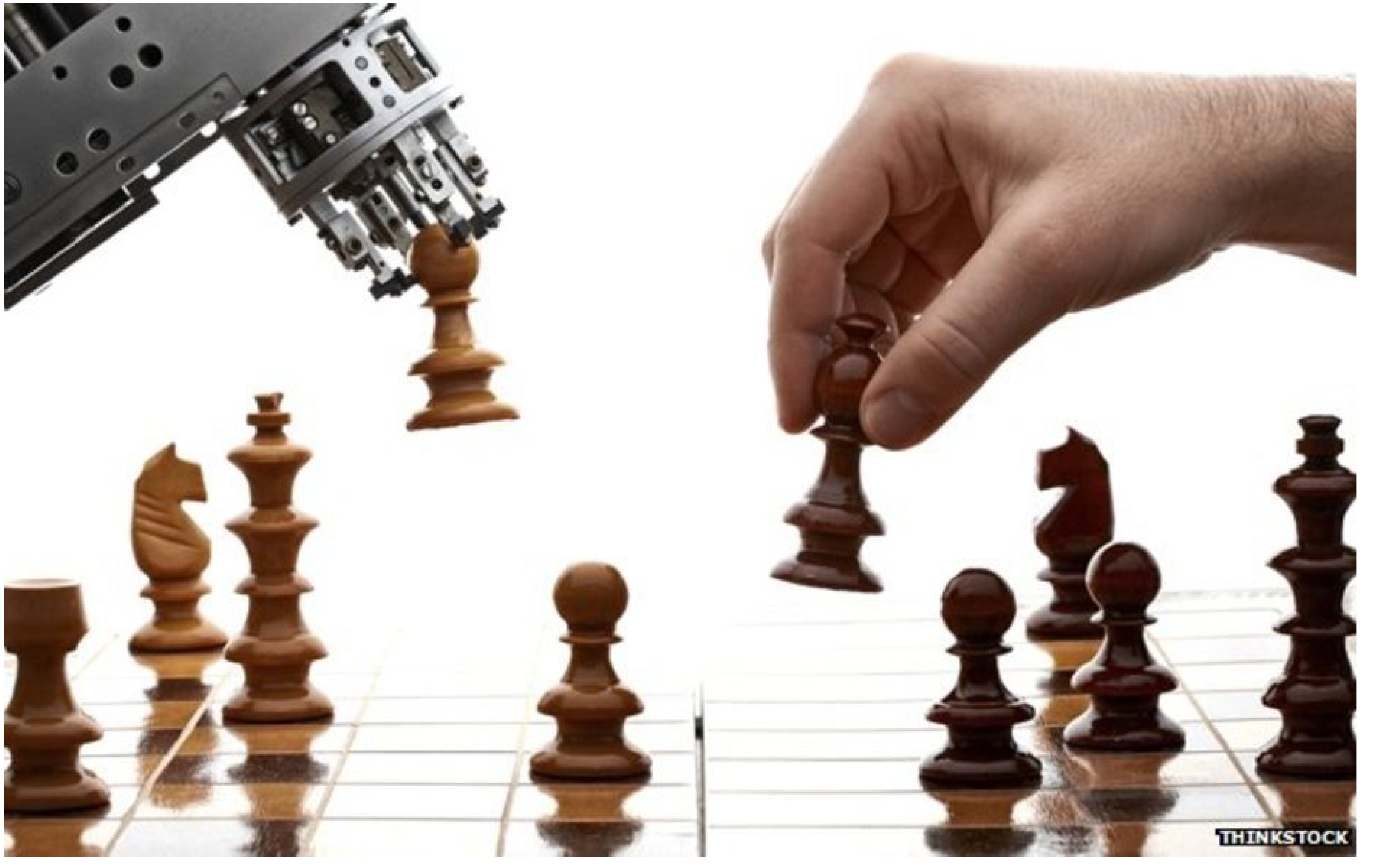


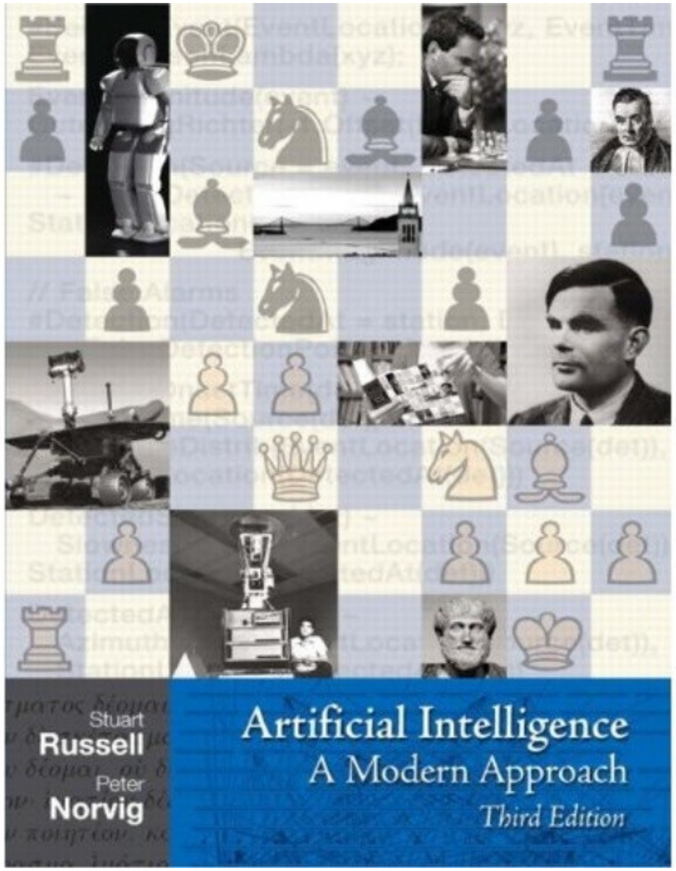
Introduction to Artificial Intelligence

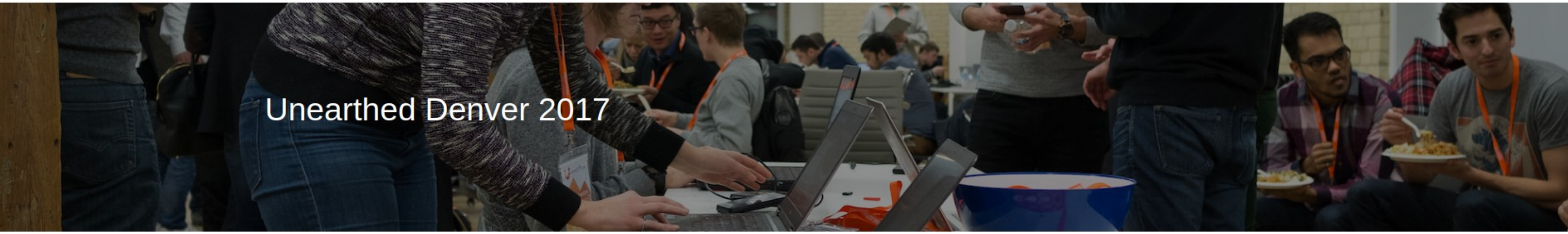
COSC 4550 / COSC 5550

Professor Cheney

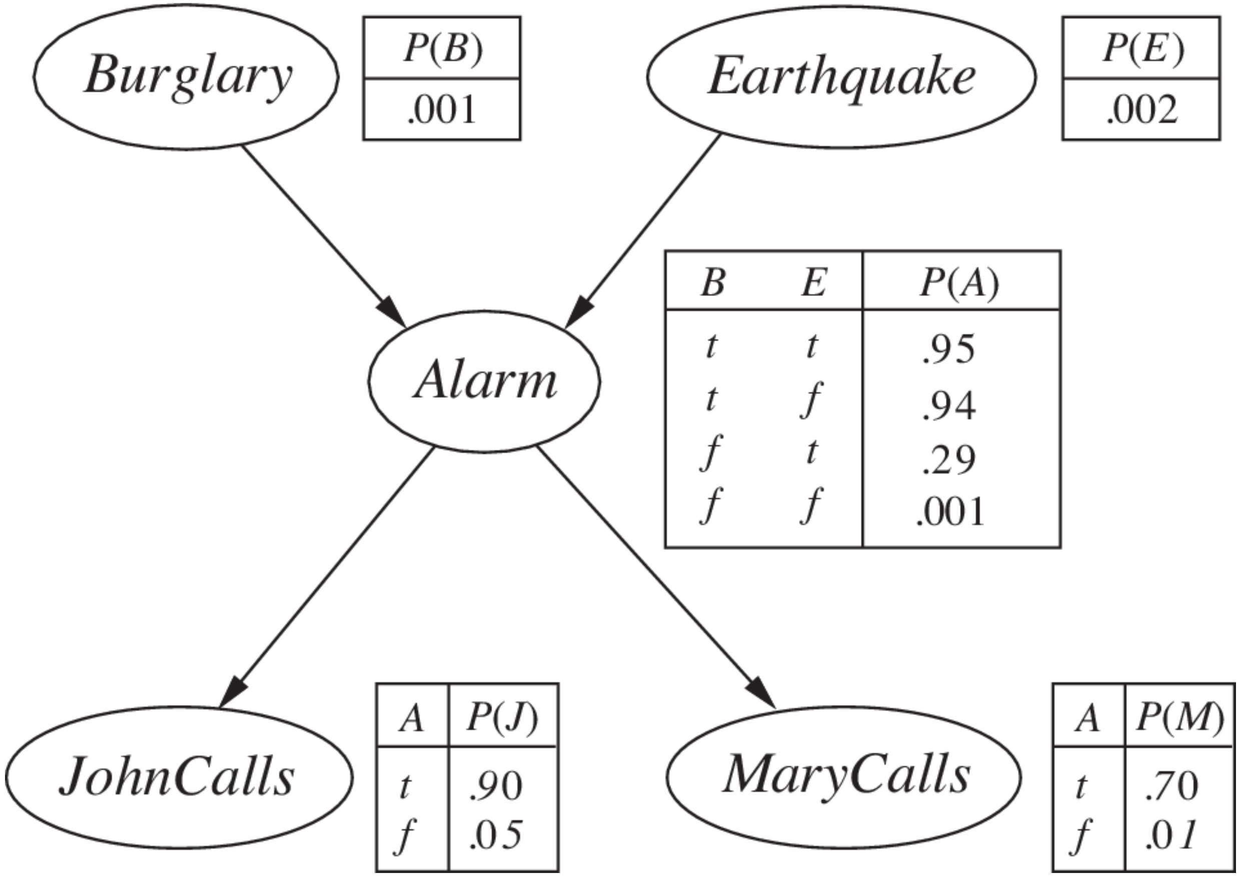
9/29/17

slides are ahead of reading on website schedule… sorry





Unearthed Denver 2017



full joint probability: $O(2^n)$

worst-case Bayes net: $O(n*2^n)$

how to we actually find the values for the conditional probability tables?

