

# CS 1571 Introduction to AI

## Lecture 20

### Bayesian belief networks: inference

Milos Hauskrecht

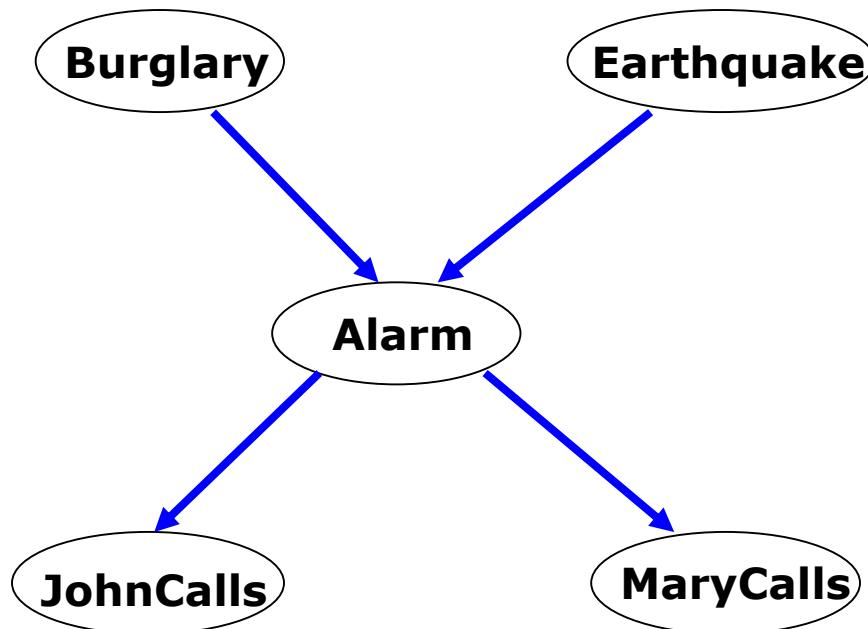
[milos@cs.pitt.edu](mailto:milos@cs.pitt.edu)

5329 Sennott Square

# Bayesian belief network

## 1. Directed acyclic graph

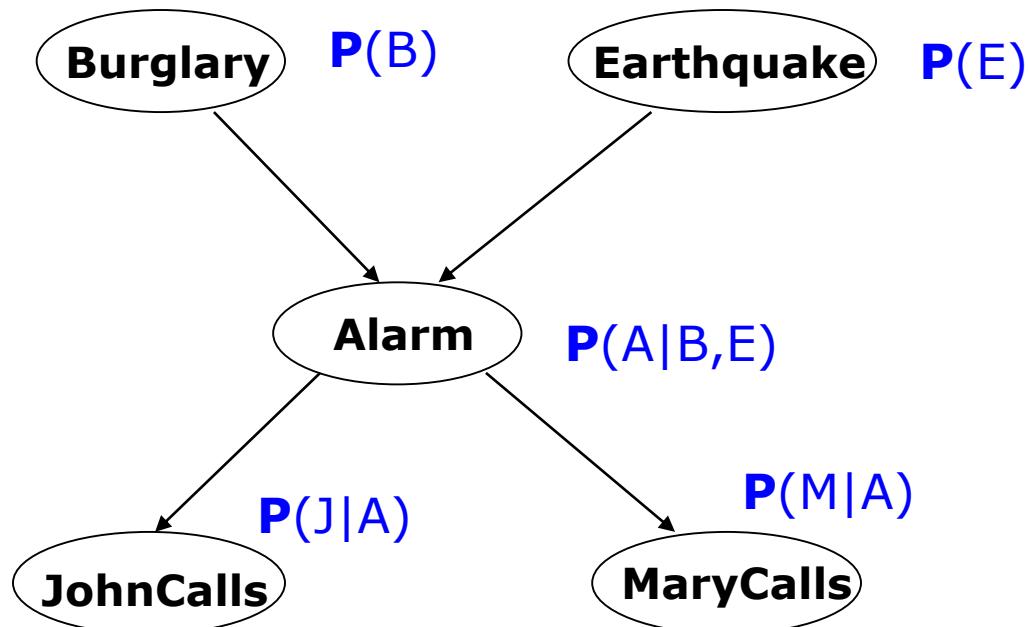
- **Nodes** = random variables  
Burglary, Earthquake, Alarm, Mary calls and John calls
- **Links** = direct (causal) dependencies between variables.  
Missing links model conditional independences among variables.



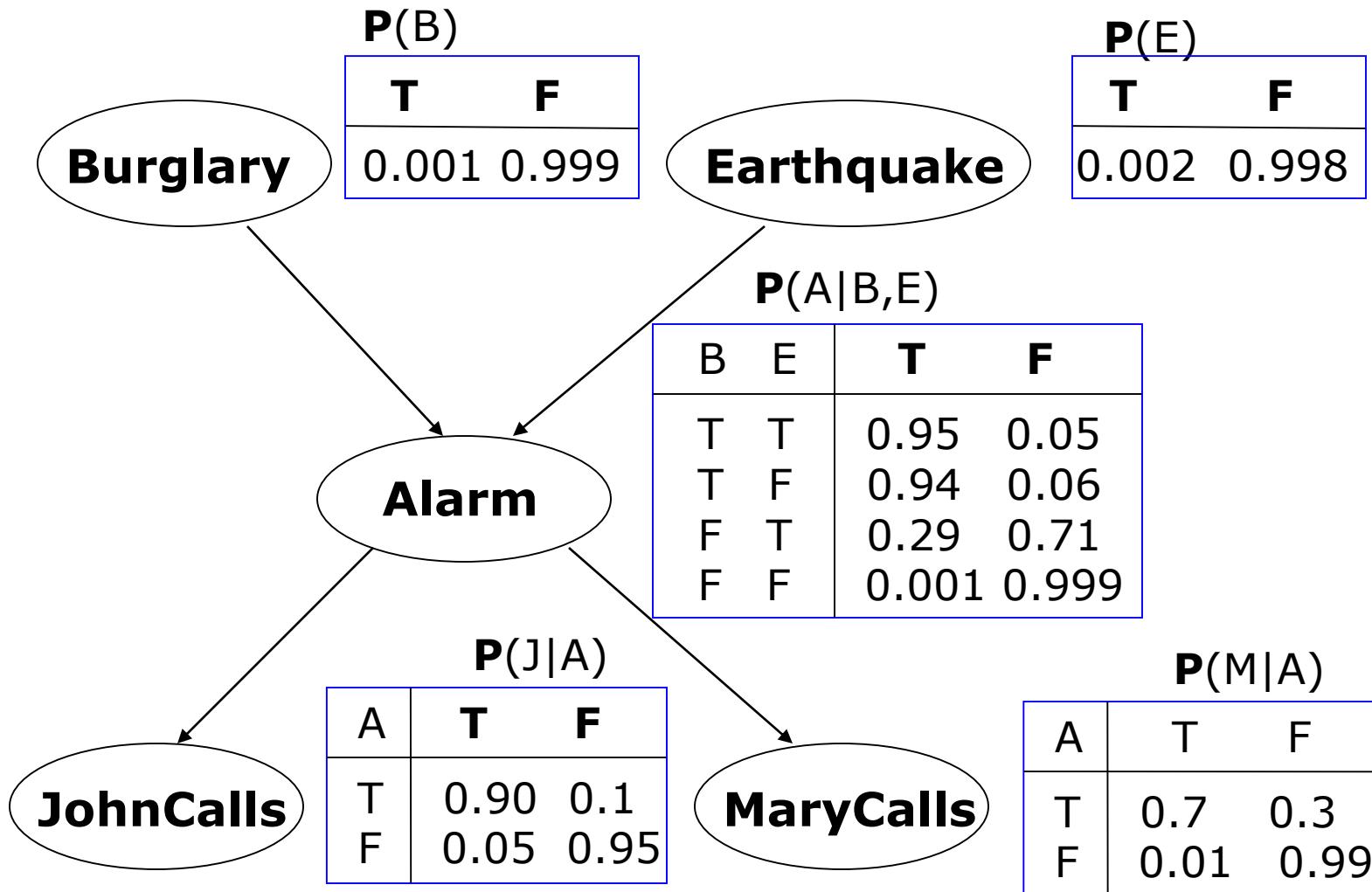
# Bayesian belief network

## 2. Local conditional distributions

- relate variables and their parents



# Bayesian belief network



# Full joint distribution in BBNs

**Full joint distribution** is defined in terms of local conditional distributions (obtained via the chain rule):

$$\mathbf{P}(X_1, X_2, \dots, X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

**Example:**

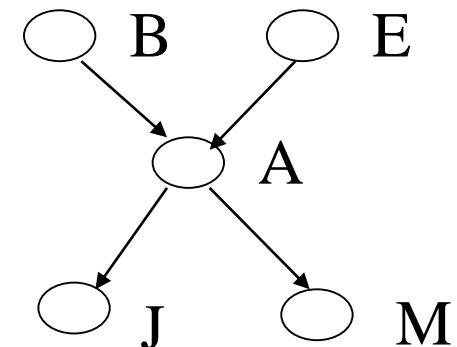
Assume the following assignment  
of values to random variables

$$B = T, E = T, A = T, J = T, M = F$$

Then its probability is:

$$P(B = T, E = T, A = T, J = T, M = F) =$$

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



# Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

$$\mathbf{P}(X_1, X_2, \dots, X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

- What did we save?**

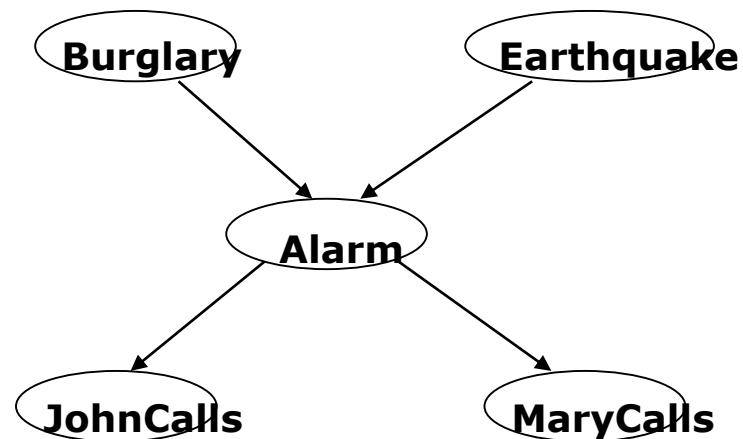
Alarm example: 5 binary (True, False) variables

# of parameters of the full joint:

$$2^5 = 32$$

One parameter is for free:

$$2^5 - 1 = 31$$



# Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

$$\mathbf{P}(X_1, X_2, \dots, X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

- What did we save?**

Alarm example: 5 binary (True, False) variables

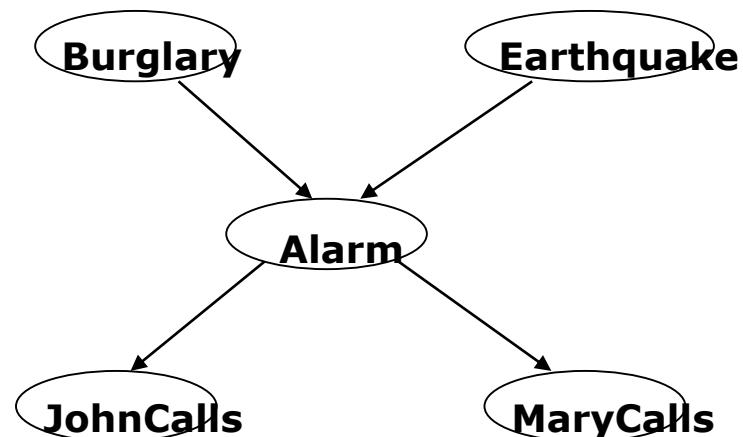
# of parameters of the full joint:

$$2^5 = 32$$

One parameter is for free:

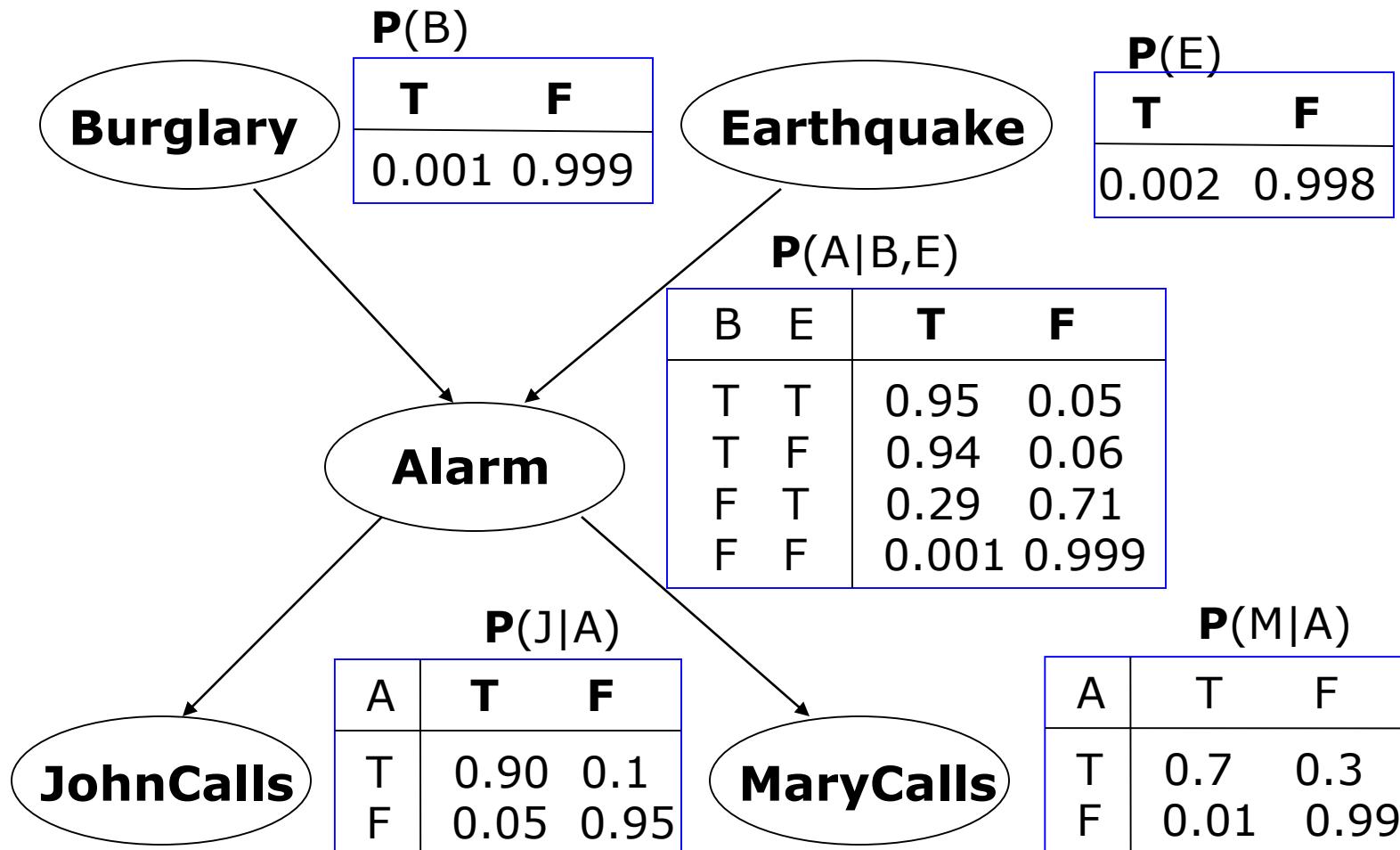
$$2^5 - 1 = 31$$

# of parameters of the BBN: ?



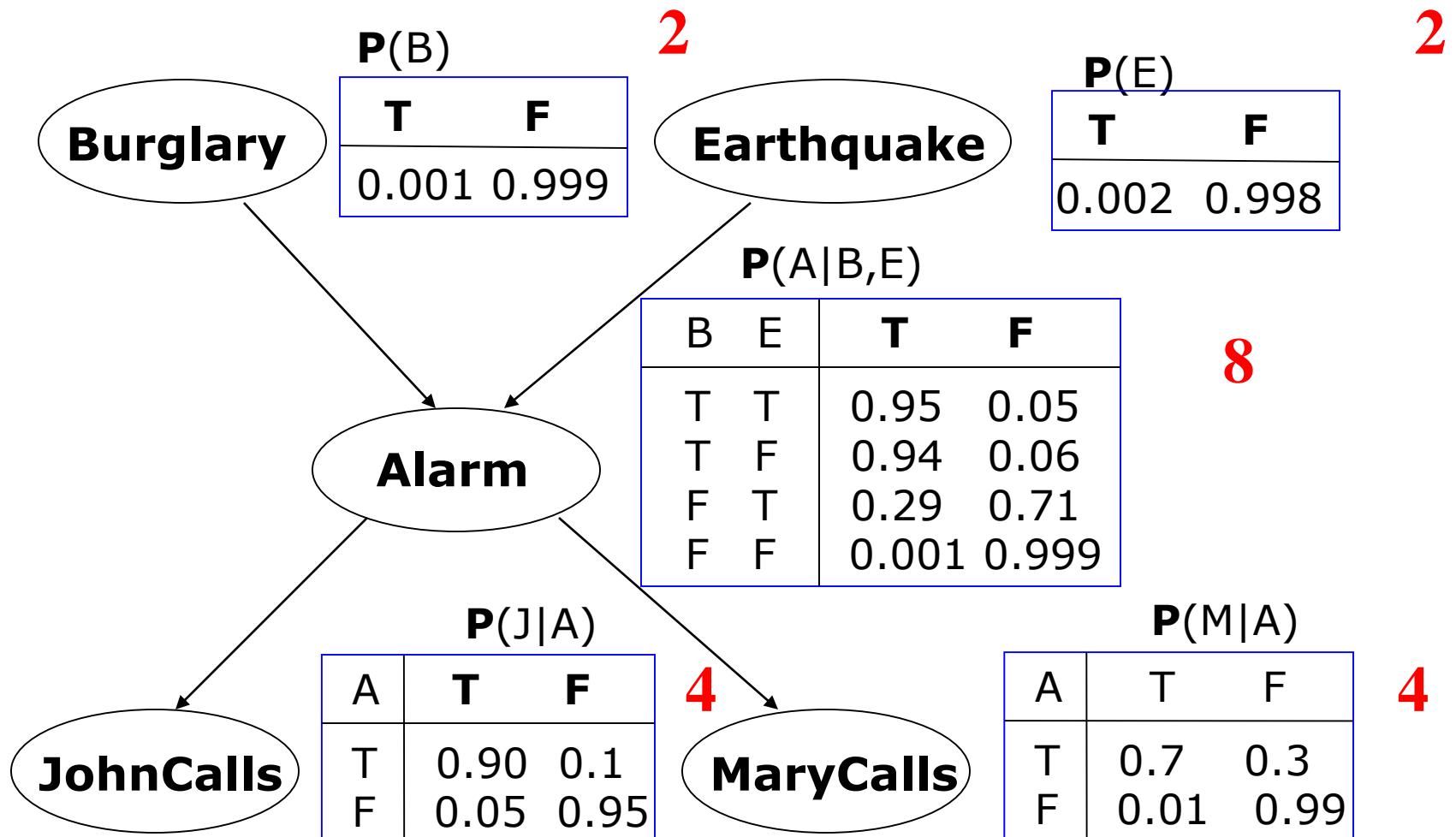
# Bayesian belief network.

- In the BBN the **full joint distribution** is expressed using a set of local conditional distributions



# Bayesian belief network.

- In the BBN the **full joint distribution** is expressed using a set of local conditional distributions



# Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

$$\mathbf{P}(X_1, X_2, \dots, X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

- What did we save?**

Alarm example: 5 binary (True, False) variables

# of parameters of the full joint:

$$2^5 = 32$$

One parameter is for free:

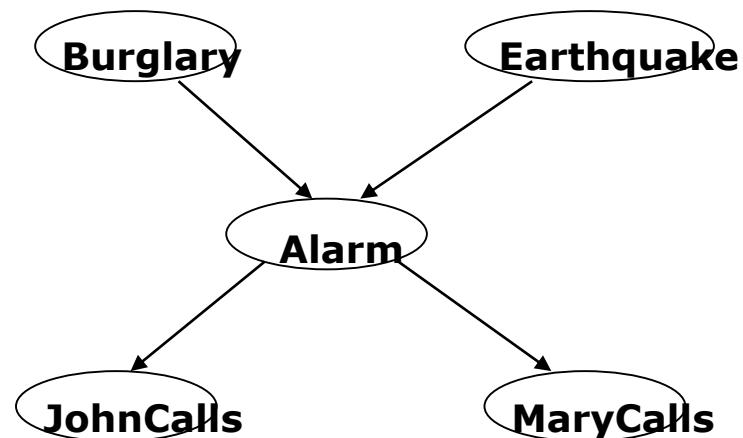
$$2^5 - 1 = 31$$

# of parameters of the BBN:

$$2^3 + 2(2^2) + 2(2) = 20$$

One parameter in every conditional is for free:

?



# Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

$$\mathbf{P}(X_1, X_2, \dots, X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

- What did we save?**

Alarm example: 5 binary (True, False) variables

# of parameters of the full joint:

$$2^5 = 32$$

One parameter is for free:

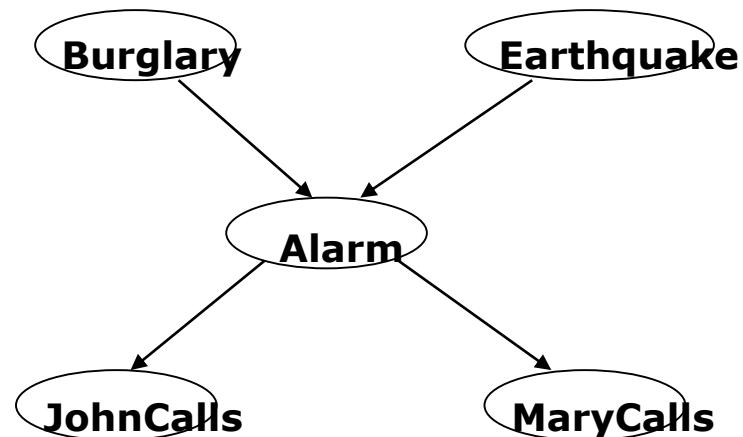
$$2^5 - 1 = 31$$

# of parameters of the BBN:

$$2^3 + 2(2^2) + 2(2) = 20$$

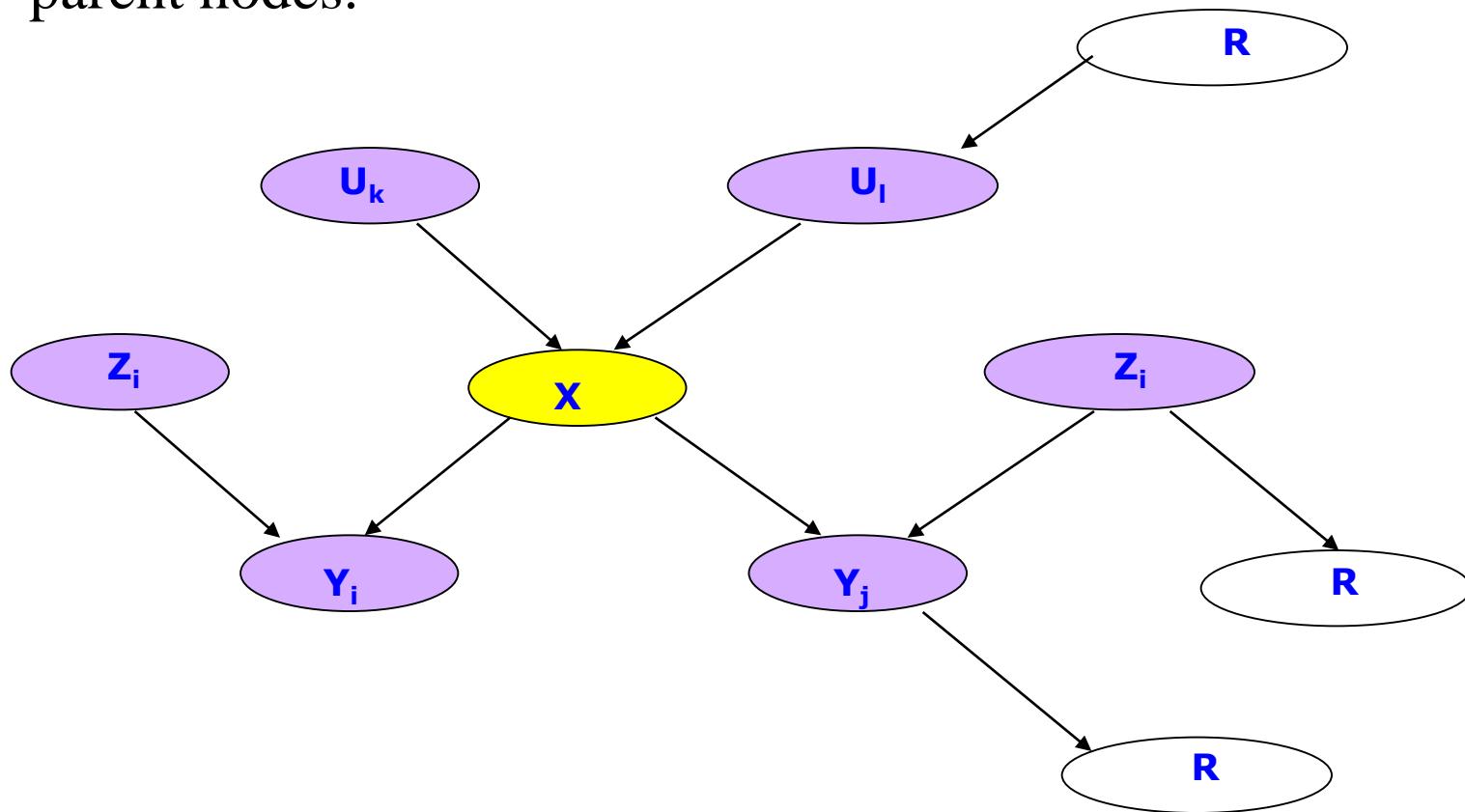
One parameter in every conditional is for free:

$$2^2 + 2(2) + 2(1) = 10$$



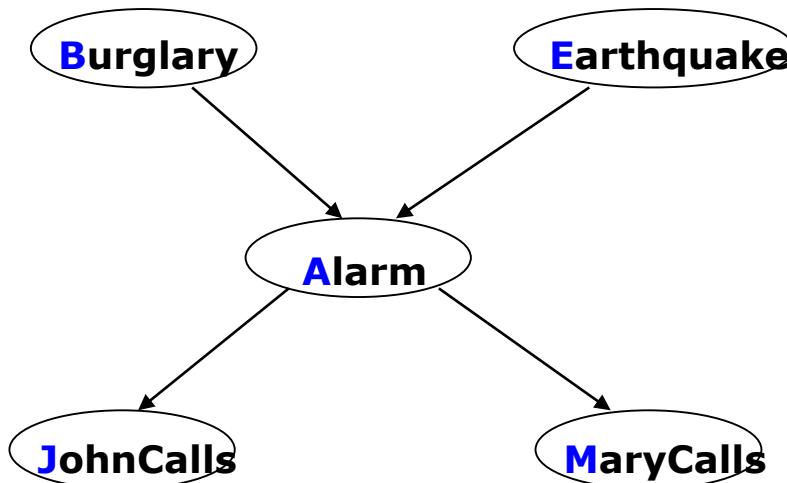
# Markov Blanket

- The Markov blanket of a node is the set of nodes that makes it independent of the rest of the network.
- The Markov blanket consists of all parent, children and co-parent nodes.



# Inference in Bayesian network

- **Bad news:**
  - Exact inference problem in BBNs is NP-hard (Cooper)
  - Approximate inference is NP-hard (Dagum, Luby)
- **But** very often we can achieve significant improvements
- Assume our Alarm network



- Assume we want to compute:  $P(J = T)$

# Inference in Bayesian networks

**Computing:**  $P(J = T)$

**Approach 1. Blind approach.**

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$P(J = T) =$$

$$= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(B = b, E = e, A = a, J = T, M = m)$$

$$= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T \mid A = a) P(M = m \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e)$$

**Computational cost:**

Number of additions: ?

Number of products: ?

# Inference in Bayesian networks

**Computing:**  $P(J = T)$

**Approach 1. Blind approach.**

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$P(J = T) =$$

$$= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(B = b, E = e, A = a, J = T, M = m)$$

$$= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T \mid A = a) P(M = m \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e)$$

**Computational cost:**

Number of additions: 15

Number of products: ?

# Inference in Bayesian networks

**Computing:**  $P(J = T)$

**Approach 1. Blind approach.**

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$P(J = T) =$$

$$= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(B = b, E = e, A = a, J = T, M = m)$$

$$= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T \mid A = a) P(M = m \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e)$$

**Computational cost:**

Number of additions: 15

Number of products:  $16 * 4 = 64$

# Inference in Bayesian networks

## Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$P(J = T) =$$

$$\begin{aligned} &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\ &= \sum_{b \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(B = b) \left[ \sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \\ &= \sum_{a \in T, F} P(J = T | A = a) \left[ \sum_{m \in T, F} P(M = m | A = a) \right] \left[ \sum_{b \in T, F} P(B = b) \left[ \sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \right] \end{aligned}$$

## Computational cost:

Number of additions: ?

Number of products: ?

# Inference in Bayesian networks

## Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$= \sum_{a \in T, F} P(J = T \mid A = a) \left[ \sum_{m \in T, F} P(M = m \mid A = a) \right] \left[ \sum_{b \in T, F} P(B = b) \left[ \sum_{e \in T, F} P(A = a \mid B = b, E = e) P(E = e) \right] \right]$$

↓  
1

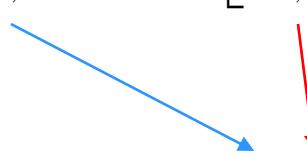
**Computational cost:**

Number of additions: ?

# Inference in Bayesian networks

## Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$= \sum_{a \in T, F} P(J = T \mid A = a) \left[ \sum_{m \in T, F} P(M = m \mid A = a) \right] \left[ \sum_{b \in T, F} P(B = b) \left[ \sum_{e \in T, F} P(A = a \mid B = b, E = e) P(E = e) \right] \right]$$


2\*1

**Computational cost:**

Number of additions: ?

# Inference in Bayesian networks

## Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$= \sum_{a \in T, F} P(J = T \mid A = a) \left[ \sum_{m \in T, F} P(M = m \mid A = a) \right] \left[ \sum_{b \in T, F} P(B = b) \left[ \sum_{e \in T, F} P(A = a \mid B = b, E = e) P(E = e) \right] \right]$$

2\*2\*1

**Computational cost:**

Number of additions: ?

# Inference in Bayesian networks

## Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$= \sum_{a \in T, F} P(J = T \mid A = a) \left[ \sum_{m \in T, F} P(M = m \mid A = a) \right] \left[ \sum_{b \in T, F} P(B = b) \left[ \sum_{e \in T, F} P(A = a \mid B = b, E = e) P(E = e) \right] \right]$$

The diagram illustrates the computational cost of the interleaved approach. It shows the expression above with blue arrows pointing from the outermost loops to the result '2\*2\*1'. There are two blue arrows pointing from the first two loops to '2\*2\*1', and one red arrow pointing from the innermost loop to the same result.

**Computational cost:**

Number of additions: ?

