INFERÊNCIA EM REDES BAYESIANAS CAP 14 (14.4 - 14.5)

Parcialmente adaptado de

http://aima.eecs.berkeley.edu

Resumo

- Inferência exacta por enumeração
- Inferência exacta por eliminação de variáveis
- Inferência aproximada por simulação estocástica

Tarefas de inferência

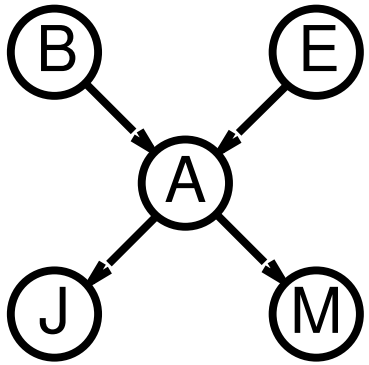
- **Interrogações simples**: calcular distribuição marginal à posteriori $\mathbf{P}(X_i|\mathbf{E}=\mathbf{e})$
  - e.g., $P(NoGas|Gauge=empty, Lights=on, Starts=false)$
- **Interrogações conjuntivas**:

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

- **Decisões óptimas**: redes de decisão incluem informação de utilidade; inferência probabilística necessária para $P(outcome|action,evidence)$
- **Valor da informação**: que evidência procurar a seguir?
- **Analise de sensibilidade**: quais os valores de probabilidade mais críticos?
- **Explicação**: por que é que preciso de um novo motor de arranque?

Inferência por enumeração

- Forma relativamente inteligente de somar variáveis da distribuição conjunta sem necessitar de construir a sua representação explícita
- Interrogação simples na rede do assaltante:



$$\mathbf{P}(B \mid j,m)$$
$$= \mathbf{P}(B,j,m) / P(j,m)$$
$$= \alpha \mathbf{P}(B,j,m)$$
$$= \alpha \sum_e \sum_a \mathbf{P}(B,e,a,j,m)$$

- Rescrever entradas da distribuição conjunta recorrendo ao produto de entradas da tabela de probabilidade condicional (CPT):

$$\mathbf{P}(B \mid j,m)$$
$$= \alpha \sum_e \sum_a \mathbf{P}(B) P(e) \mathbf{P}(a \mid B,e) P(j \mid a) P(m \mid a)$$
$$= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a \mid B,e) P(j \mid a) P(m \mid a) \qquad (1)$$

Inferência por enumeração

$$\mathbf{P}(B \mid j,m) = \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a \mid B,e) P(j \mid a) P(m \mid a) \qquad (1)$$

- Fazendo *B=true* em (1) (ignorando α):

$$\mathbf{P}(B = true) \sum_{e \in \{true,false\}} P(E = e) \sum_{a \in \{true,false\}} \mathbf{P}(A = a \mid B = true, E = e) P(J = true \mid A = a) P(M = true \mid A = a)$$

$$= 0.001 \times [0.002 \times (0.95 \times 0.9 \times 0.7 + 0.05 \times 0.05 \times 0.01) + 0.098 (0.94 \times 0.9 \times 0.7 \times 0.06 \times 0.05 \times 0.01)] = 0.000592243$$

- Fazendo *B=false* em (1) (ignorando α):

$$\mathbf{P}(B = false) \sum_{e \in \{true,false\}} P(E = e) \sum_{a \in \{true,false\}} \mathbf{P}(A = a \mid B = false, E = e) P(J = true \mid A = a) P(M = true \mid A = a)$$

$$= 0.999 \times [0.002 \times (0.29 \times 0.9 \times 0.7 + 0.71 \times 0.05 \times 0.01) + 0.098 (0.001 \times 0.9 \times 0.7 \times 0.999 \times 0.05 \times 0.01)] = 0.001491858$$

- Normalizando:

$$\mathbf{P}(B = true \mid J = true, M = true) = 0.000592243 / (0.000592243 + 0.001491858) = 0.284171835$$

$$\mathbf{P}(B = false \mid J = true, M = true) = 0.001491858 / (0.000592243 + 0.001491858) = 0.715828165$$

Algoritmo de enumeração

function ENUMERATION-ASK(X, $\mathbf{e}$, bn) **returns** a distribution over X

inputs: X, the query variable
$\mathbf{e}$, observed values for variables $\mathbf{E}$
bn, a Bayesian network with variables $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$

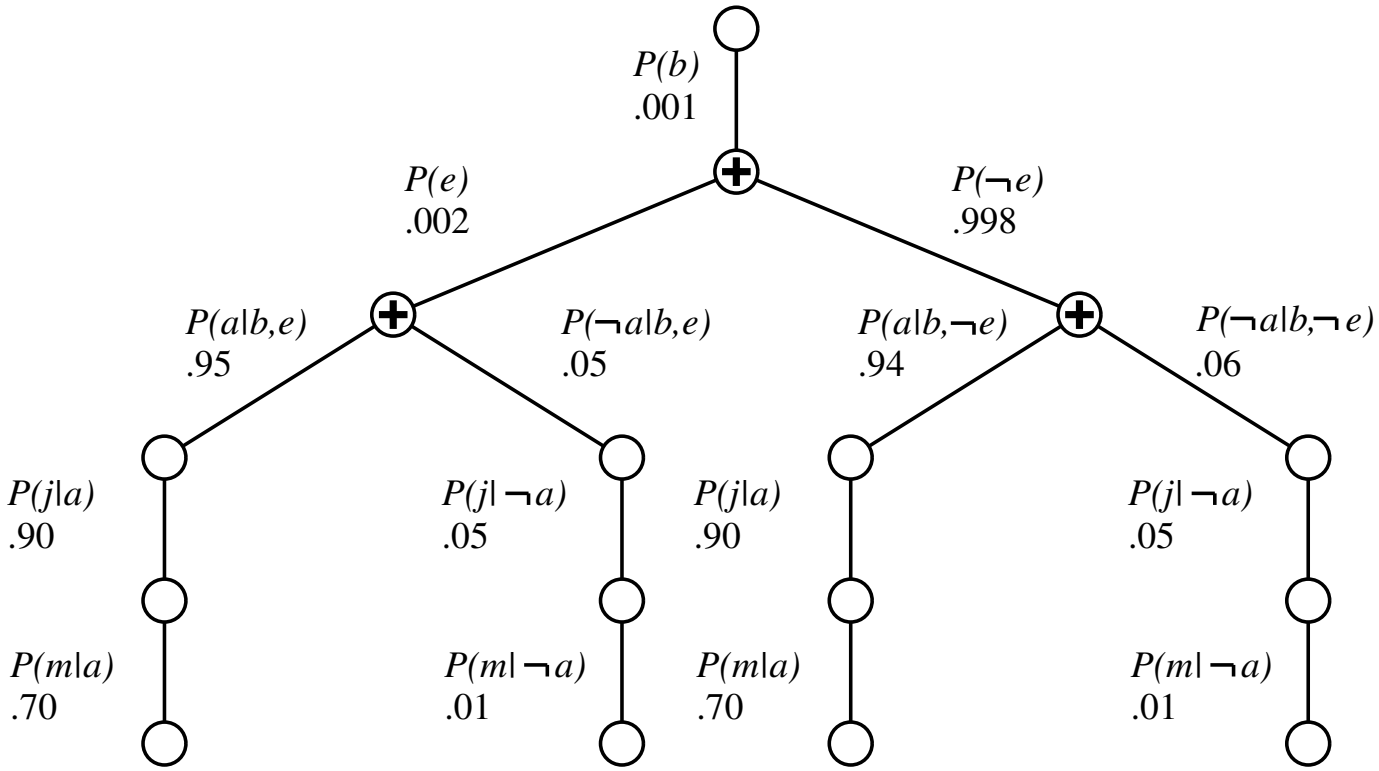
$\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
for each value x_i of X **do**
extend $\mathbf{e}$ with value x_i for X
$\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL(VARS[bn], $\mathbf{e}$)
return NORMALIZE($\mathbf{Q}(X)$)

---

function ENUMERATE-ALL($vars$, $\mathbf{e}$) **returns** a real number
if EMPTY?($vars$) **then return** 1.0
$Y \leftarrow$ FIRST($vars$)
if Y has value y in $\mathbf{e}$
then return $P(y \mid Pa(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}$)
else return $\Sigma_y \; P(y \mid Pa(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_y$)
where $\mathbf{e}_y$ is $\mathbf{e}$ extended with $Y = y$

Árvore de avaliação

- Enumeração recursiva em profundidade primeiro: espaço O(n), tempo O(d^n)
- Enumeração é ineficiente: cálculos repetidos
  - e.g., calcula $P(j|a)P(m|a)$ para cada valor de e



Inferência por eliminação de variáveis

- Eliminação de variáveis: efectuar somatórios da direita para a esquerda, armazenando resultados intermédios (factores) para evitar recomputação

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

$$= \alpha \underbrace{\mathbf{P}(B)}_{B} \sum_e \underbrace{P(e)}_{E} \sum_a \underbrace{\mathbf{P}(a \mid B,e)}_{A} \underbrace{P(j \mid a)}_{J} \underbrace{P(m \mid a)}_{M}$$

$$= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a \mid B,e) P(j \mid a) f_M(a)$$

$$= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a \mid B,e) f_J(a) f_M(a)$$

$$= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a f_A(a,B,e) f_J(a) f_M(a)$$

$$= \alpha \mathbf{P}(B) \sum_e P(e) f_{\bar{A}JM}(B,e) \quad \text{(soma-se } A\text{)}$$

$$= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(B) \quad \text{(soma-se } E\text{)}$$

$$= \alpha f_B(B) f_{\bar{E}\bar{A}JM}(B)$$

Eliminação de variáveis: operações básicas

- Somar para eliminar a variável de um produto de factores:
  - deslocar todos os factores constantes para fora do somatório
  - adicionar sub-matrizes no produto pontual dos factores restantes

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

assumindo que $f_1,\ldots,f_i$ não depende de X

- Produto pontual dos factores f_1 e f_2:

$$f_1(x_1,\ldots,x_j,y_1,\ldots,y_k) \times f_2(y_1,\ldots,y_k,z_1,\ldots,z_l)$$
$$= f(x_1,\ldots,x_j,y_1,\ldots,y_k,z_1,\ldots,z_l)$$

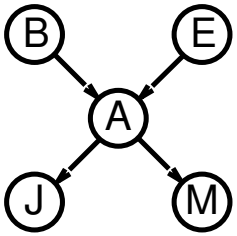
- E.g., $f_1(a,b) \times f_2(b,c) = f(a,b,c)$

Algoritmo de eliminação de variáveis

function ELIMINATION-ASK($X, \mathbf{e}, bn$) **returns** a distribution over X
  inputs: X, the query variable
    $\mathbf{e}$, evidence specified as an event
    bn, a belief network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

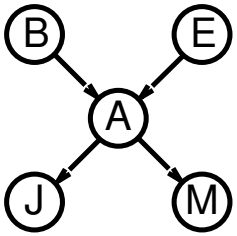
  $factors \leftarrow [\,]$; $vars \leftarrow$ REVERSE(VARS[bn])
  for each var **in** $vars$ **do**
    $factors \leftarrow [$MAKE-FACTOR$(var, \mathbf{e})|factors]$
    if var is a hidden variable **then** $factors \leftarrow$ SUM-OUT$(var, factors)$
  return NORMALIZE(POINTWISE-PRODUCT($factors$))

Algoritmo de eliminação de variáveis: exemplo



- Variável de Interrogação: $Burglary$
- Evidência: $JohnCalls=true$, $MaryCalls=true$
- Ordenação da variáveis (das folhas para a raíz): M, J, A, B, E
- Factores: []

