CPS 170: Artificial Intelligence

http://www.cs.duke.edu/courses/spring09/cps170/

Bayesian networks

Instructor: Vincent Conitzer

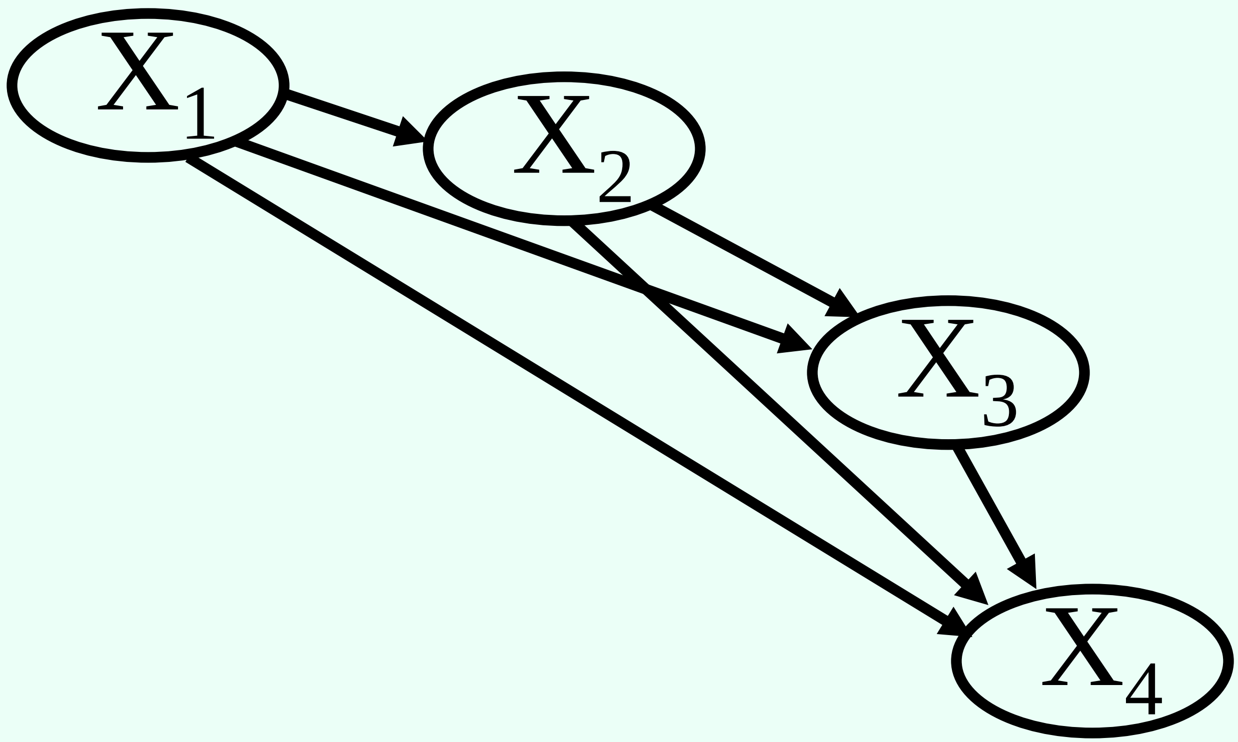
Specifying probability distributions

- Specifying a probability for every atomic event is impractical
- $P(X_1, \ldots, X_n)$ would need to be specified for **every** combination $x_1, \ldots, x_n$ of values for $X_1, \ldots, X_n$
  - If there are k possible values per variable…
  - … we need to specify $k^n - 1$ probabilities!
- We have already seen it can be easier to specify probability distributions by using (conditional) independence
- Bayesian networks allow us
  - to specify any distribution,
  - to specify such distributions concisely if there is (conditional) independence, in a natural way

A general approach to specifying probability distributions

- Say the variables are $X_1, \ldots, X_n$
- $P(X_1, \ldots, X_n) = P(X_1)P(X_2|X_1)P(X_3|X_1,X_2)\ldots P(X_n|X_1, \ldots, X_{n-1})$
- or:
- $P(X_1, \ldots, X_n) = P(X_n)P(X_{n-1}|X_n)P(X_{n-2}|X_n,X_{n-1})\ldots P(X_1|X_n, \ldots, X_2)$
- Can specify every component
  - For **every** combination of values for the variables on the right of |, specify the probability over the values for the variable on the left
- If every variable can take k values,
- $P(X_i|X_1, \ldots, X_{i-1})$ requires $(k-1)k^{i-1}$ values
- $\Sigma_{i=\{1,..,n\}}(k-1)k^{i-1} = \Sigma_{i=\{1,..,n\}}k^i-k^{i-1} = k^n - 1$
- Same as specifying probabilities of all atomic events – of course, because we can specify any distribution!

Graphically representing influences

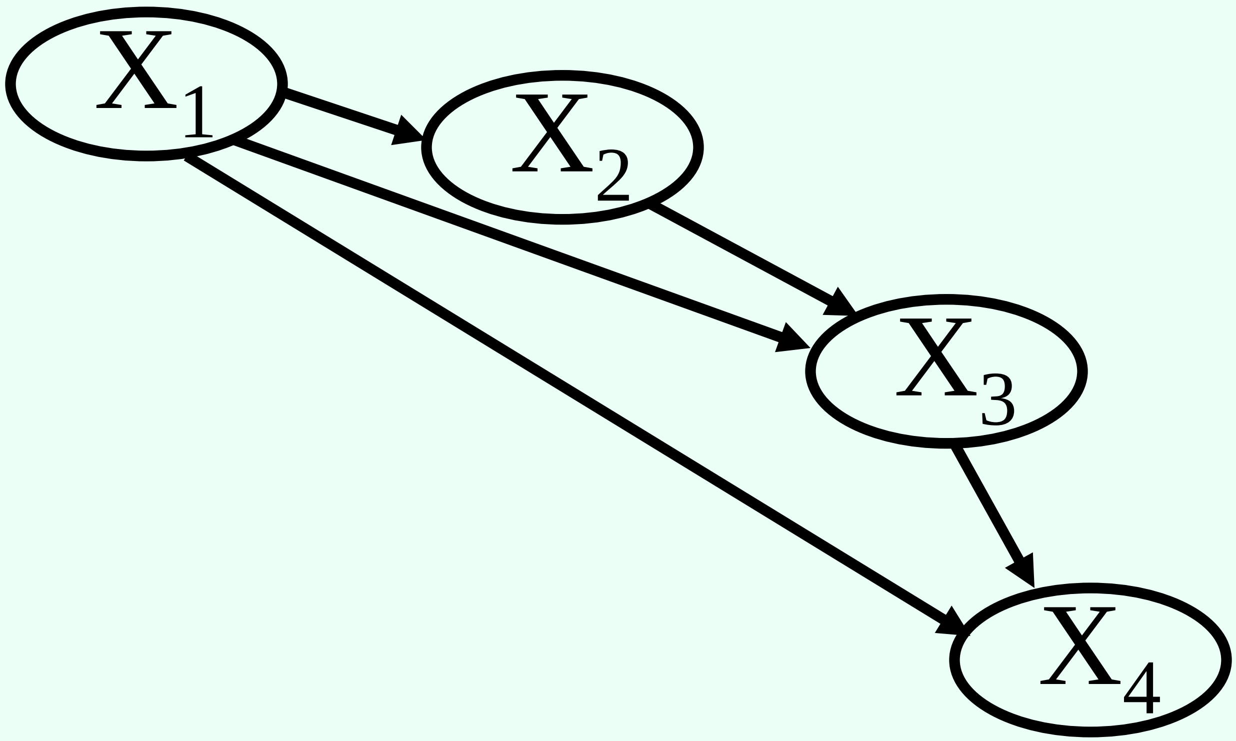


Conditional independence to the rescue!

