# COMP 424 – Tutorial 3

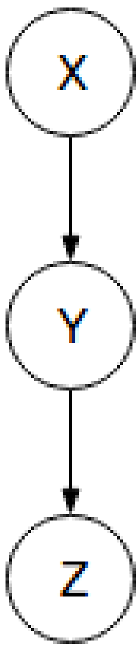
## Practice Questions for Assignment 3

~ Nicolas Angelard-Gontier ~
nicolas.angelard-gontier@mail.mcgill.ca

~ Rudolf Lam ~
rudolf.lam@mail.mcgill.ca

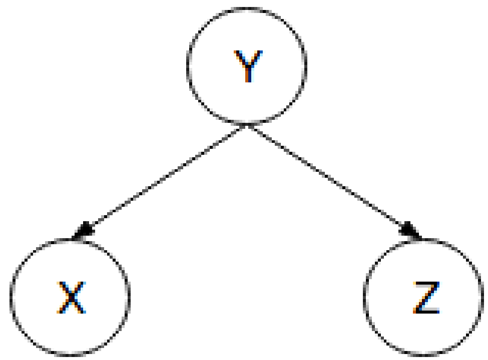
# (Recall Bayes Net)

Indirect Connection



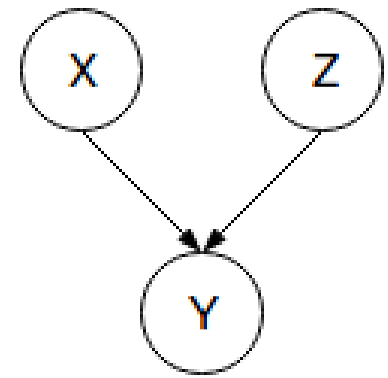

Z indep X : FALSE
Z indep X given Y: TRUE

Common Cause




Z indep X : FALSE
Z indep X given Y: TRUE

V-Structure




Z indep X : TRUE
Z indep X given Y: FALSE

# 1. Designing a Bayesian Network

- In your garden, there are sprinklers that sense when grass humidity gauge is lower than a given threshold. Consider the Boolean variables:

  S (sprinklers: on / off)
  OK_s (sprinklers are not broken: true / false)
  OK_g (gauge is not broken: true / false)
  G (humidity gauge reading: high / low)
  H (actual grass humidity: high / low

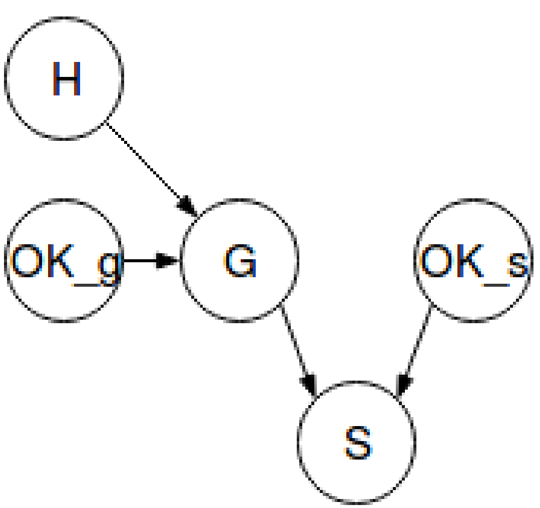
- Draw Bayesian Network for this domain:

# 1. Designing a Bayesian Network

- In your garden, there are sprinklers that sense when grass humidity gauge is lower than a given threshold. Consider the Boolean variables:

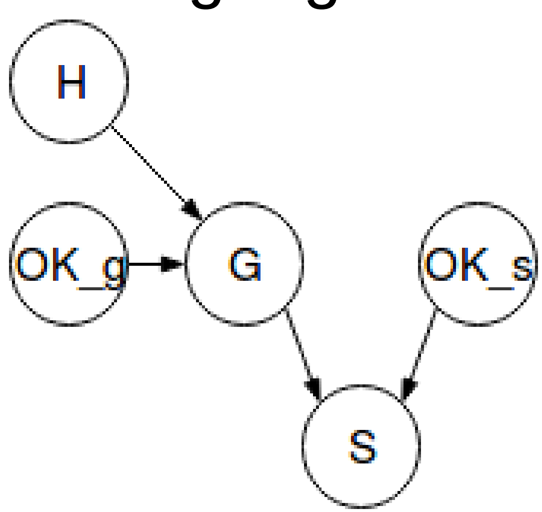
    S (sprinklers: on / off)
    OK_s (sprinklers are not broken: true / false)
    OK_g (gauge is not broken: true / false)
    G (humidity gauge reading: high / low)
    H (actual grass humidity: high / low

- Draw Bayesian Network for this domain:



