CSC242: Intro to AI

Lecture 16 Bayesian Networks II

Learning Bayesian Networks from Data

Kinds of Learning Problems

- Learning the structure of the graph
- Learning the numbers in the conditional probability tables (aka “parameter learning”)

Kinds of Data

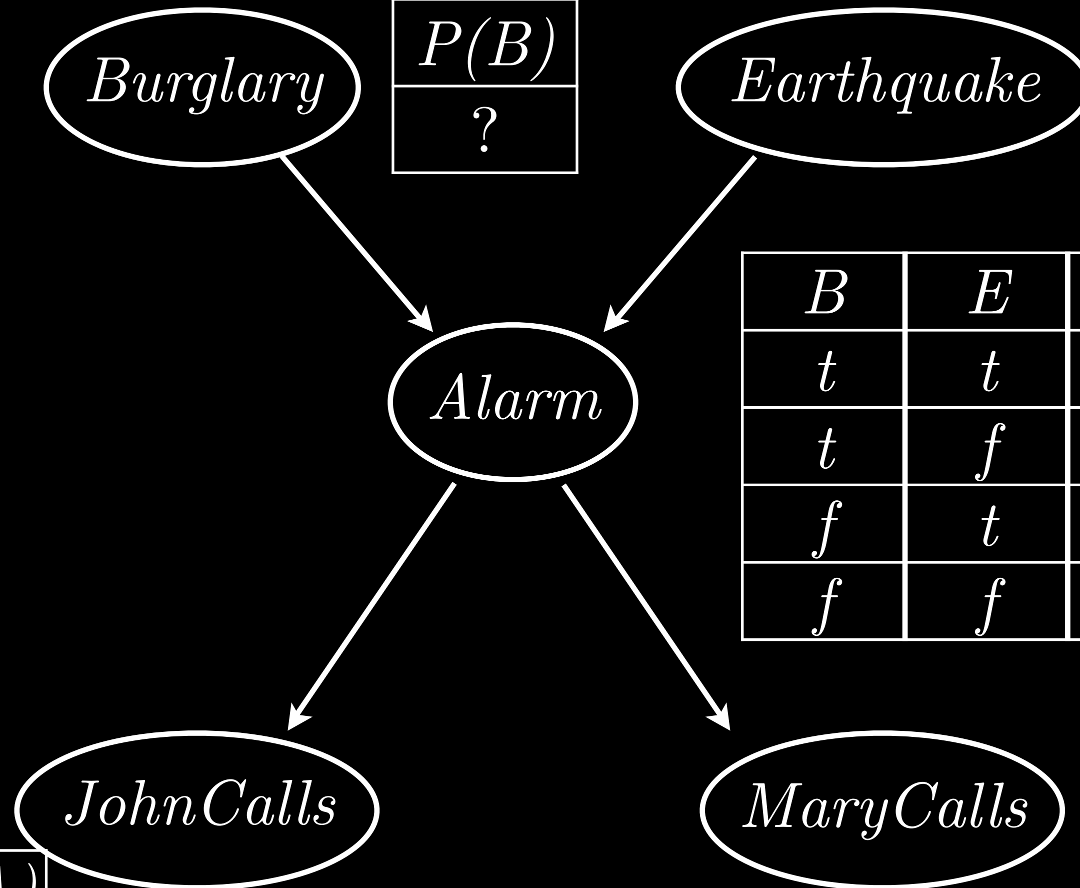
- Each piece of data is a sample of some of the random variables
- Each piece of data is a sample of all of the random variables (aka "complete data")

Easiest Case

- Learning the numbers in the conditional probability tables (aka "parameter learning")
- Each piece of data is a sample of all of the random variables (aka "complete data")

Parameter Learning from Complete Data

- Parameter values for a variable given its parents are the observed frequencies
- Learning = Counting!



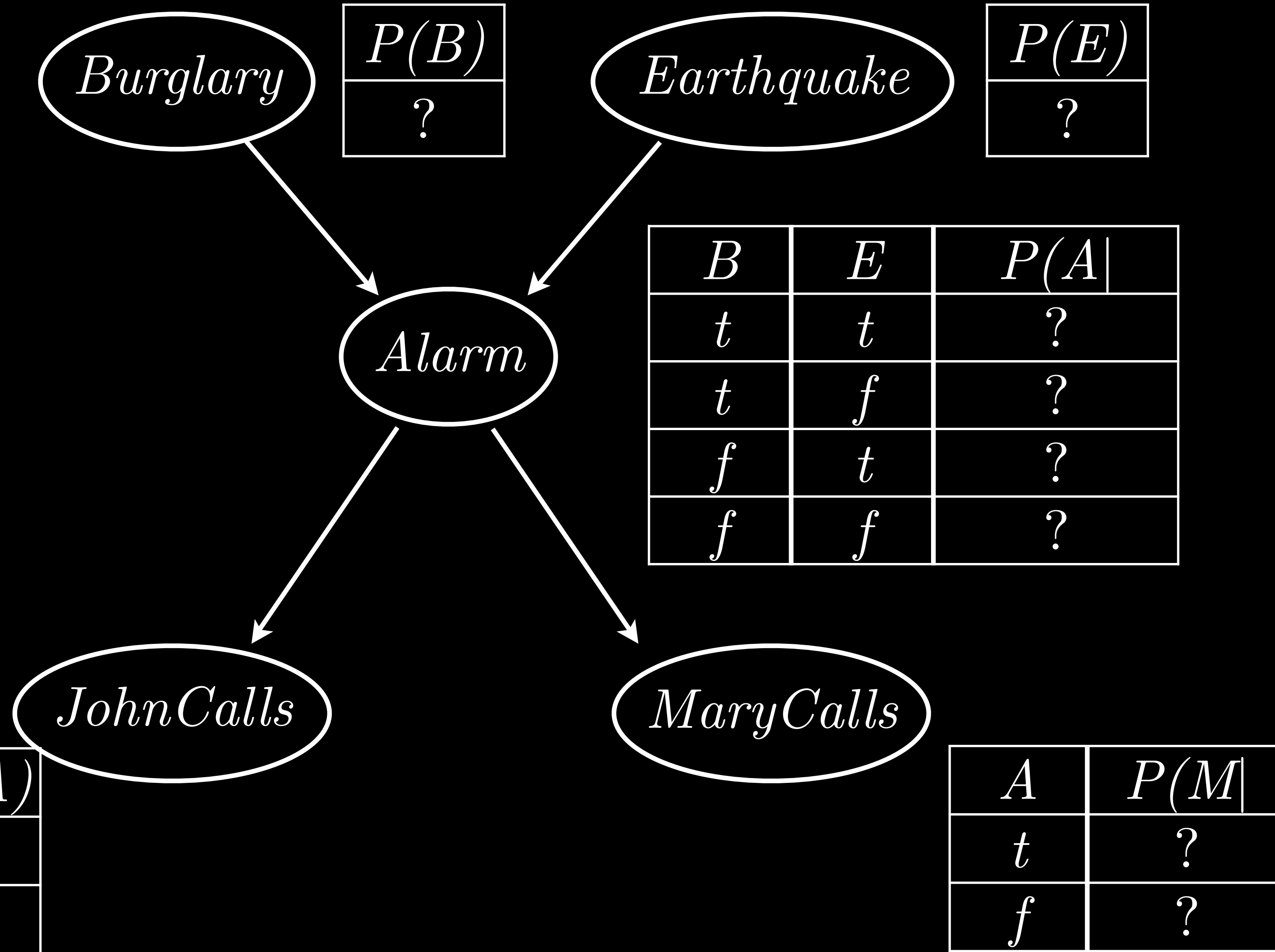
$P(B)$
?

$P(E)$
?

B	E	$P(A\|$
t	t	?
t	f	?
f	t	?
f	f	?

A	$P(J\|A)$
t	?
f	?

A	$P(M\|$
t	?
f	?



$P(B)$
?

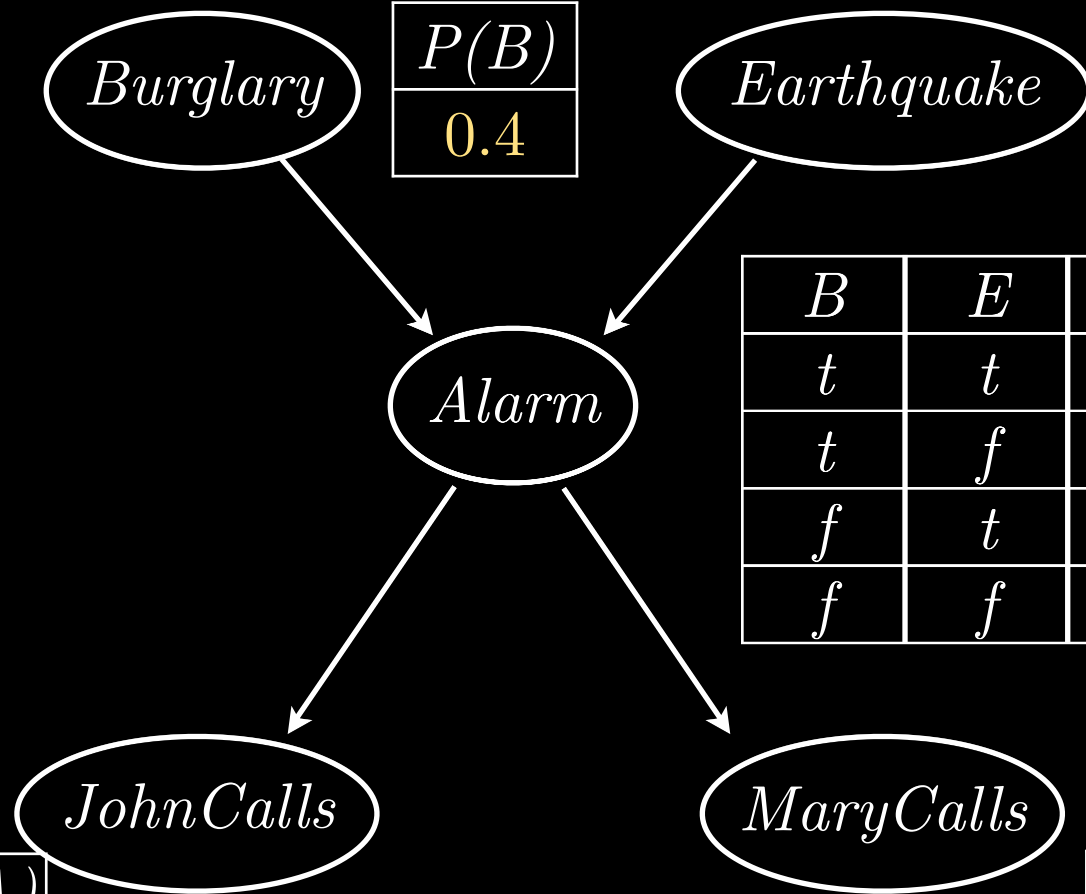
$P(E)$
?

B	E	$P(A\|$
t	t	?
t	f	?
f	t	?
f	f	?

A	$P(J\|A)$
t	?
f	?

A	$P(M\|$
t	?
f	?

