Bayesian networks (2)

Lirong Xia

Rensselaer

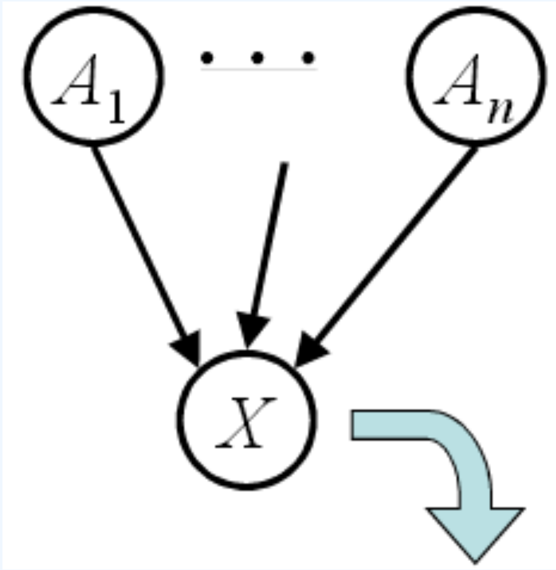
Last class

- Bayesian networks
  - compact, graphical representation of a joint probability distribution
  - conditional independence

Bayesian network

- Definition of Bayesian network (Bayes' net or BN)
- A set of nodes, one per variable X
- A directed, acyclic graph
- A conditional distribution for each node
  - A collection of distributions over X, one for each combination of parents' values

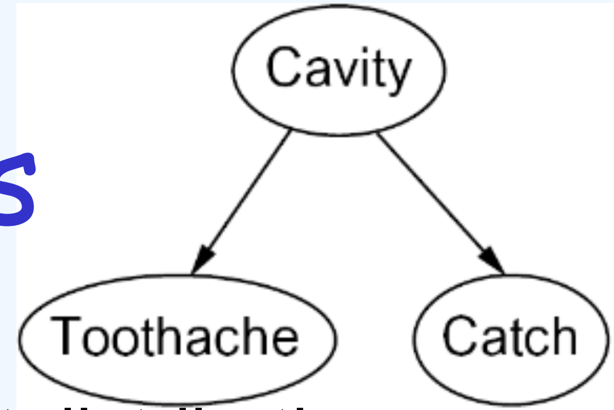
    $p(X|a_1,\ldots, a_n)$
  - CPT: conditional probability table
  - Description of a noisy "causal" process



$$p\left(X\middle|A_1\ldots A_n\right)$$

A Bayesian network = Topology (graph) + Local Conditional Probabilities

Probabilities in BNs



- Bayesian networks implicitly encode joint distributions
  - As a product of local conditional distributions

$$p(x_1, x_2, \cdots x_n) = \prod_{i=1}^{n} p(x_i \mid parents(X_i))$$

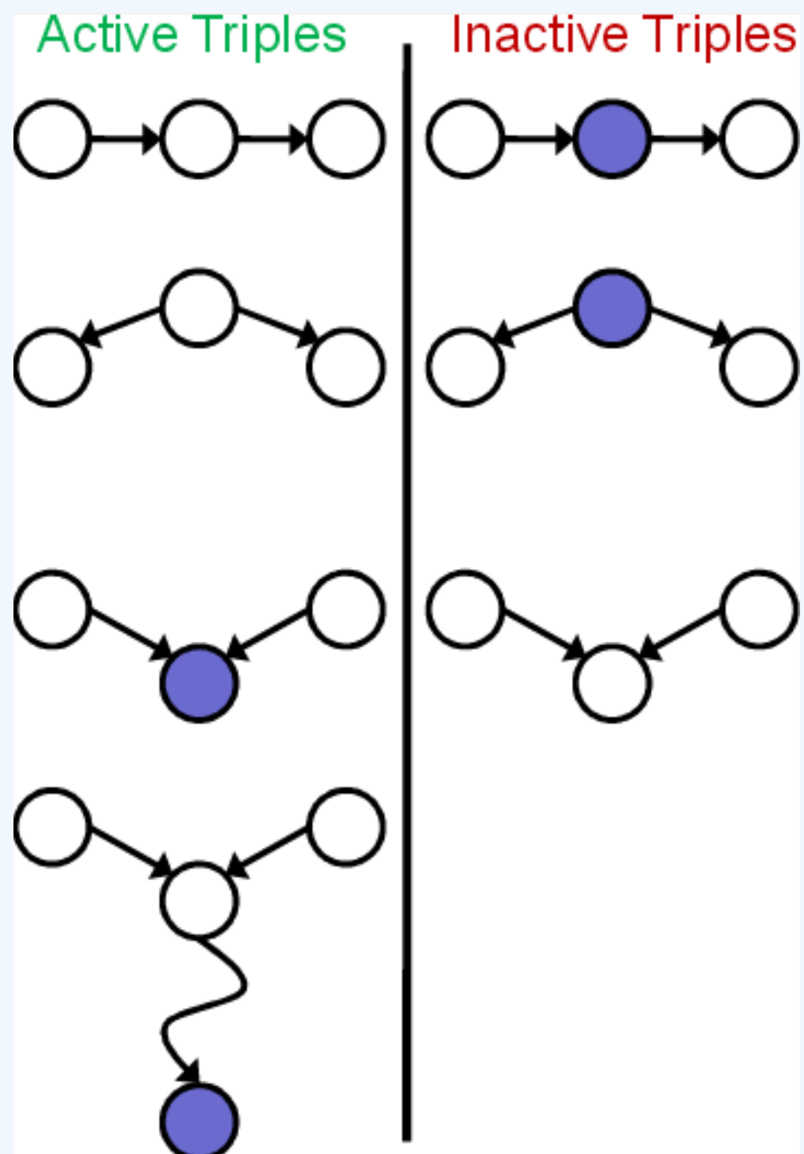
  - Example:

$$p(\text{+Cavity, +Catch, -Toothache})$$

- This lets us reconstruct any entry of the full joint
- Not every BN can represent every joint distribution
  - The topology enforces certain conditional independencies