# 1. Designing a Bayesian Network

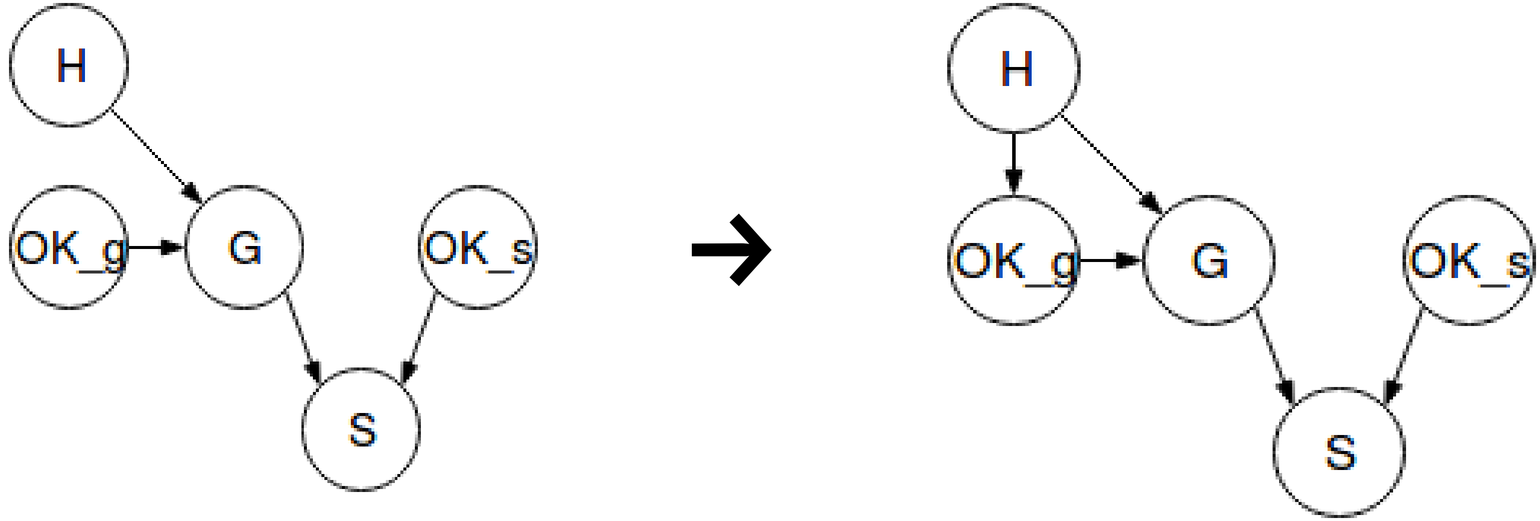
- In your garden, there are sprinklers that sense when grass humidity gauge is lower than a given threshold. Consider the Boolean variables:
  S (sprinklers: on / off)
  OK_s (sprinklers are not broken: true / false)
  OK_g (gauge is not broken: true / false)
  G (humidity gauge reading: high / low)
  H (actual grass humidity: high / low
- Draw Bayesian Network for this domain:
  now what if “gauge is more likely to fail when grass is too dry”?




→

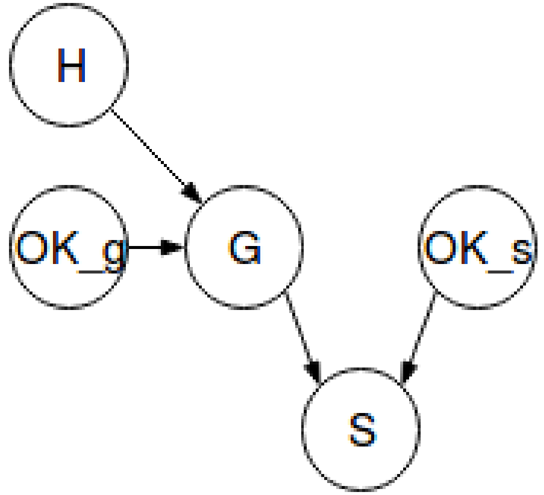
# 1. Designing a Bayesian Network

- In your garden, there are sprinklers that sense when grass humidity gauge is lower than a given threshold. Consider the Boolean variables:
  S (sprinklers: on / off)
  OK_s (sprinklers are not broken: true / false)
  OK_g (gauge is not broken: true / false)
  G (humidity gauge reading: high / low)
  H (actual grass humidity: high / low
- Draw Bayesian Network for this domain:
  now what if “gauge is more likely to fail when grass is too dry”?

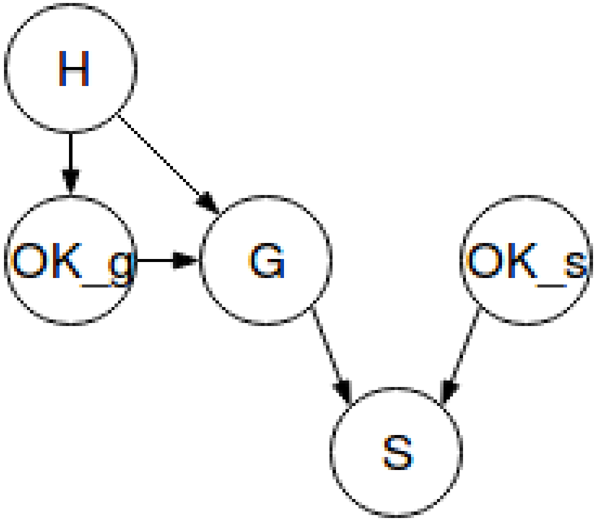


# 1. Designing a Bayesian Network

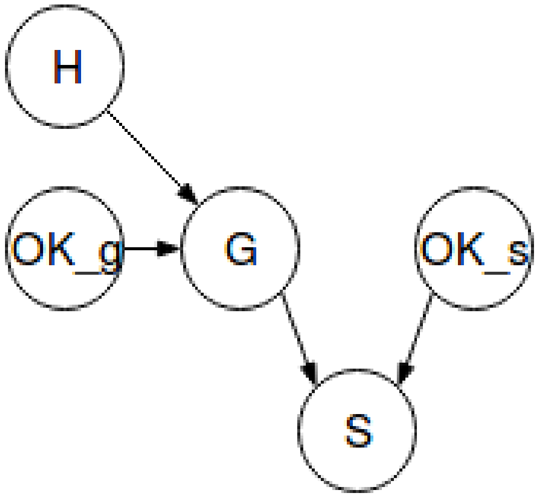
- Is your graph a polytree ?
- Polytree = it contains no cycle when you remove the direction of the edges
  - YES