# Inference in Bayesian networks

## Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$= \sum_{a \in T, F} P(J = T \mid A = a) \left[ \sum_{m \in T, F} P(M = m \mid A = a) \right] \left[ \sum_{b \in T, F} P(B = b) \left[ \sum_{e \in T, F} P(A = a \mid B = b, E = e) P(E = e) \right] \right]$$

The diagram illustrates the computation of the expression above. It shows blue arrows pointing from the outer summations to intermediate results, and red arrows pointing from the inner summations to the final result. The intermediate results are labeled 1, 2\*1, 2\*1, and 2\*2\*1.

**Computational cost:**

Number of additions: ?

# Inference in Bayesian networks

## Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$= \sum_{a \in T, F} P(J = T \mid A = a) \left[ \sum_{m \in T, F} P(M = m \mid A = a) \right] \left[ \sum_{b \in T, F} P(B = b) \left[ \sum_{e \in T, F} P(A = a \mid B = b, E = e) P(E = e) \right] \right]$$

The diagram shows the computation graph for the given formula. It starts with four summations at the top. Blue arrows point from the first three summations to intermediate results: 1, 2\*1, and 2\*1. These intermediate results then point to a final result of 2\*2\*1. Red arrows point from the fourth summation to the final result.

**Computational cost:**

Number of additions:  $1 + 2 * [1 + 1 + 2 * 1] = 9$

# Inference in Bayesian networks

## Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$= \sum_{a \in T, F} P(J = T \mid A = a) \left[ \sum_{m \in T, F} P(M = m \mid A = a) \right] \left[ \sum_{b \in T, F} P(B = b) \left[ \sum_{e \in T, F} P(A = a \mid B = b, E = e) P(E = e) \right] \right]$$

↓  
1

**Computational cost:**

Number of products: ?

# Inference in Bayesian networks

## Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$= \sum_{a \in T, F} P(J = T \mid A = a) \left[ \sum_{m \in T, F} P(M = m \mid A = a) \right] \left[ \sum_{b \in T, F} P(B = b) \left[ \sum_{e \in T, F} P(A = a \mid B = b, E = e) P(E = e) \right] \right]$$

2\*2 \*2\*1

**Computational cost:**

Number of products: ?

# Inference in Bayesian networks

## Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$= \sum_{a \in T, F} P(J = T \mid A = a) \left[ \sum_{m \in T, F} P(M = m \mid A = a) \right] \left[ \sum_{b \in T, F} P(B = b) \right] \left[ \sum_{e \in T, F} P(A = a \mid B = b, E = e) P(E = e) \right]$$

2\*2      2\*2\*1      2\*2 \*2\*1

**Computational cost:**

Number of products: ?

# Inference in Bayesian networks

## Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$= \sum_{a \in T, F} P(J = T \mid A = a) \left[ \sum_{m \in T, F} P(M = m \mid A = a) \right] \left[ \sum_{b \in T, F} P(B = b) \right] \left[ \sum_{e \in T, F} P(A = a \mid B = b, E = e) P(E = e) \right]$$

The diagram illustrates the computation cost of the expression. It shows the nested summations and the resulting cost. Blue arrows point from the outermost summations to the result  $2*2 *2*1$ . Red arrows point from the inner summations to the result  $2*2$ .

**Computational cost:**

Number of products:  $2 * [2 + 2 * (1 + 2 * 1)] = 16$

# Inference in Bayesian networks

## Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$P(J = T) =$$

$$\begin{aligned} &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\ &= \sum_{b \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(B = b) \left[ \sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \\ &= \sum_{a \in T, F} P(J = T | A = a) \left[ \sum_{m \in T, F} P(M = m | A = a) \right] \left[ \sum_{b \in T, F} P(B = b) \left[ \sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \right] \end{aligned}$$

### Computational cost:

Number of additions:  $1 + 2 * [1 + 1 + 2 * 1] = 9$

Number of products:  $2 * [2 + 2 * (1 + 2 * 1)] = 16$

# Variable elimination

- **Variable elimination:**
  - Similar idea but interleave sum and products one variable at the time during inference
  - E.g. Query  $P(J = T)$  requires to eliminate A,B,E,M and this can be done in different order

$$P(J = T) =$$

$$= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(J = T \mid A = a) P(M = m \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e)$$

# Variable elimination

Assume order: M, E, B,A to calculate  $P(J = T)$

$$\begin{aligned} &= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\ &= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \left[ \sum_{m \in T,F} P(M = m | A = a) \right] \\ &= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) - 1 \quad \text{1} \quad \text{red arrow} \\ &= \sum_{a \in T,F} \sum_{b \in T,F} P(J = T | A = a) P(B = b) \left[ \sum_{e \in T,F} P(A = a | B = b, E = e) P(E = e) \right] \\ &= \sum_{a \in T,F} \sum_{b \in T,F} P(J = T | A = a) P(B = b) \tau_1(A = a, B = b) \\ &= \sum_{a \in T,F} P(J = T | A = a) \left[ \sum_{b \in T,F} P(B = b) \tau_1(A = a, B = b) \right] \\ &= \sum_{a \in T,F} P(J = T | A = a) \quad \tau_2(A = a) \quad = \boxed{P(J = T)} \end{aligned}$$

# Inference in Bayesian network

- **Exact inference algorithms:**



- Variable elimination
- Recursive decomposition (Cooper, Darwiche)
- Symbolic inference (D'Ambrosio)
- Belief propagation algorithm (Pearl)



- Clustering and joint tree approach (Lauritzen, Spiegelhalter)
- Arc reversal (Olmsted, Schachter)

- **Approximate inference algorithms:**



- Monte Carlo methods:
  - Forward sampling, Likelihood sampling
- Variational methods

# Monte Carlo approaches

- **MC approximation:**

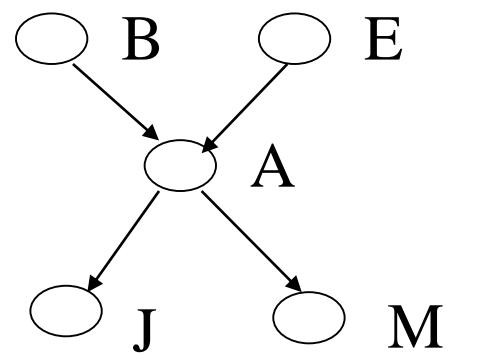
- The probability is approximated using sample frequencies
- **Example:**

$$\tilde{P}(B = T, J = T) = \frac{N_{B=T, J=T}}{N}$$

#samples with  $B = T, J = T$

total # samples

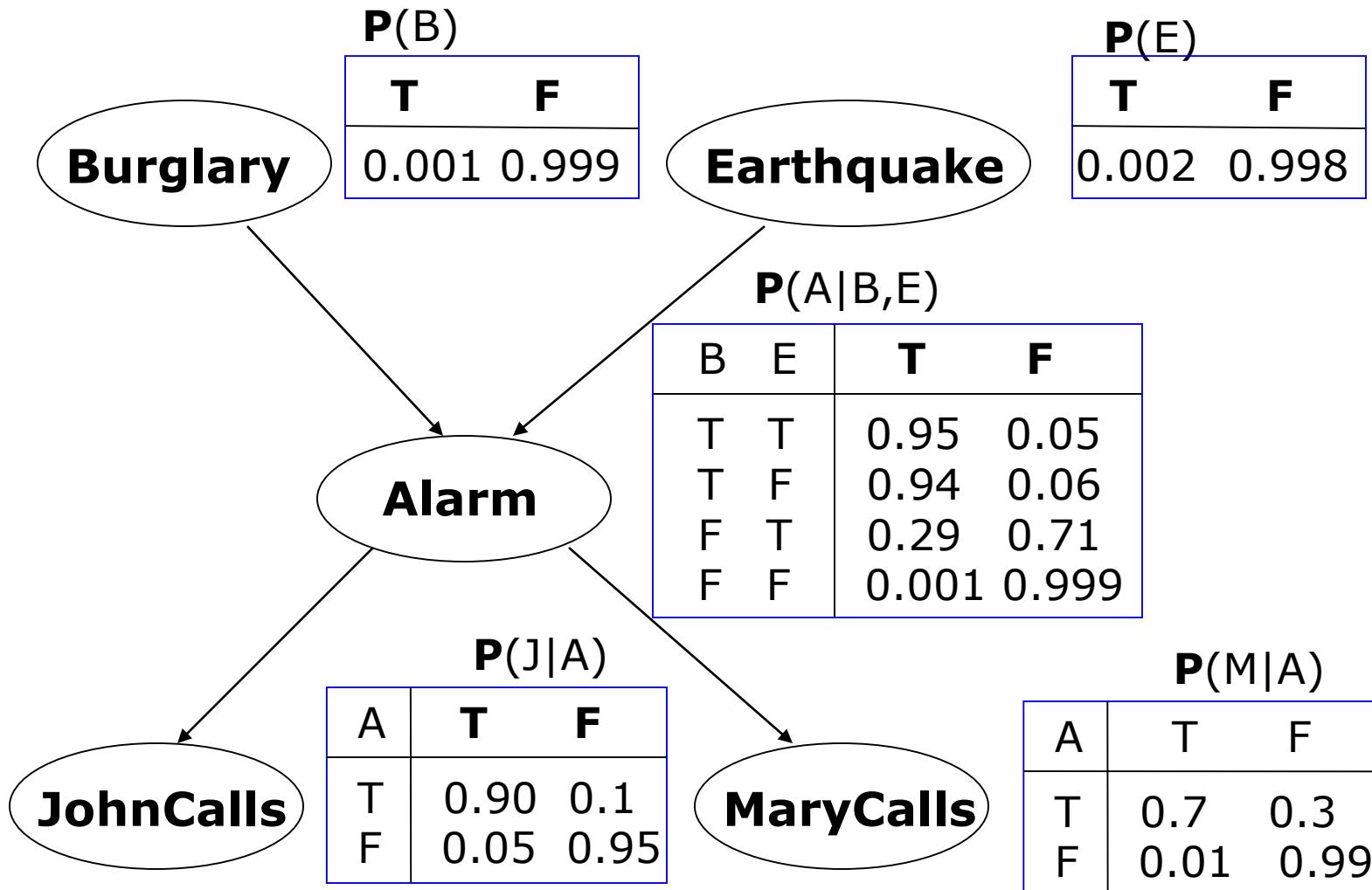
- **BBN sampling:**



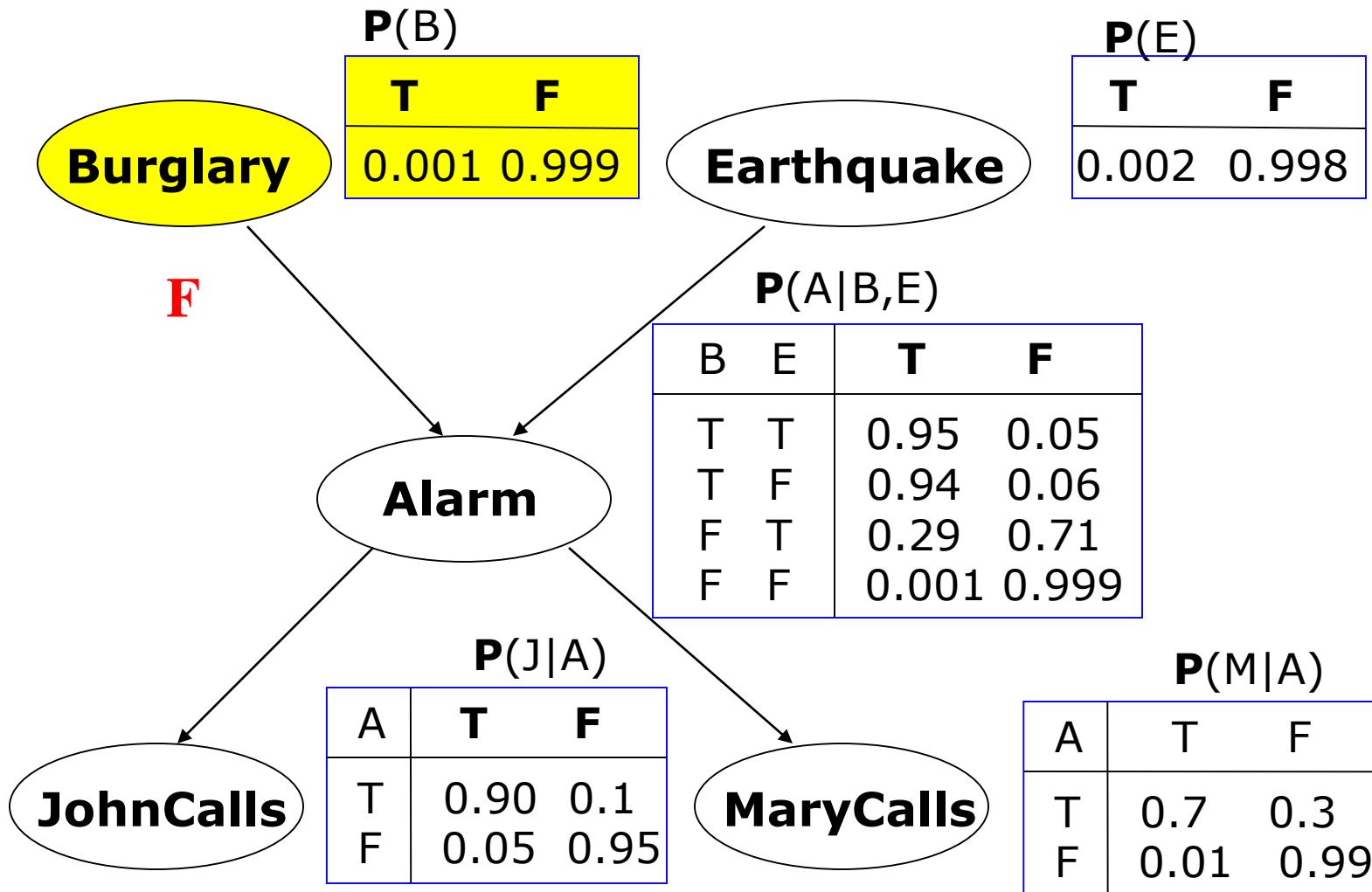
Generate sample in a top down manner, following the links