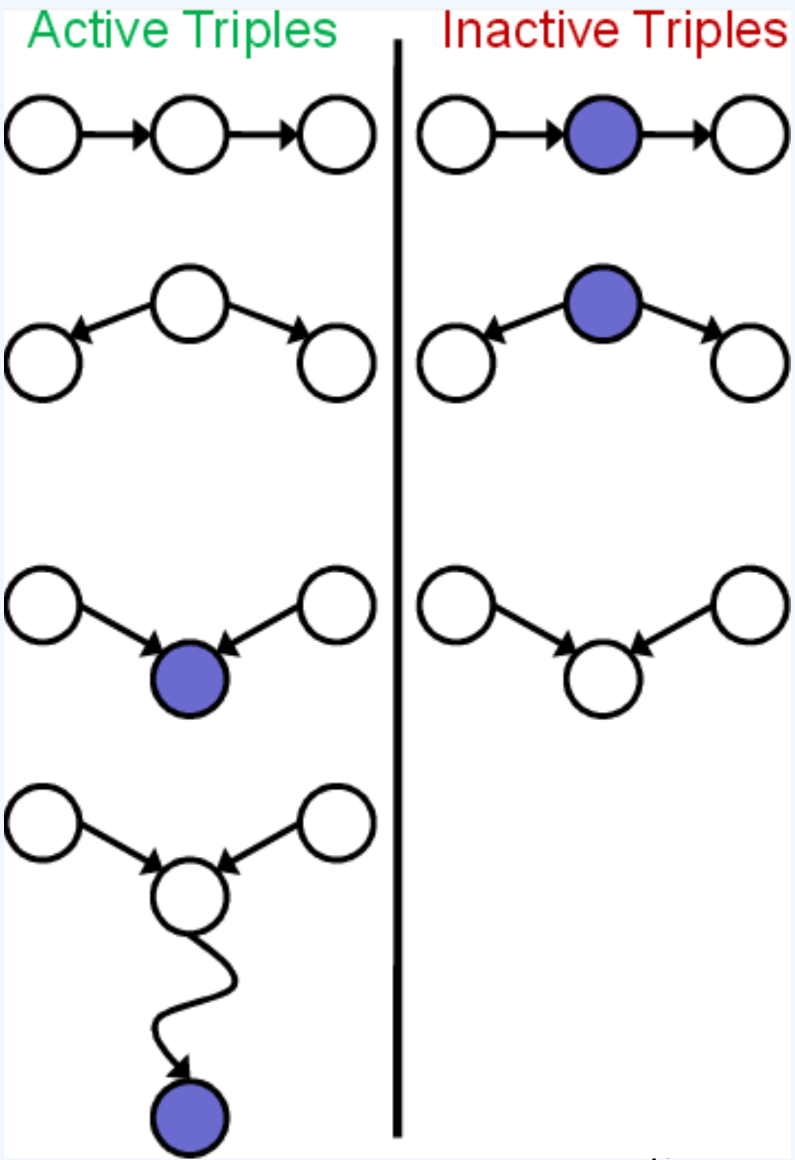
Reachability (D-Separation)

- Question: are X and Y conditionally independent given evidence vars {Z}?
  - Yes, if X and Y "separated" by Z
  - Look for active paths from X to Y
  - No active paths = independence!
- A path is active if each triple is active:
  - Causal chain $A \rightarrow B \rightarrow C$ where B is unobserved (either direction)
  - Common cause $A \leftarrow B \rightarrow C$ where B is unobserved
  - Common effect $A \rightarrow B \leftarrow C$ where B or one of its descendents is observed
- All it takes to block a path is a single inactive segment

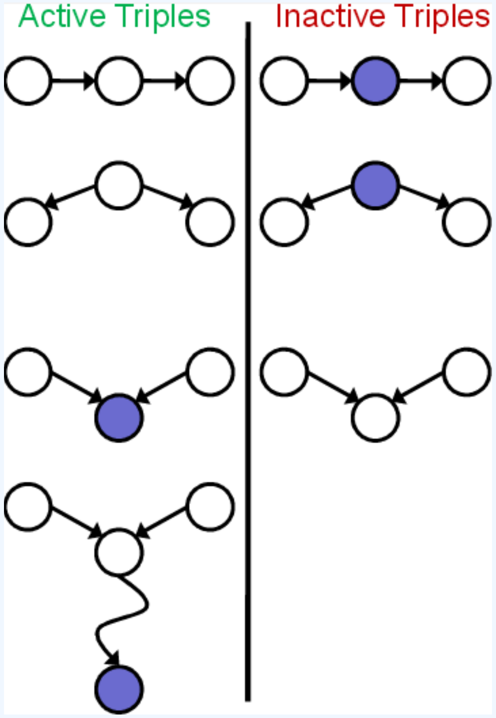


Checking conditional independence from BN graph

- Given random variables $Z_1,\ldots Z_p$, we are asked whether $X \perp Y | Z_1,\ldots Z_p$
- Step 1: shade $Z_1,\ldots Z_p$
- Step 2: for each undirected path from X to Y
  - if all triples are active, then X and Y are NOT conditionally independent
- If all paths have been checked and none of them is active, then $X \perp Y | Z_1,\ldots Z_p$