Algoritmo de eliminação de variáveis: exemplo

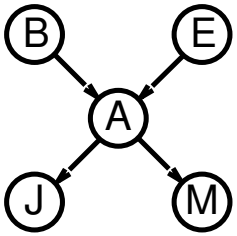


- Variável de Interrogação: *Burglary*
- Evidência: *JohnCalls=true*, *MaryCalls=true*
- Ordenação da variáveis (das folhas para a raíz): M, J, A, B, E
- Factores: [$f_M(A)$]

- Factor da variável M: $f_M(A)=P(M=true|A)$

A	$f_M(A)$
true	0.70
false	0.01

Algoritmo de eliminação de variáveis: exemplo



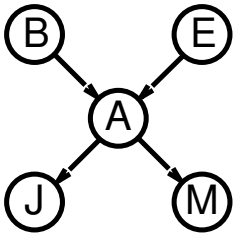
- Variável de Interrogação: *Burglary*
- Evidência: *JohnCalls=true*, *MaryCalls=true*
- Ordenação da variáveis (das folhas para a raíz): M, J, A, B, E
- Factores: [$f_J(A)$,$f_M(A)$]

- Factor da variável J: $f_J(A)=P(J=true|A)$

A	$f_J(A)$
true	0.90
false	0.05

A	$f_M(A)$
true	0.70
false	0.01

Algoritmo de eliminação de variáveis: exemplo



- Variável de Interrogação: *Burglary*
- Evidência: *JohnCalls=true*, *MaryCalls=true*
- Ordenação da variáveis (das folhas para a raíz): M, J, A, B, E
- Factores: [$f_A(A,B,E), f_J(A), f_M(A)$]

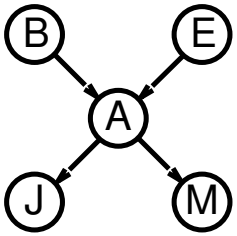
- Factor da variável A: $f_A(A,B,E)=P(A|B,E)$

A	B	E	$f_A(A,B,E)$
true	true	true	0.95
true	true	false	0.94
true	false	true	0.29
true	false	false	0.001
false	true	true	0.05
false	true	false	0.06
false	false	true	0.71
false	false	false	0.999

A	$f_J(A)$
true	0.90
false	0.05

A	$f_M(A)$
true	0.70
false	0.01

Algoritmo de eliminação de variáveis: exemplo



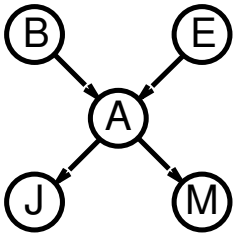
- Variável de Interrogação: $Burglary$
- Evidência: $JohnCalls=true$, $MaryCalls=true$
- Ordenação da variáveis (das folhas para a raíz): M, J, A, B, E
- Factores: $[f_A(A,B,E), f_J(A), f_M(A)]$

- Como a variável A é oculta vai-se eliminá-la (através da soma). Para isso é necessário:
  - Efectuar o produto pontual dos factores que têm o parâmetro A

$$f_{AJM}(A,B,E) = f_A(A,B,E) \times f_J(A) \times f_M(A)$$

  - Eliminar a variável A obtendo o factor $f_{\bar{A}JM}(B,E)$

Algoritmo de eliminação de variáveis: exemplo



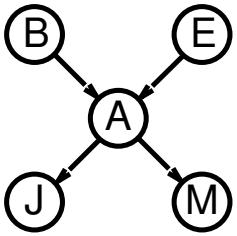
- Variável de Interrogação: *Burglary*
- Evidência: *JohnCalls=true*, *MaryCalls=true*
- Ordenação da variáveis (das folhas para a raíz): M, J, A, B, E
- Factores: [$f_A(A,B,E), f_J(A), f_M(A)$]

- Produto pontual:

$$f_{AJM}(A,B,E) = f_A(A,B,E) \times f_{JM}(A)$$
$$f_{JM}(A) = f_J(A) \times f_M(A)$$

A	$f_{JM}(A)$		A	$f_J(A)$		A	$f_M(A)$
true	0.90×0.70=0.63	=	true	0.90	×	true	0.70
false	0.05×0.01=0.0005		false	0.05		false	0.01

Algoritmo de eliminação de variáveis: exemplo



- Variável de Interrogação: *Burglary*
- Evidência: *JohnCalls=true*, *MaryCalls=true*
- Ordenação da variáveis (das folhas para a raíz): M, J, A, B, E
- Factores: $[f_A(A,B,E), f_J(A), f_M(A)]$

- Produto pontual: $f_{AJM}(A,B,E) = f_A(A,B,E) \times f_{JM}(A)$

A	B	E	$f_{AJM}(A,B,E)$
true	true	true	0.95×0.63=0.5985
true	true	false	0.94×0.63=0.5922
true	false	true	0.29×0.63=0.1827
true	false	false	0.001×0.63=0.00063
false	true	true	0.05×0.0005=0.000025
false	true	false	0.06×0.0005=0.00003
false	false	true	0.71×0.0005=0.000355
false	false	false	0.999×0.0005=0.0004995

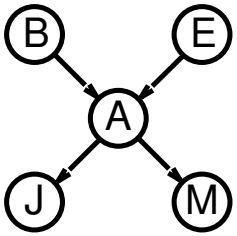
=

A	B	E	$f_A(A,B,E)$
true	true	true	0.95
true	true	false	0.94
true	false	true	0.29
true	false	false	0.001
false	true	true	0.05
false	true	false	0.06
false	false	true	0.71
false	false	false	0.999

×

A	$f_{JM}(A)$
true	0.90×0.70=0.63
false	0.05×0.01=0.0005

Algoritmo de eliminação de variáveis: exemplo



- Variável de Interrogação: *Burglary*
- Evidência: *JohnCalls=true*, *MaryCalls=true*
- Ordenação da variáveis (das folhas para a raíz): M, J, A, B, E
- Factores: [$f_{\bar{A}JM}(B,E)$]

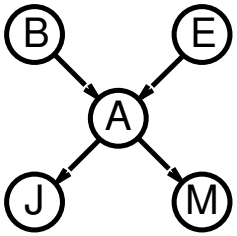
- Eliminação por soma da variável A: $f_{\bar{A}JM}(B,E)$

B	E	$f_{\bar{A}JM}(B,E)$
true	true	0.5985+0.000025=0.598525
true	false	0.5922+0.00003=0.59223
false	true	0.1827+0.000355=0.183055
false	false	0.00063+0.0004995=0.00113

$=\text{SUM}_A$

A	B	E	$f_{AJM}(A,B,E)$
true	true	true	0.5985
true	true	false	0.5922
true	false	true	0.1827
true	false	false	0.00063
false	true	true	0.000025
false	true	false	0.00003
false	false	true	0.000355
false	false	false	0.0004995

Algoritmo de eliminação de variáveis: exemplo



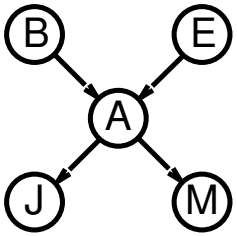
- Variável de Interrogação: *Burglary*
- Evidência: *JohnCalls=true*, *MaryCalls=true*
- Ordenação da variáveis (das folhas para a raíz): M, J, A, B, E
- Factores: [$f_B(B), f_{\bar{A}JM}(B,E)$]

- Factor da variável B: $f_B(B)=P(B)$

B	$f_B(B)$
true	0.001
false	0.999

B	E	$f_{\bar{A}JM}(A,B,E)$
true	true	0.598525
true	false	0.59223
false	true	0.183055
false	false	0.00113

Algoritmo de eliminação de variáveis: exemplo



- Variável de Interrogação: *Burglary*
- Evidência: *JohnCalls=true*, *MaryCalls=true*
- Ordenação da variáveis (das folhas para a raíz): M, J, A, B, E
- Factores: [$f_E(E), f_B(B), f_{\bar{A}JM}(B,E)$]

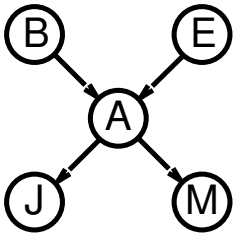
- Factor da variável E: $f_E(E)=P(E)$

E	$f_E(E)$
true	0.002
false	0.998

B	$f_B(B)$
true	0.001
false	0.999

B	E	$f_{\bar{A}JM}(A,B,E)$
true	true	0.598525
true	false	0.59223
false	true	0.183055
false	false	0.00113

Algoritmo de eliminação de variáveis: exemplo



- Variável de Interrogação: *Burglary*
- Evidência: *JohnCalls=true*, *MaryCalls=true*
- Ordenação da variáveis (das folhas para a raíz): *M*, *J*, *A*, *B*, *E*
- Factores: [$f_B(B), f_{\bar{E}\bar{A}JM}(B,E)$]

- Eliminação da variável E: $f_{\bar{E}\bar{A}JM}(B) = SUM_E(f_E(E) \times f_{\bar{A}JM}(B,E))$

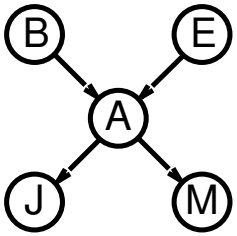
B	*E*	$f_{E\bar{A}JM}(B,E)$		*E*	$f_E(E)$	×	*B*	*E*	$f_{\bar{A}JM}(A,B,E)$
true	true	0.002×0.598525=0.001197	=	true	0.002	×	true	true	0.598525
true	false	0.998×0.59223=0.591046		false	0.998		true	false	0.59223
false	true	0.002×0.183055=0.000366					false	true	0.183055
false	false	0.998×0.00113=0.001127					false	false	0.00113

- Factores finais:

B	$f_B(B)$
true	0.001
false	0.999

B	$f_{\bar{E}\bar{A}JM}(B)$
true	0.001197+0.591046=0.592243
false	0.000366+0.001127=0.001493

Algoritmo de eliminação de variáveis: exemplo



- Variável de Interrogação: *Burglary*
- Evidência: *JohnCalls=true*, *MaryCalls=true*
- Ordenação da variáveis (das folhas para a raíz): M, J, A, B, E
- Factores: $[f_B(B), f_{\bar{E}\bar{A}JM}(B,E)]$

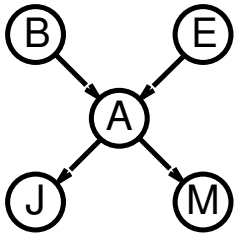
- Cálculos finais: $f_{B\bar{E}\bar{A}JM}(B) = f_B(B) \times f_{\bar{E}\bar{A}JM}(B)$

B	$f_{B\bar{E}\bar{A}JM}(B,E)$		B	$f_B(B)$		B	$f_{\bar{E}\bar{A}JM}(B)$
true	0.001×0.592243=0.000592243	=	true	0.001	×	true	0.592243
false	0.999×0.001493=0.001491858		false	0.999		false	0.001493

- Resultado: $P(B|JohnCalls=true, MaryCalls=false) = Normalize(f_{B\bar{E}\bar{A}JM}(B))$

B	$P(B\|J=true, M=false)$
true	0.000592243/(0.000592243+0.001491858) = **0.284172**
false	0.001491858/(0.000592243+0.001491858) = **0.715828**

Variáveis irrelevantes



- Considere-se a interrogação $P(JohnCalls|Burglary = true)$

$$\mathbf{P}(J \mid b) = \alpha P(b) \sum_e P(e) \sum_a P(a \mid b,e) \mathbf{P}(J \mid a) \sum_m P(m \mid a)$$

