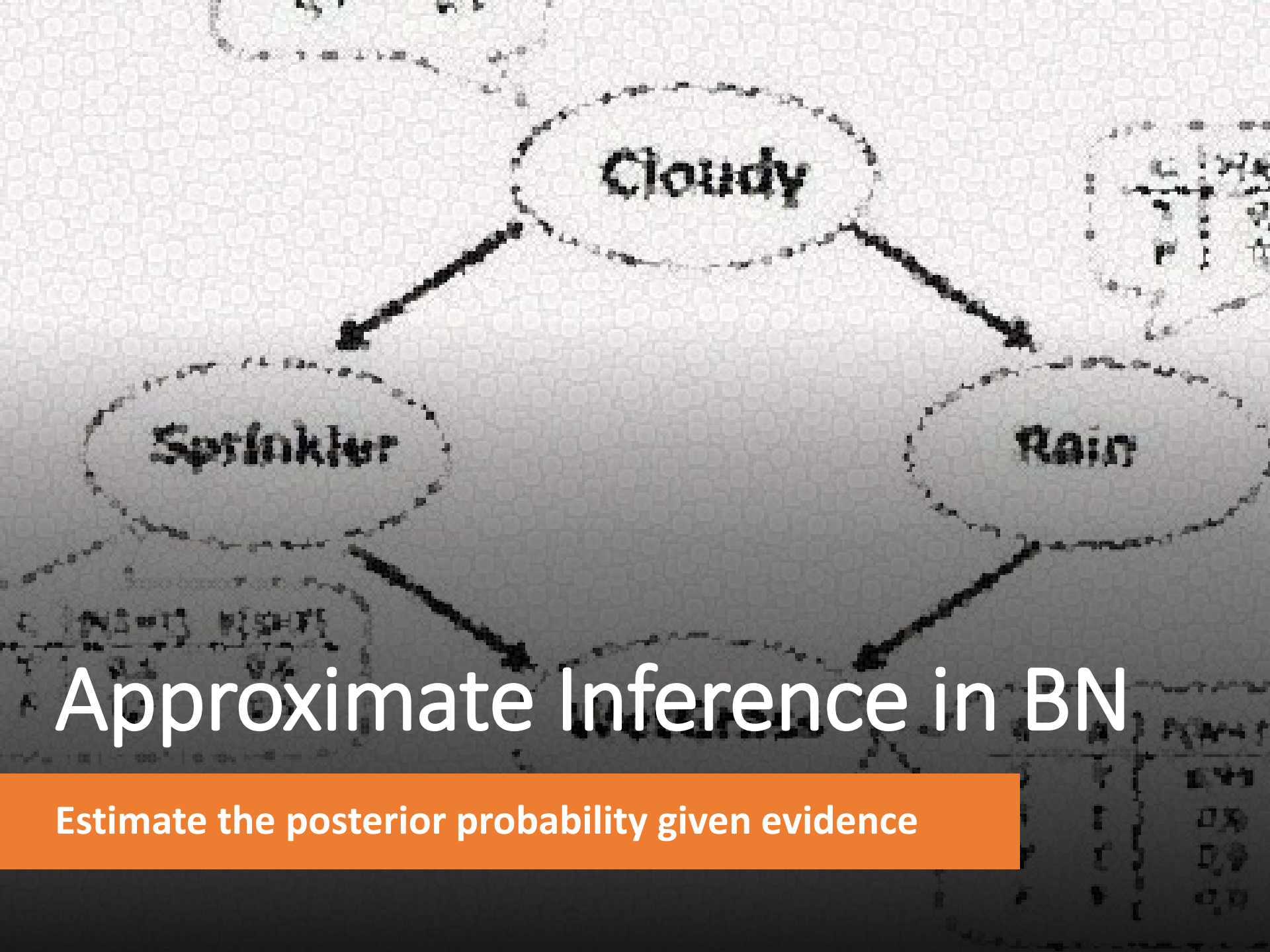
1. **Full joint distributions are too large** to store.

Bayes nets provide significant savings for representing the conditional probability structure.

2. Marginalizing out many unobservable variables  $Y$  may involve **too many summation terms**.

This summation is called **exact inference by enumeration**. Unfortunately, it does not scale well (#p-hard).

In praxis, **approximate inference by sampling** is used.



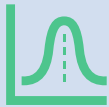
# Approximate Inference in BN

Estimate the posterior probability given evidence

# BN as a Generative Model



Bayesian networks can be used as ***generative models***.



Allows us to efficiently generate samples from the joint distribution.