build full joint distribution table
(don't need to know any structure of the problem – all independent)

	toothache		*¬toothache*	
	catch	*¬catch*	*catch*	*¬catch*
cavity	0.108	0.012	0.072	0.008
¬cavity	0.016	0.064	0.144	0.576

0.12

0.08

Figure 13.3 A full joint distribution for the *Toothache*, *Cavity*, *Catch* world.

sum over all unspecified variables…

P(Cavity | toothache) = Σ_{catch} P(Cavity, toothache, catch)

"exact inference"

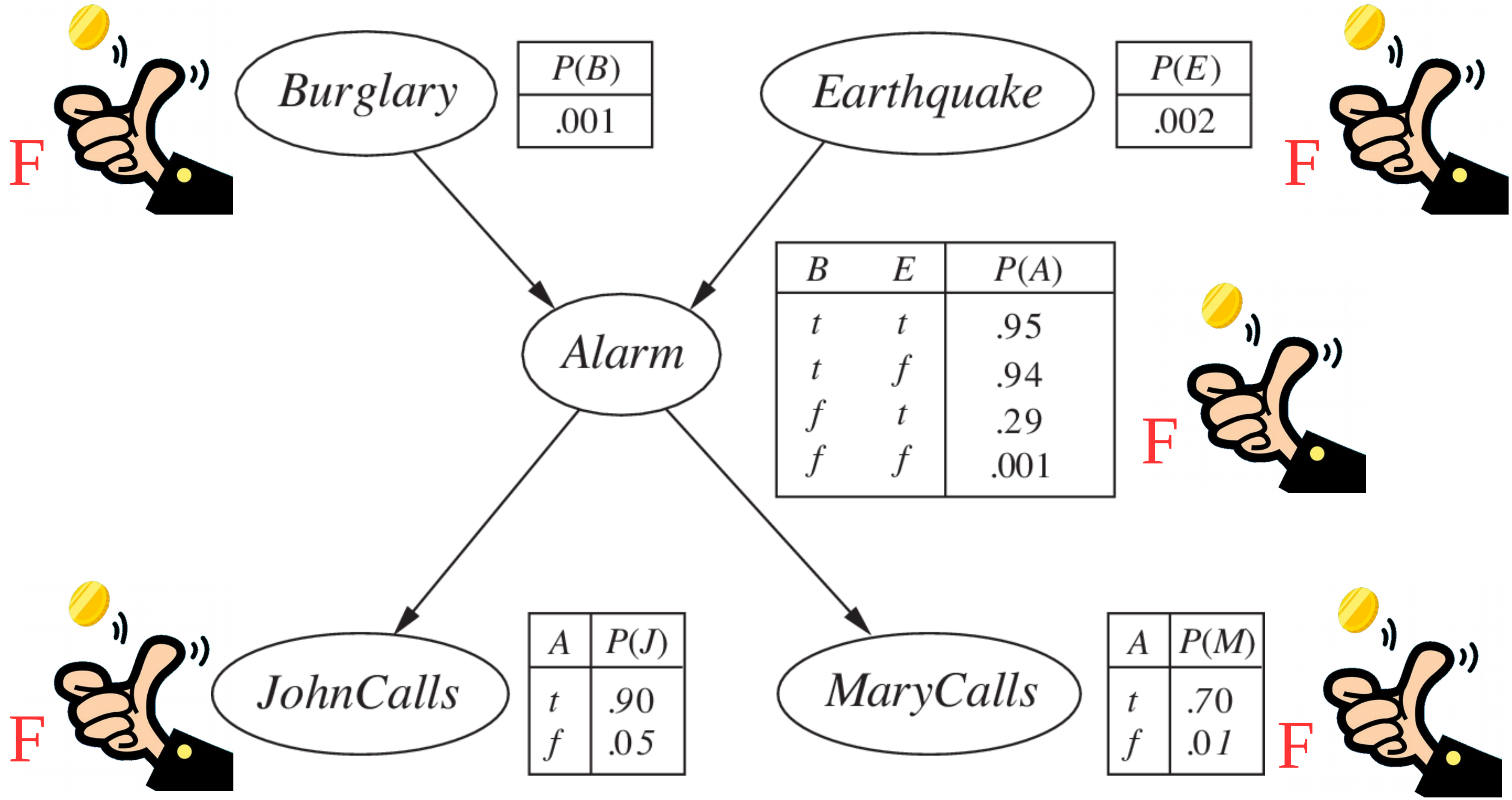
exact inference is exponential in the number of nodes
(full joint distribution is 2^n)

in real systems this is expensive (lots of parameters to learn)!!!

(even after some small tricks – see book)

let's sample on the network instead!

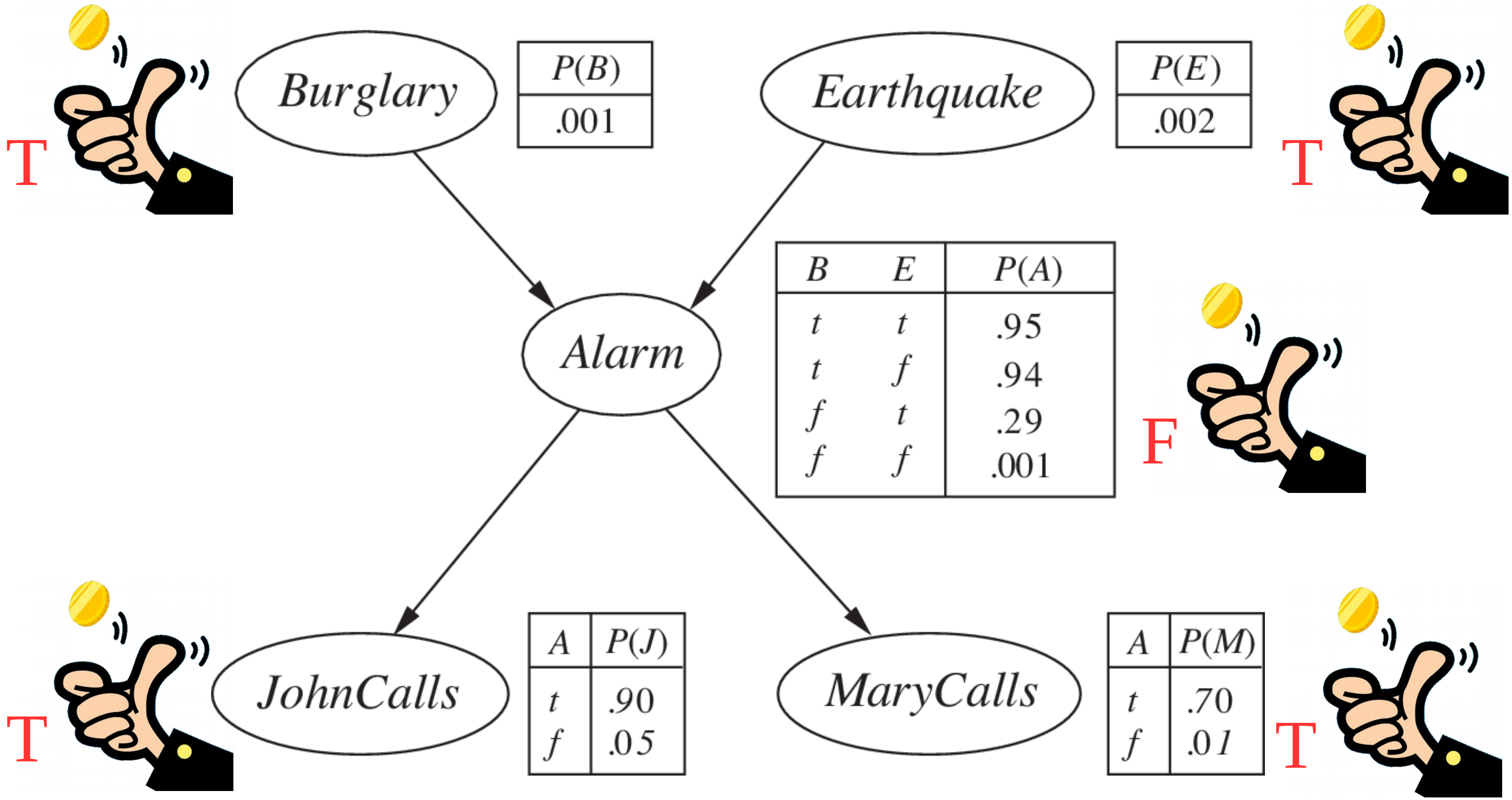
approximate inference in Bayes nets
(Monte Carlo methods)



what's the problem here?

over-sampling/under-sampling some (extreme) trajectories

P(happy | winLottery)?



keep sampling until you get what you're looking for
(keep that one and throw out all the ones before it)

“rejection sampling”

always get what you're looking for

but may take you a long time

especially if some events are unlikely

(or even if you have to get many likely events simultaneously)

$0.90^{100} = 0.00003$

“likelihood weighting”

rather than throwing out samples that don't match you evidence

use all data, but weight it by the likihood that
the (parent) variables you sampled up to that point
would occur by chance

