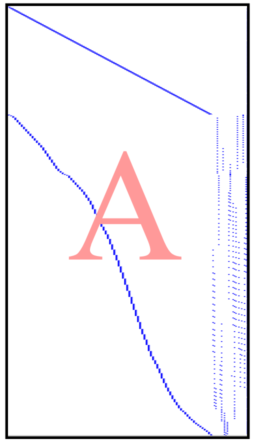
# SLAM and Graphical Models

# Robot Localization and Mapping
# 16-833

Michael Kaess

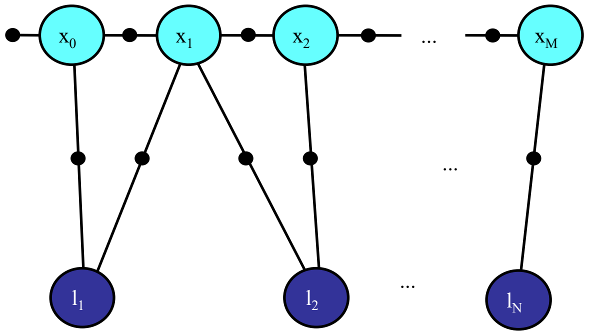
November 4+11, 2024

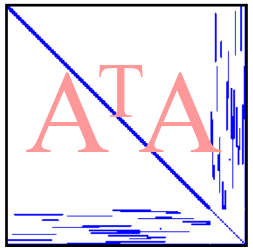
# Matrix vs. Graph




Measurement Jacobian

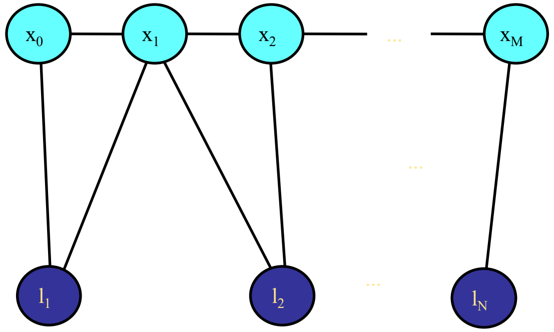
Factor Graph

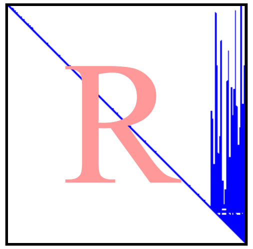



Information Matrix

Markov Random Field

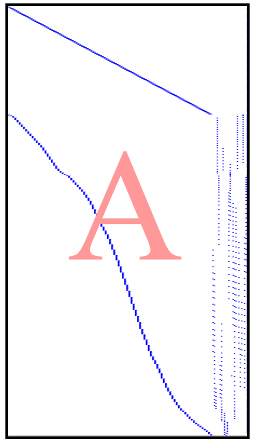



Square Root Inf. Matrix

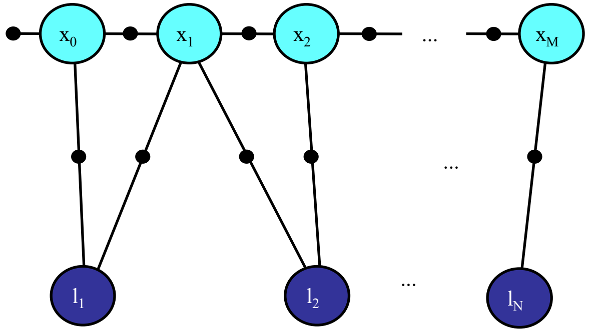
?

