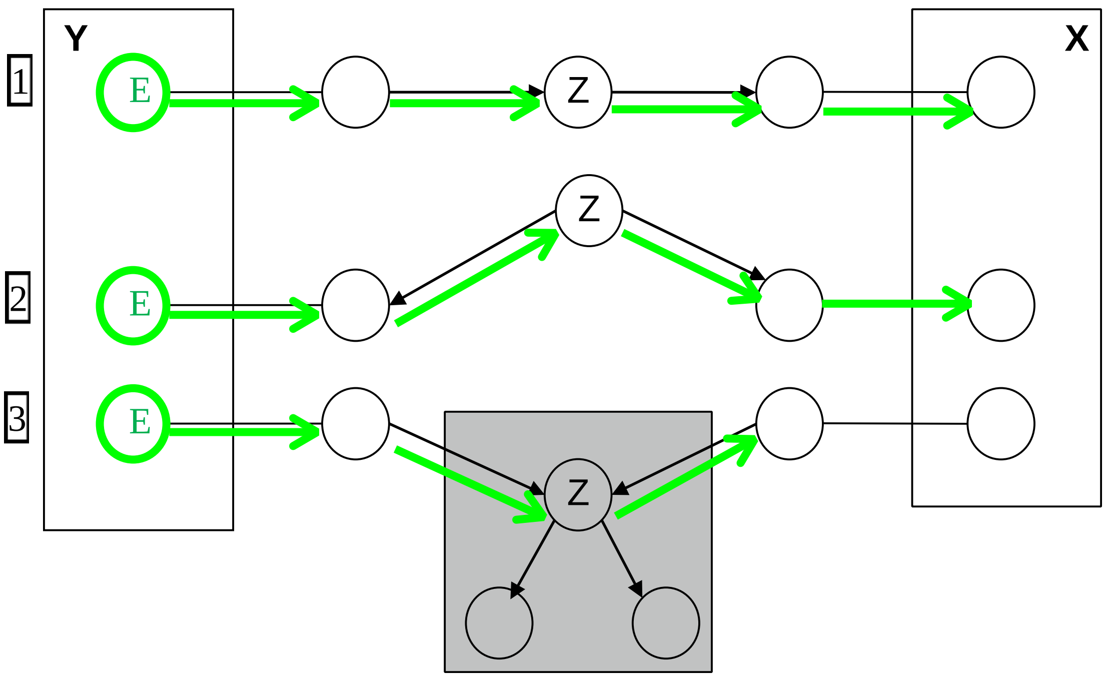
Lecture 22

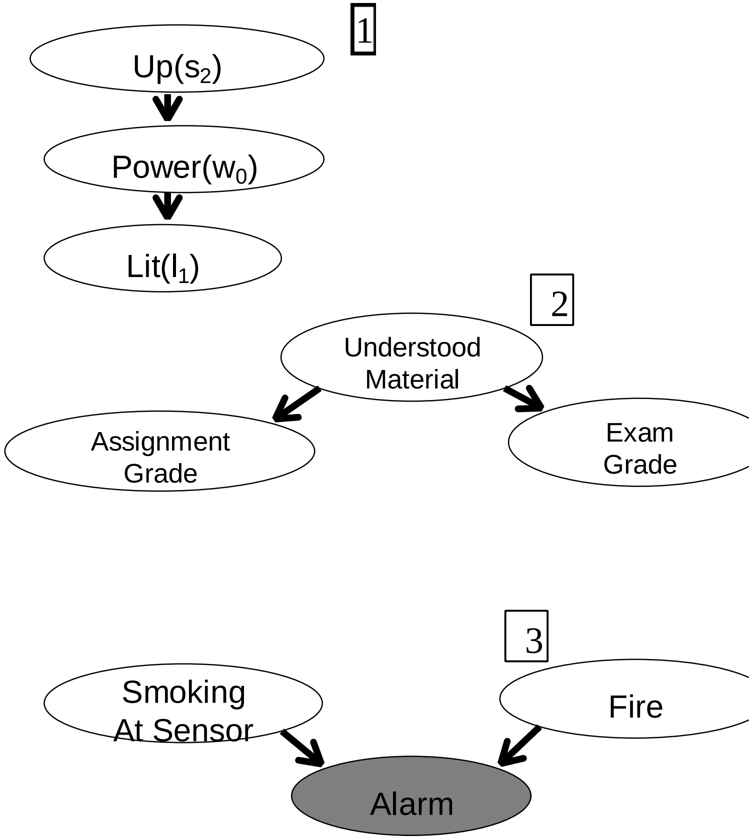
Variable Elimination

Lecture Overview

- Recap
- Variable Elimination
- Algorithm
- VE example

Dependencies in a Bayesian Network

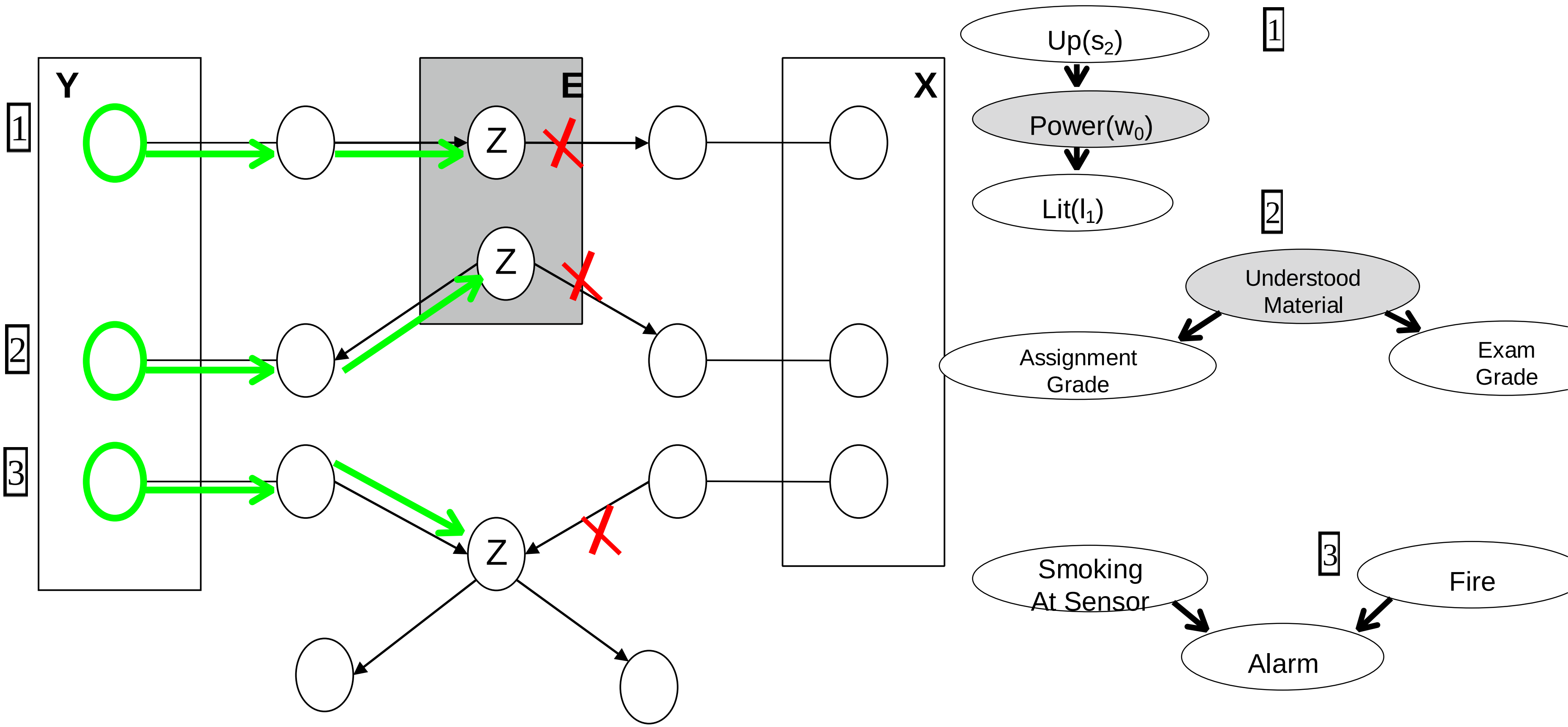




In 1, 2 and 3, X and Y are dependent (grey areas represent existing evidence/observations)

Or Conditional Independencies

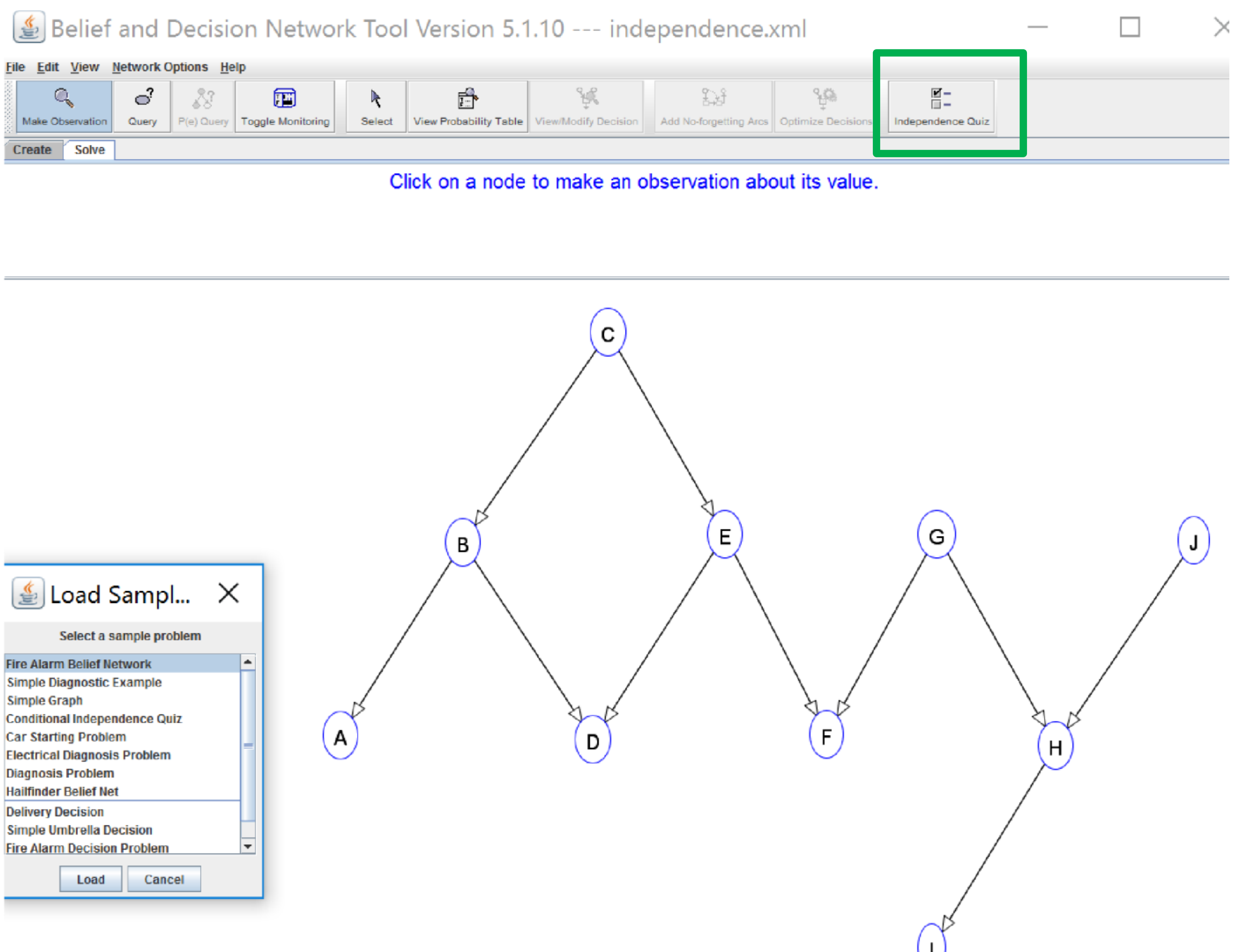
Or, blocking paths for probability propagation. Three ways in which a path



between Y to X (or viceversa) can be blocked, given evidence E

Practice in the AISpace Applet

- Open the Belief and Decision Networks applet



- Load the problem: Conditional Independence Quiz
- Click on Independence Quiz

Independence

What is the minimal set of nodes that must be observed in order to make node X independent from all the non-observed nodes in the network

Independence

What is the minimal set of nodes that must be observed in order to make node X independent from all the non-observed nodes in the network

B - This is the Markov Blanket for node X

Slide 7

Inference Under Uncertainty

- Y: subset of variables that is queried
- E: subset of variables that are observed .
- $Z_1, \ldots, Z_k$ remaining variables in the JPD

Bayesian Networks - AISpace

Try queries with the Belief Networks applet in AISpace,

- Load the Fire Alarm example (ex. 6.10 in textbook)
- Compare the probability of each query node before and after the new evidence is observed
- try first to predict how they should change using your understanding of the domain
- Make sure you explore and understand the various other example networks in textbook and Aispace

Inference Under Uncertainty

Remember our example from Inference by Enumeration?

Windy W	*Cloudy C*	*Temperature T*	*P(W, C, T)*
yes	no	hot	0.04
yes	no	mild	0.09
yes	no	cold	0.07
yes	yes	hot	0.01
yes	yes	mild	0.10
yes	yes	cold	0.12
no	no	hot	0.06
no	no	mild	0.11
no	no	cold	0.03
no	yes	hot	0.04
no	yes	mild	0.25
no	yes	cold	0.08

Find P(T=cold | W=yes)

- Y: subset of variables that is queried => Temperature
- E: subset of variables that are observed => Windy

- $Z_1, \ldots, Z_k$ remaining variables in the JPD => Cloudy

We get the same result if, after removing the rows inconsistent with evidence, we

- first marginalize over Cloudy in the original P(W,C,T), for the entries consistent with Wind = yes. This gives us P(T∧ $WW = yyyyyy$)

Windy W	Cloudy C	Temperature T	P(W,C,T)
yes	no	hot	0.04
yes	no	mild	0.09
yes	no	cold	0.07
yes	yes	hot	0.01
yes	yes	mild	0.10
yes	yes	cold	0.12
~~no~~	~~no~~	~~hot~~	~~0.06~~
~~no~~	~~no~~	~~mild~~	~~0.11~~
~~no~~	~~no~~	~~cold~~	~~0.03~~
~~no~~	~~yes~~	~~hot~~	~~0.04~~
~~no~~	~~yes~~	~~mild~~	~~0.25~~
~~no~~	~~yes~~	~~cold~~	~~0.08~~

$$For\ every\ value\ t\ of\ T, find$$
$$P(T = t, W = yes, C = no) + P(T = t, W = yes, C = yes)$$

Wind W	Temperature T	P(T, W = yes)
yes	hot	0.05
yes	mild	0.19
yes	cold	0.19

This is P(T∧ $W = yes$)

then normalize

$$\frac{P(T \wedge W{=}yes)}{P(W{=}yes \wedge T{=}hot) + P(W{=}yes \wedge T{=}mild) + P(W{=}yes \wedge T{=}co}$$

Temperature T	*P(T| W=yes)*
hot	0.05/0.43 = ~0.12
mild	0.19/0.43= ~0.44
cold	0.19/0.43 = ~0.44

Inference in General

- Y: subset of variables that is queried (e.g. Temperaturein previous example)
- E: subset of variables that are observed . E = e (W = yesin previous example)
- $Z_1, \ldots, Z_k$ remaining variables in the JPD (Cloudyin previous example)

We need to compute this numerator for each value of Y, y_i

We need to marginalize over all the variables $Z_1, \ldots Z_k$ not involved in the query $P(Y$

$$= y_i, E = e) = \sum \ldots \sum P(Z_1, \ldots, Z_k, Y = y_i, E = e)$$

Z_1 Z_k

$P(Y,$

$P(Y \mid E=e) =$ $E=e)$ Def of conditional probability

$P(E=e)$

$P(Y,E=e)$

To compute the denominator, marginalize over Y

$\sum_Y P(Y,E=e)$

constant- Same value for every ensuring that P(Y=$\sum_Y$ y$P(Y$i).