- Problem: $P(X_i|X_1, \ldots, X_{i-1})$ requires us to specify too many values
- Suppose $X_1, \ldots, X_{i-1}$ partition into two subsets, *S* and *T*, so that X_i is conditionally independent from *T* given *S*
- $P(X_i|X_1, \ldots, X_{i-1}) = P(X_i|S, T) = P(X_i|S)$
- Requires only $(k-1)k^{|S|}$ values instead of $(k-1)k^{i-1}$ values

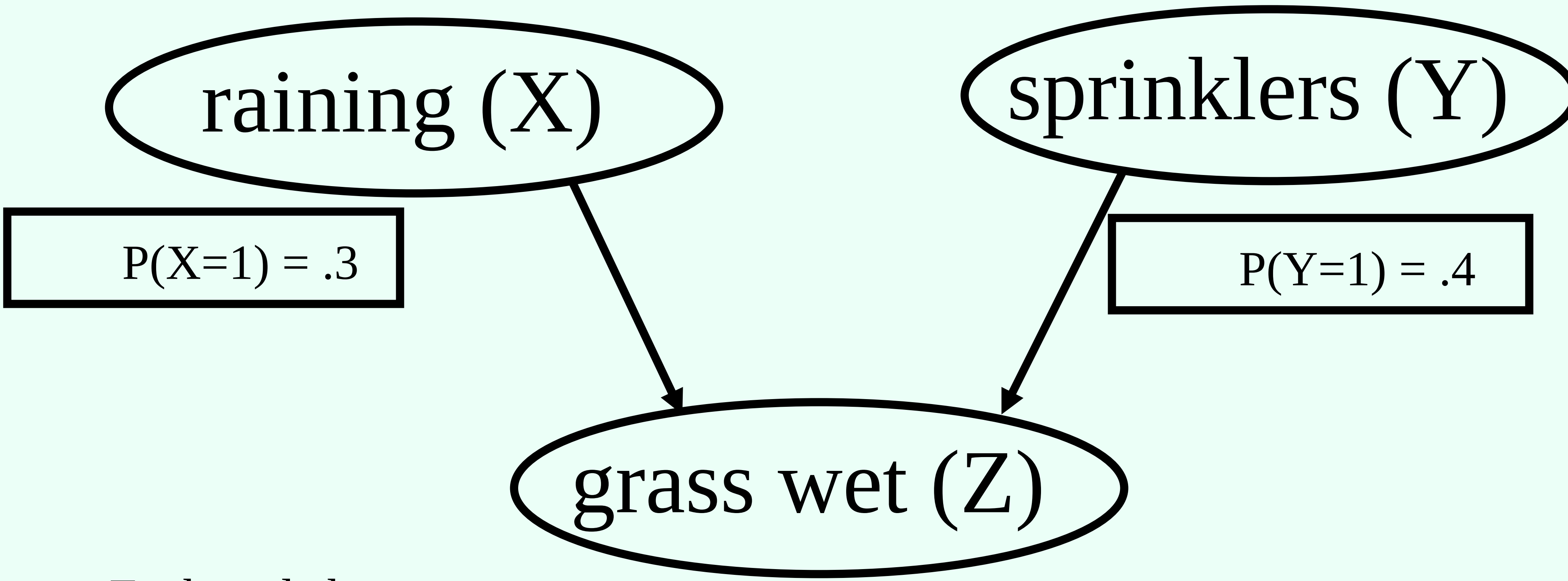
Graphically representing influences

- … if X_4 is conditionally independent from X_2 given X_1 and X_3



Rain and sprinklers example

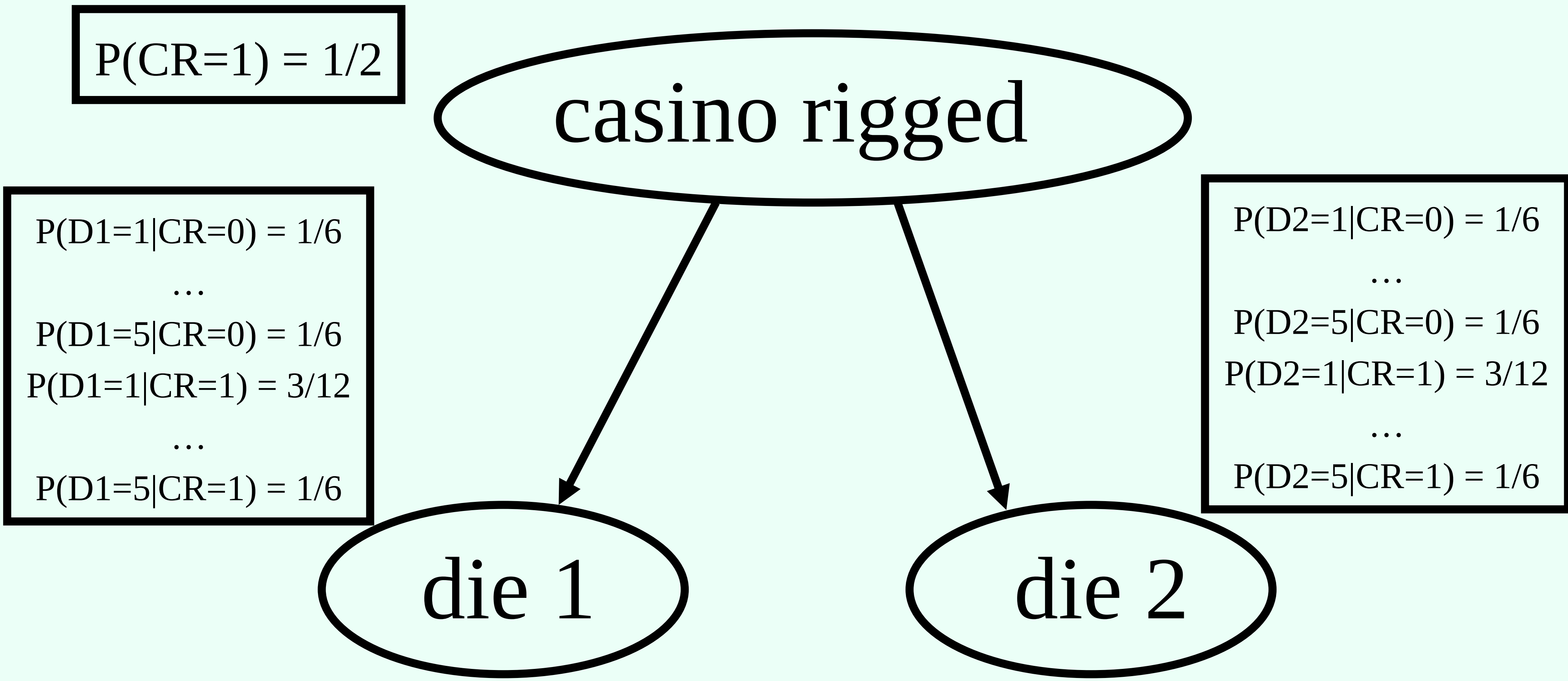
sprinklers is independent of raining, so no edge between them



Each node has a conditional probability table (CPT)

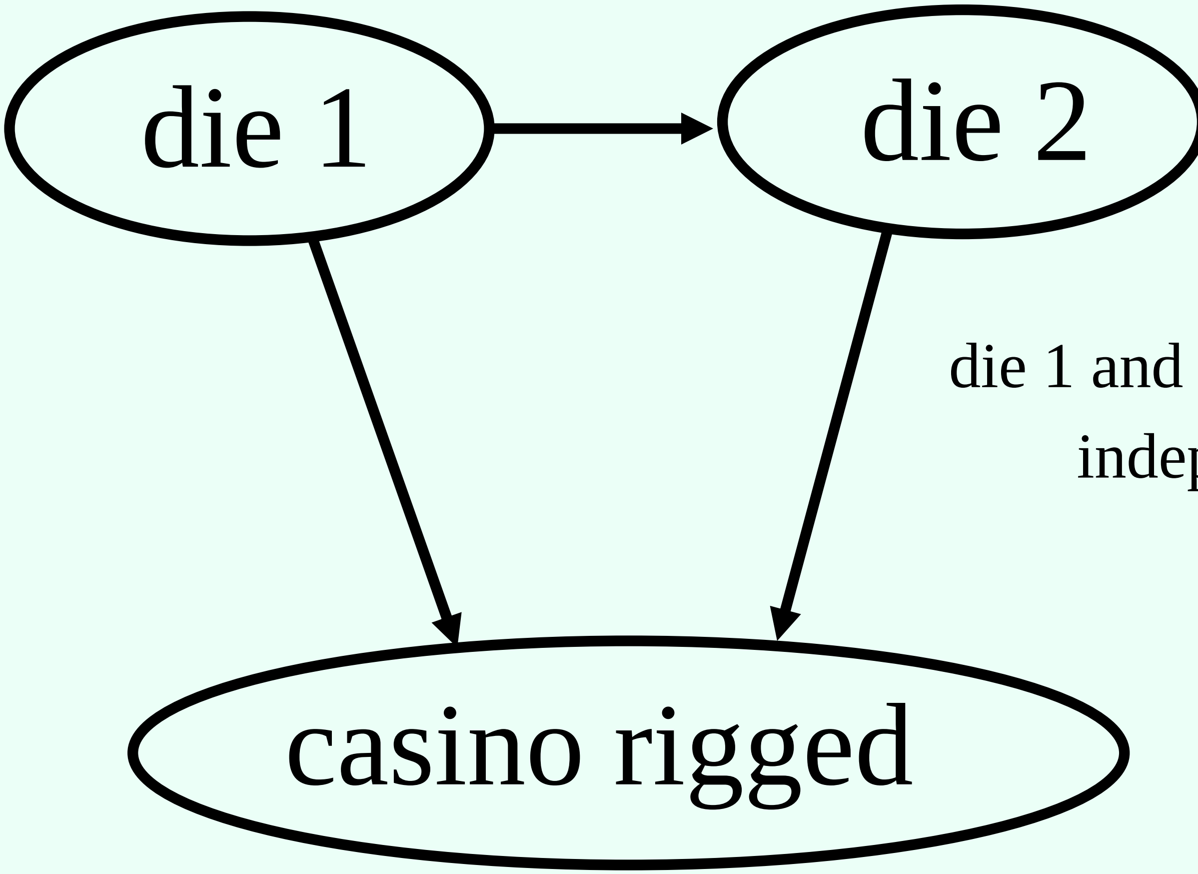
P(Z=1 | X=0, Y=0) = .1
P(Z=1 | X=0, Y=1) = .8
P(Z=1 | X=1, Y=0) = .7
P(Z=1 | X=1, Y=1) = .9