  - NO

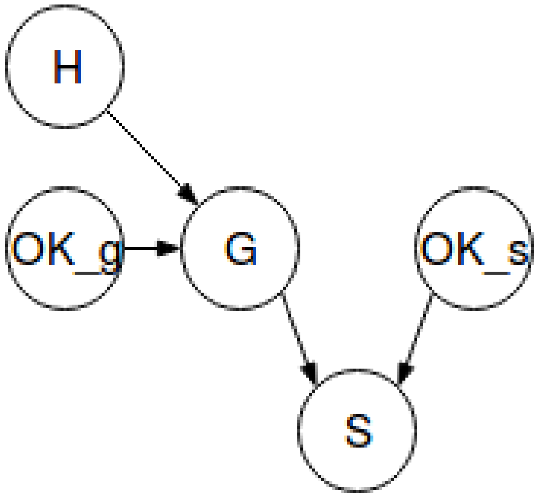
# 1. Designing a Bayesian Network




- Boolean variables:
  S (sprinklers: on / off)
  OK_s (sprinklers are not broken: true / false)
  OK_g (gauge is not broken: true / false)
  G (humidity gauge reading: high / low)
  H (actual grass humidity: high / low
- the probability that the gauge gives the correct humidity is *x* when it is working, but y when it is faulty. Give the conditional probability table associated with G:

| H | OK_g | | P(G=high \| H, OK_g) | P(G=low \| H, OK_g) |
|---|---|---|---|---|
| high | true | | | |
| high | false | | | |
| low | true | | | |
| low | false | | | |

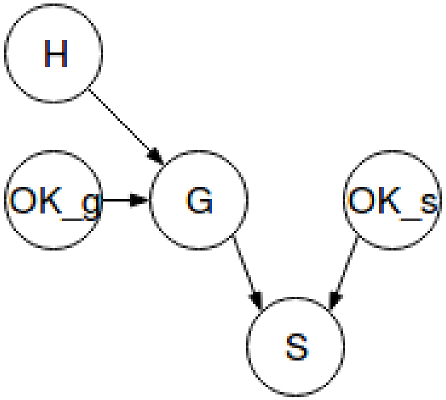
# 1. Designing a Bayesian Network




- Boolean variables:
  S (sprinklers: on / off)
  OK_s (sprinklers are not broken: true / false)
  OK_g (gauge is not broken: true / false)
  G (humidity gauge reading: high / low)
  H (actual grass humidity: high / low
- the probability that the gauge gives the correct humidity is *x* when it is working, but y when it is faulty. Give the conditional probability table associated with G:

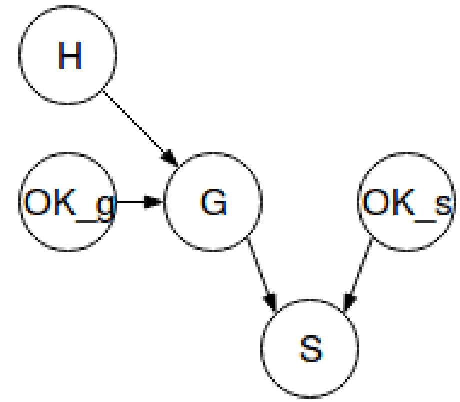
| H | OK_g | | P(G=high \| H, OK_g) | P(G=low \| H, OK_g) |
|---|---|---|---|---|
| high | true | | x | 1 - x |
| high | false | | y | 1 - y |
| low | true | | 1 - x | x |
| low | false | | 1 - y | y |

# 1. Designing a Bayesian Network




- Boolean variables:
  - S (sprinklers: on / off)
  - OK_s (sprinklers are not broken: true / false)
  - OK_g (gauge is not broken: true / false)
  - G (humidity gauge reading: high / low)
  - H (actual grass humidity: high / low
- Suppose the sprinklers and gauge are working and the sprinklers are on. Calculate an expression for the probability that the air humidity is high, in terms of the various conditional probabilities in the network.
- P(H=high | OK_s=1, OK_g=1, S=1) = ?

# 1. Designing a Bayesian Network

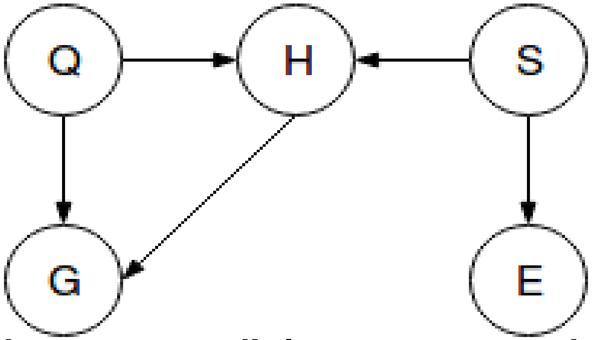



- Boolean variables:
  S (sprinklers: on / off)
  OK_s (sprinklers are not broken: true / false)
  OK_g (gauge is not broken: true / false)
  G (humidity gauge reading: high / low)
  H (actual grass humidity: high / low
- Suppose the sprinklers and gauge are working and the sprinklers are on. Calculate an expression for the probability that the air humidity is high, in terms of the various conditional probabilities in the network.
- P(H=high | OK_s=1, OK_g=1, S=1) = $\dfrac{P(H=high, OK_s=1, OK_g=1, S=1)}{P(OK_s=1, OK_g=1, S=1)}$

$$=\frac{\sum_g P(H=high, OK_s=1, OK_g=1, S=1, G=g)}{\sum_{g,h} P(H=h, OK_s=1, OK_g=1, S=1, G=g)}$$

