# Inferenza in Reti Bayesiane

## Corso di Sistemi Intelligenti, Capitolo 14.4–5

# Outline

- ♢ Inferenza esatta tramite enumerazione
- ♢ Inferenza esatta tramite eliminazione di variabile
- ♢ Inferenza approssimata tramite simulazione stocastica
- ♢ Inferenza approssimata tramite Markov chain Monte Carlo

# Compiti di inferenza

Query semplici: calcolare la probabilità a posteriori marginale $\mathbf{P}(X_i|\mathbf{E}=\mathbf{e})$
p.e., $P(NoGas|Gauge=empty, Lights=on, Starts=false)$

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

Decisioni ottimali: reti di decisioni includono informazioni di utilità;
inferenza probabilistica richiesta per $P(outcome|action, evidence)$

Recupero informazione: quale evidenza si deve cercare?

Analisi della sensitività: quali valori di probabilità sono i più critici?

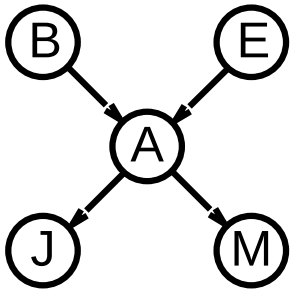
Spiegazione: perché ho bisogno di un nuovo motore di avviamento?

# Inferenza tramite enumerazione

Modo un pò più furbo per marginalizzare alcune variabili dalla distribuzione congiunta senza costruire esplicitamente la sua rappresentazione

Query semplice sulla rete dell'allarme:

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




Riscrittura di entrate della distribuzione congiunta usando il prodotto di entrate di CPT:

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

Enumerazione ricorsiva depth-first: $O(n)$ in spazio, $O(d^n)$ in tempo

# Algoritmo di enumerazione

**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($\mathbf{Q}(X)$)

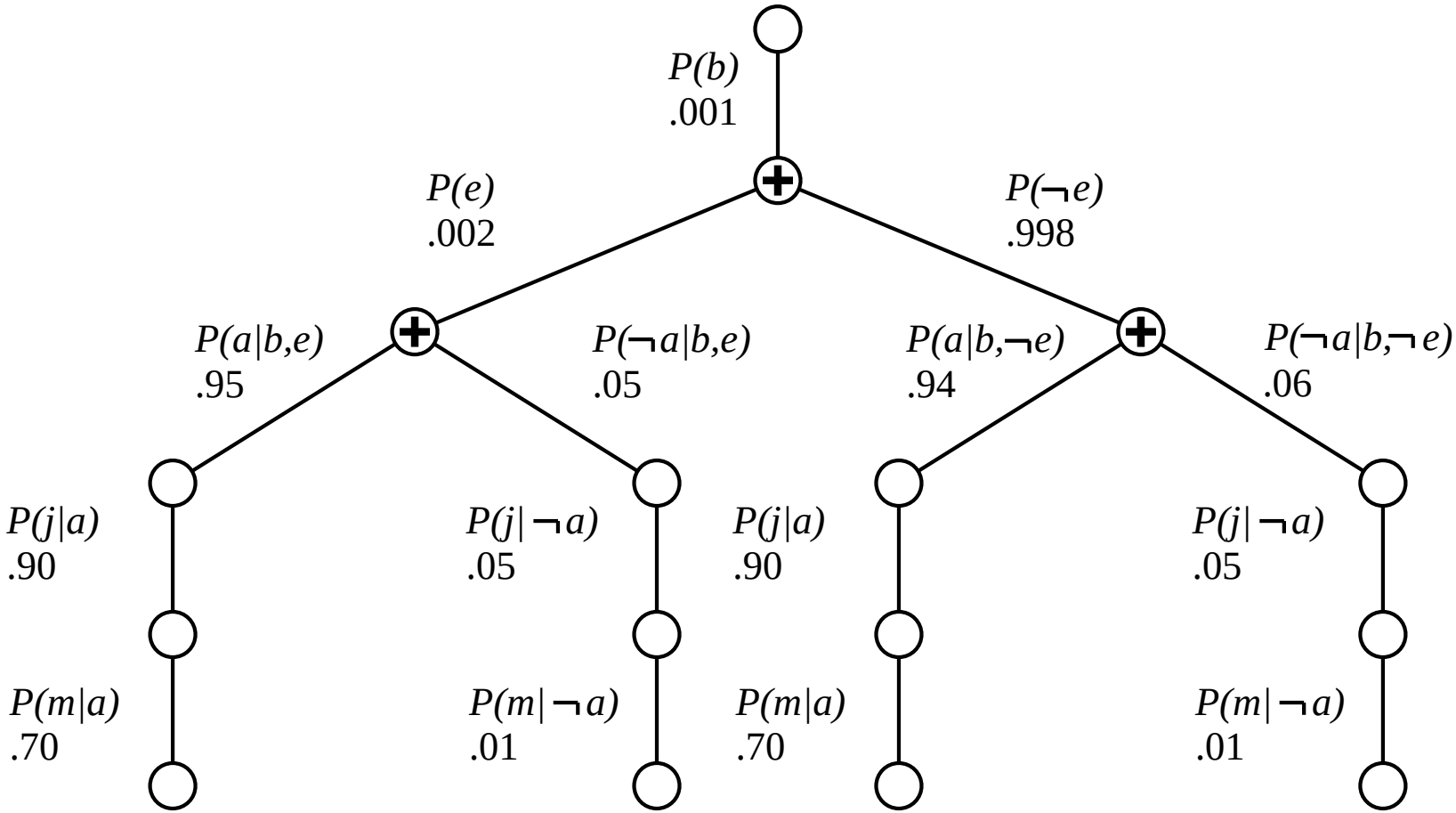
---

**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 $Y = y$

# Albero di valutazione

L'enumerazione è inefficiente: calcoli ripetuti

p.e., calcola $P(j|a)P(m|a)$ per ogni valore di $e$



# Inferenza tramite eliminazione di variabile

Eliminazione di variabile: effettuare le somme da destra a sinista, memorizzare i risultati intermedi (fattori) per evitare di ricalcolarli

$$
\begin{aligned}
&\mathbf{P}(B|j,m) \\
&\quad = \alpha \underbrace{\mathbf{P}(B)}_{B} \Sigma_e \underbrace{P(e)}_{E} \Sigma_a \underbrace{\mathbf{P}(a|B,e)}_{A} \underbrace{P(j|a)}_{J} \underbrace{P(m|a)}_{M} \\
&\quad = \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a \mathbf{P}(a|B,e) P(j|a) f_M(a) \\
&\quad = \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a \mathbf{P}(a|B,e) f_J(a) f_M(a) \\
&\quad = \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a f_A(a,b,e) f_J(a) f_M(a) \\
&\quad = \alpha \mathbf{P}(B) \Sigma_e P(e) f_{\bar{A}JM}(b,e) \text{ (elimina } A\text{)} \\
&\quad = \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (elimina } E\text{)} \\
&\quad = \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
\end{aligned}
$$

# Eliminazione di variabile: operazioni base

Eliminare una variabile da un prodotto di fattori:

1. muovere i fattori costanti al di fuori della somma
2. aggiungere le sottomatrici al prodotto "pointwise" dei fattori rimanenti

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

assumendo che $f_1, \ldots, f_i$ non dipendano da $X$

Prodotto pointwise di fattori $f_1$ e $f_2$:

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

P.e., $f_1(a, b) \times f_2(b, c) = f(a, b, c)$

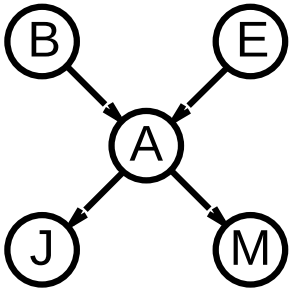
# Algoritmo di eliminazione di variabile

**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**
    $factors \leftarrow [\text{MAKE-FACTOR}(var, \mathbf{e}) | factors]$
    **if** $var$ is a hidden variable **then** $factors \leftarrow$ SUM-OUT($var, factors$)
  **return** NORMALIZE(POINTWISE-PRODUCT($factors$))

# Variabili irrilevanti

Consideriamo la query $P(JohnCalls|Burglary = true)$




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

La somma su $m$ è uguale a 1; $M$ è **irrilevante** per la query

Thm 1: $Y$ è irrilevante a meno che $Y \in Ancestors(\{X\} \cup \mathbf{E})$

Qui, $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$, e
$Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$
quindi $M$ è irrilevante

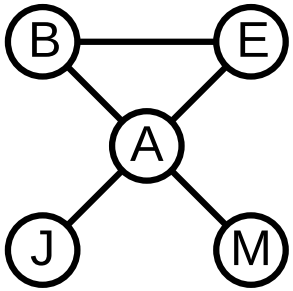
(Confrontare con backward chaining a partire dalla query in KB con clausole di Horn)

# Variabili irrilevanti

Defn: grafo moralizzato di una rete bayesiana: sposare tutti i genitori ed eliminare la direzione degli archi