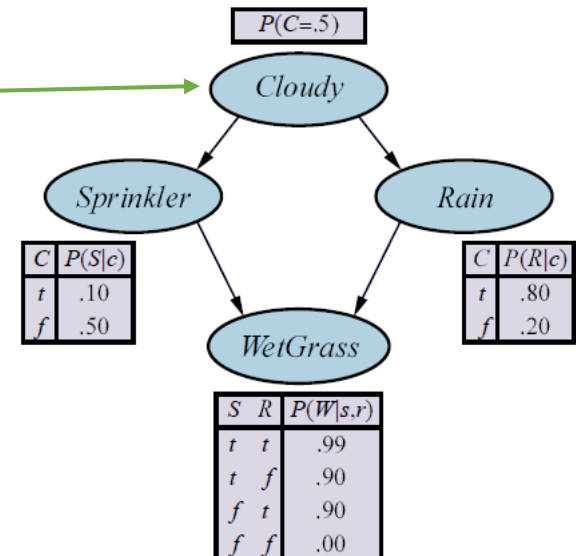


**Idea:** Generate samples from the network to estimate joint and conditional probability distributions.

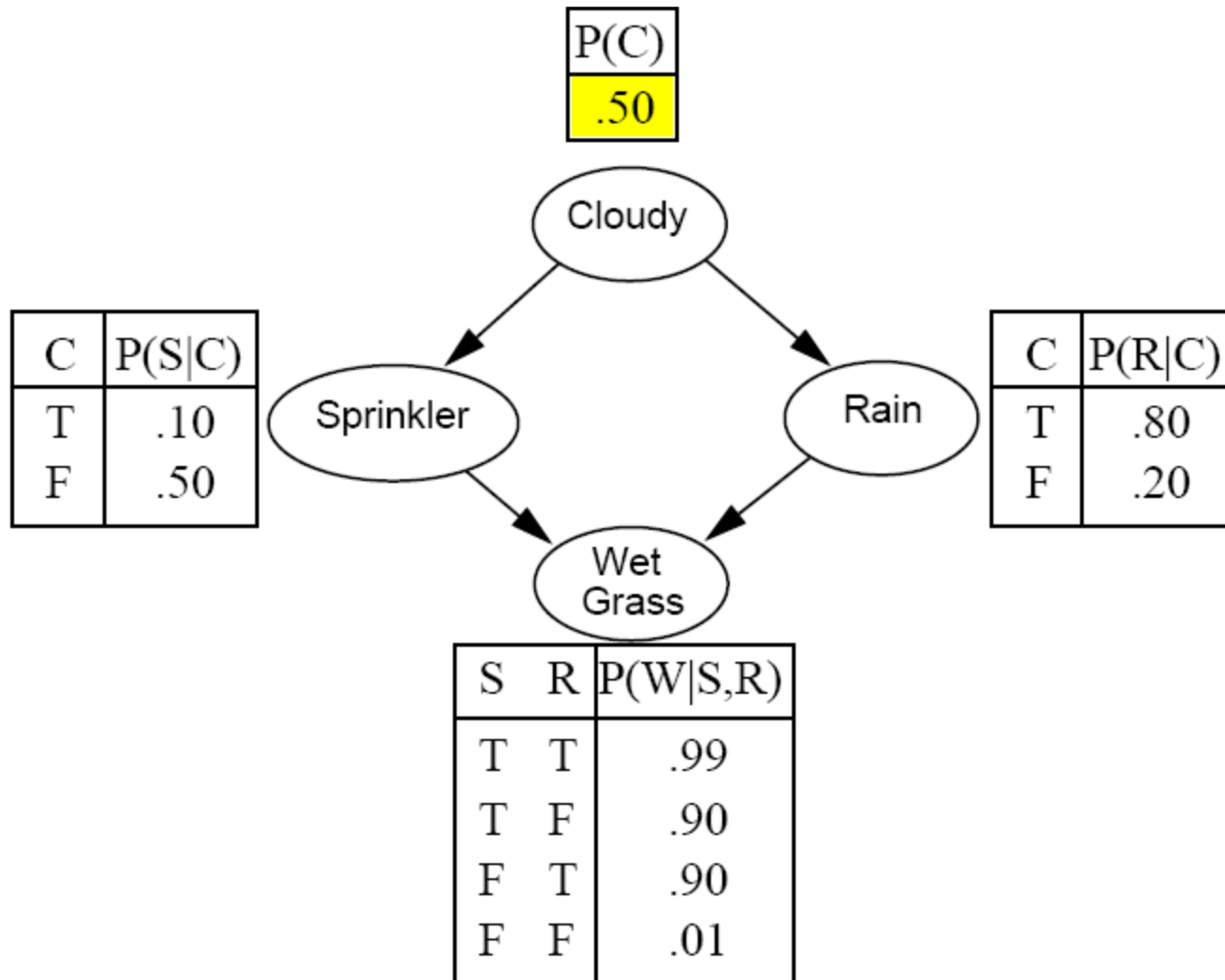
# Prior-Sample Algorithm to Create a Sample (Event)

```
function PRIOR-SAMPLE( $bn$ ) returns an event sampled from the prior specified by  $bn$   
inputs:  $bn$ , a Bayesian network specifying joint distribution  $\mathbf{P}(X_1, \dots, X_n)$   
  
 $\mathbf{x} \leftarrow$  an event with  $n$  elements  
for each variable  $X_i$  in  $X_1, \dots, X_n$  do  
     $\mathbf{x}[i] \leftarrow$  a random sample from  $\mathbf{P}(X_i \mid \text{parents}(X_i))$   
return  $\mathbf{x}$ 
```

We need to start with the random variables that have no parents.