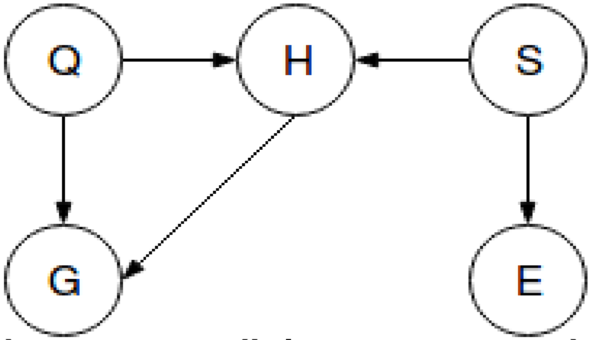
$$=\frac{\sum_g P(H=high)P(OK_g=1)P(G=g|H=high, OK_g=1)P(OK_s=1)P(S=1|G=g, OK_s=1)}{\sum_{g,h} P(H=h)P(OK_g=1)P(G=g|H=h, OK_g=1)P(OK_s=1)P(S=1|G=g, OK_s=1)}$$

# 2. Inference in Bayesian Networks




- Consider the following Bayesian Network:
  with the following properties:
  P(q) = 0.8    P(s) = 0.7
  P(e | s) = 0.7    P(e | not s) = 0.5
  P(h|q,s)=.9    P(h|q,not s)=.85    P(h|not q,s)=.15    P(h|not q,not s)=.3
  P(g|q,h)=.75    P(g|q,not h)=.4    P(g|not q,h)=.6    P(g|not q,not h)=.3
- P(h, not g) = ?

# 2. Inference in Bayesian Networks




- Consider the following Bayesian Network:
  with the following properties:
  P(q) = 0.8 P(s) = 0.7
  P(e | s) = 0.7 P(e | not s) = 0.5
  P(h|q,s)=.9 P(h|q,not s)=.85 P(h|not q,s)=.15 P(h|not q,not s)=.3
  P(g|q,h)=.75 P(g|q,not h)=.4 P(g|not q,h)=.6 P(g|not q,not h)=.3
- P(h, not g) = $\sum_{q,s,e} P(Q=q, S=s, h, \neg g, E=e)$

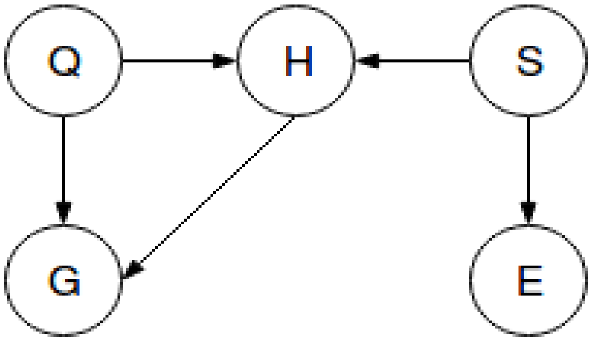
$$= \sum_{q,s,e} P(Q=q) P(S=s) P(h|Q=q, S=s) P(\neg g|Q=q, h) P(E=e|S=s)$$

| Q | S | E | P(Q=q)*P(S=s)*P(h\|Q=q, S=s)*P(not g\|Q=q, h)*P(E=e\|S=s) | |
|---|---|---|---|---|
| q | s | e | 0.8 * 0.7 * 0.9 * (1 - 0.75) * 0.7 | =0.00882 |
| q | s | not e | 0.8 * 0.7 * 0.9 * (1 - 0.75) * (1 - 0.7) | =0.03780 |
| q | not s | e | 0.8 * (1 - 0.7) * 0.85 * (1 - 0.75) * 0.5 | =0.02550 |
| q | not s | not e | 0.8 * (1 - 0.7) * 0.85 * (1 – 0.75) * (1 - 0.5) | =0.02550 |
| not q | s | e | (1 - 0.8) * 0.7 * 0.15 * (1 - 0.6) * 0.7 | =0.00588 |
| not q | s | not e | (1 - 0.8) * 0.7 * 0.15 * (1 - 0.6) * (1 - 0.7) | =0.00252 |
| not q | not s | e | (1 - 0.8) * (1 - 0.7) * 0.3 * (1 - 0.6) * 0.5 | =0.00360 |
| not q | not s | not e | (1 - 0.8) * (1 - 0.7) * 0.3 * (1 - 0.6) * (1 - 0.5) | =0.00360 |
| | | | | **=0.11322** |

# 3. Variable elimination

1. Impose order over all variables
   – Note: Query variable is LAST in the ordering
2. Create a list of factors
3. For each variable in ordering in (1), marginalize (ie: sum over all possible values) to replace factors by 'messages'
4. Memorize intermediate results

# 3. Variable elimination

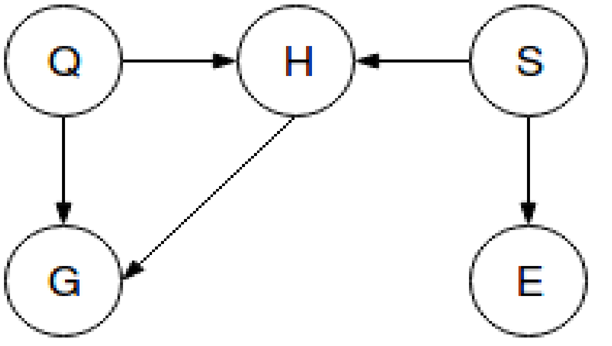



- Consider the following Bayesian Network:
  with the following properties:
  P(q) = 0.8 P(s) = 0.7
  P(e | s) = 0.7 P(e | not s) = 0.5
  P(h|q,s)=.9 P(h|q,not s)=.85 P(h|not q,s)=.15 P(h|not q,not s)=.3
  P(g|q,h)=.75 P(g|q,not h)=.4 P(g|not q,h)=.6 P(g|not q,not h)=.3

- Compute the MAP result of querying P(**Q**|e) using variable elimination with the following order: G, H, S, E, **Q**.