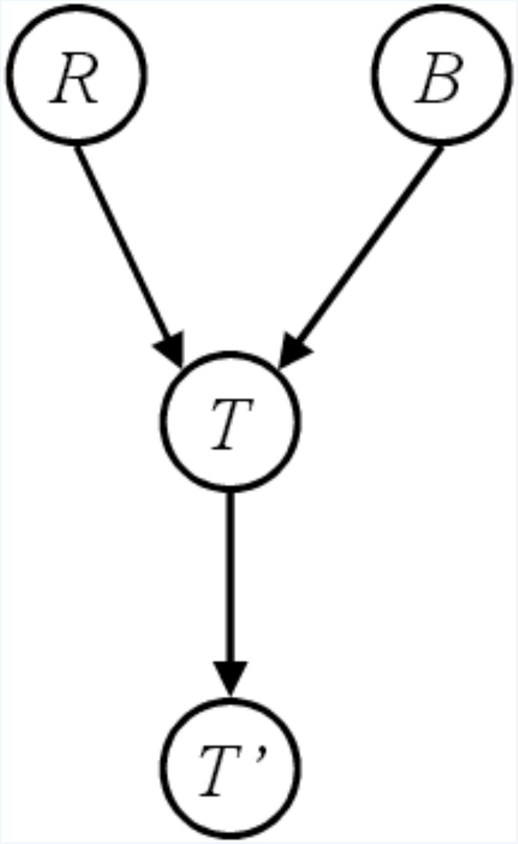
Example



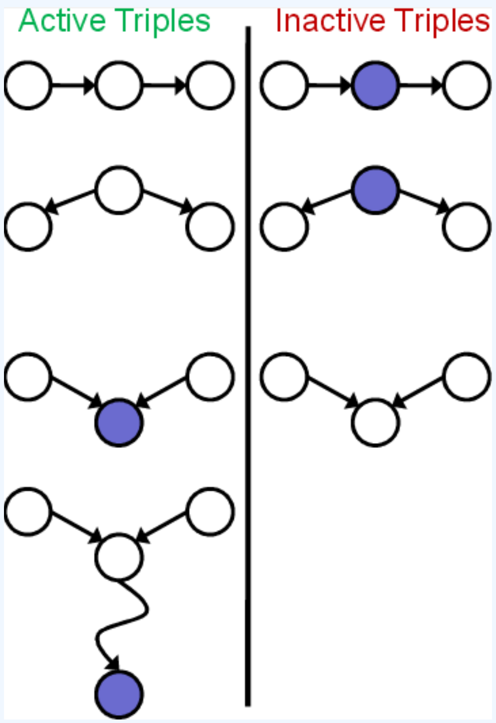
$R \perp B$ *Yes!*

$R \perp B | T$

$R \perp B | T'$



Example



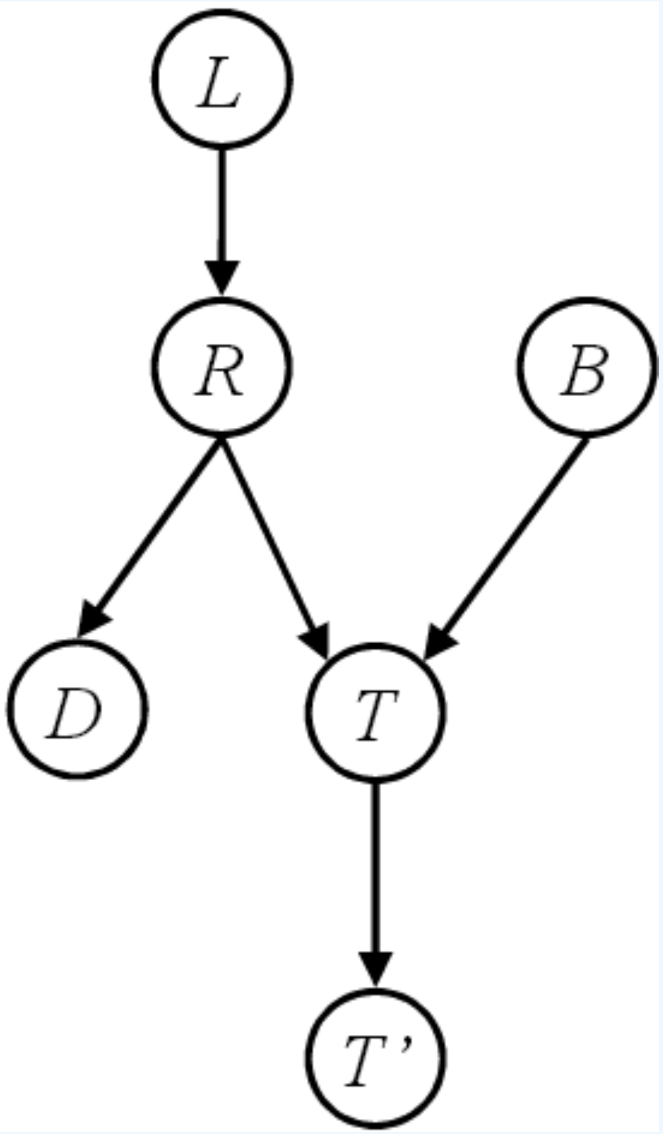
$L \perp T' | T$ Yes!

$L \perp B$ Yes!

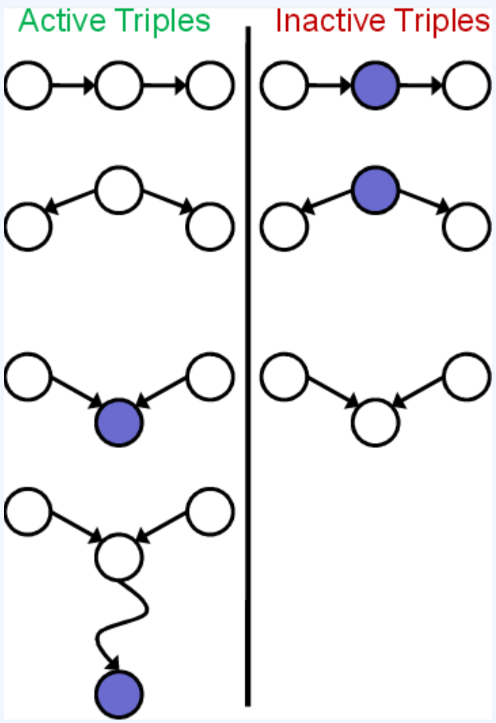
$L \perp B | T$

$L \perp B | T'$

$L \perp B | T, R$ Yes!



Example

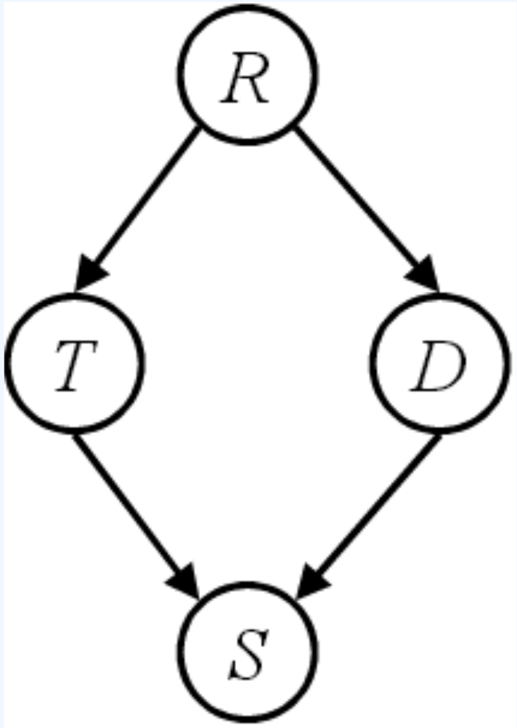


- Variables:
  - R: Raining
  - T: Traffic
  - D: Roof drips
  - S: I am sad
- Questions:

$$T \perp D$$

$$T \perp D \mid R$$ *Yes!*

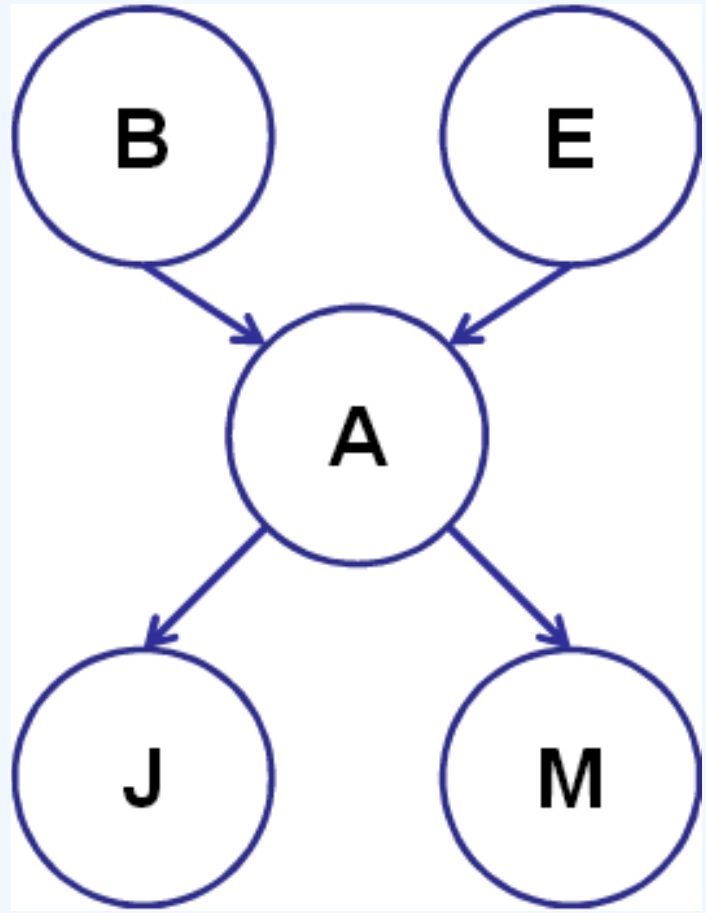
$$T \perp D \mid R, S$$



Today: Inference---variable elimination (dynamic programming)

Inference

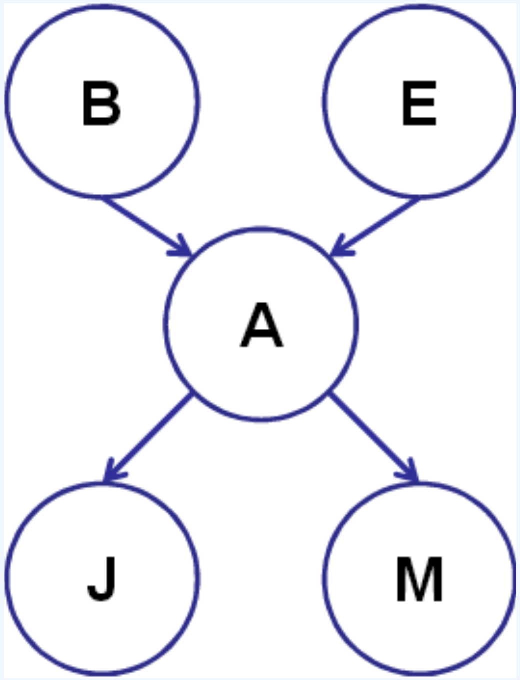
- Inference: calculating some useful quantity from a joint probability distribution
- Examples:
  - Posterior probability:
    $$p\left(Q \middle| E_1 = e_1, \cdots E_k = e_k\right)$$
  - Most likely explanation:
    $$\arg\max_q p\left(Q = q \middle| E_1 = e_1, \cdots\right)$$



Inference

- Given unlimited time, inference in BNs is easy
- Recipe:
  - State the marginal probabilities you need
  - Figure out ALL the atomic probabilities you need
  - Calculate and combine them
- Example:

$$p(+b \mid +j,+m) = \frac{p(+b,+j,+m)}{p(+j,+m)}$$



Example: Enumeration

- In this simple method, we only need the BN to synthesize the joint entries

$$p(+b,+j,+m) =$$

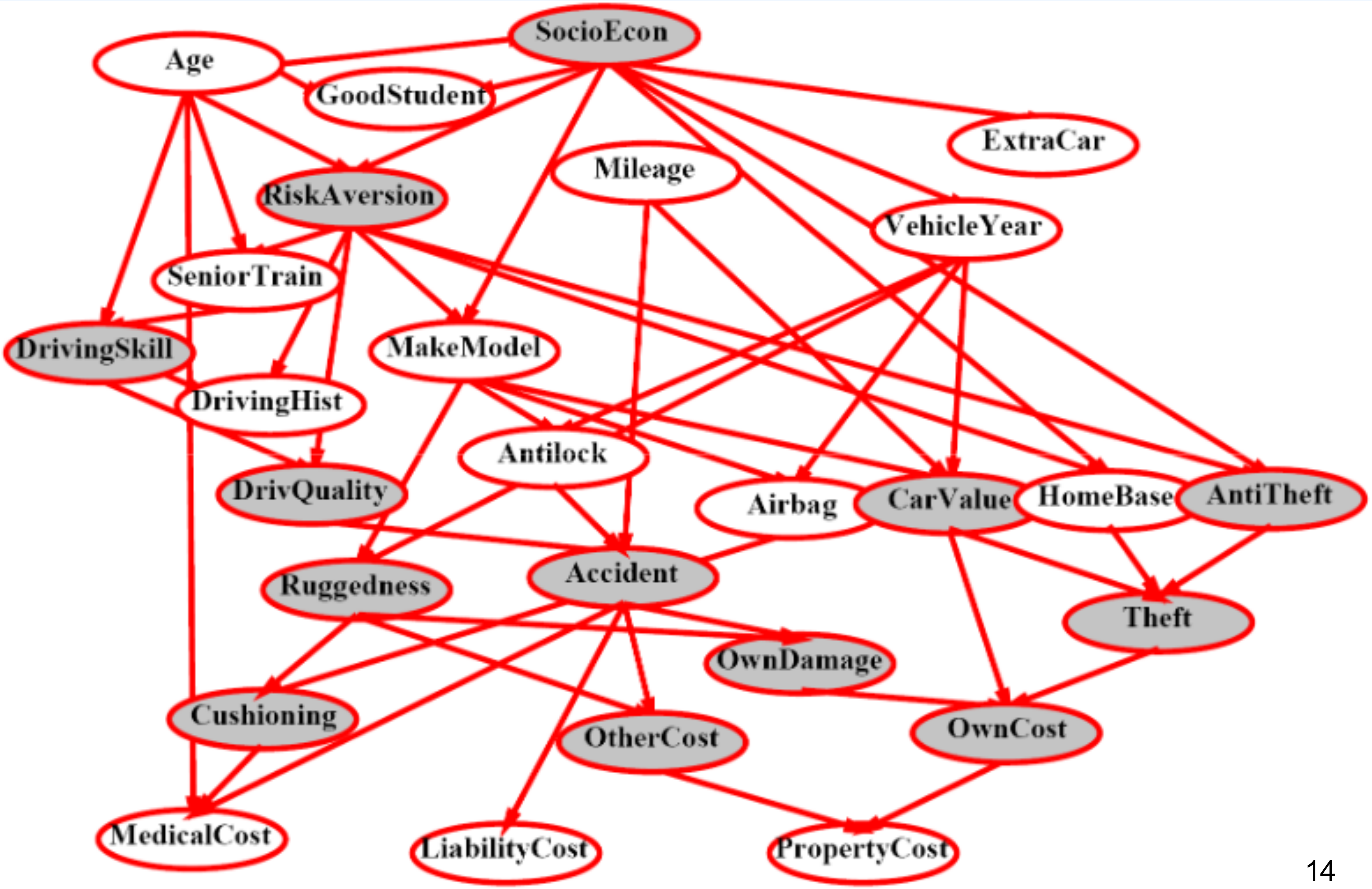
$$p(+b)p(+e)p(+a|+b,+e)p(+j|+a)p(+m|+a)+$$

$$p(+b)p(+e)p(-a|+b,+e)p(+j|-a)p(+m|-a)+$$

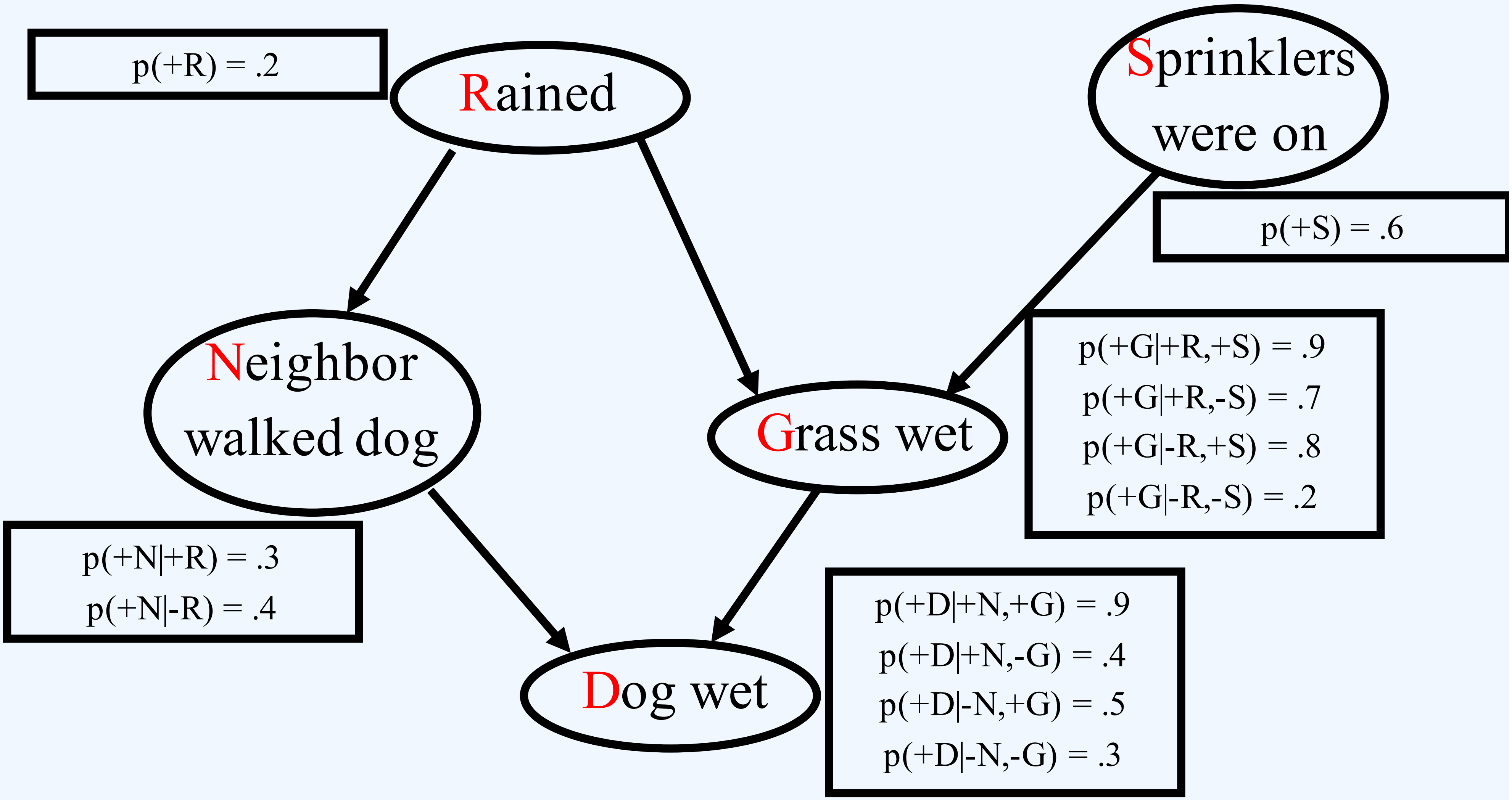
$$p(+b)p(-e)p(+a|+b,-e)p(+j|+a)p(+m|+a)+$$