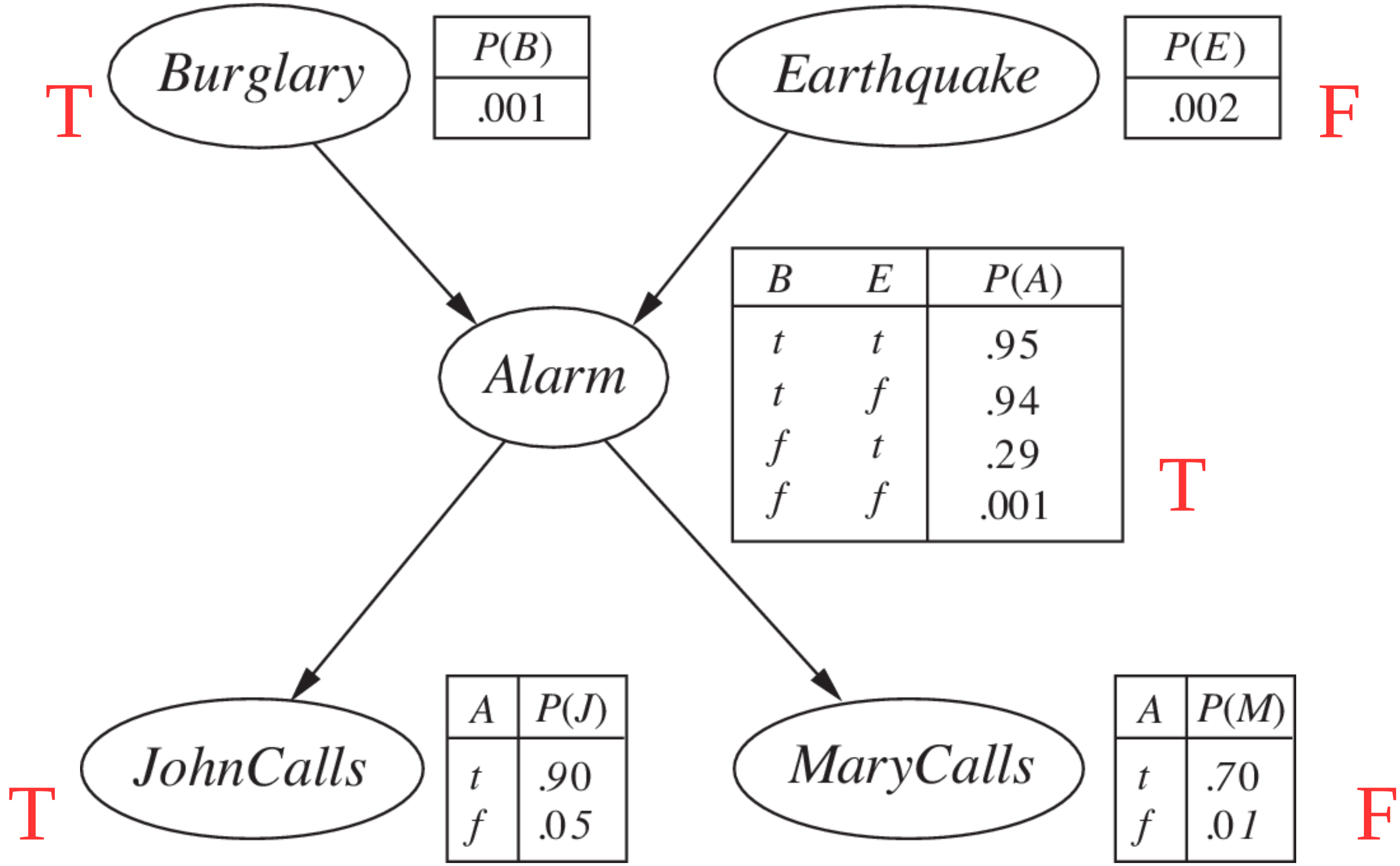
puts less emphasis on unlikely data

(pseudocode in your book)

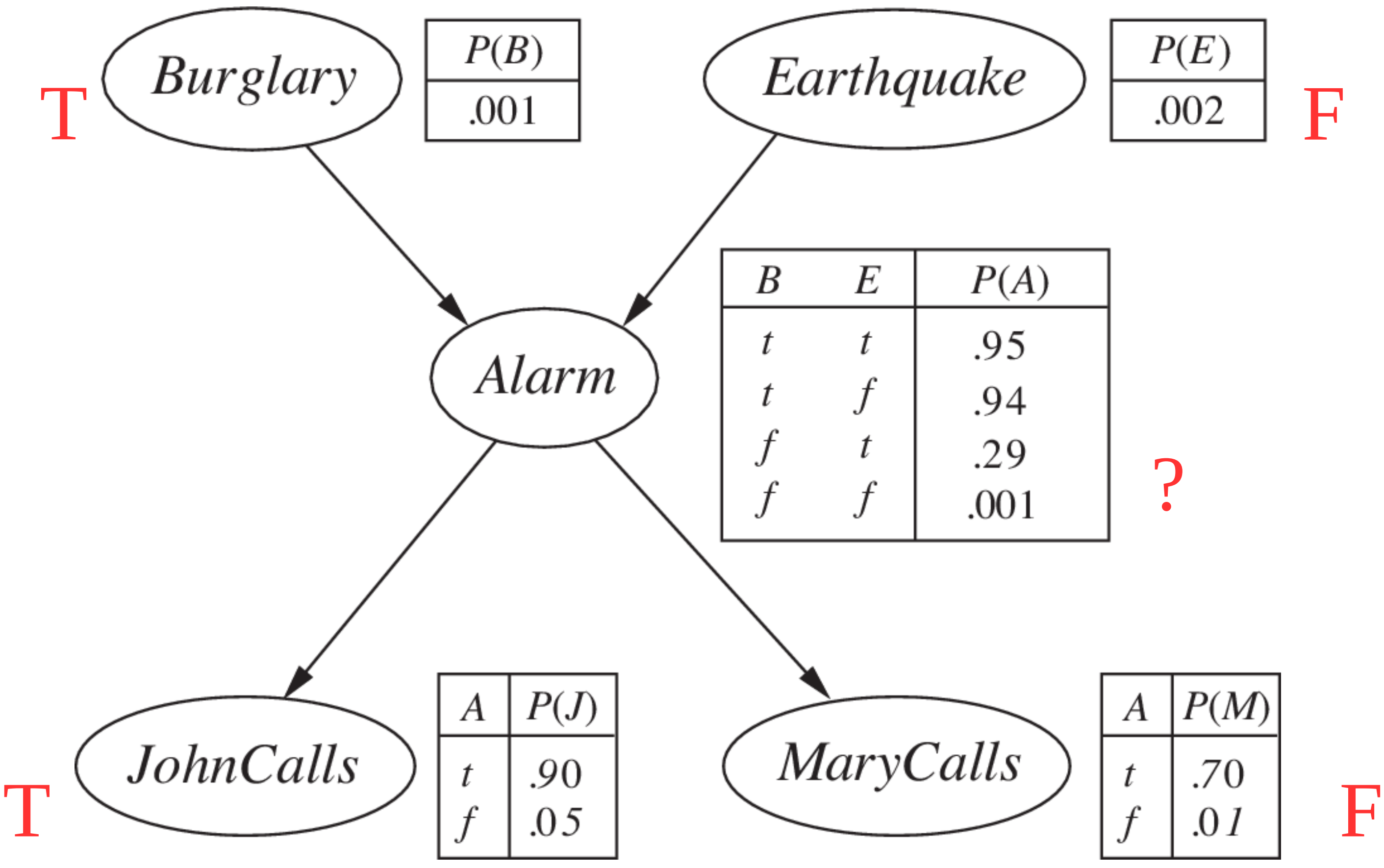
Monte Carlo Markov Chain simulation
(e.g. “Gibbs sampling”)

rather than drawing random samples of all variables,
only sample on random variable at a time
(conditioned on the current values of the others)

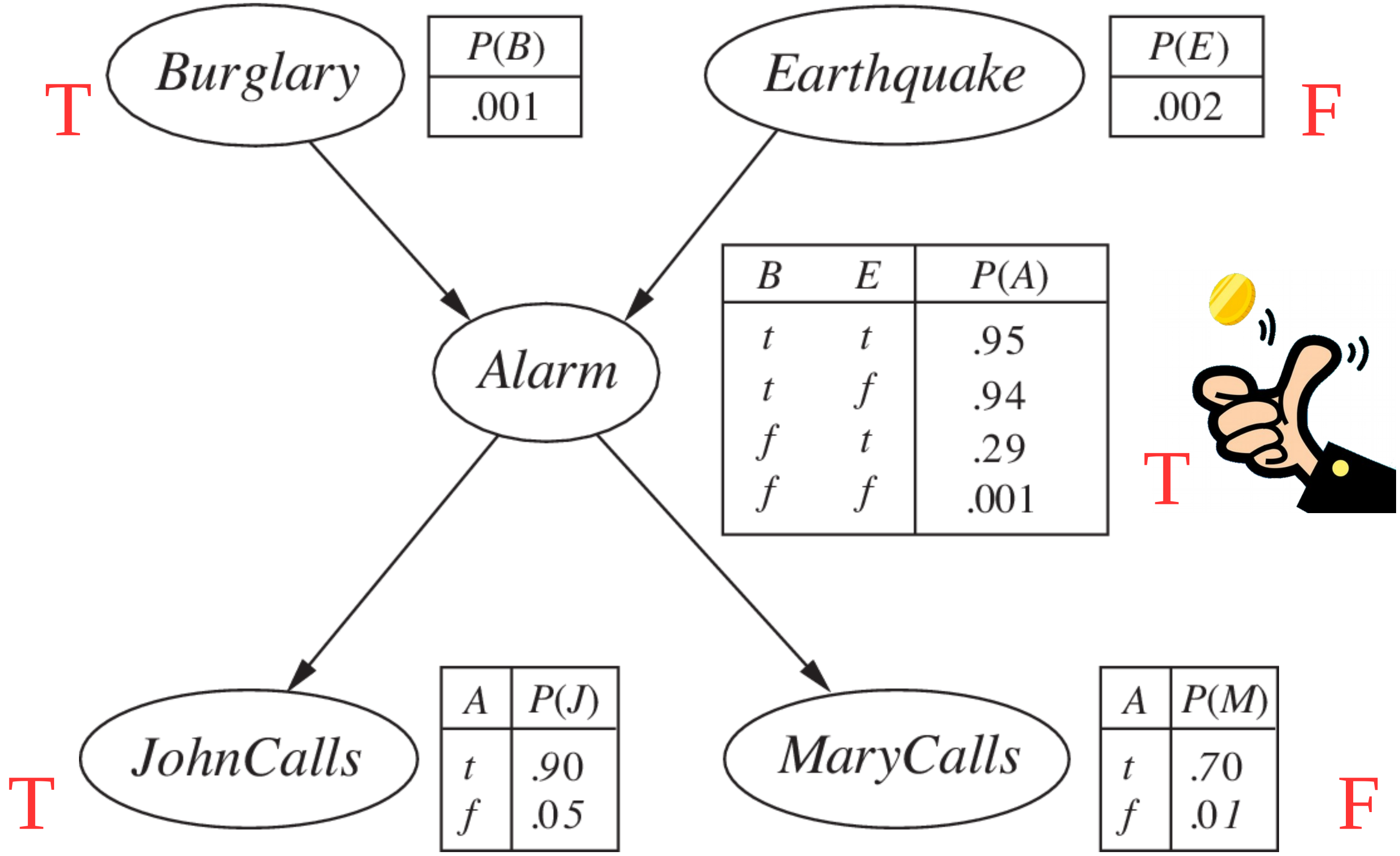
iterative method that randomly tweaks just
one small part of the sample at a given time
(as is done in simulated annealing or genetic algorithms)