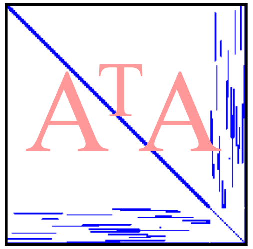
# Matrix vs. Graph




Measurement Jacobian

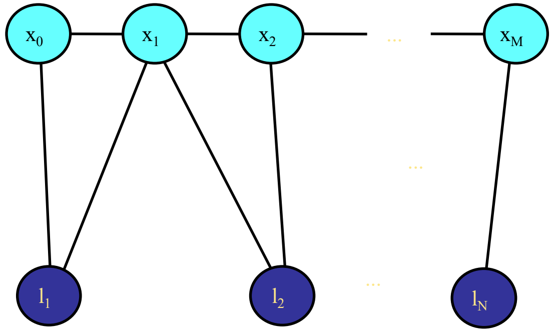
Factor Graph

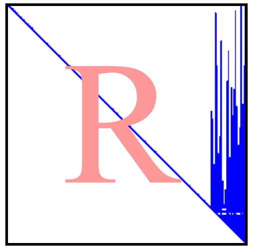



Information Matrix

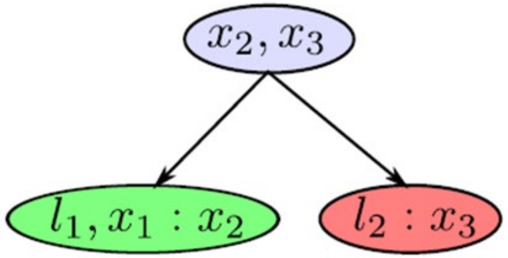
Markov Random Field



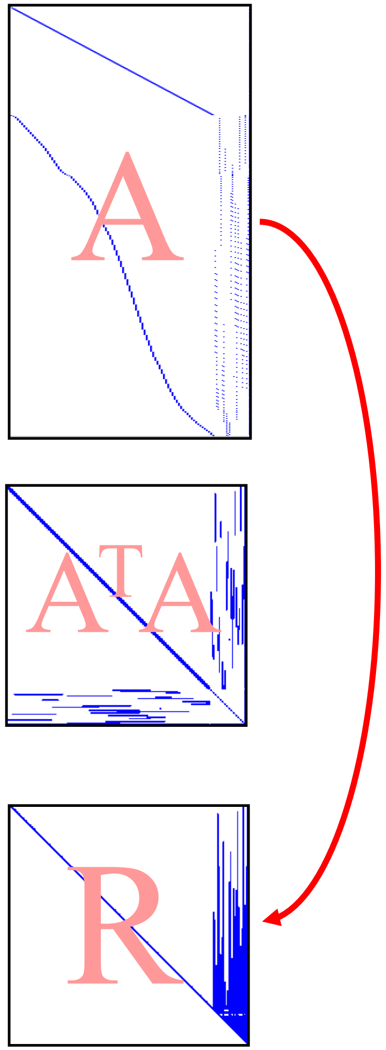




Square Root Inf. Matrix

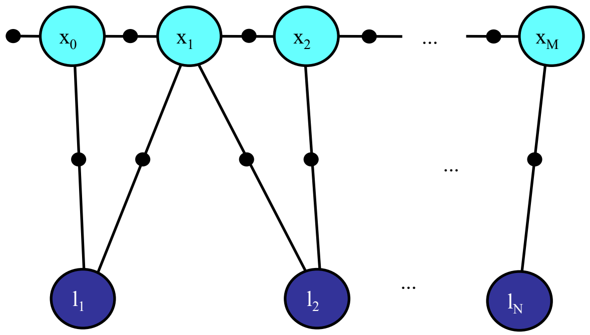