$$p(+b)p(-e)p(-a|+b,-e)p(+j|-a)p(+m|-a)$$

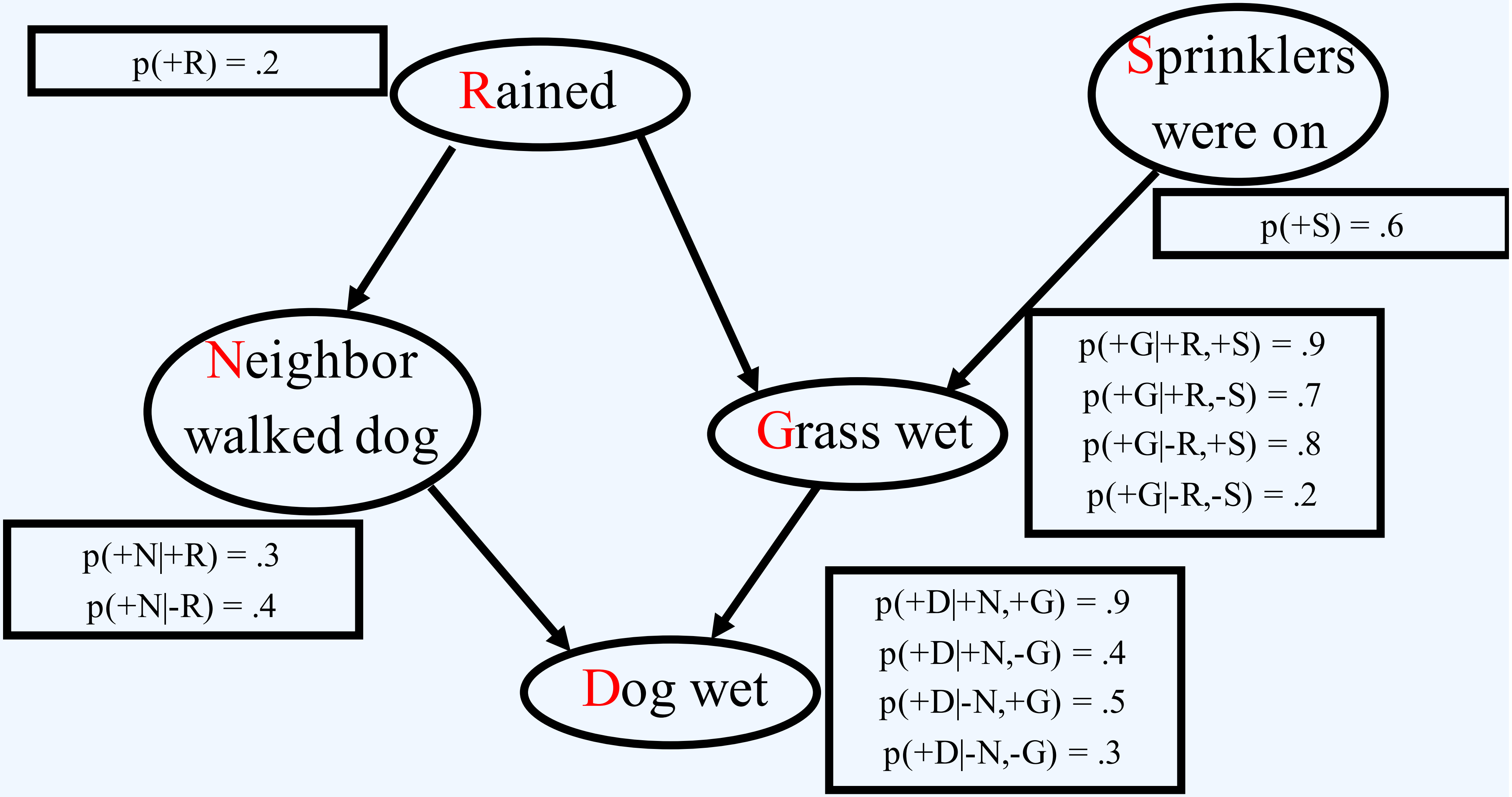
Inference by Enumeration?



