Reading:
Chapters 5&6 of Koller&Friedman

Variable elimination

Graphical Models – 10708

Carlos Guestrin
Carnegie Mellon University

September 26th, 2005

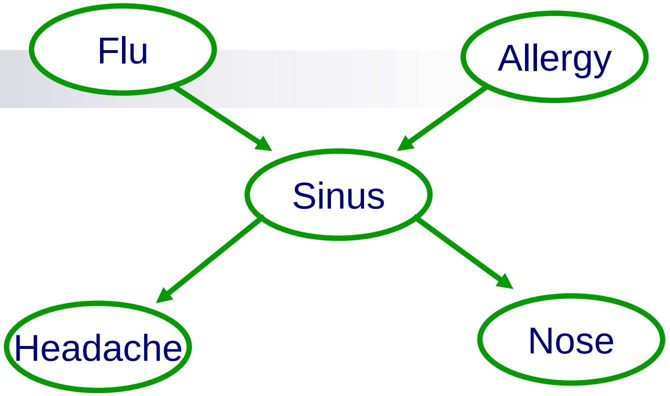
Announcements

- Waiting List
  - Anyone still wants to be registered?

Inference in BNs hopeless?

- In general, yes!
  - Even approximate!
- In practice
  - Exploit structure
  - Many effective approximation algorithms (some with guarantees)
- For now, we'll talk about exact inference
  - Approximate inference later this semester

General probabilistic inference



- Query: $P(X \mid e)$

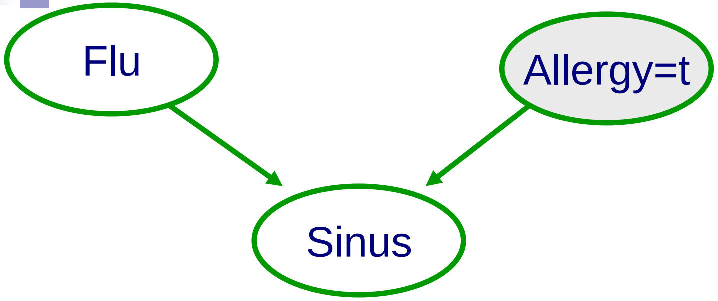
- Using def. of cond. prob.:

$$P(X \mid e) = \frac{P(X, e)}{P(e)}$$

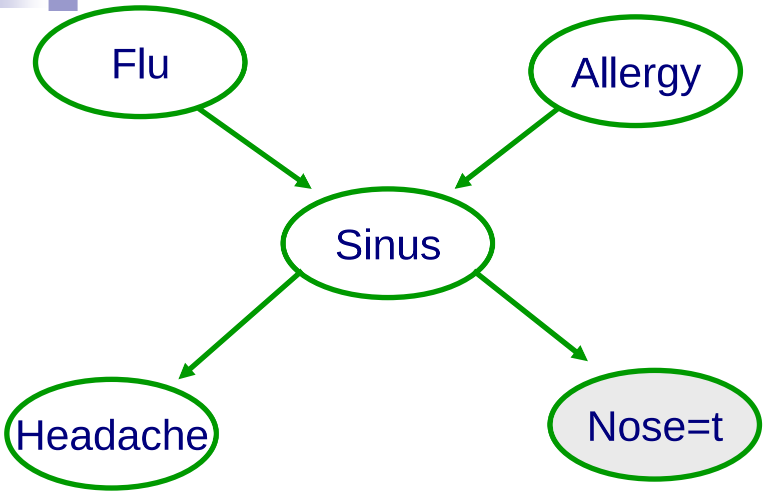
- Normalization:

$$P(X \mid e) \propto P(X, e)$$

Marginalization

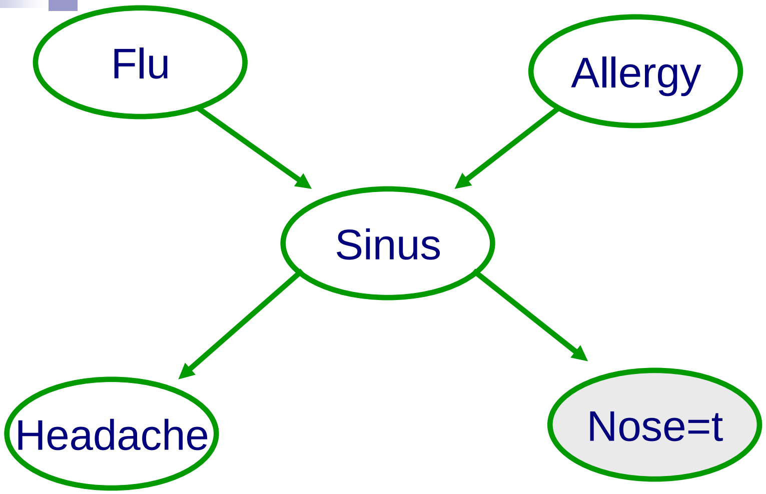


Probabilistic inference example



Inference seems exponential in number of variables!