- **One sample gives one assignment of values to all variables**

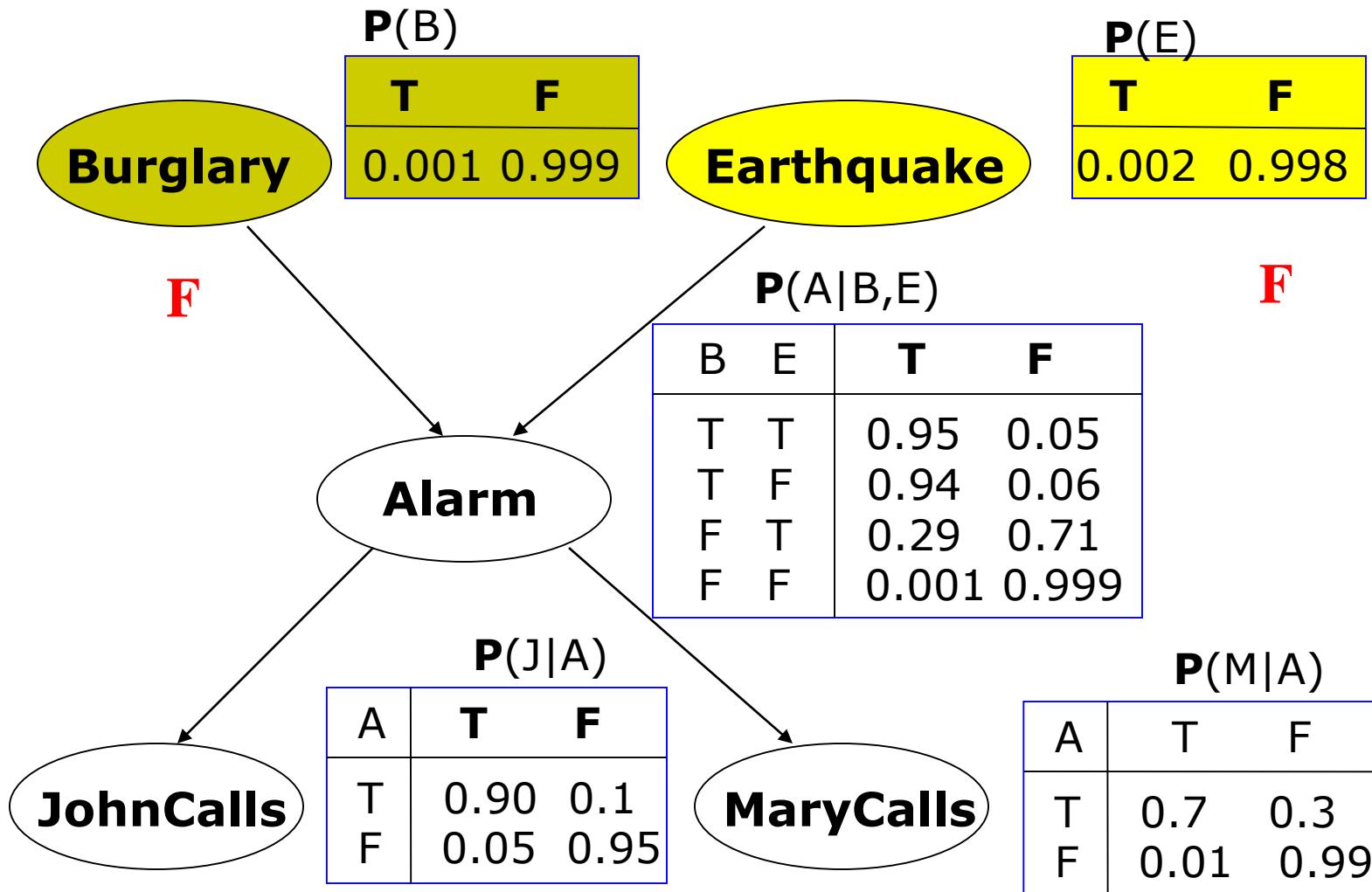
# BBN sampling example



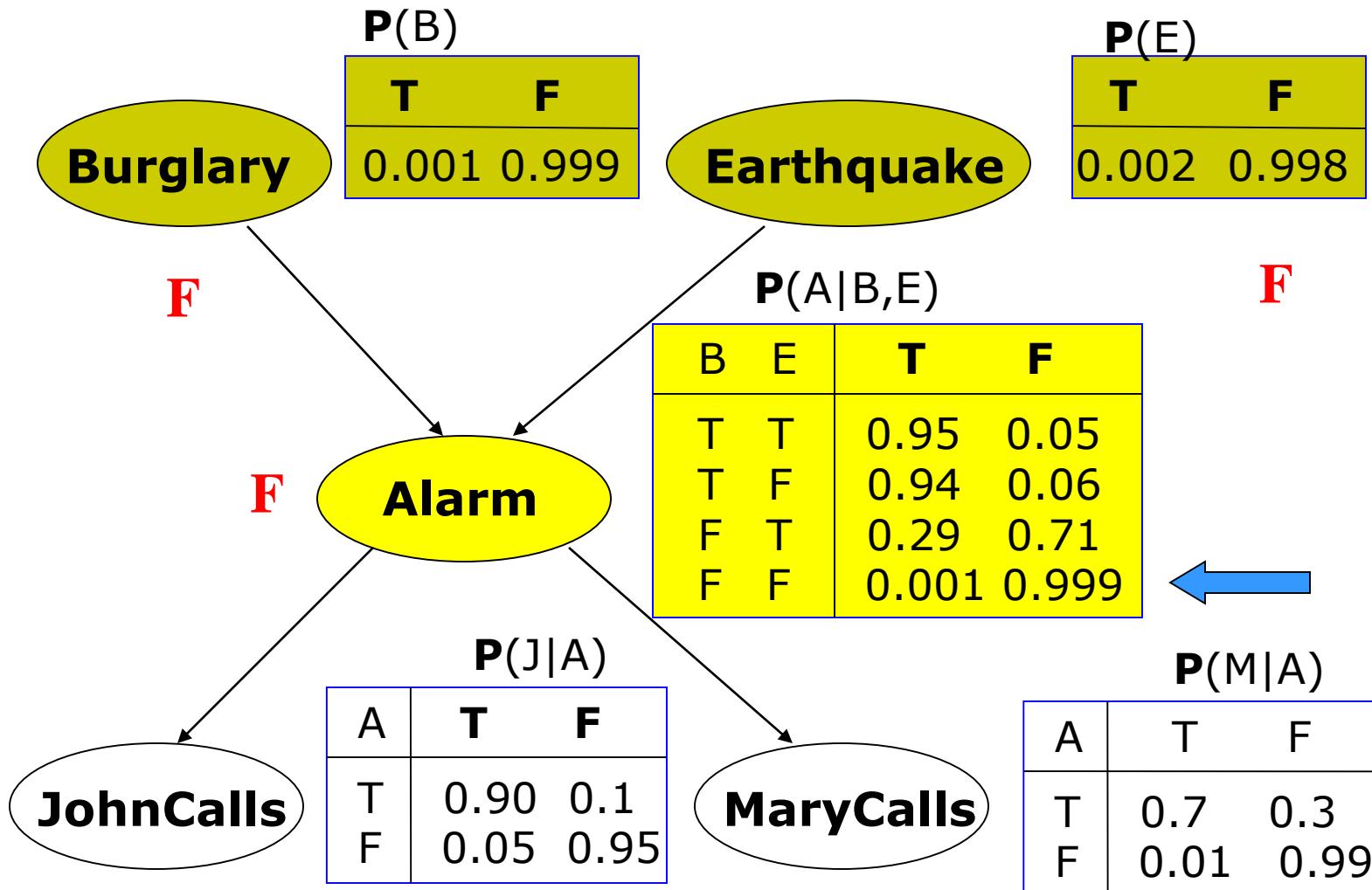
# BBN sampling example



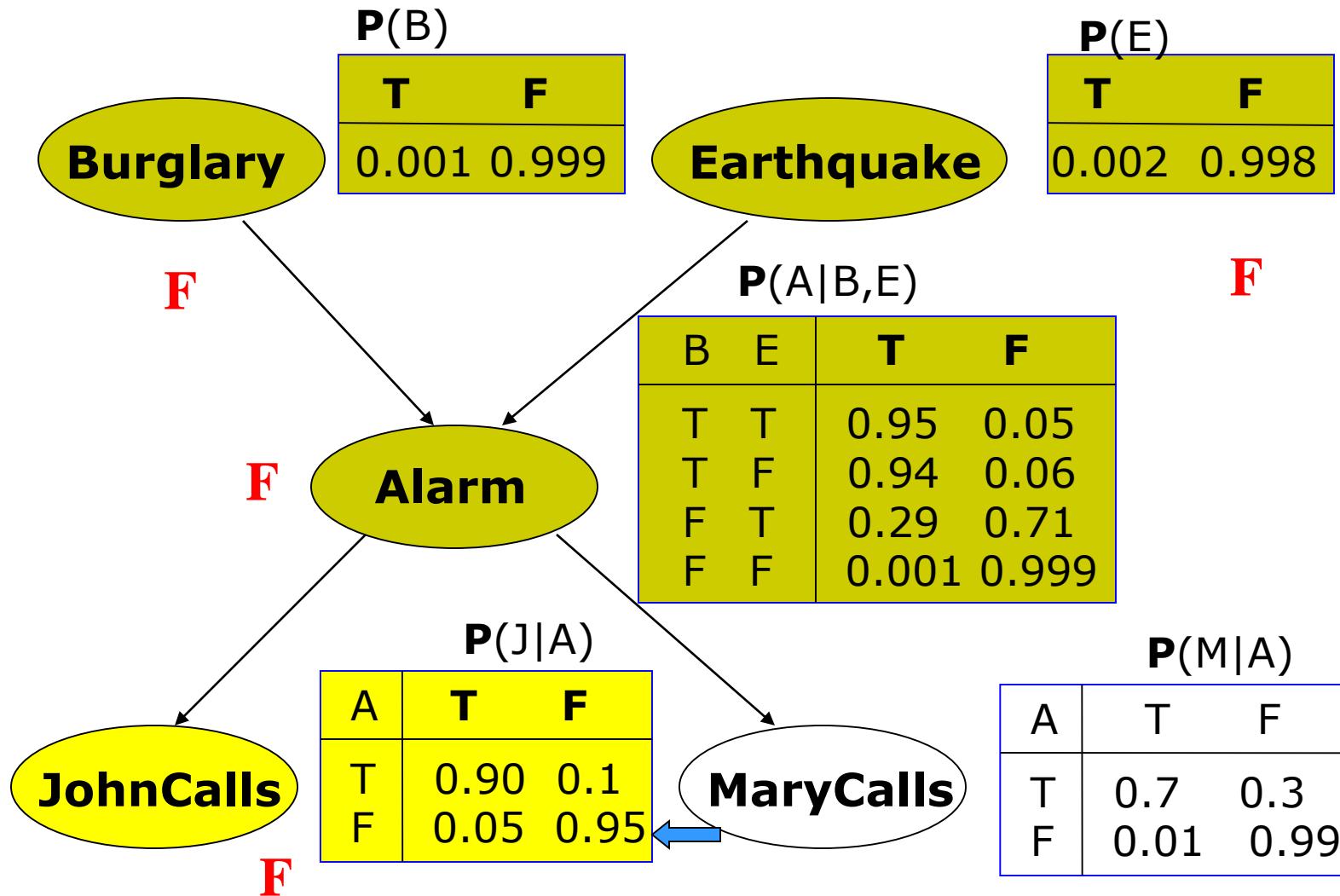
# BBN sampling example



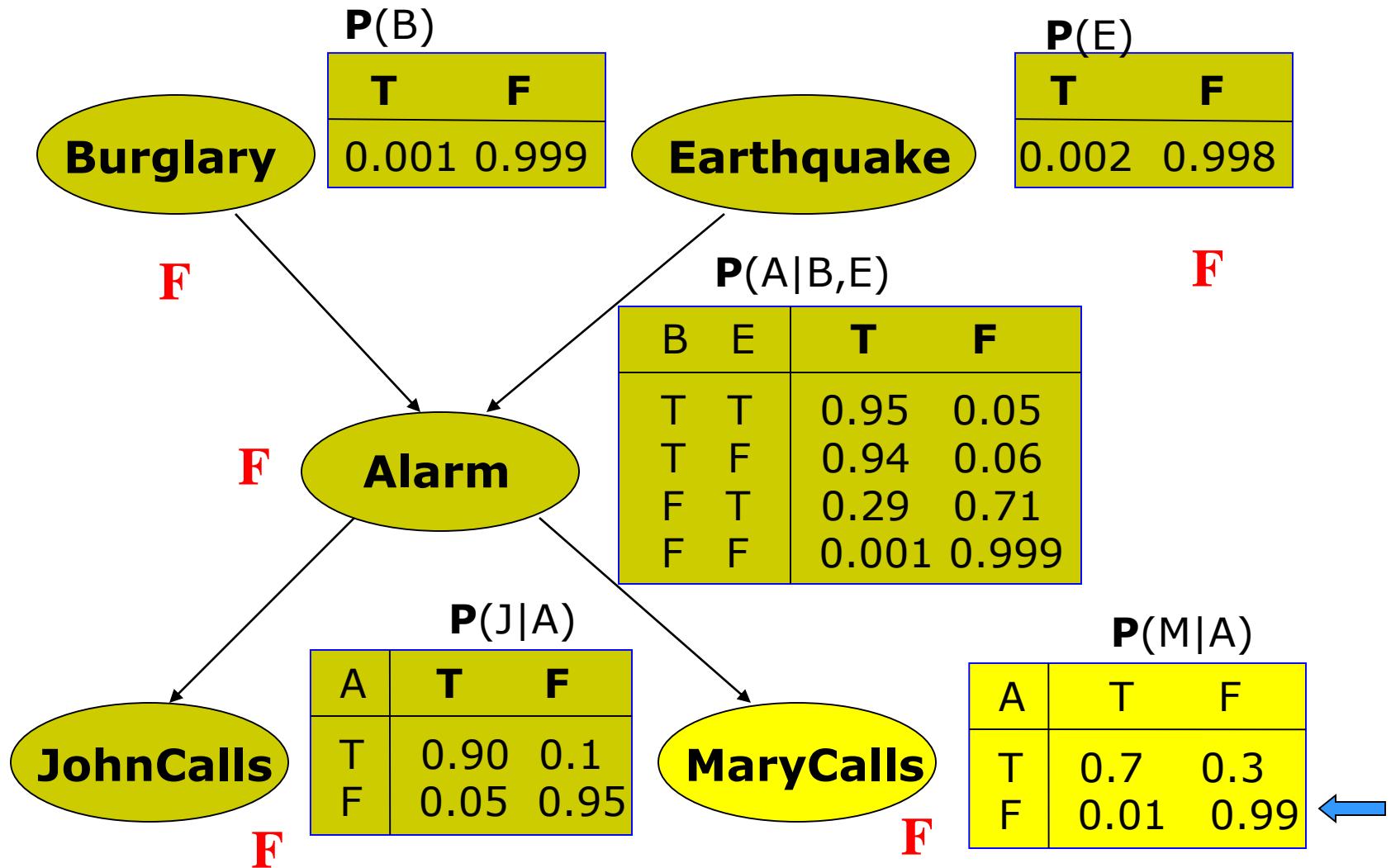
# BBN sampling example



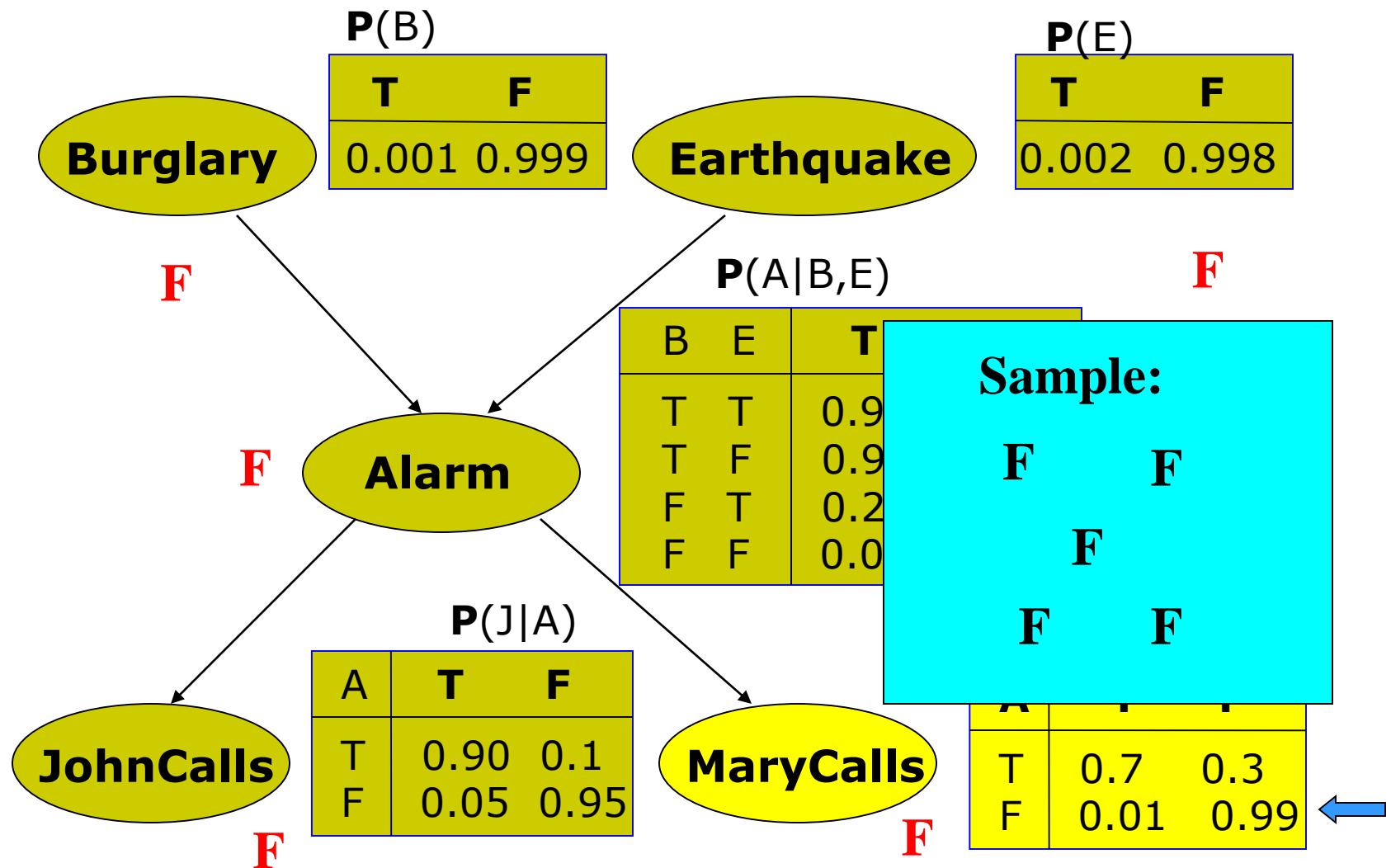
# BBN sampling example



# BBN sampling example



# BBN sampling example



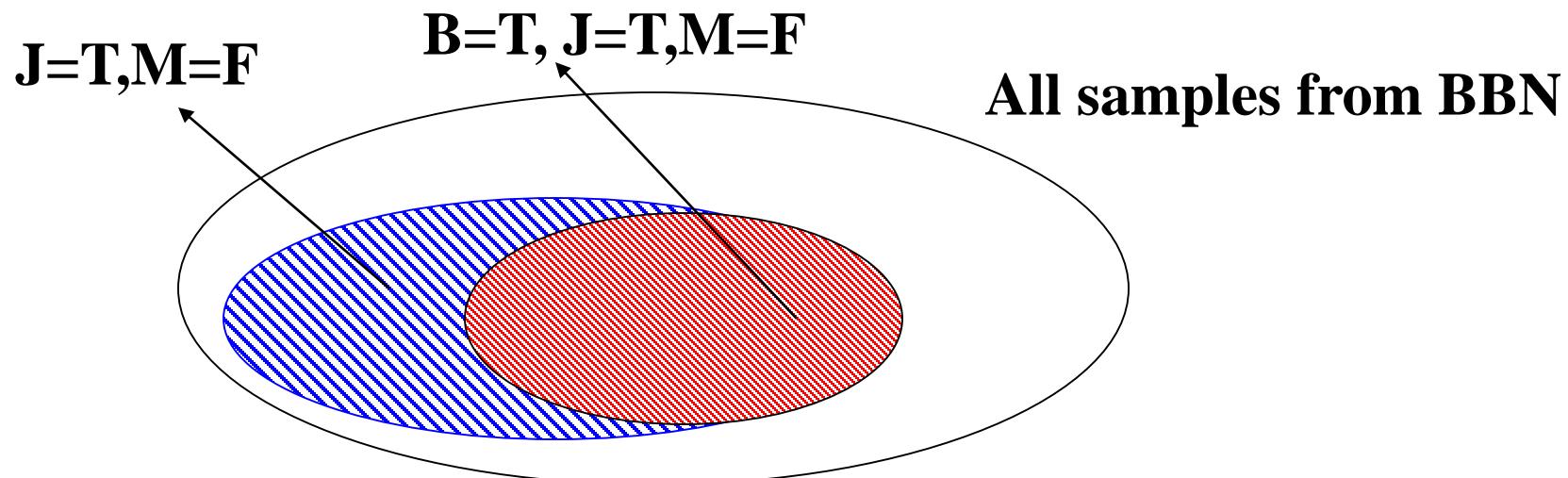
# Monte Carlo approaches

- MC approximation of conditional probabilities:
  - The probability is approximated using sample frequencies
  - Example:

# samples with  $B = T, J = T, M = F$

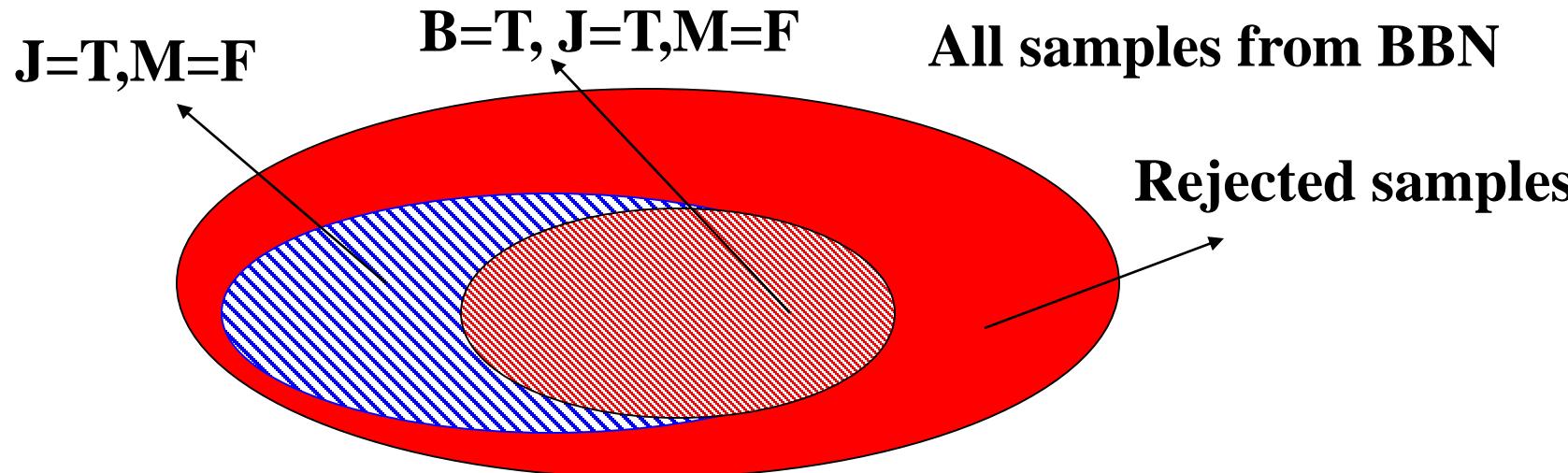
$$\tilde{P}(B = T \mid J = T, M = F) = \frac{N_{B=T, J=T, M=F}}{N_{J=T, M=F}}$$

# samples with  $J = T, M = F$



# Monte Carlo approaches

- **Rejection sampling**