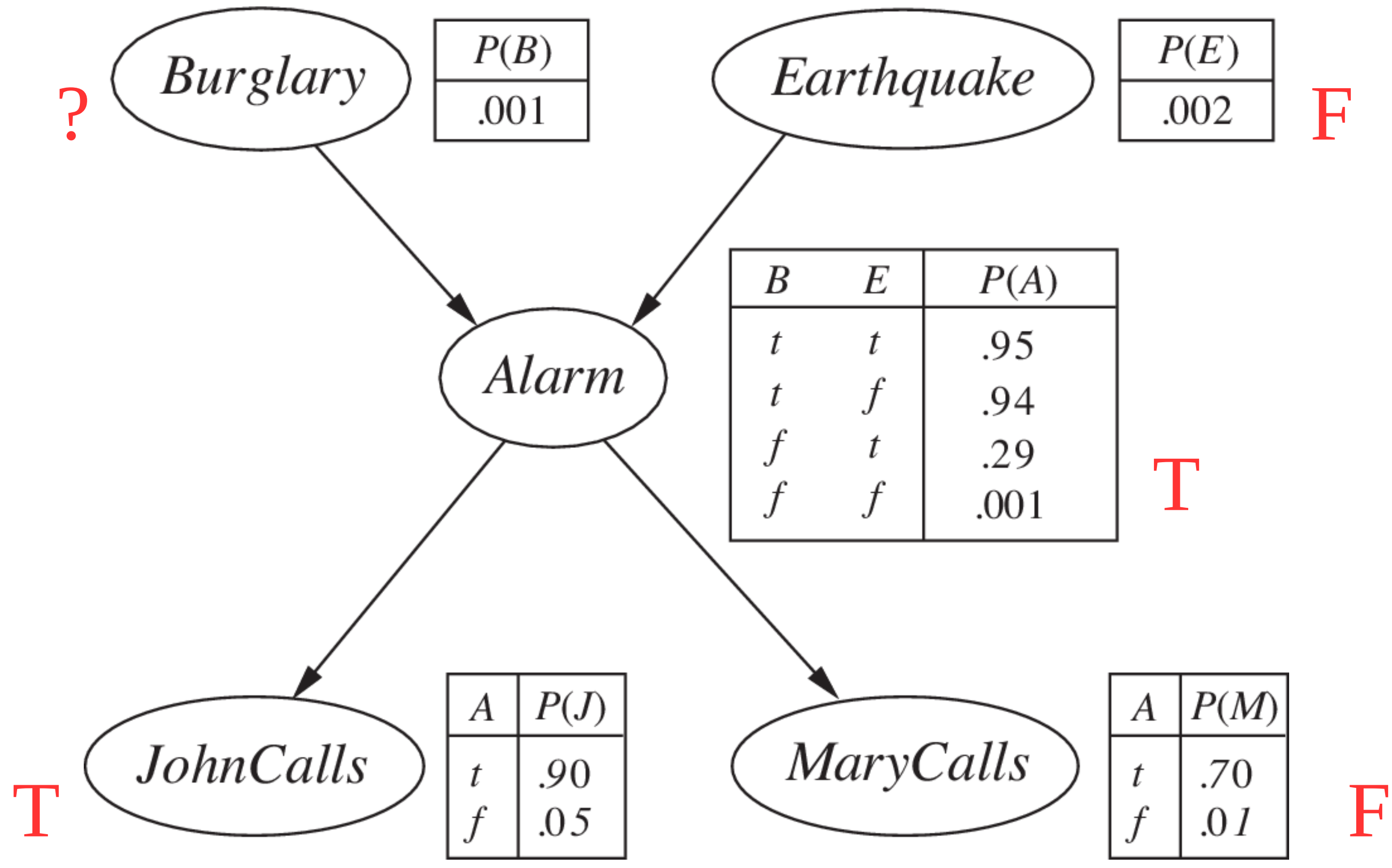


Burglary
P(B)
.001
T
Earthquake
P(E)
.002
F
Alarm
B E P(A)
t t .95
t f .94
f t .29
f f .001
T
JohnCalls
A P(J)
t .90
f .05
T
MaryCalls
A P(M)
t .70
f .01
F

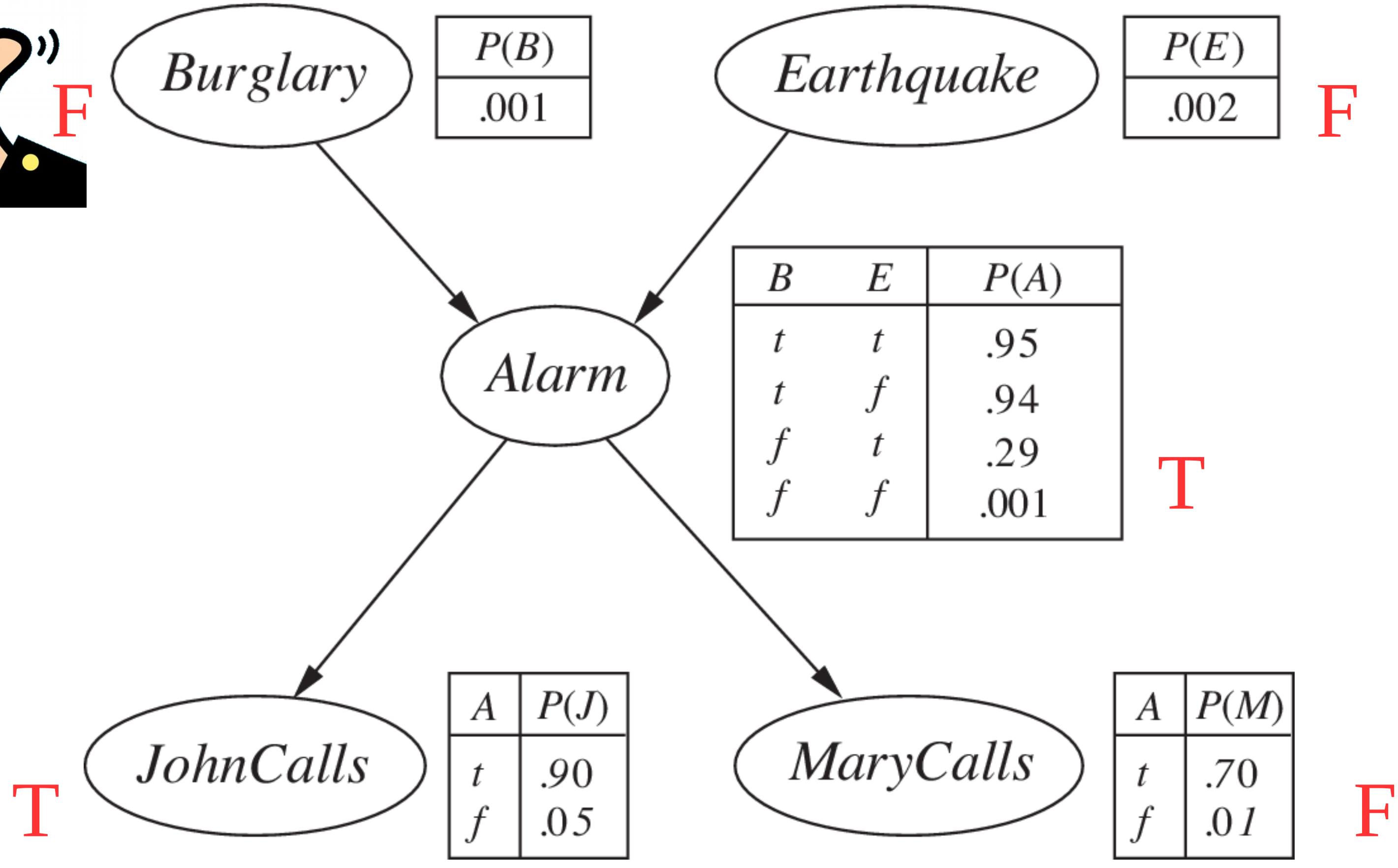


P(Alarm | burglary, ~earthquake, johnCalls, ~maryCalls)