Normalization $= y_i \mid E) = 1$

- All we need to compute is the numerator: joint probability of the query variable(s) and the evidence!

- Variable Elimination is an algorithm that efficiently performs this operation by casting it as operations between factors - introduced next

Factors

- A factor is a function from a tuple of random variables to the real numbers R

- $P(X_1, X_2)$ is a factor $f(X_1, X_2)$

Distribution

- $P(Z \mid X,Y)$ is a factor $f(Z,X,Y)$

Set of Distributions

One for each combination of values for X and Y

- $P(Z=f|X,Y)$ is a factor $f(X,Y)$

Set of partial Distributions

$f(X, Y)_{Z = f}$

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

- Note: Factors do not have to sum to one

- We write a factor on variables $X_1, \ldots, X_j$ as $f(X_1, \ldots, X_j)$
- A factor denotes one or more (possibly partial) distributions over the given tuple of variables, e.g.,

Operation 1: assigning a variable

- We can make new factors out of an existing factor

- Our first operation: we can assign some or all of the variables of a factor.

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4

f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

f(X,Y,Z):

Factor of Y,Z

What is the result of assigning X= t ?

$f(X=t,Y,Z) = f(X, Y, Z)_{X=t}$

Y	Z	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

Operation 3: multiplying factors

The product of factors $f_1(A, B)$ and $f_2(B, C)$, where B is the variable (or set of variables) in common, is the factor $(f_1 \times f_2)(A, B, C)$ defined by:

$f_1(A,B)$:

A	B	Val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

$f_2(B,C)$:

B	C	Val
t	t	0.3
t	f	0.7
f	t	0.6
f	f	0.4

$f_1(A,B) \times f_2(B,C)$:

A	B	C	val
t	t	t	0.03
t	t	f	0.1x0.7
t	f	t	0.9x0.6
t	f	f	…
f	t	t	
f	t	f	
f	f	t	
f	f	f	

The domain of f1 × f2 is the union of the domains of f1 and f2

$$(f_1 \times f_2)(A,B,C) = f_1(A,B) \times f_2(B,C)$$

Recap

If we assign variable A=a in factor f(A,B), what is the correct form for the resulting factor?

- f(B).
  When we assign variable A we remove it from the factor's domain

If we marginalize variable A out from factor f(A,B), what is the correct form for the resulting factor?

- f(B).
  When we marginalize out variable A we remove it from the factor's domain

If we multiply factors $f_4(X,Y)$ and $f_6(Z,Y)$, what is the correct form for the resulting factor?

- f(X,Y,Z)
- When multiplying factors, the resulting factor's domain is the union of the multiplicands' domains

Lecture Overview

- Recap
- Variable Elimination
- Algorithm
    - VE example

Inference in General

- Y: subset of variables that is queried
- E: subset of variables that are observed . E = e
- Z_1, …,Z_k remaining variables in the JPD

We need to compute this numerator for each value of Y, y_i

We need to marginalize over all the variables Z_1,…Z_k not involved in the query $P(Y = y_i, E = e) = \sum_{Z_1} \ldots \sum_{Z_k} P(Z_1, \ldots, Z_k, Y = y_i, E = e)$

$$P(Y \mid E=e) = P(Y, E=e)$$ Def of conditional probability

$$\frac{P(Y, E=e)}{P(E=e)}$$

$$\sum_Y P(Y, E=e)$$

To compute the denominator, marginalize over Y
- Same value for every $P(Y=y_i)$. Normalization constant ensuring that $\sum_Y P(Y=y_i \mid E)=1$

- All we need to compute is the numerator: joint probability of the query variable(s) and the evidence!
- Variable Elimination is an algorithm that efficiently performs this operation by casting it as operations between factors

Variable Elimination: Intro (1)

- We can express the joint probability as a factor

observed Other variables not involved in the query

- $f(Y, E_1 \ldots, E_j, Z_1 \ldots, Z_k)$

- We can compute $P(Y, E_1{=}e_1, \ldots, E_j{=}e_j)$ by

- Assigning $E_1{=}e_1, \ldots, E_j{=}e_j$

- Marginalizing out variables $Z_1, \ldots, Z_k$, one at a time
    - ✓the order in which we do this is called our elimination ordering

$$P(Y, E_1 = e_1, \ldots, E_j = e_j) = \sum_{Z_k} \cdots \sum_{Z_1} f(Y, E_1, .., E_j, Z_1, .., Z_k)_{E_1 = e_1, \ldots, E_j = e_j}$$

- Are we done?

No, this still represents the whole JPD (as a single factor)!
Need to exploit the compactness of Bayesian networks

Variable Elimination Intro (2)

$$P(Y, E_1 = e_1, \ldots, E_j = e_j) = \sum_{Z_k} \cdots \sum_{Z_1} f(Y, E_1, .., E_j, Z_1, .., Z_k)_{E_1 = e_1, \ldots, E_j = e_j}$$

Recall the JPD of a Bayesian network

$$\mathrm{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid X_1, \ldots, X_{i-1}) = \prod_{i=1}^{n} P(X_i \mid pa(X_i))$$

We can express the joint factor as a product of factors, one for each conditional probability

$$P(X_i \mid pa(X_i)) = f(X_i, pa(X_i)) = f_i$$

$$P(Y, E_1 = e_1, \ldots, E_j = e_j) = \sum_{Z_k} \cdots \sum_{Z_1} f(Y, E_1, .., E_j, Z_1, .., Z_k)_{E_1=e_1,\ldots,E_j=e_j}$$

$$= \sum_{Z_k} \cdots \sum_{Z_1} \prod_{i=1}^{n} (f_i)_{E_1=e_1,\ldots,E_j=e_j}$$

Computing sums of products

Inference in Bayesian networks thus reduces to computing the sums of products

$$\sum \cdots \sum \prod_{i=1}^{n} (f_i)_{E_1=e_1,\ldots,E_j=e_j}$$

Z_k Z

To compute efficiently n

$$\sum_{Z_k} \prod_{i=1} f_i$$

- Factor out those terms that don't involve Z_k, e.g.:

$$\sum_A \boxed{f(C,D)} \times f(A,B,D) \times f(E,A) \times \boxed{f(D)}$$

$$\boxed{f(C,D) \times f(D)} \sum_A f(A,B,D) \times f(E,A)$$

$$f(C,D) \times f(D) \times f'(B,D,E)$$

Example

A	B	Val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

Imagine we want $\sum_A f_1(A,B) \times f_2(B,C)$

If we do the product as is, and then sum out A,

Example

Imagine we want $\sum_A f_1(A,B) \times f_2(B,C)$

B	C	Val
t	t	0.3
t	f	0.7
f	t	0.6
f	f	0.4

A	B	C	val
t	t	t	0.03
t	t	f	0.07
t	f	t	0.54
t	f	f	0.36
f	t	t	0.06
f	t	f	0.14
f	f	t	0.48
f	f	f	0.32

B	C	Val
t	t	0.09
t	f	0.21
f	t	1.02
f	f	0.68

we need 8 multiplications for the factors product, and 4 sums to marginalize out A

If we partition the product as $f_2(B,C)\ \sum_A f_1(A,B)$

A	B	Val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

$\sum_A f_1(A,B)$ →

B	Val
t	0.3
f	1.7

B	C	Val
t	t	0.3
t	f	0.7
f	t	0.6
f	f	0.4

$f_2(B,C)f_3(B)$ →

B	C	Val
t	t	0.09
t	f	0.21
f	t	1.12
f	f	0.68

we need 2 sums to marginalize out A, and 4 products

Analogy with “Computing sums of products”

This simplification is similar to what you can do in basic algebra with multiplication and addition

Example: it takes 14 multiplications or additions to evaluate the expression ab + ac + ad + aeh + afh + agh.

How can this expression be evaluated efficiently?

- Factor out the a and then the h giving a(b + c + d + h(e + f + g))
- This takes only 7 operations

Summing out a variable efficiently

To sum out a variable Z from a product $f_1 \times \ldots \times f_k$ of factors

- Partition the factors into
  - ✓Those that do not contain Z, say $f_1 ,.., f_i$
  - ✓Those that contain Z, say $f_{i+1} ,..., f_k$

- Rewrite

$$\sum_{Z} f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_i \times \left(\sum_{Z} f_{i+1} \times \cdots \times f_k \right)$$

- We thus have

New factor *f'* obtained by

$$\sum_{Z} f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_i \times f'$$

then summing out Z multiplying

f_{i+1} ,..,f_k and Z

- Now we have summed out Z

Z_k

$$= \sum \cdots \sum$$

Simplify Sum of Product: General Case

Factors that do not contain Z_1 | Factors that contain Z_1

$$\sum_{Z_k}\cdots\sum_{Z_1} f_1\times\cdots\times f_h=\sum_{Z_k}\cdots\sum_{Z_2}(f_1\times\cdots\times f_i)\left[\sum_{Z_1} f_{Z_1 1}\times\cdots\times f_{Z_1 k}\right]$$

$$=\sum\cdots\sum_{Z_2} f_1\times\cdots\times f_i\times f'$$

Factors that contain Z_2

$$(f_m\times\cdots\times f_j)\sum_{Z_2}(f_{Z_2 1}\times\cdots\times f_{Z_2 k})$$

Factors that do not contain Z_2

$$=\sum_{Z_k}\cdots\sum_{Z_3} f_m\times\cdots\times f_j\times f''$$

Z_k Z_3

Etc., continue given a predefined simplification ordering of the variables: variable elimination ordering

The variable elimination algorithm,

To compute $P(Y=y_i | E = e)$

1. Construct a factor for each conditional probability.
2. For each factor, assign the observed variables E to their observed values.
3. Given an elimination ordering, decompose sum of products
4. Sum out all variables Z_i not involved in the query

5 Multiply the remaining factors, which only involve

.

A. Y

B. Z

C. E

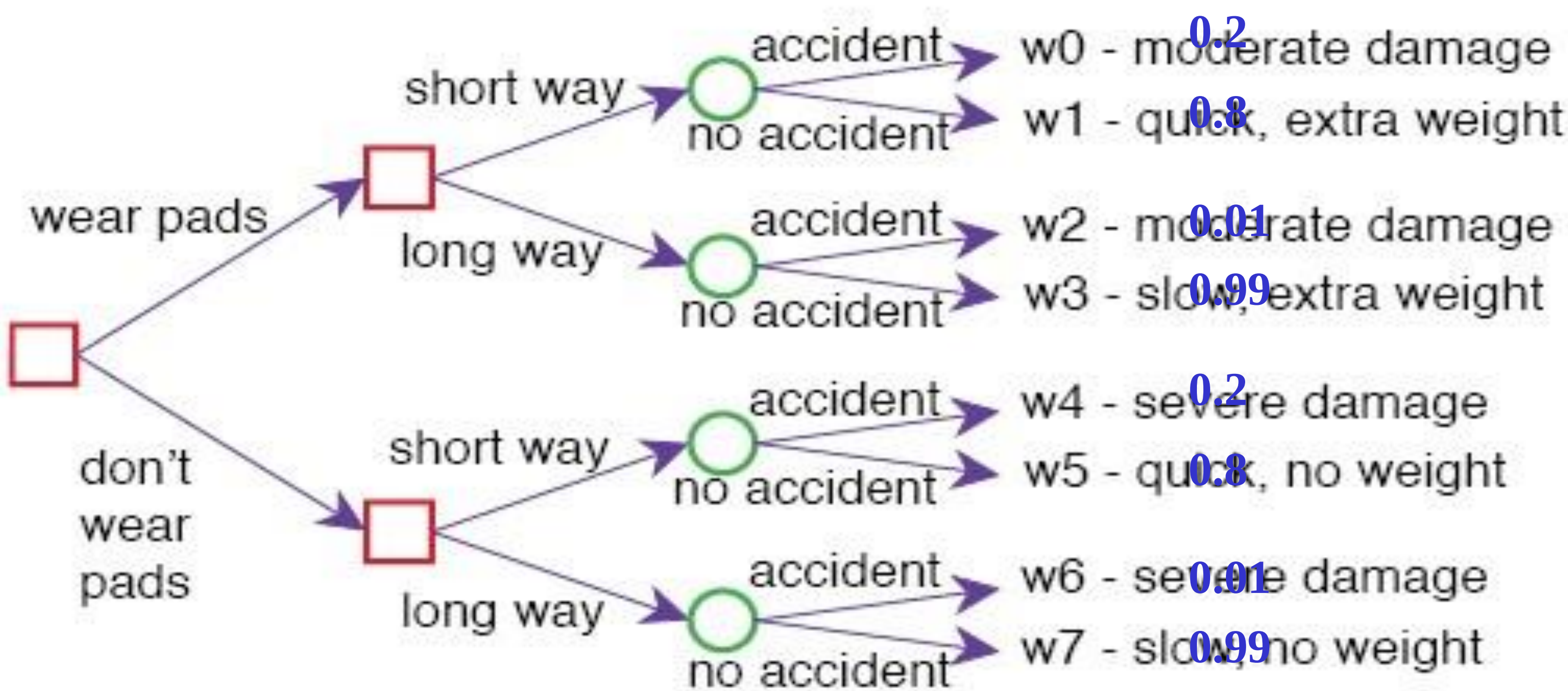
Utility

- Utility: a measure of desirability of possible worlds to an agent
- Let U be a real-valued function such that U(w) represents an agent's degree of preference for world w
- Expressed by a number in [0,100]