  - Generate samples from the full joint by sampling BBN
  - Use only samples that agree with the condition, the remaining samples are rejected
- **Problem:** many samples can be rejected



# Likelihood weighting

**Idea:** generate only samples consistent with an evidence (or conditioning event)

- Benefit: Avoids inefficiencies of rejection sampling

**Problem:**

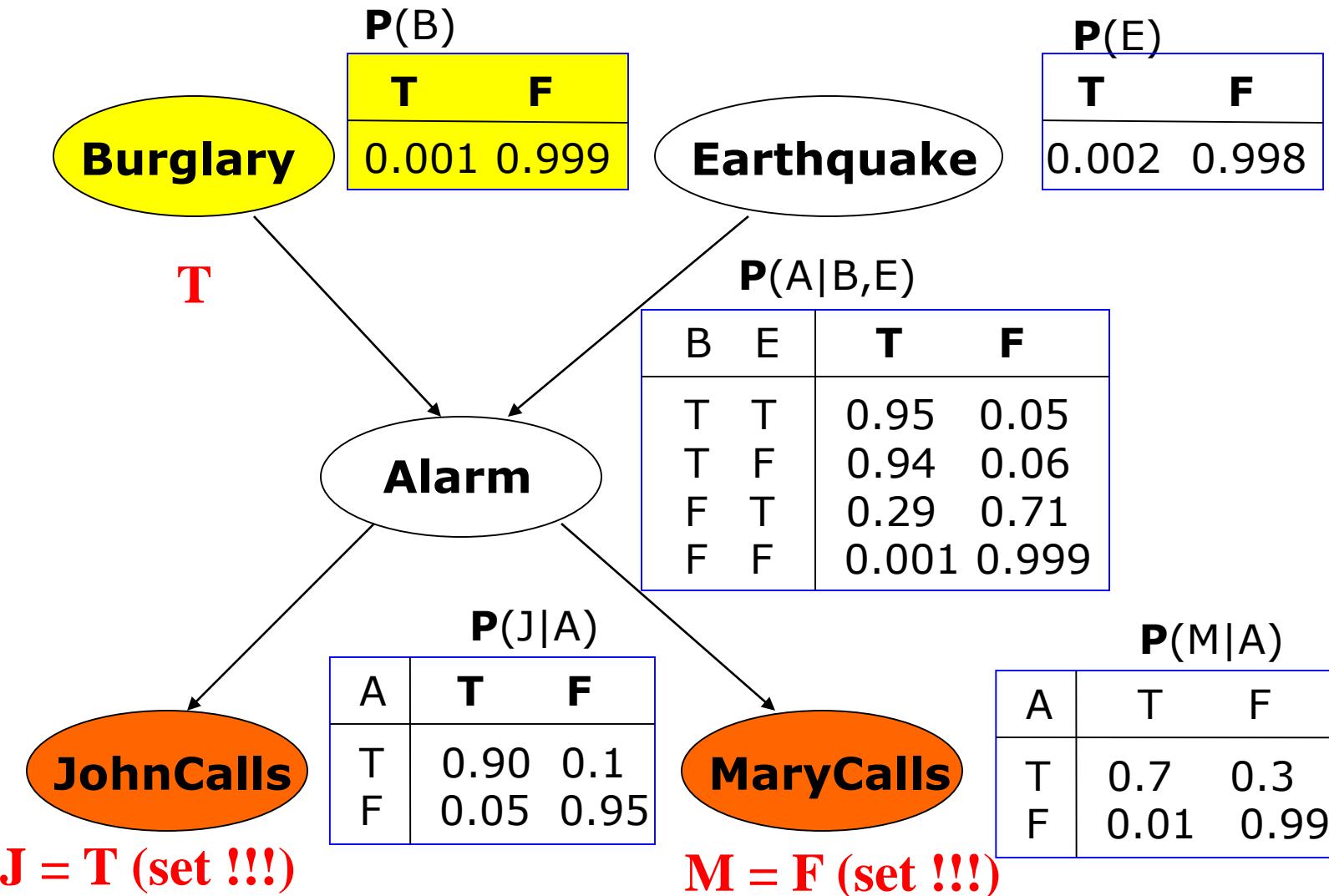
- the distribution generated by enforcing the conditioning variables to set values is biased
- simple counts are not sufficient to estimate the probabilities

**Solution:**

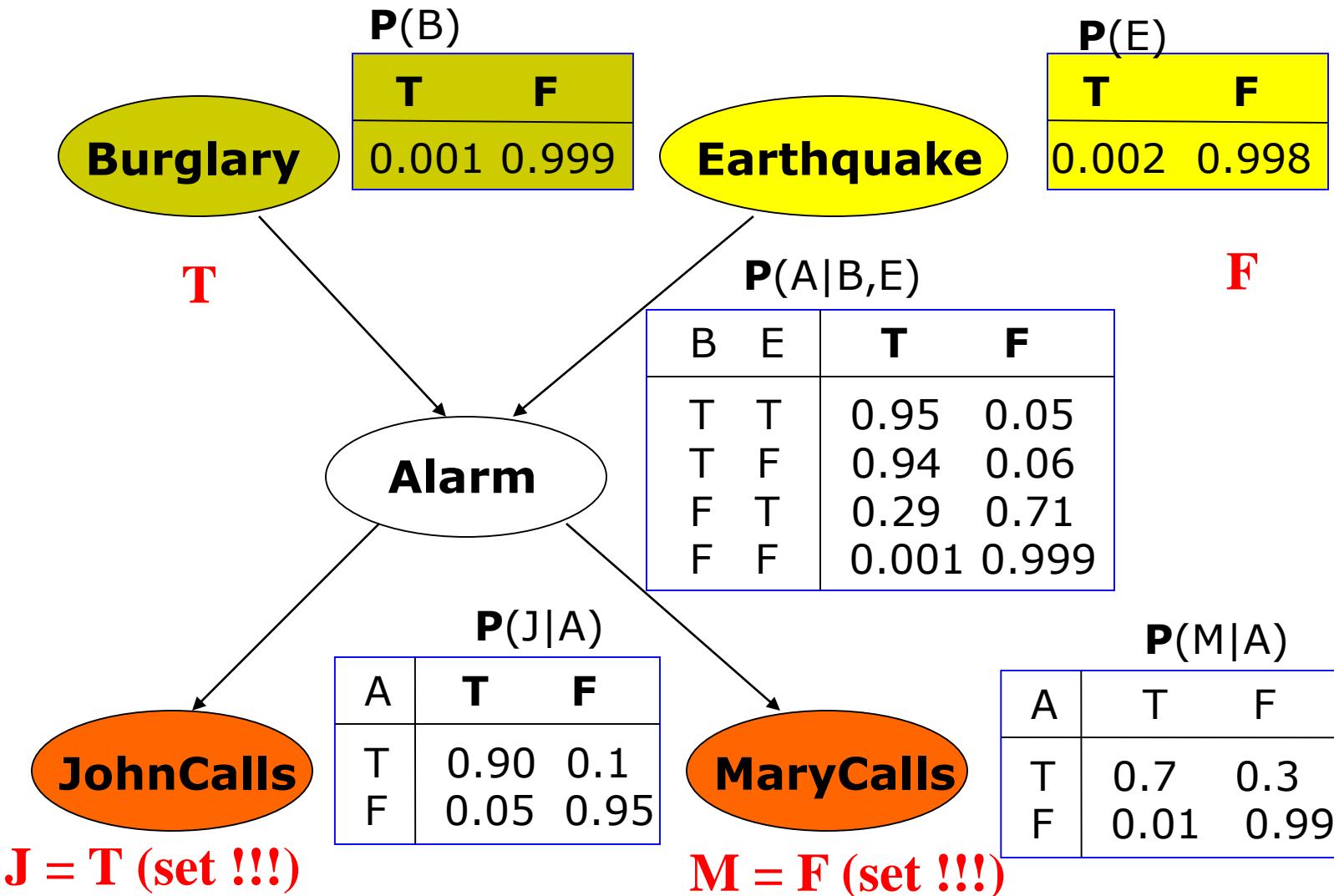
- With every sample keep a weight with which it should count towards the estimate

$$\tilde{P}(B = T \mid J = T, M = F) = \frac{\sum_{\substack{\text{samples with } B=T, M=F \text{ and } J=T}} w_{B=T|J=T, M=F}}{\sum_{\substack{\text{samples with any value of } B \text{ and } J=T, M=F}} w_{B=x|J=T, M=F}}$$