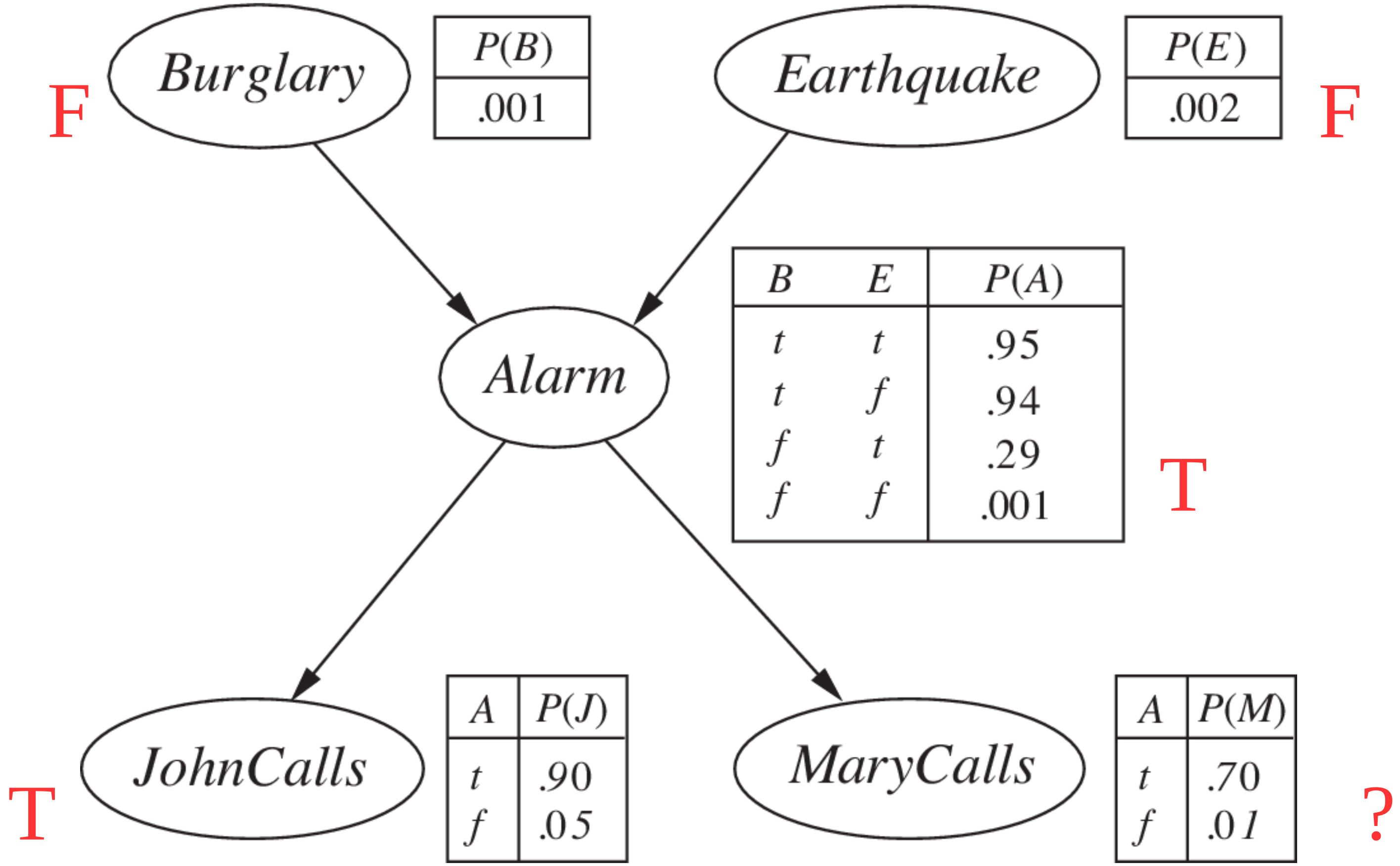

Burglary
P(B)
.001
T
Earthquake
P(E)
.002
F
Alarm
B E P(A)
t t .95
t f .94
f t .29
f f .001
T
JohnCalls
A P(J)
t .90
f .05
T
MaryCalls
A P(M)
t .70
f .01
F



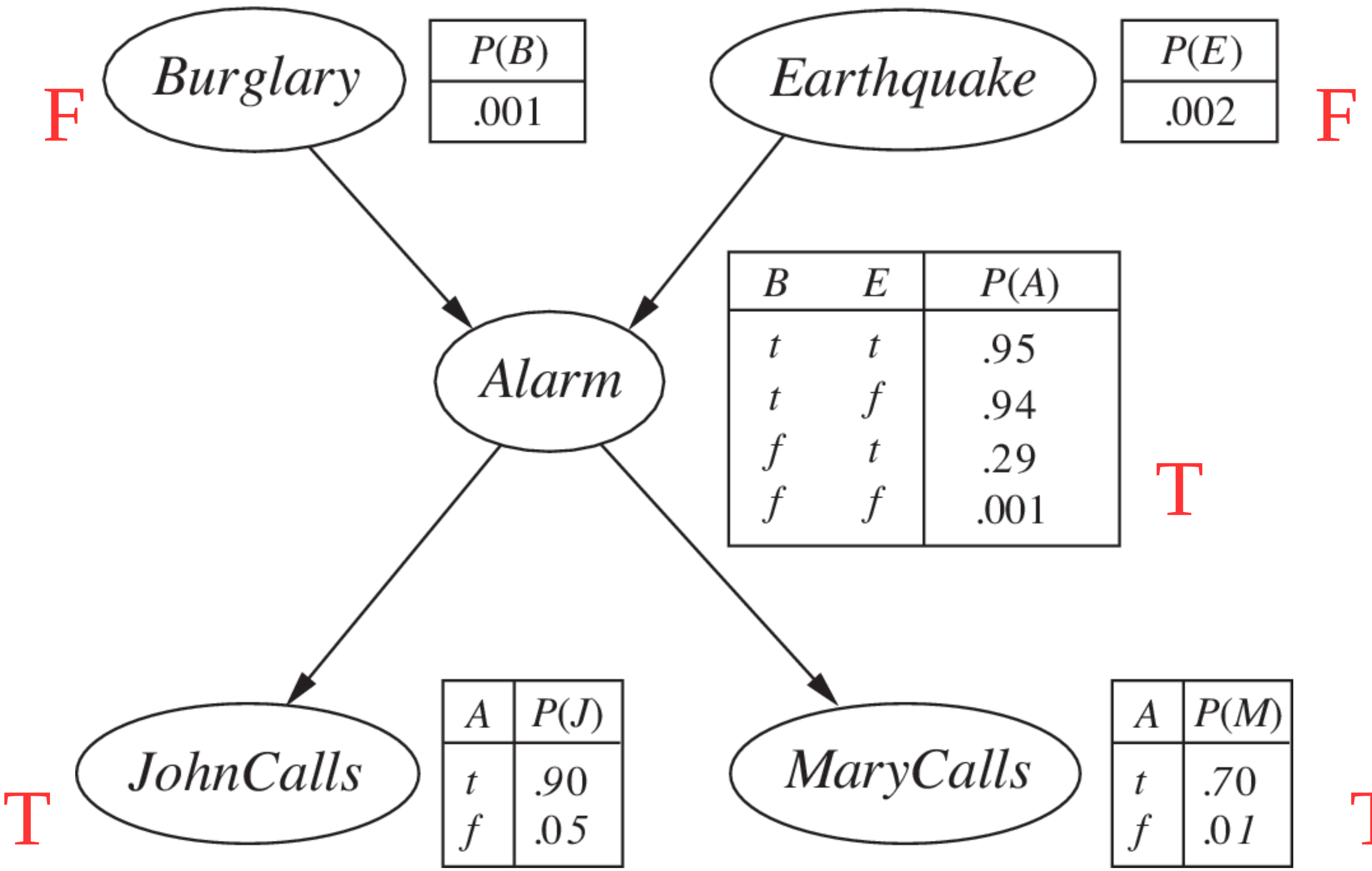
P(Burglary | alarm, ~earthquake, johnCalls, ~maryCalls)



Burglary
P(B)
.001
F
Earthquake
P(E)
.002
F
Alarm
B E P(A)
t t .95
t f .94
f t .29
f f .001
T
JohnCalls
A P(J)
t .90
f .05
T
MaryCalls
A P(M)
t .70
f .01
F



P(MaryCalls | alarm, ~burglary, ~earthquake, johnCalls)