Defn: $\mathbf{F}$ è m-separato da $\mathbf{G}$ tramite $\mathbf{H}$ sse è separato tramite $\mathbf{H}$ nel grafo moralizzato

Thm 2: $Y$ è irrilevante se m-separato da $X$ tramite $\mathbf{E}$




Per $P(JohnCalls|Alarm = true)$, sia
$Burglary$ che $Earthquake$ sono irrilevanti

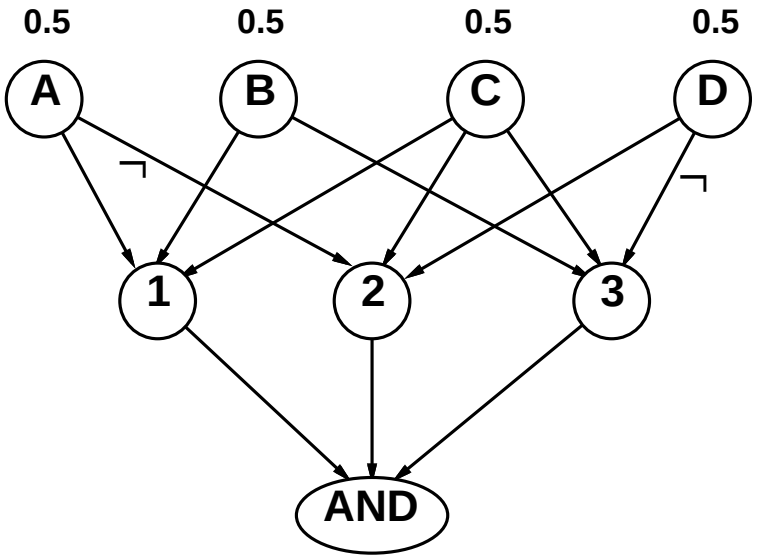
# Complessità dell'inferenza esatta

Reti singolarmente connesse (o polytree):
- ogni coppia di nodi è connessa da al più un cammino (non diretto)
- il costo in tempo e spazio della eliminazione di variabile è $O(d^k n)$

Reti connesse più che singolarmente:
- possibile ridurre 3SAT alla inferenza esatta $\Rightarrow$ NP-hard
- equivalente a modelli 3SAT con **conteggio** (del numero di soluzioni)

$\Rightarrow$ #P-complete

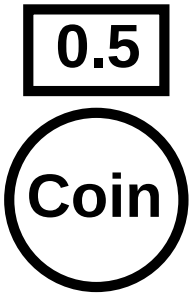
1. A v B v C
2. C v D v ¬A
3. B v C v ¬D



# Inferenza tramite simulazione stocastica

Idea base:

1) Estrarre $N$ campioni da una distribuzione di campionamento $S$
2) Calcolare la probabilità a posteriori approssimata $\hat{P}$
3) Mostrare che converge alla vera probabilità $P$




Outline:

- Campionamento da una rete vuota
- Rejection sampling: rigettare i campioni in disaccordo con l'evidenza
- Likelihood weighting: usare l'evidenza per pesare i campioni
- Markov chain Monte Carlo (MCMC): campiona in accordo ad un processo stocastico la cui distribuzione stazionaria è la vera probabilità

# Campionamento da una rete vuota

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

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

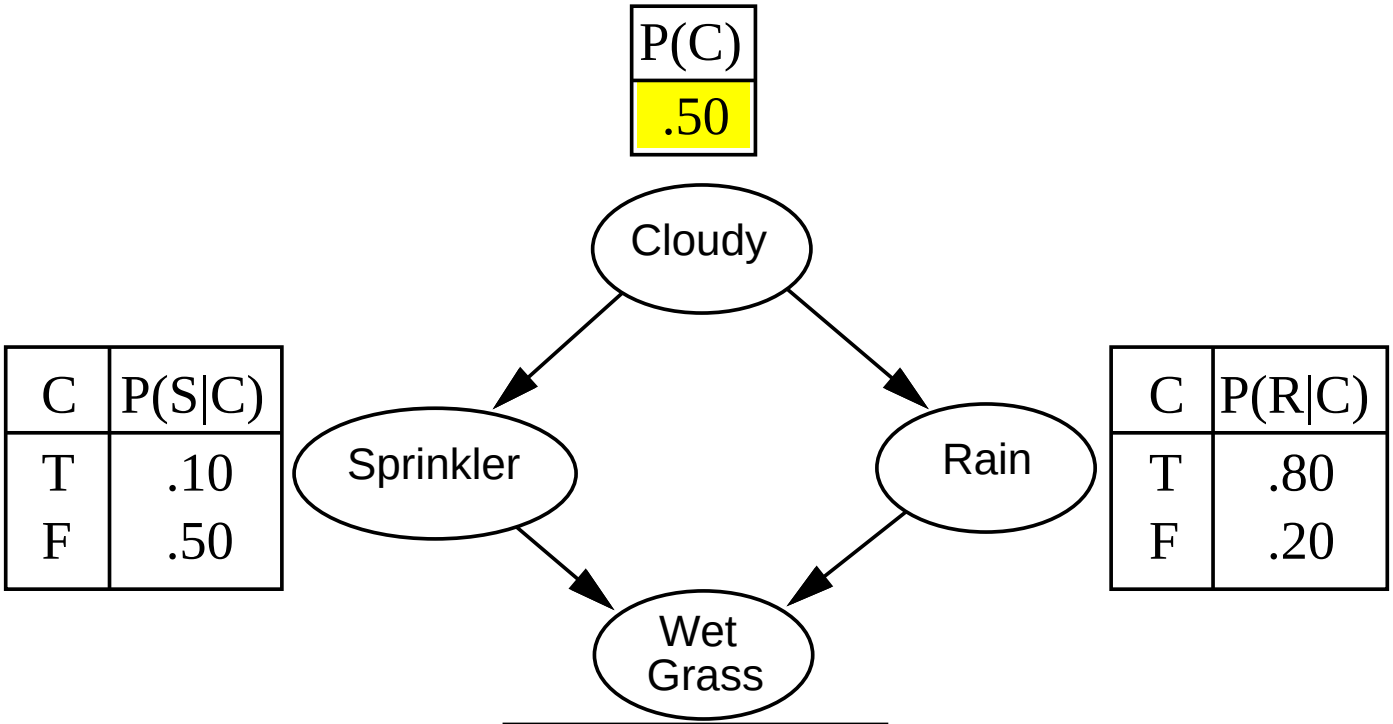
$\mathbf{x} \leftarrow$ an event with $n$ elements

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

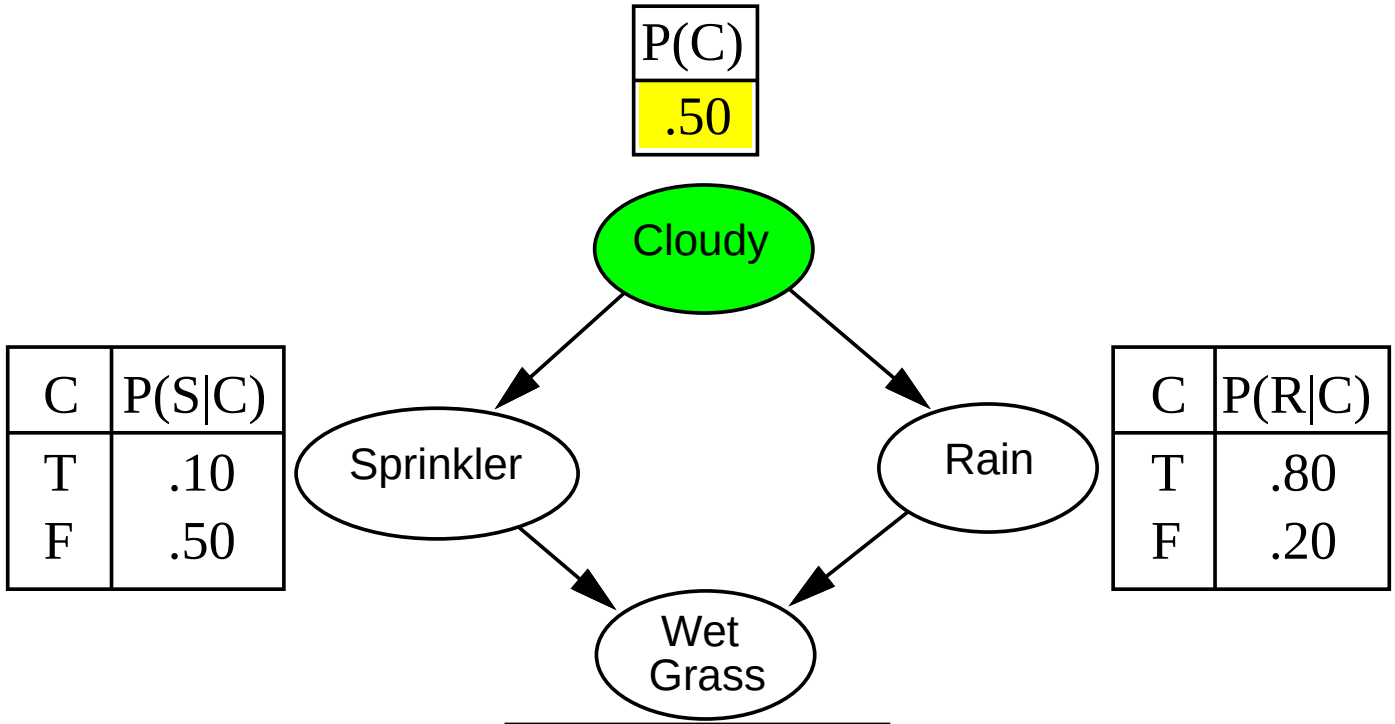
$x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid Parents(X_i))$