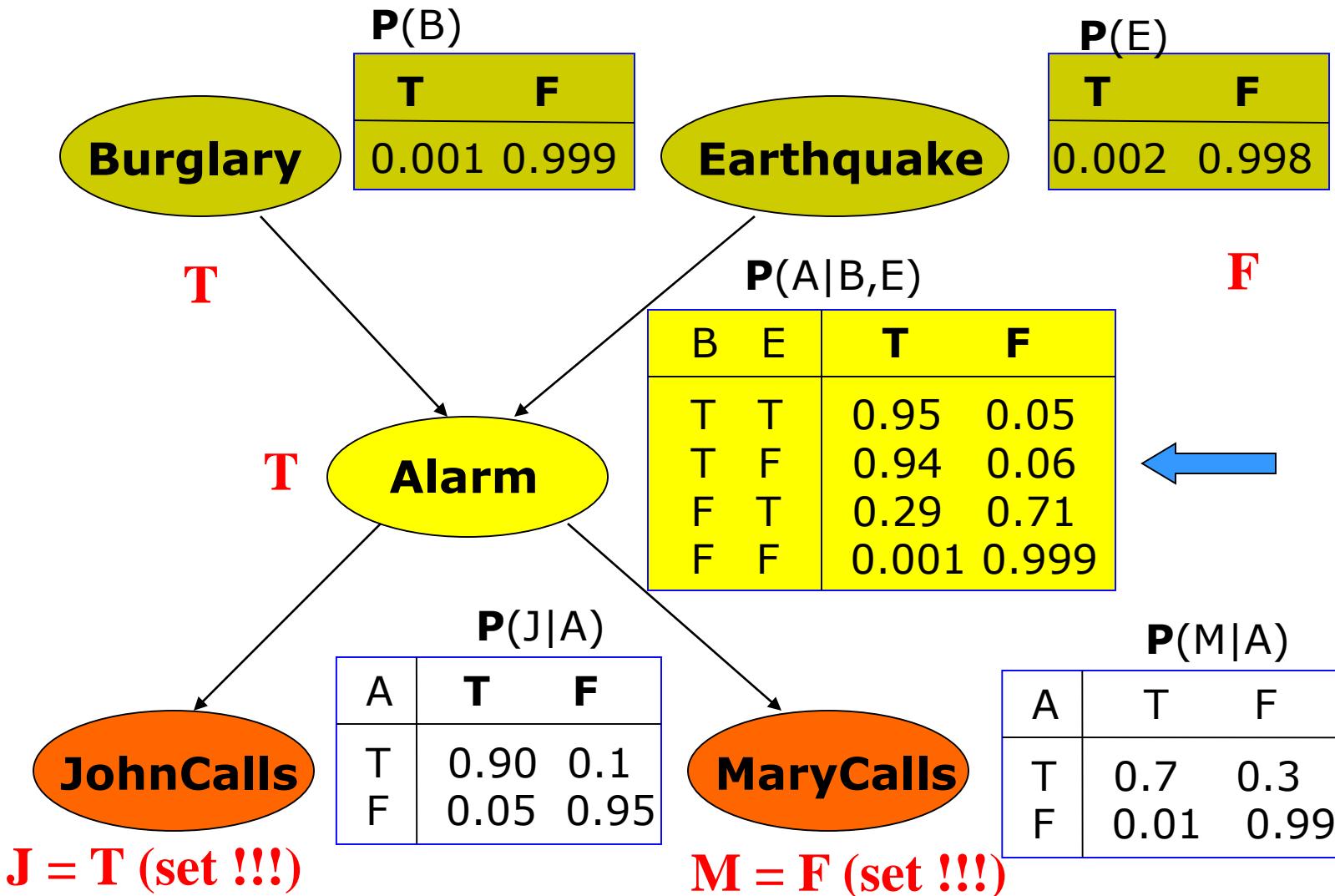
# BBN likelihood weighting example



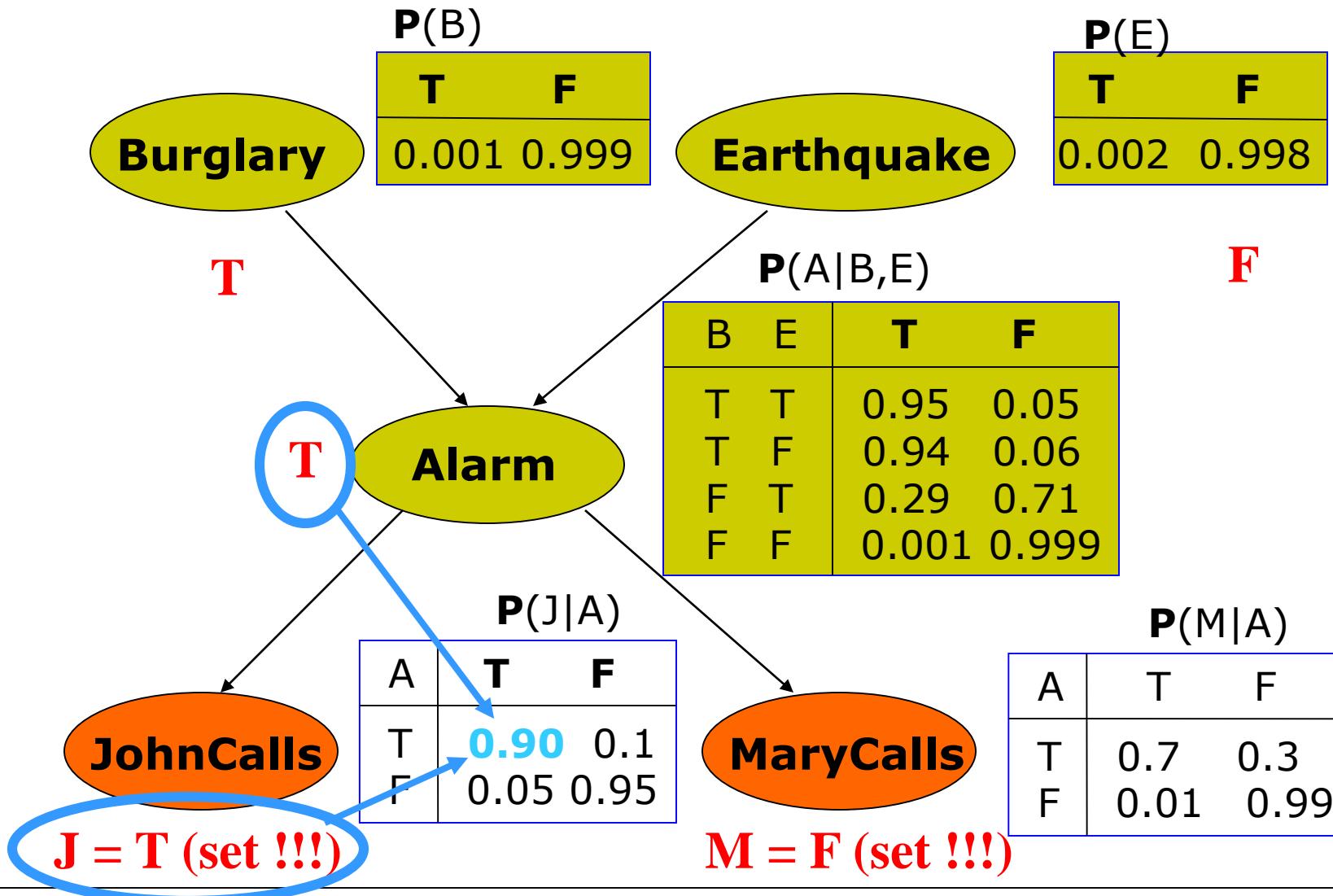
# BBN likelihood weighting example



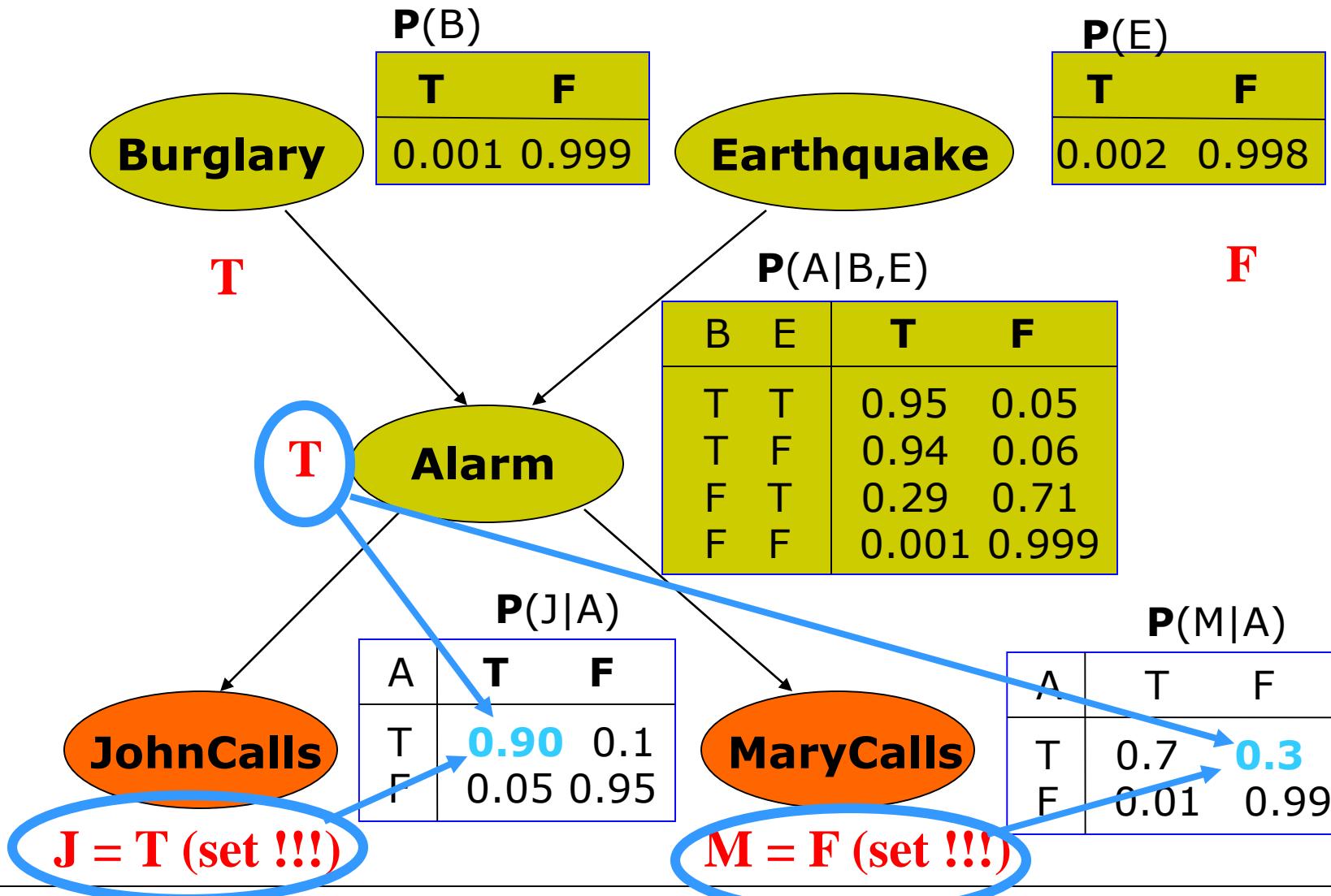
# BBN likelihood weighting example



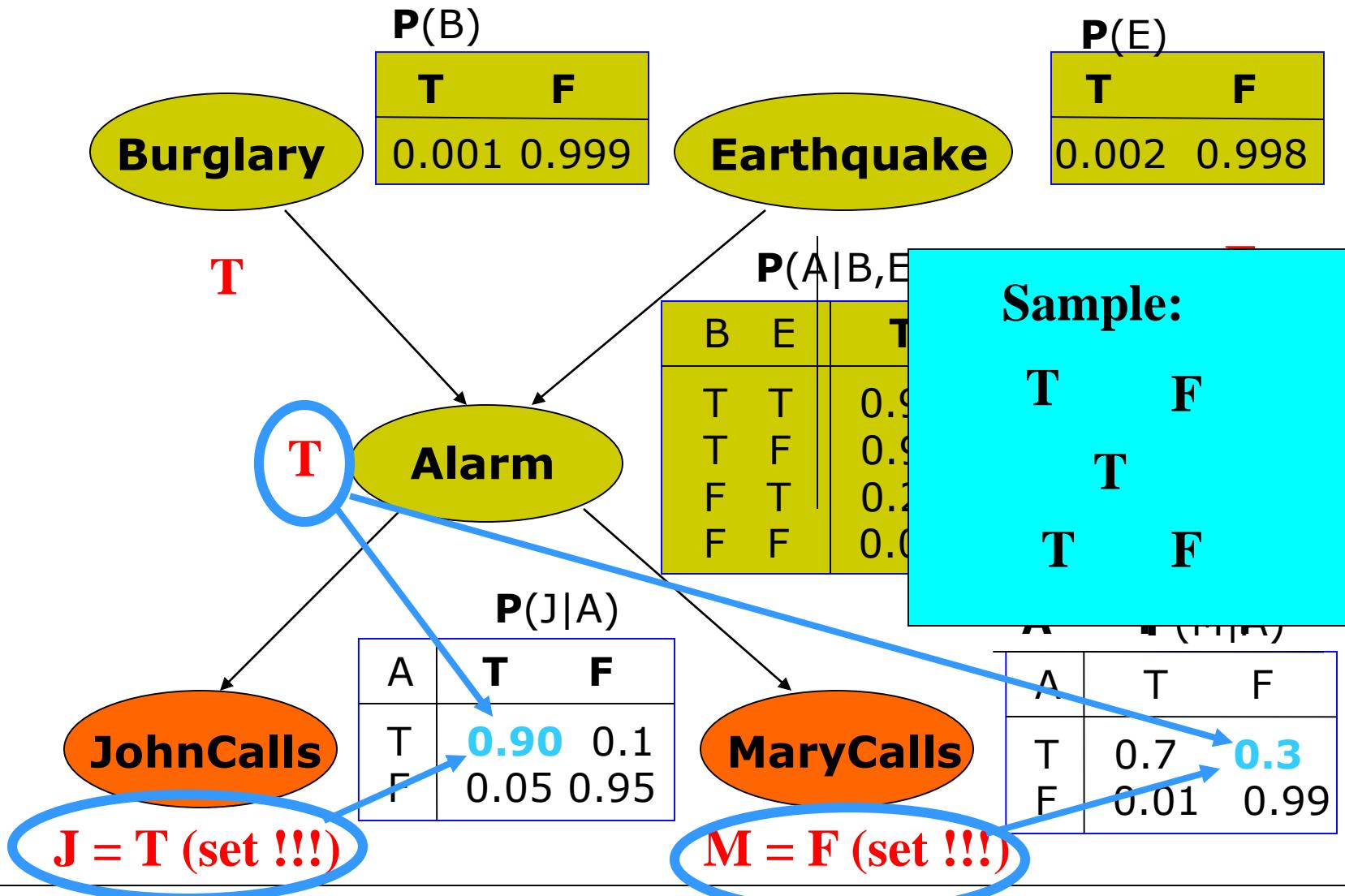
# BBN likelihood weighting example



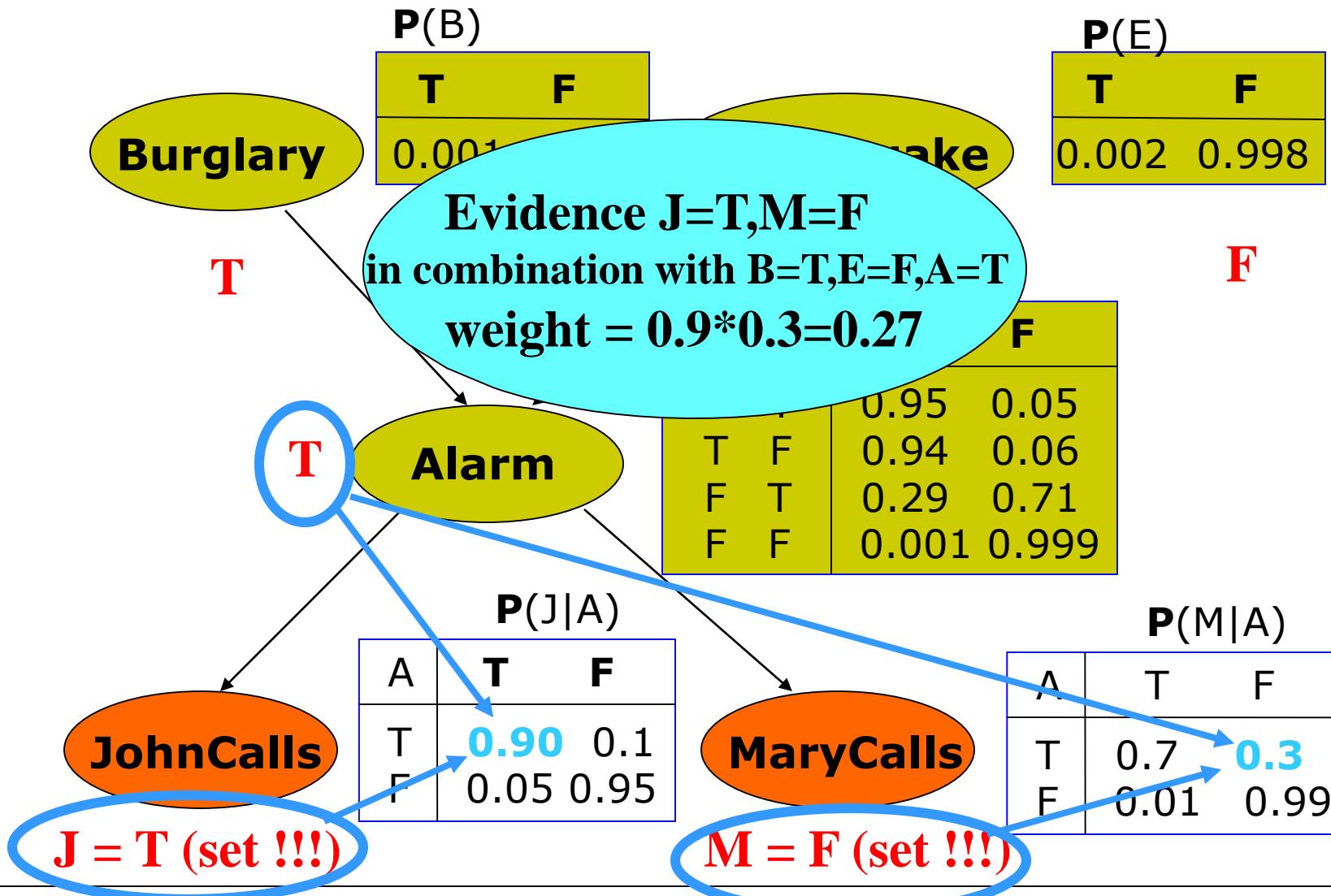
# BBN likelihood weighting example



# BBN likelihood weighting example

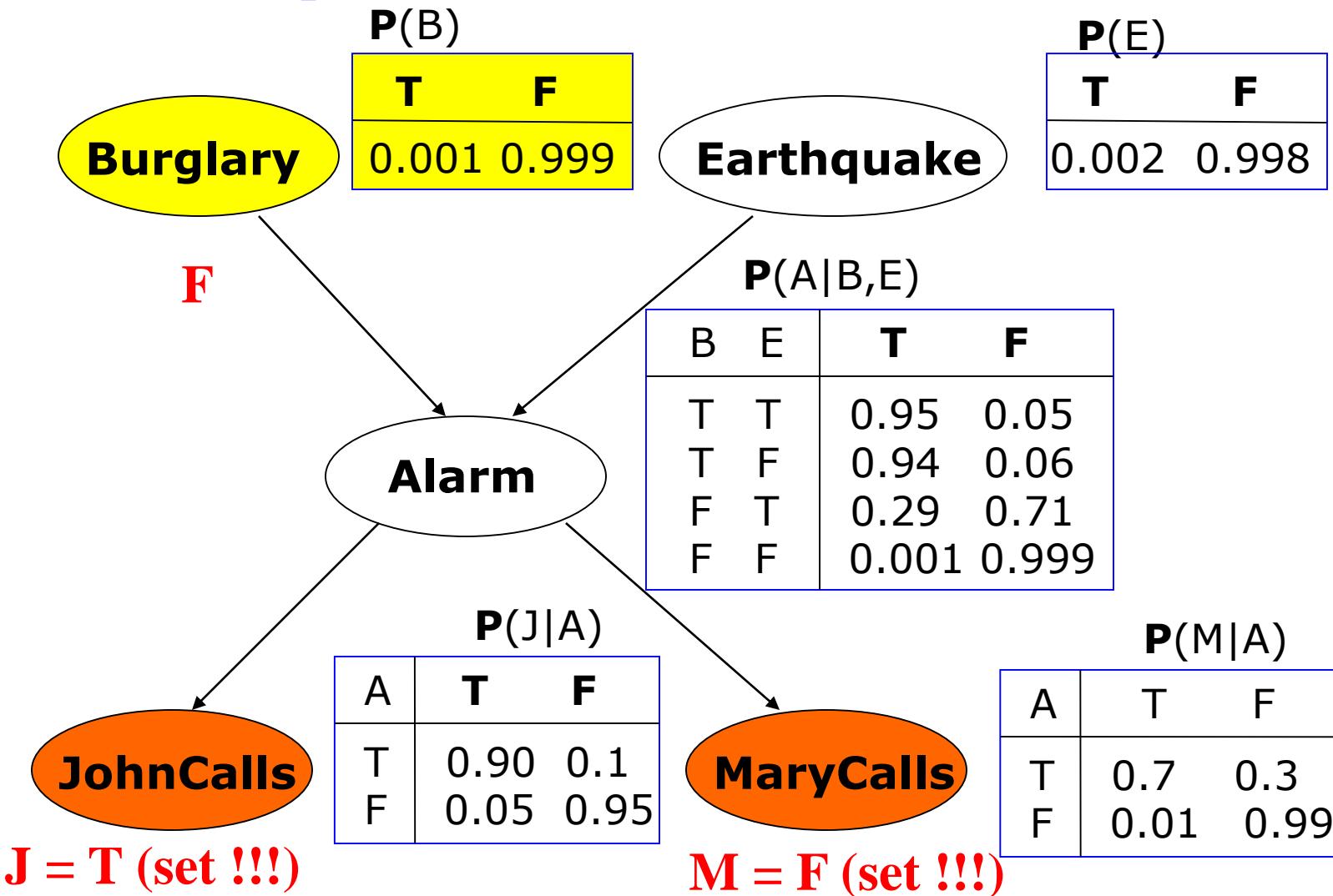


# BBN likelihood weighting example



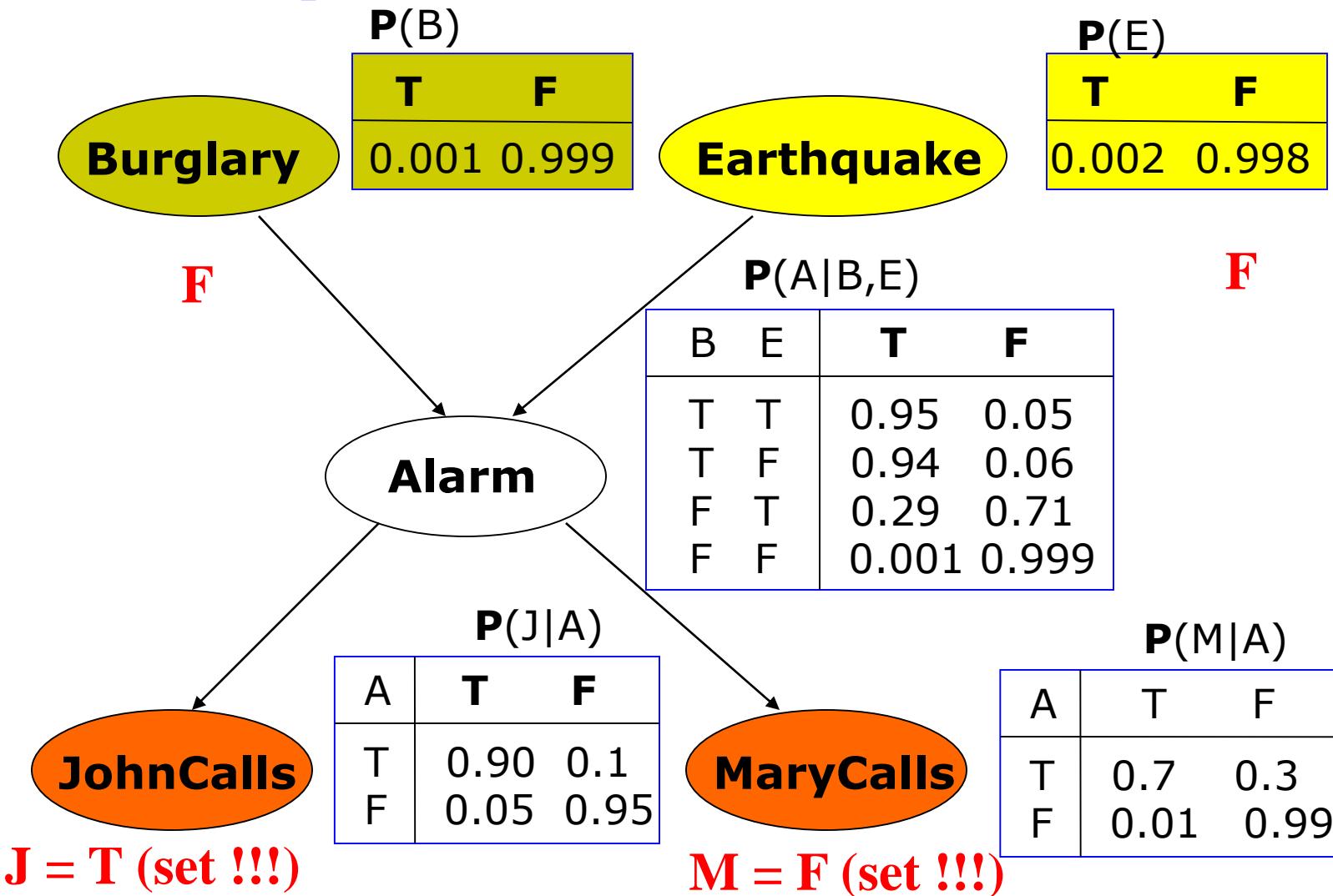
# BBN likelihood weighting example

## Second sample



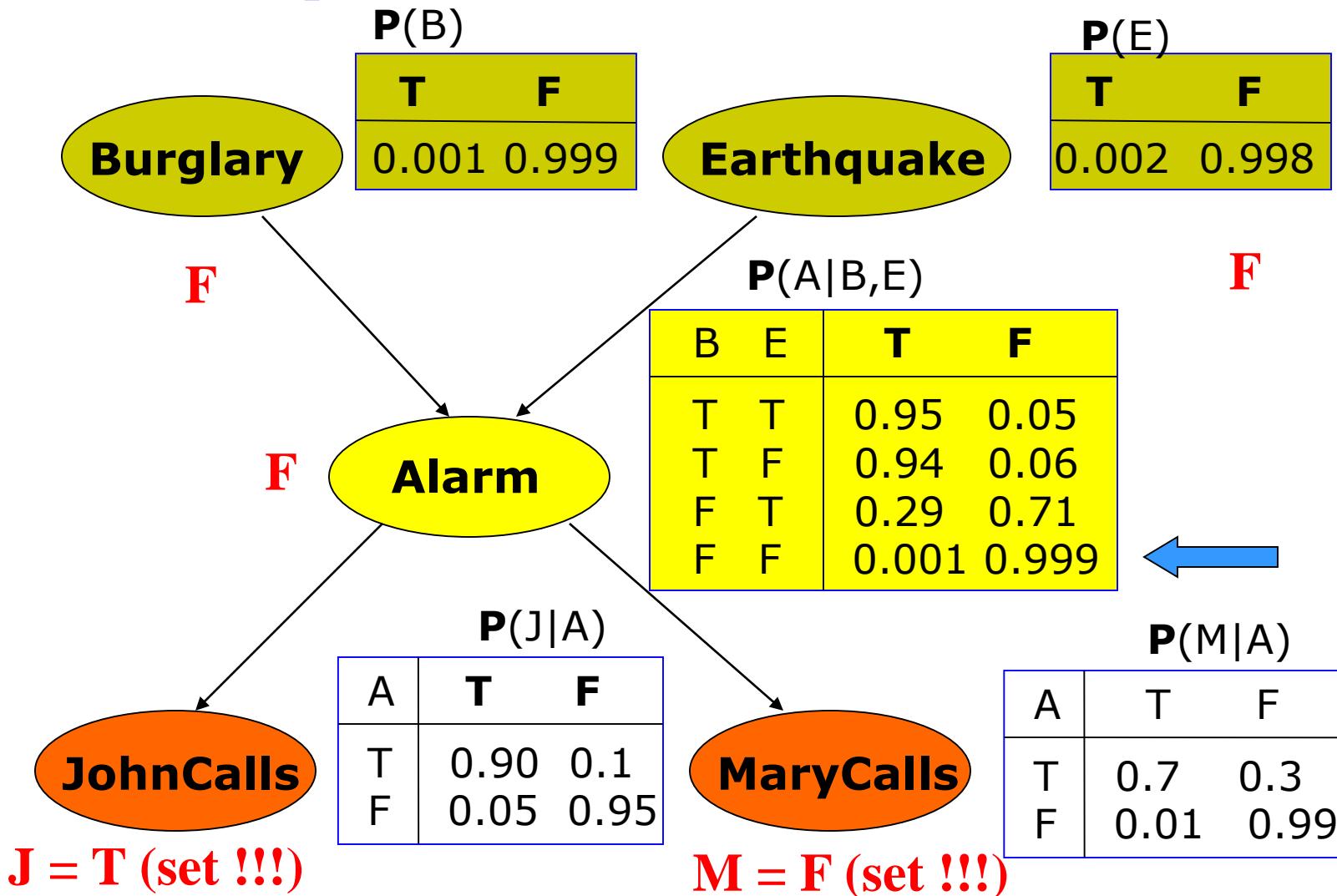