Burglary	*Earthquake*	*Alarm*	*JohnCalls*	*MaryCalls*
T	*T*	*F*	*T*	*F*
F	*F*	*T*	*F*	*T*
F	*T*	*T*	*T*	*T*
F	*F*	*F*	*F*	*F*
T	*T*	*T*	*T*	*T*



$P(B)$
0.4

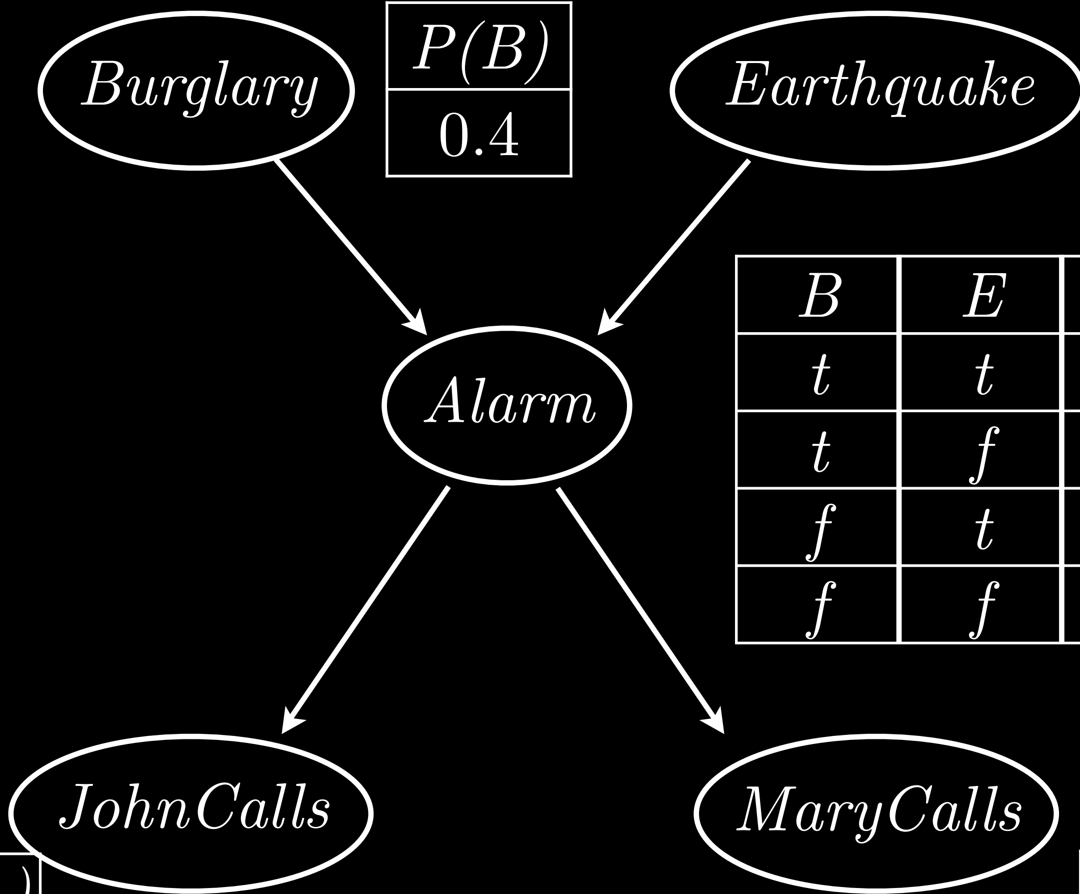
$P(E)$
?

B	E	$P(A\|$
t	t	?
t	f	?
f	t	?
f	f	?

A	$P(J\|A)$
t	?
f	?

A	$P(M\|$
t	?
f	?

Burglary	*Earthquake*	*Alarm*	*JohnCalls*	*MaryCalls*
T	*T*	*F*	*T*	*F*
F	*F*	*T*	*F*	*T*
F	*T*	*T*	*T*	*T*
F	*F*	*F*	*F*	*F*
T	*T*	*T*	*T*	*T*



$P(B)$
0.4

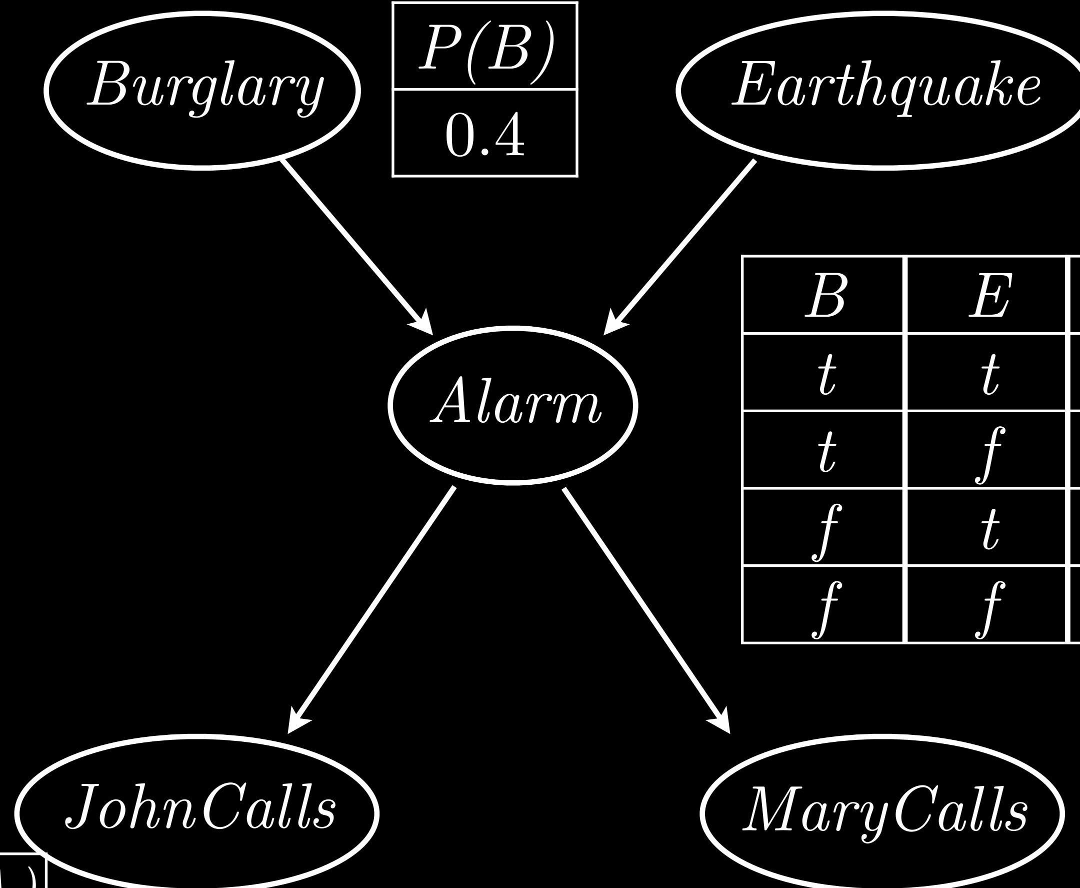
$P(E)$
0.6

B	E	$P(A\|$
t	t	?
t	f	?
f	t	?
f	f	?

A	$P(J\|A)$
t	?
f	?

A	$P(M\|$
t	?
f	?

Burglary	*Earthquake*	*Alarm*	*JohnCalls*	*MaryCalls*
T	*T*	*F*	*T*	*F*
F	*F*	*T*	*F*	*T*
F	*T*	*T*	*T*	*T*
F	*F*	*F*	*F*	*F*
T	*T*	*T*	*T*	*T*



$P(B)$
0.4

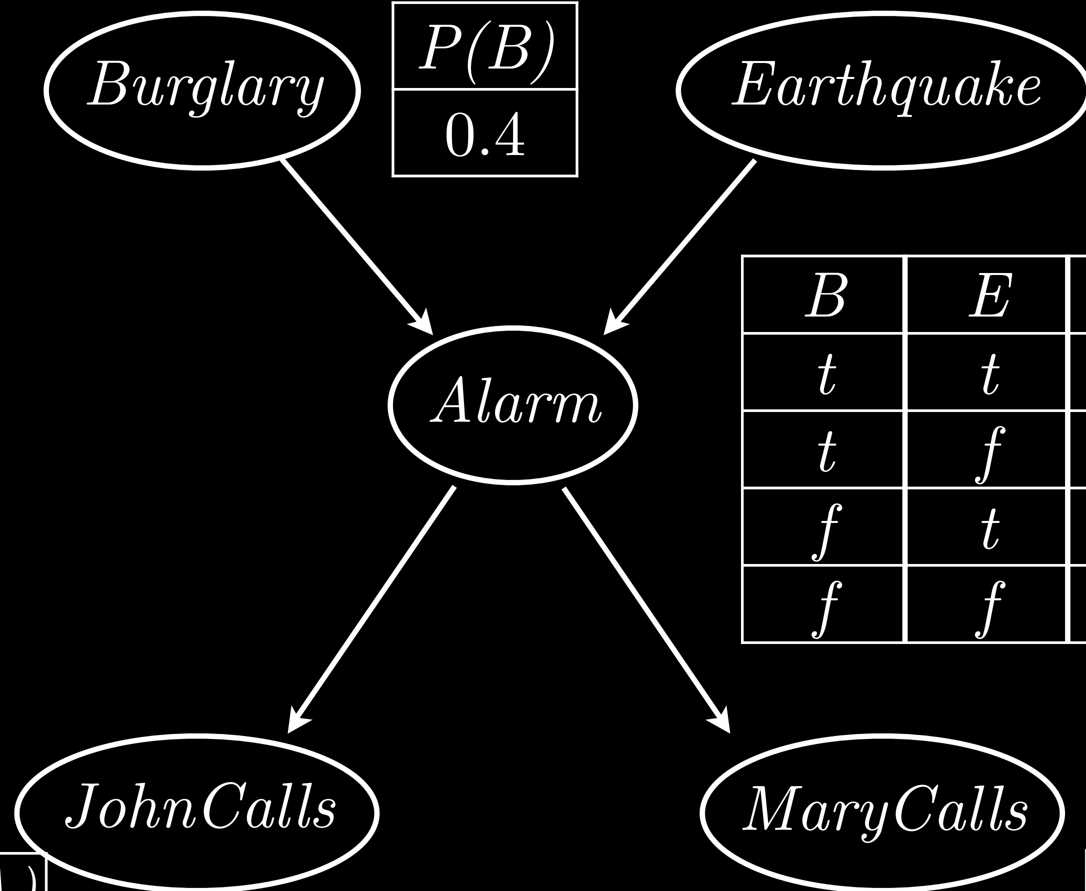
$P(E)$
0.6

B	E	$P(A\|$
t	t	0.5
t	f	?
f	t	1.0
f	f	0.5

A	$P(J\|A)$
t	?
f	?

A	$P(M\|$
t	?
f	?

Burglary	*Earthquake*	*Alarm*	*JohnCalls*	*MaryCalls*
T	T	F	T	F
F	F	T	F	T
F	T	T	T	T
F	F	F	F	F
T	T	T	T	T



$P(B)$
0.4

$P(E)$
0.6

B	E	$P(A\|$
t	t	0.5
t	f	?
f	t	1.0
f	f	0.5

Need more data!

A	$P(J\|A)$
t	?
f	?

A	$P(M\|$
t	?
f	?