Rigged casino example



die 2 is conditionally independent of die 1 given casino rigged, so no edge between them

Rigged casino example with poorly chosen order

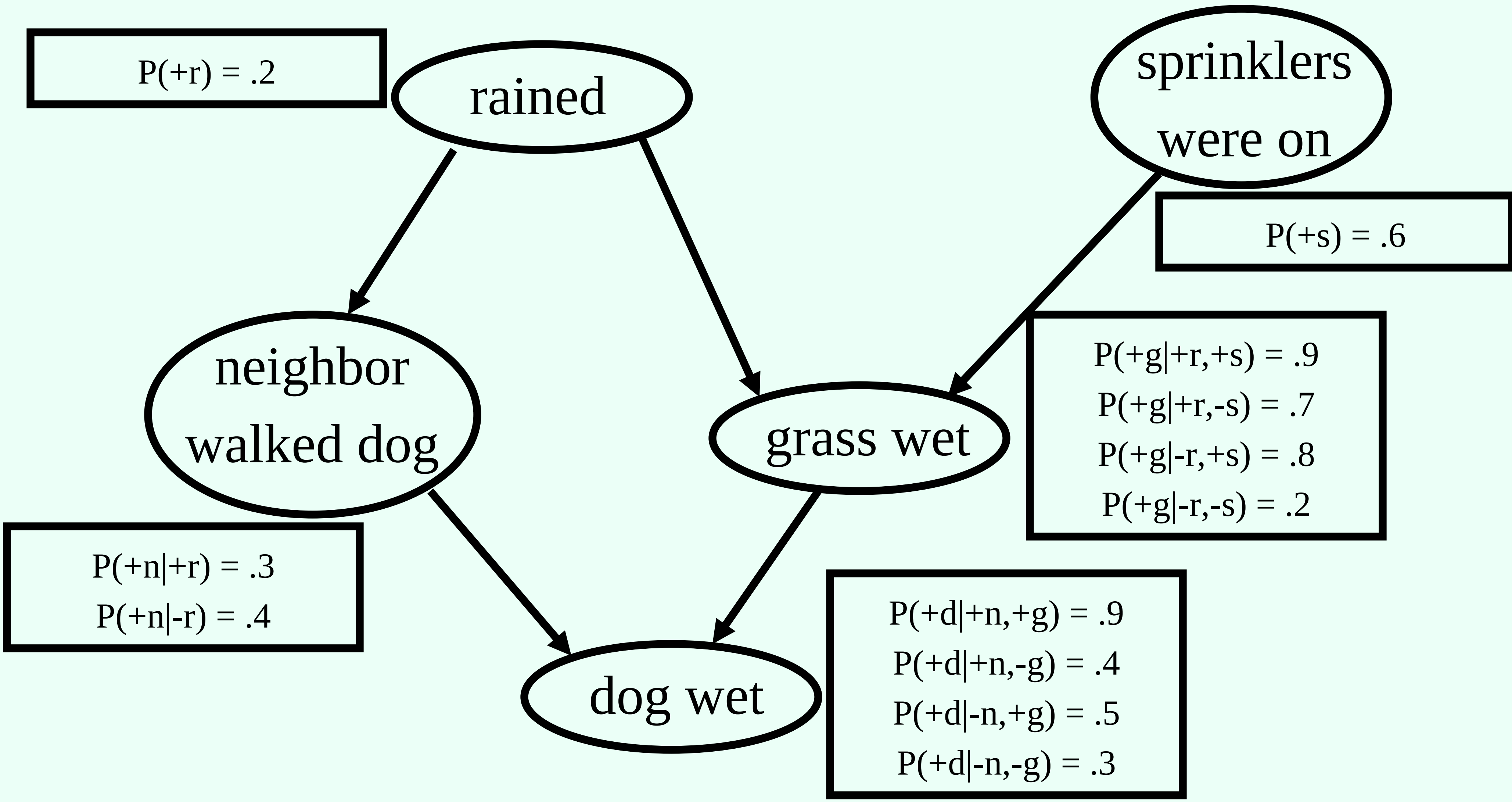


die 1 and die 2 are not independent

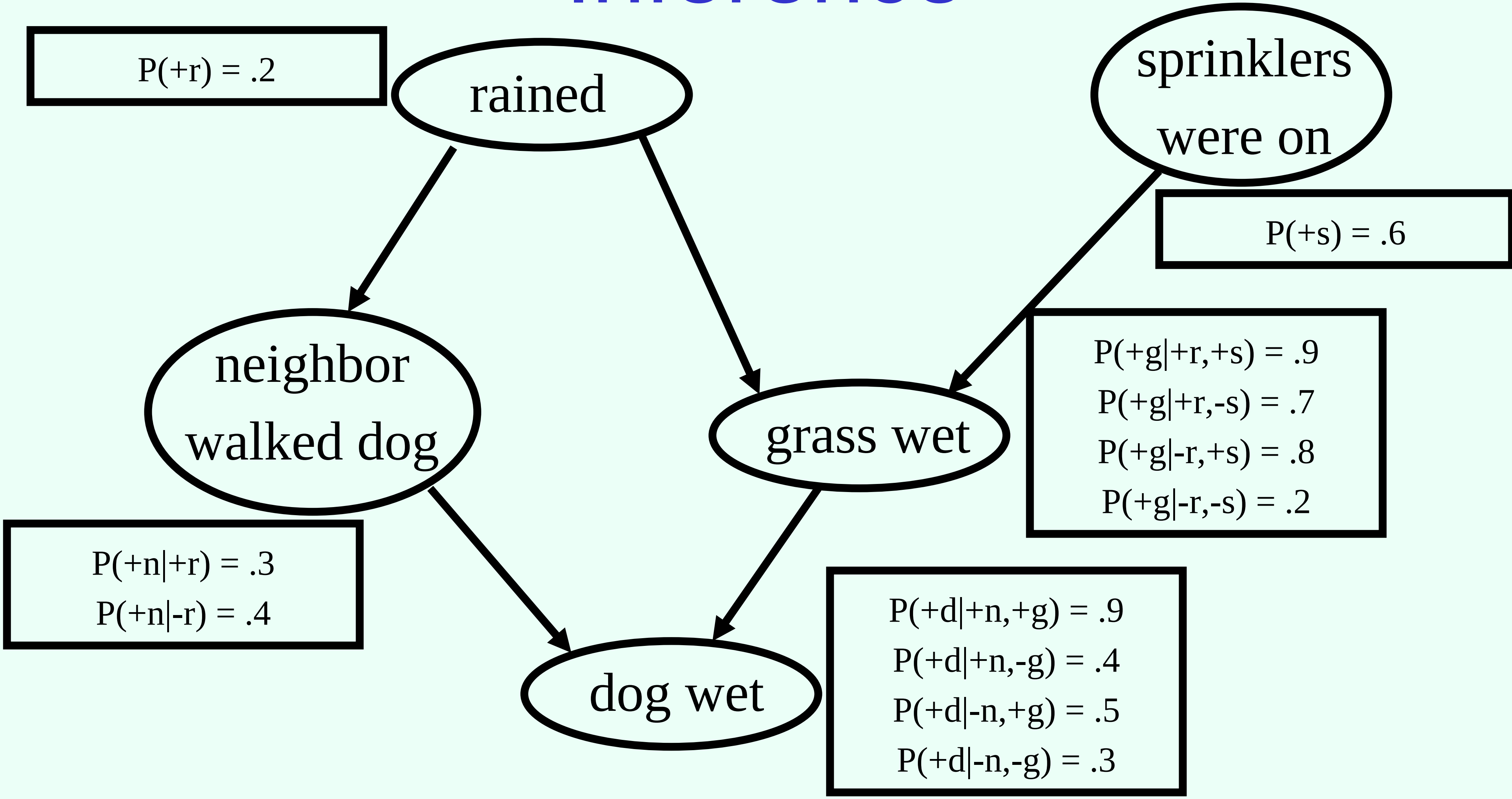
both the dice have relevant information for whether the casino is rigged