**return** $\mathbf{x}$

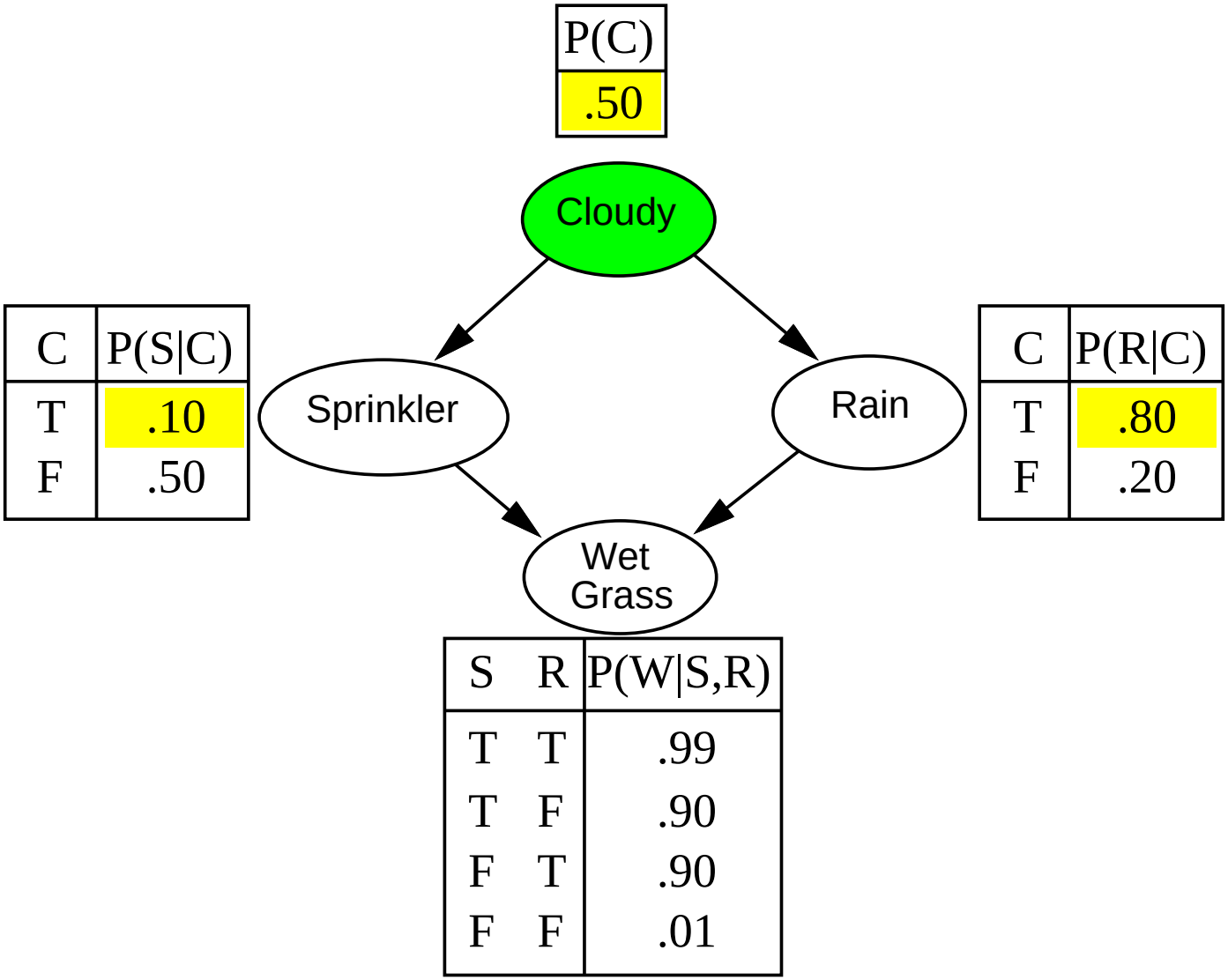
# Esempio


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

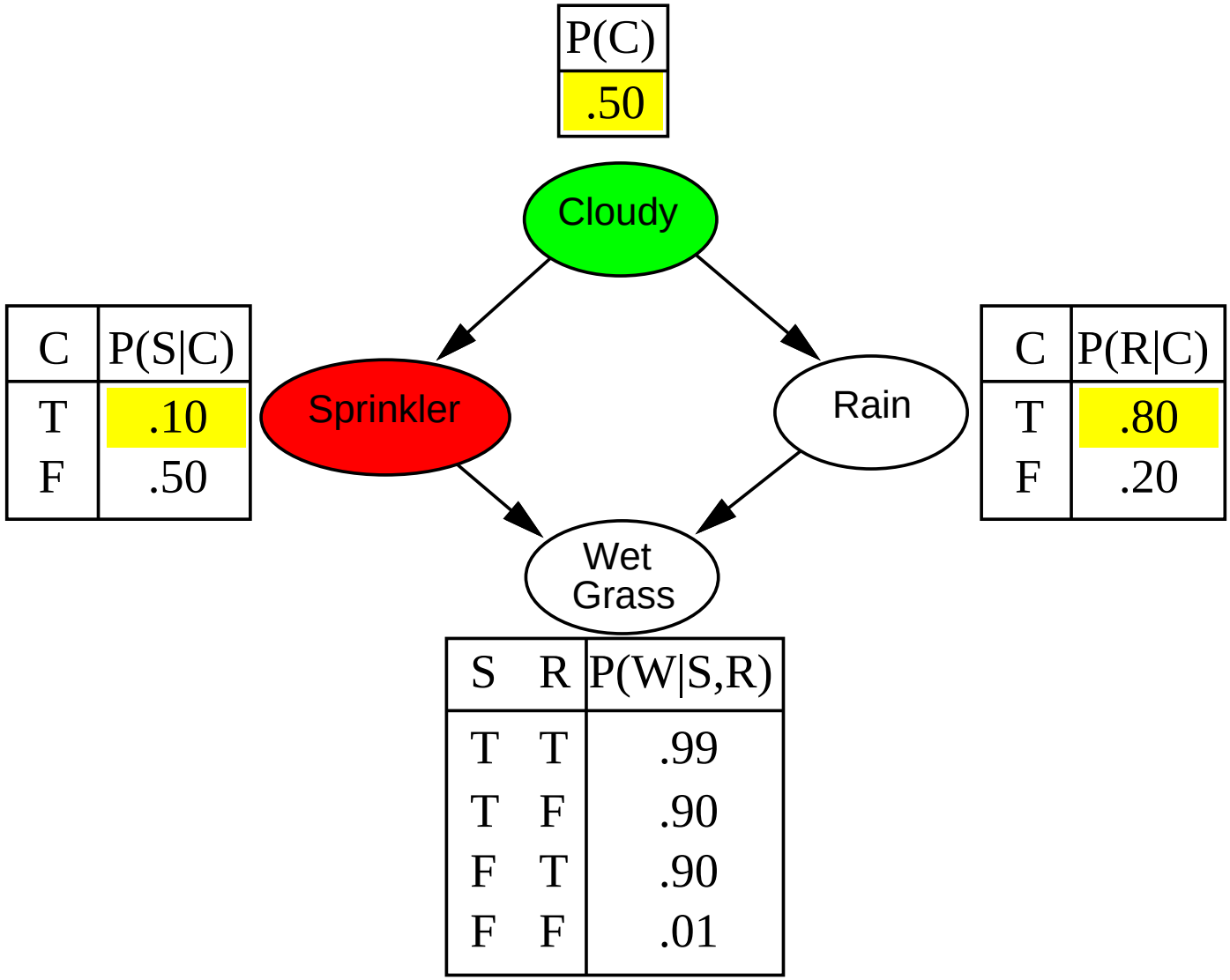
# Esempio


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

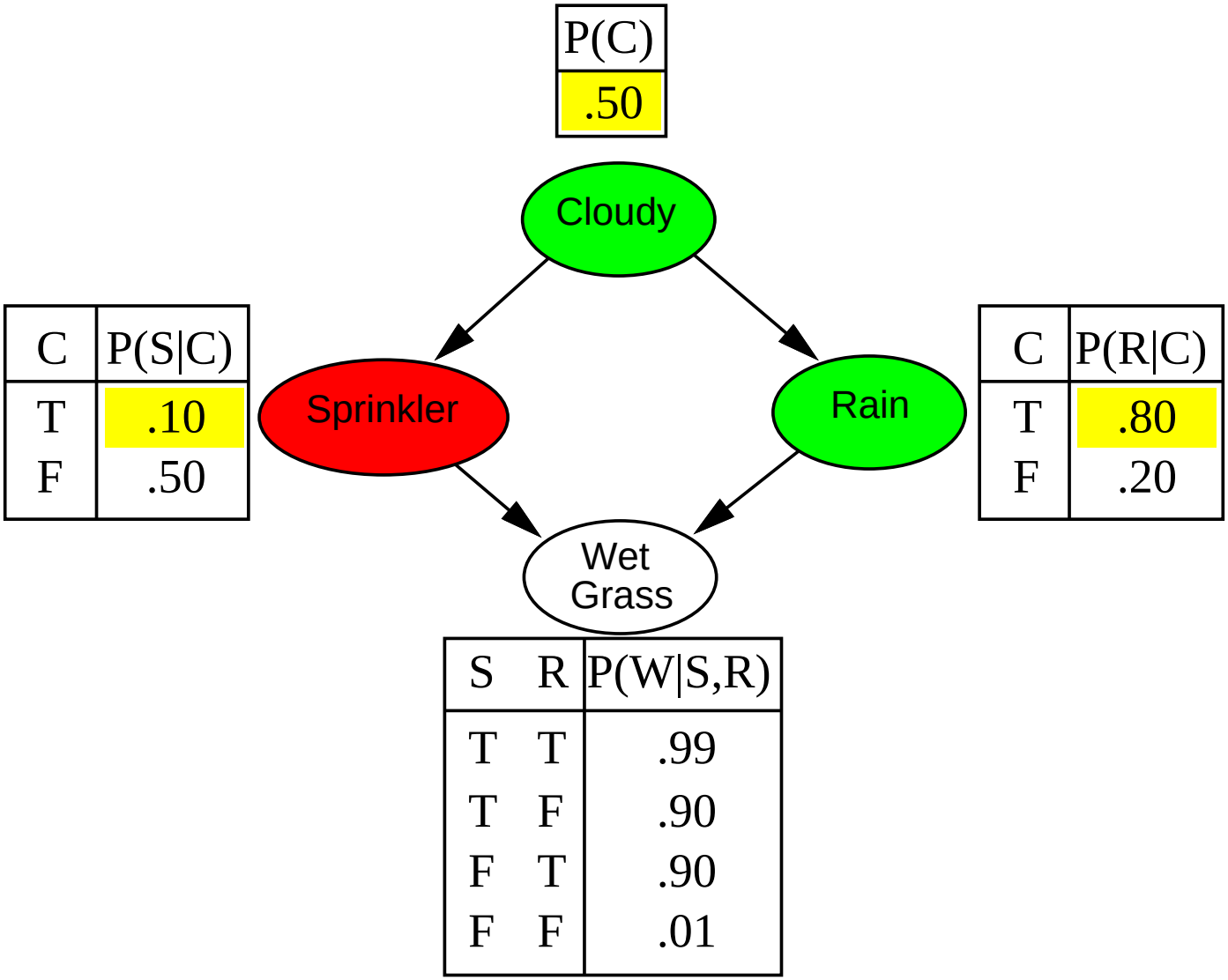
# Esempio


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

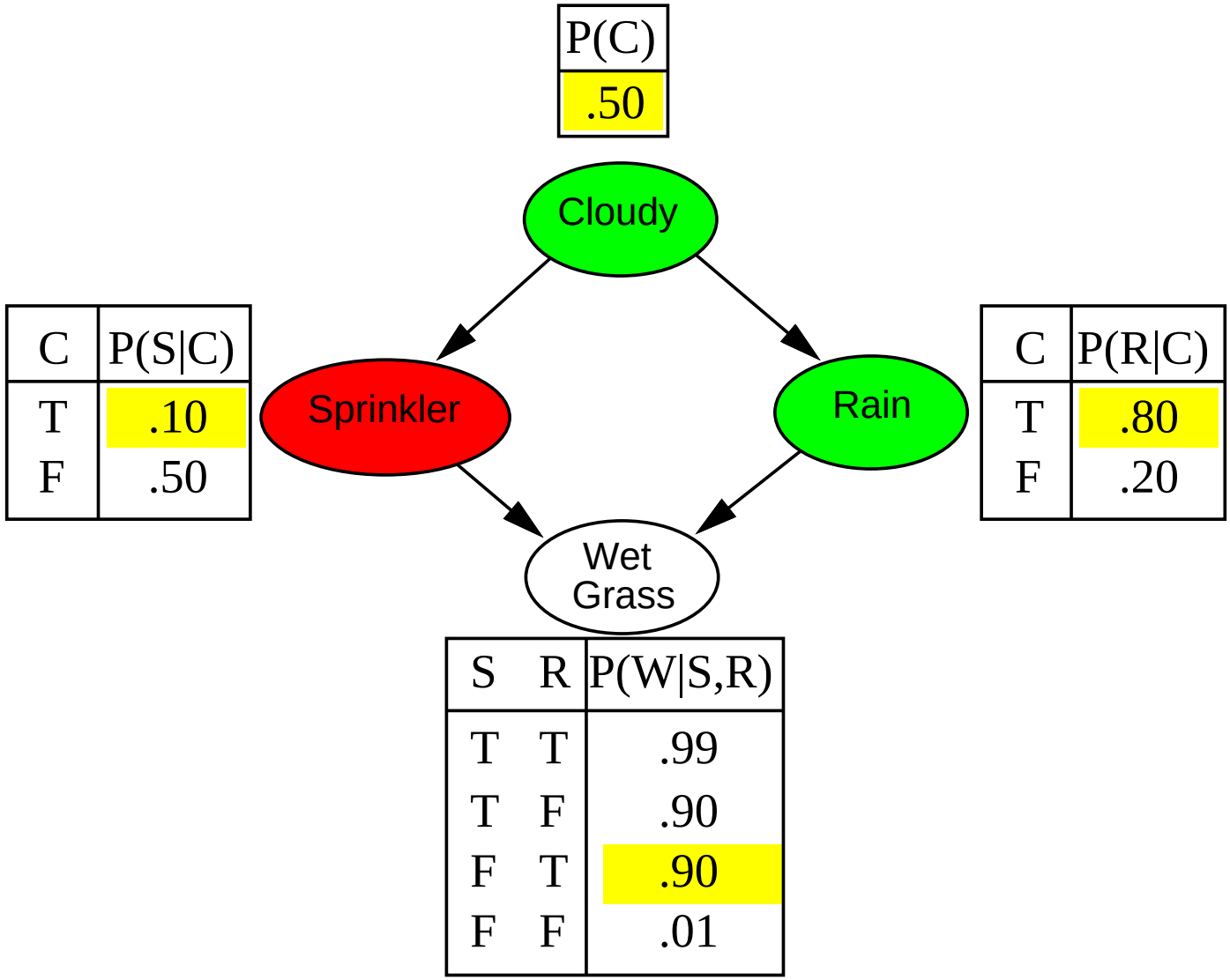
# Esempio


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

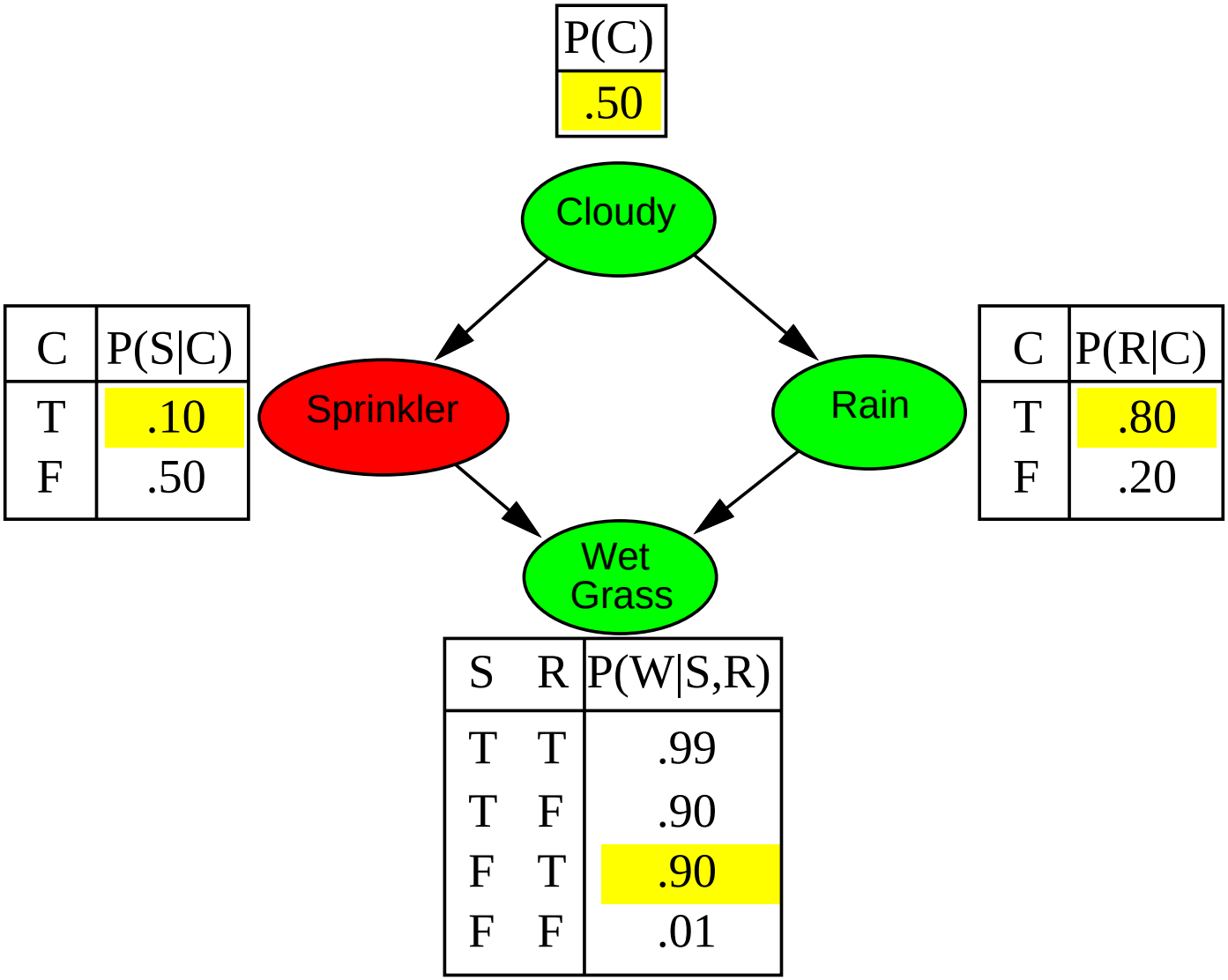
# Esempio


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

# Esempio


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

# Esempio


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

# Campionamento da una rete vuota

Probabilità che PriorSample generi un evento particolare

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

cioè, la vera probabilità a priori

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

Posto $N_{PS}(x_1 \dots x_n)$ essere il numero di campioni generati per l'evento $x_1, \dots, x_n$

Allora abbiamo

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

Cioè, stime derivate da PriorSample sono consistenti

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

# Rejection sampling

$\hat{\mathbf{P}}(X|\mathbf{e})$ stimate da campioni in accordo con $\mathbf{e}$