- A soma de m é igual 1; M é irrelevante para a interrogação

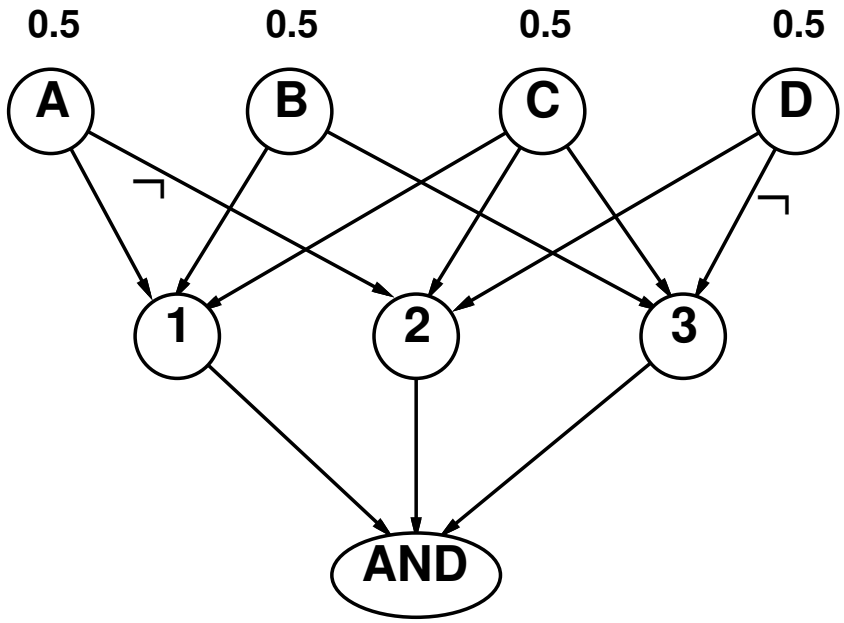
- **Teorema 1**: Y é irrelevante a não ser que $Y \in Ancestors(\{X\} \cup \mathbf{E})$

- Fazendo $X = JohnCalls$, $E = \{Burglary\}$, e $Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$ conclui-se que M é irrelevante
- (Compare-se com o método de encadeamento para trás a partir da interrogação em KBs de cláusulas de Horn)

Complexidade da inferência exacta

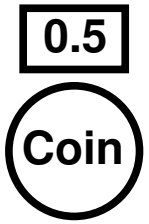
- Redes singularmente conexas (ou polytrees):
  - quaisquer dois nós estão ligados no máximo por um caminho (não dirigido)
  - complexidade temporal e espacial da eliminação de variáveis é $O(d^k n)$
- Redes multiplamente conexas:
  - pode-se reduzir 3SAT a inferência exacta ⇒ NP-difícil
  - equivalente contar modelos 3SAT ⇒ #P-completo

1. **A v B v C**
2. **C v D v ¬A**
3. **B v C v ¬D**



Inferência por simulação estocástica

- Ideia básica:
  1. Efectuar N amostras de uma distribuição de amostragem S
  2. Calcular um probabilidade à posteriori aproximada $\hat{P}$
  3. Mostrar que processo converge para a probabilidade real P
- Métodos:
  - Amostragem a partir de uma rede vazia
  - Amostragem por rejeição: rejeitar amostras que não estão de acordo com evidência
  - Pesagem por verosimilhança: utilizar evidência para pesar amostras
  - Markov chain Monte Carlo (MCMC): amostrar a partir de um processo estocástico cuja distribuição estacionária é a probabilidade à posteriori real



Amostragem a partir de uma rede vazia

function PRIOR-SAMPLE(bn) **returns** an event sampled from bn

inputs: bn, a belief network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

$\mathbf{x} \leftarrow$ an event with n elements

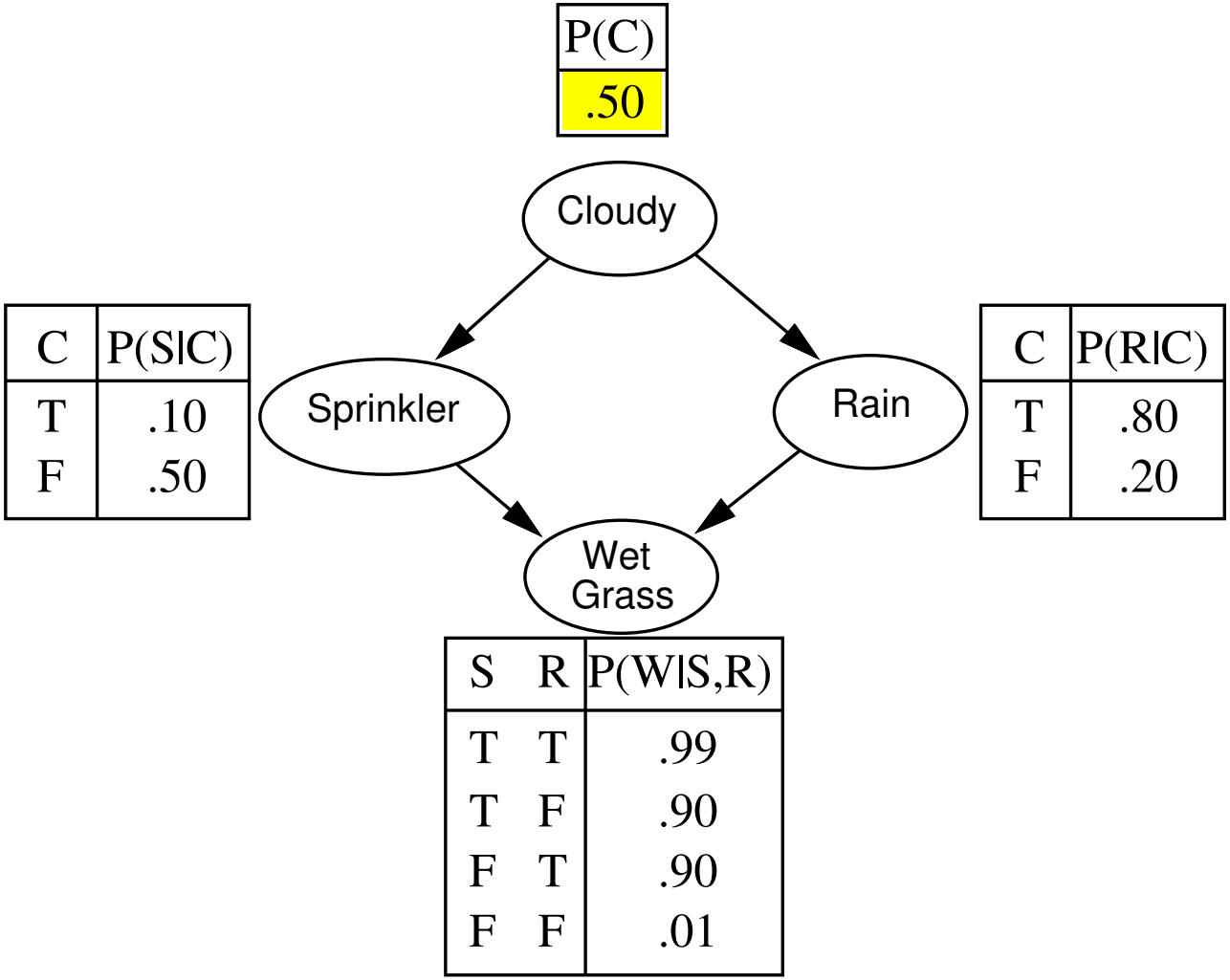
for $i = 1$ **to** n **do**

$x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$

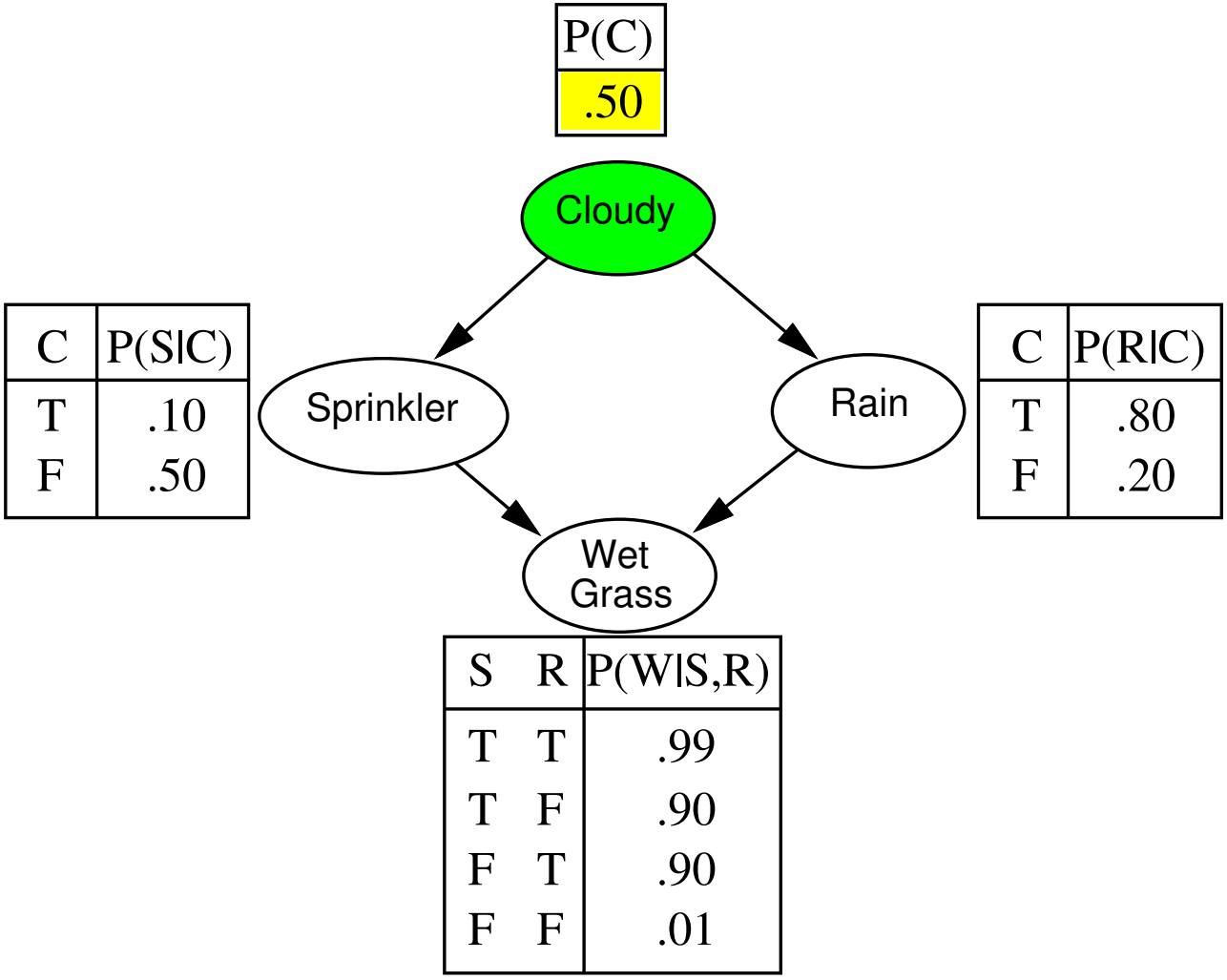
given the values of $Parents(X_i)$ in $\mathbf{x}$