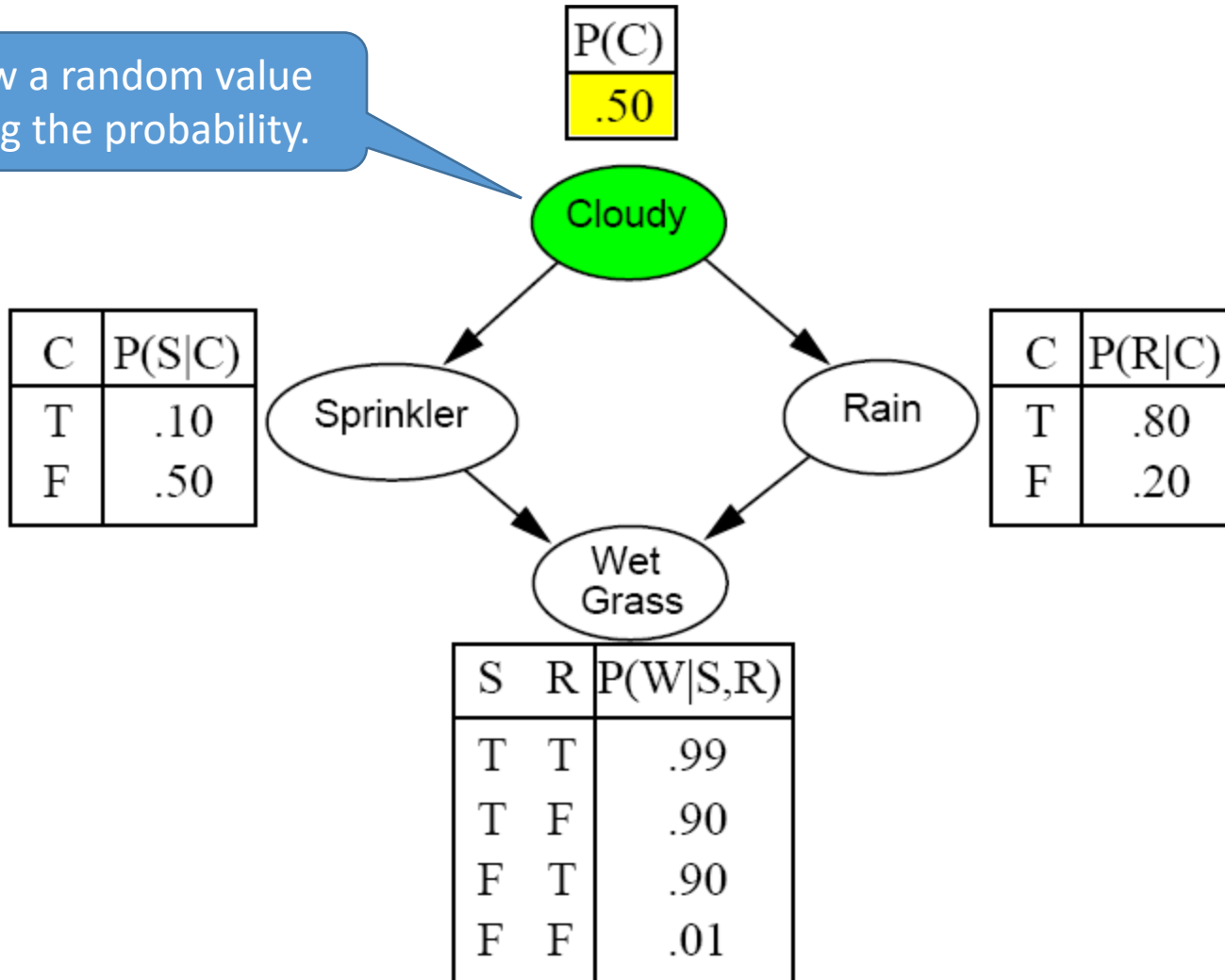


# Example: Sampling from a Bayesian Network

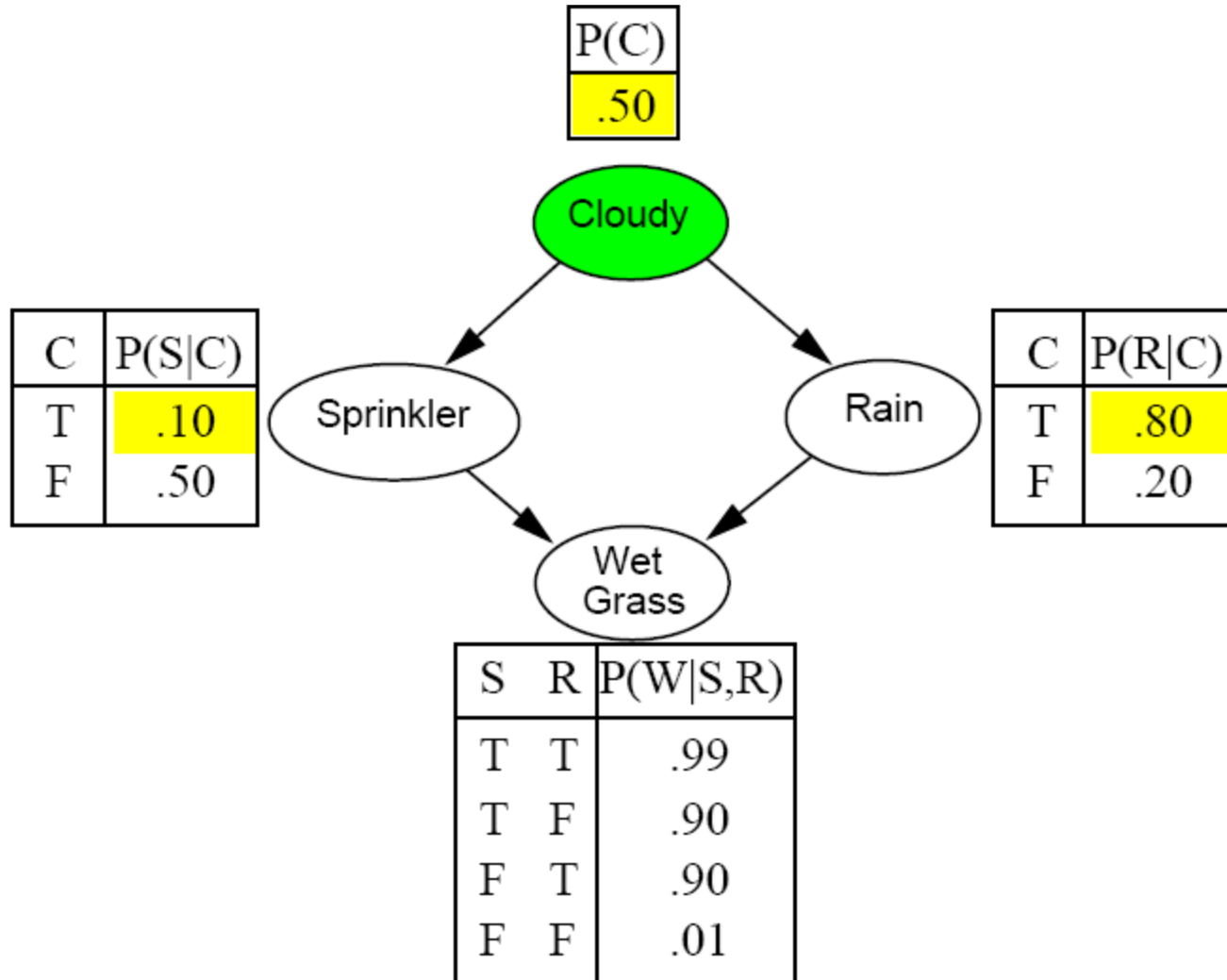