Burglary	*Earthquake*	*Alarm*	*JohnCalls*	*MaryCalls*
T	T	F	T	F
F	F	T	F	T
F	T	T	T	T
F	F	F	F	F
T	T	T	T	T

Later in Course:

- Partial data (no specifying all variables)
- Structure learning

Approximate Inference in Bayesian Networks

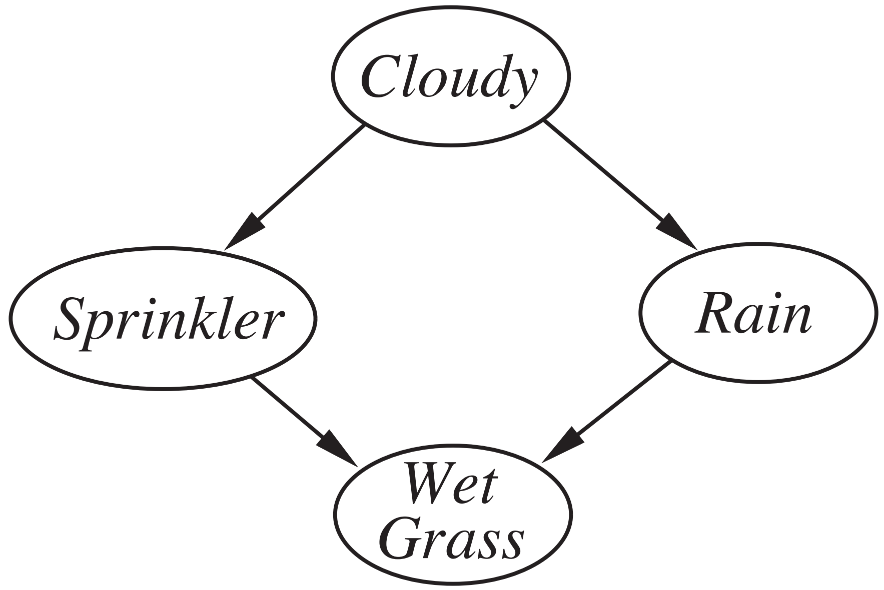
Case 1: No Evidence

- Query variable X
- Non-evidence, non-query ("hidden") variables: Y
- Approximate: $\mathbf{P}(X \mid \mathrm{e})$

Sampling

- Generate assignments of values to the random variables …
- So that in the limit (as number of samples increase), the probability of any event is equal to the frequency of its occurrence in the sample set

$P(C) = .5$



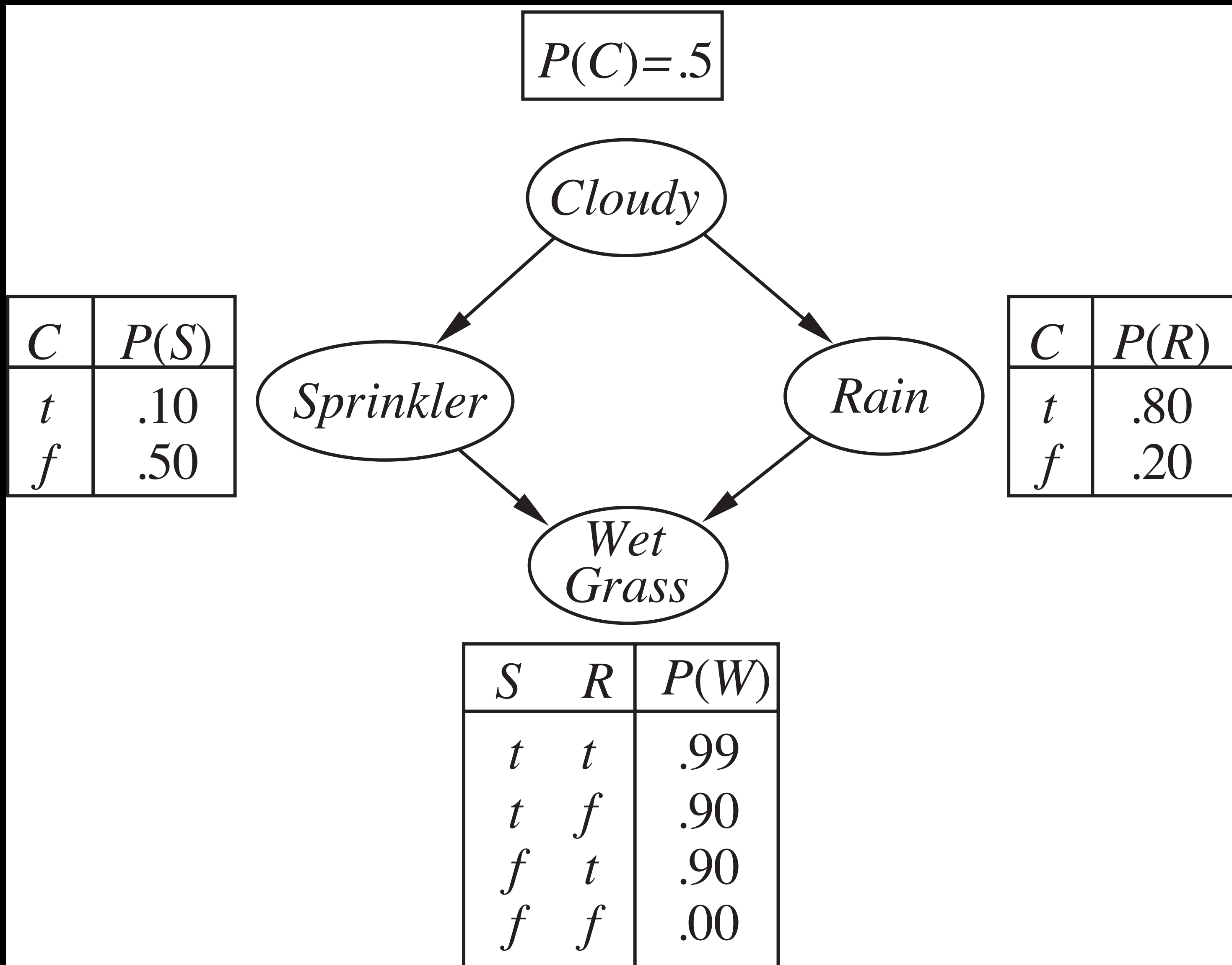
C	$P(S)$
t	.10
f	.50

C	$P(R)$
t	.80
f	.20

S	R	$P(W)$
t	t	.99
t	f	.90
f	t	.90
f	f	.00

Generating Samples

- Sample each variable in topological order
  - Child appears after its parents
- Choose the value for that variable conditioned on the values already chosen for its parents