# BBN likelihood weighting example

## Second sample



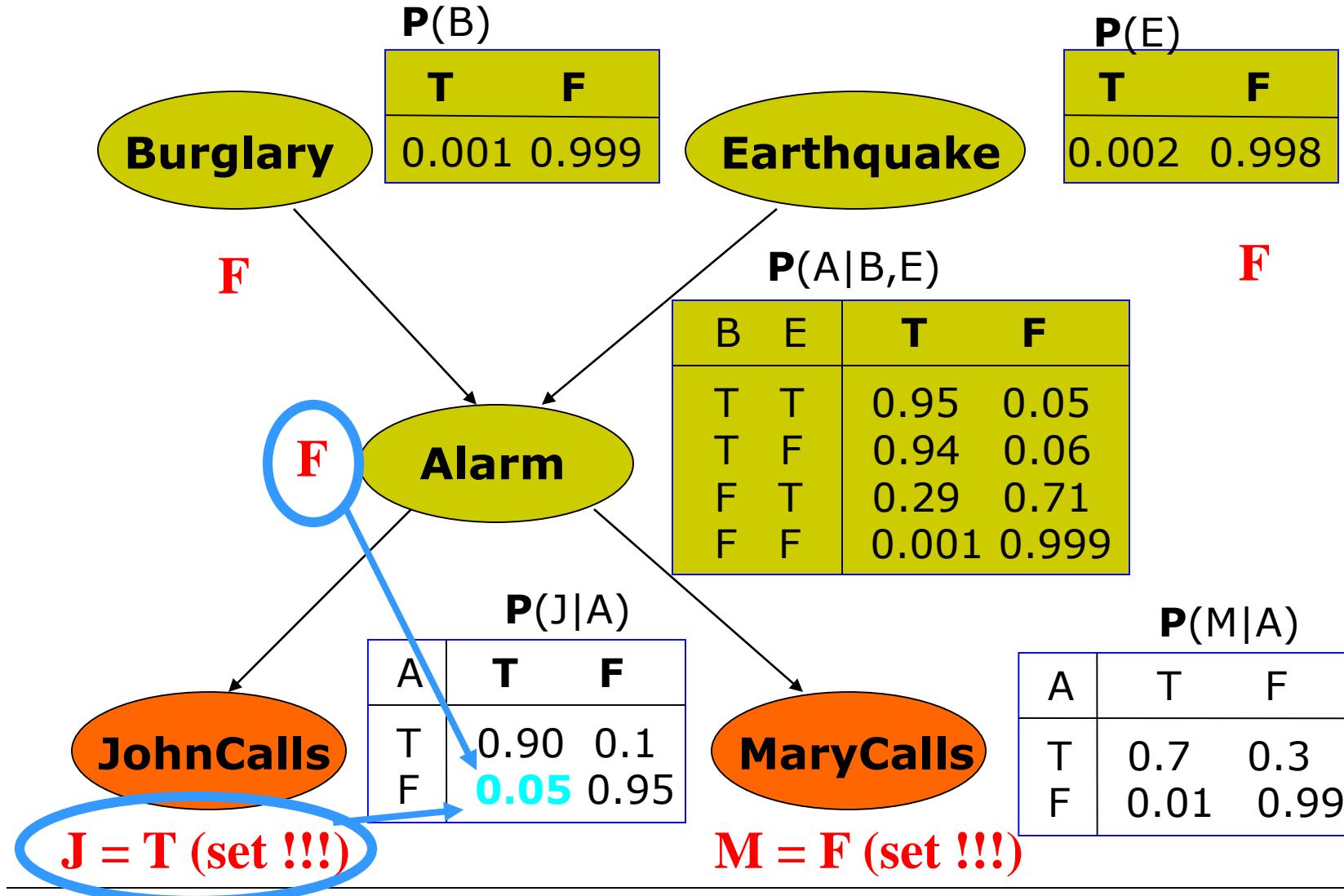
# BBN likelihood weighting example

## Second sample



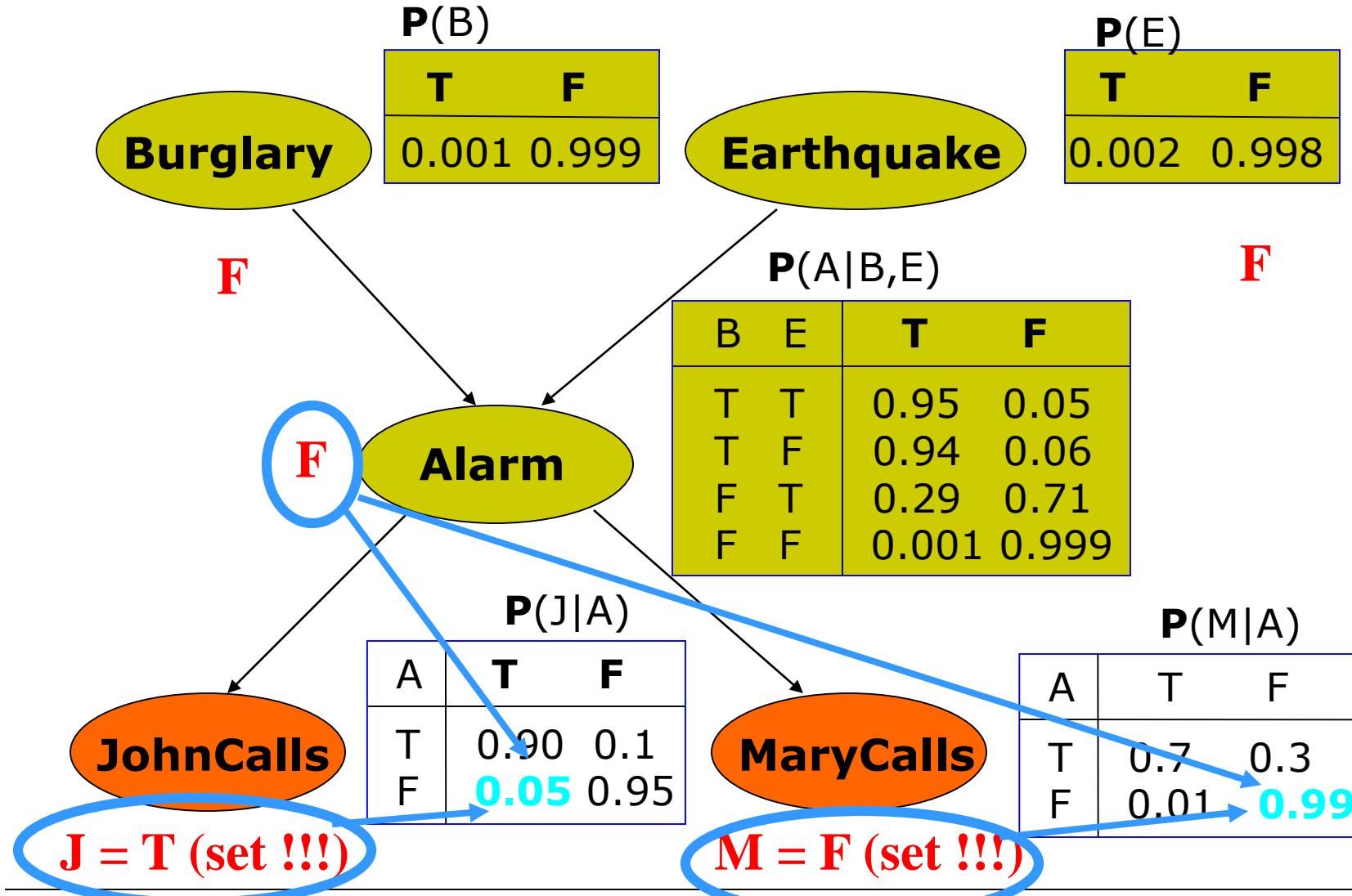
# BBN likelihood weighting example

## Second sample



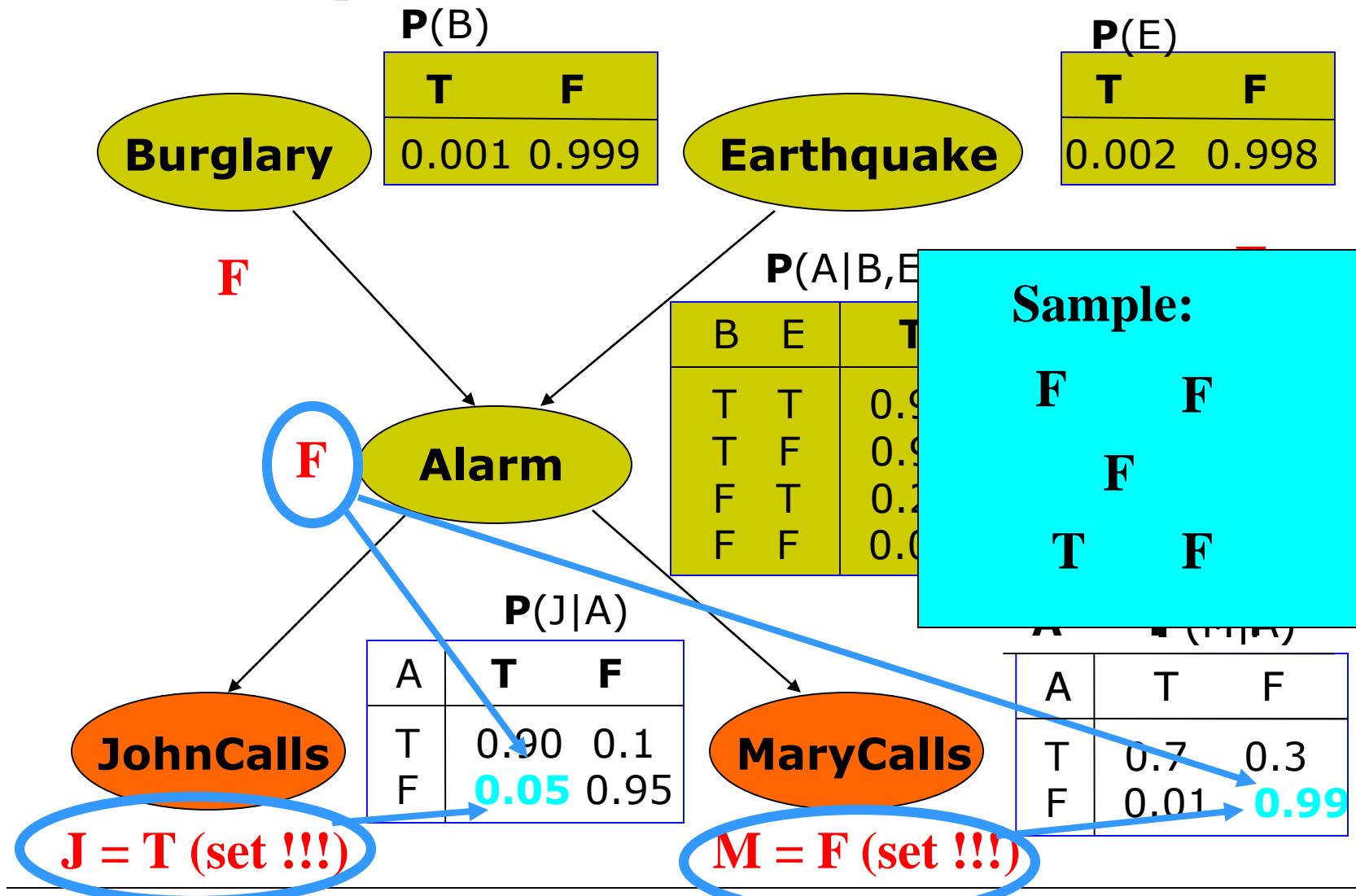
# BBN likelihood weighting example

## Second sample



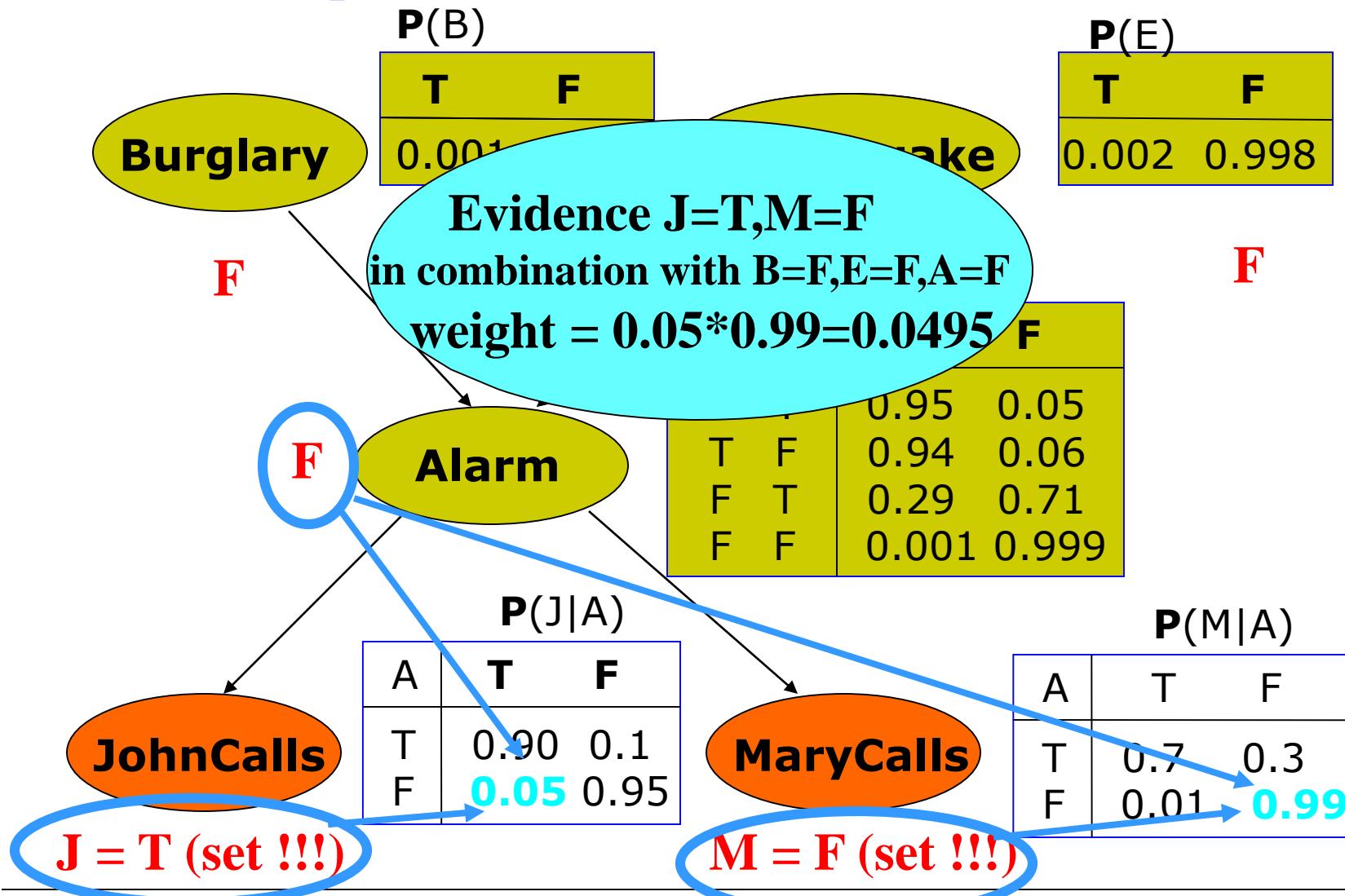
# BBN likelihood weighting example

## Second sample



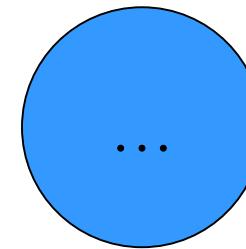
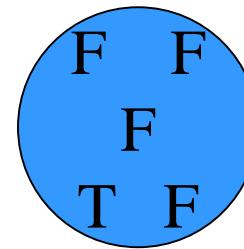
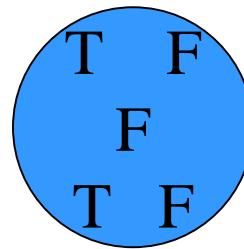
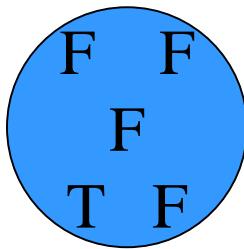
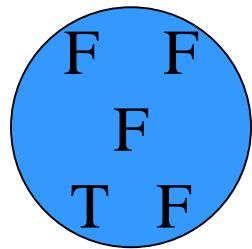
# BBN likelihood weighting example

## Second sample



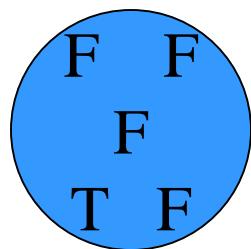
# Likelihood weighting

- Assume we have generated the following M samples:

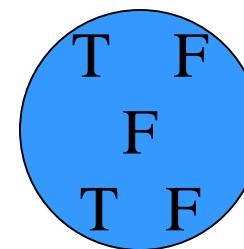


M

**How to make the samples consistent?** Weight each sample by probability with which it agrees with the conditioning evidence P(e).



← Weight 0.0495



← Weight 0.27

$$\tilde{P}(B = T \mid J = T, M = F) = \frac{\sum_{\substack{\text{samples with } B=T, M=F \text{ and } J=T}} w_{B=T|J=T, M=F}}{\sum_{\substack{\text{samples with any value of } B \text{ and } J=T, M=F}} w_{B=x|J=T, M=F}}$$