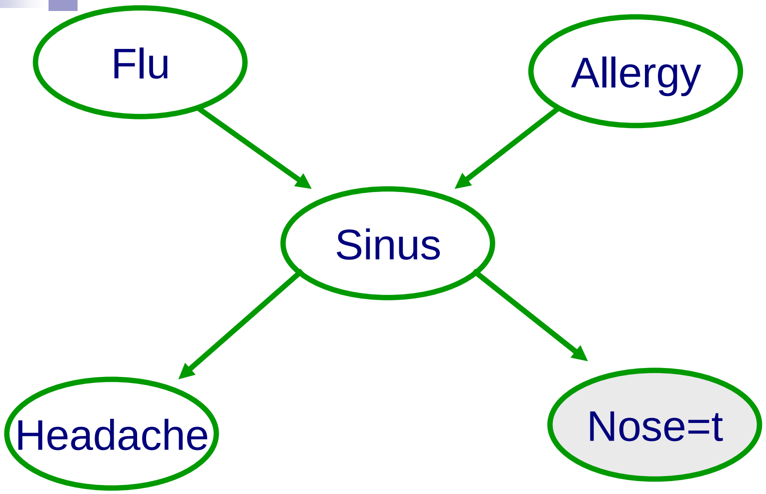
Fast probabilistic inference example – Variable elimination



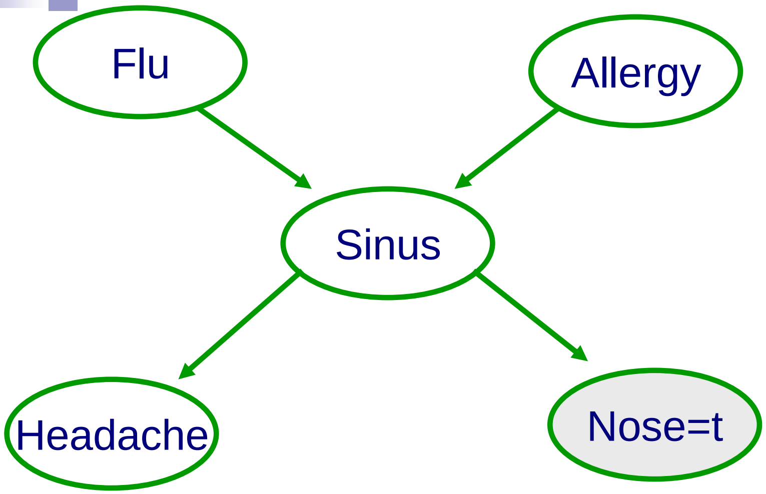
(Potential for) Exponential reduction in computation!

Understanding variable elimination – Exploiting distributivity

Understanding variable elimination – Order can make a HUGE difference

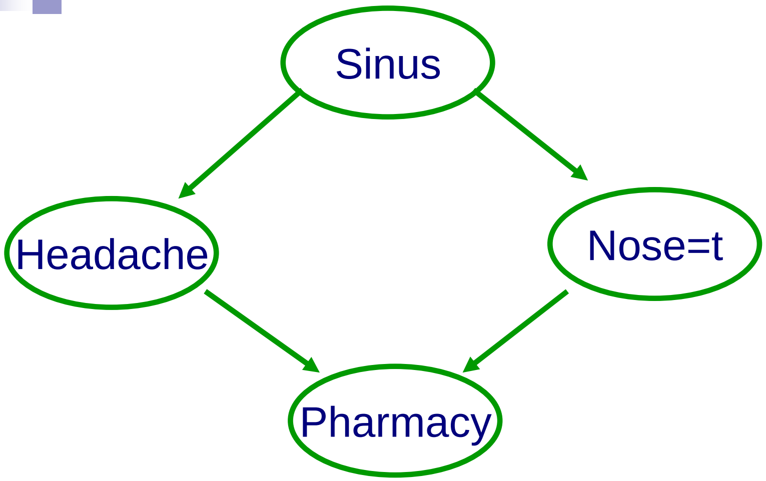


Understanding variable elimination – Intermediate results

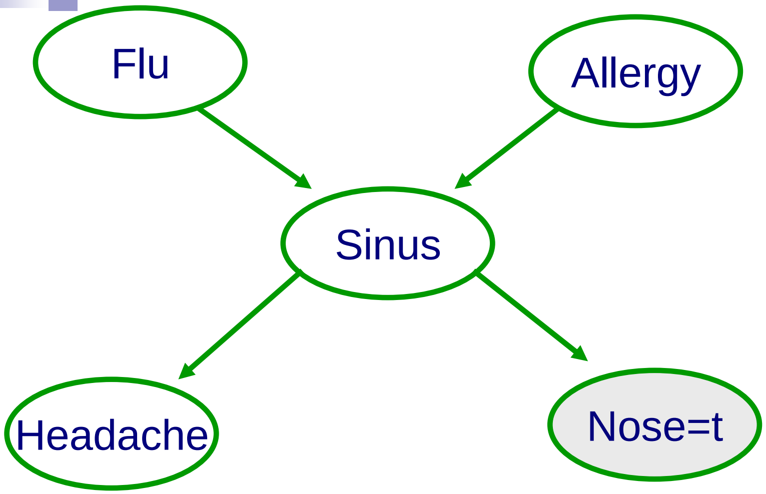


Intermediate results are probability distributions

Understanding variable elimination – Another example



Pruning irrelevant variables



Prune all non-ancestors of query variables

More generally: Prune all nodes not on active trail between evidence and query vars

Variable elimination algorithm

- Given a BN and a query $P(X|e) \propto P(X,e)$
- Instantiate evidence **e**
- Prune non-active vars for {X,**e**}

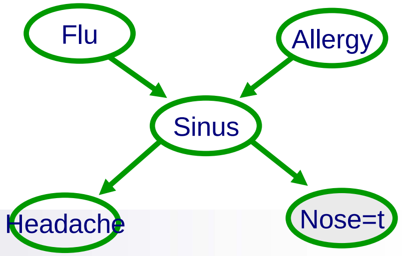
IMPORTANT!!!

- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- Initial *factors* $\{f_1,\ldots,f_n\}$: $f_i = P(X_i|\mathbf{Pa}_{X_i})$ (CPT for X_i)
- For i = 1 to n, If $X_i \notin \{X,\mathbf{E}\}$
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

$$g = \sum_{X_i} \prod_{j=1}^{k} f_j$$

  - Variable X_i has been eliminated!
- Normalize $P(X,\mathbf{e})$ to obtain $P(X|\mathbf{e})$

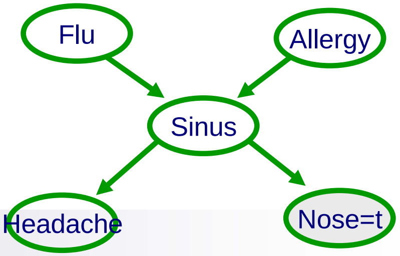
Operations on factors



$$g = \sum_{X_i} \prod_{j=1}^{k} f_j$$

Multiplication:

Operations on factors



$$g = \sum_{X_i} \prod_{j=1}^{k} f_j$$

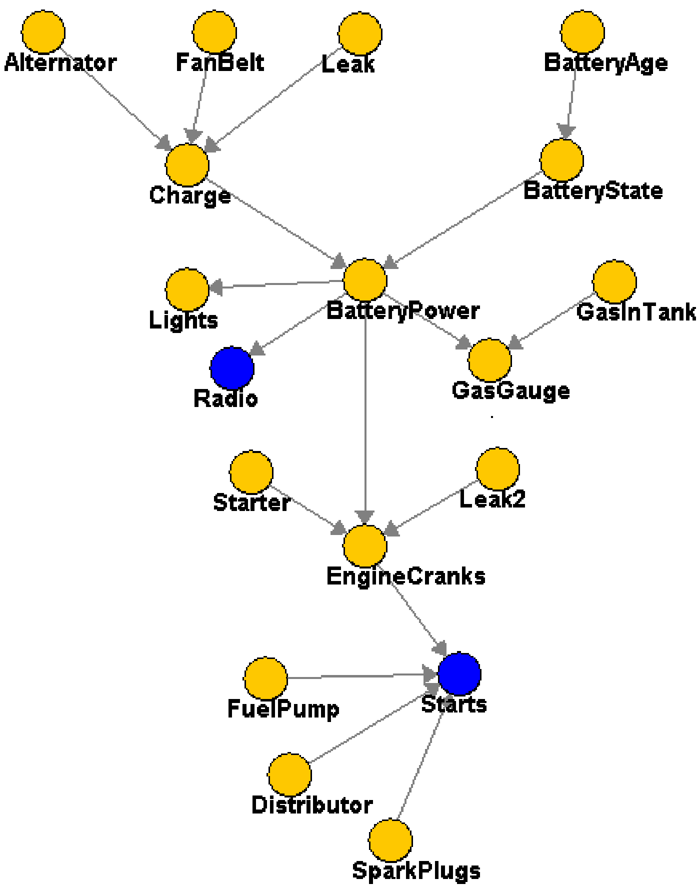
Marginalization:

Complexity of VE – First analysis

- Number of multiplications:

- Number of additions:

Complexity of variable elimination – (Poly)-tree graphs

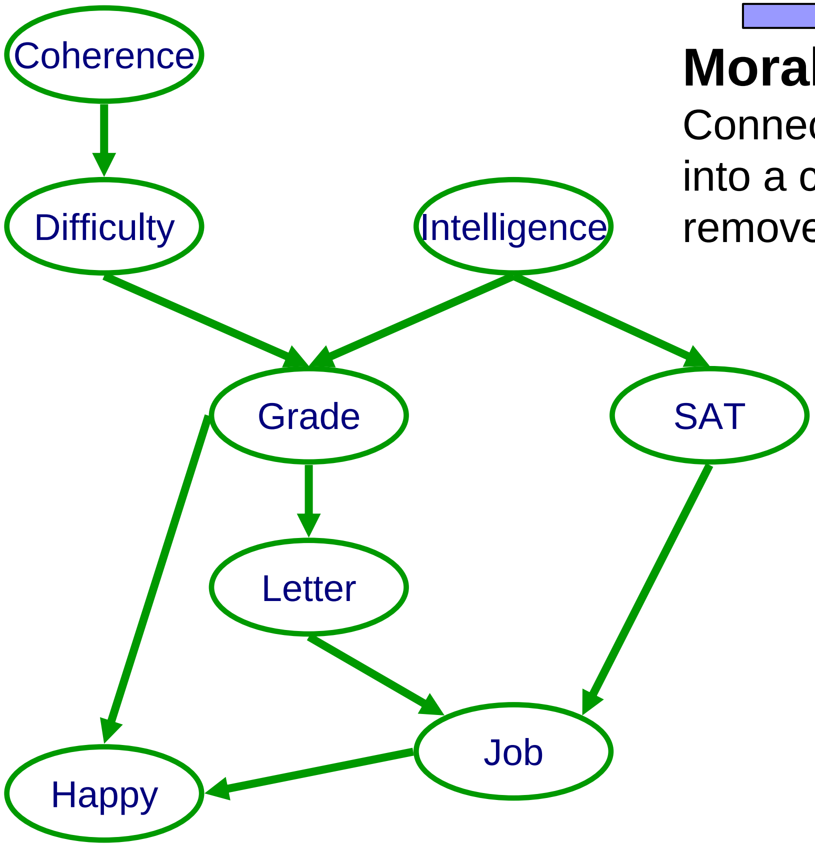


Variable elimination order:
Start from "leaves" inwards:

- Start from skeleton!
- Choose a "root", any node
- Find topological order for root
- Eliminate variables in reverse order

Linear in CPT sizes!!! (versus exponential)

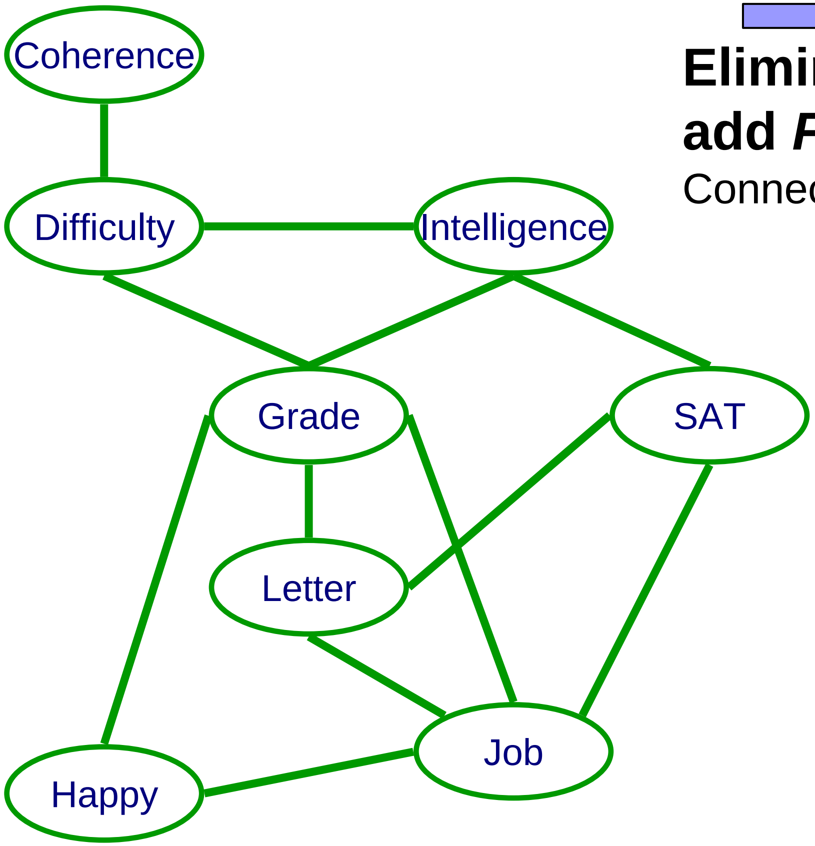
Complexity of variable elimination – Graphs with loops