need 36 probabilities here!

More elaborate rain and sprinklers example

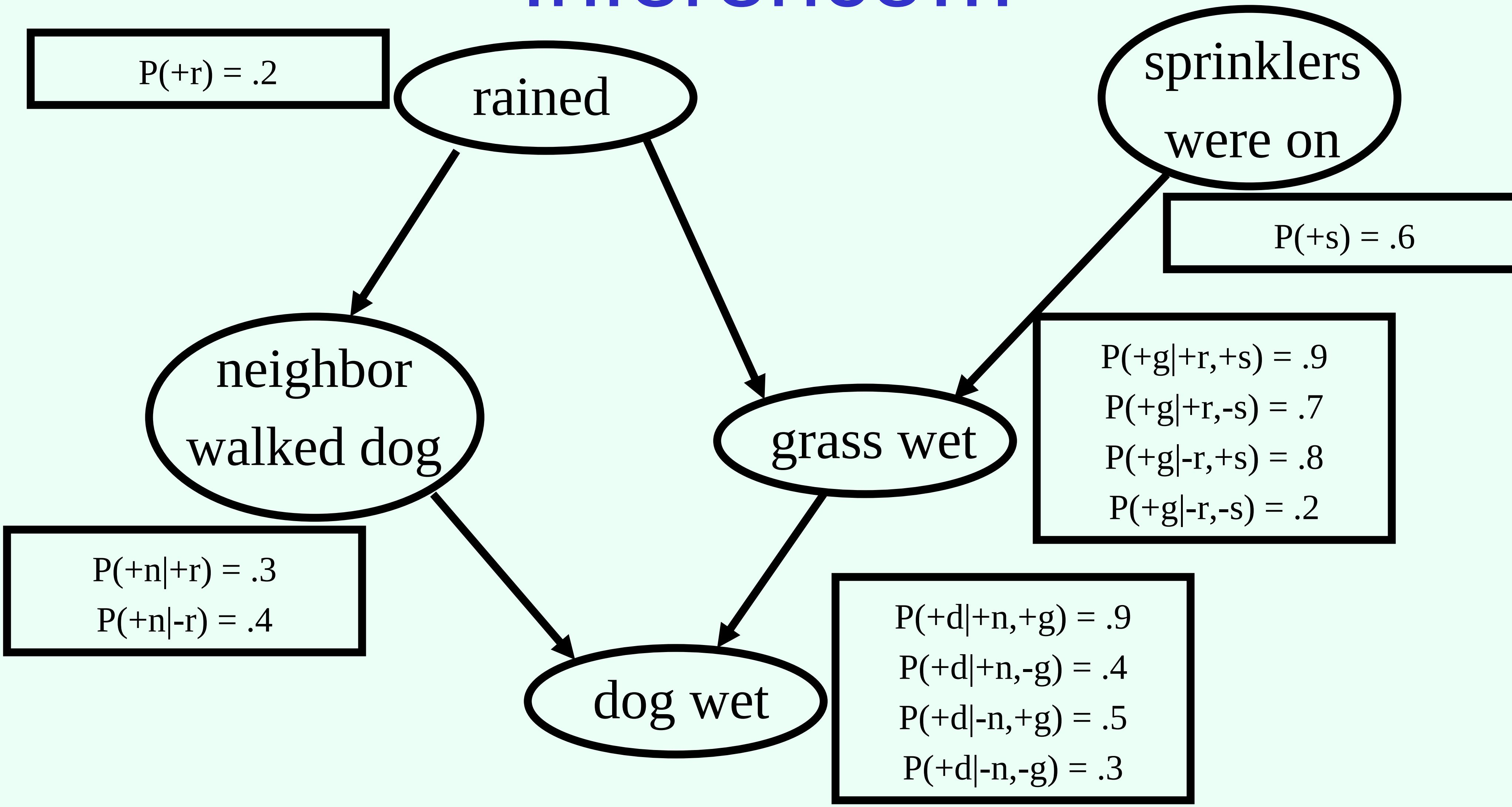


Inference



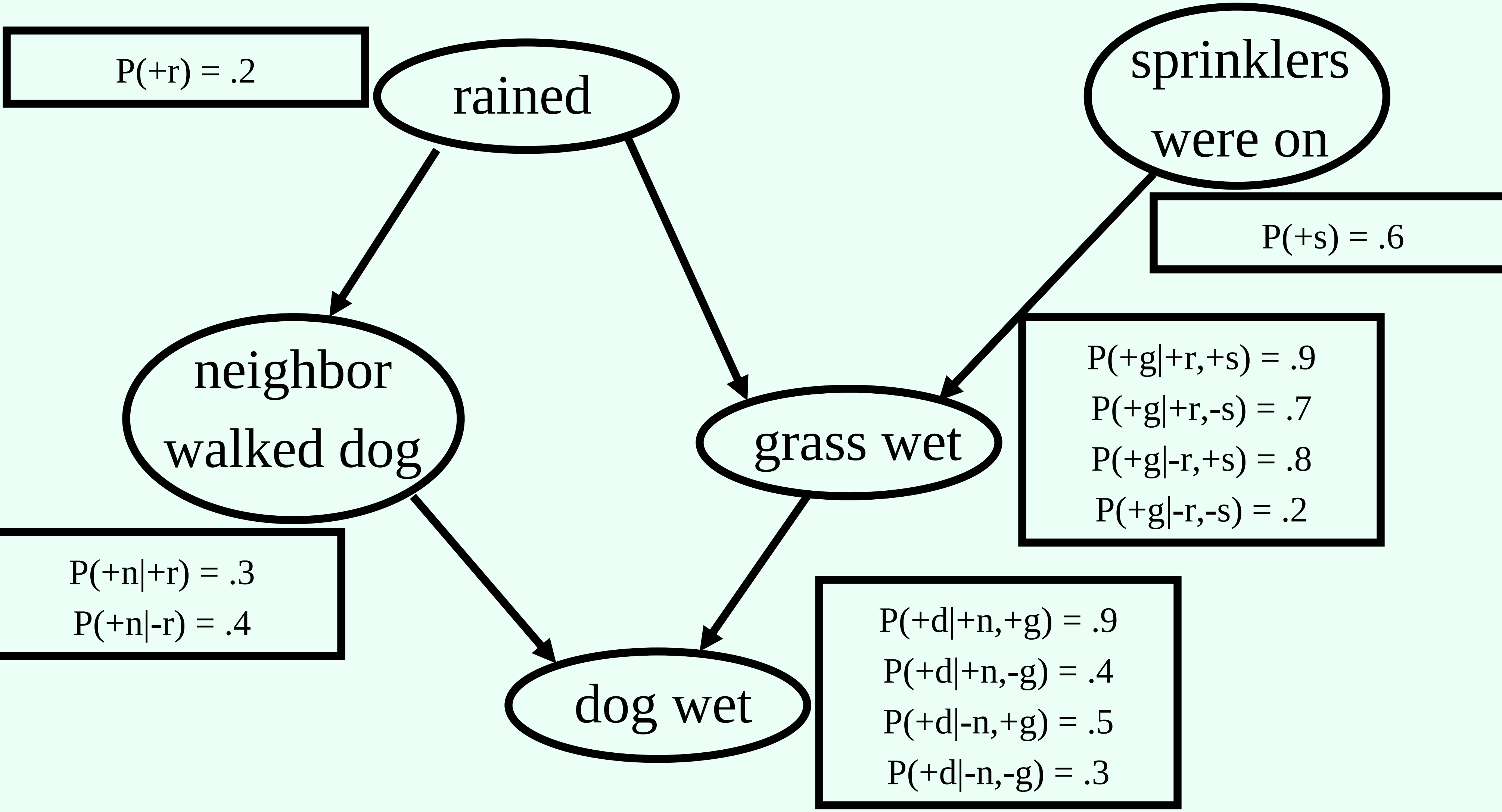
- Want to know: $P(+r|+d) = P(+r,+d)/P(+d)$
- Let's compute $P(+r,+d)$