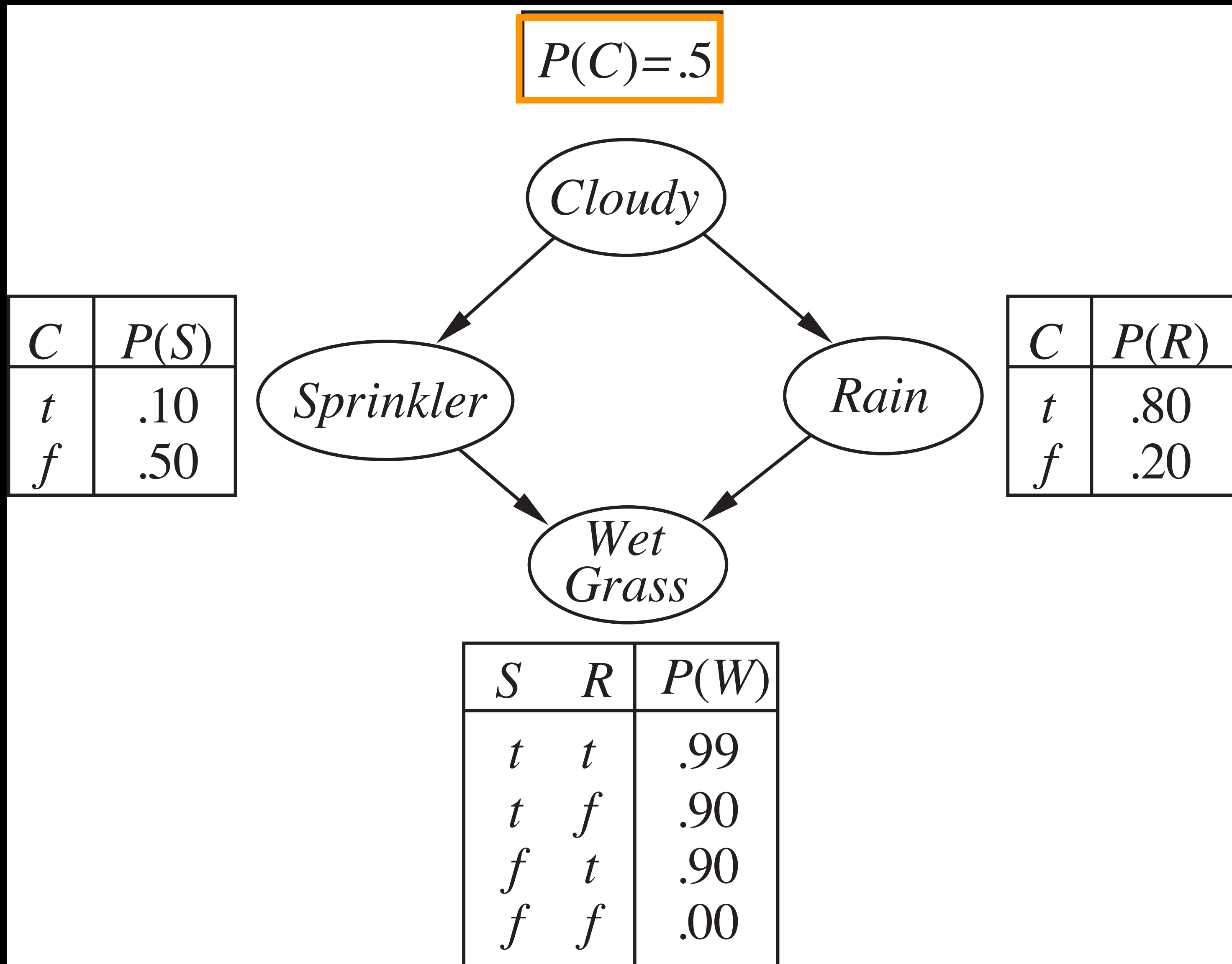


Cloudy

Sprinkler

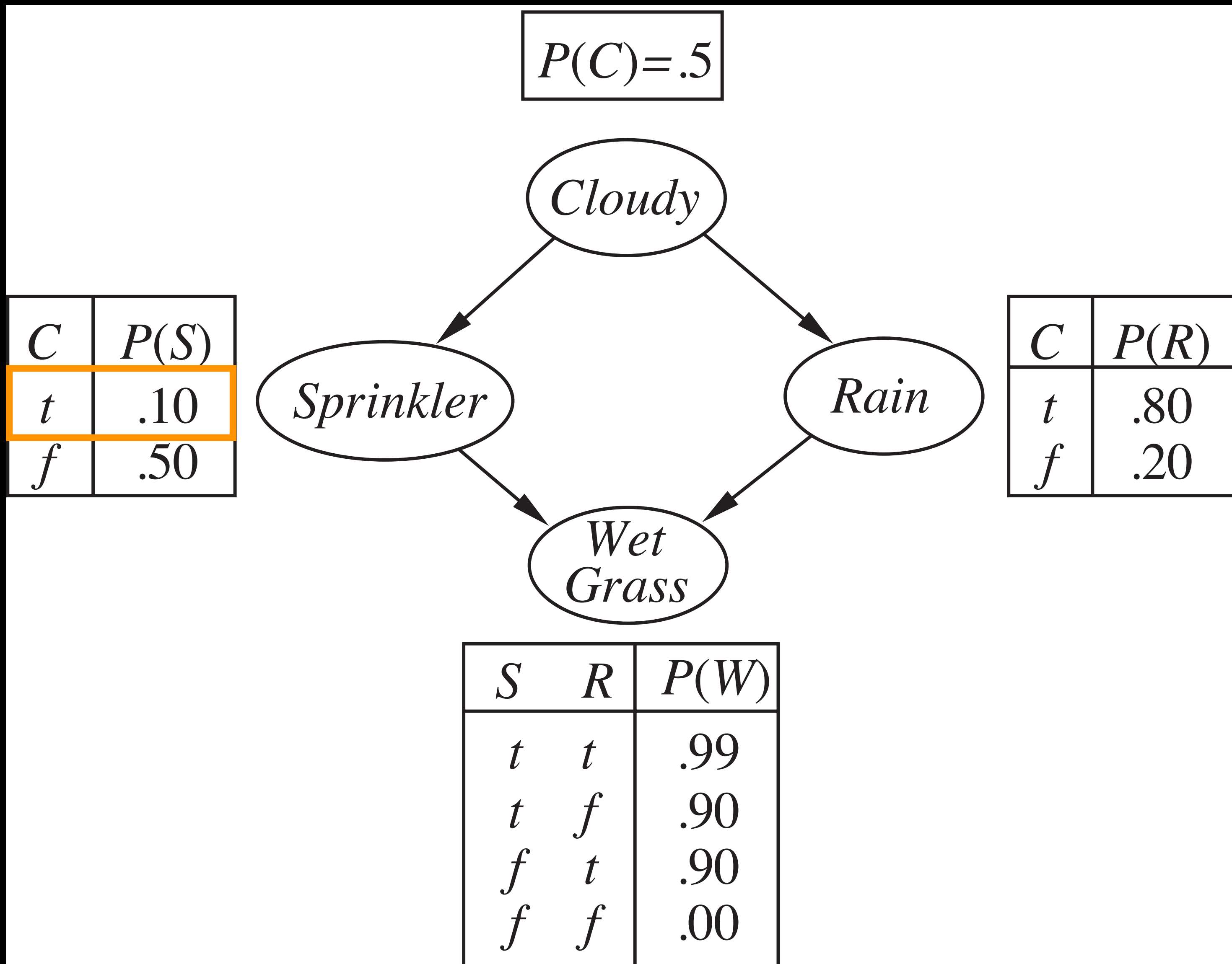
Rain

WetGrass



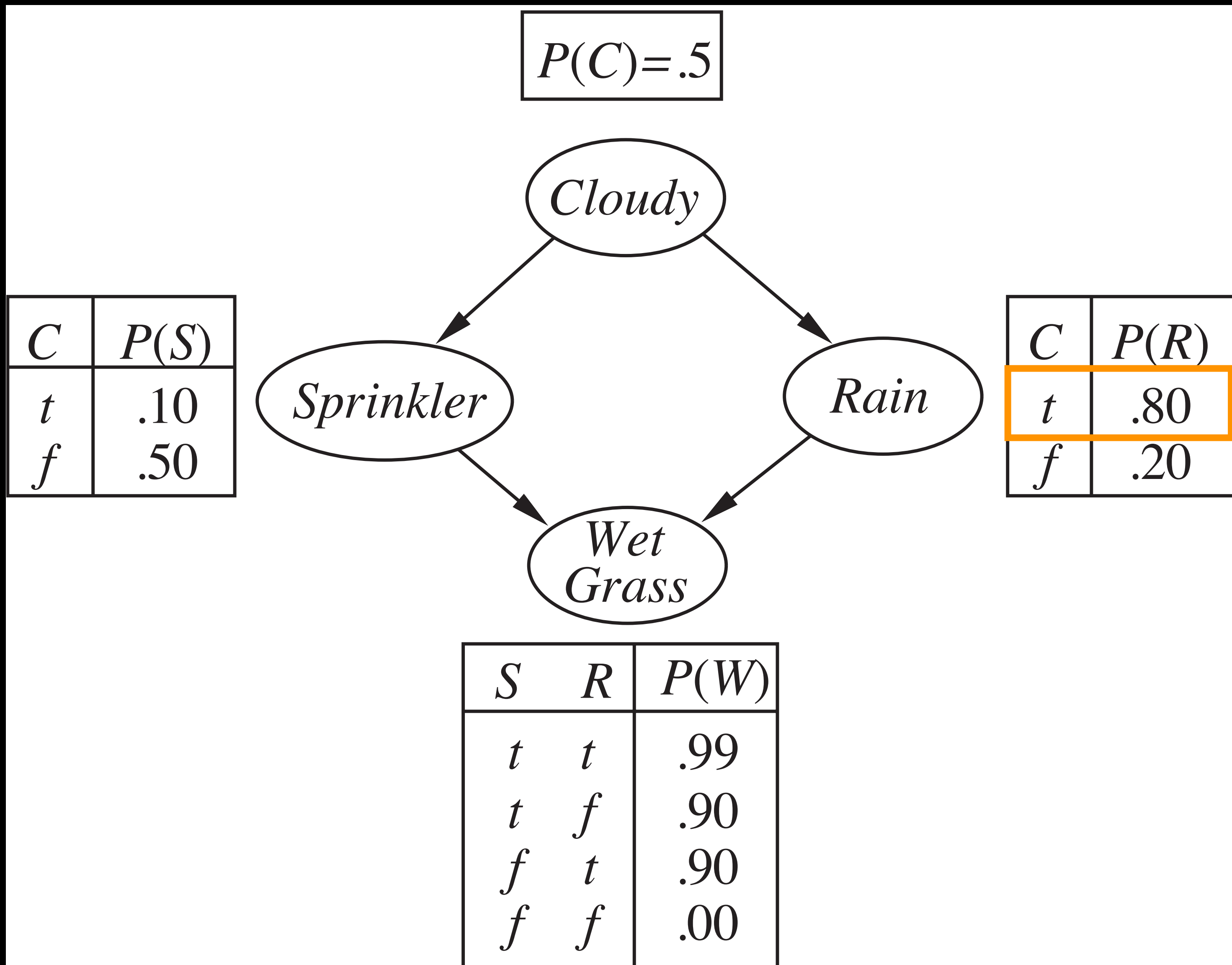
$Cloudy$	$true$
$Sprinkler$	
$Rain$	
$WetGrass$	

$$\mathbf{P}(Cloudy) = \langle 0.5, 0.5 \rangle$$



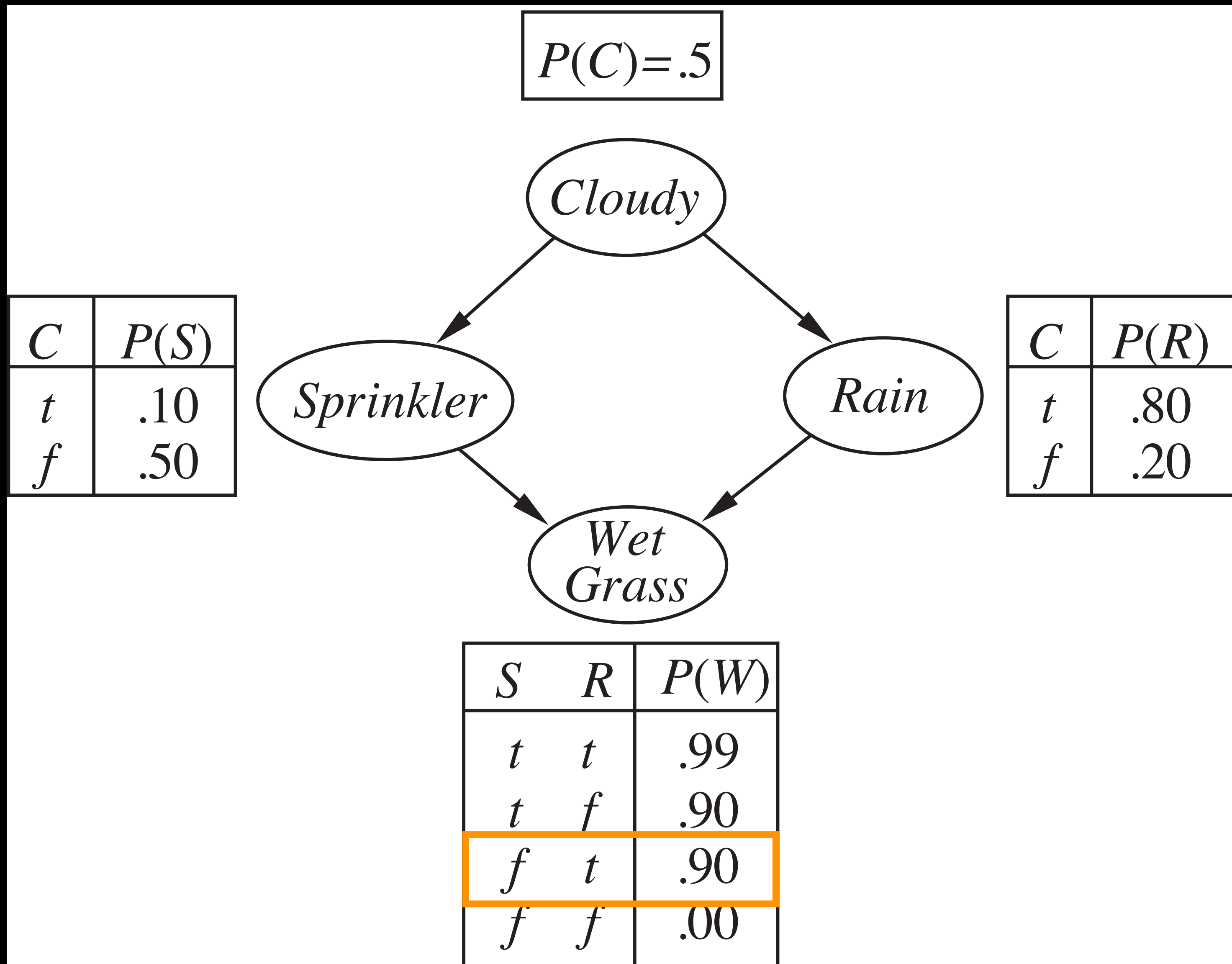
$Cloudy$	$true$
$Sprinkler$	$false$
$Rain$	
$WetGrass$	