Moralize graph:
Connect parents
into a clique and
remove edge directions

Connect nodes that appear together in an initial factor

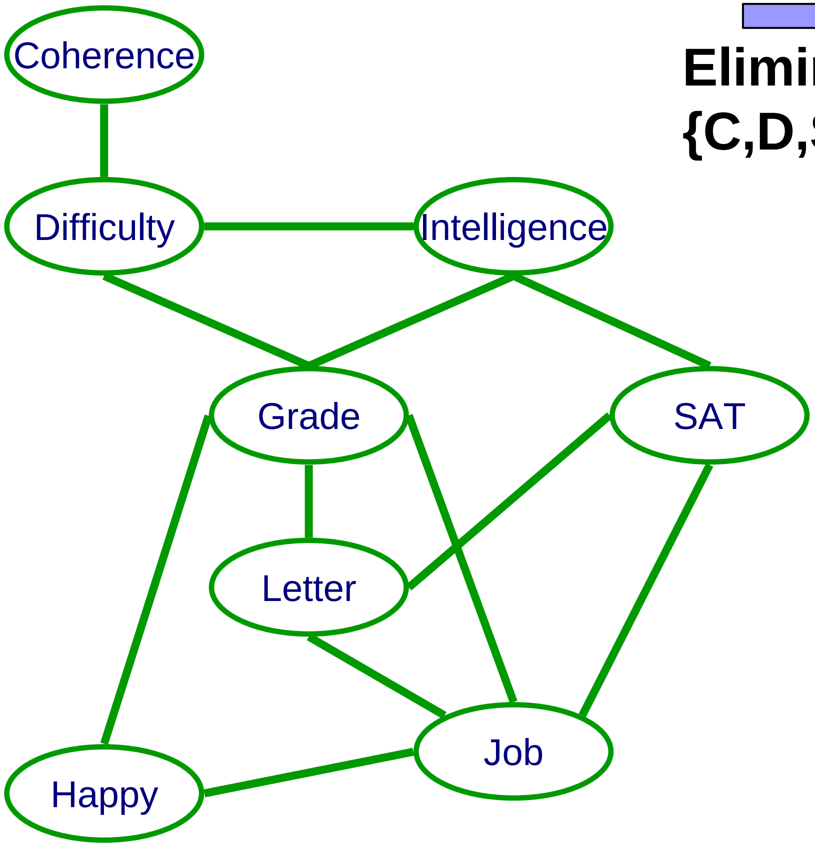
Eliminating a node – Fill edges



Eliminate variable
add *Fill Edges*:
Connect neighbors

Induced graph

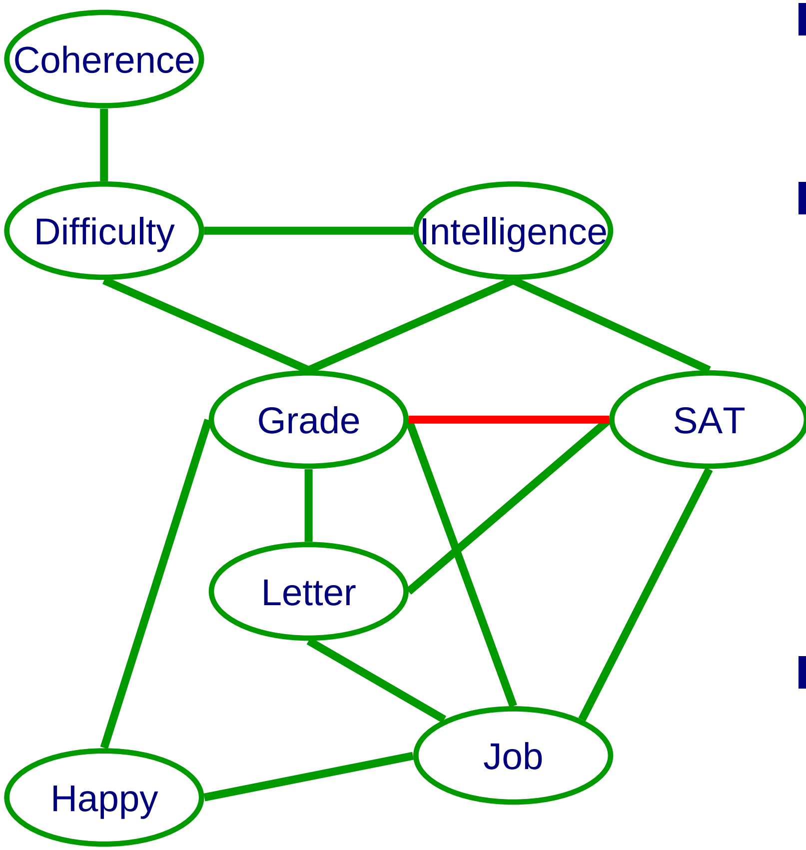
The **induced graph $I_{F \prec}$** for elimination order $\prec$ has an edge $X_i - X_j$ if X_i and X_j appear together in a factor generated by VE for elimination order $\prec$ on factors F



Elimination order: {C,D,S,I,L,H,J,G}

Induced graph and complexity of VE

Read complexity from cliques in induced graph



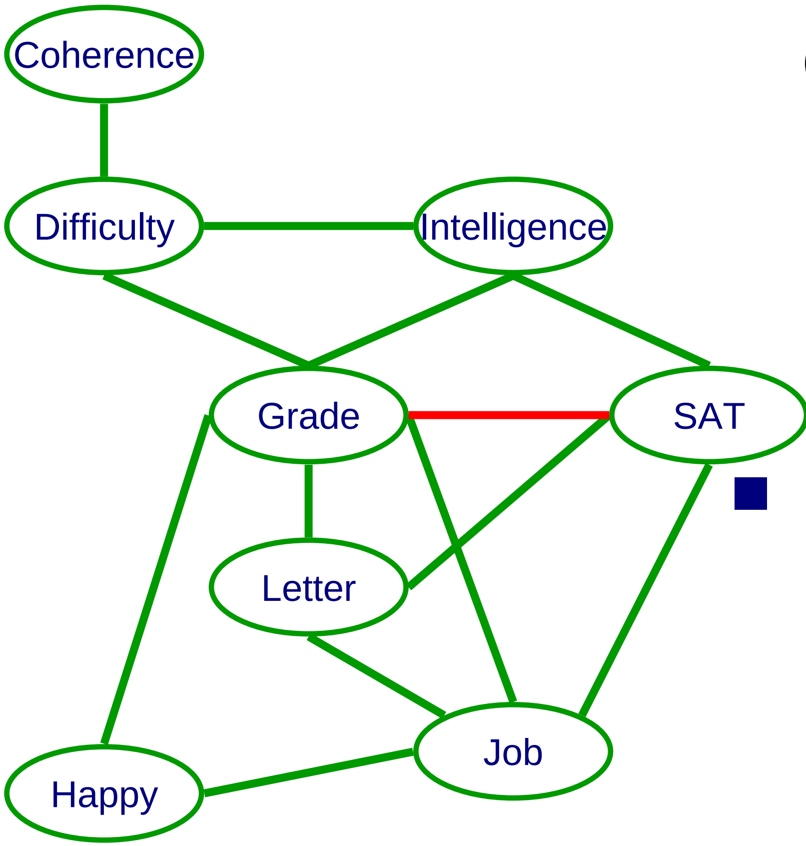
Elimination order:
{C,D,I,S,L,H,J,G}

- Structure of induced graph encodes complexity of VE!!!
- **Theorem**:
  - Every factor generated by VE subset of a maximal clique in $I_{F\prec}$
  - For every maximal clique in $I_{F\prec}$ corresponds to a factor generated by VE
- **Induced width** (or treewidth)
  - Size of largest clique in $I_{F\prec}$ minus 1
  - *Minimal induced width* – induced width of best order $\prec$