**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**
    $\mathbf{x} \leftarrow$ 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($\mathbf{N}[X]$)

P.e., stimare $\mathbf{P}(Rain|Sprinkler = true)$ usando 100 campioni
  27 campioni hanno $Sprinkler = true$
    Di questi, 8 hanno $Rain = true$ e 19 hanno $Rain = false$.

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

# Analisi di rejection sampling

$\hat{\mathbf{P}}(X|\mathbf{e}) = \alpha \mathbf{N}_{PS}(X,\mathbf{e})$ (def. algoritmo)
$= \mathbf{N}_{PS}(X,\mathbf{e})/N_{PS}(\mathbf{e})$ (normalizzato tramite $N_{PS}(\mathbf{e})$)
$\approx \mathbf{P}(X,\mathbf{e})/P(\mathbf{e})$ (properietà di PriorSample)
$= \mathbf{P}(X|\mathbf{e})$ (def. di probabilità condizionale)

Quindi rejection sampling restituisce stime consistenti della prob. a posteriori

Problemi: costosissimo se $P(\mathbf{e})$ è piccola

$P(\mathbf{e})$ converge esponenzialmente con il numero di variabili di evidenza!

# Likelihood weighting

Idea: fissare le variabili di evidenza, campionare solo variabili non di evidenza, e pesare ogni campione con la likelihood accordata dall'evidenza

**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

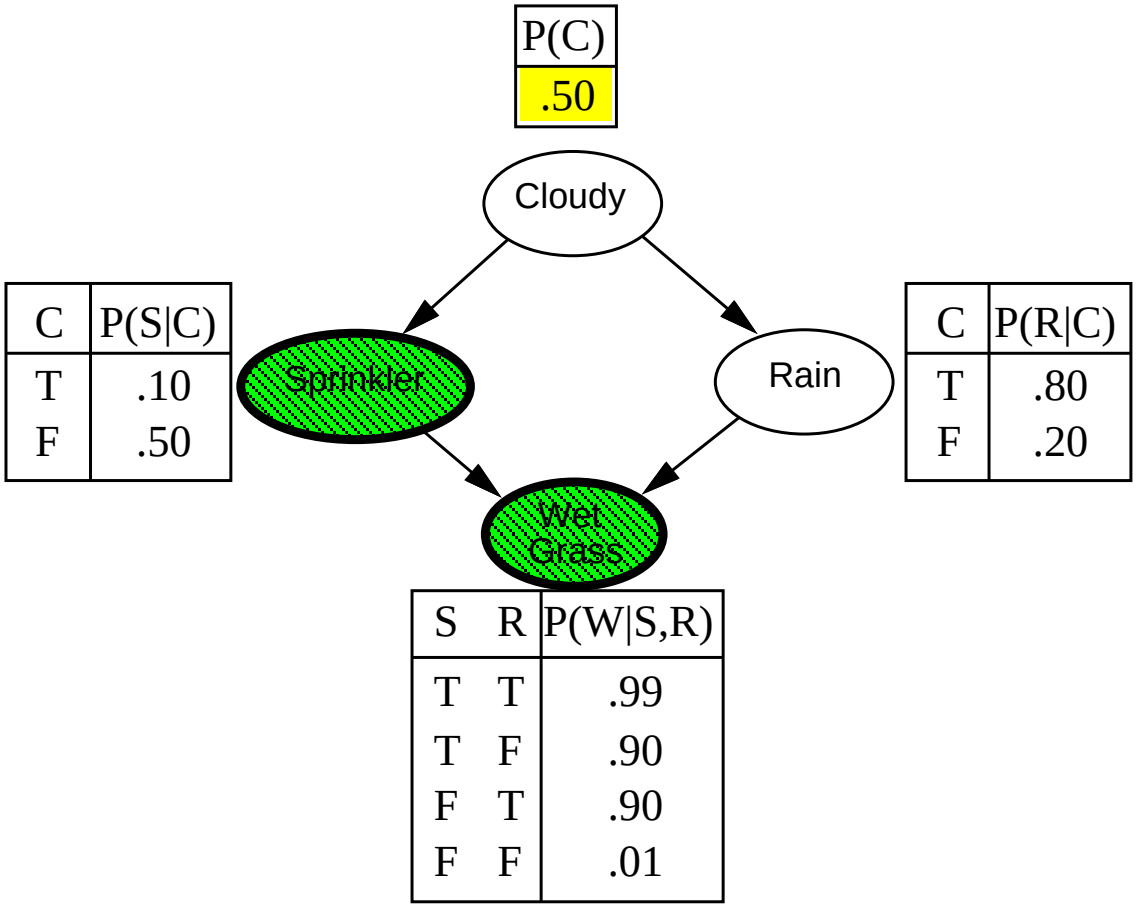
**for** $j = 1$ to $N$ **do**
  $\mathbf{x}, w \leftarrow$ WEIGHTED-SAMPLE($bn$)
  $\mathbf{W}[x] \leftarrow \mathbf{W}[x] + w$ where $x$ is the value of $X$ in $\mathbf{x}$
**return** NORMALIZE($\mathbf{W}[X]$)