$$\mathbf{P}(Sprinkler \mid Cloudy = true) = \langle 0.1, 0.9 \rangle$$



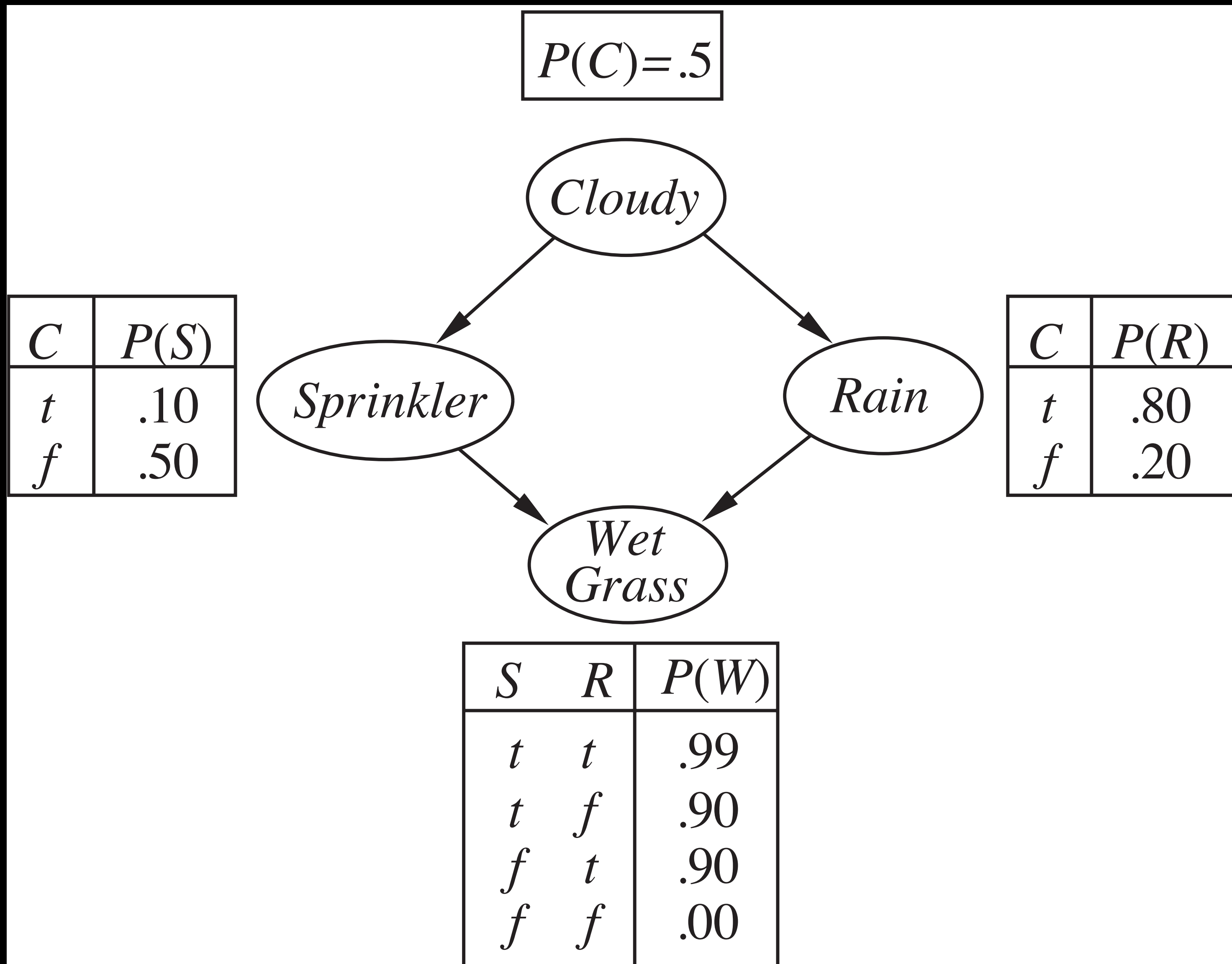
Cloudy	*true*
Sprinkler	*false*
Rain	*true*
WetGrass	

$$\mathbf{P}(Rain \mid Cloudy = true) = \langle 0.8, 0.2 \rangle$$



Cloudy	*true*
Sprinkler	*false*
Rain	*true*
WetGrass	*true*

$$\mathbf{P}(\,WetGrass \mid Sprinkler = false, Rain = true) = \langle 0.9, 0.1 \rangle$$



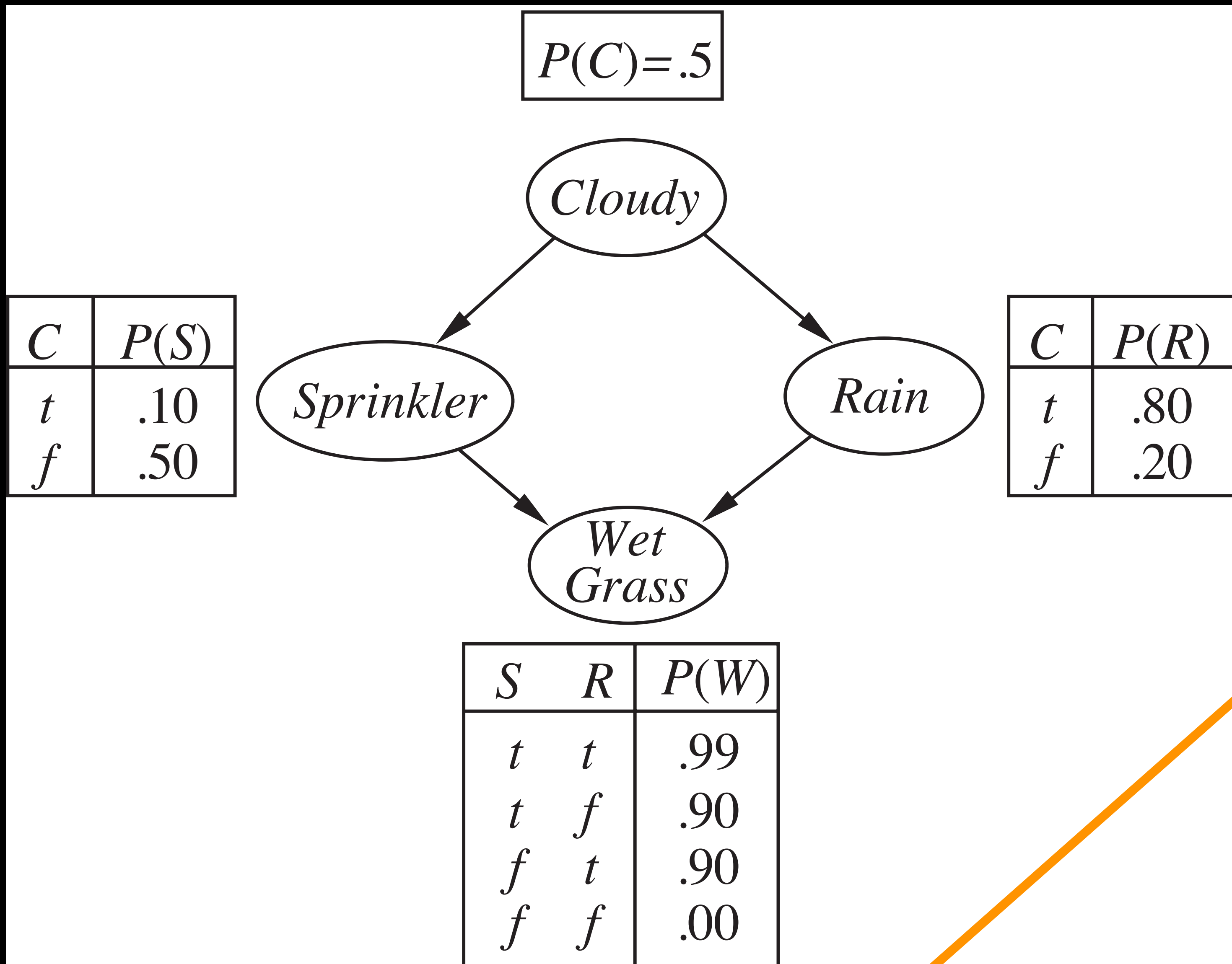
$Cloudy$	$true$
$Sprinkler$	$false$
$Rain$	$true$
$WetGrass$	$true$

$$\langle Cloudy = true, Sprinkler = false, Rain = true, WetGrass = true \rangle$$

Guaranteed to be a consistent estimate
(becomes exact in the large-sample limit)

Case II: Handling Evidence

- Query variable X
- Evidence variables E_1, ..., E_m
  - Observed values: $e = < e_1 , \ldots, e_m >$
- Non-evidence, non-query ("hidden") variables: Y
- Approximate: $\mathbf{P}(X \mid \mathbf{e})$

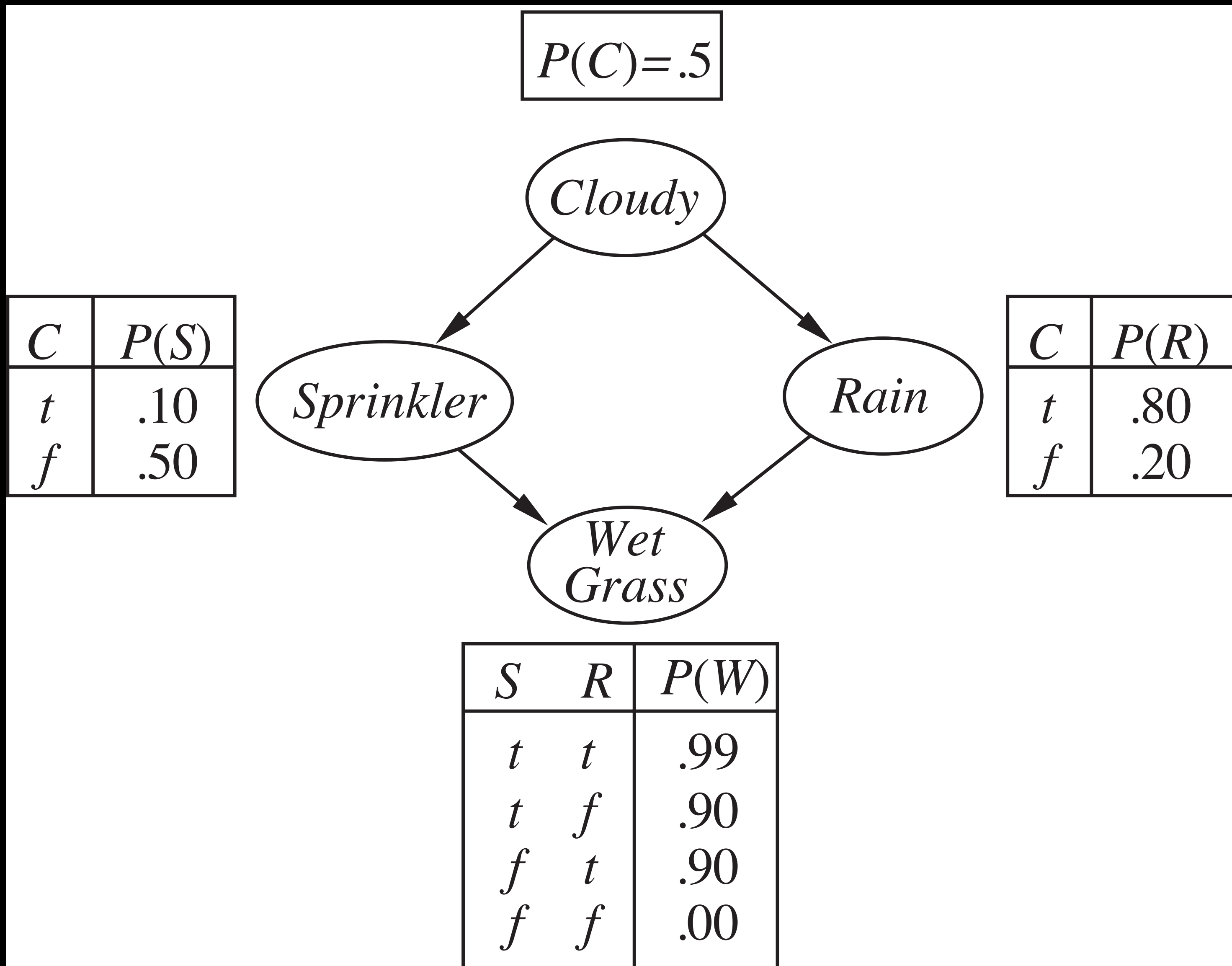


$$\mathbf{P}(Rain \mid Sprinkler = true)$$

$$\langle Cloudy = true, Sprinkler = false, Rain = true, WetGrass = true \rangle$$