- (unordered) list of factors:
  –

# 3. Variable elimination



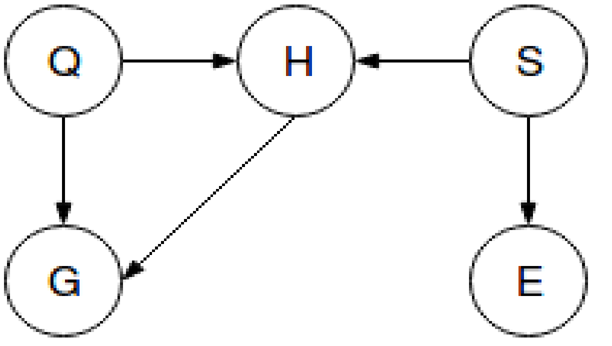

- Consider the following Bayesian Network:
  with the following properties:
  P(q) = 0.8 P(s) = 0.7
  P(e | s) = 0.7 P(e | not s) = 0.5
  P(h|q,s)=.9 P(h|q,not s)=.85 P(h|not q,s)=.15 P(h|not q,not s)=.3
  P(g|q,h)=.75 P(g|q,not h)=.4 P(g|not q,h)=.6 P(g|not q,not h)=.3

- Compute the MAP result of querying P(Q|e) using variable elimination with the following order: G, H, S, E, Q.

- (unordered) list of factors:
  - P(Q), P(S), P(H | Q, S), P(G | Q, H), P(E | S), $\delta(E, e)$

    - Note: $\delta(E,e)=\left\{\begin{matrix} 1\, if\ E=e \\ 0\, otherise \end{matrix}\right\}$

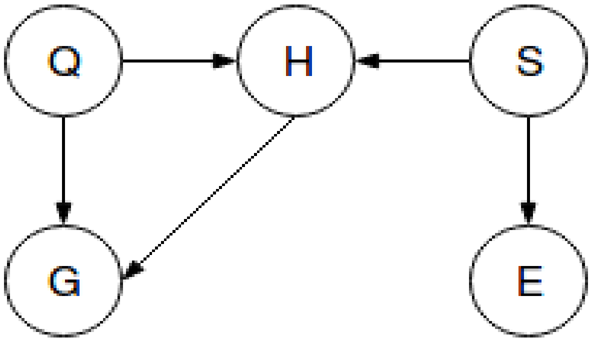
# 3. Variable elimination




- Consider the following Bayesian Network:
  with the following properties:
  P(q) = 0.8        P(s) = 0.7
  P(e | s) = 0.7   P(e | not s) = 0.5
  P(h|q,s)=.9      P(h|q,not s)=.85   P(h|not q,s)=.15   P(h|not q,not s)=.3
  P(g|q,h)=.75    P(g|q,not h)=.4     P(g|not q,h)=.6    P(g|not q,not h)=.3

- Compute the MAP result of querying P(Q|e) using variable elimination with the following order: **G**, H, S, E, Q.
- (unordered) list of factors:
  - P(Q), P(S), P(H | Q, S), **P(G | Q, H)**, P(E | S), $\delta(E, e)$
- Eliminate **G**:

# 3. Variable elimination




- Consider the following Bayesian Network:
  with the following properties:
  P(q) = 0.8 P(s) = 0.7
  P(e | s) = 0.7 P(e | not s) = 0.5
  P(h|q,s)=.9 P(h|q,not s)=.85 P(h|not q,s)=.15 P(h|not q,not s)=.3
  P(g|q,h)=.75 P(g|q,not h)=.4 P(g|not q,h)=.6 P(g|not q,not h)=.3

- Compute the MAP result of querying P(Q|e) using variable elimination with the following order: **G**, H, S, E, Q.

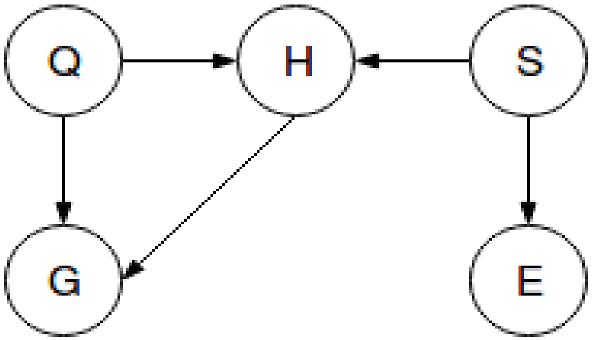
- (unordered) list of factors:
  - P(Q), P(S), P(H | Q, S), **P(G | Q, H)**, P(E | S), $\delta(E,e)$

- Eliminate **G**:

$$m_G(Q,H)=\sum_g P(g|Q,H)$$

$$\begin{bmatrix}(q,h) & (q,\neg h)\\ (\neg q,h) & (\neg q,\neg h)\end{bmatrix}: m_G(Q,H)=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}\stackrel{\text{def}}{=}[1]$$

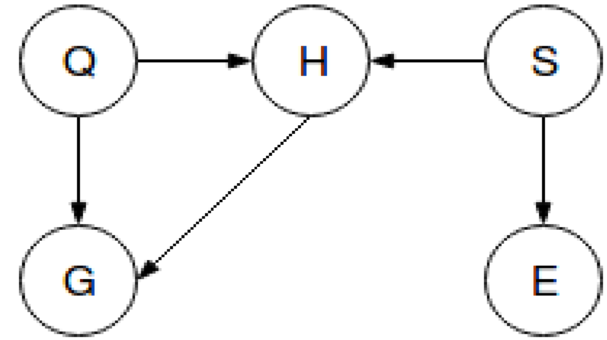
# 3. Variable elimination




- Consider the following Bayesian Network: with the following properties:
  P(q) = 0.8 P(s) = 0.7
  P(e | s) = 0.7 P(e | not s) = 0.5
  P(h|q,s)=.9 P(h|q,not s)=.85 P(h|not q,s)=.15 P(h|not q,not s)=.3
  P(g|q,h)=.75 P(g|q,not h)=.4 P(g|not q,h)=.6 P(g|not q,not h)=.3