Utility for the Robot Example

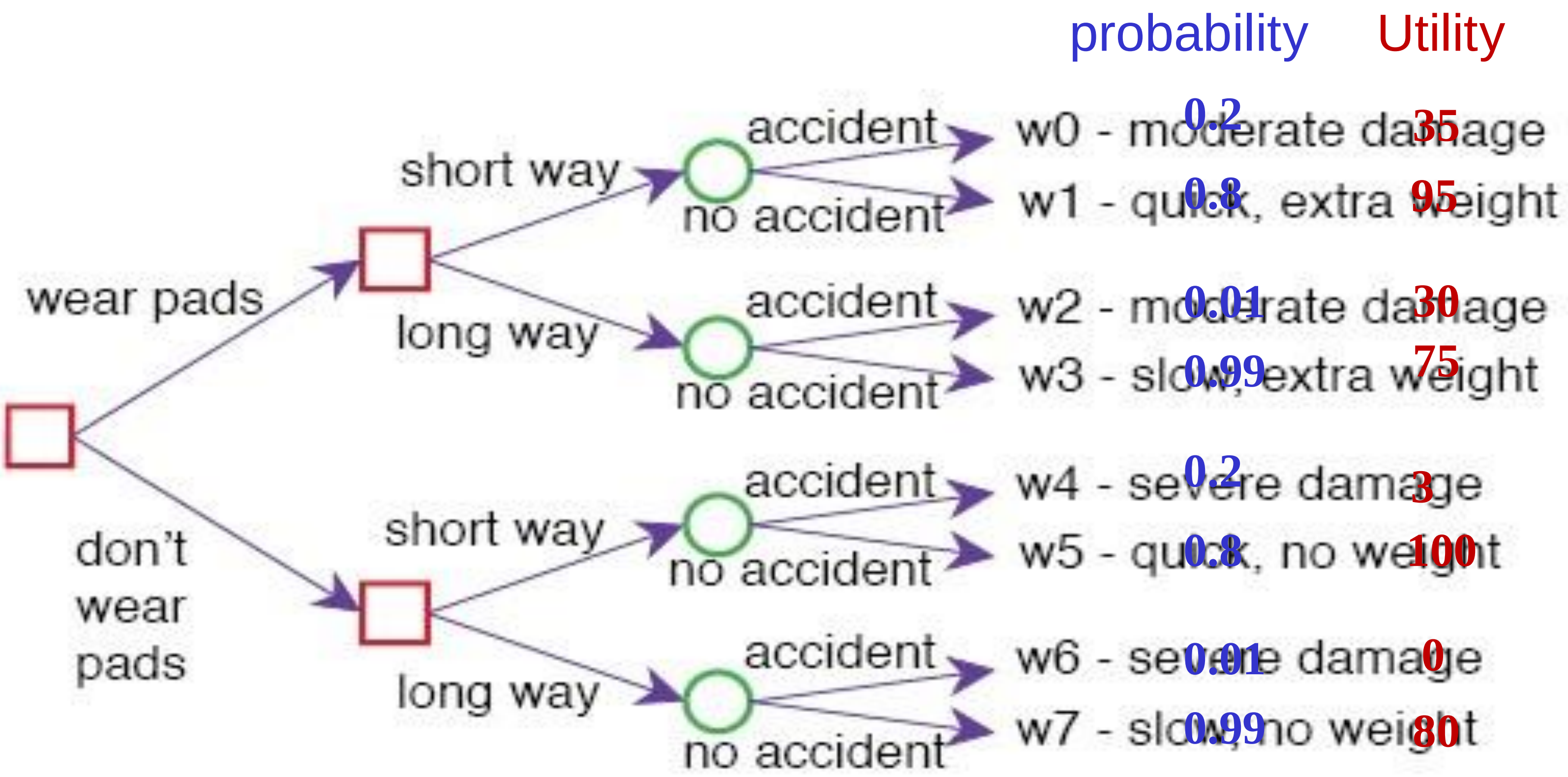
- Which would be a reasonable utility function for our robot?
- Which are the best and worst scenarios? probability

Utility



Utility for the Robot Example

- Which would be a reasonable utility function for our robot?



probability
Utility
wear pads
don't wear pads
short way
long way
short way
long way
accident
no accident
accident
no accident
accident
no accident
accident
no accident
w0 - moderate damage
0.2
35
w1 - quick, extra weight
0.8
95
w2 - moderate damage
0.01
30
w3 - slow, extra weight
0.99
75
w4 - severe damage
0.2
3
w5 - quick, no weight
0.8
100
w6 - severe damage
0.01
0
w7 - slow, no weight
0.99
80

Utility: Simple Goals

A.

Which way	Accident	Wear Pads	Utility
long	true	true	0
long	true	false	0 0
long	false	true	0
long	false	false	0
short	true	true	0
short	true	false	0
short	false	true	100
short	false	false	90

- How can the simple (boolean) goal "reach the room" be specified? **B.**

Which way	Accident	Wear Pads	Utility

Which way Accident Wear Pads Utility long

true true 0 long true

false 0 long false true 0 long

false false 0 short true true

0 short true false 0 short

false true 100 short false

false 0

long	true	true	0
long	true	false	0
long	false	true	100
long	false	false	100
short	true	true	0
short	true	false	0
short	false	true	100
short	false	false	100

C .

D. Not possible

Utility

- Utility: a measure of desirability of possible worlds to an agent • Let U be a real-valued function such that U(w) represents an agent's degree of preference for world w

- Expressed by a number in [0,100]

- Simple goals can still be specified
- Worlds that satisfy the goal have utility 100 • Other worlds have utility 0

Optimal decisions: combining Utility and Probability

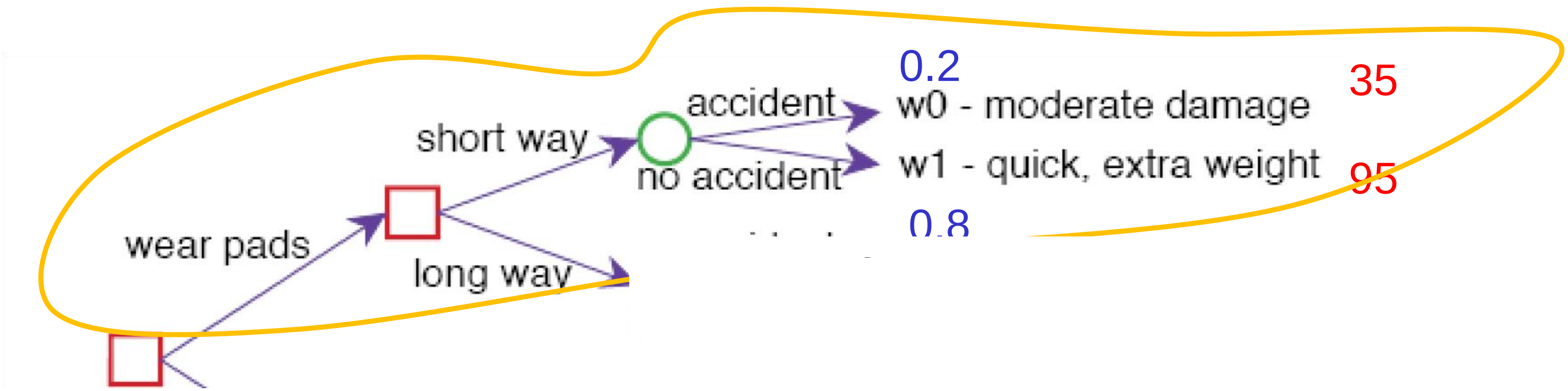
Which way Pads	Accident	Wear	Utility
long	true	true	0
long	true	false	0
long	false	true	100
long	false	false	100
short	true	true	0
short	true	false	0
short	false	true	100
short	false	false	100

e.g., goal "reach the room"

- Each set of decisions defines a probability distribution over possible outcomes
- Each outcome has a utility
- For each set of decisions, we need to know their expected utility

- the value for the agent of achieving a certain probability distribution over outcomes (possible worlds)

- The expected utility of a set of decisions is obtained by
- weighting the utility of the relevant possible worlds by their probability.



- We want to find the decision with maximum expected utility

Expected utility of a decision

- The expected utility of a specific decision $D = d$ is indicated as $E(U|\ D = d)$ and it is computed as follows

$$P(w_1)\times U(w_1) + P(w_2)\times U(w_2) + P(w_3)\times U(w_3) + \ldots + P(w_n)\times U(w_n)$$

Where

- $w_1, w_2, w_3 \ldots, w_n$ are all the possible worlds in which d is true
- $P(w_1), P(w_2), \ldots P(w_n)$ are their probabilities
- $U(w_1), U(w_2), \ldots U(w_n)$ are the corresponding utilities for $w_1, \ldots, w_n$

That is,

- for each possible world w_i in which the decision d is true, multiply the probability of that world and its utility $P(w_i) \times U(w_i)$

- Sum all these products together

This notation indicates all the possible worlds in which dis true

In a formula

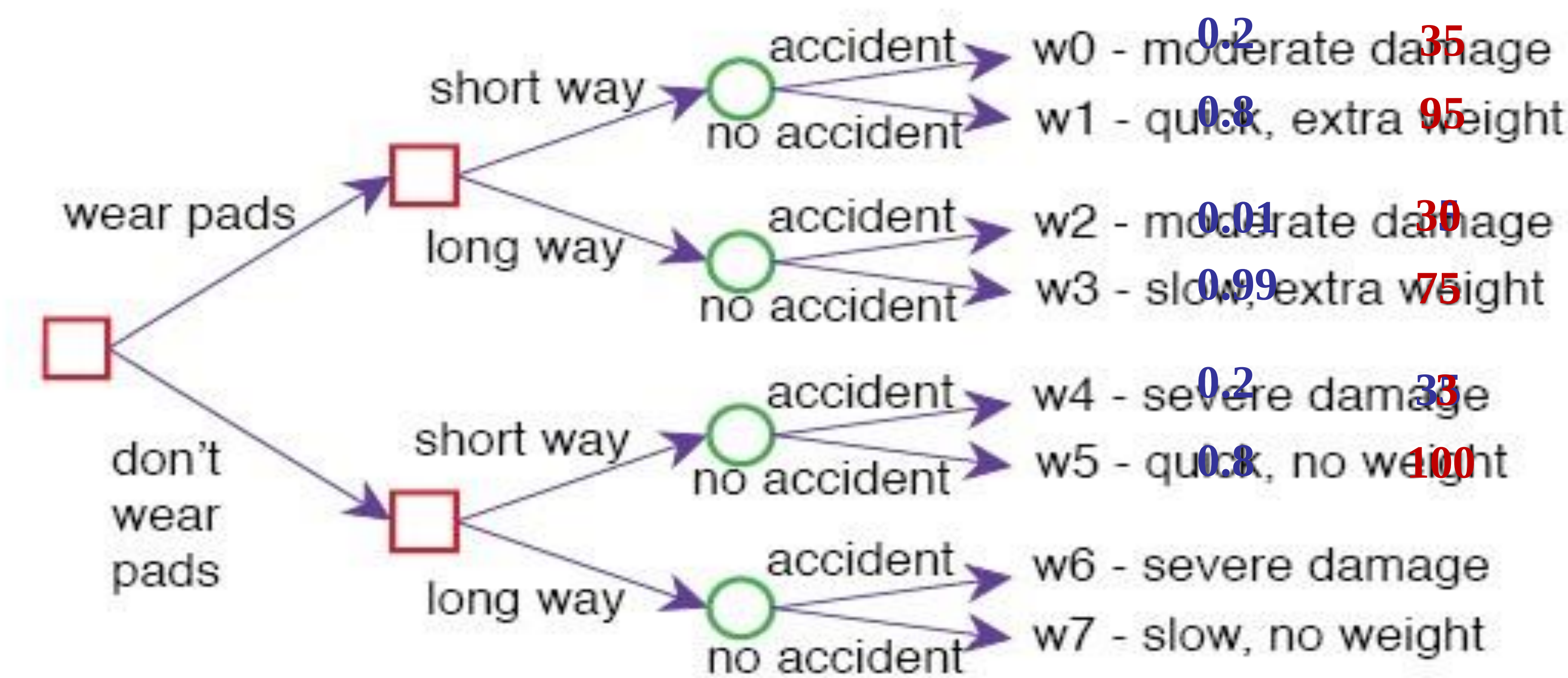
$$E(U \mid D = d_i) = \sum_{w \models (D = d_i)} P(w) \times U(w)$$

Example of Expected Utility

- The expected utility of decision $D = d$ is

$$E(U \mid D = d) = \sum_{w \models (D = d)} P(w)\, U(w) = \mathbf{P(w_1) \times U(w_1) + \ldots + P(w_n) \times U(w_n)}$$

- What is the expected utility of Wearpads=yes, Way=short ?



A. 7 B. 83 C. 76 D. 157.55

Probability Utility E[U|D]

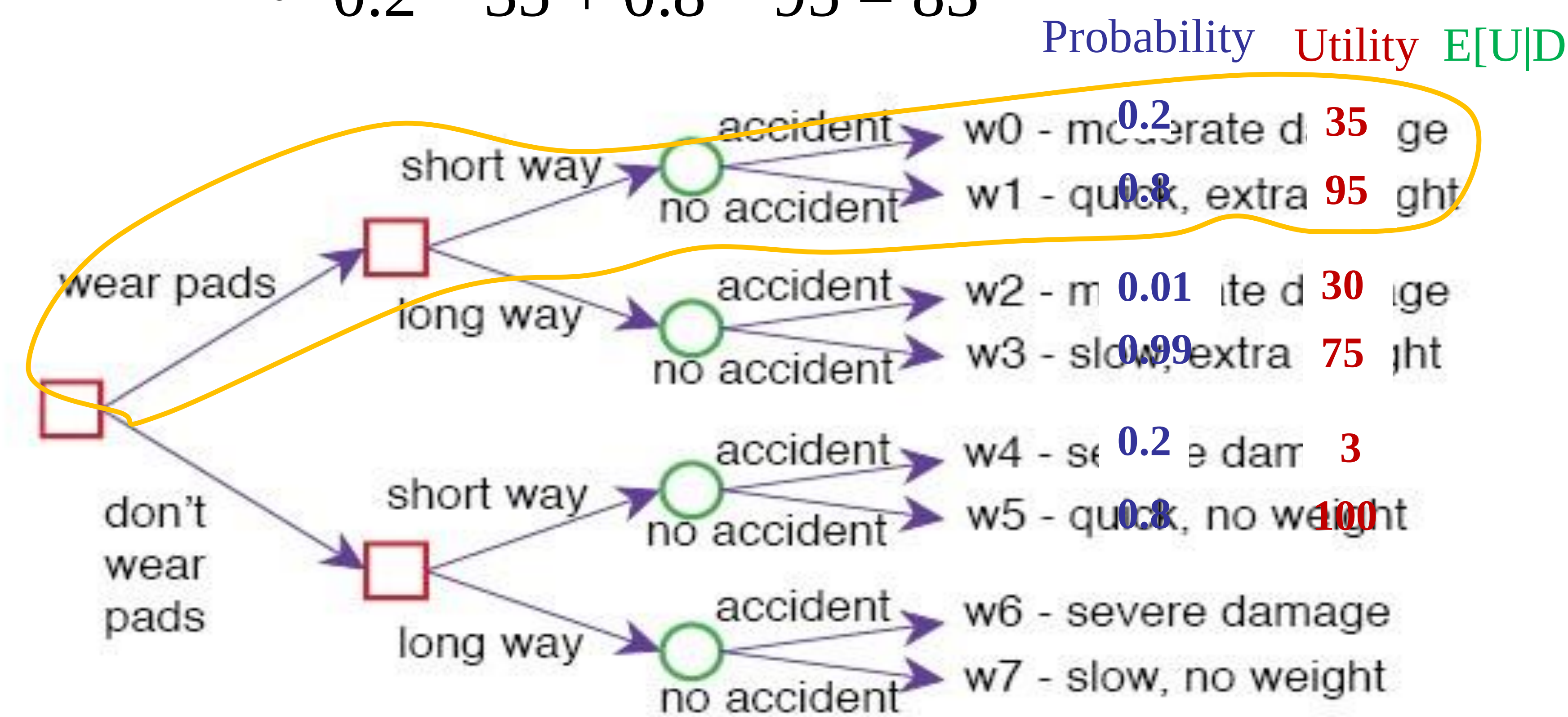
0.0135 **0** **0.99** **80**

Expected utility of a decision

- The expected utility of decision $D = d$ is

$$E(U \mid D = d) = \sum_{w \models (D = d)} P(w)\, U(w) = \mathbf{P(w_1) \times U(w_1) + \ldots + P(w_n) \times U(w_n)}$$