return $\mathbf{x}$

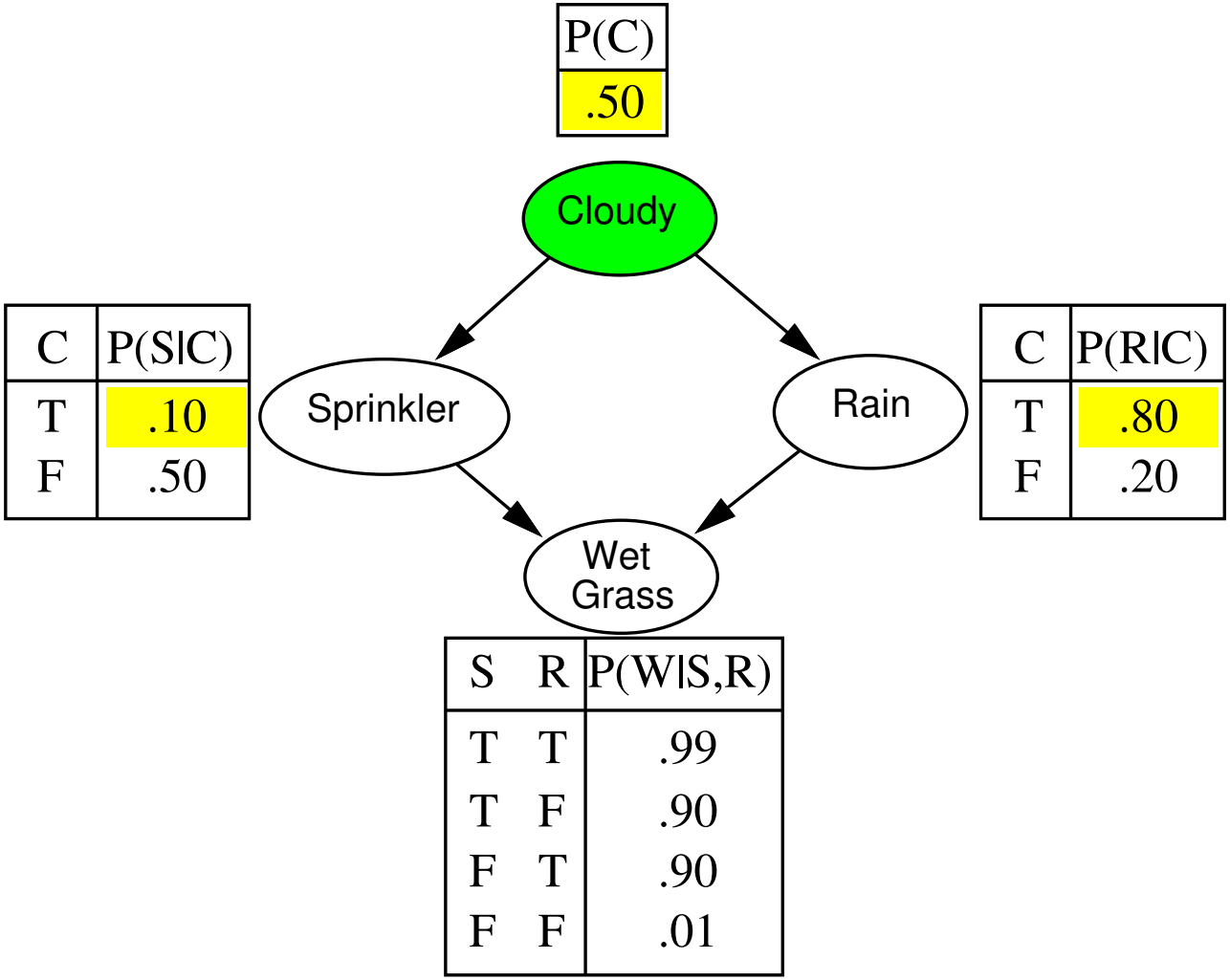
Exemplo


P(C)
.50
Cloudy
C P(S|C)
T .10
F .50
Sprinkler
Rain
C P(R|C)
T .80
F .20
Wet Grass
S R P(W|S,R)
T T .99
T F .90
F T .90
F F .01

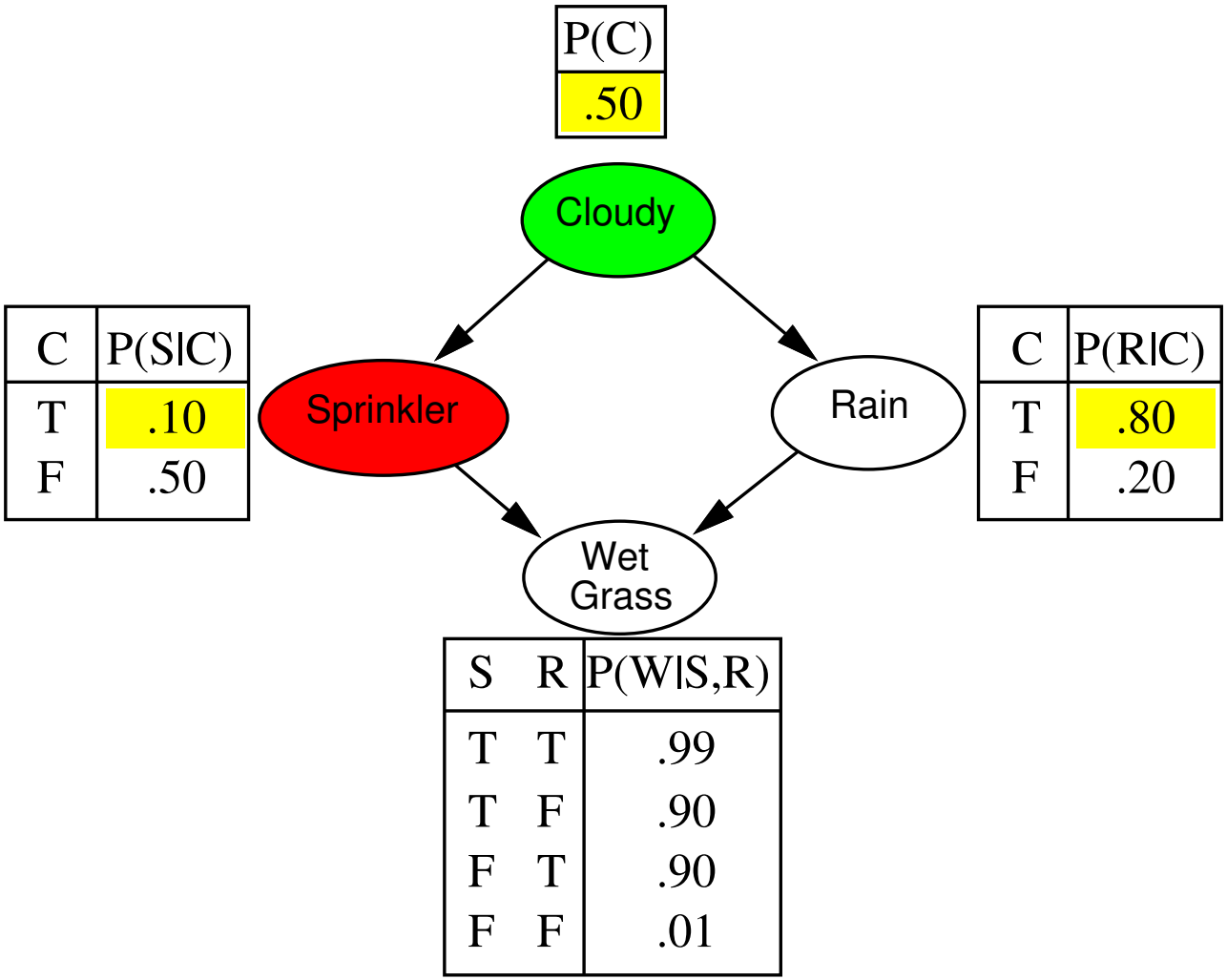
Exemplo


P(C)
.50
Cloudy
C
P(S|C)
T
.10
F
.50
Sprinkler
Rain
C
P(R|C)
T
.80
F
.20
Wet
Grass
S
R
P(W|S,R)
T
T
.99
T
F
.90
F
T
.90
F
F
.01

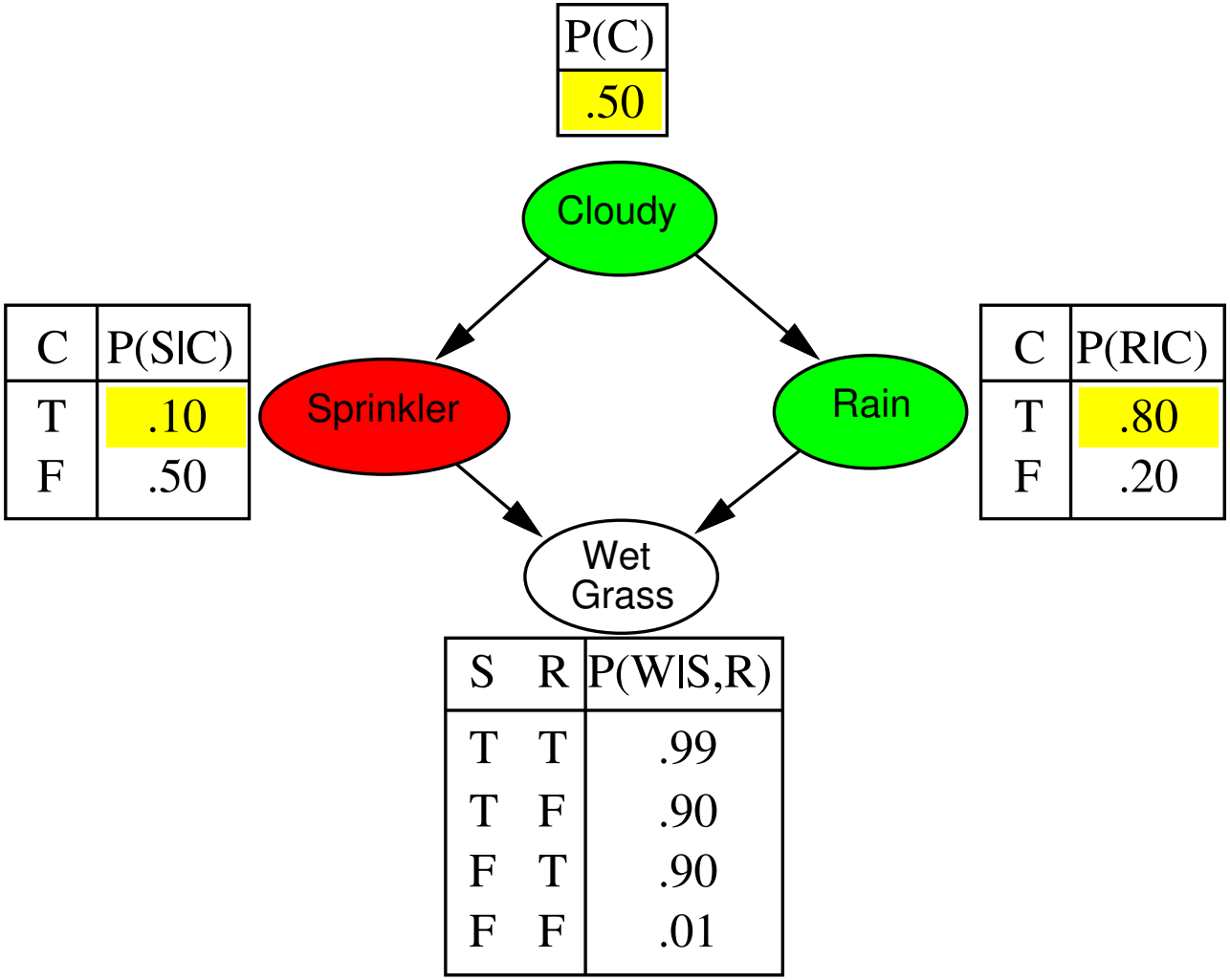
Exemplo


P(C)
.50
Cloudy
C
P(S|C)
T
.10
F
.50
Sprinkler
Rain
C
P(R|C)
T
.80
F
.20
Wet
Grass
S
R
P(W|S,R)
T
T
.99
T
F
.90
F
T
.90
F
F
.01