Inference…



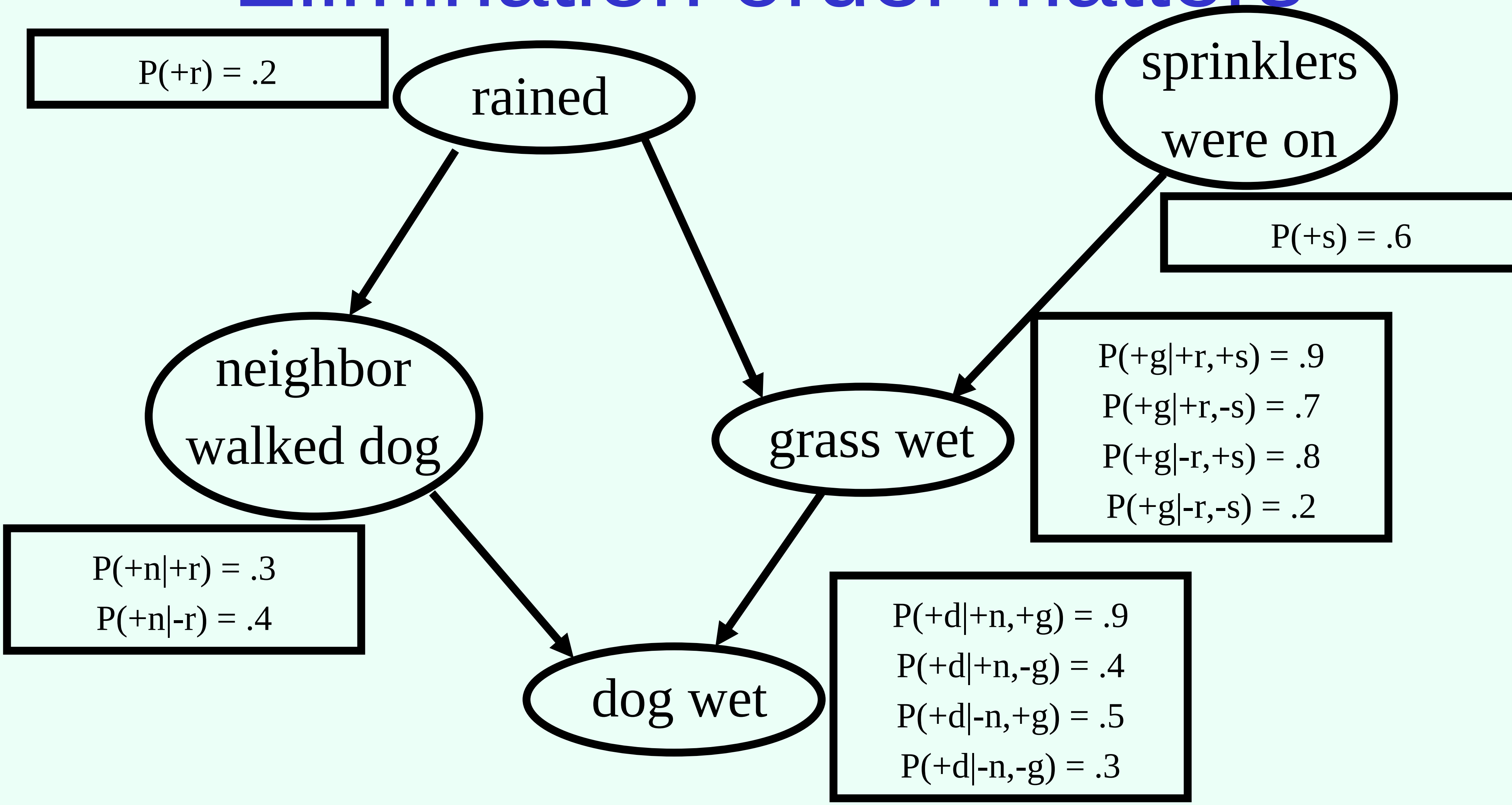
- $P(+r,+d) = \Sigma_s \Sigma_g \Sigma_n P(+r)P(s)P(n|+r)P(g|+r,s)P(+d|n,g) = P(+r)\Sigma_s P(s)\Sigma_g P(g|+r,s)\Sigma_n P(n|+r)P(+d|n,g)$

Variable elimination



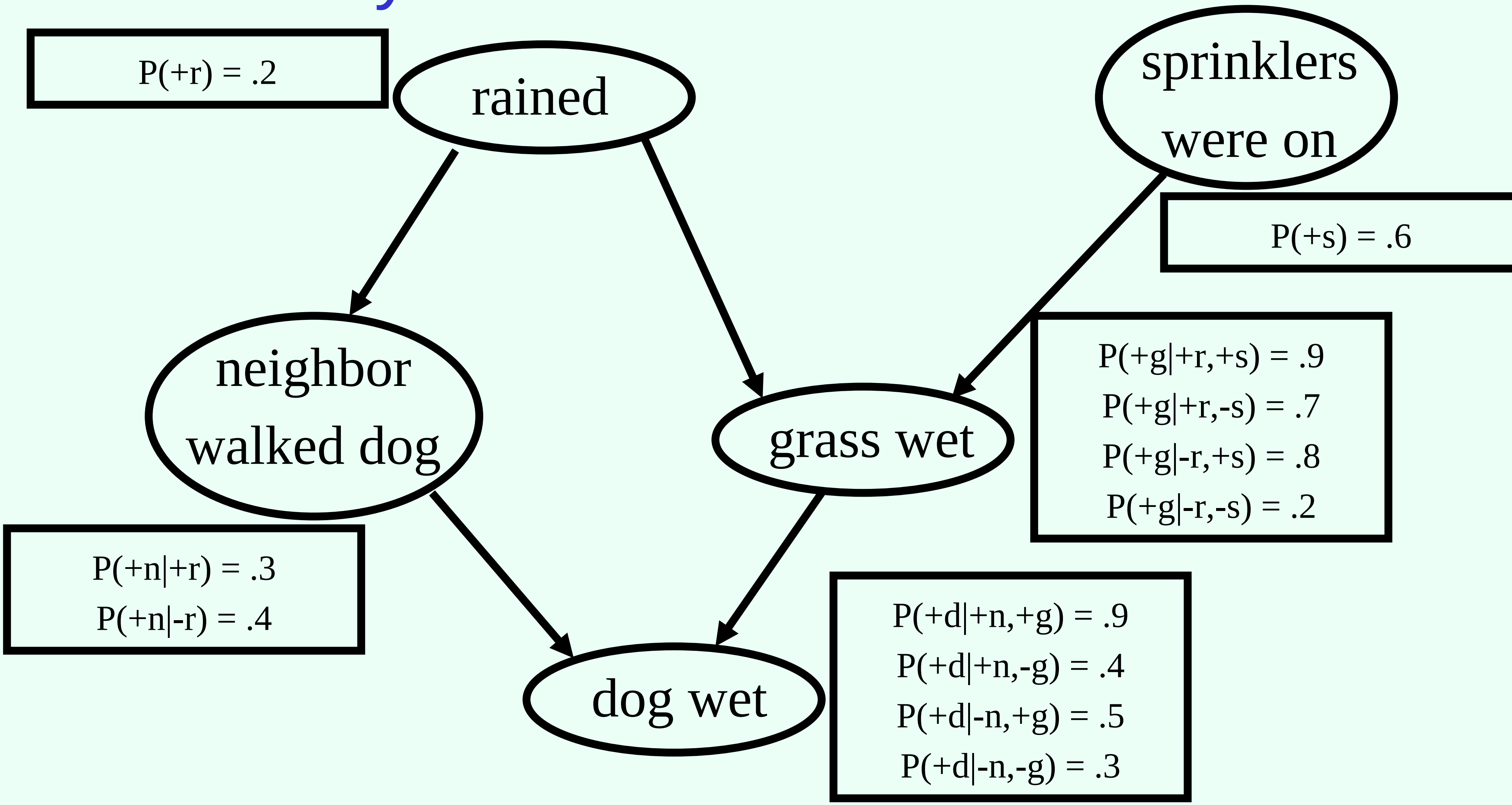
- From the factor Σ_n P(n|+r)P(+d|n,g) we sum out n to obtain a factor only depending on g
- [Σ_n P(n|+r)P(+d|n,+g)] = P(+n|+r)P(+d|+n,+g) + P(-n|+r)P(+d|-n,+g) = .3*.9+.7*.5 = .62
- [Σ_n P(n|+r)P(+d|n,-g)] = P(+n|+r)P(+d|+n,-g) + P(-n|+r)P(+d|-n,-g) = .3*.4+.7*.3 = .33
- Continuing to the left, g will be summed out next, etc. (continued on board)