Example: Large induced-width with small number of parents

Compact representation ⇏ Easy inference ☹

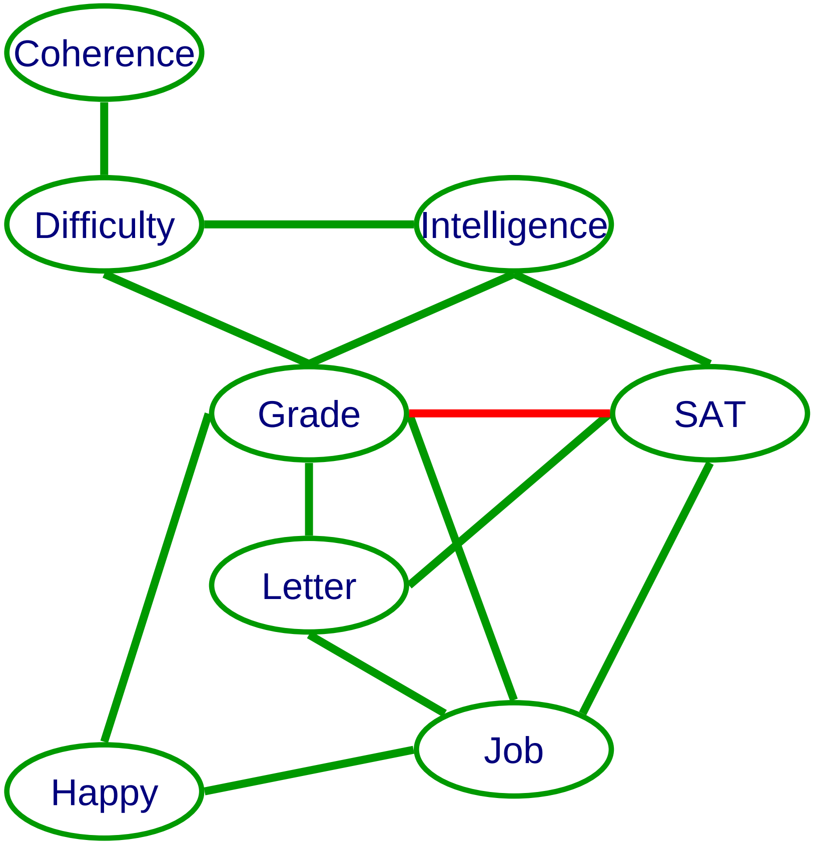
Finding optimal elimination order



Elimination order:
{C,D,I,S,L,H,J,G}

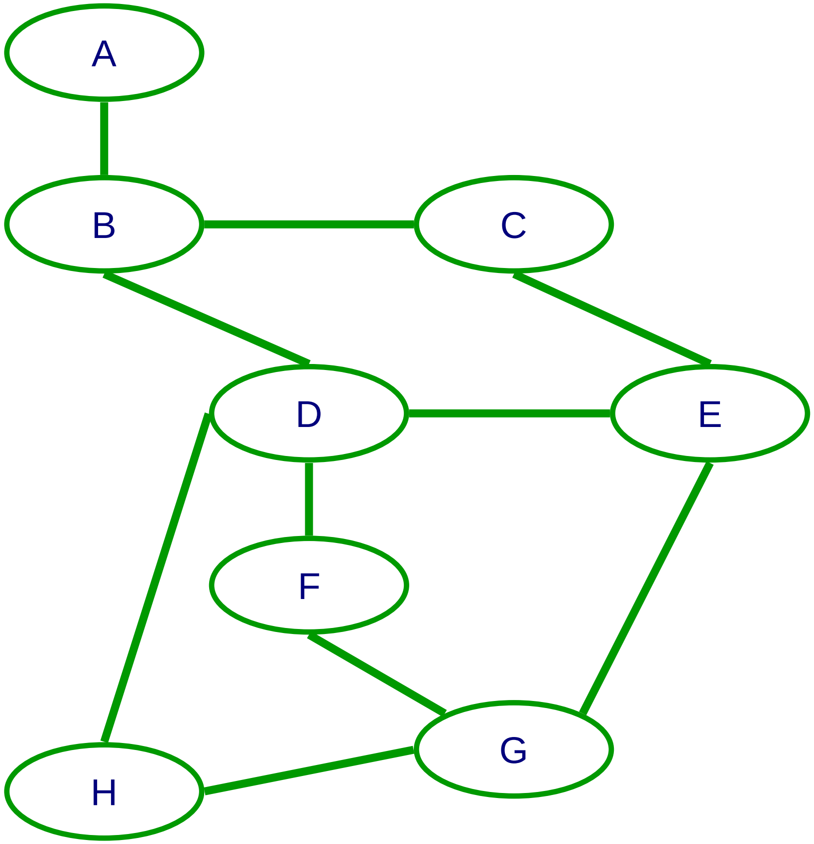
- **Theorem**: Finding best elimination order is NP-complete:
  - Decision problem: Given a graph, determine if there exists an elimination order that achieves induced width $\leq$ K
- **Interpretation**:
  - Hardness of elimination order "orthogonal" to hardness of inference
  - Actually, can find elimination order in time exponential in size of largest clique – same complexity as inference (next week)