- $0.2 * 35 + 0.8 * 95 = 83$



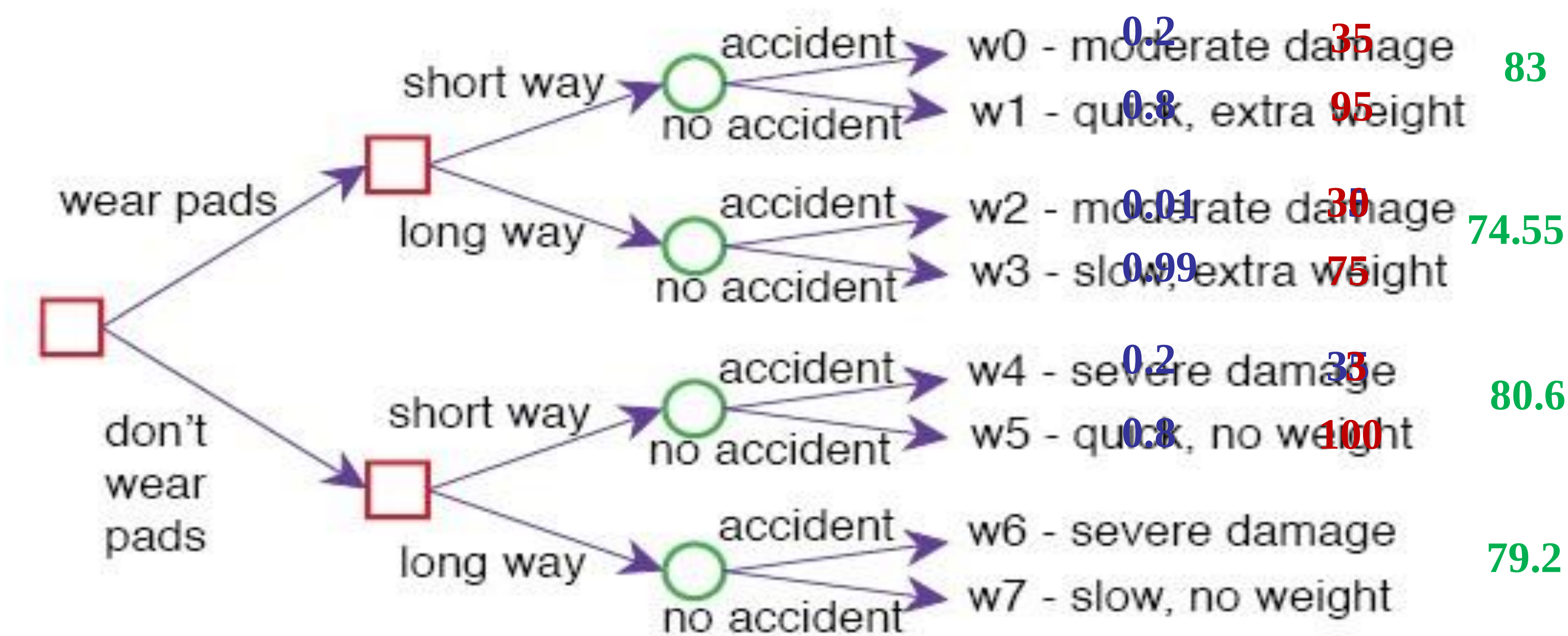
- What is the expected utility of Wearpads=yes, Way=short ?

0.01350 0.99 80

Expected utility of a decision

- The expected utility of decision $D = d$ is

$$E(U \mid D = d) = \sum_{w \models (D = d)} P(w)\, U(w) = \mathbf{P(w_1) \times U(w_1) + \ldots + P(w_n) \times U(w_n)}$$



Probability Utility E[U|D]

0.01350 0.99 80

The variable elimination algorithm,

To compute $P(Y=y_i | E = e)$

1. Construct a factor for each conditional probability.
2. For each factor, assign the observed variables E to their observed values.
3. Given an elimination ordering, decompose sum of products
4. Sum out all variables Z_i not involved in the query
5. Multiply the remaining factors (which only involve)
6. Normalize by dividing the resulting factor f( ) by $\sum_{y} f(Y)$

See the algorithm VE_BN in the P&M text, Section 6.4.1, Figure 6.8, p. 254.

The variable elimination algorithm,

To compute $P(Y=y_i | E = e)$

1. Construct a factor for each conditional probability.
2. For each factor, assign the observed variables E to their observed values.
3. Given an elimination ordering, decompose sum of products
4. Sum out all variables Z_i not involved in the query
5. Multiply the remaining factors (which only involve Y) 6.

   Normalize by dividing the resulting factor f(Y) by $\sum_{y} f(Y)$

See the algorithm VE_BN in the P&M text, Section 6.4.1, Figure 6.8, p. 254.

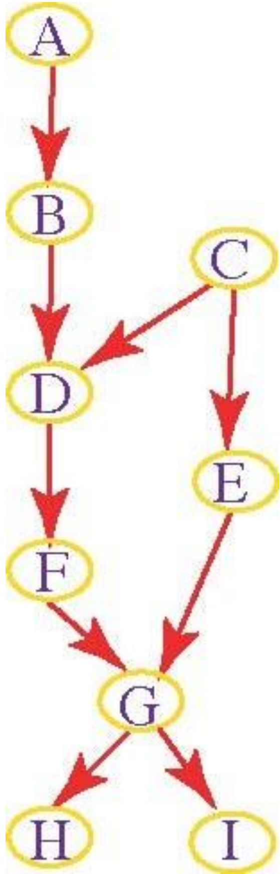
Lecture Overview

- Recap
- Variable Elimination
- Algorithm
  - VE example

Variable elimination example

Compute $P(G|H=h_1)$ in the network below .

$P(G,H) = \sum_{A,B,C,D,E,F,I} P(A,B,C,D,E,F,G,H,I) =$



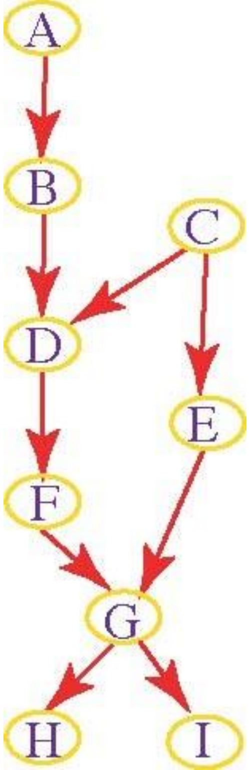
$$= \sum_{A,B,C,D,E,F,I} P(A)P(B|A)P(C)P(D|B,C)P(E|C)P(F|D)P(G|F,E)P(H|G)P(I|G)$$

Step 1: Construct a factor for each cond. probability

Compute $P(G|H=h_1)$ in the network below .

$$P(G,H) = \sum_{A,B,C,D,E,F,I} P(A)P(B|A)P(C)P(D|B,C)P(E|C)P(F|D)P(G|F,E)P(H|G)P(I|G)$$

$$P(G,H) = \sum_{A,B,C,D,E,F,I} f_0(A)\, f_1(B,A)\, f_2(C)\, f_3(D,B,C)\, f_4(E,C)\, f_5(F, D)\, f_6(G,F,E)\, f_7(H,G)\, f_8(I,G)$$



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$

- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

Step 2: assign to observed variables their values

Compute $P(G|H=h_1)$.

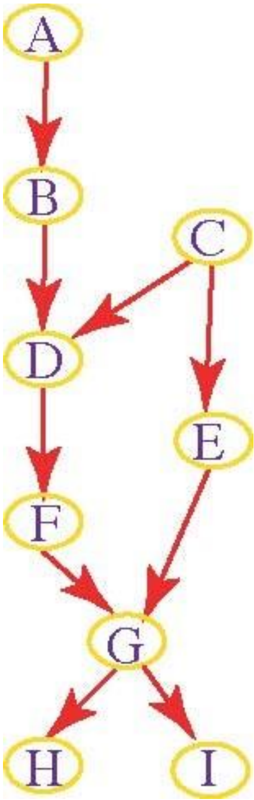
Previous state:

$P(G,H) = \sum_{A,B,C,D,E,F,I} f_0(A)\ f_1(B,A)\ f_2(C)\ f_3(D,B,C)\ f_4(E,C)\ f_5(F, D)\ f_6(G,F,E) f_7(H,G)$
$f_8(I,G)$

Observe H :

$P(G,H=h_1)=\sum_{A,B,C,D,E,F,I} f_0(A)\ f_1(B,A)\ f_2(C)\ f_3(D,B,C)\ f_4(E,C)\ f_5(F, D)\ f_6(G,F,E) f_9(G)\ f_8(I,G)$

- $f_7(H,G)$
- $f_8(I,G)$



- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$

$H=h_1$

- $f_0(A)$ • $f_9(G)$

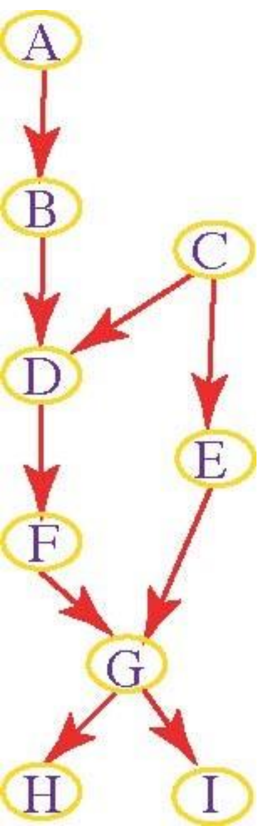
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

Step 3: Decompose sum of products

Compute $P(G|H=h_1)$.

Previous state:

$P(G,H=h_1) = \sum_{A,B,C,D,E,F,I} f_0(A)\, f_1(B,A)\, f_2(C)\, f_3(D,B,C)\, f_4(E,C)\, f_5(F, D)\, f_6(G,F,E) f_9(G) f_8(I,G)$



Elimination ordering A, C, E, I, B, D, F:

$P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B \sum_I f_8(I,G) \sum_E f_6(G,F,E) \sum_C f_2(C)\, f_3(D,B,C)\, f_4(E,C) \sum_A f_0(A)\, f_1(B,A)$

- $f_0(A)$
- $f_9(G)$
- $f_1(B,A)$
- $f_2(C)$
- $f_7(H,G)$
- $f_8(I,G)$

- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$

Step 4: sum out non query variables (one at a time)

Compute $P(G|H=h_1)$.

Elimination order: A,C,E,I,B,D,F

Previous state:

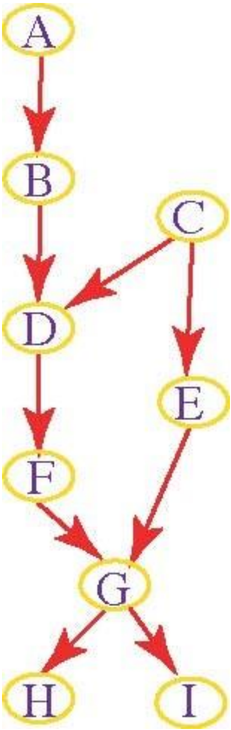
$P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B \sum_I f_8(I,G) \sum_E f_6(G,F,E) \sum_C f_2(C)\ f_3(D,B,C)\ f_4(E,C) \sum_A f_0(A)\ f_1(B,A)$

Eliminate A: perform product and sum out A in

- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

$$P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B) \sum_I f_8(I,G) \sum_E f_6(G,F,E) \sum_C f_2(C)\, f_3(D,B,C)\, f_4(E,C)$$

- $f_7(H,G)$
- $f_8(I,G)$



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

- $f_9(G)$
- $f_{10}(B)$

$f_{10}(B)$ does not depend on C, E, or I, so we can push it outside of those sums.

Step 4: sum out non query variables (one at a time)

Compute $P(G|H=h_1)$.

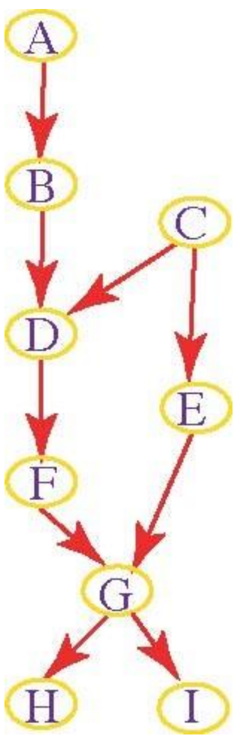
Elimination order: A,C,E,I,B,D,F

Previous state:

$P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B) \sum_I f_8(I,G) \sum_E f_6(G,F,E) \sum_C f_2(C)\, f_3(D,B,C)\, f_4(E,C)$

- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

Eliminate C: perform product and sum out C in



•f_0(A)

•f_1(B,A)

•f_2(C) a

•f_3(D,B,C)

•f_4(E,C)

•f_5(F, D)

•f_9(G)

•f_{10}(B)

f_{11}(B,C,D,E)

b f_{11}(B,D,E)

c f_{11}(D,E)

Step 4: sum out non query variables (one at a time)

•f_6(G,F,E)

•f_7(H,G)

•f_8(I,G)

d f_{11}(C, D,E)

Step 4: sum out non query variables (one at a time)

Compute $P(G|H=h_1)$.

Compute $P(G|H=h_1)$.

Elimination order: A,C,E,I,B,D,F

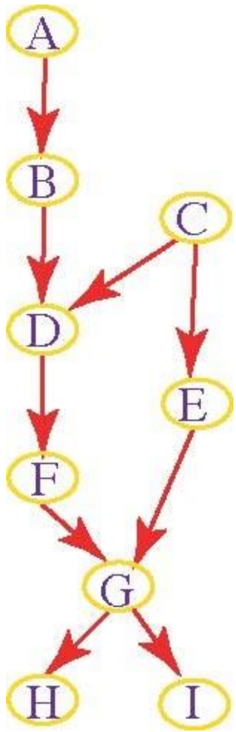
Previous state:

$P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B) \sum_I f_8(I,G) \sum_E f_6(G,F,E) \sum_C f_2(C)\ f_3(D,B,C)\ f_4(E,C)$

Eliminate C: perform product and sum out C in

$P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B) \sum_I f_8(I,G) \sum_E f_6(G,F,E) f_{11}(B,D,E)$

- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$



•$f_0(A)$

•$f_1(B,A)$

•$f_3(D,B,C)$

•$f_4(E,C)$

•$f_5(F, D)$

•$f_9(G)$

•$f_{10}(B)$

•$f_2(C)$•$f_{11}(B,D,E)$

Elimination order: A,C,E,I,B,D,F

Previous state:

$P(G,H=h_1) = P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B) \sum_I f_8(I,G) \sum_E f_6(G,F,E) f_{11}(B,D,E)$

Eliminate E: perform product and sum out E in

•$f_6(G,F,E)$

•$f_7(H,G)$

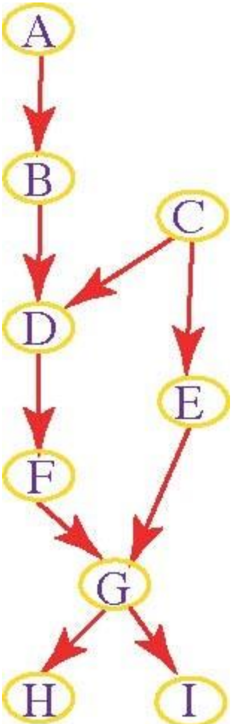
•$f_8(I,G)$

Step 4: sum out non query variables (one at a time)

Compute $P(G|H=h_1)$.