**Bayes Tree**

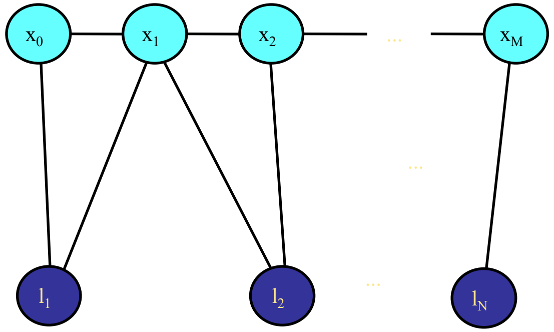


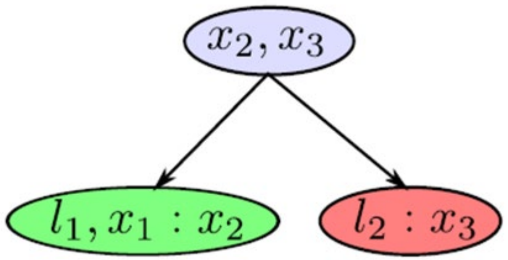
# Matrix vs. Graph



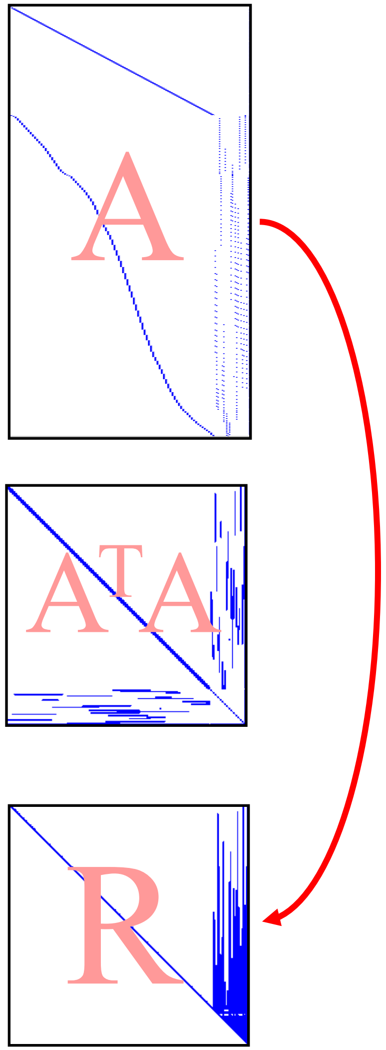

Matrix factorization



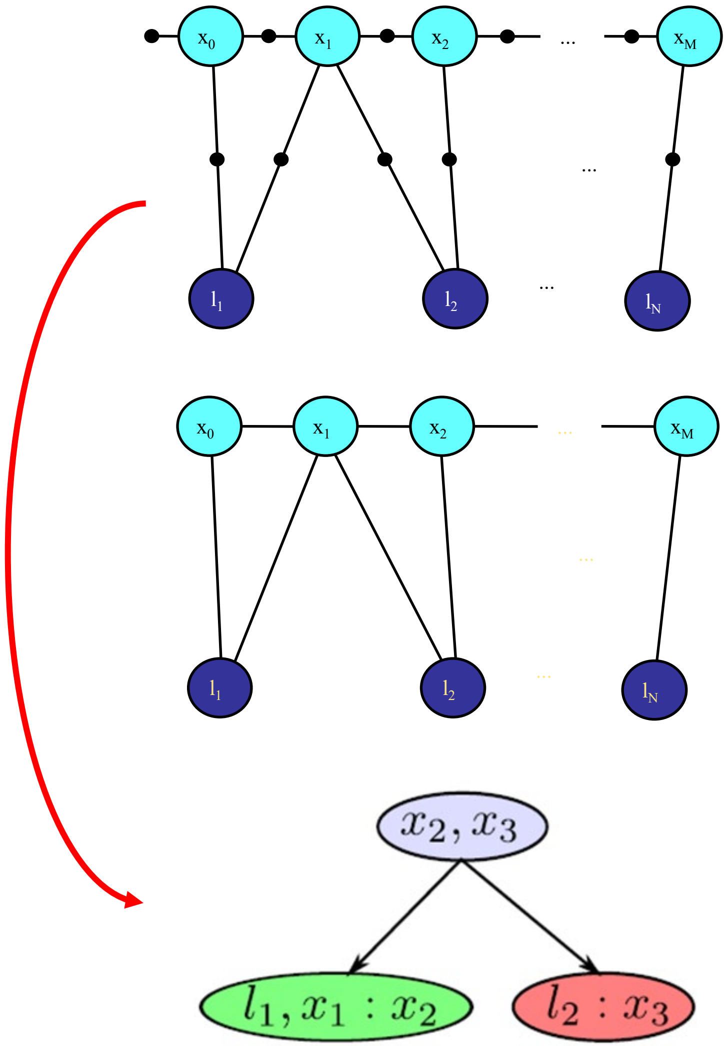



# Matrix vs. Graph




Matrix factorization

Variable elimination



# iSAM2: Bayes Tree

Inference in tree structure is easy

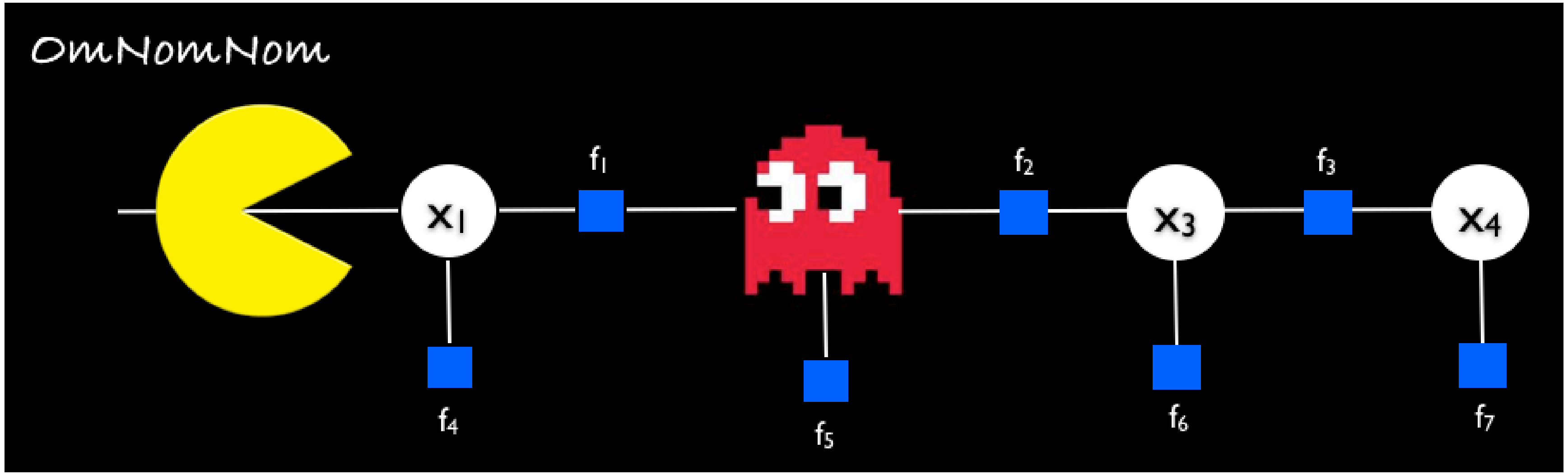