Elimination order matters



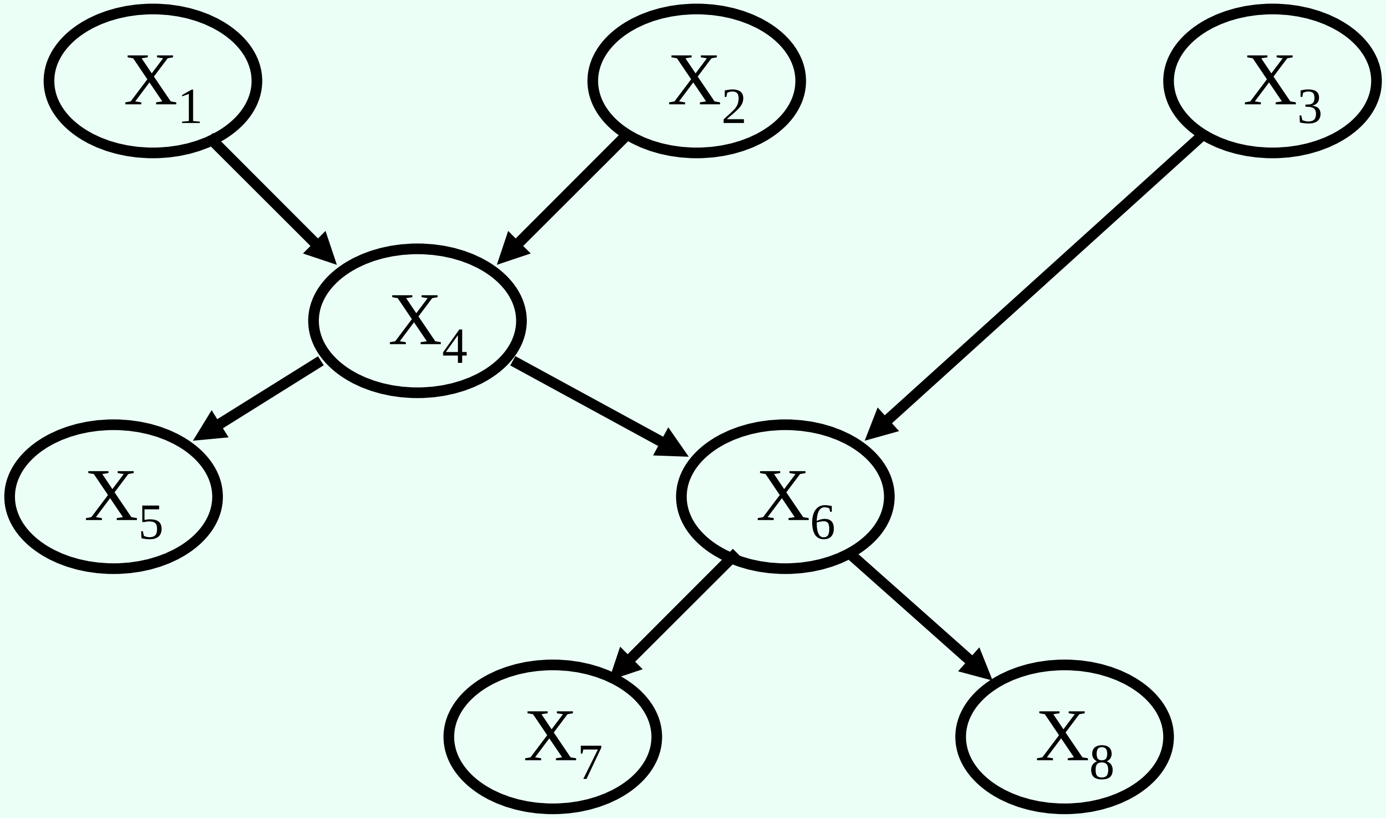
- $P(+r,+d) = \Sigma_n \Sigma_s \Sigma_g P(+r)P(s)P(n|+r)P(g|+r,s)P(+d|n,g) =$
  $P(+r)\Sigma_n P(n|+r)\Sigma_s P(s)\Sigma_g P(g|+r,s)P(+d|n,g)$
- Last factor will depend on two variables in this case!

Don’t always *need* to sum over **all** variables



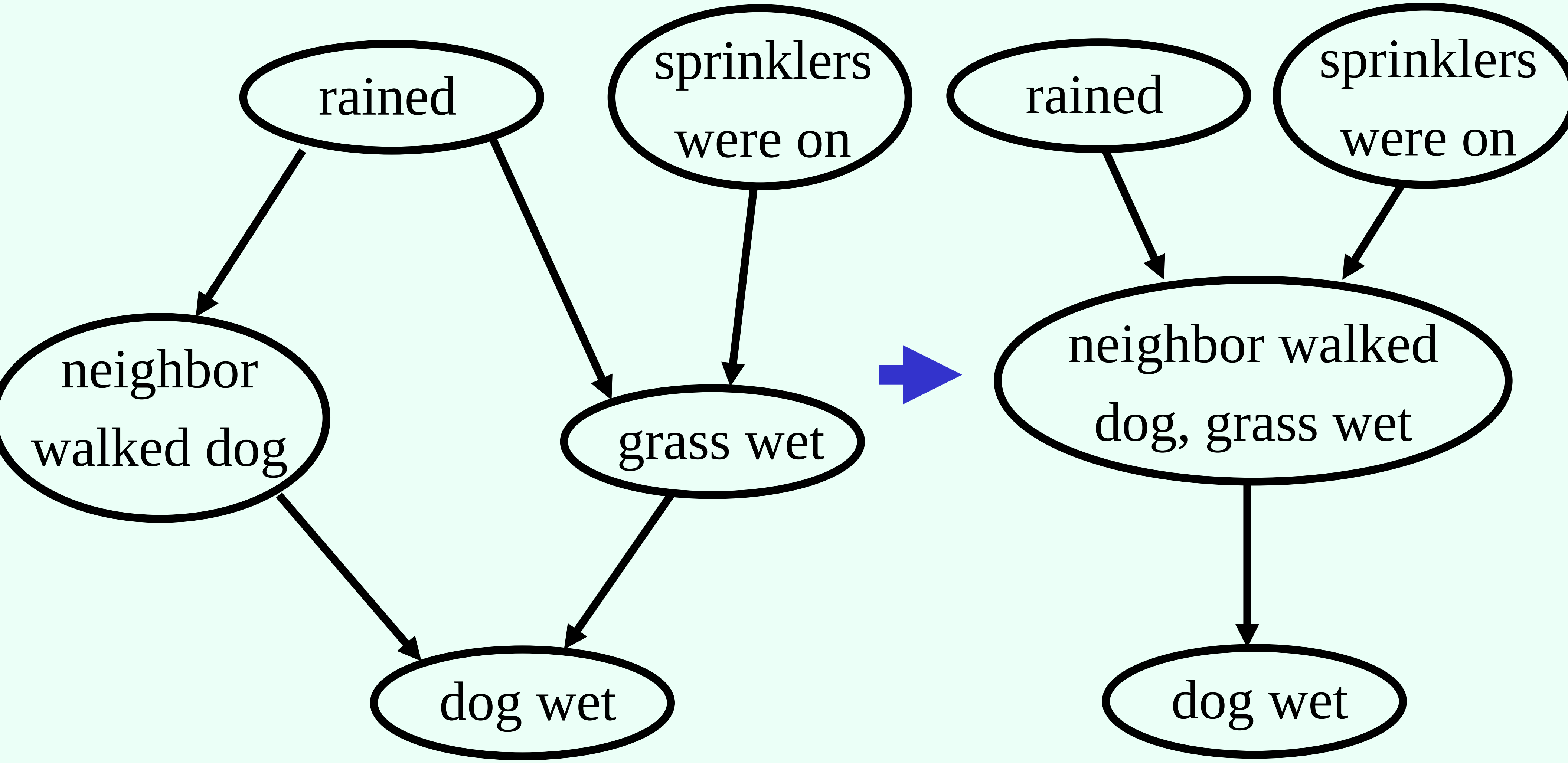
- Can drop parts of the network that are irrelevant
- P(+r, +s) = P(+r)P(+s) = .6*.2 = .12
- P(+n, +s) = Σ_rP(r, +n, +s) = Σ_rP(r)P(+n|r)P(+s) = P(+s)Σ_rP(r)P(+n|r) = P(+s)(P(+r)P(+n|+r) + P(-r)P(+n|-r)) = .6*(.2*.3 + .8*.4) = .6*.38 = .228
- P(+d | +n, +g, +s) = P(+d | +n, +g) = .9