---

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

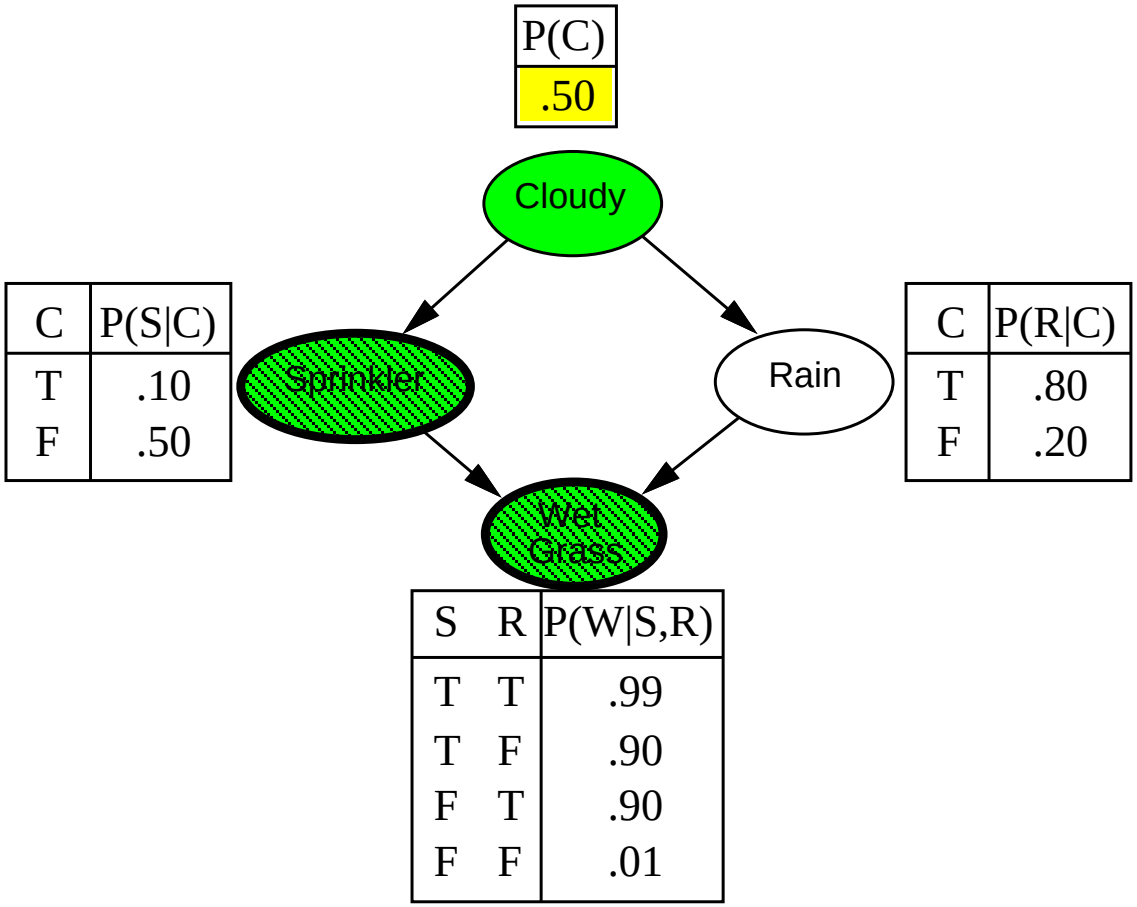
$\mathbf{x} \leftarrow$ an event with $n$ elements; $w \leftarrow 1$
**for** $i = 1$ **to** $n$ **do**
  **if** $X_i$ has a value $x_i$ in $\mathbf{e}$
    **then** $w \leftarrow w \times P(X_i = x_i \mid Parents(X_i))$
    **else** $x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid Parents(X_i))$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