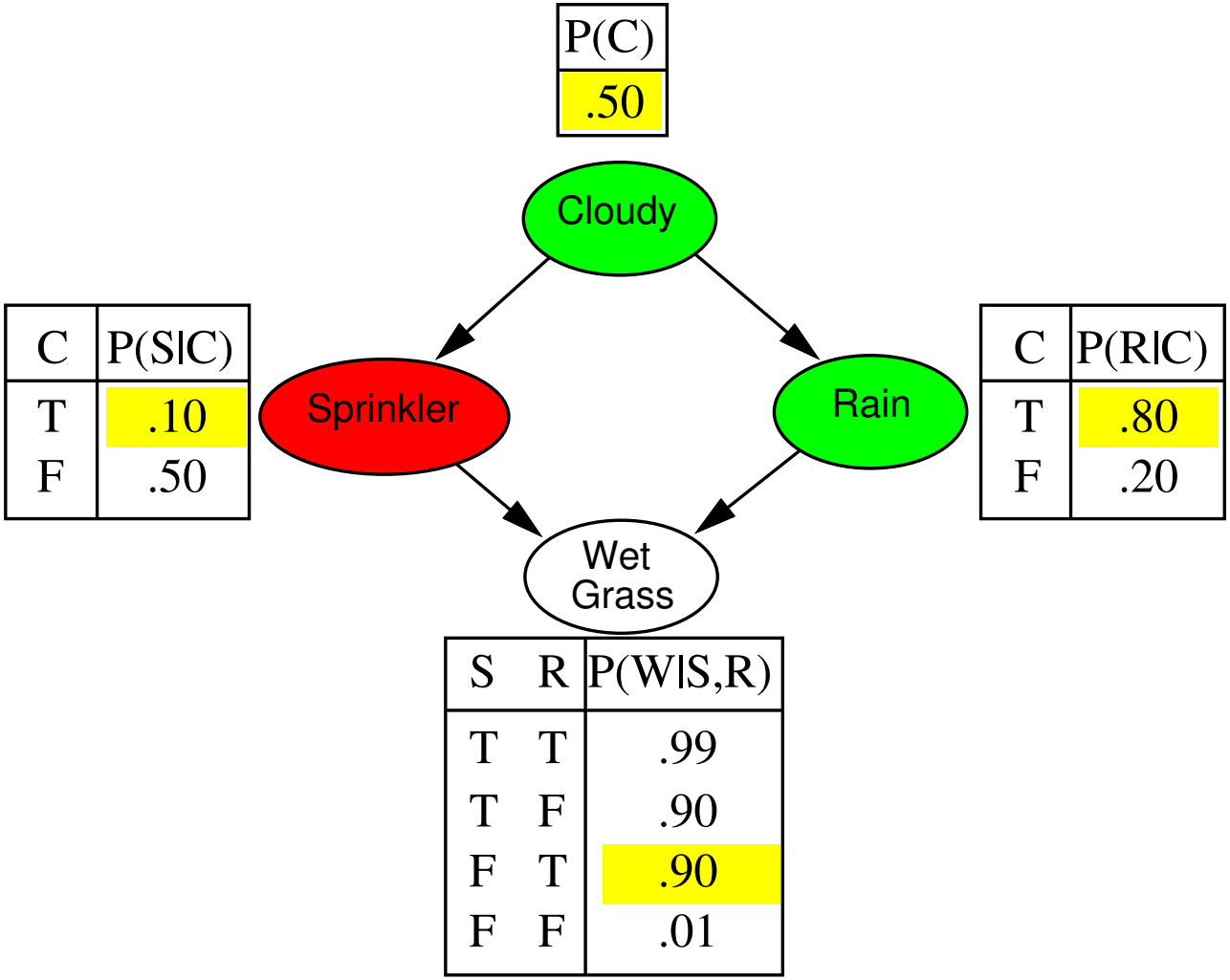
Exemplo


P(C)
.50
Cloudy
C
P(S|C)
T
.10
F
.50
Sprinkler
Rain
C
P(R|C)
T
.80
F
.20
Wet
Grass
S
R
P(W|S,R)
T
T
.99
T
F
.90
F
T
.90
F
F
.01

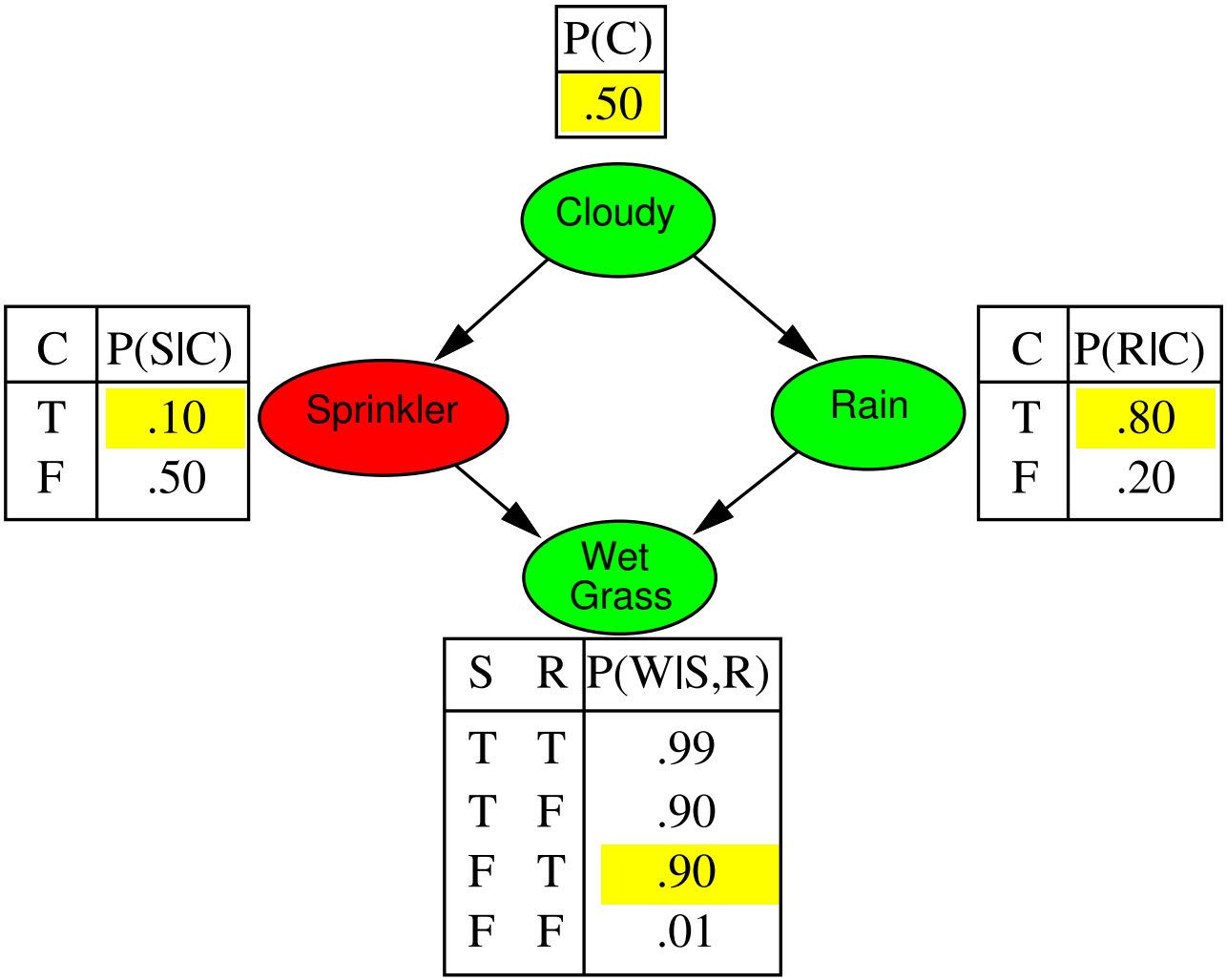
Exemplo


P(C)
.50
Cloudy
C
P(S|C)
T
.10
F
.50
Sprinkler
Rain
C
P(R|C)
T
.80
F
.20
Wet
Grass
S
R
P(W|S,R)
T
T
.99
T
F
.90
F
T
.90
F
F
.01

Exemplo


P(C)
.50
Cloudy
C
P(S|C)
T
.10
F
.50
Sprinkler
Rain
C
P(R|C)
T
.80
F
.20
Wet
Grass
S
R
P(W|S,R)
T
T
.99
T
F
.90
F
T
.90
F
F
.01

Exemplo


P(C)
.50
Cloudy
C
P(S|C)
T
.10
F
.50
Sprinkler
Rain
C
P(R|C)
T
.80
F
.20
Wet Grass
S
R
P(W|S,R)
T
T
.99
T
F
.90
F
T
.90
F
F
.01

Amostragem a partir de uma rede vazia

- Probabilidade de PRIORSAMPLE gerar um acontecimento particular

$$S_{PS}(x_1,...,x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(X_i)) = P(x_1,...,x_n)$$

- i.e., a probabilidade à priori real
- E.g., $S_{PS}(t,f,t,t) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324 = P(t,f,t,t)$
- Seja $N_{PS}(x_1,...,x_n)$ o número de amostras geradas para o acontecimento $x_1,...,x_n$
- Então obtemos

$$\lim_{N \to \infty} N_{PS}(x_1,...,x_n) / N = S_{PS}(x_1,...,x_n) = P(x_1,...,x_n)$$

- Ou seja, estimativas derivadas de PRIORSAMPLE são consistentes

$$P(x_1,...,x_n) \approx N_{PS}(x_1,...,x_n) / N = \hat{P}(x_1,...,x_n)$$

  - onde ≈ significa que a probabilidade estimada se torna exacta no limite de amostras grandes.

Amostragem por rejeição

- $\hat{\mathbf{P}}(X|\mathbf{e})$ estimado a partir das amostras que concordam com $\mathbf{e}$

```
function REJECTION-SAMPLING(X, e, bn, N) returns an estimate of P(X|e)
  local variables: N, a vector of counts over X, initially zero

  for j = 1 to N do
      x ← PRIOR-SAMPLE(bn)
      if x is consistent with e then
          N[x] ← N[x]+1 where x is the value of X in x
  return NORMALIZE(N[X])
```

- E.g., estimar *P(Rain|Sprinkler=true)* utilizando 100 amostras
  - 27 amostras têm *Sprinkler =true*
  - Destas, 8 têm *Rain=true* e 19 têm *Rain=false*.
- $\hat{\mathbf{P}}(Rain|Sprinkler = true) = Normalize(\langle 8,19\rangle) = \langle 0.296, 0.704\rangle$

Análise da amostragem por rejeição

$$\hat{\mathbf{P}}(X \mid \mathbf{e}) = \alpha \mathbf{N}_{PS}(X, \mathbf{e}) \qquad \text{(definição do algoritmo)}$$

$$= \mathbf{N}_{PS}(X, \mathbf{e}) / N_{PS}(\mathbf{e}) \qquad \text{(normalizado por } N_{PS}(\mathbf{e})\text{)}$$

$$\approx \mathbf{P}(X, \mathbf{e}) / P(\mathbf{e}) \qquad \text{(propriedade de PriorSample)}$$

$$= \mathbf{P}(X \mid \mathbf{e}) \qquad \text{(definição de probabilidade condicional)}$$

- Logo, amostragem por rejeição devolve estimativas consistentes da probabilidade à posteriori
- Problema: terrivelmente ineficiente se *P(e)* é pequeno
- *P(e)* diminui exponencialmente com o número de variáveis evidência!
- Este método é semelhante aos procedimentos empíricos de estimativa (estimação de probabilidades condicionais diretamente do mundo real):
  - E.g. Para estimar *P(Rain|RedskyAtNight=true)* basta contar a frequência com que chove quando na véspera esteve um céu avermelhado, ignorando os casos em que o céu não esteve dessa cor. Pode demorar muito tempo pois o céu raramente está avermelhado, sendo esta uma das fraquezas deste método.