Trees are easy



- Choose an extreme variable to eliminate first
- Its probability is “absorbed” by its neighbor
- … $\Sigma_{x4} P(x_4|x_1,x_2) \ldots \Sigma_{x5} P(x_5|x_4) = \ldots \Sigma_{x4} P(x_4|x_1,x_2)[\Sigma_{x5} P(x_5|x_4)] \ldots$

Clustering algorithms

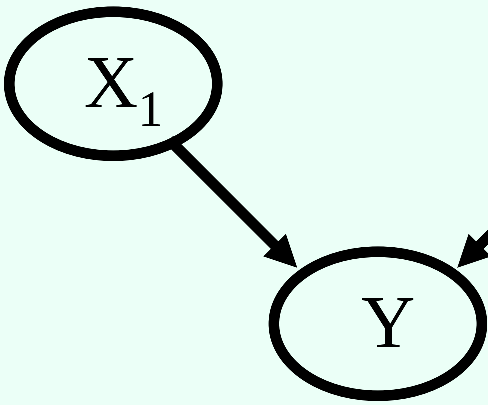


- Merge nodes into “meganodes” until we have a tree
  – Then, can apply special-purpose algorithm for trees
- Merged node has values {+n+g,+n-g,-n+g,-n-g}
  – Much larger CPT