$P(G,H=h_1) = P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B) f_{12}(B,D,F,G) \sum_I f_8(I,G)$

- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$



- f_0(A)
- f_1(B,A)
- f_2(C)
- f_3(D,B,C)
- f_5(F, D)
- f_6(G,F,E)
- f_7(H,G)
- f_8(I,G)

- f_9(G)
- f_{10}(B)
- f_{11}(B,D,E)
- f_{12}(B,D,F,G)
- f_4(E,C)

Step 4: sum out non query variables (one at a time)

Compute $P(G|H=h_1)$.

Elimination order: A,C,E,**I**,B,D,F

Previous state:

$P(G,H=h_1) = P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B) f_{12}(B,D,F,G) \sum_I f_8(I,G)$

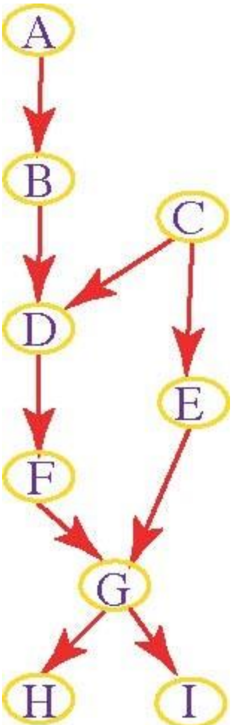
Eliminate I: perform product and sum out I in

$P(G,H=h_1) = P(G,H=h_1) = f_9(G)\, f_{13}(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B) f_{12}(B,D,F,G)$

- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

Step 4: sum out non query variables (one at a time)

Compute $P(G|H=h_1)$.



Previous state:

- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

- $f_9(G)$
- $f_{10}(B)$
- $f_{11}(B,D,E)$
- $f_{12}(B,D,F,G)$
- $f_4(E,C)$
- $f_{13}(G)$

Elimination order:
A,C,E,I,**B**,D,F

Step 4: sum out non query variables (one at a time)

Compute $P(G|H=h_1)$.

$P(G,H=h_1) = P(G,H=h_1) = f_9(G)\, f_{13}(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B)\, f_{12}(B,D,F,G)$

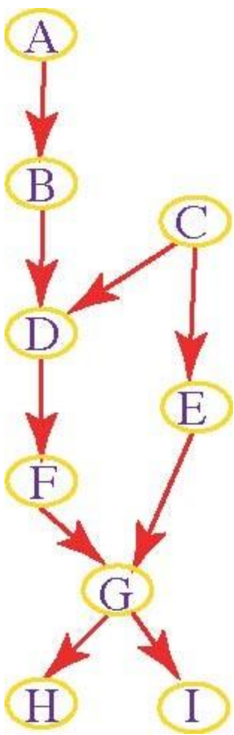
Eliminate B: perform product and sum out B in

$P(G,H=h_1) = P(G,H=h_1) = f_9(G)\, f_{13}(G) \sum_F \sum_D f_5(F, D)\, f_{14}(D,F,G)$

- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

Step 4: sum out non query variables (one at a time)

Compute $P(G|H=h_1)$.



•$f_0(A)$

•$f_1(B,A)$•$f_{10}(B)$

•$f_2(C)$•$f_{11}(B,D,E)$

•$f_3(D,B,C)$

•$f_{12}(B,D,F,G)$

•$f_9(G)$

•$f_4(E,C)$•$f_{13}(G)$

•$f_5(F, D)$•$f_{14}(D,F,G)$

Elimination order: A,C,E,I,B,**D**,F

Previous state:

•$f_6(G,F,E)$

•$f_7(H,G)$

•$f_8(I,G)$

Step 4: sum out non query variables (one at a time)

Compute $P(G|H=h_1)$.

$P(G,H=h_1) = P(G,H=h_1) = f_9(G)\, f_{13}(G) \sum_F \sum_D f_5(F, D)\, f_{14}(D,F,G)$

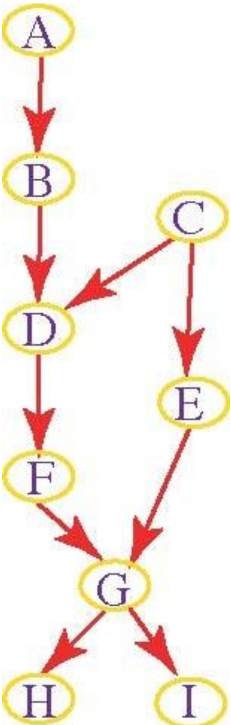
Eliminate D: perform product and sum out D in

$P(G,H=h_1) = P(G,H=h_1) = f_9(G)\, f_{13}(G) \sum_F f_{15}(F,G)$

- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

Step 4: sum out non query variables (one at a time)

Compute $P(G|H=h_1)$.



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$
- $f_9(G)$
- $f_{10}(B)$
- $f_{11}(B,D,E)$
- $f_{12}(B,D,F,G)$
- $f_{13}(G)$
- $f_{14}(D,F,G)$
- $f_{15}(F,G)$

Step 4: sum out non query variables (one at a time)

Compute $P(G|H=h_1)$.

Elimination order: A,C,E,I,B,D,F

Previous state:

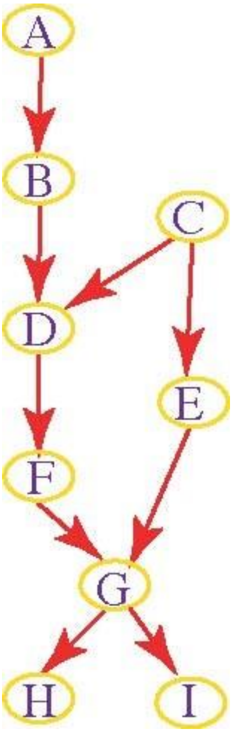
$P(G,H=h_1) = P(G,H=h_1) = f_9(G)\ f_{13}(G) \sum_F f_{15}(F,G)$

Eliminate F: perform product and sum out F in $f_9(G)$
$f_{13}(G) f_{16}(G)$

- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

Step 4: sum out non query variables (one at a time)

Compute $P(G|H=h_1)$.



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

- $f_9(G)$
- $f_{10}(B)$
- $f_{11}(B,D,E)$
- $f_{12}(B,D,F,G)$
- $f_{13}(G)$
- $f_{14}(D,F,G)$
- $f_{15}(F,G)$

Step 4: sum out non query variables (one at a time)

Compute $P(G|H=h_1)$.

- $f_{16}(G)$

- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

Step 5: Multiply remaining factors

Compute

$P(G|H=h_1)$.

Elimination order: A,C,E,I,B,D,F

Previous state:

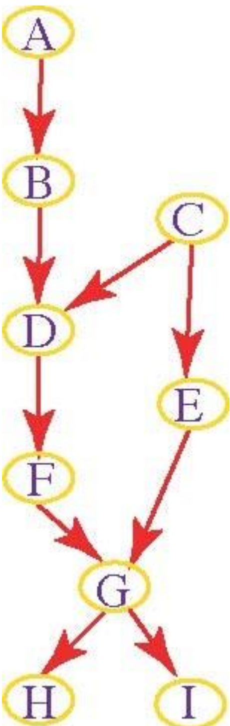
$P(G,H=h_1) = f_9(G)\, f_{13}(G) f_{16}(G)$

Multiply remaining factors (all in G):

- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

$P(G,H=h_1) = f_{17}(G)$

Now we just need to normalize



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$ 12
- $f_4(E,C)$ $f_{17}(G)$
- 13
- $f_5(F, D)$
- $f_{15}(F,G)$

- $f_9(G)$
- $f_{10}(B)$
- $f_{11}(B,D,E)$
- 14

- f (B,D,F,G)
- f (G)
- f (D,F,G)
- $f_{16}(G)$

Inference in General

- Y: subset of variables that is queried (e.g. Temperaturein previous example)

- E: subset of variables that are observed . E = e (W = yesin previous example)
- $Z_1, \ldots, Z_k$ remaining variables in the JPD (Cloudyin previous example)

We need to compute this numerator for each value of Y, y_i

We need to marginalize over all the variables $Z_1,\ldots Z_k$ not involved in the query $P(Y = y_i, E = e) = \sum_{Z_1} \ldots \sum_{Z_k} P(Z_1,\ldots,Z_k, Y = y_i, E = e)$

$$P(Y \mid E=e) = \frac{P(Y, E=e)}{P(E=e)}$$ Def of conditional probability

$\underline{P(Y,E=e)}$ To compute the denominator, marginalize over Y

$\sum P(Y,E=e)$ constant- Same value for every ensuring that P(Y=$\sum$y$P(Y_i)$.

Normalization $= y_i \mid E) = 1$

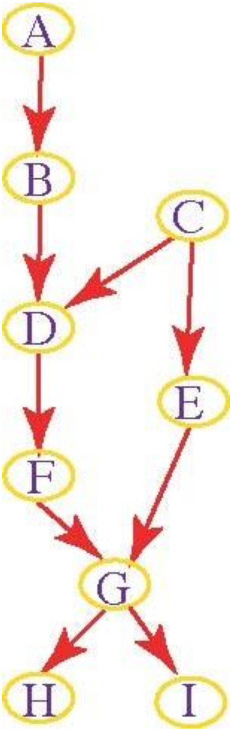
Y Y

We are just missing Step 6: Normalize

Compute P(G|H=
from P(G , H=h_1)

$$P(G = g \mid H = h_1) = \frac{P(G = g, H = h_1)}{P(H = h_1)}$$

$$= \frac{P(G = g, H = h_1)}{\sum_{g' \in dom(G)} P(G = g', H = h_1)} = \frac{f_{17}(g)}{\sum_{g' \in dom(G)} f_{17}(g')}$$



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$
- $f_9(G)$
- $f_{10}(B)$
- $f_{11}(B,D,E)$
- $f_{12}(B,D,F,G)$
- $f_{13}(G)$
- $f_{14}(D,F,G)$
- $f_{15}(F,G)$
- $f_{16}(G)$

$f_{17}(G)$

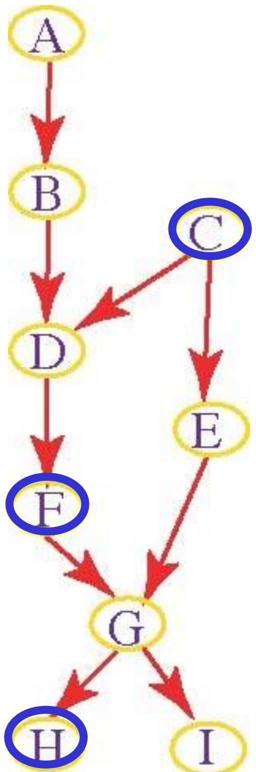
VE and conditional independence

So far, we haven't use conditional independence!

- Before running VE, we can prune all variables Z that are conditionally independent of the query Y given evidence E: $Z \perp\!\!\!\perp Y \mid E$

a	A,B,D
b	D,E

- They cannot change the belief over Y given E! • Example: which variables can we prune for the query $P(G=g| C=c_1, F=f_1, H=h_1)$?



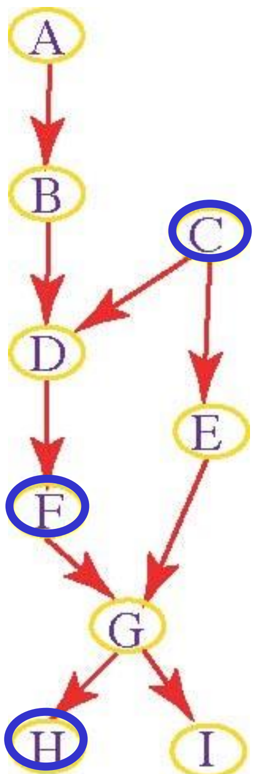
VE and conditional independence

So far, we haven't use conditional independence!

- Before running VE, we can prune all variables Z that are conditionally independent of the query Y given evidence E: $Z ⫫ Y \mid E$

c	A,B,D,E
d	None

- They cannot change the belief over Y given E! • Example: which

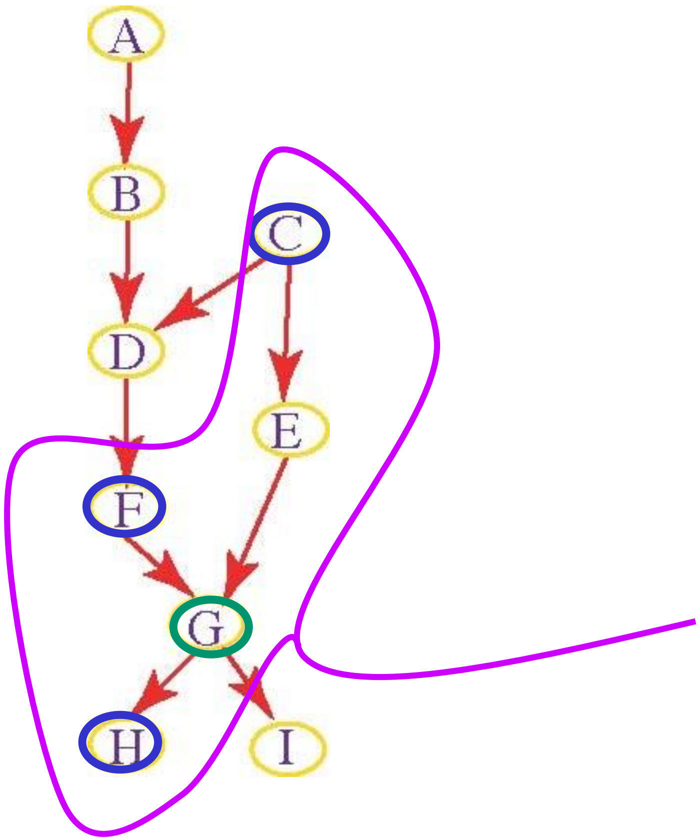


variables can we prune for the query P(G=g| C=c_1, F=f_1, H=h_1) ?

- Both a A,B,D paths from these nodes to G are blocked
- F is observed node in chain structure
- C is an observed common parent

Variable elimination: pruning

- We can also prune unobserved leaf nodes
- Since they are unobserved and not predecessors of the query nodes, they cannot influence the posterior probability of the query nodes



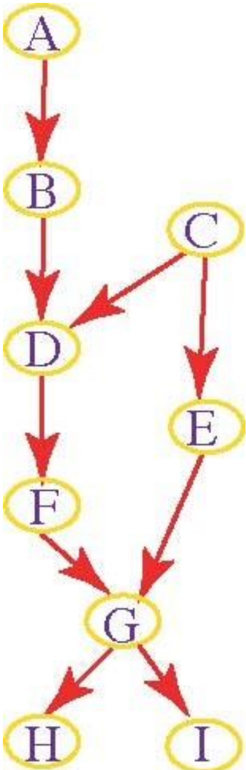
Thus, if the query is

$P(G=g \mid C=c_1, F=f_1, H=h_1)$
we only need to consider
this subnetwork

Slide 58

One last trick

- We can also prune unobserved leaf nodes
- And we can do so recursively



E.g., which nodes can we prune if the query is P(A)?