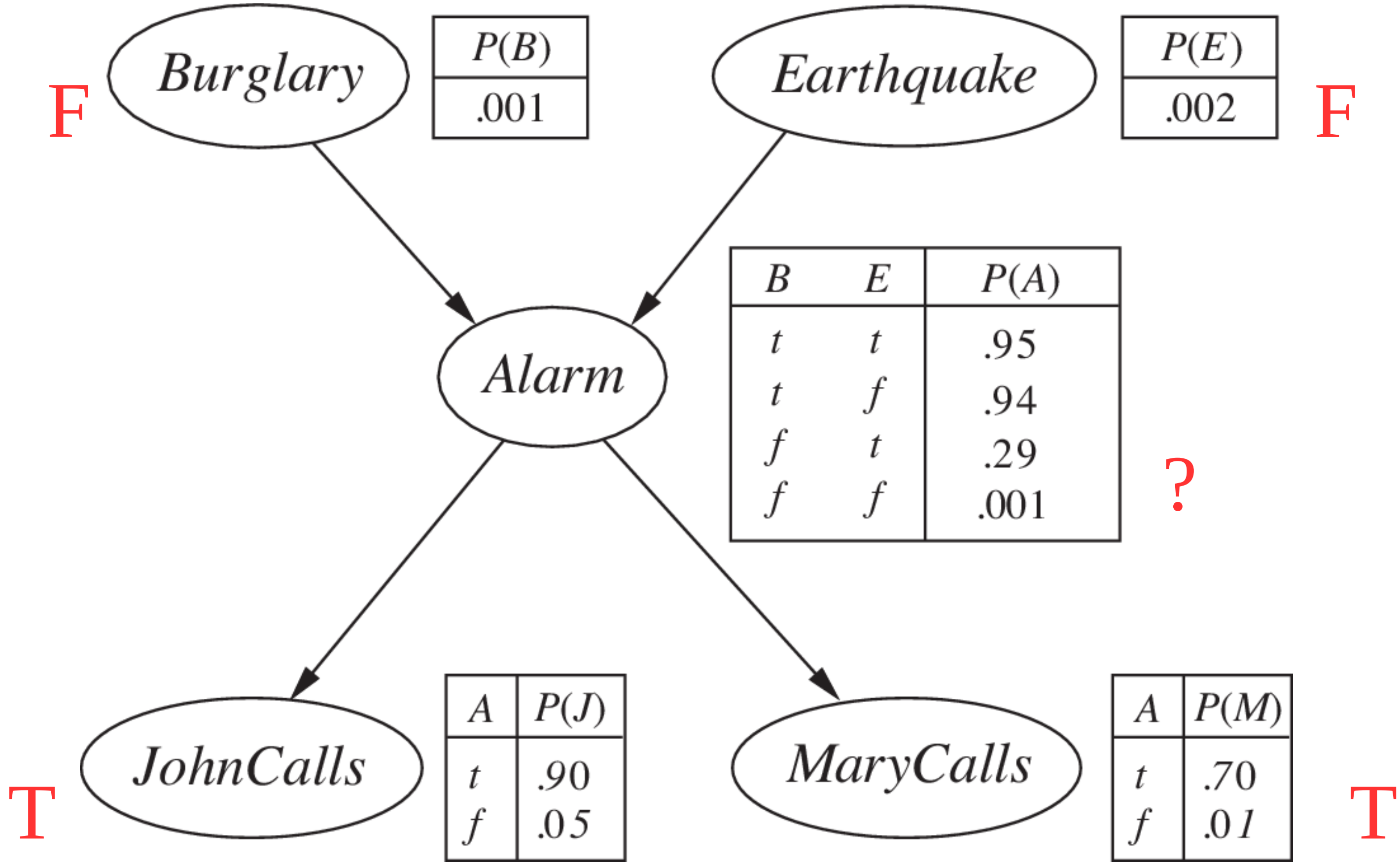
P(B)
.001

P(E)
.002

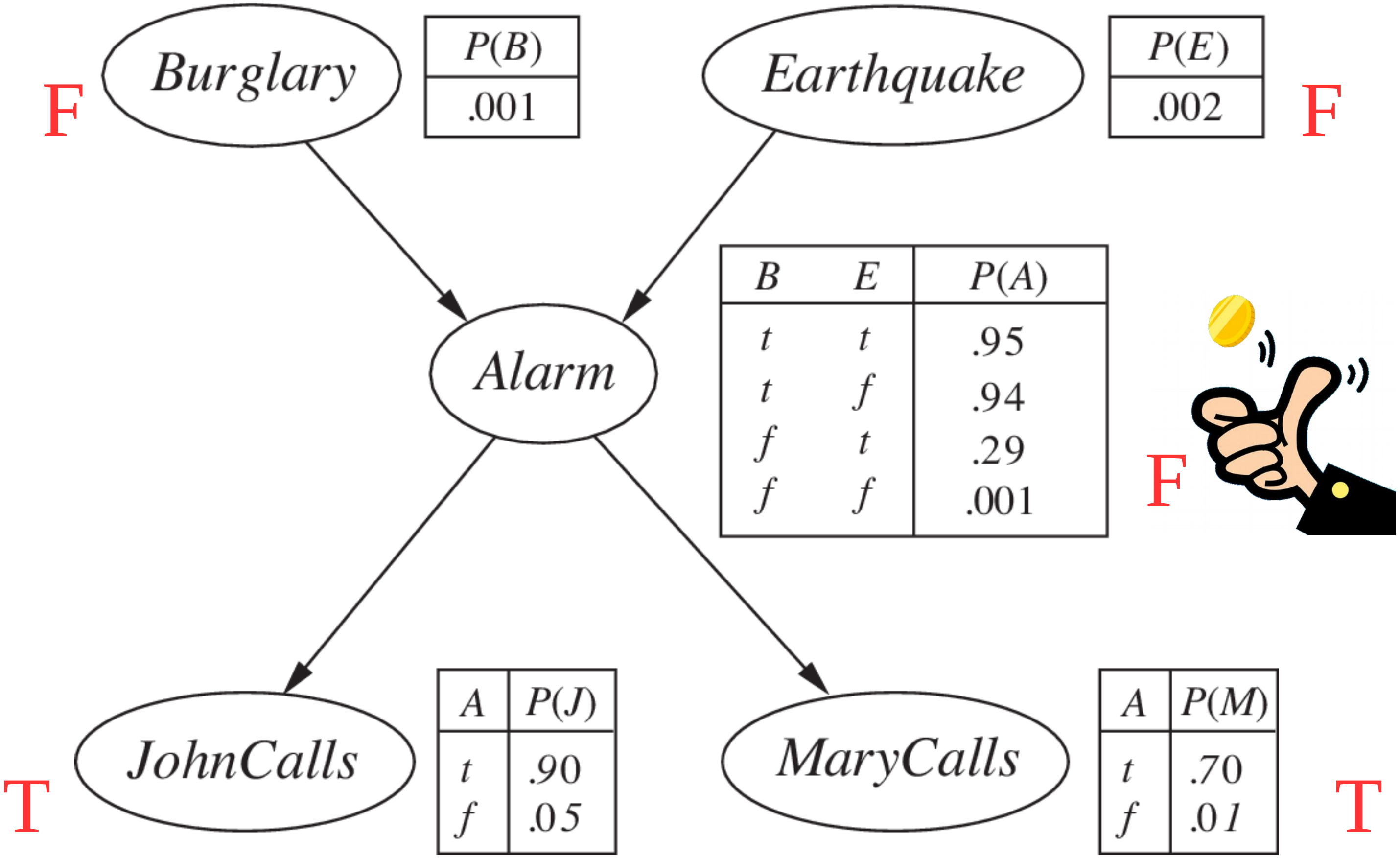
B	E	P(A)
t	t	.95
t	f	.94
f	t	.29
f	f	.001

A	P(J)
t	.90
f	.05

A	P(M)
t	.70
f	.01



P(Alarm | ~burglary, ~earthquake, johnCalls, maryCalls)