- Compute the MAP result of querying P(Q|e) using variable elimination with the following order: G, **H**, S, E, Q.
- List: P(Q), P(S), **P(H | Q, S)**, $\boldsymbol{m_G(Q,H)}$ , P(E | S), $\delta(E,e)$
- Eliminate **H**:

# 3. Variable elimination



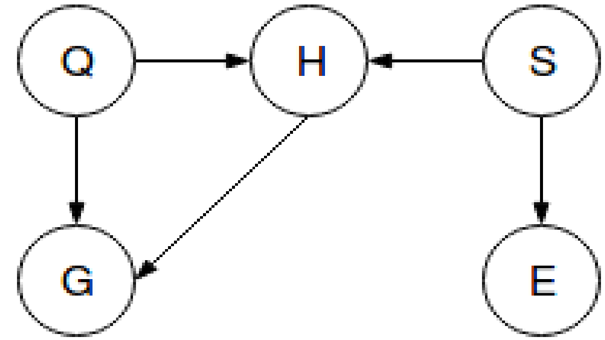

- Consider the following Bayesian Network:
  with the following properties:
  P(q) = 0.8      P(s) = 0.7
  P(e | s) = 0.7   P(e | not s) = 0.5
  P(h|q,s)=.9    P(h|q,not s)=.85   P(h|not q,s)=.15   P(h|not q,not s)=.3
  P(g|q,h)=.75    P(g|q,not h)=.4    P(g|not q,h)=.6    P(g|not q,not h)=.3

- Compute the MAP result of querying P(Q|e) using variable elimination with the following order: G, **H**, S, E, Q.
- List: P(Q), P(S), **P(H | Q, S)**, $\boldsymbol{m_G(Q,H)}$ , P(E | S), $\delta(E,e)$
- Eliminate **H**:

$$m_H(Q,S)=\sum_h P(h|Q,S)m_G(Q,h)=\sum_h P(h|Q,S)*[1]$$

$$\begin{bmatrix}(q,s) & (q,\neg s)\\ (\neg q,s) & (\neg q,\neg s)\end{bmatrix}: m_H(Q,S)=\begin{bmatrix}1*[1] & 1*[1]\\ 1*[1] & 1*[1]\end{bmatrix}\overset{\text{def}}{=}[1]$$

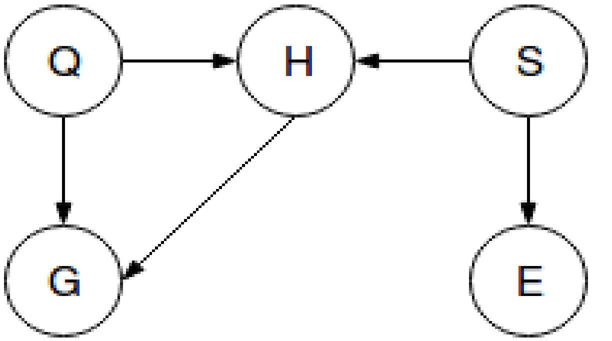
# 3. Variable elimination




- Consider the following Bayesian Network:
  with the following properties:
  P(q) = 0.8 P(s) = 0.7
  P(e | s) = 0.7 P(e | not s) = 0.5
  P(h|q,s)=.9 P(h|q,not s)=.85 P(h|not q,s)=.15 P(h|not q,not s)=.3
  P(g|q,h)=.75 P(g|q,not h)=.4 P(g|not q,h)=.6 P(g|not q,not h)=.3

- Compute the MAP result of querying P(Q|e) using variable elimination with the following order: G, H, **S**, E, Q.
- List: P(Q), **P(S),** $\boldsymbol{m_H(Q,S)}$ **, P(E | S)**, $\delta(E,e)$
- Eliminate **S**:

# 3. Variable elimination




- Consider the following Bayesian Network:
  with the following properties:
  P(q) = 0.8 P(s) = 0.7
  P(e | s) = 0.7 P(e | not s) = 0.5
  P(h|q,s)=.9 P(h|q,not s)=.85 P(h|not q,s)=.15 P(h|not q,not s)=.3
  P(g|q,h)=.75 P(g|q,not h)=.4 P(g|not q,h)=.6 P(g|not q,not h)=.3

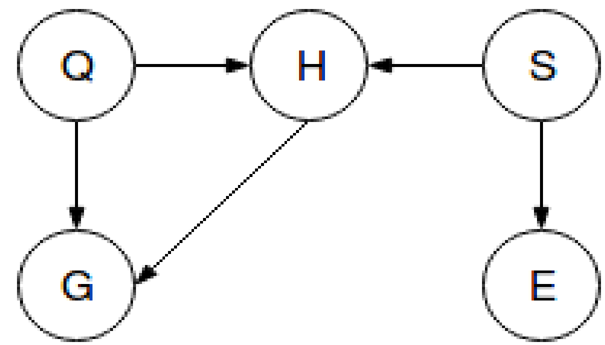
- Compute the MAP result of querying P(Q|e) using variable elimination with the following order: G, H, **S**, E, Q.
- List: P(Q), **P(S),** $\boldsymbol{m_H(Q,S)}$ **, P(E | S)**, $\delta(E,e)$
- Eliminate **S**:

$$m_S(Q,E)=\sum_s P(s)P(E|s)m_H(Q,S)=\sum_s P(s)P(E|s)[1]$$

$$m_S(Q,E)=P(s)P(E|s)+P(\neg s)P(E|\neg s)$$

$$[\begin{matrix}E=e\\E=\neg e\end{matrix}]:m_S(Q,E)=0.7*[\begin{matrix}0.7\\0.3\end{matrix}]+0.3*[\begin{matrix}0.5\\0.5\end{matrix}]=[\begin{matrix}0.64\\0.36\end{matrix}]$$