Pesagem por verosimilhança

- **Ideia**: fixar variáveis evidência, amostrar apenas as outras variáveis, e pesar cada amostra pela verosimilhança de acordo com a evidência

function LIKELIHOOD-WEIGHTING($X, \mathbf{e}, bn, N$) **returns** an estimate of $P(X|\mathbf{e})$
  local variables: $\mathbf{W}$, a vector of weighted counts over X, initially zero

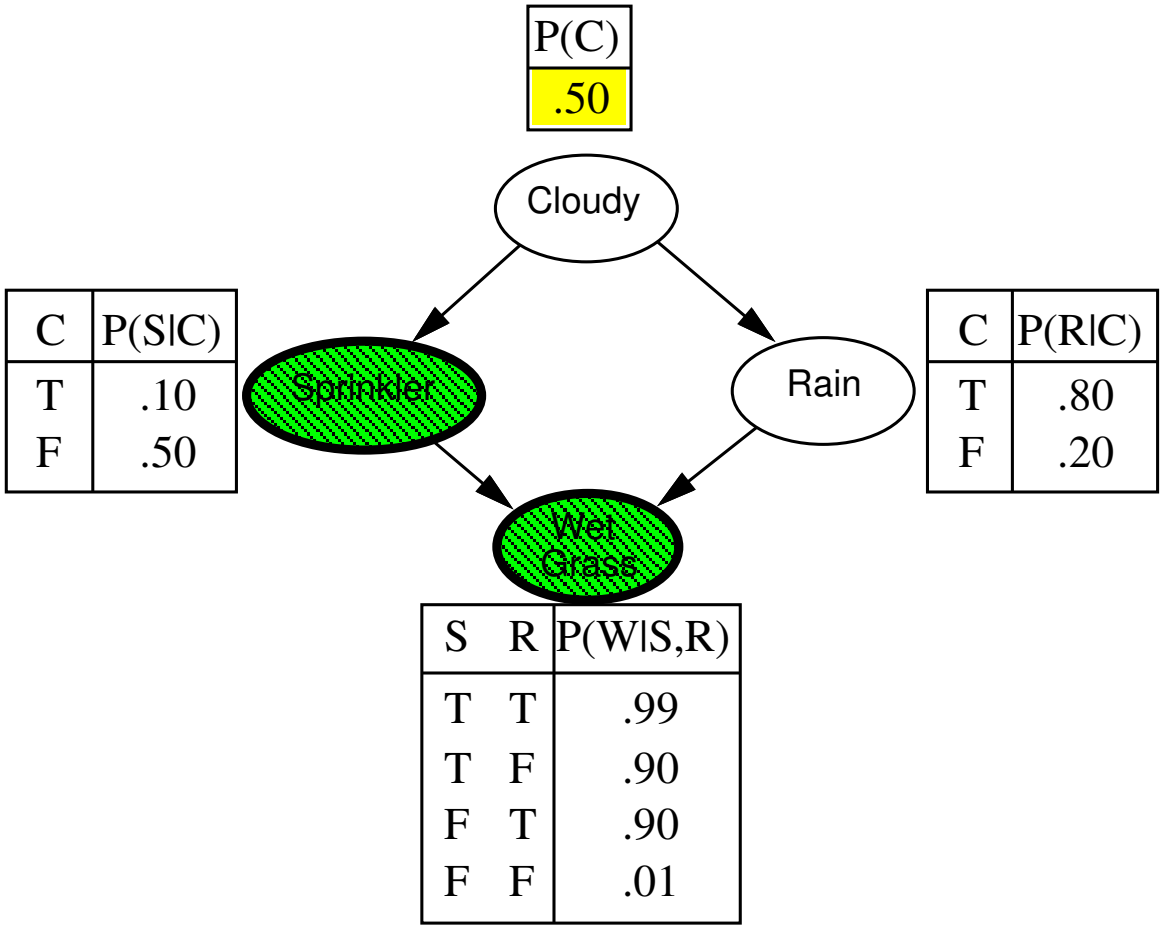
  for $j = 1$ to N **do**
    $\mathbf{x}, w \leftarrow$ WEIGHTED-SAMPLE(bn)
    $\mathbf{W}[x] \leftarrow \mathbf{W}[x] + w$ where x is the value of X in $\mathbf{x}$
  return NORMALIZE($\mathbf{W}[X]$)

---

function WEIGHTED-SAMPLE($bn, \mathbf{e}$) **returns** an event and a weight

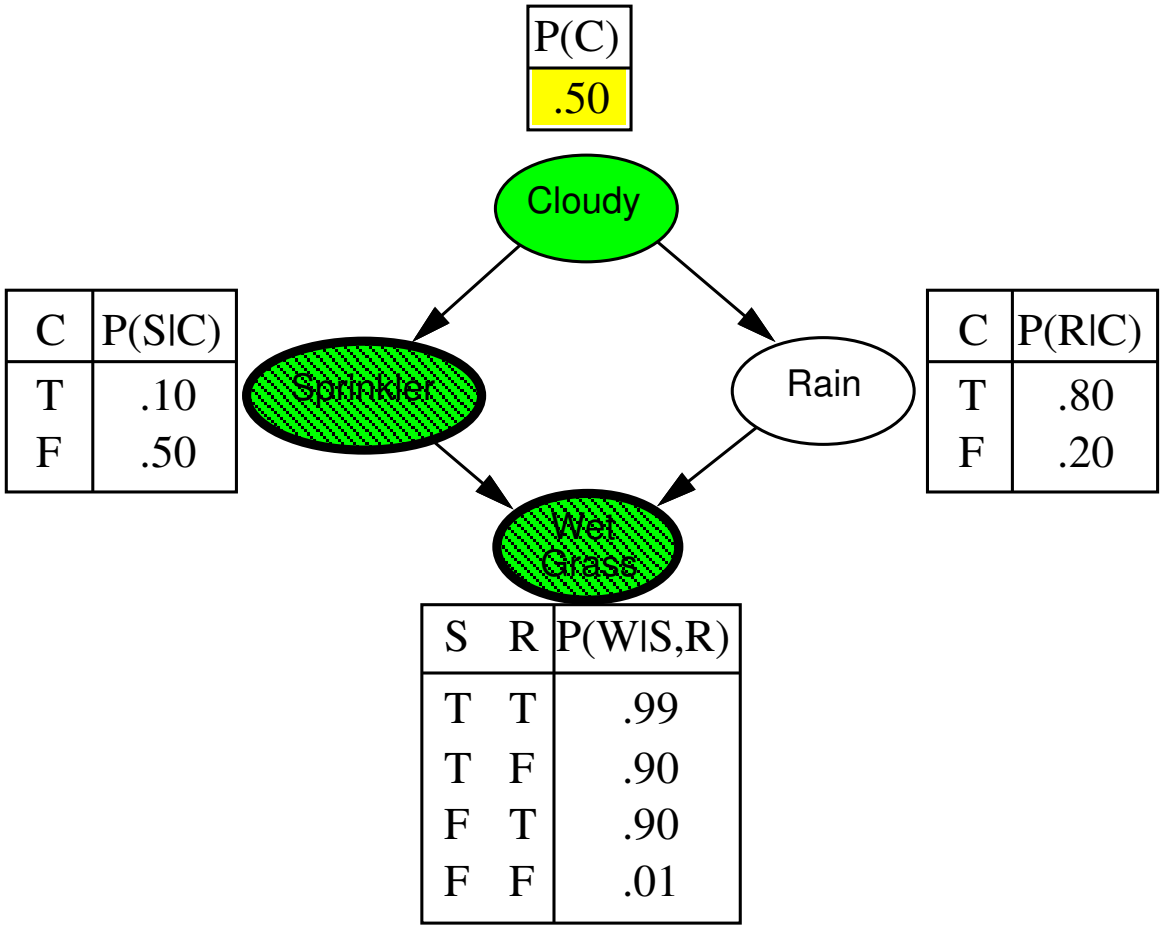
  $\mathbf{x} \leftarrow$ an event with n elements; $w \leftarrow 1$
  for $i = 1$ **to** n **do**
    if X_i has a value x_i in $\mathbf{e}$
      then $w \leftarrow w \times P(X_i = x_i \mid parents(X_i))$
      else $x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return $\mathbf{x}$, w