# Example: Sampling from a Bayesian Network

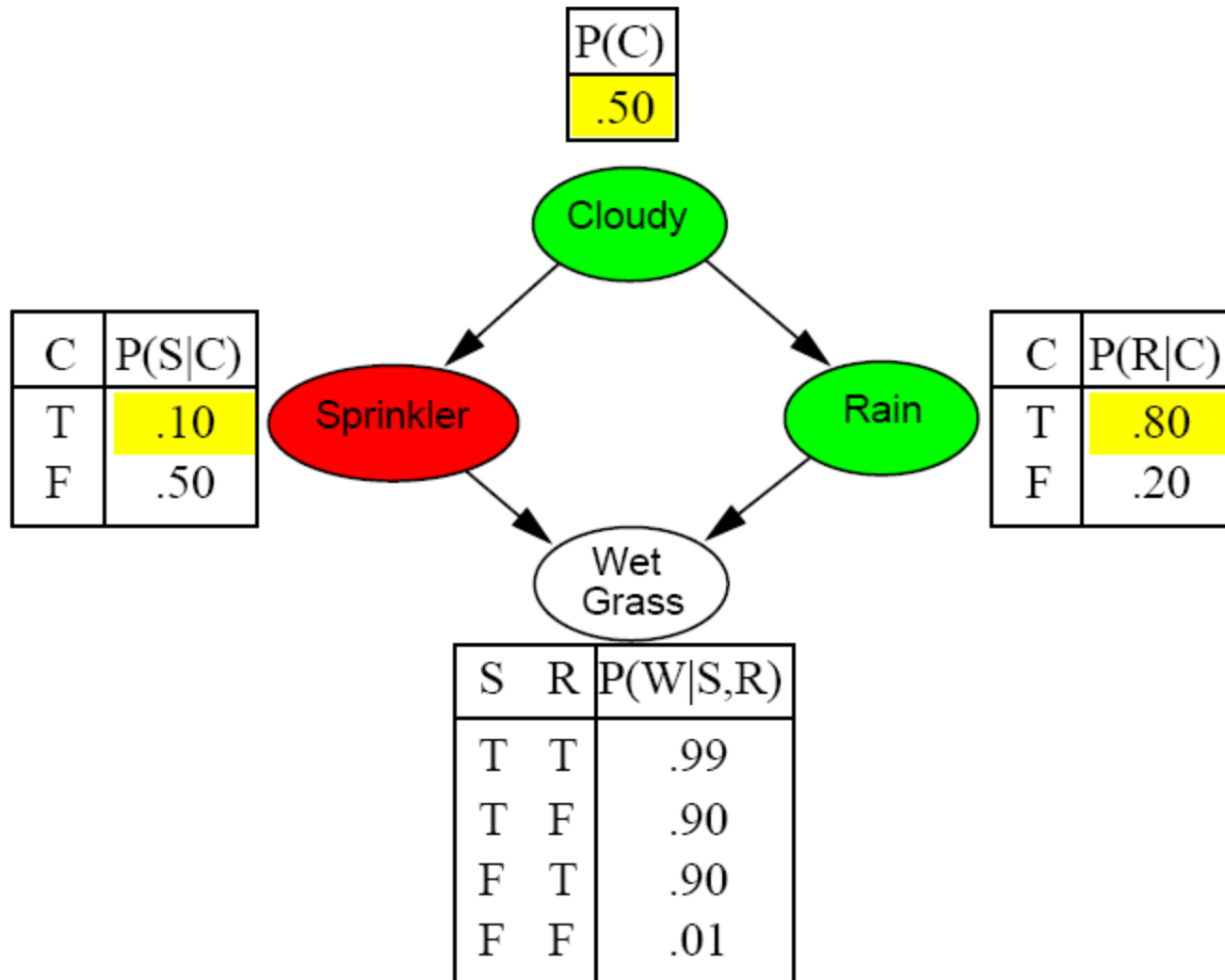
Draw a random value using the probability.