# Esempio di likelihood weighting



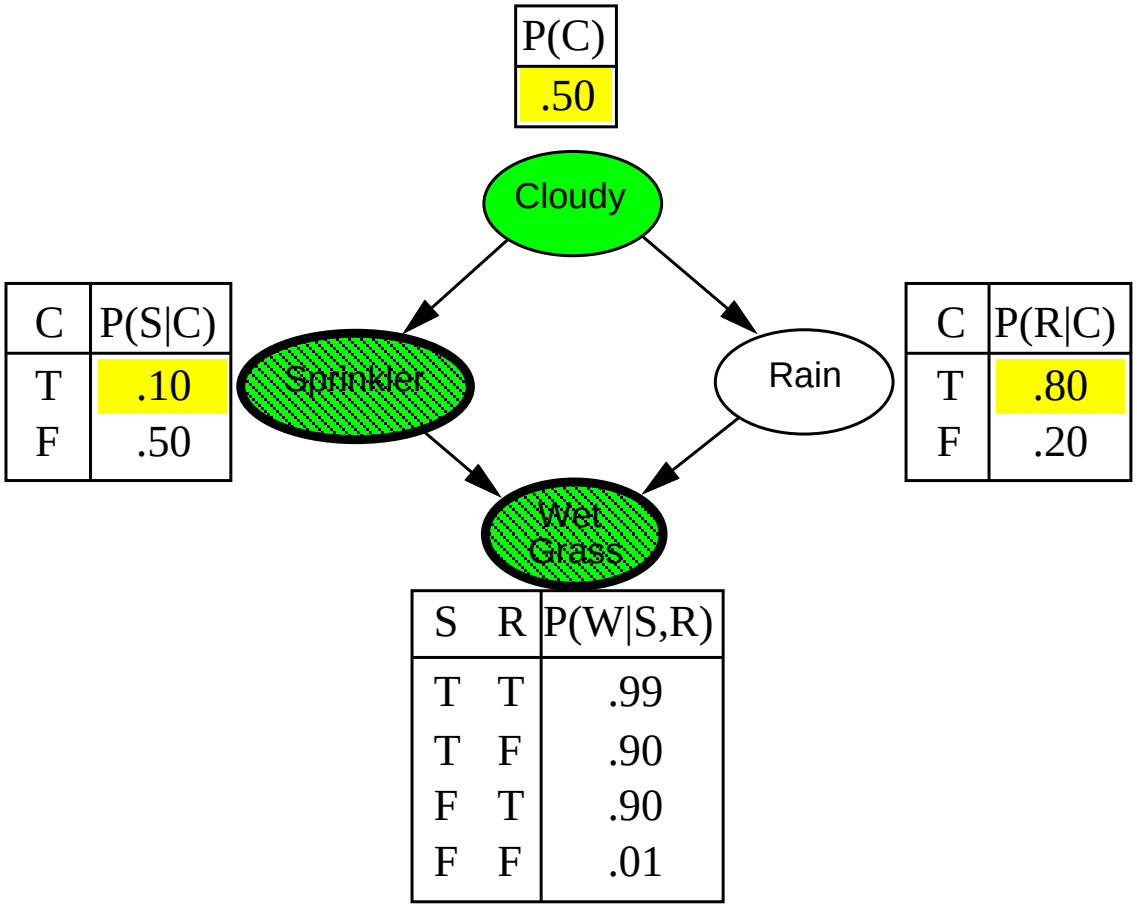

$w = 1.0$

# Esempio di likelihood weighting



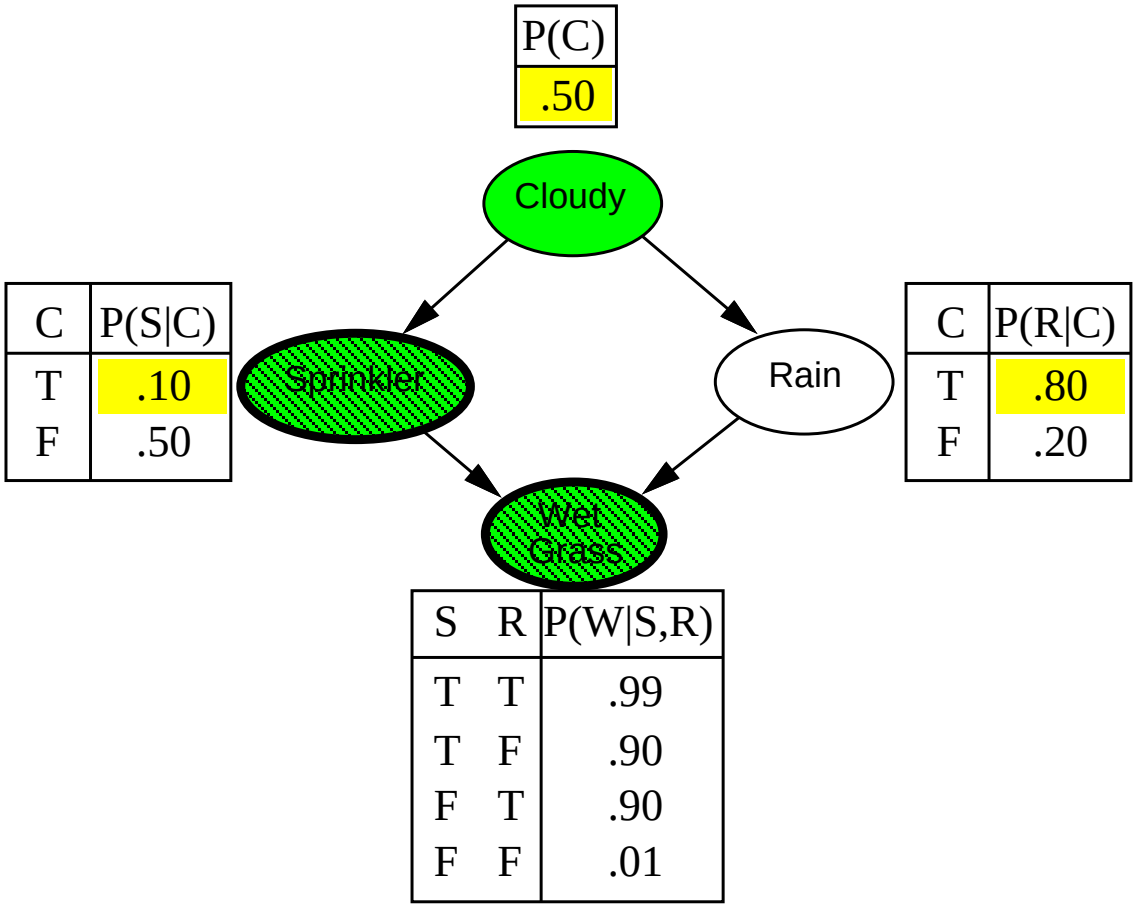

$w = 1.0$

# Esempio di likelihood weighting



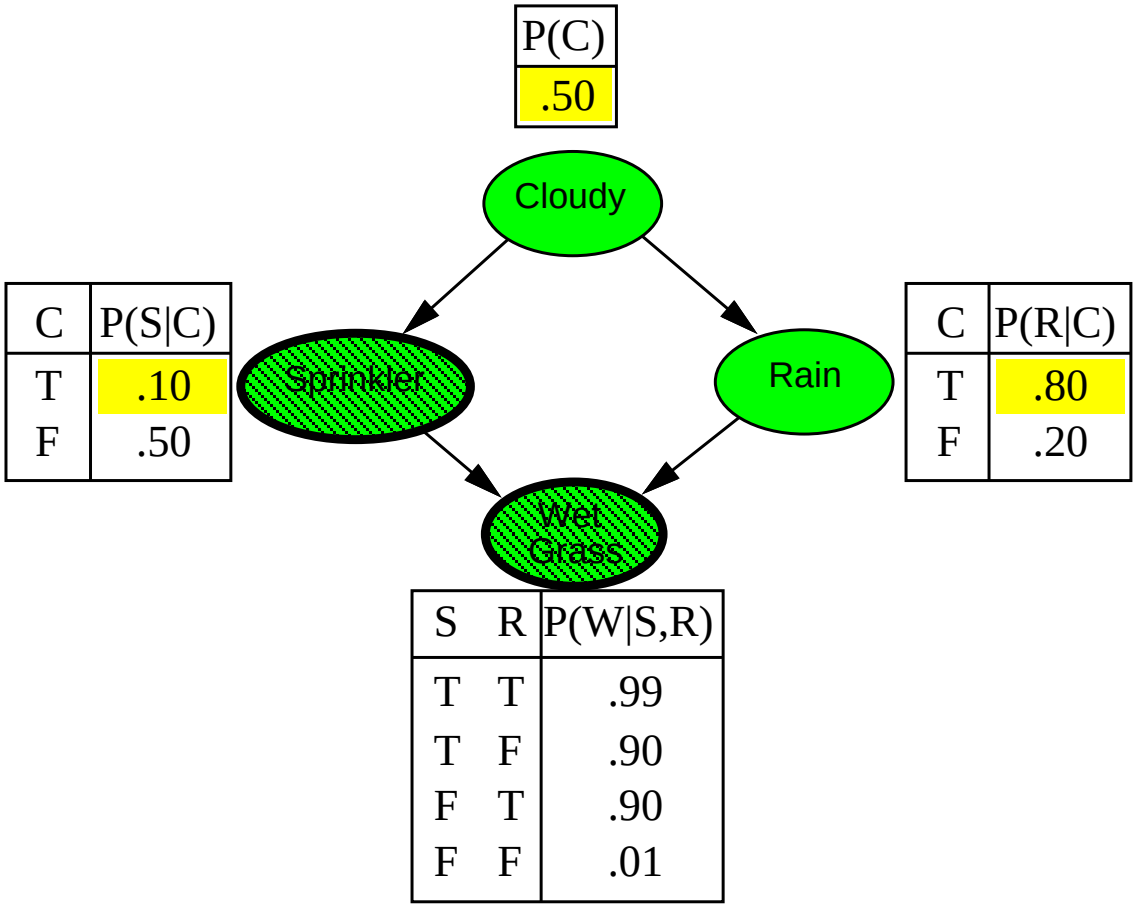

$w = 1.0$

# Esempio di likelihood weighting



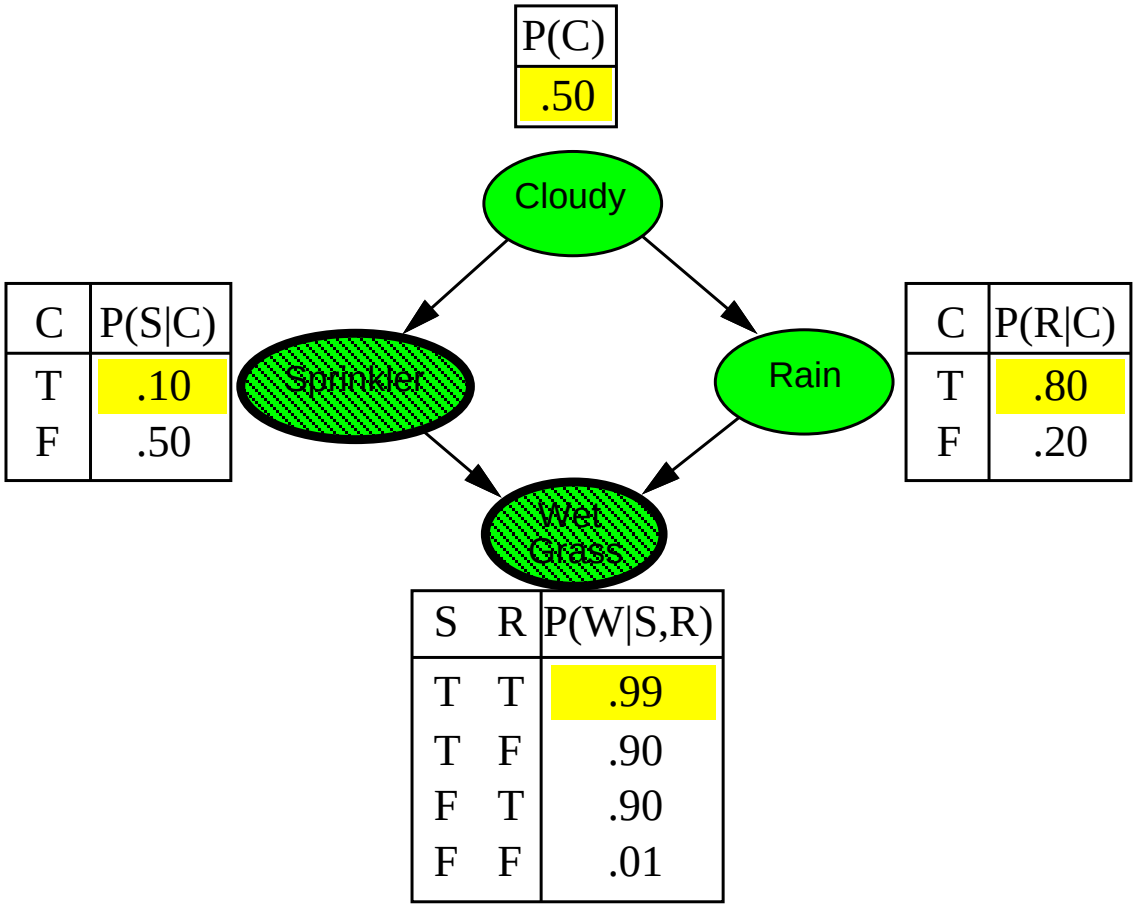

$w = 1.0 \times 0.1$

# Esempio di likelihood weighting



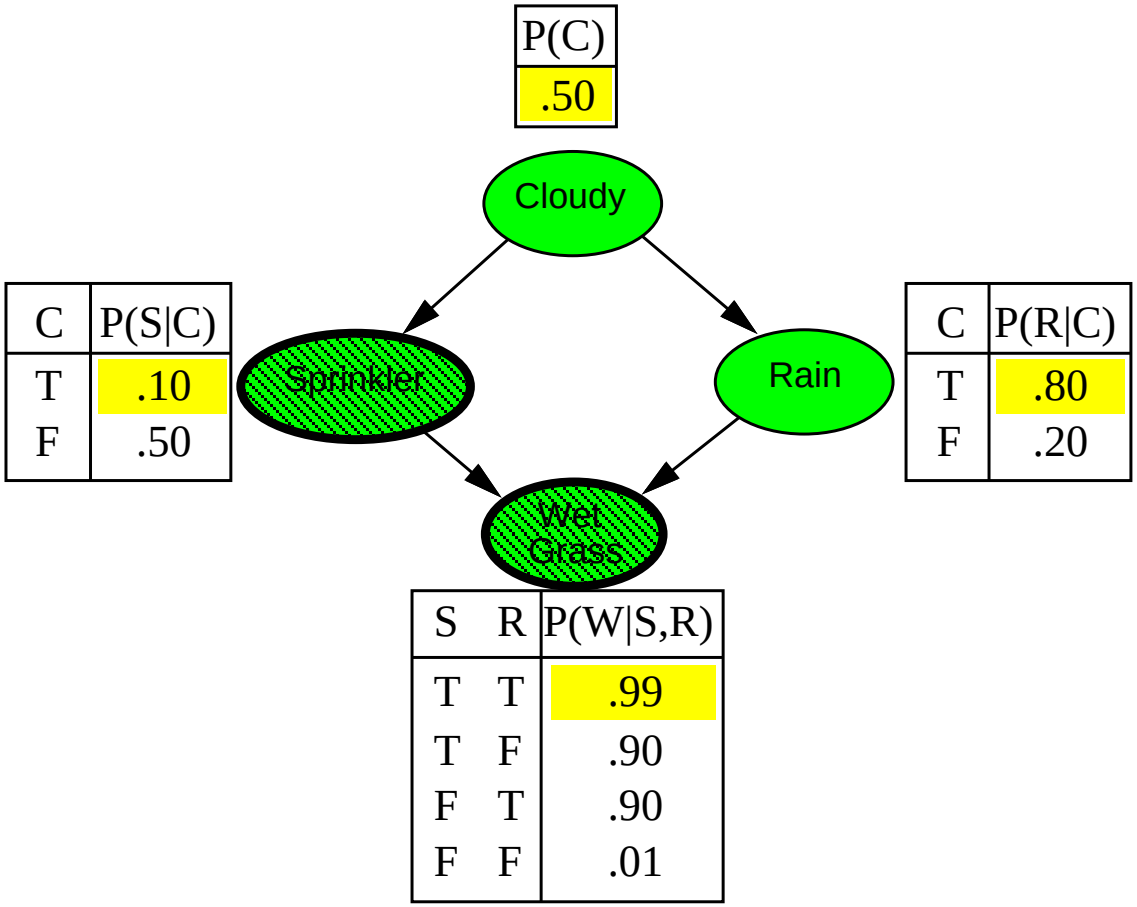

$w = 1.0 \times 0.1$

# Esempio di likelihood weighting




$w = 1.0 \times 0.1$

# Esempio di likelihood weighting




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

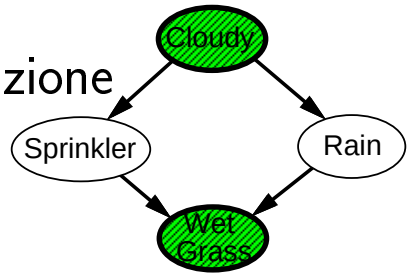
# Analisi di likelihood weighting

La probabilità di campionamento per WeightedSample è

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

Nota: pone attenzione solo all'evidenza negli **antenati**

$\Rightarrow$ da qualche parte "nel mezzo" fra la distribuzione a priori e quella a posteriori




Il peso per un dato campione $\mathbf{z},\mathbf{e}$ è

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

La probabilità di campionamento pesata è

$$\begin{aligned} &S_{WS}(\mathbf{z},\mathbf{e})w(\mathbf{z},\mathbf{e}) \\ &= \Pi_{i=1}^{l} P(z_i|Parents(Z_i)) \ \ \Pi_{i=1}^{m} P(e_i|Parents(E_i)) \\ &= P(\mathbf{z},\mathbf{e}) \text{ (per la semantica standard globale della rete)} \end{aligned}$$