Logic gates in Bayes nets

- Not everything needs to be random…

AND gate



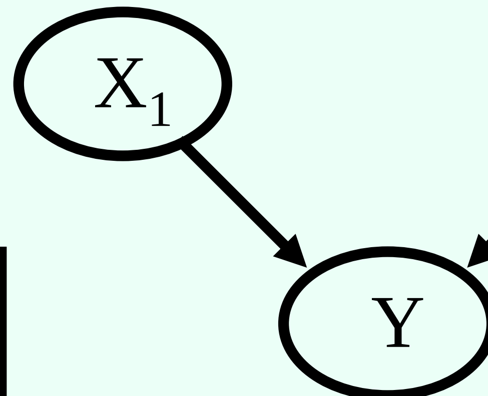
$P(+y|+x_1,+x_2) = 1$

$P(+y|-x_1,+x_2) = 0$

$P(+y|+x_1,-x_2) = 0$

$P(+y|-x_1,-x_2) = 0$

OR gate



$P(+y|+x_1,+x_2) = 1$

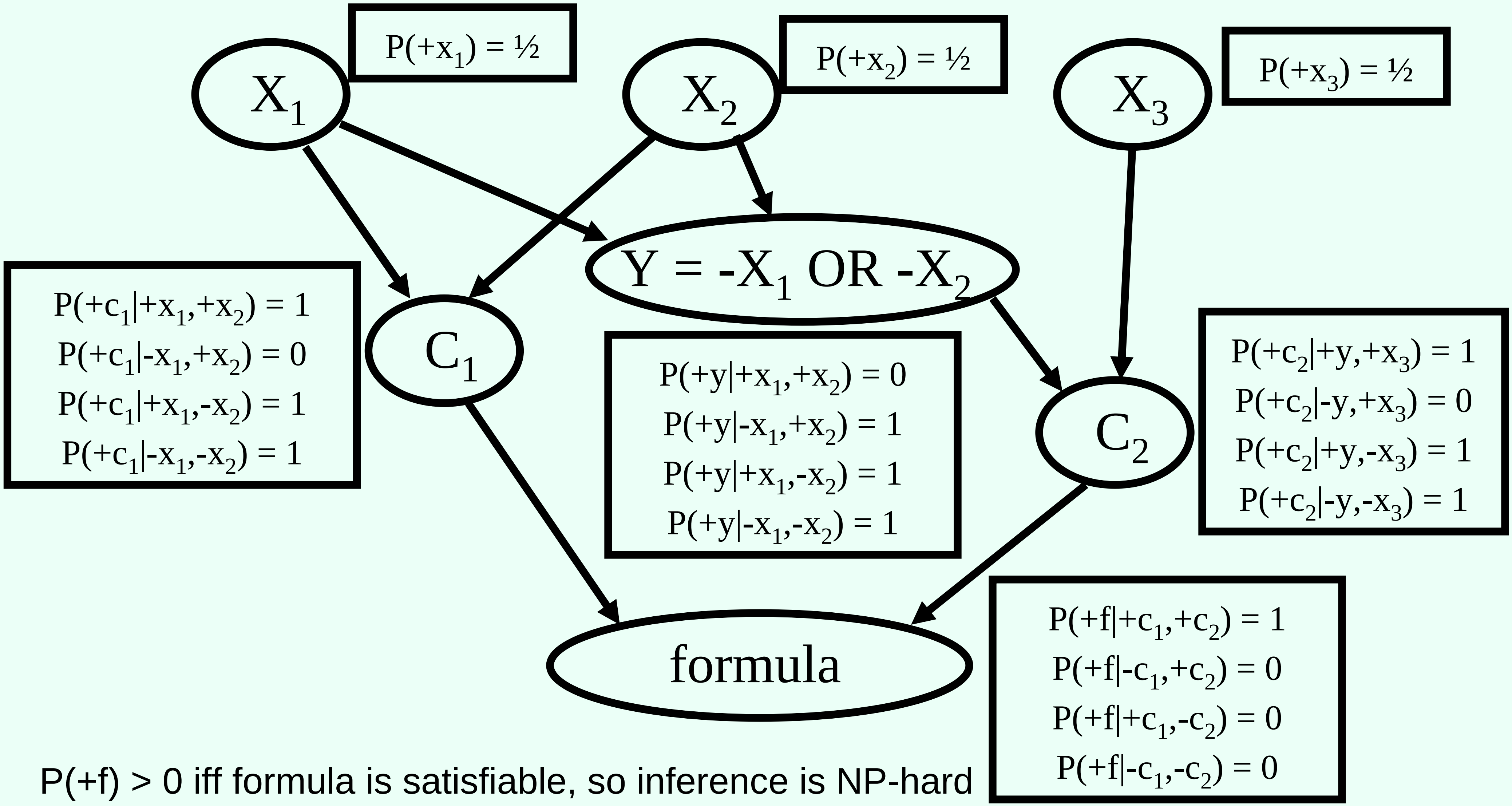
$P(+y|-x_1,+x_2) = 1$

$P(+y|+x_1,-x_2) = 1$

$P(+y|-x_1,-x_2) = 0$

Modeling satisfiability as a Bayes Net

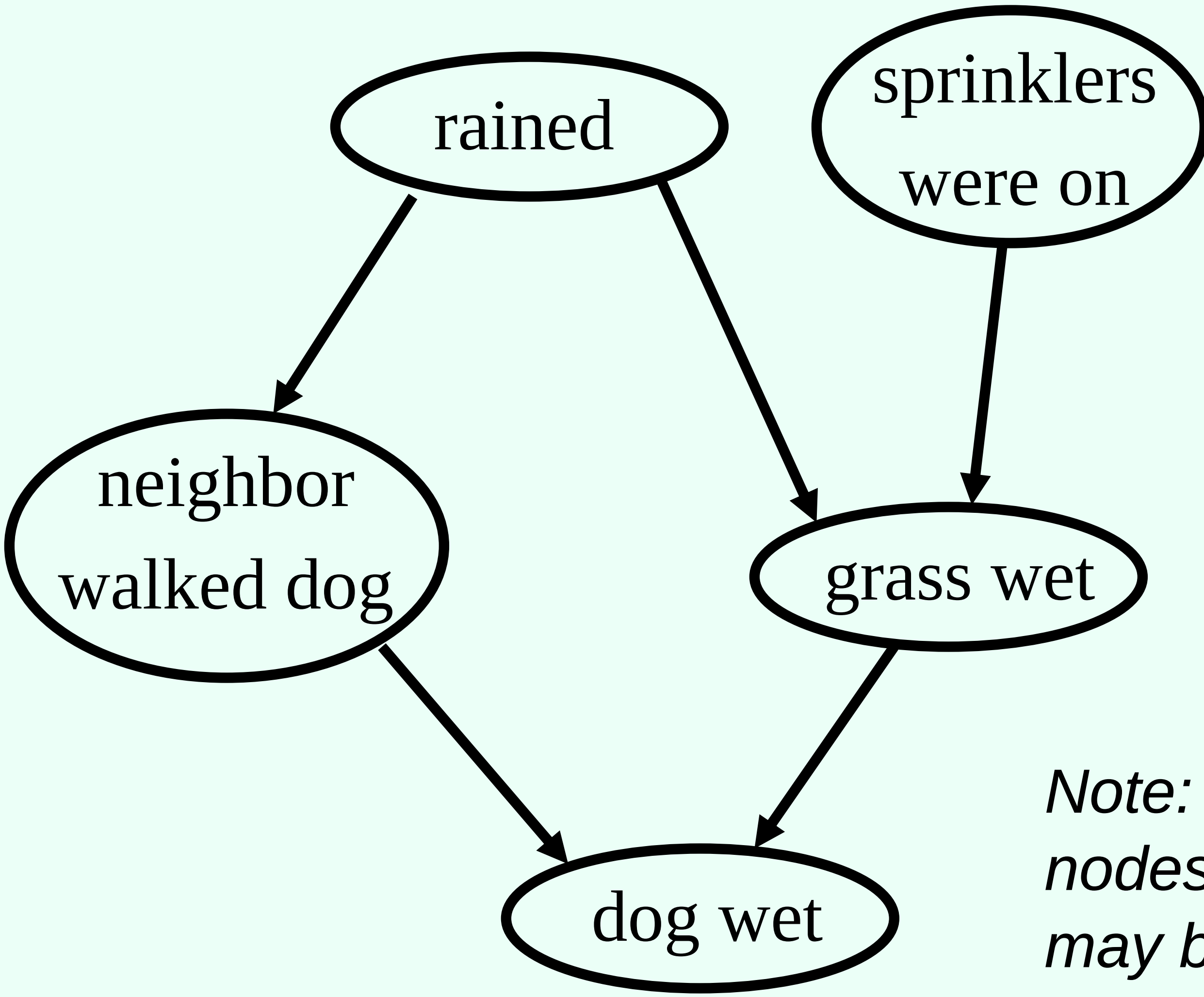
- $(+X_1 \text{ OR } -X_2) \text{ AND } (-X_1 \text{ OR } -X_2 \text{ OR } -X_3)$



- P(+f) > 0 iff formula is satisfiable, so inference is NP-hard
- P(+f) = (#satisfying assignments/2^n), so inference is #P-hard (because counting number of satisfying assignments is)

More about conditional independence

- A node is conditionally independent of its non-descendants, given its parents
- A node is conditionally independent of everything else in the graph, given its parents, children, and children's parents (its Markov blanket)

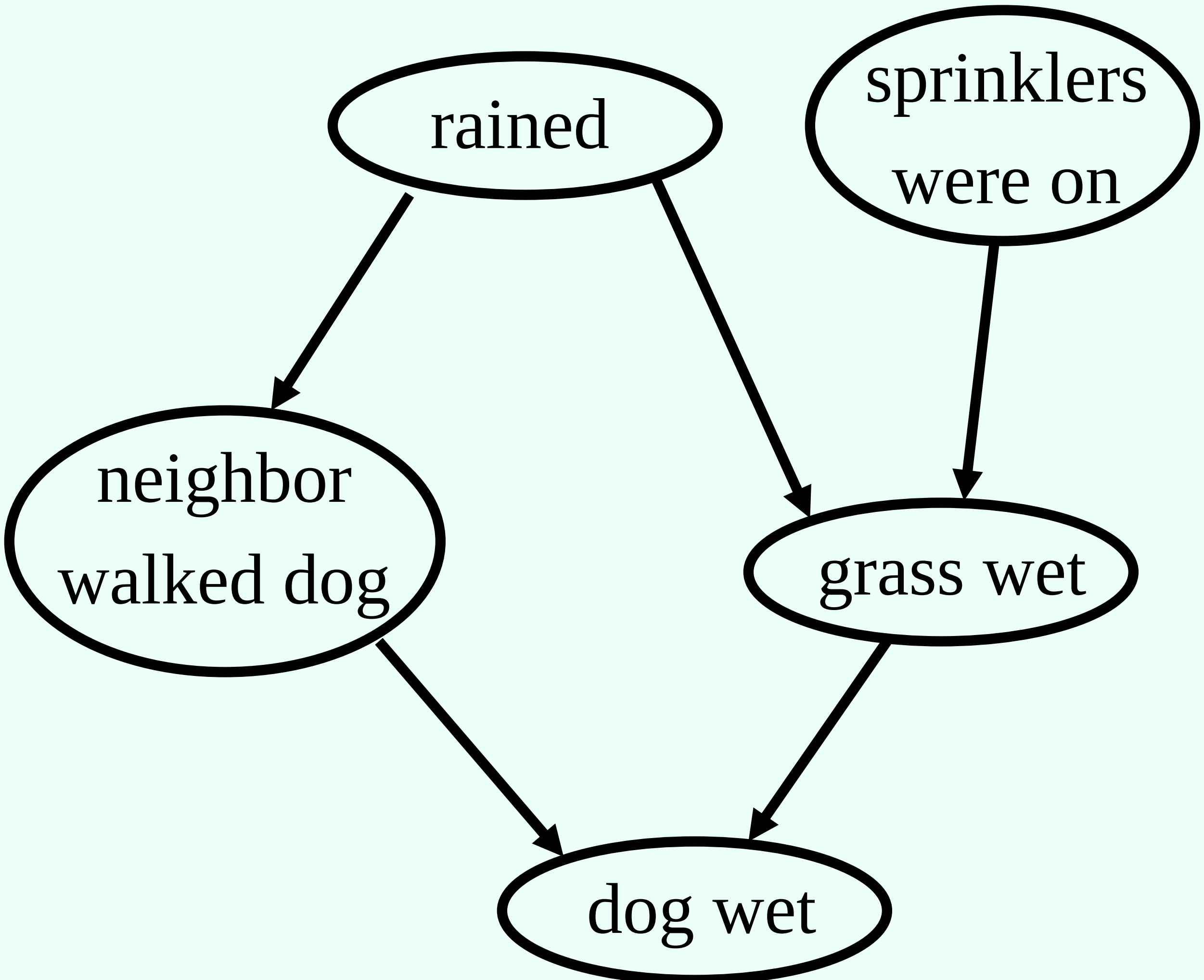


- N is independent of G given R
- N is **not** independent of G given R and D
- N is independent of S given R, G, D

*Note: can't know **for sure** that two nodes are not independent: edges may be dummy edges*

General criterion: d-separation

- Sets of variables *X* and *Y* are conditionally independent given variables in *Z* if all paths between *X* and *Y* are blocked; a path is blocked if one of the following holds:
  - it contains U -> V -> W or U <- V <- W or U <- V -> W, and V is in *Z*
  - it contains U -> V <- W, and neither V nor any of its descendants are in *Z*



- N is independent of G given R
- N is not independent of S given R and D