# Example: Sampling from a Bayesian Network

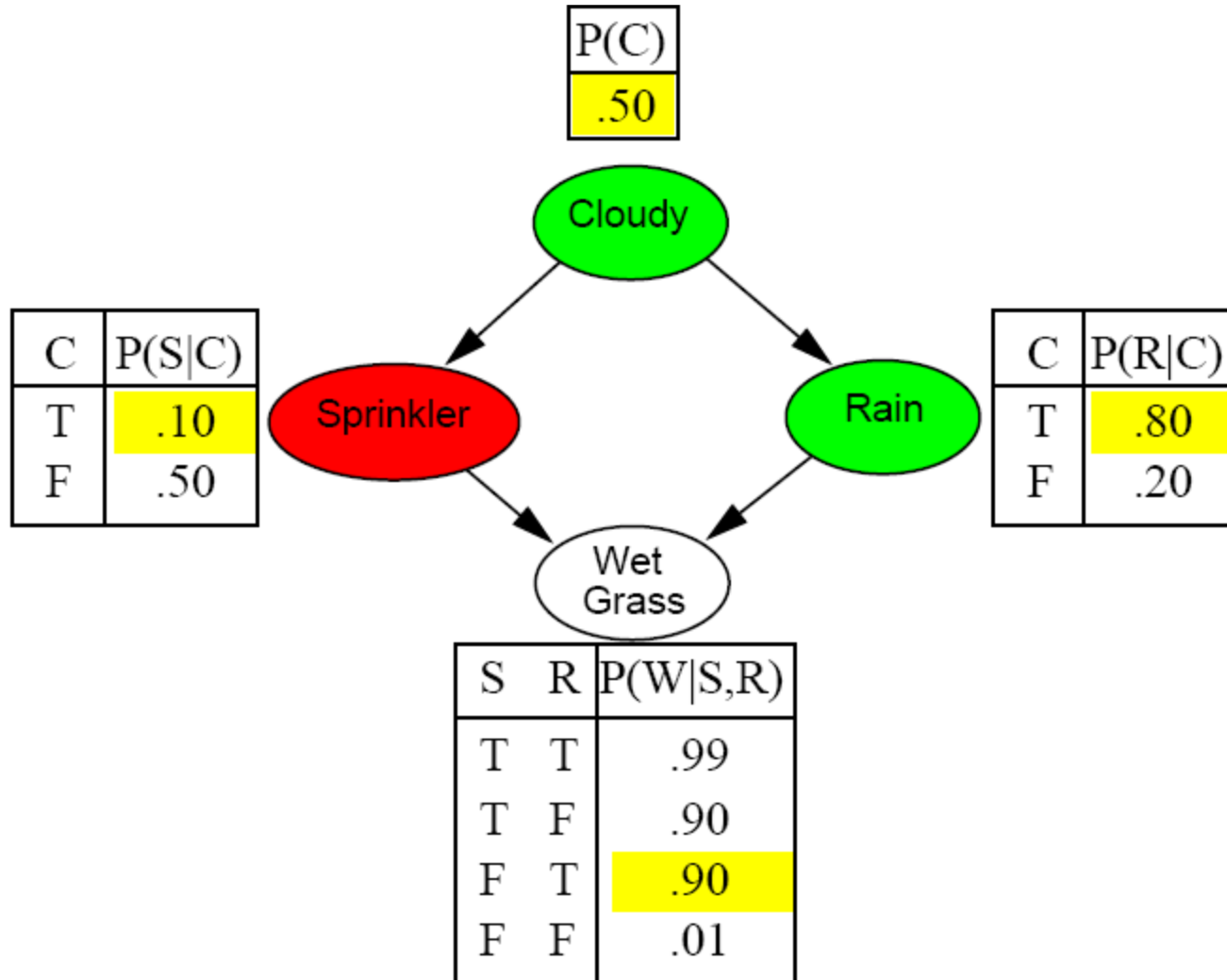


# Example: Sampling from a Bayesian Network