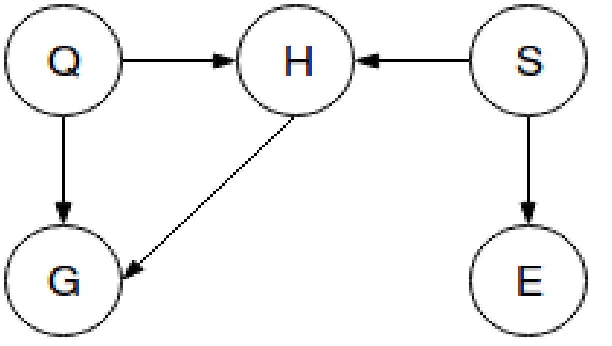
# 3. Variable elimination




- Consider the following Bayesian Network:
  with the following properties:
  P(q) = 0.8 P(s) = 0.7
  P(e | s) = 0.7 P(e | not s) = 0.5
  P(h|q,s)=.9 P(h|q,not s)=.85 P(h|not q,s)=.15 P(h|not q,not s)=.3
  P(g|q,h)=.75 P(g|q,not h)=.4 P(g|not q,h)=.6 P(g|not q,not h)=.3

- Compute the MAP result of querying P(Q|e) using variable elimination with the following order: G, H, S, **E**, Q.
- List: P(Q), $\boldsymbol{m_S(Q,E)}$ , $\boldsymbol{\delta(E,e)}$
- Eliminate **E**: $$m_E(Q)=\sum_e m_S(Q,E)\delta(E,e)=0.64*1+0.36*0=0.64$$
- New list: P(Q), $m_E(Q)$

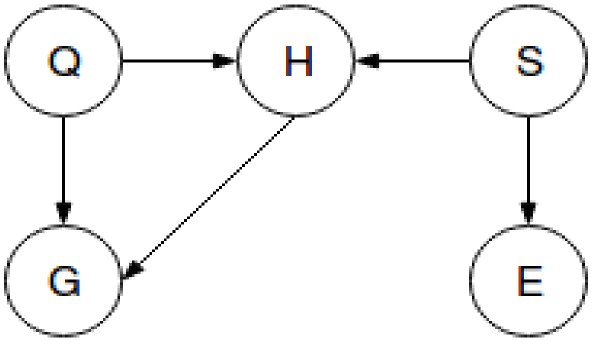
# 3. Variable elimination




- Consider the following Bayesian Network:
  with the following properties:
  P(q) = 0.8 P(s) = 0.7
  P(e | s) = 0.7 P(e | not s) = 0.5
  P(h|q,s)=.9 P(h|q,not s)=.85 P(h|not q,s)=.15 P(h|not q,not s)=.3
  P(g|q,h)=.75 P(g|q,not h)=.4 P(g|not q,h)=.6 P(g|not q,not h)=.3

- Compute the MAP result of querying P(Q|e) using variable elimination with the following order: G, H, S, E, **Q**.
- List: P(Q), $m_E(Q)$
- Query **Q**:

# 3. Variable elimination

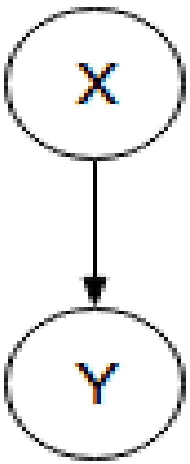



- Consider the following Bayesian Network:
  with the following properties:
  P(q) = 0.8 P(s) = 0.7
  P(e | s) = 0.7 P(e | not s) = 0.5
  P(h|q,s)=.9 P(h|q,not s)=.85 P(h|not q,s)=.15 P(h|not q,not s)=.3
  P(g|q,h)=.75 P(g|q,not h)=.4 P(g|not q,h)=.6 P(g|not q,not h)=.3

- Compute the MAP result of querying P(Q|e) using variable elimination with the following order: G, H, S, E, **Q**.
- List: P(Q), $m_E(Q)$
- Query **Q**:
  - Q = q : 0.8 * 0.64 <----
  - Q = not q : 0.2 * 0.64

# 4. Learning with Bayesian Networks

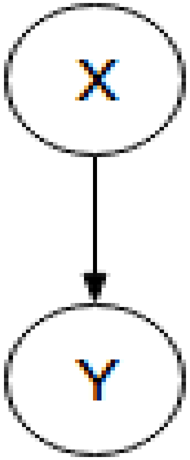
Consider the following graph:




- Enumerate the parameters that must be learned.