Recursively prune unobserved leaf nodes:
we can prune all nodes other than A !

Applet for Bayesian and Decision Networks

The Belief and Decision Networks applet you to load predefined Bayesian and Decision networks (see next topic) for various domains and run queries on them.

Select one of the available examples via “File -> Load Sample Problem

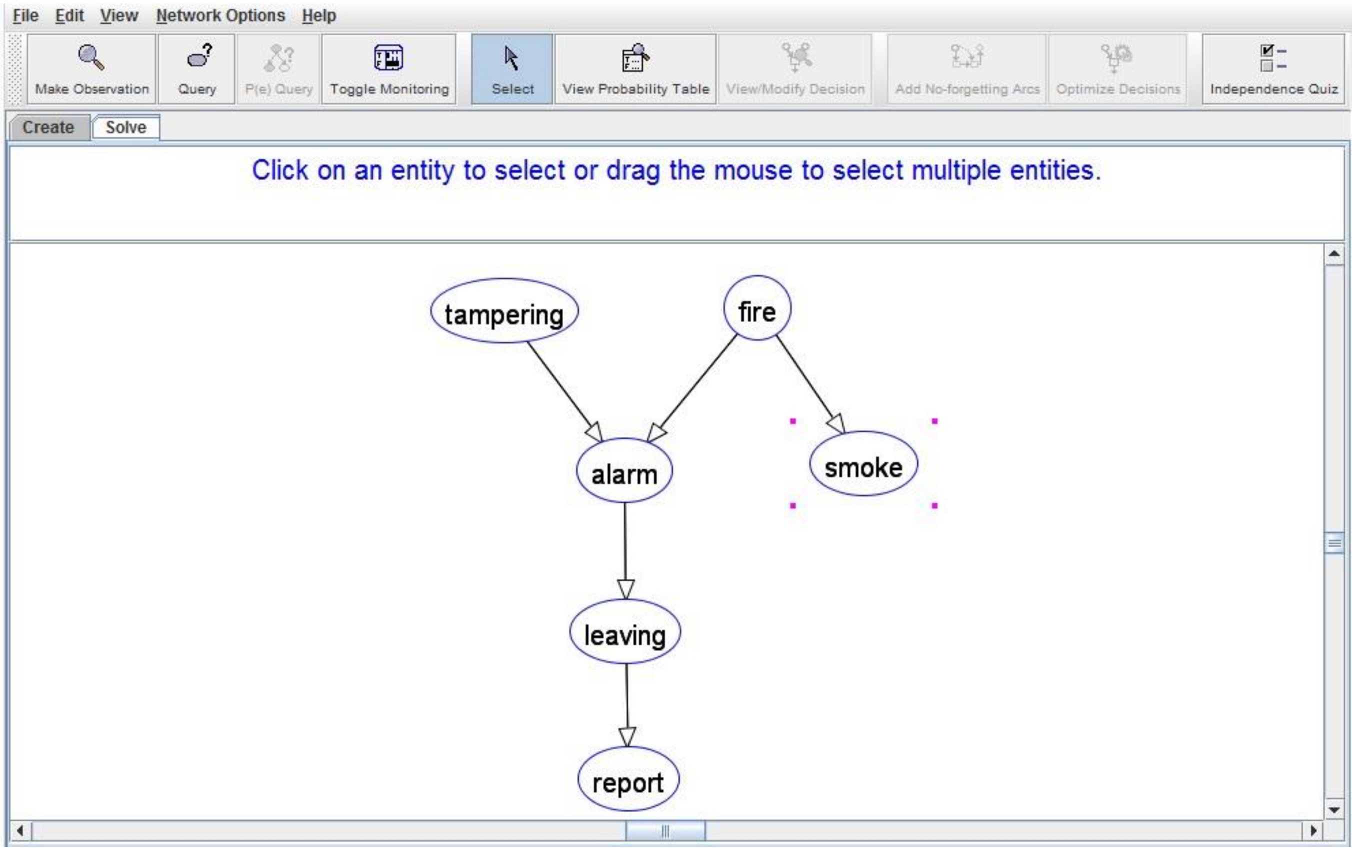
For Bayesian Networks

- Choose any of the examples above the blue line in the list that appears
- Right click on a node to perform any of these operations
- View the CPT table for a node
- Make an observation for a variable (i.e., set it to one of its values)
- Query the current probability distribution for a node given the observations made
- A dialogue box will appear the first time you do this. Select
- "Brief" if you just want to see the new probability
- "Verbose" if you want to see how VE computes it

See available help pages and video tutorials for more details on how to use the Bayes applet (http://www.aispace.org/bayes/)

VE in AISpace

- To see how variable elimination works in the Aispace Applet
- Select “Network options -> Query Models > verbose”
- Compare what happens when you select “Prune Irrelevant variables” or not in the VE window that pops up when you query a node



- Try different heuristics for elimination ordering

Query P(A given L=F,S=T)

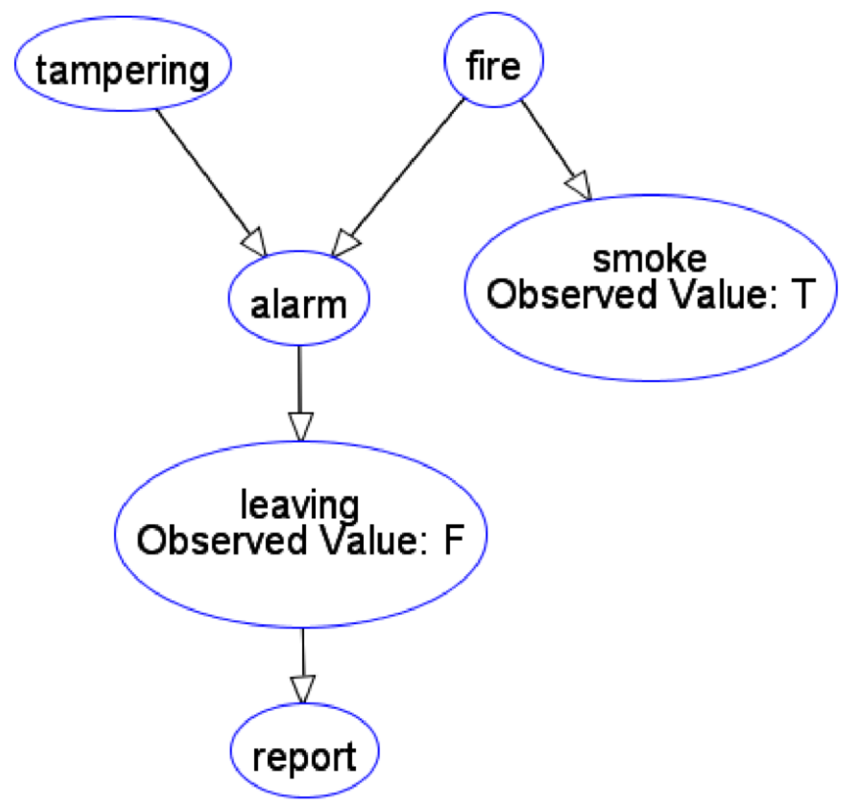
Making Observations in the Applet

Right click on a node and select “Make Observations”

Below we observed Smoke = T and Leaving = F

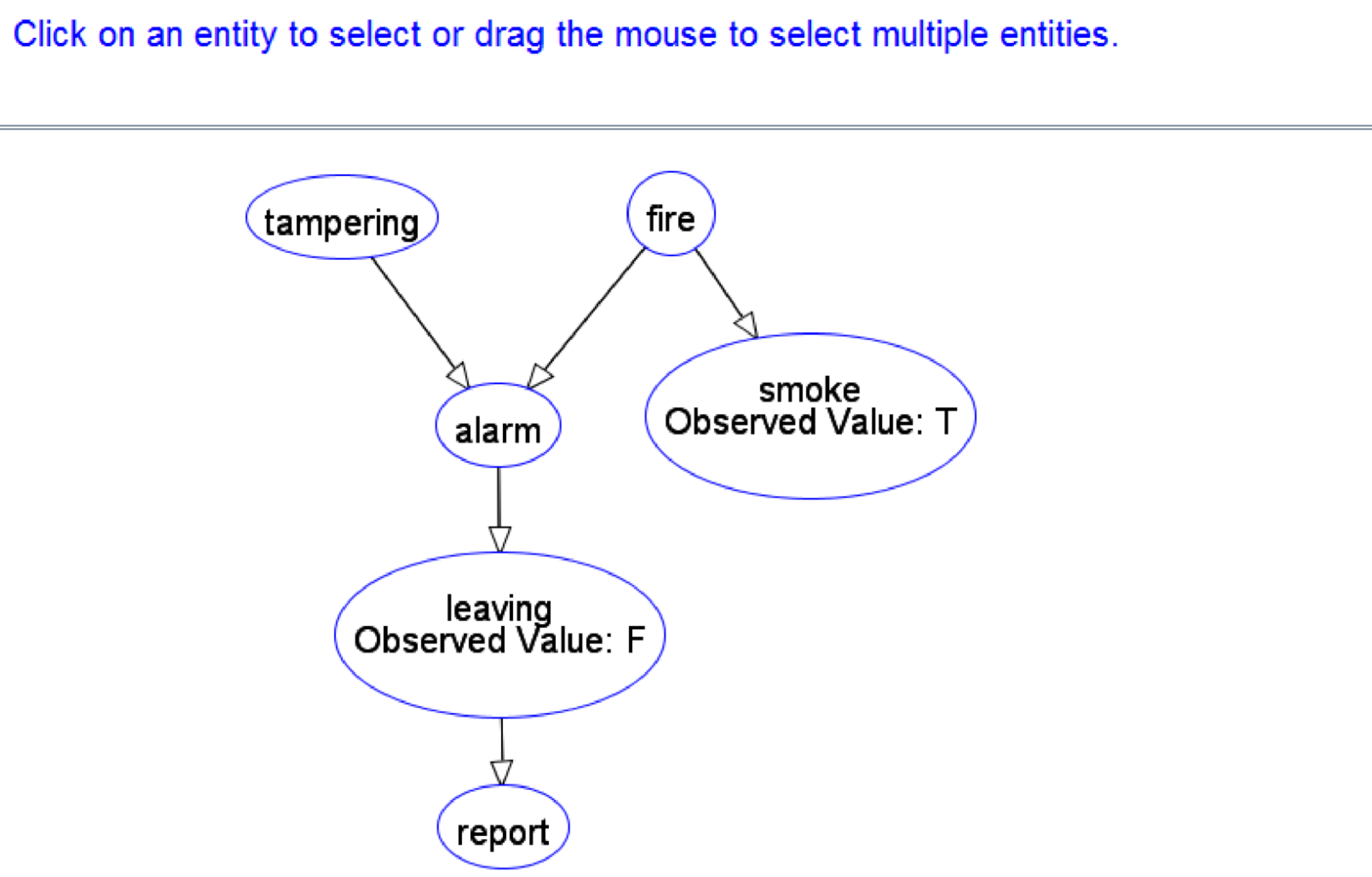
Click on an entity to select or drag the mouse to select multiple entities.

Query P(A given L=F,S=T)