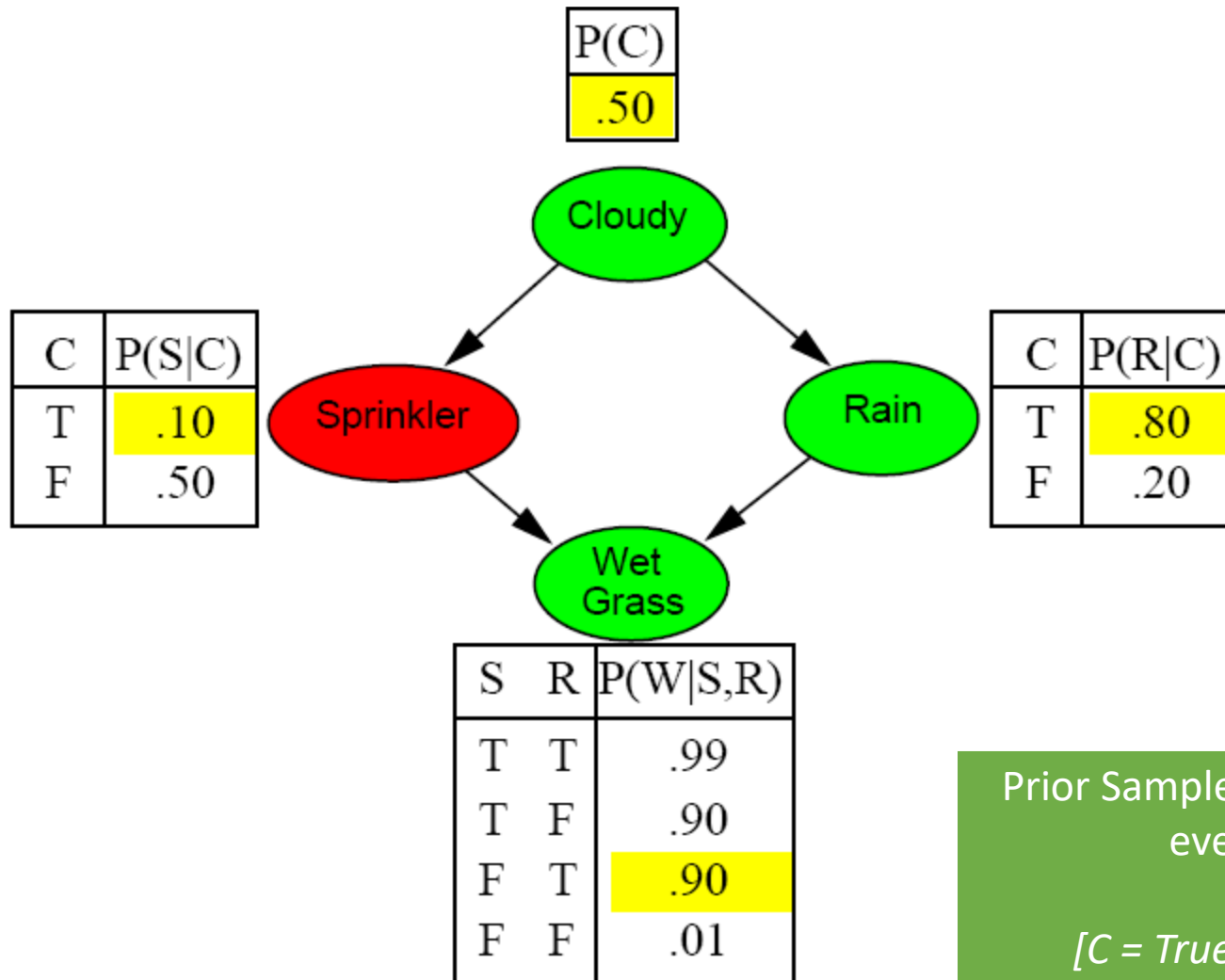




# Example: Sampling from a Bayesian Network



# Example: Sampling from a Bayesian Network



Prior Sample returns the event:

*[C = True, S = False,  
R = True, W = True]*

# Estimating the Joint Probability Distribution

Sample  $N$  times and determine  $N_{PS}(x_1, x_2, \dots, x_n)$ , the count of how many times Prior-Sample produces event  $(x_1, x_2, \dots, x_n)$ .

$$\hat{P}(x_1, x_2, \dots, x_n) = \frac{N_{PS}(x_1, x_2, \dots, x_n)}{N}$$

The marginal probability of partially specified event (some  $x$  values are known) can also be calculated. E.g.,

$$\hat{P}(x_1) = \frac{N_{PS}(x_1)}{N}$$

# Estimating Conditional Probabilities:

## Rejection sampling

Sample  $N$  times and **ignore the samples that are not consistent with the evidence  $e$ .**

$$\hat{P}(X|e) = \alpha N_{PS}(X, e) = \frac{N_{PS}(X, e)}{N_{PS}(e)}$$

**Issue:** What if  $e$  is a rare event?

- Example: burglary  $\wedge$  earthquake
- Rejection sampling ends up throwing away most of the samples. This is very inefficient!

# Estimating Conditional Probabilities:

## Rejection sampling

**function** REJECTION-SAMPLING( $X, \mathbf{e}, bn, N$ ) **returns** an estimate of  $\mathbf{P}(X \mid \mathbf{e})$   
**inputs:**  $X$ , the query variable  
           $\mathbf{e}$ , observed values for variables  $\mathbf{E}$   
           $bn$ , a Bayesian network  
           $N$ , the total number of samples to be generated  
**local variables:**  $\mathbf{C}$ , a vector of counts for each value of  $X$ , initially zero

**for**  $j = 1$  **to**  $N$  **do**  
     $\mathbf{x} \leftarrow \text{PRIOR-SAMPLE}(bn)$   
    **if**  $\mathbf{x}$  is consistent with  $\mathbf{e}$  **then**  
         $\mathbf{C}[j] \leftarrow \mathbf{C}[j] + 1$  where  $x_j$  is the value of  $X$  in  $\mathbf{x}$   
**return** NORMALIZE( $\mathbf{C}$ )

We throw away many samples  
if  $\mathbf{e}$  is rare!

# Estimating Conditional Probabilities: Importance sampling (likelihood weighting)

Goal: Avoid the need of rejection sampling to throw out samples.

**1. Fix the evidence  $E = e$**  for sampling and estimate the probability for the non-evidence variables using prior-sampling.

$$Q_{WS}(x)$$

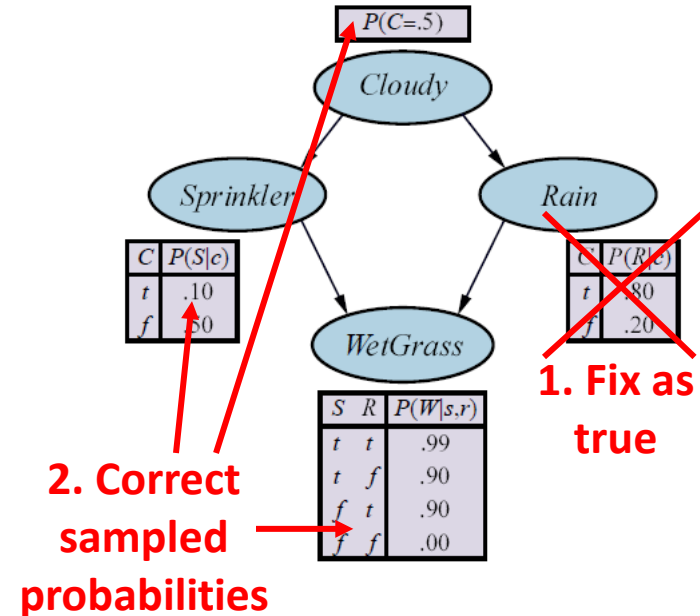
**2. Correct the probabilities** using weights

$$P(x|e) = w(x)Q_{WS}(x)$$

Turns out the weights in this case can be easily calculated

$$w(x) = \frac{1}{P(e)} \prod_{i=1}^m P(e_i | \text{parents}(E_i))$$

Example: Evidence = it rains



# Estimating Conditional Probabilities: Markov Chain Monte Carlo Sampling (MCMC)

- **Generates a sequence of samples** instead of creating each sample individually from scratch.
- Create a state by making random changes to the current state. The sequence of states forms a random process called a **Markov Chain** (MC).
- The MCs stationary distribution turns out to be the posterior distribution of the non-evidence variables.
- Estimate the stationary distribution using **Monte Carlo** simulation by counting how often each state is reached and normalize to obtain probability estimates.
- Algorithms:
  1. Gibbs sampling (works well for BNs)
  2. Metropolis-Hastings sampling