Exemplo de pesagem por verosimilhança



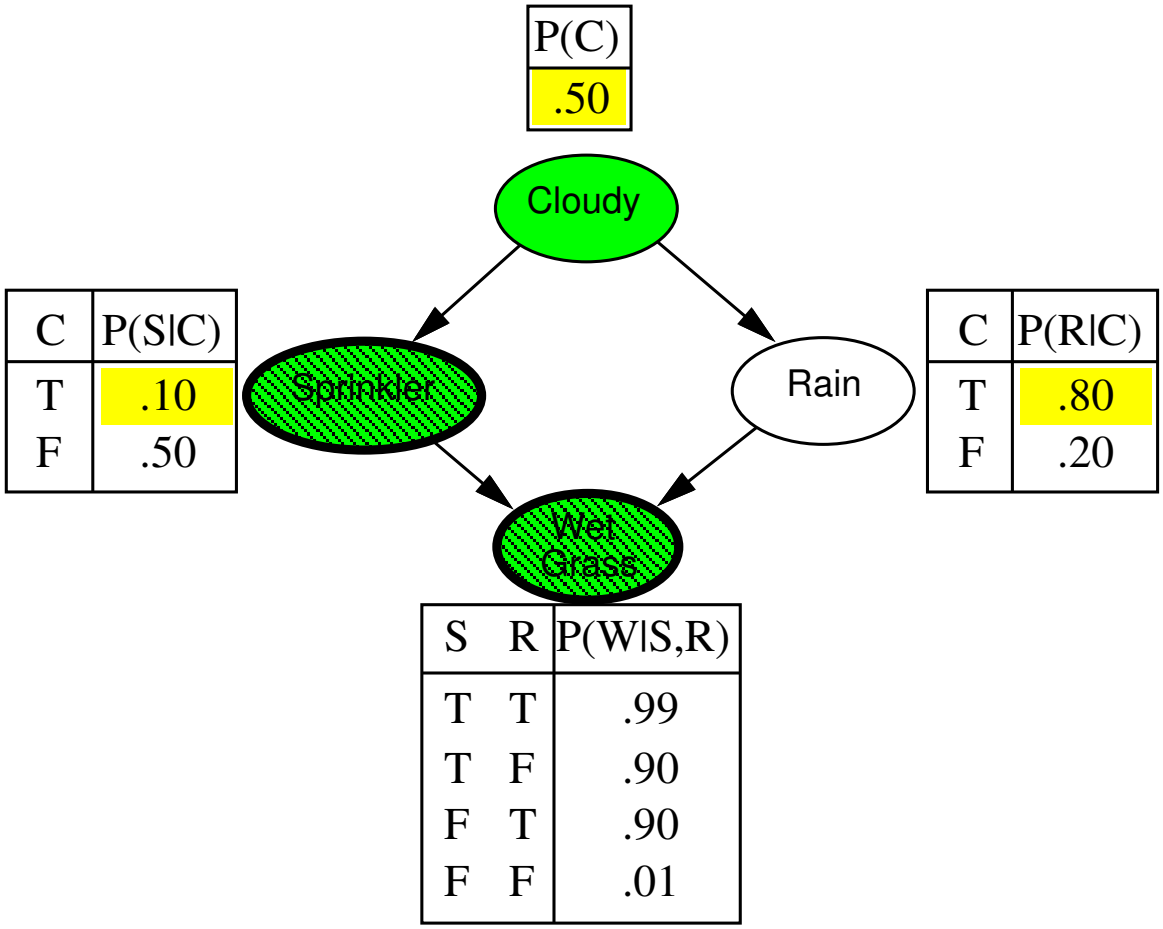
$w = 1.0$

Exemplo de pesagem por verosimilhança



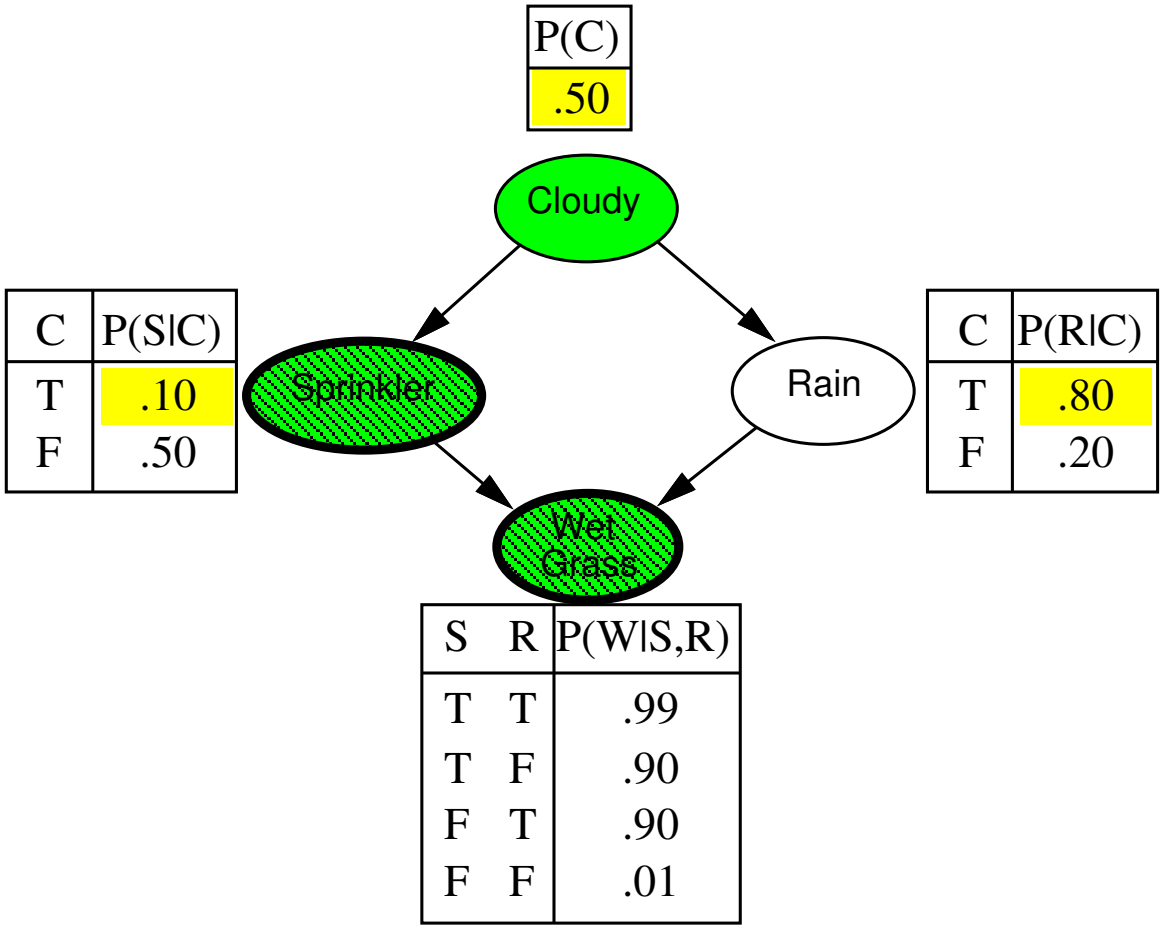
$w = 1.0$

Exemplo de pesagem por verosimilhança



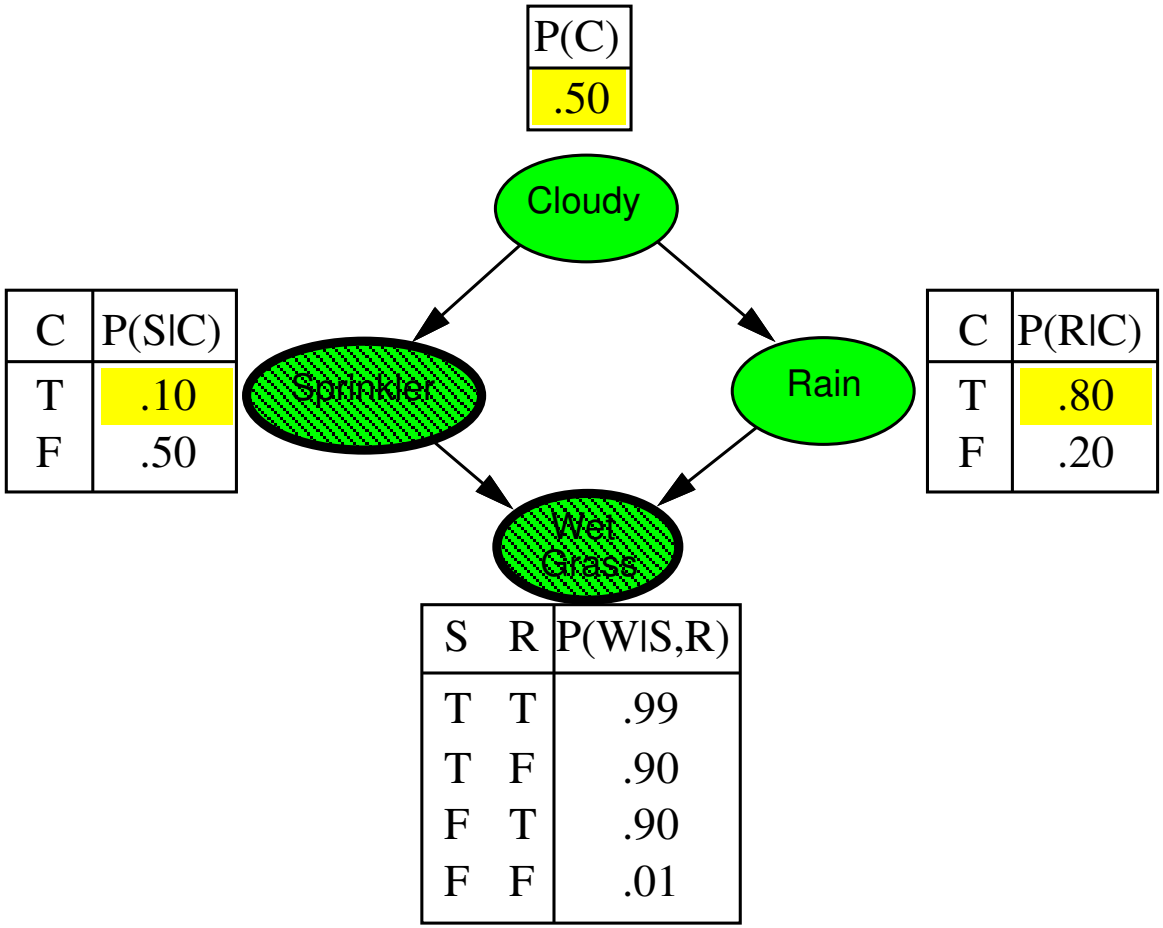
$w = 1.0$

Exemplo de pesagem por verosimilhança



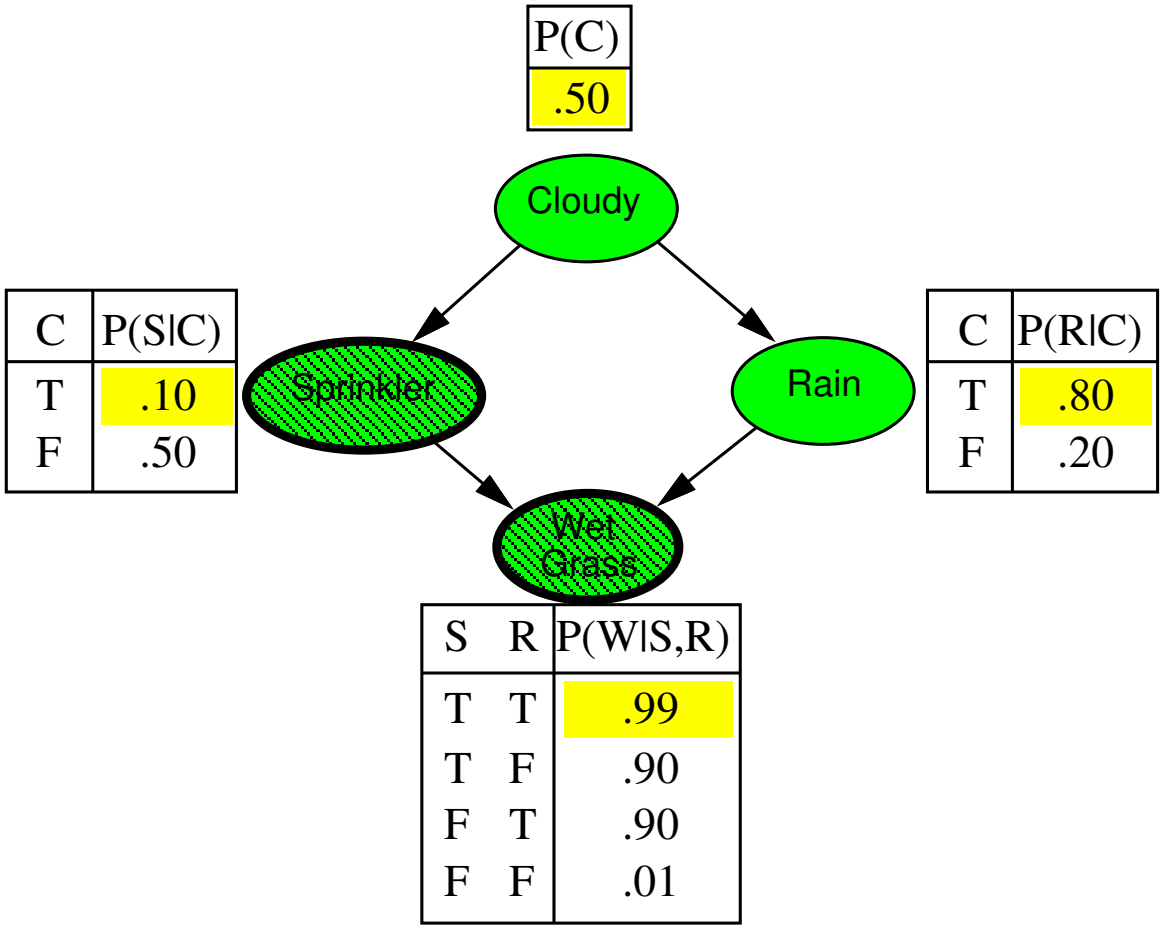
$w = 1.0 \times 0.1$

Exemplo de pesagem por verosimilhança



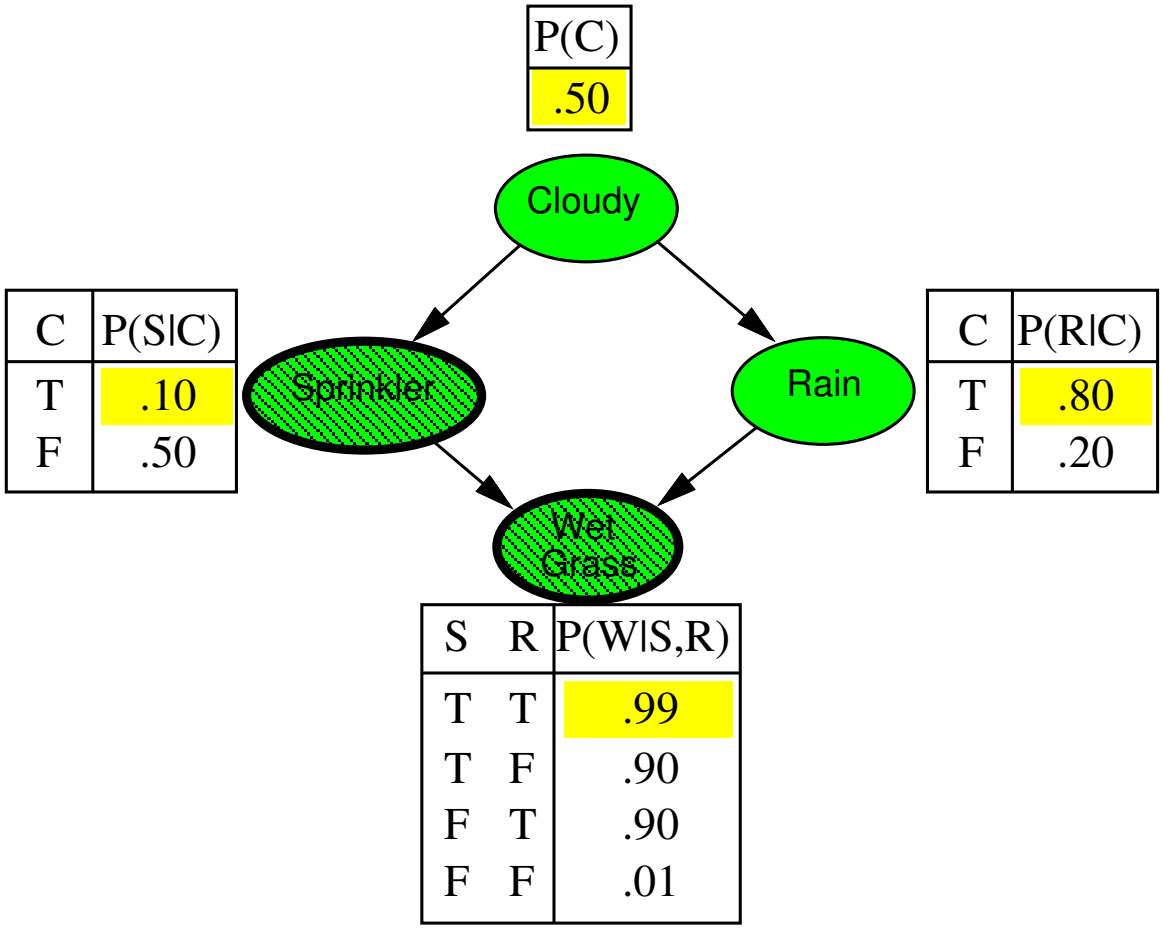
$w = 1.0 \times 0.1$

Exemplo de pesagem por verosimilhança



$w = 1.0 \times 0.1$

Exemplo de pesagem por verosimilhança



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

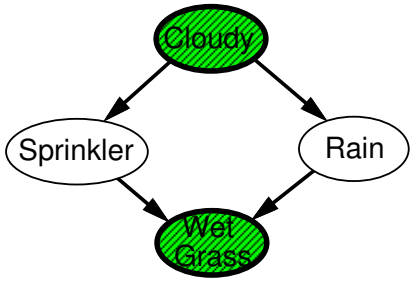
Análise de pesagem por verosimilhança

- Probabilidade de amostragem de WeightedSample é

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

- Nota: entra em conta apenas com a evidência dos antecessores
  - Algures "entre" a distribuição à priori e a à posteriori
- Peso para uma dada amostra $\mathbf{z}$, $\mathbf{e}$ é

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



- Probabilidade de amostragem pesada é

$$S_{WS}(\mathbf{z},\mathbf{e})\,w(\mathbf{z},\mathbf{e}) = \prod_{i=1}^{l} P(z_i \mid Parents(Z_i)) \prod_{i=1}^{m} P(e_i \mid Parents(E_i))$$
$$= P(\mathbf{z},\mathbf{e}) \quad \text{(pela semântica global da rede)}$$

- Portanto a pesagem por verosimilhança retorna estimativas consistentes.
  - Mais eficiente do que amostragem por rejeição, pois usa todas as amostras
  - Mas o desempenho continua a degradar-se com muitas variáveis evidência porque algumas (poucas) amostras têm quase todo o peso

Markov Chain Monte Carlo (MCMC)

- "Estado" da rede = atribuição corrente a todas as variáveis.
- Gerar o estado seguinte amostrando uma variável dado a sua cobertura de Markov (Markov blanket)
- Altera-se uma variável de cada vez, por amostragem, mantendo a evidência.

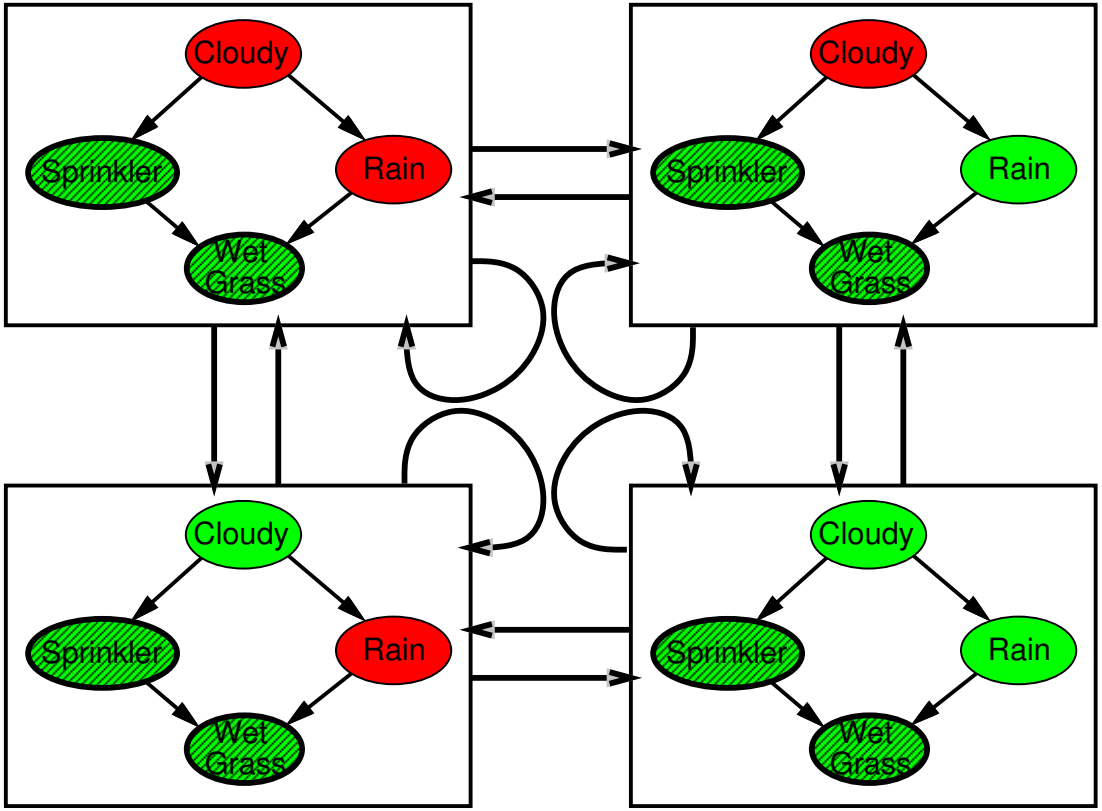
function MCMC-ASK($X, \mathbf{e}, bn, N$) **returns** an estimate of $P(X|\mathbf{e})$
  local variables: $\mathbf{N}[X]$, a vector of counts over X, initially zero
    $\mathbf{Z}$, the nonevidence variables in bn
    $\mathbf{x}$, the current state of the network, initially copied from $\mathbf{e}$