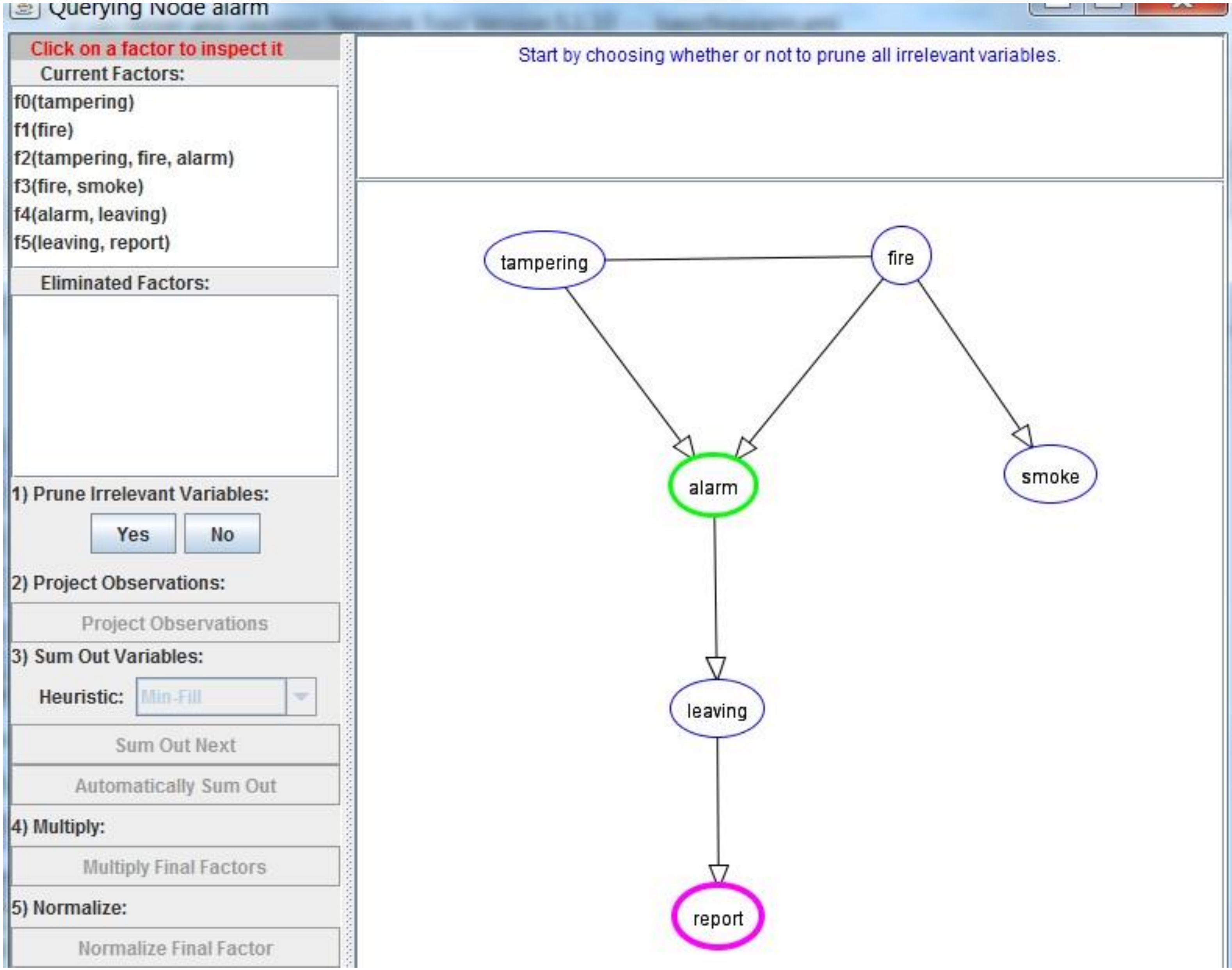
Making Observations in the Applet

Now query “alam”, that is



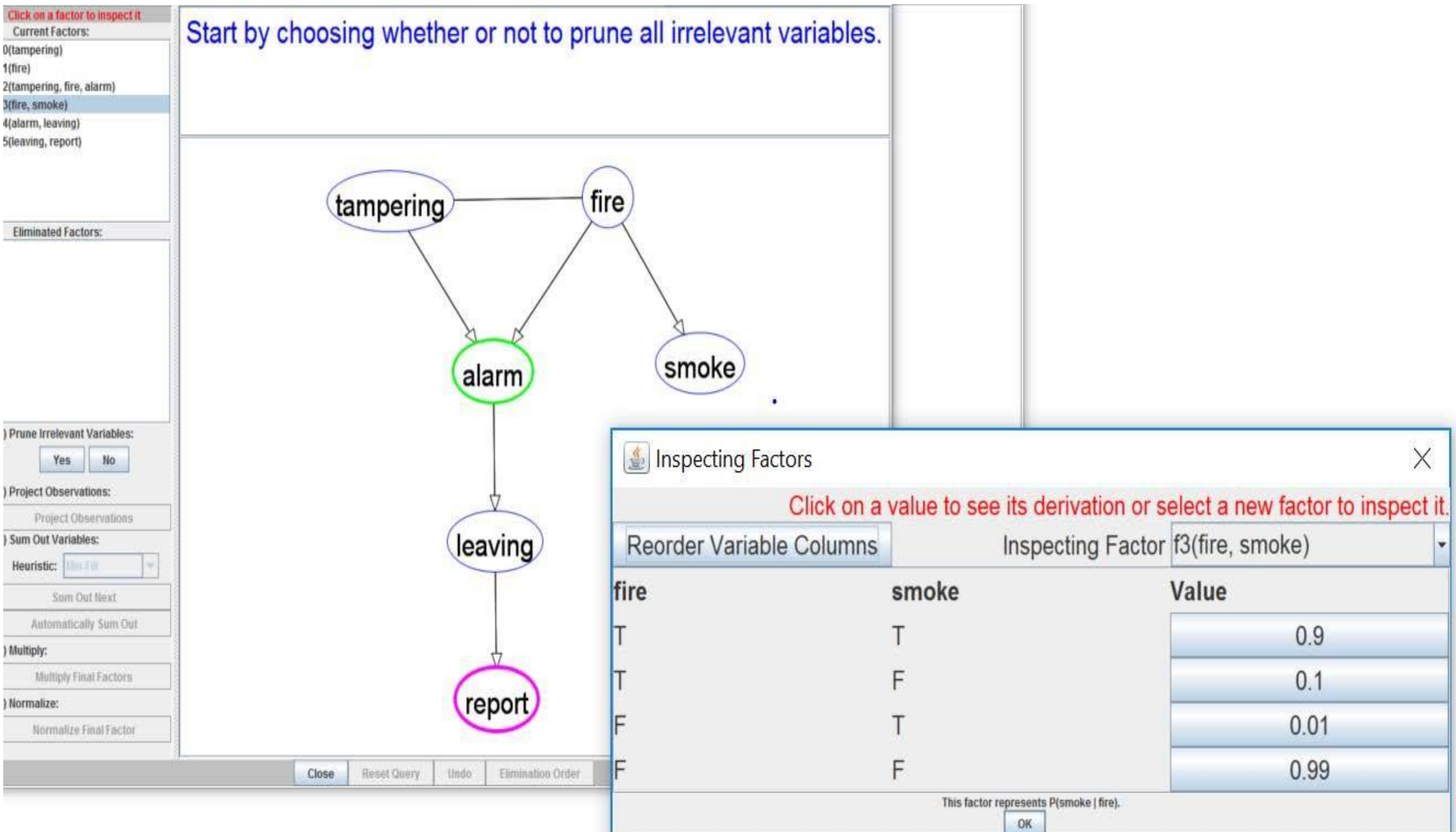
P(A given L=F,S=T)

Query P(A given L=F,S=T) - initial screen for VE


Querying Node alarm
Click on a factor to inspect it
Current Factors:
f0(tampering)
f1(fire)
f2(tampering, fire, alarm)
f3(fire, smoke)
f4(alarm, leaving)
f5(leaving, report)
Eliminated Factors:
1) Prune Irrelevant Variables:
Yes
No
2) Project Observations:
Project Observations
3) Sum Out Variables:
Heuristic:
Min-Fill
Sum Out Next
Automatically Sum Out
4) Multiply:
Multiply Final Factors
5) Normalize:
Normalize Final Factor
Start by choosing whether or not to prune all irrelevant variables.
tampering
fire
alarm
smoke
leaving
report

View Factors

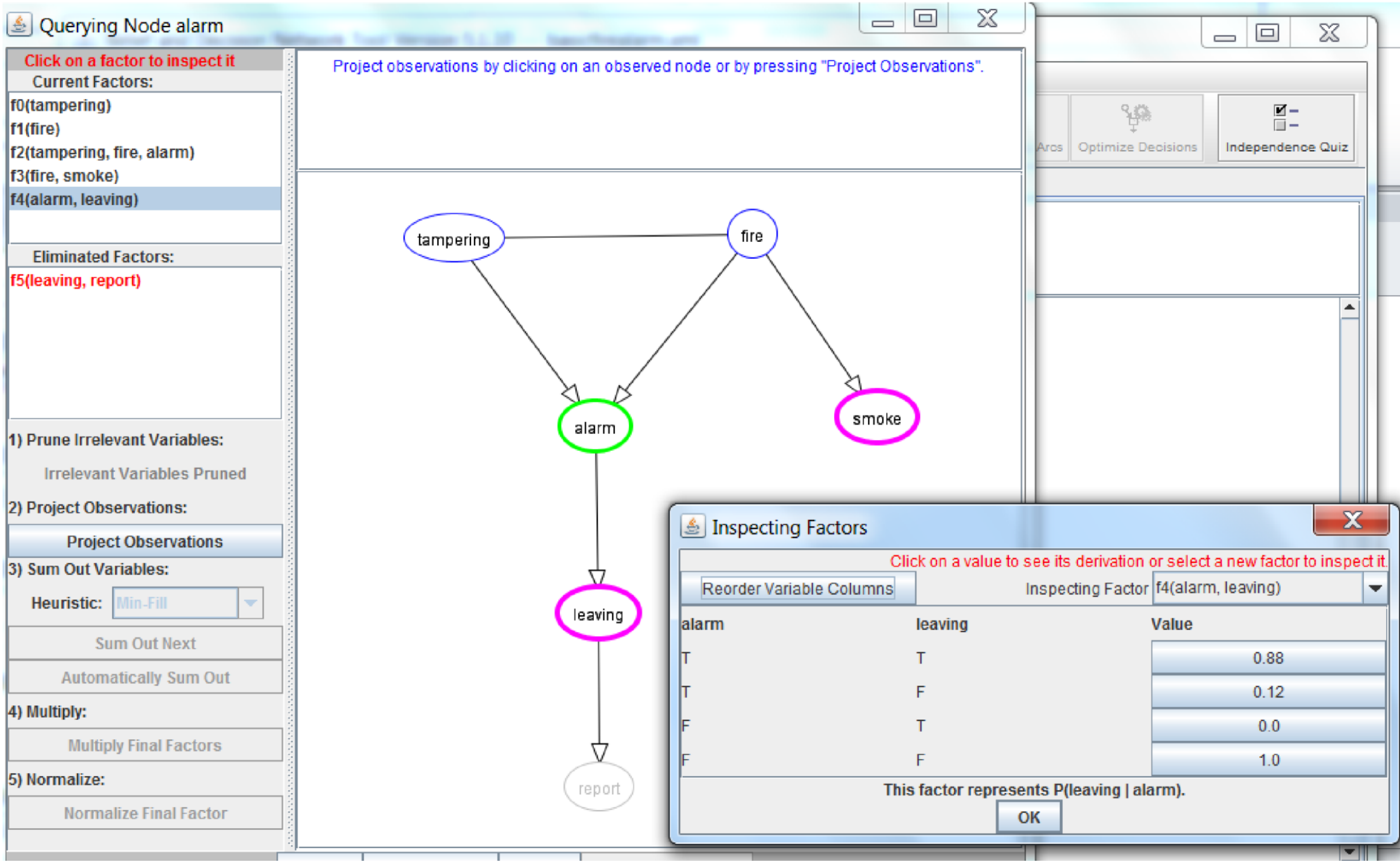
You can click on any factor in the “current factors” box to see its table ,



e.g. f(fire, smoke) below, which is just P(smoke|fire)

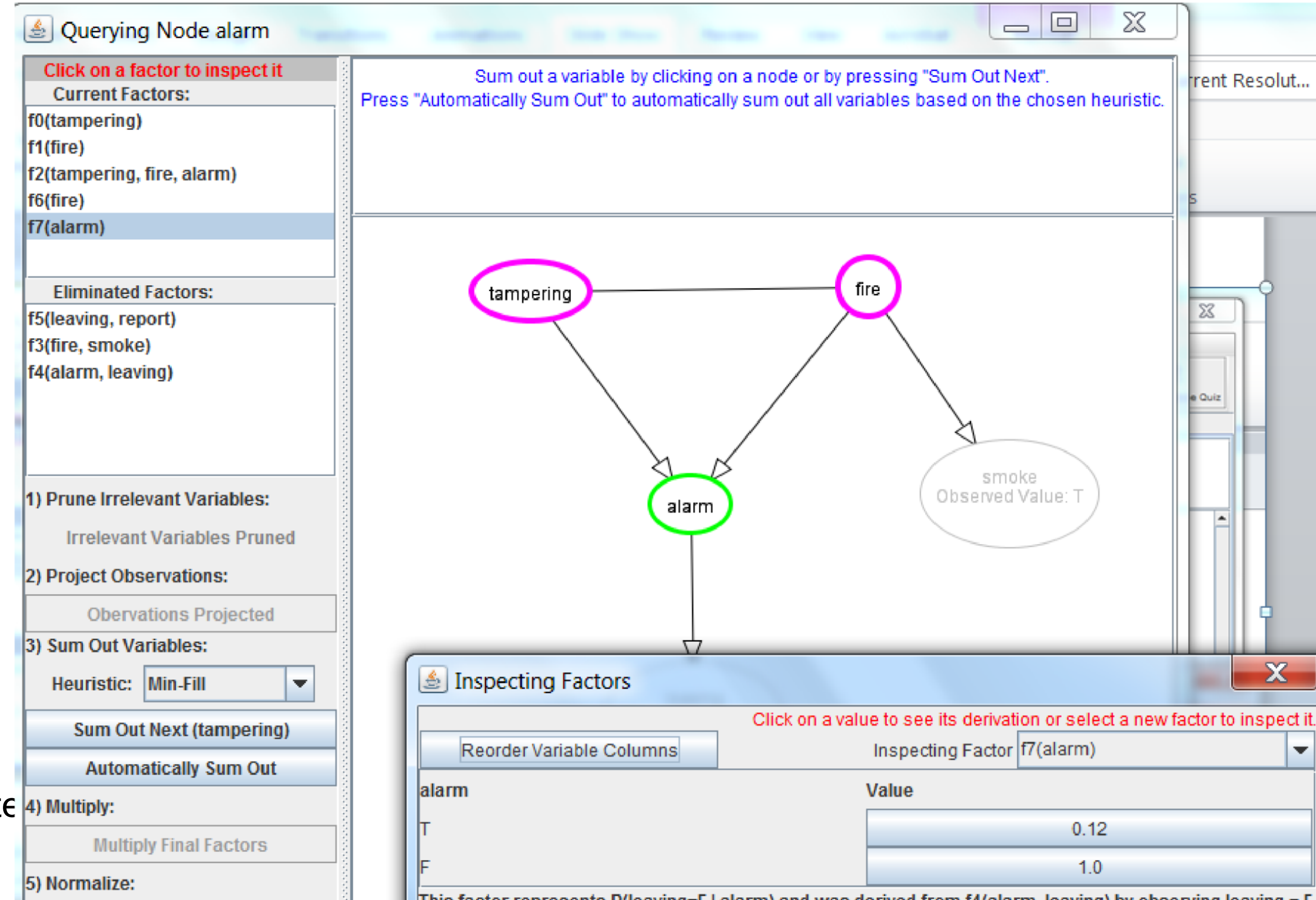
View Factors

You can click on any factor in the “current factors” or “ Eliminated factors” box to



see its table , e.g. f(alarm, leaving) below, which is just P(leaving|alarm)

Assigning evidence (Project Observations)



Afte

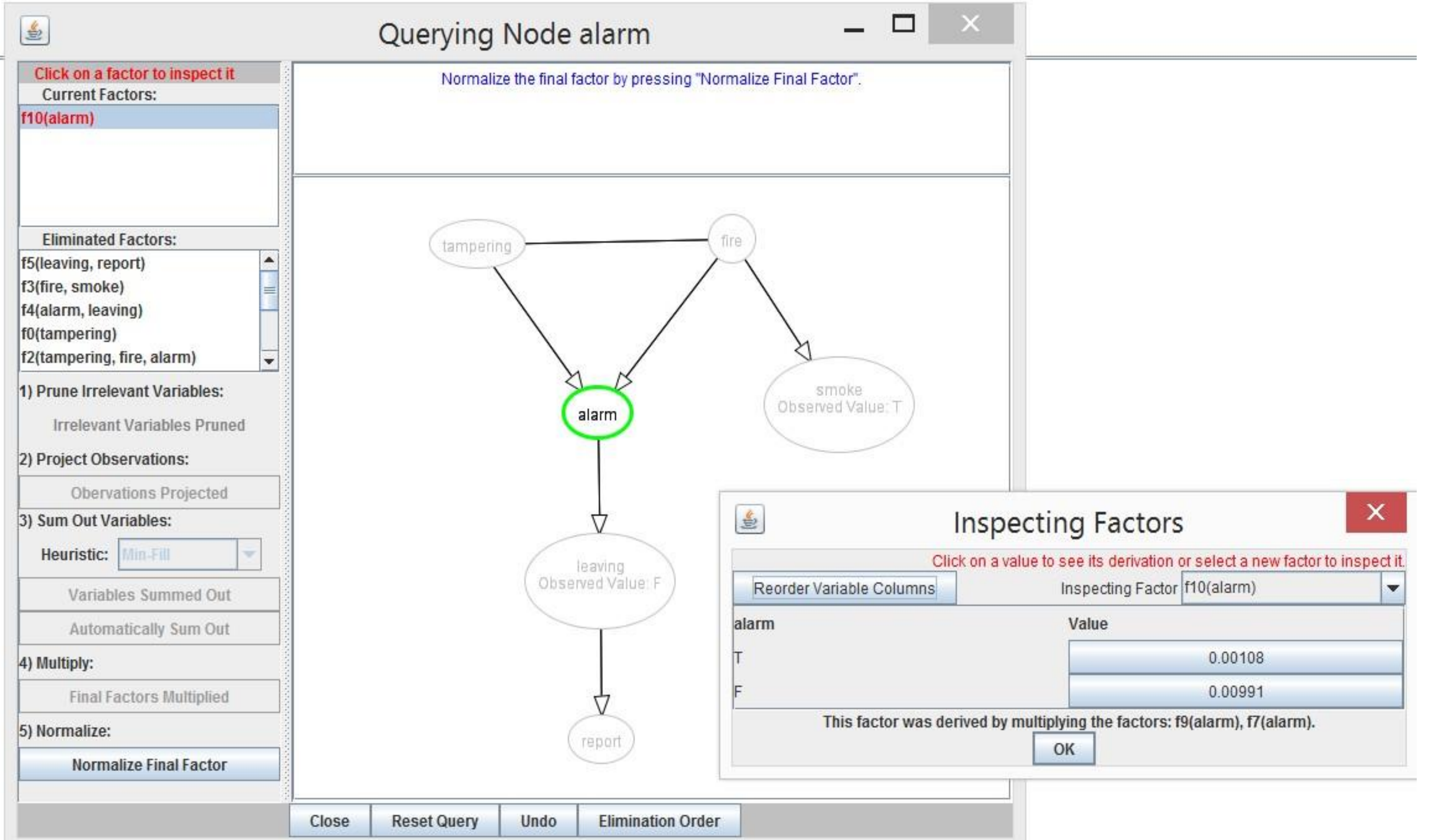
Assigning evidence (Project Observations)

f4(alarm, leaving) that correspond to L=F

Assigning evidence (Project Observations)