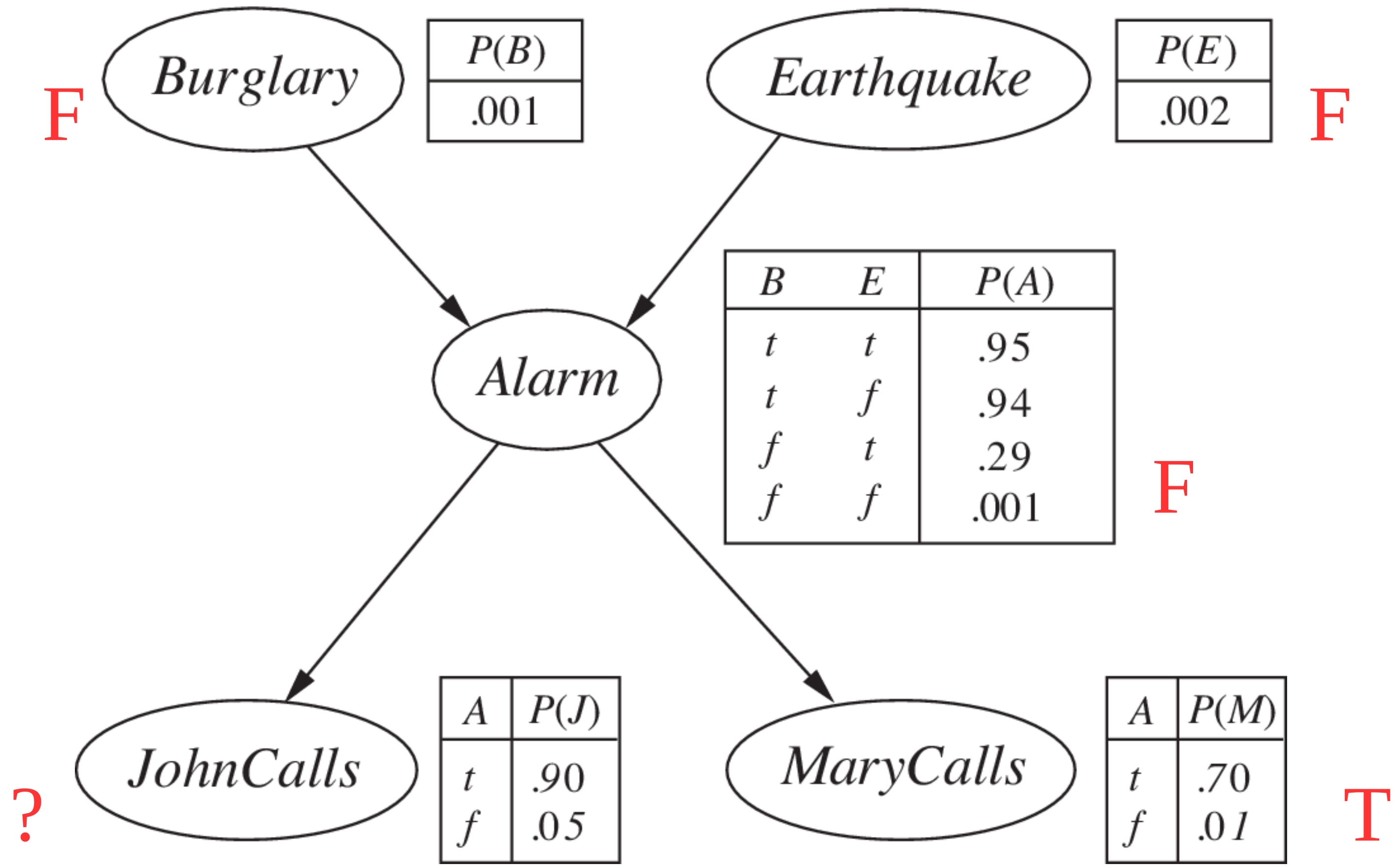
P(B)
.001

P(E)
.002

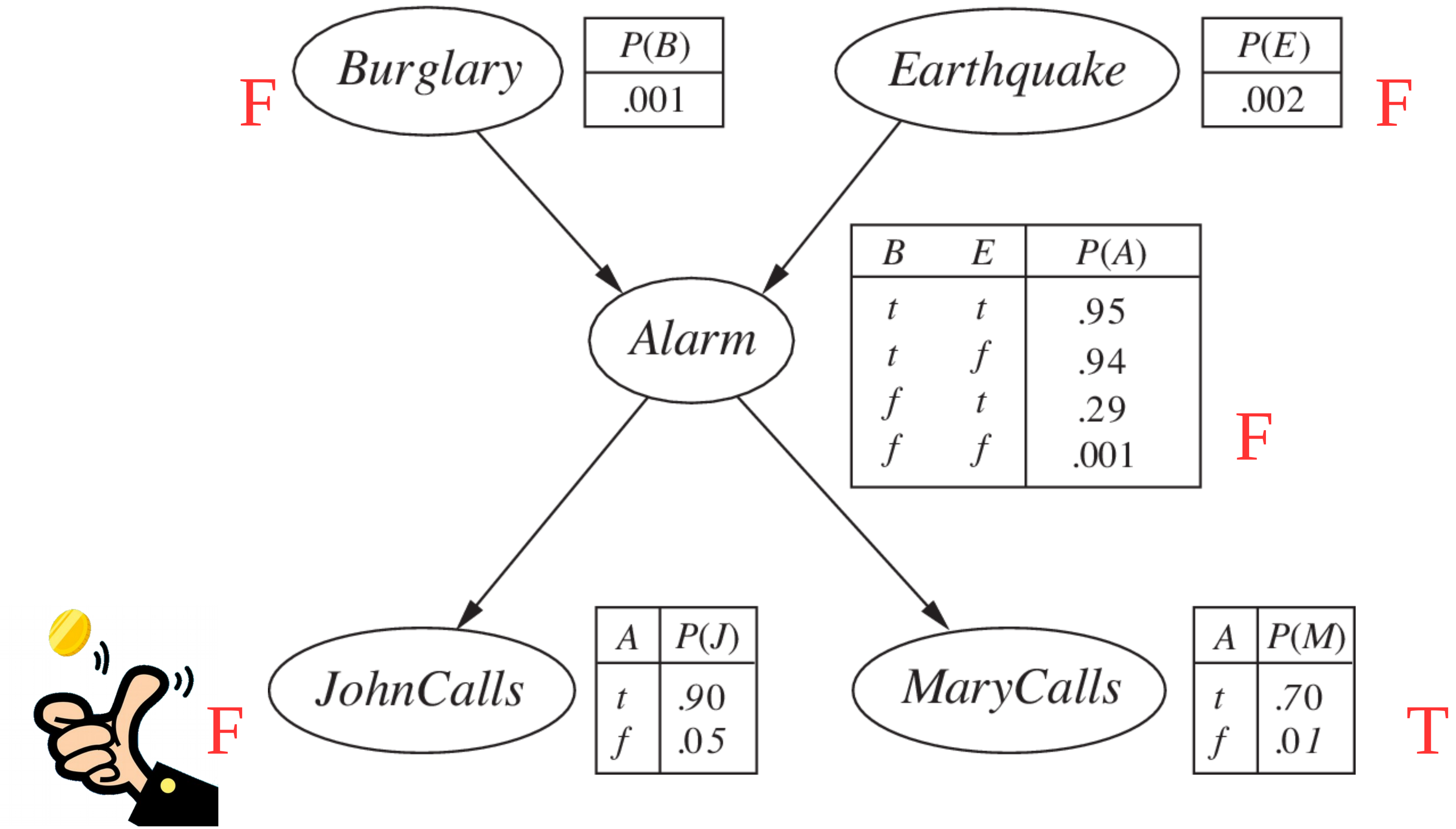
B	E	P(A)
t	t	.95
t	f	.94
f	t	.29
f	f	.001

A	P(J)
t	.90
f	.05

A	P(M)
t	.70
f	.01



P(JohnCalls | ~burglary, ~earthquake, ~alarm, maryCalls)



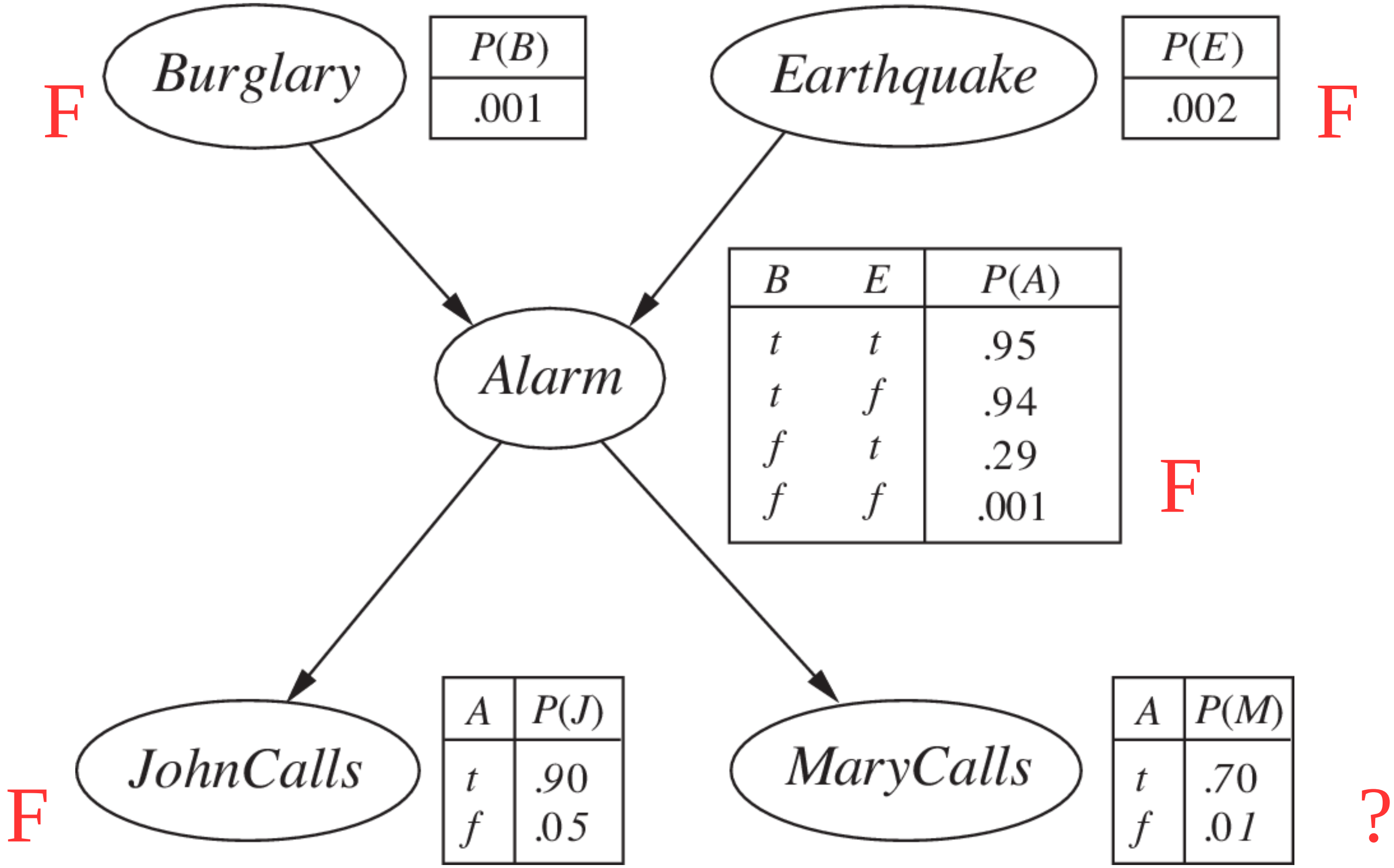
P(B)
.001

P(E)
.002

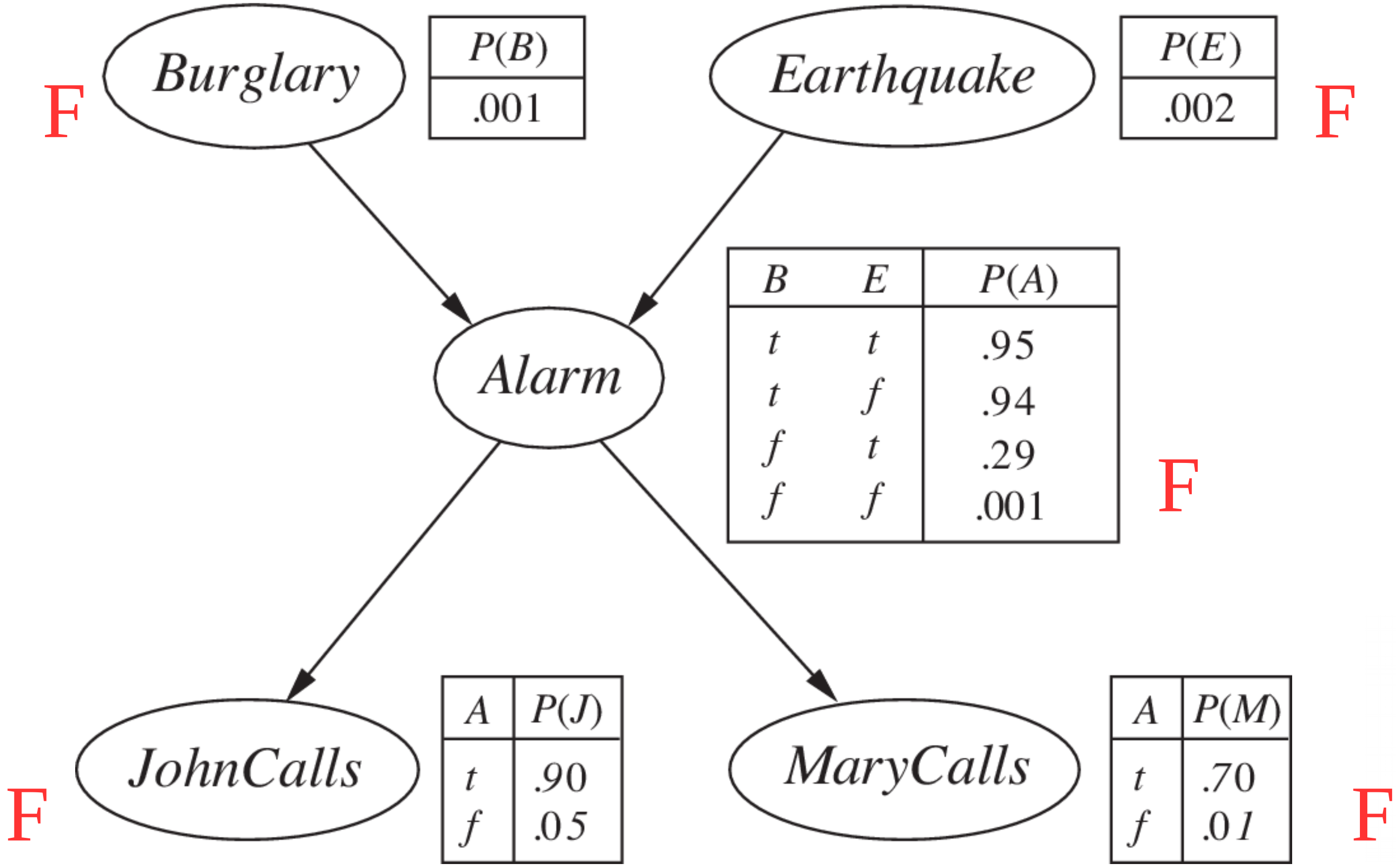
B	E	P(A)
t	t	.95
t	f	.94
f	t	.29
f	f	.001