Quindi likelihood weighting restituisce stime consistenti
però le prestazioni degradano con la presenza di tante variabili di evidenza
poiché pochi esempi hanno quasi tutto il peso totale

# Inferenza approssimata tramite MCMC

"Stato" della rete = assegnamento corrente a tutte le variabili

Genera lo stato successivo campionando una variabile dato il suo Markov blanket
Campiona ogni variabile a turno, mantenendo l'evidenza fissa

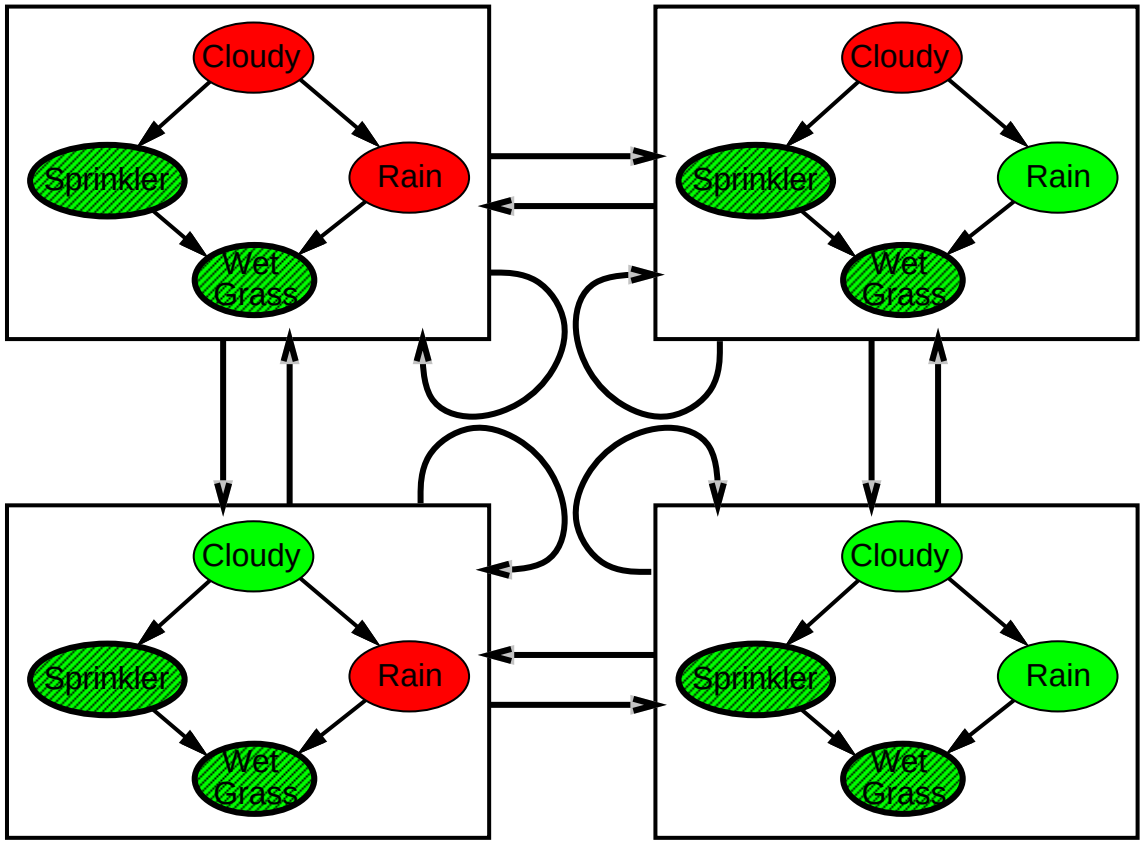
**function** MCMC-ASK($X, \mathbf{e}, bn, N$) **returns** an estimate of $P(X|\mathbf{e})$
  **local variables**: $\mathbf{N}[X]$, a vector of counts over $X$, initially zero
    $\mathbf{Z}$, the nonevidence variables in $bn$
    $\mathbf{x}$, the current state of the network, initially copied from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

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

Può anche scegliere una variabile da campionare a caso ogni volta

# La catena di Markov (Markov chain)

Con $Sprinkler = true, WetGrass = true$, ci sono quattro stati:




"gironzolare" per un pò, fare la media di quello che si osserva

# MCMC: esempio

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

Campionare $Cloudy$ o $Rain$ dato il suo Markov blanket, ripetere.
Contare il numero di volte in cui $Rain$ è true e false nei campioni.

P.e., visita 100 stati

31 hanno $Rain = true$, 69 hanno $Rain = false$