Rejection Sampling

- Generate sample from the prior distribution specified by the network
- Reject sample if inconsistent with the evidence
- Use remaining samples to estimate probability of event



$$\mathbf{P}(Rain \mid Sprinkler = true)$$

100 samples

$Sprinkler{=}false$: 73

$Sprinkler{=}true$: 27

$Rain{=}true$: 8

$Rain{=}false$: 19

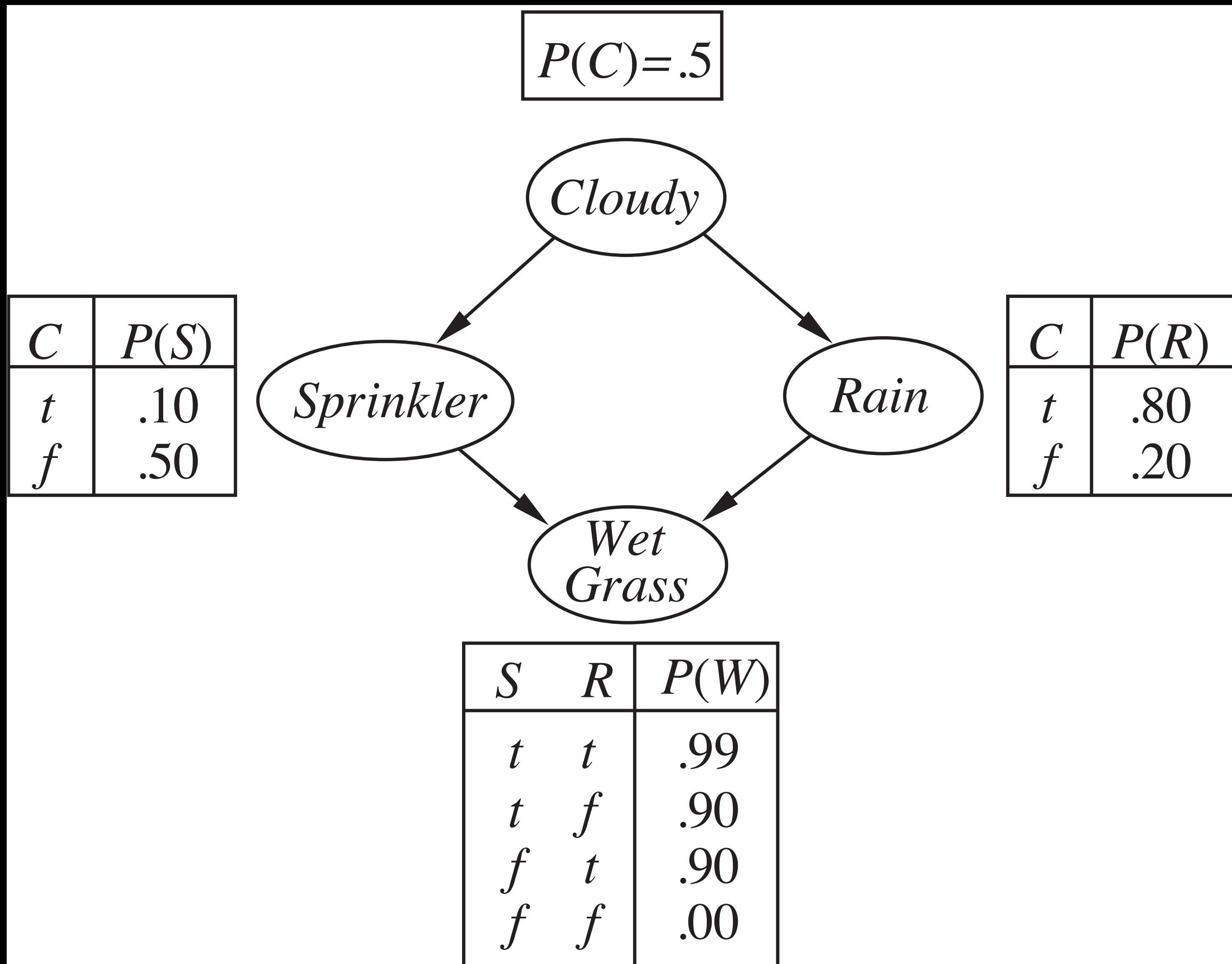
$$\mathbf{P}(Rain \mid Sprinkler = true) \approx \alpha\,\langle \frac{8}{27}, \frac{19}{27} \rangle = \langle 0.296, 0.704 \rangle$$

Rejection Sampling

- Generate sample from the prior distribution specified by the network
- Reject sample if inconsistent with the evidence
- Use remaining samples to estimate probability of event
- Problem: Fraction of samples consistent with the evidence drops exponentially with number of evidence variables

Likelihood Weighting

- Generate only samples consistent with the evidence
  - i.e., fix values of evidence variables
- Instead of counting 1 for each non-rejected sample, weight the count by the likelihood (probability) of the sample given the evidence