  initialize $\mathbf{x}$ with random values for the variables in $\mathbf{Y}$
  for $j = 1$ to N **do**
    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 $\mathbf{x}$
      $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$ where x is the value of X in $\mathbf{x}$
  return NORMALIZE($\mathbf{N}[X]$)

- Pode-se também escolher aleatoriamente a variável a amostrar

A cadeia de Markov

- Com *Sprinkler =true*, *WetGrass=true*, existem quatro estados:



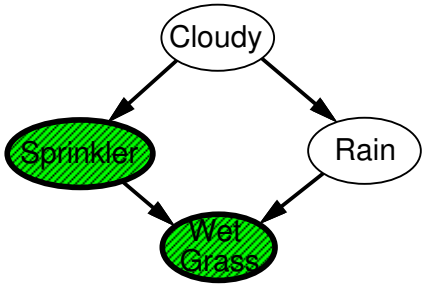
- Vagueia durante algum tempo, efectua a média do que observa

MCMC: exemplo

- Estimar $P(Rain|Sprinkler = true, WetGrass = true)$
- Amostrar $Cloudy$ ou $Rain$ dado o seu Markov blanket, repetir.
- Contar o numero de vezes que $Rain$ é verdadeiro e falso nas amostras.
  - E.g., visita 100 estados, 31 tem $Rain = true$, 69 tem $Rain = false$
- $\hat{\mathbf{P}}(Rain|Sprinkler = true, WetGrass = true) = Normalize(\langle 31, 69 \rangle) = \langle 0.31, 0.69 \rangle$
- **Teorema**: cadeia aproxima-se da distribuição estacionária: a fração de tempo gasto em cada espaço é exactamente proporcional à sua probabilidade à posteriori

Amostragem da Cobertura de Markov

- Cobertura de Markov de *Cloudy* é:
  - *Sprinkler* e *Rain*
- Cobertura de Markov de *Rain* é:
  - *Cloudy*, *Sprinkler*, e *WetGrass*



- Probabilidade dada a Cobertura de Markov (MB) é obtida como se segue:

$$P\left(x_i' \mid MB\left(X_i\right)\right) = P\left(x_i' \mid Parents\left(X_i\right)\right)\prod_{z_j \in Children\left(X_i\right)} P\left(z_j \mid Parents\left(Z_j\right)\right)$$

- Problemas computacionais principais:
  - Dificuldade em detectar que se atingiu a convergência
  - Pode ser dispendioso se Cobertura de Markov é grande:
    - $P(X_i|MB(X_i))$ não varia muito

Sumário

- Inferência exacta por eliminação de variáveis:
  - tempo polinomial em polytrees, NP-difícil em grafos arbitrários
  - espaço = tempo, muito sensível à topologia
- Inferência aproximada por PV, MCMC:
  - PV comporta-se mal quando existe muita evidência
  - PV, MCMC geralmente insensíveis à topologia
  - Convergência pode ser muito lenta com probabilidades perto de 0 ou 1
  - Pode tratar combinações arbitrárias de variáveis discretas e contínuas