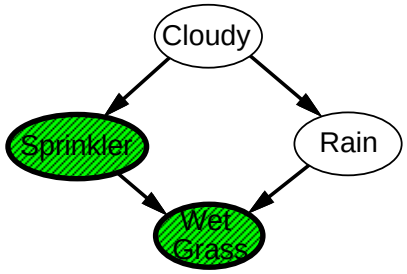
$\hat{\mathbf{P}}(Rain|Sprinkler = true, WetGrass = true)$
$= \textsc{Normalize}(\langle 31, 69\rangle) = \langle 0.31, 0.69\rangle$

Teorema: la catena raggiunge la distribuzione stazionaria:

la frazione di tempo speso in ogni stato è esattamente proporzionale alla sua probabilità a posteriori

# Markov blanket: campionamento

Il Markov blanket di $Cloudy$ è
$Sprinkler$ e $Rain$
Il Markov blanket di $Rain$ è
$Cloudy$, $Sprinkler$, e $WetGrass$




La probabilità dato il Markov blanket è calcolata come segue:
$P(x'_i|MB(X_i)) = P(x'_i|Parents(X_i))\Pi_{Z_j \in Children(X_i)} P(z_j|Parents(Z_j))$

Facilmente implementabile in un sistema parallelo a scambio di messaggi

Principali problemi computazionali:
1) Difficoltà a riconoscere se la convergenza è avvenuta
2) Molto costoso se il Markov blanket è grande:
$P(X_i|MB(X_i))$ non cambierà molto (legge dei grandi numeri)

# Riassunto

Inferenza esatta tramite l'eliminazione di variabile:
- polinomiale su polialberi, NP-hard in generale
- spazio = tempo, dipendente dalla topologia

Inferenza approssimata tramite LW e MCMC:
- LW si comporta male quando c'è molta evidenza (soprattutto a "valle")
- LW, MCMC in genere indipendenti dalla topologia
- La convergenza può essere molto lenta per probabilità vicine a 1 o 0
- Possono trattare combinazioni arbitrarie di variabili discrete e continue