*Note: Simulated annealing belongs to the family of MCMC algorithms.*

# Gibbs sampling in Bayes Networks

**function** GIBBS-ASK( $X, \mathbf{e}, bn, N$ ) **returns** an estimate of  $\mathbf{P}(X | \mathbf{e})$

**local variables:**  $\mathbf{C}$ , a vector of counts for each value of  $X$ , initially zero

$\mathbf{Z}$ , the nonevidence variables in  $bn$

$\mathbf{x}$ , the current state of the network, initialized from  $\mathbf{e}$

initialize  $\mathbf{x}$  with random values for the variables in  $\mathbf{Z}$

**for**  $k = 1$  **to**  $N$  **do**

**choose** any variable  $Z_i$  from  $\mathbf{Z}$  according to any distribution  $\rho(i)$

    set the value of  $Z_i$  in  $\mathbf{x}$  by sampling from  $\mathbf{P}(Z_i | mb(Z_i))$

$\mathbf{C}[j] \leftarrow \mathbf{C}[j] + 1$  where  $x_j$  is the value of  $X$  in  $\mathbf{x}$

**return** NORMALIZE( $\mathbf{C}$ )

Random  
State

Change one  
variable

Count

- $mb(Z_i)$  is the Markov blanket of random variable  $Z_i$  (all variables it can be dependent of, i.e., parents, children and parents of children).

$$P(z_i | mb(Z_i)) = \alpha P(z_i | parents(Z_i)) \prod_{Y_j \in children(X_i)} P(y_j | parents(Y_j))$$



# Gibbs Sampling: Example

Find

$$P(\text{Rain} \mid \text{Sprinkler} = \text{true}, \text{WetGrass} = \text{true}).$$

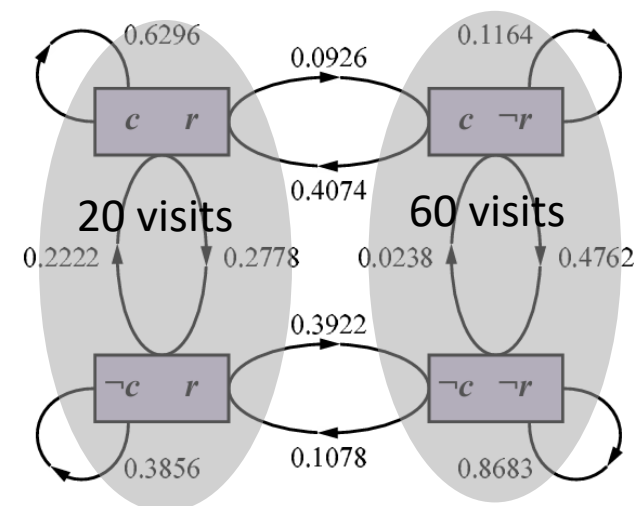
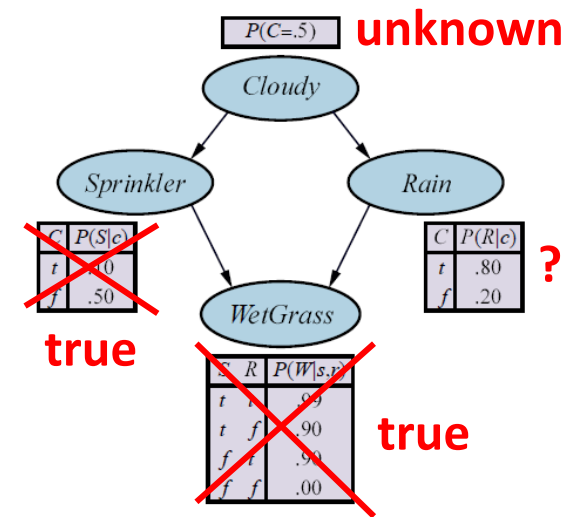
Determine states and calculate transition probabilities of the Markov chain for changing one variable using  $P(z_i \mid mb(Z_i))$ .

The algorithm randomly wanders around in this graph using the stated transition probabilities.

Assume that we observe 20 states with  $\text{Rain} = \text{true}$  and 60 with  $\text{rain} = \text{false}$ :

$$\text{NORMALIZE}(\langle 20, 60 \rangle) = \langle 0.25, 0.75 \rangle$$

$$P(\text{Rain} \mid \text{Sprinkler} = \text{true}, \text{WetGrass} = \text{true}) \approx 0.75$$



Note the self-loops: the state stays the same when either variable is chosen and then resamples the same value it already has.



## Conclusion

---

- Bayesian networks provide an efficient way to store a complete probabilistic model by exploiting (conditional) independence between variables.
- Inference means querying the model for a conditional probability given some evidence.
- Exact inference is difficult, for all but tiny models.
- State of the art is to use approximate inference by sampling from the model.
- Software libraries provide general inference engines.