More elaborate rain and sprinklers example

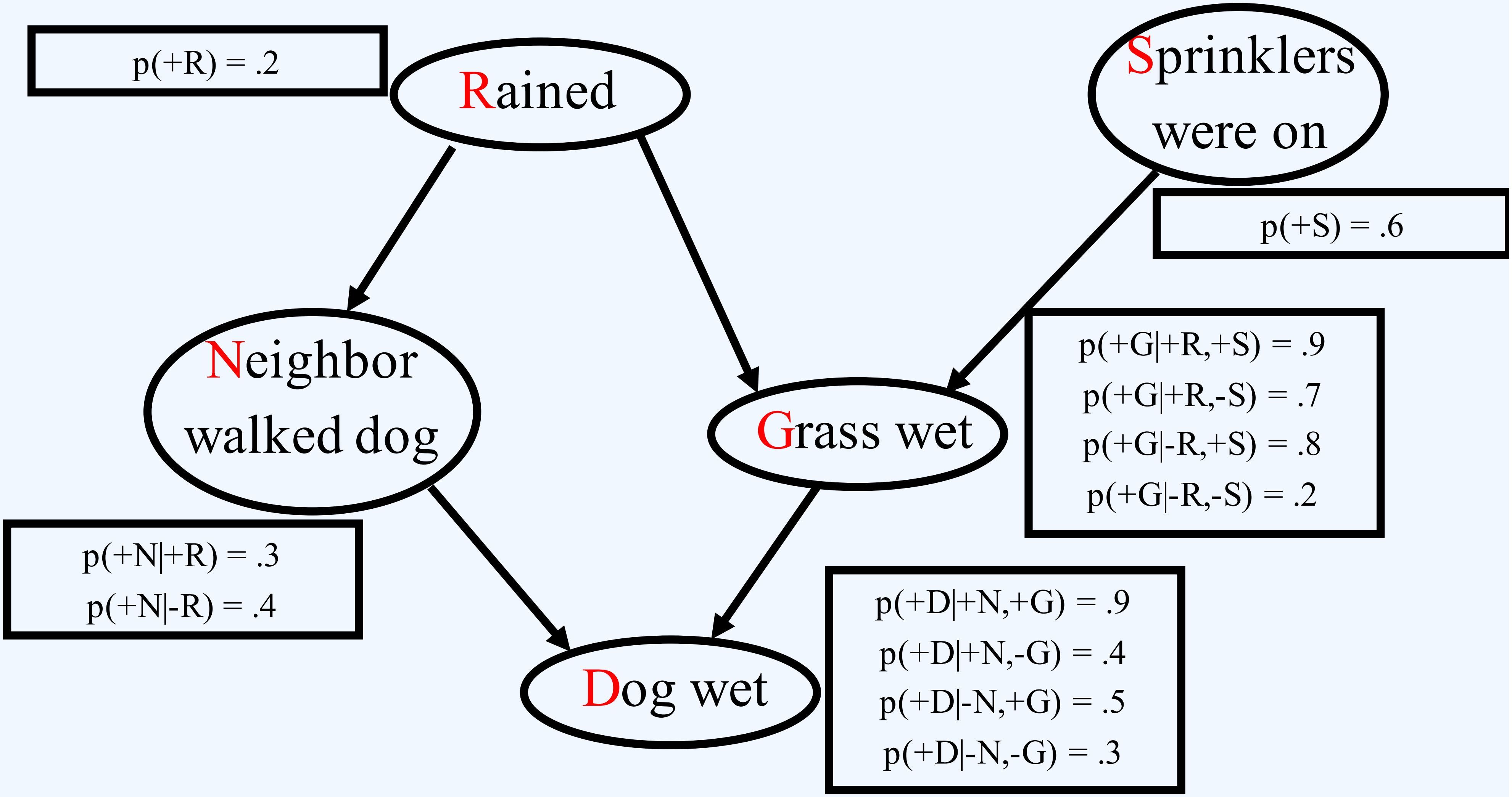


Inference



- Want to know: p(+R|+D) = p(+R,+D)/P(+D)
- Let's compute p(+R,+D)

Inference



- $p(+R,+D) = \Sigma_s \Sigma_g \Sigma_n \, p(+R)p(s)p(n|+R)p(g|+R,s)p(+D|n,g) =$
  $p(+R)\Sigma_s p(s)\Sigma_g p(g|+R,s)\Sigma_n \, p(n|+R)p(+D|n,g)$

The formula

$p(+R,+D) =$

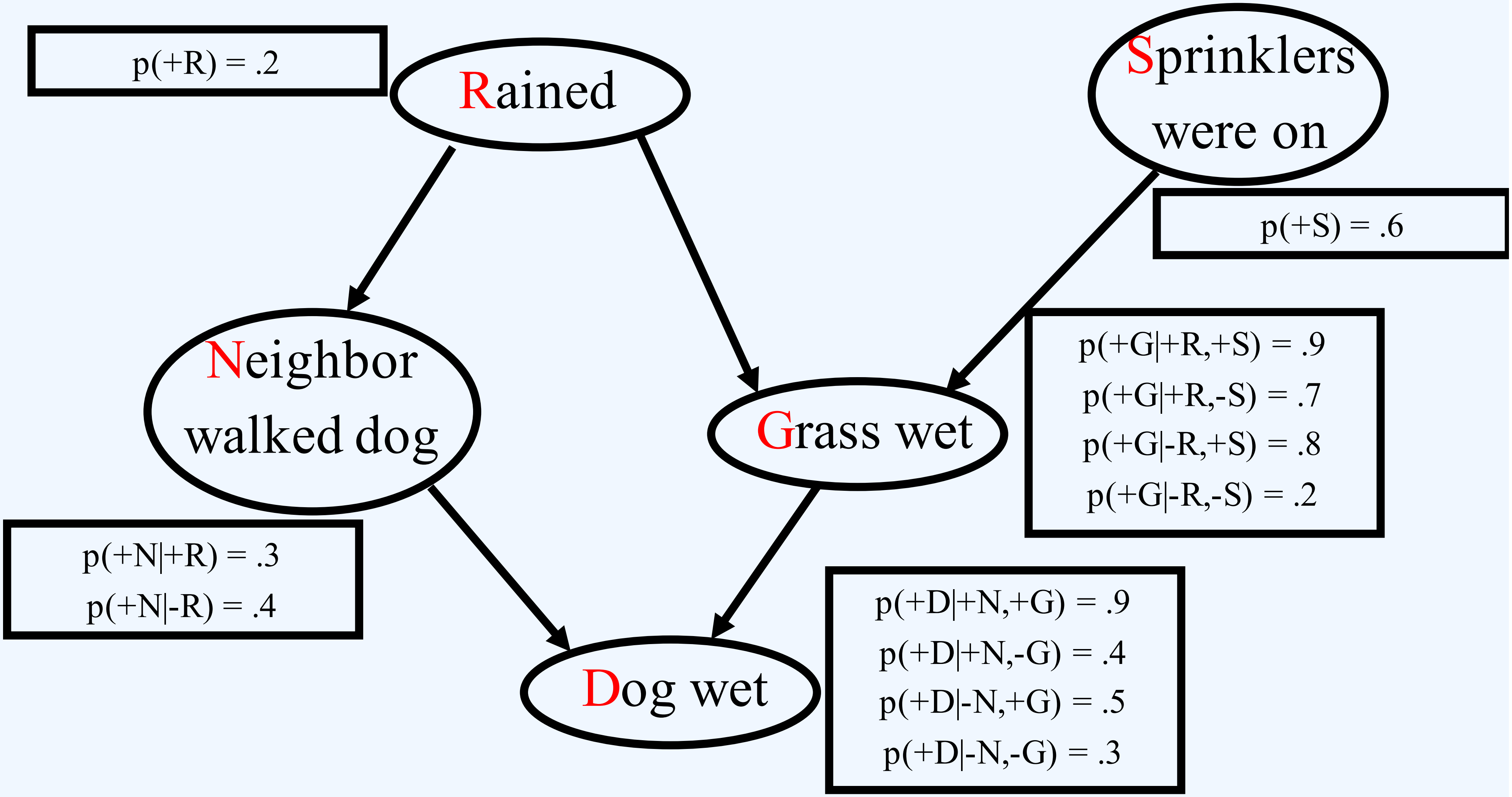
$$p(+R)\Sigma_s p(s)\ \Sigma_g p(g|+R,s)\ \Sigma_n\ p(n|+R)p(+D|n,g)$$

$f_2(s,g)$

$f_1(s)$

- Order: s>g>n
- is what we want to compute
- only involves s
- only involves s, g