Cloudy

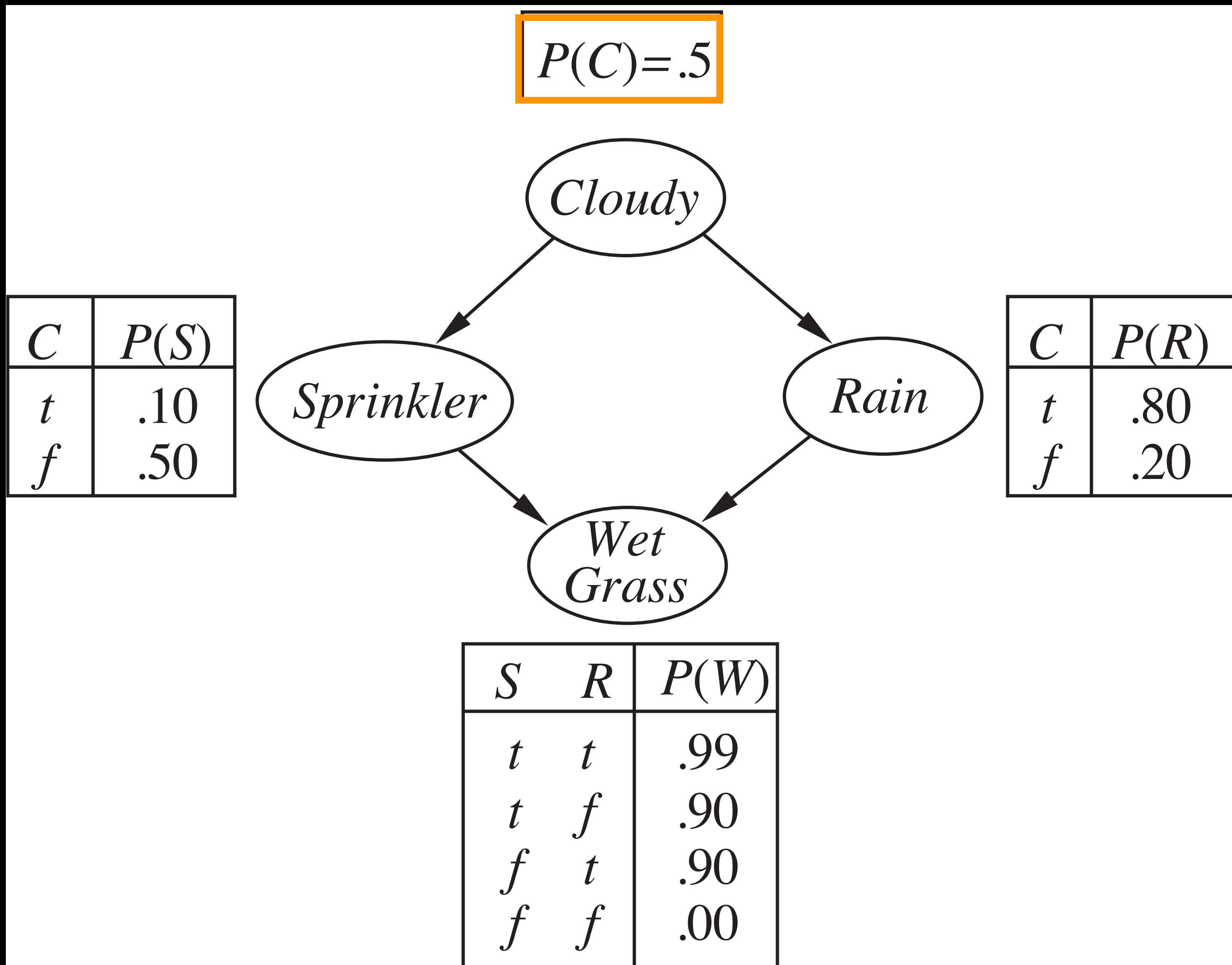
Sprinkler

Rain

WetGrass

$$w = 1.0$$

$$P(Rain|Sprinkler = true, WetGrass = true)$$



$Cloudy$ $true$

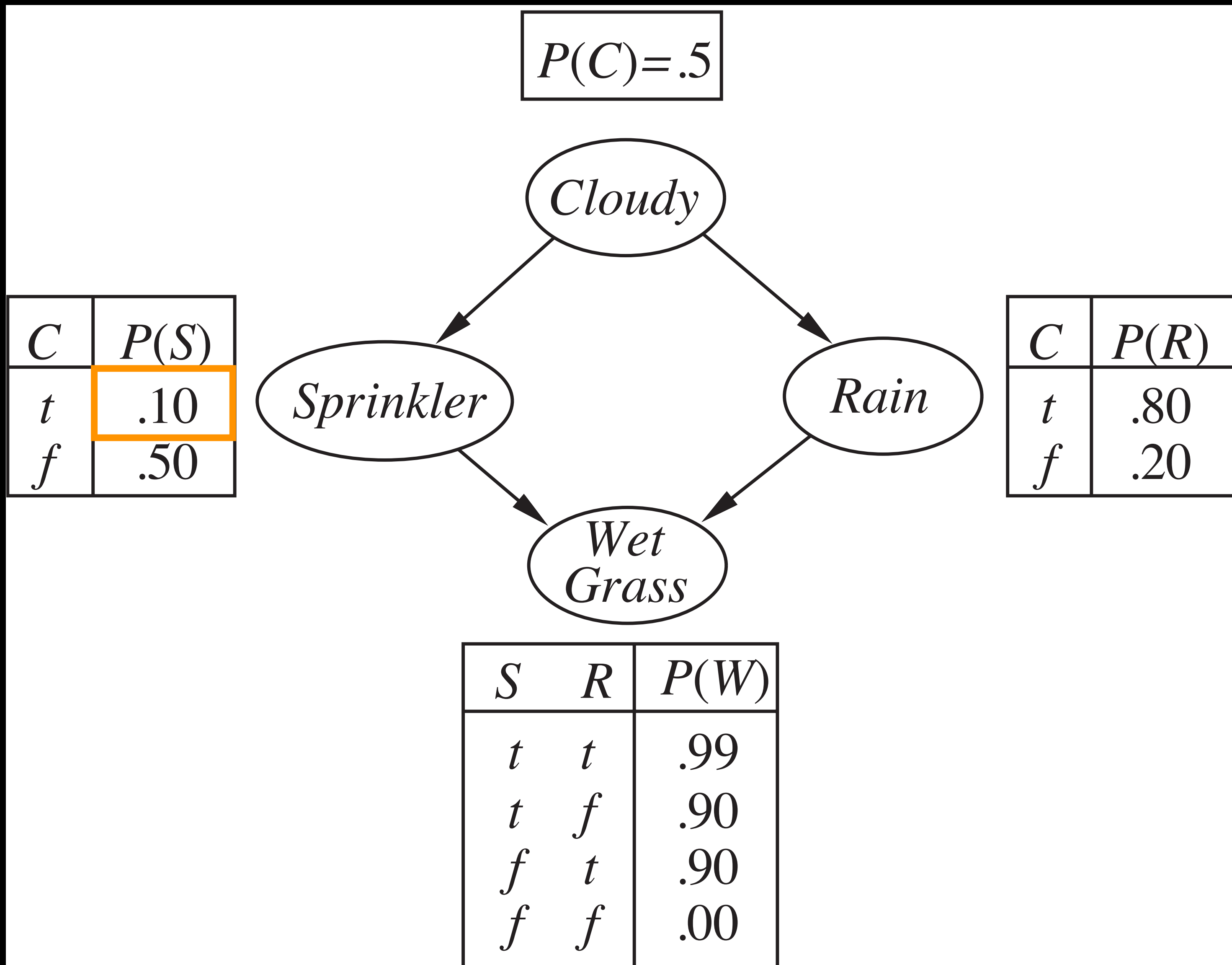
$Sprinkler$

$Rain$

$WetGrass$

$$w = 1.0$$

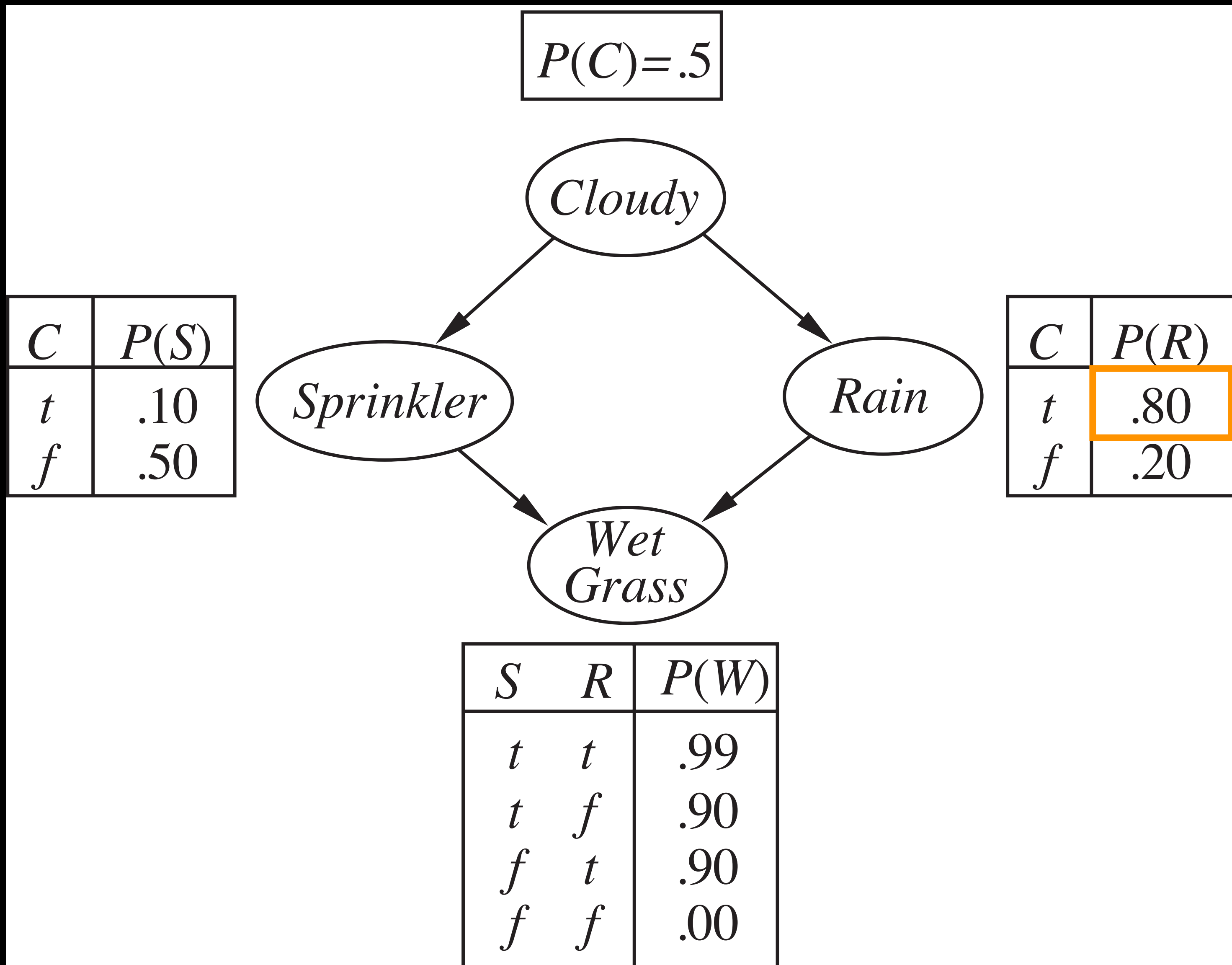
$$P(Rain|Sprinkler = true, WetGrass = true)$$



Cloudy	*true*
Sprinkler	*true*
Rain	
WetGrass	

$$w = 1.0 \times 0.1 = 0.10$$

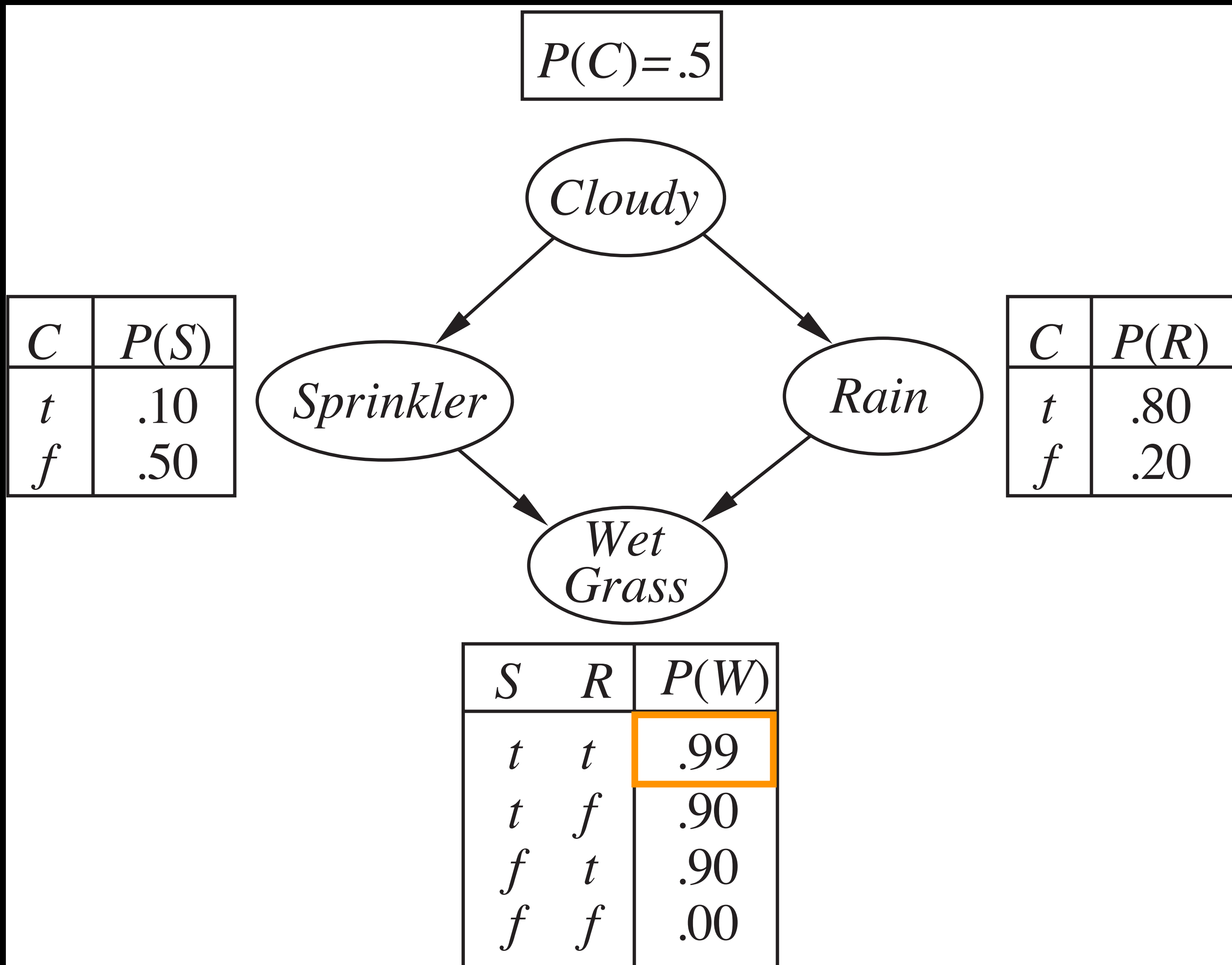
$$P(Rain|Sprinkler = true, WetGrass = true)$$