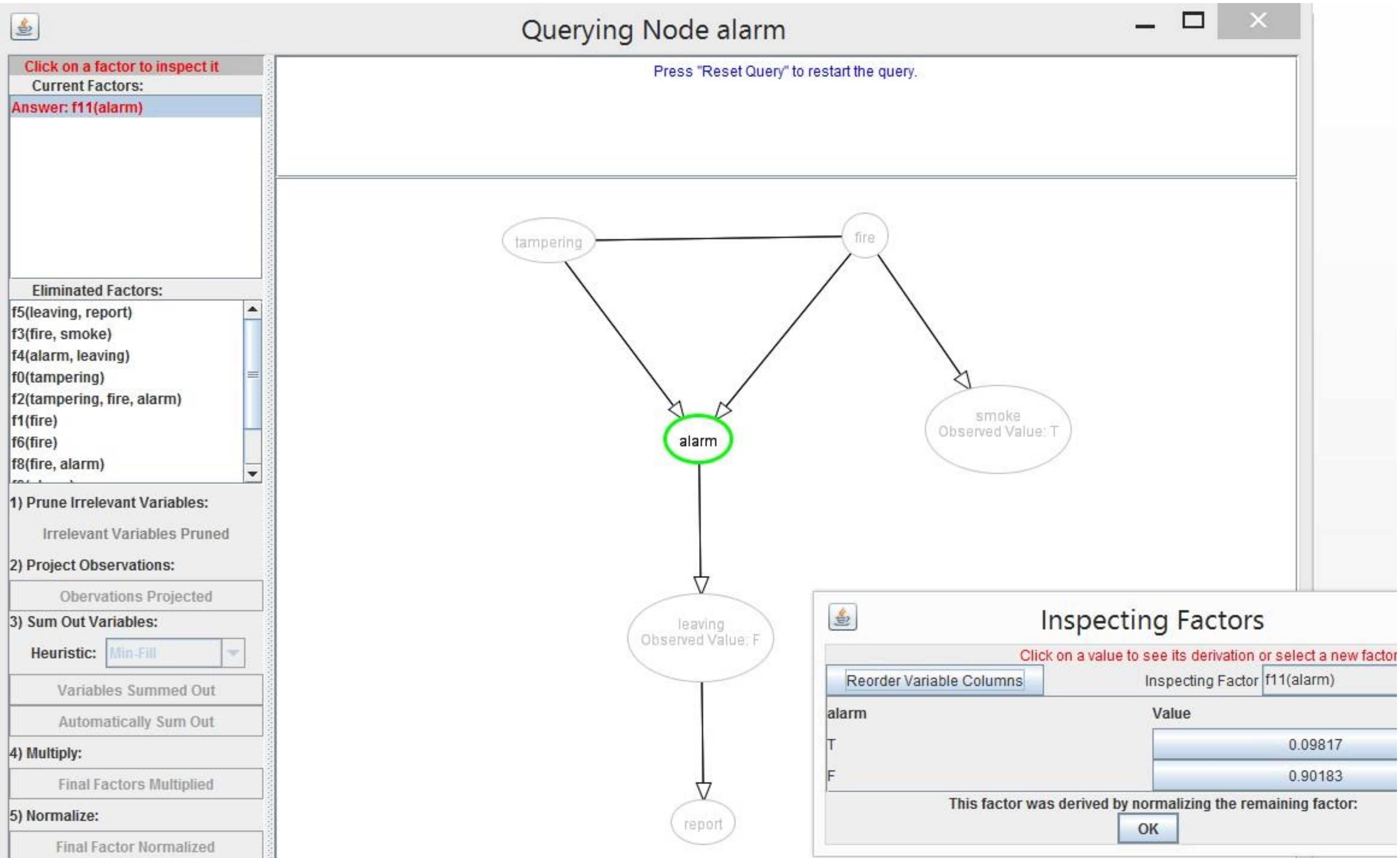
Keep summing our variables and check out the factors (current and eliminated) to understand the underling operations. Below is the outcome for this query, before normalization

Assigning evidence (Project Observations)



Assigning evidence (Project Observations)

Keep summing our variables and check out the factors (current and eliminated) to



Assigning evidence (Project Observations)

understand the underling operations. Below is the outcome for this query, after normalization

Complexity of Variable Elimination (VE) (not required)

- A factor over n binary variables has to store 2^n numbers
- The initial factors are typically quite small (variables typically only have few parents in Bayesian networks)
- But variable elimination constructs larger factors by multiplying factors together
- The complexity of VE is exponential in the maximum number of variables in any factor during its execution
- This number is called the treewidth of a graph (along an ordering)
- Elimination ordering influences treewidth
- Finding the best ordering is NP complete

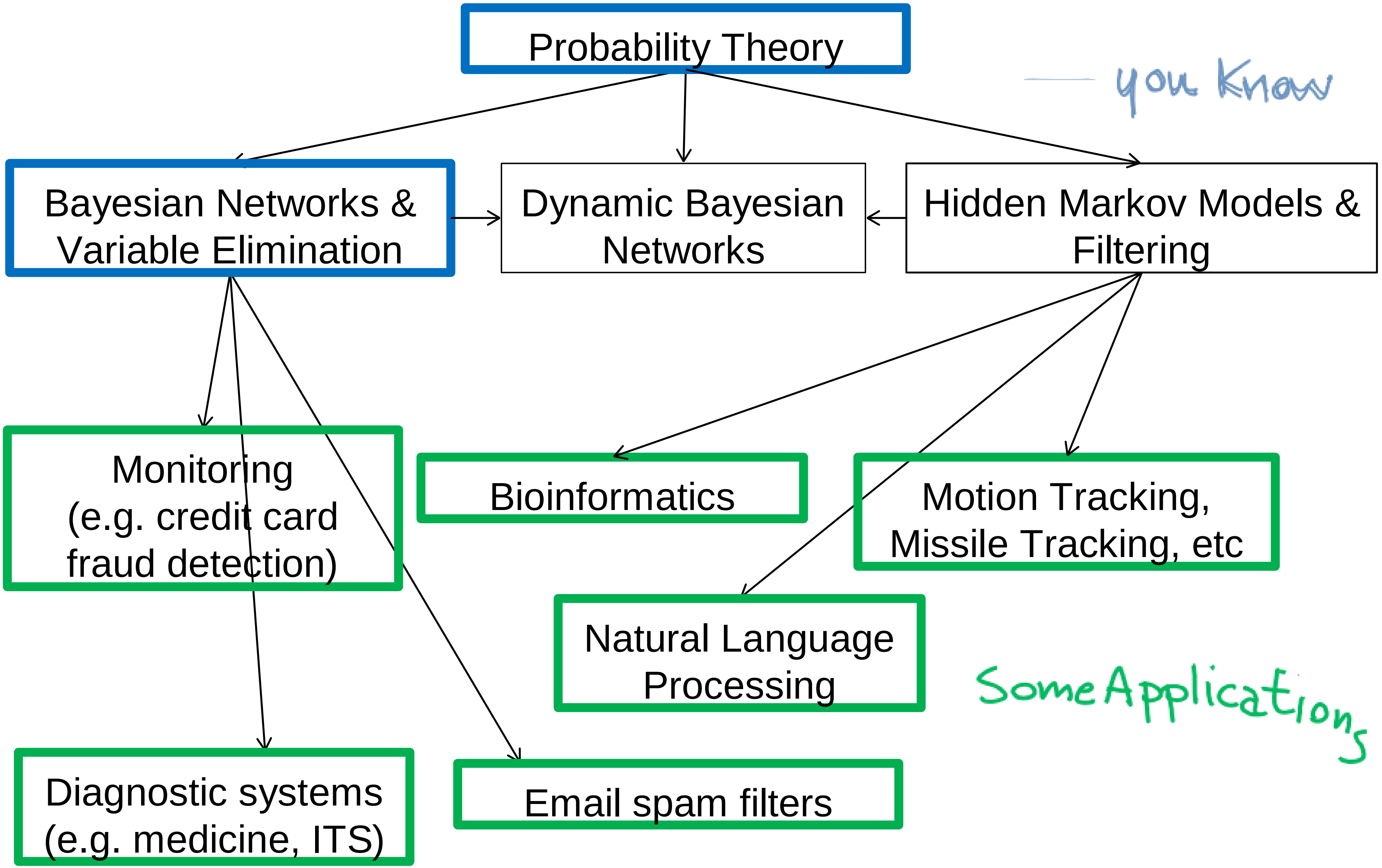
- I.e., the ordering that generates the minimum treewidth
- Heuristics work well in practice (e.g. least connected variables first) • Even with best ordering, inference is sometimes infeasible
  - ✓ In those cases, we need approximate inference. See CS422 & CS540

Learning Goals For Bnets

- Build a Bayesian Network for a given domain
- Identify the necessary CPTs
- Compare different network structures
- Understand dependencies and independencies
- Variable elimination

- Understating factors and their operations
- Carry out variable elimination by using factors and the related operations
- Use techniques to simplify variable elimination

Big picture: Reasoning Under Uncertainty



Probability Theory
— you know
Bayesian Networks & Variable Elimination
Dynamic Bayesian Networks
Hidden Markov Models & Filtering
Monitoring (e.g. credit card fraud detection)
Bioinformatics
Motion Tracking, Missile Tracking, etc
Natural Language Processing
Some Applications
Diagnostic systems (e.g. medicine, ITS)
Email spam filters

Where are we?

Representation

Reasoning

Technique

Environment

Deterministic

Stochastic

Problem Type

Static

Sequential

Constraint Satisfaction

VarsConstraints+

Search

Arc

Consistency

Query

Logics

Search

STRIPS

Planning

Search

This concludes the module on answering queries in stochastic environments

Belief Nets

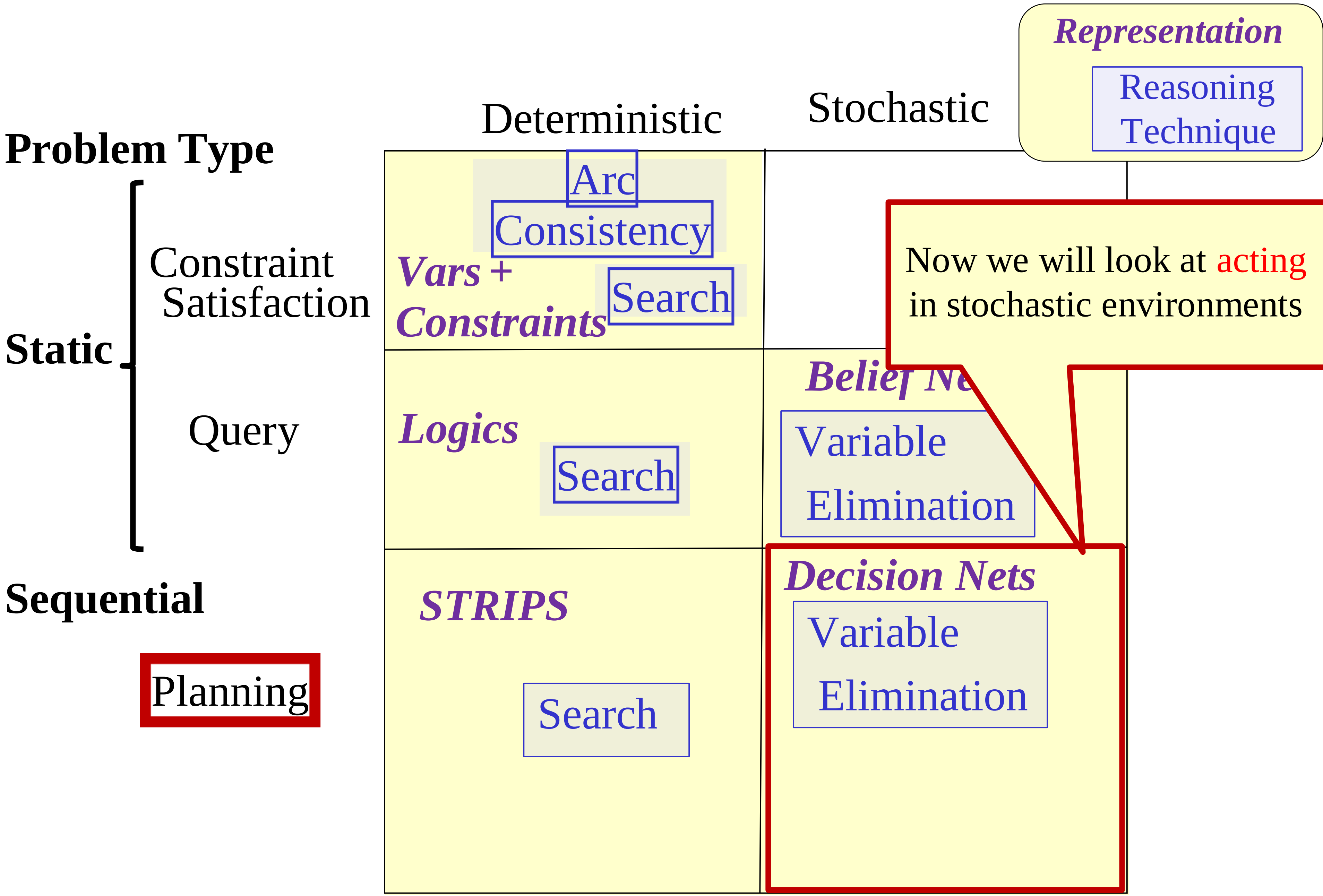
Variable

Elimination

Decision Nets

Variable

Elimination



Representation
Reasoning Technique
Deterministic
Stochastic
Problem Type
Static
Constraint Satisfaction
Query
Sequential
Planning
Arc Consistency
Vars + Constraints
Search
Logics
Search
STRIPS
Search
Belief Ne
Variable Elimination
Decision Nets
Variable Elimination
Now we will look at acting in stochastic environments

Environment