Induced graphs and chordal graphs



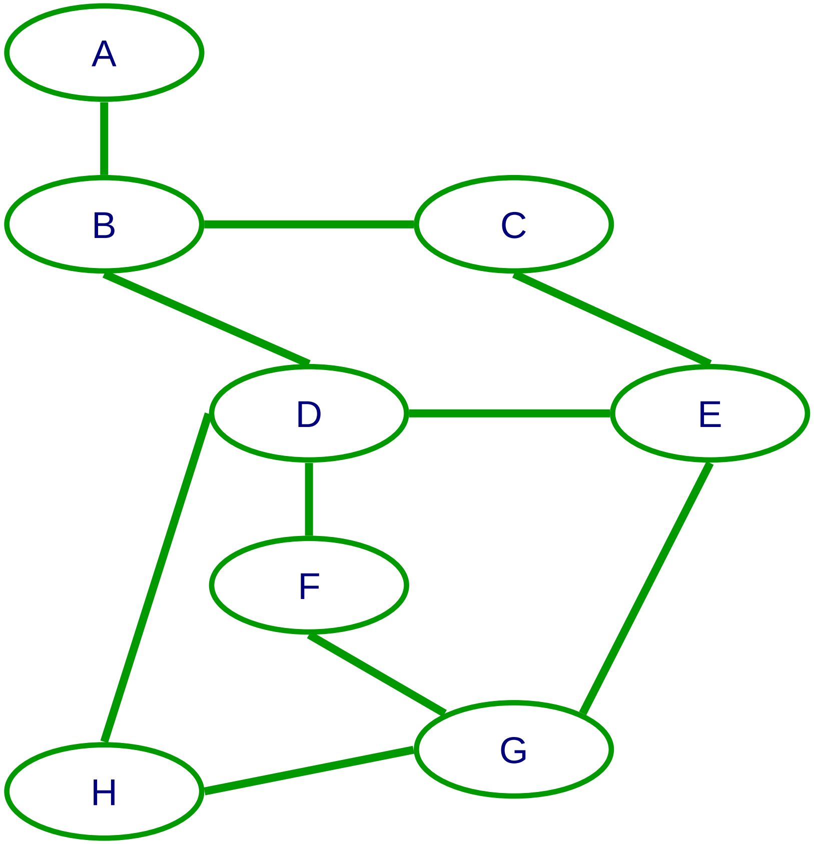
- **Chordal graph**:
  - Every cycle $X_1 - X_2 - \ldots - X_k - X_1$ with $k \geq 3$ has a chord
  - Edge $X_i - X_j$ for non-consecutive i & j
- **Theorem**:
  - Every induced graph is chordal
- "Optimal" elimination order easily obtained for chordal graph

Chordal graphs and triangulation



- **Triangulation**: turning graph into chordal graph
- **Max Cardinality Search**:
  - Simple heuristic
- Initialize unobserved nodes **X** as unmarked
- For k = |**X**| to 1
  - X ← unmarked var with most marked neighbors
  - $\prec$(X) ← k
  - Mark X
- **Theorem**: Obtains optimal order for chordal graphs
- Often, not so good in other graphs!

Minimum fill/size/weight heuristics

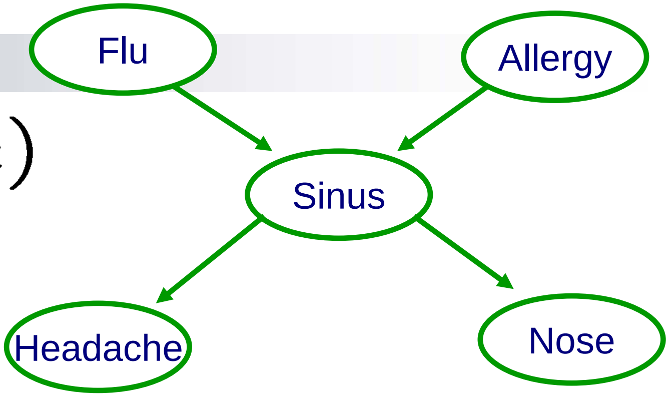


- Many more effective heuristics
  - page 262 of K&F
- **Min (weighted) fill heuristic**
  - Often very effective
- Initialize unobserved nodes **X** as unmarked
- For k = 1 to |**X**|
  - X $\leftarrow$ unmarked var whose elimination adds fewest edges
  - $\prec$(X) $\leftarrow$ k
  - Mark X
  - Add fill edges introduced by eliminating X
- Weighted version:
  - Consider size of factor rather than number of edges

Choosing an elimination order

- Choosing best order is NP-complete
  - Reduction from MAX-Clique
- Many good heuristics (some with guarantees)
- Ultimately, can't beat NP-hardness of inference
  - Even optimal order can lead to exponential variable elimination computation
- In practice
  - Variable elimination often very effective
  - Many (many many) approximate inference approaches available when variable elimination too expensive
  - Most approximate inference approaches build on ideas from variable elimination

Most likely explanation (MLE)