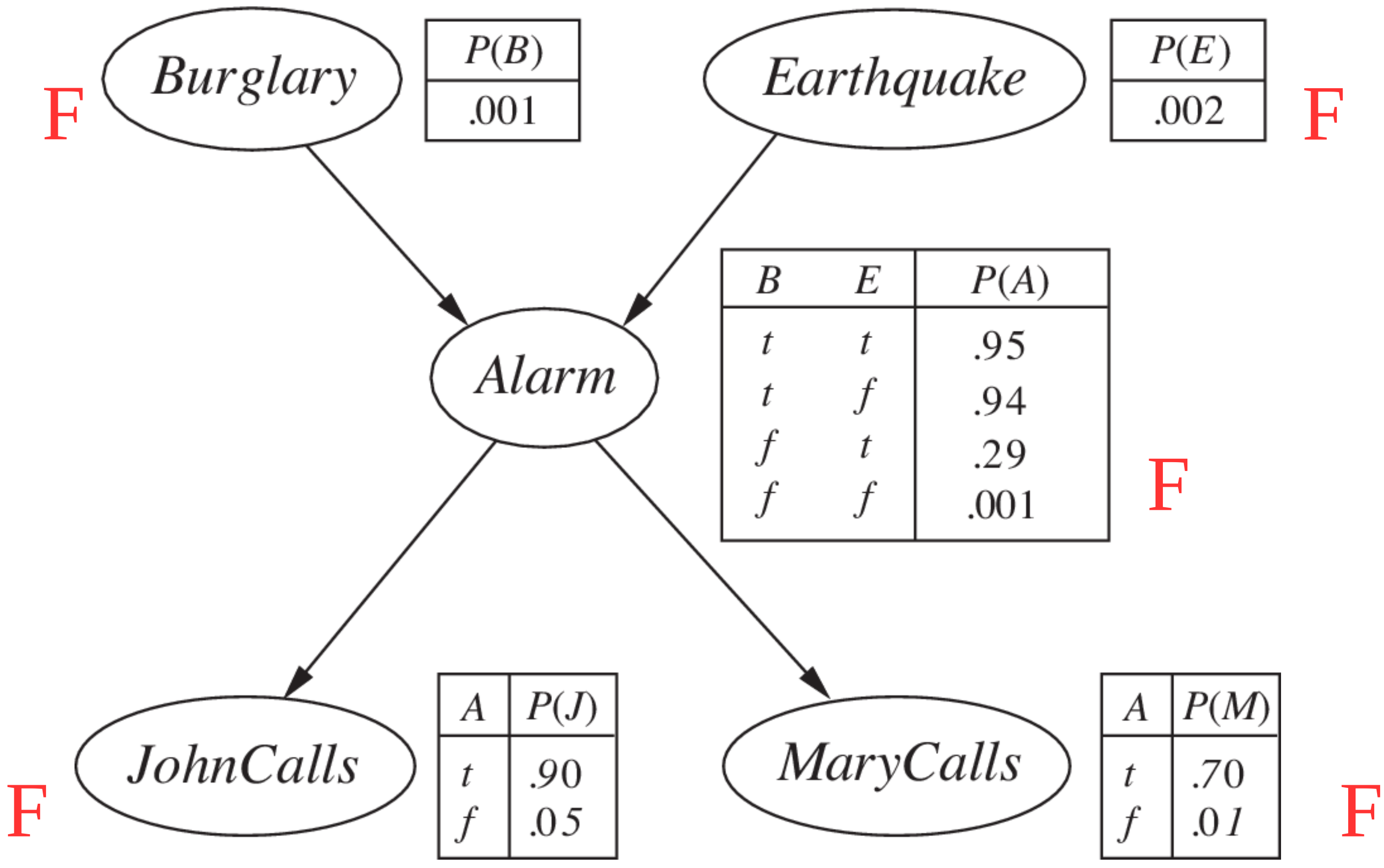
A	P(J)
t	.90
f	.05

A	P(M)
t	.70
f	.01



P(MaryCalls | ~burglary, ~earthquake, ~alarm, ~johnCalls)


Burglary
P(B)
.001
Earthquake
P(E)
.002
Alarm
B E P(A)
t t .95
t f .94
f t .29
f f .001
JohnCalls
A P(J)
t .90
f .05
MaryCalls
A P(M)
t .70
f .01
F
F
F
F
F
F



oh look... the most likely scenario is most likely to come up!

(and other states are as likely as their true posteriors)
(guarantees in your book)

probabilistic reasoning over time

so far our inference methods have assumed that we are continuing to sample from a static distribution (e.g. the chance that an earthquake will happen doesn't change over time)

in many systems, the world is changing around you so past events will influence future outcomes

e.g. in a game of blackjack or poker, you might want to keep track of what cards have already been played, as they change the future probability of what cards might come next

P(dealer has an Ace hidden)

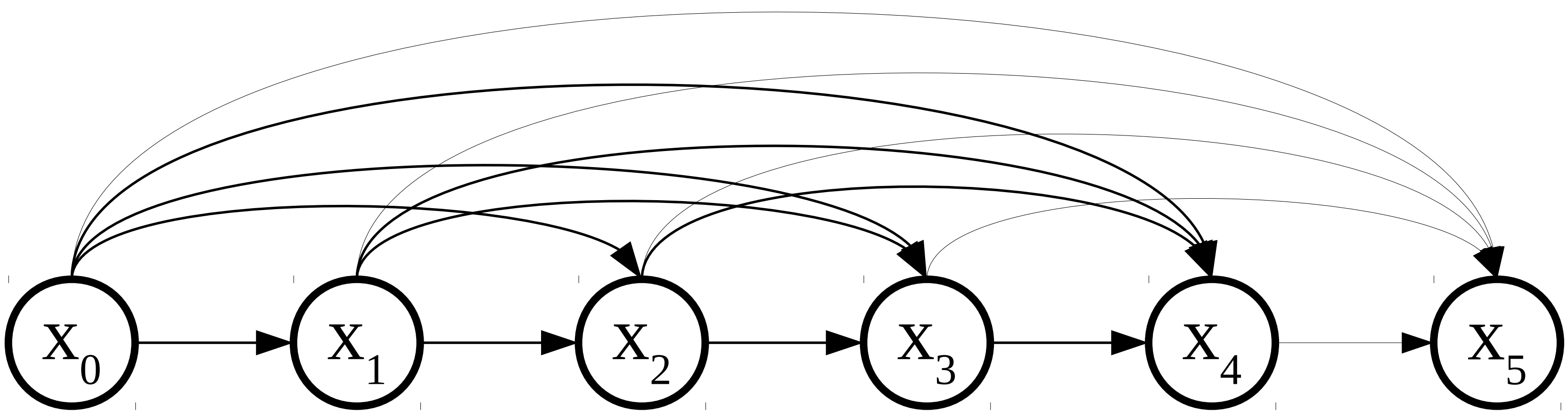
$\neq$

P(dealer has an Ace hidden | previous player was dealt an Ace) to model this, we'll consider the world in discrete snapshots or timesteps

$t = 1, 2, 3, \ldots, N$

$x_{t=0}, x_{t=1}, x_{t=2}, x_{t=3}, \ldots, x_{t=N}$

consider the dependencies of some state x over time...

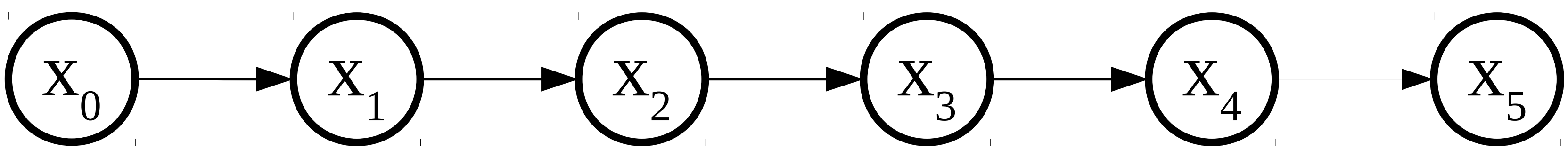


exponential dependencies in time!

instead, let's assume that all the information necessary to know what state X_t will be is contained in state X_{t-1}

(i.e. conditionally independent of $X_{0:t-2}$)

$$P(X_t \mid X_{t-1}, X_{t-2}, X_{t-3}, \ldots, X_1, X_0) = P(X_t \mid X_{t-1})$$



"Markov assumption"

(resulting process is called a "Markov process" or "Markov chain")