# 4. Learning with Bayesian Networks

Consider the following graph:



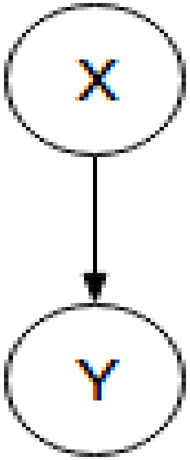

- Enumerate the parameters that must be learned.

$$\theta_X = P(X) \qquad \theta_{Y,1} = P(Y|X=1)$$

$$\theta_{Y,0} = P(Y|X=0)$$

# 4. Learning with Bayesian Networks

- Consider the following graph:




$$\theta_X = P(X)$$

$$\theta_Y = P(Y|X=1)$$

$$\theta_Y = P(Y|X=0)$$

- Given samples:

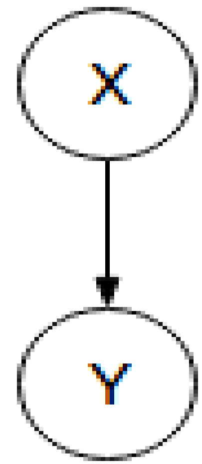
| X | Y | # of instances |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 2 |
| 1 | 0 | 3 |
| 1 | 1 | 4 |

- Compute MLE:

# 4. Learning with Bayesian Networks

- Consider the following graph:
- Given samples:

| X | Y | # of instances |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 2 |
| 1 | 0 | 3 |
| 1 | 1 | 4 |




$$\theta_X = P(X)$$

$$\theta_Y = P(Y|X=1)$$

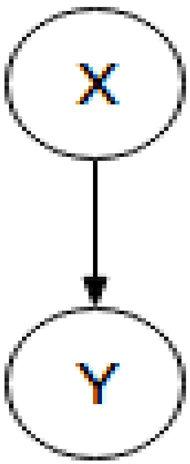
$$\theta_Y = P(Y|X=0)$$

- Compute MLE:
  - P(X=1) = #(X=1) / [#(X=1) + #(X=0)] = 7 / 10
  - P(Y=1 | X=1) = #(Y=1, X=1) / [#(Y=1, X=1) + #(Y=0, X=1)]
    = 4 / (4+3) = 4 / 7
  - P(Y=1 | X=0) = #(Y=1, X=0) / [#(Y=1, X=0) + #(Y=0, X=0)]
    = 2 / (2+1) = 2 / 3

# 4. Learning with Bayesian Networks

- Consider the following graph:
- Given samples:

| X | Y | # of instances |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 2 |
| 1 | 0 | 3 |
| 1 | 1 | 4 |




$$\theta_X = P(X)$$

$$\theta_Y = P(Y|X=1)$$

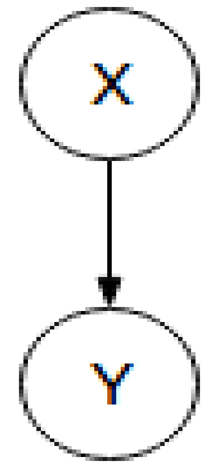
$$\theta_Y = P(Y|X=0)$$

- Give the maximum a posterior estimate for each parameter after applying Laplace smoothing:

# 4. Learning with Bayesian Networks

- Consider the following graph:
- Given samples:

| X | Y | # of instances |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 2 |
| 1 | 0 | 3 |
| 1 | 1 | 4 |




$$\theta_X = P(X)$$

$$\theta_Y = P(Y|X=1)$$

$$\theta_Y = P(Y|X=0)$$

- Give the maximum a posterior estimate for each parameter after applying Laplace smoothing:
  - P(X=1) = #(X=1)**+1** / [#(X=1)**+1** + #(X=0)**+1**] = 8 / 12
  - P(Y=1 | X=1) = #(Y=1, X=1)**+1** / [#(Y=1, X=1)**+1** + #(Y=0, X=1)**+1**]
    = 5 / (5+4) = 5 / 9
  - P(Y=1 | X=0) = #(Y=1, X=0)**+1** / [#(Y=1, X=0)**+1** + #(Y=0, X=0)**+1**]
    = 3 / (3+2) = 3 / 5

# 4. Learning with Bayesian Networks

- Missing Data: same as before but with extra entry: <X=0, Y=?>
- We will apply the (soft) EM algorithm on these instances. Initialize the model using MLE parameter estimates.
- Show the computation of the first E-step:

| X | Y |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 0 | 1 |
| 1 | 0 |
| 1 | 0 |
| 1 | 0 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 0 | ? |

# 4. Learning with Bayesian Networks

- Missing Data: same as before but with extra entry: <X=0, Y=?>
- We will apply the (soft) EM algorithm on these instances. Initialize the model using MLE parameter estimates.
- Show the computation of the first E-step:
  - From before we have:
    P(Y=1 | X=0) = 2 / 3
    so we have weights
    Y=1 :  2 / 3
    Y=0 :  1 / 3

| X | Y |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 0 | 1 |
| 1 | 0 |
| 1 | 0 |
| 1 | 0 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 0 | ? |

# 4. Learning with Bayesian Networks

- Missing Data: same as before but with extra entry: <X=0, Y=?>
- We will apply the (soft) EM algorithm on these instances. Initialize the model using MLE parameter estimates.
- What are the parameters obtained for the first M-step?

| X | Y |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 0 | 1 |
| 1 | 0 |
| 1 | 0 |
| 1 | 0 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 0 | ? |

# 4. Learning with Bayesian Networks

| X | Y |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 0 | 1 |
| 1 | 0 |
| 1 | 0 |
| 1 | 0 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 0 | ? |

- Missing Data: same as before but with extra entry: <X=0, Y=?>
- We will apply the (soft) EM algorithm on these instances. Initialize the model using MLE parameter estimates.
- What are the parameters obtained for the first M-step?
  - P(X=1) same as before
  - P(Y=1 | X=1) same as before
  - P(Y=1 | X=0) = #(Y=1, X=0) / [#(Y=1, X=0) + #(Y=0, X=0)]
    = (2**+2/3**) / [(2**+2/3**)+(1**+1/3**) = 2 / 3
    converged!
- Note: for this example, the data does not provide us with more info and has converged in 1 step. No further E-step is required.

# Questions ?

~ Nicolas Angelard-Gontier ~
nicolas.angelard-gontier@mail.mcgill.ca

~ Rudolf Lam ~
rudolf.lam@mail.mcgill.ca