$Cloudy$	$true$
$Sprinkler$	$true$
$Rain$	$true$
$WetGrass$	

$$w = 1.0 \times 0.1 = 0.10$$

$$P(Rain|Sprinkler = true, WetGrass = true)$$

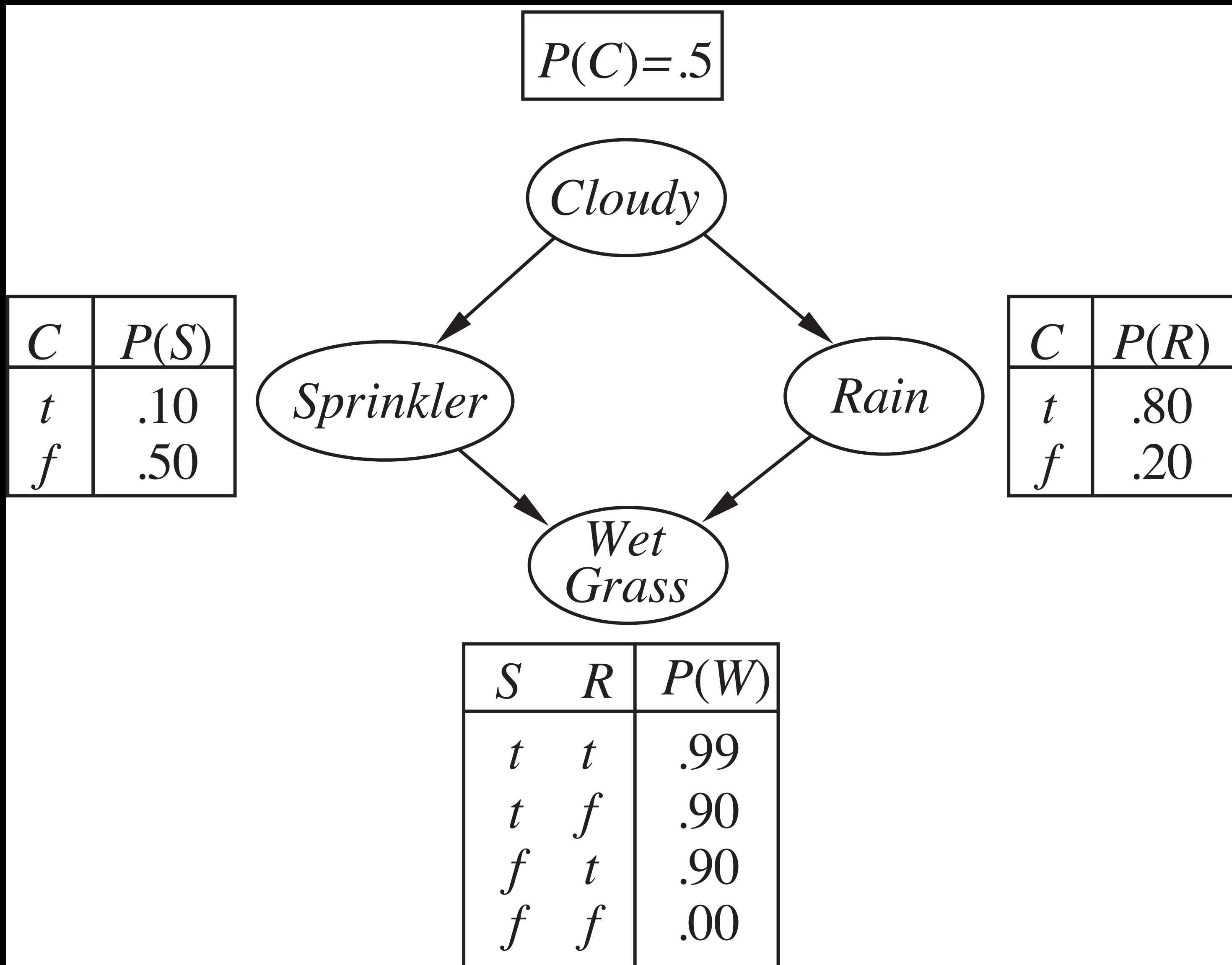


$Cloudy$	$true$
$Sprinkler$	$false$
$Rain$	$true$
$WetGrass$	$true$

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

$$0.099$$

$$P(Rain|Sprinkler = true, WetGrass = true)$$



Cloudy	*true*
Sprinkler	*false*
Rain	*true*
WetGrass	*true*

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

$$P(Rain|Sprinkler = true, WetGrass = true)$$

$$w = 0.099$$
$$\langle Cloudy = true, Sprinkler = true, Rain = true, Wetgrass = true \rangle$$

Likelihood Weighting

- Generate sample using topological order
  - Evidence variable: Fix value to evidence value and update weight of sample using probability in network
  - Non-evidence variable: Sample from values using probabilities in the network (given parents)

Likelihood Weighting

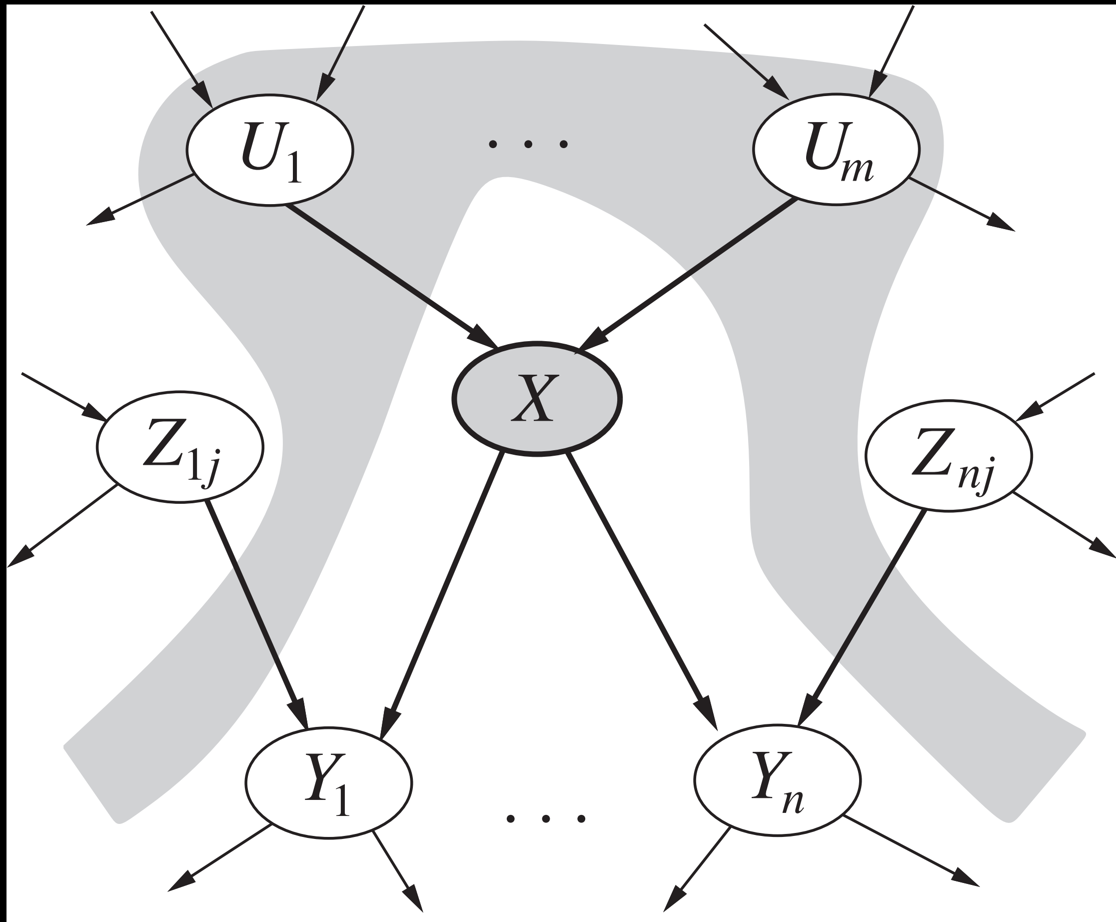
- Pros:
  - Doesn't reject any samples
- Cons:
  - More evidence $\Rightarrow$ lower weight
  - Affected by order of evidence vars in topological sort (later = worse)

Approximate Inference in Bayesian Networks

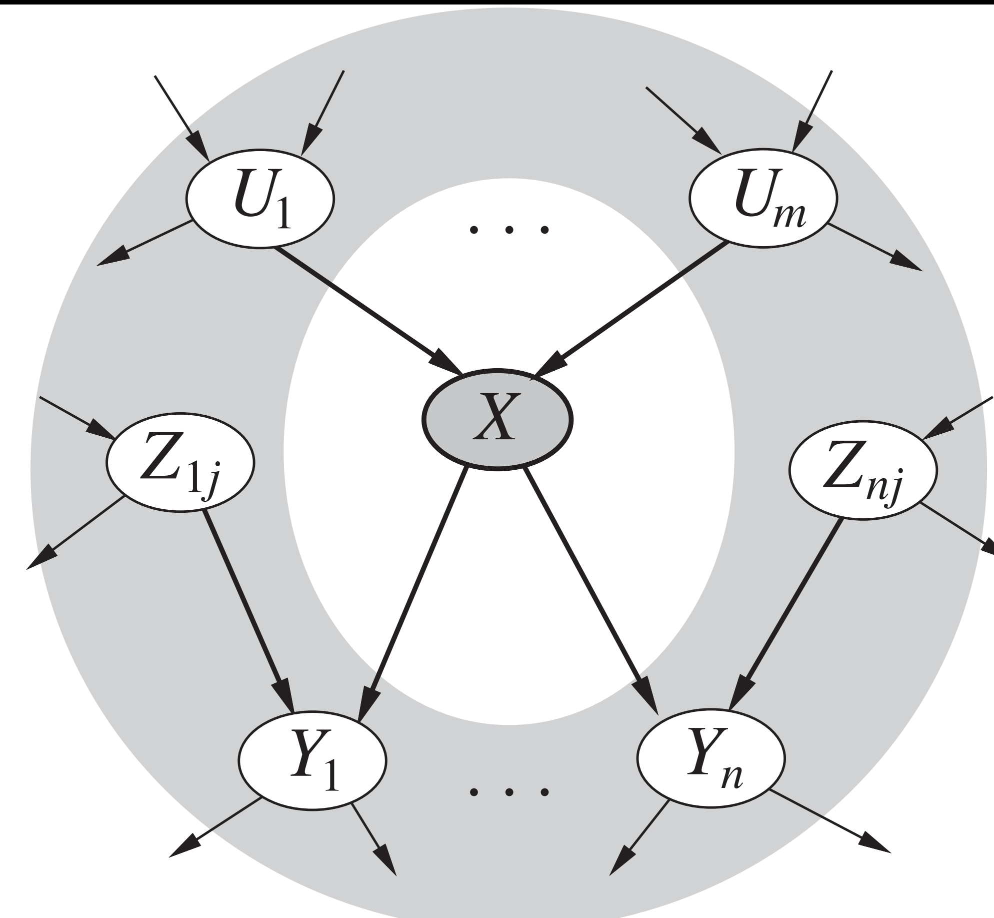
- Rejection Sampling
- Likelihood Weighting

Markov Chain Monte Carlo Simulation

- To approximate: $\mathbf{P}(X \mid \mathbf{e})$
- Start with a random state (complete assignment to the random variables)
- Move to a neighboring state (change one variable)
- Repeating gives a “chain” of sampled states



Conditional Independence



Markov Blanket

Markov Blanket

- The Markov Blanket of a node is its parents, its children, and its children's parents.
- A node is conditionally independent of all other nodes in the network given its Markov Blanket

MCMC Simulation
Gibbs Sampling

- To approximate: $\mathbf{P}(X \mid \mathbf{e})$
- Start in a state with evidence variables set to evidence values (others arbitrary)
- On each step, sample the non-evidence variables conditioned on the values of the variables in their Markov Blankets
- A form of local search! See book for details!

Approximate Inference in Bayesian Networks

- Sampling consistent with a distribution
- Rejection Sampling: simple but inefficient
- Likelihood Weighting: better
- Gibbs Sampling: a Markov-Chain Monte Carlo algorithm, similar to local search
- All generate <u>consistent</u> estimates (equal to exact probability in the large-sample limit)