Variable elimination



- From the factor Σ_n p(n|+R)p(+D|n,g) we sum out n to obtain a factor only depending on g
- f_2(s, g) happens to be insensitive to s, lucky!
- f_2(s,+G) = [Σ_n p(n|+R)p(+D|n,+G)] = p(+N|+R)P(+D|+N,+G) + p(-N|+R)p(+D|-N,+G) = .3*.9+.7*.5 = .62
- f_2(s,-G) = [Σ_n p(n|+R)p(+D|n,-G)] = p(+N|+R)p(+D|+N,-G) + p(-N|+R)p(+D|-N,-G) = .3*.4+.7*.3 = .33

Calculating $f_1(s)$

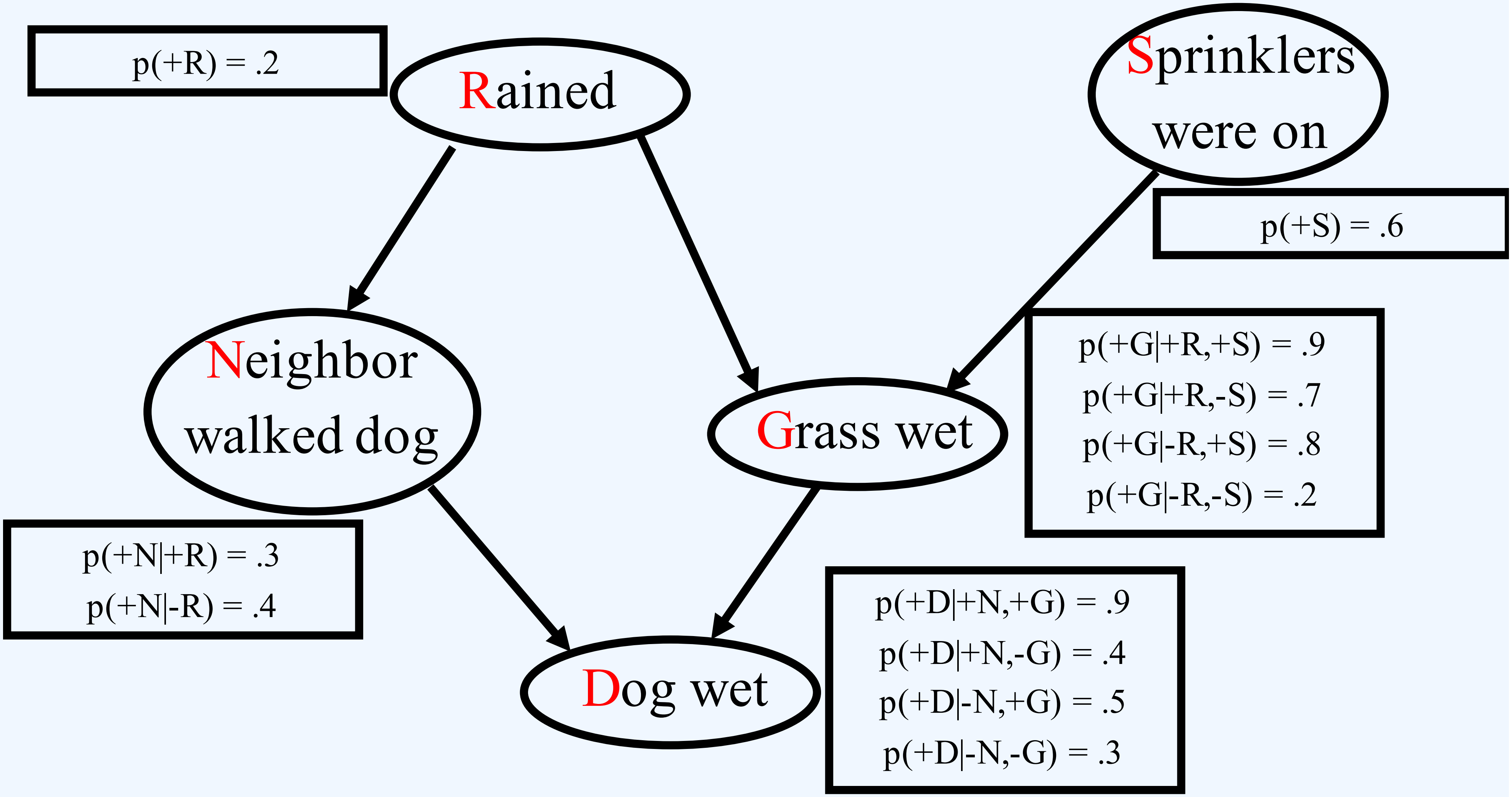
- $f_1(s) = p(+G|+R,s)\, f_2(s,+G) + p(-G|+R,s)\, f_2(s,-G)$
- $f_1(+S) = p(+G|+R,+S)\, f_2(+S,+G) + p(-G|+R, +S)\, f_2(+S,-G) =$ $0.9*0.62+0.1*0.33=0.591$
- $f_1(-S) = p(+G|+R,-S)\, f_2(-S,+G) + p(-G|+R, -S)\, f_2(-S,-G) = 0.7*0.62+0.3*0.33=0.533$

Calculating p(+R,+D)

- p(+R,+D)= p(+R)p(+S) f_1(+S) + p(+R)p(-S) f_1(-S)

=0.2*(0.6*0.591+0.4*0.533)=0.11356

Elimination order matters



- $p(+R,+D) = \Sigma_n \Sigma_s \Sigma_g\ p(+r)p(s)p(n|+R)p(g|+r,s)p(+D|n,g) = p(+R)\Sigma_n p(n|+R)\Sigma_s p(s)\Sigma_g\ p(g|+R,s)p(+D|n,g)$
- Last factor will depend on two variables in this case!

General method for variable elimination

- Compute a marginal probability p($x_1,\ldots,x_p$) in a Bayesian network
  - Let $Y_1,\ldots,Y_k$ denote the remaining variables
  - Step 1: fix an order over the Y's (wlog $Y_1 > \ldots > Y_k$)
  - Step 2: rewrite the summation as

sth only involving X's	Σ_{y_1}	sth only involving Y_1 and X's	Σ_{y_2}	sth only involving Y_1, Y_2 and X's	$\ldots\Sigma_{y_{k-1}}$	sth only involving Y_1, $Y_2,\ldots,Y_{k-1}$ and X's	Σ_{y_k} anything

  - Step 3: variable elimination from right to left