- Query: $\underset{x_1,\ldots,x_n}{\text{argmax}} P(x_1,\ldots,x_n \mid e)$

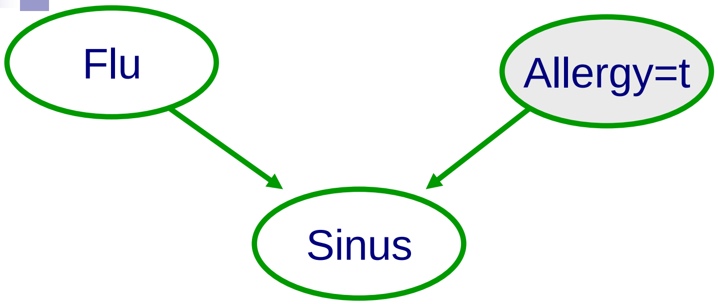
- Using Bayes rule:

$$\underset{x_1,\ldots,x_n}{\text{argmax}} P(x_1,\ldots,x_n \mid e) = \underset{x_1,\ldots,x_n}{\text{argmax}} \frac{P(x_1,\ldots,x_n,e)}{P(e)}$$

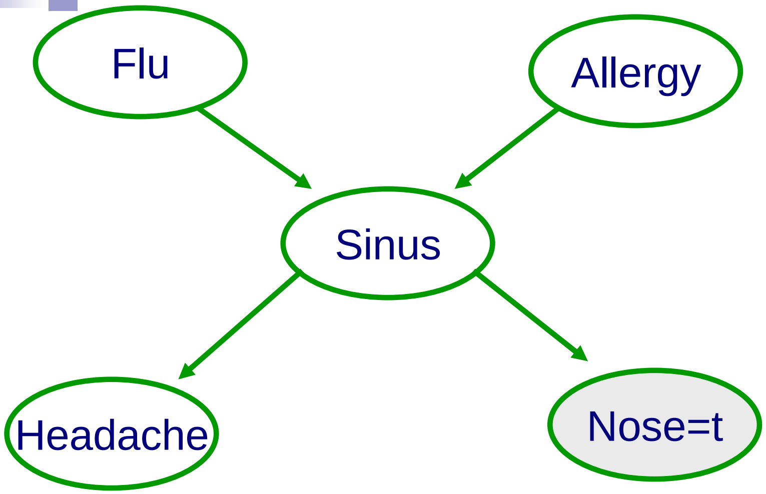
- Normalization irrelevant:

$$\underset{x_1,\ldots,x_n}{\text{argmax}} P(x_1,\ldots,x_n \mid e) = \underset{x_1,\ldots,x_n}{\text{argmax}} P(x_1,\ldots,x_n,e)$$

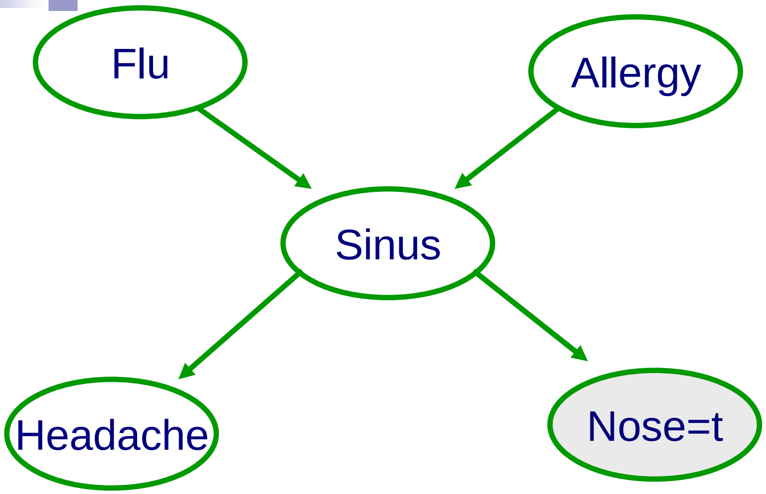
Max-marginalization



Example of variable elimination for MLE – Forward pass



Example of variable elimination for MLE – Backward pass



MLE Variable elimination algorithm – Forward pass

- Given a BN and a MLE query $\max_{x_1,\ldots,x_n} P(x_1,\ldots,x_n,\mathbf{e})$
- Instantiate evidence **E**=**e**
- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- For i = 1 to n, If $X_i \notin \mathbf{E}$
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

$$g = \max_{x_i} \prod_{j=1}^{k} f_j$$

  - Variable X_i has been eliminated!

MLE Variable elimination algorithm – Backward pass

- $\{x_1^*, \ldots, x_n^*\}$ will store maximizing assignment
- For i = n to 1, If $X_i \notin \mathbf{E}$
  - Take factors $f_1, \ldots, f_k$ used when X_i was eliminated
  - Instantiate $f_1, \ldots, f_k$, with $\{x_{i+1}^*, \ldots, x_n^*\}$
    - Now each f_j depends only on X_i
  - Generate maximizing assignment for X_i:

$$x_i^* \in \underset{x_i}{\text{argmax}} \prod_{j=1}^{k} f_j$$

What you need to know

- Variable elimination algorithm
  - Eliminate a variable:
    - Combine factors that include this var into single factor
    - Marginalize var from new factor
  - Cliques in induced graph correspond to factors generated by algorithm
  - Efficient algorithm ("only" exponential in induced-width, not number of variables)
    - If you hear: "Exact inference only efficient in tree graphical models"
    - You say: "No!!! Any graph with low induced width"
    - And then you say: "And even some with very large induced-width" (next week)
- Elimination order is important!
  - NP-complete problem
  - Many good heuristics
- Variable elimination for MLE
  - Only difference between probabilistic inference and MLE is "sum" versus "max"