Idea: Convert factor graph to tree structure

Two stage process:

- Variable elimination converts factor graph to Bayes net
- Discovering cliques provides Bayes tree

“iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree”
M. Kaess, H. Johannsson, R. Roberts, V. Ila, J.J. Leonard, and F. Dellaert.
IJRR 2012

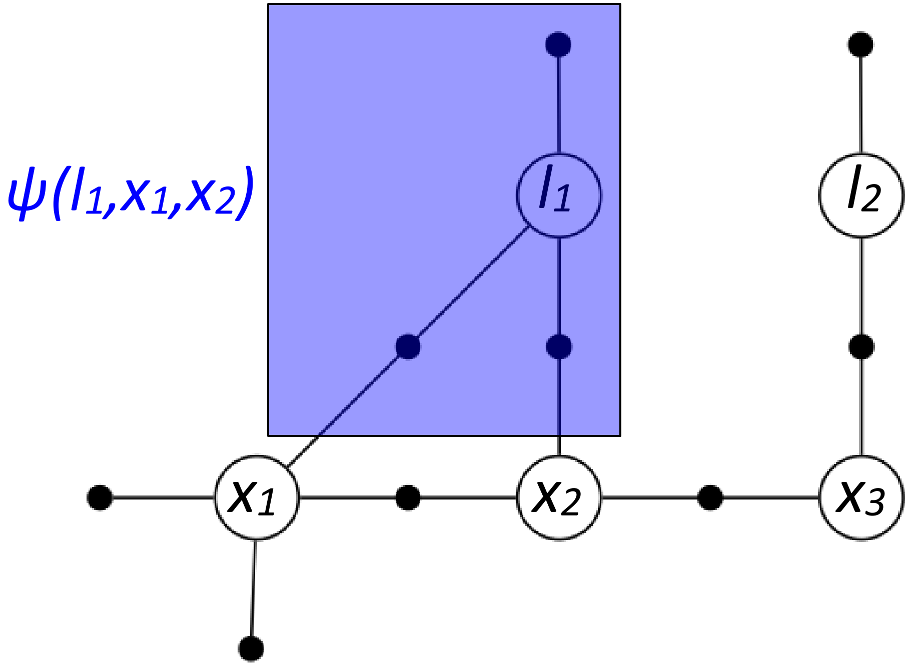
# Variable Elimination

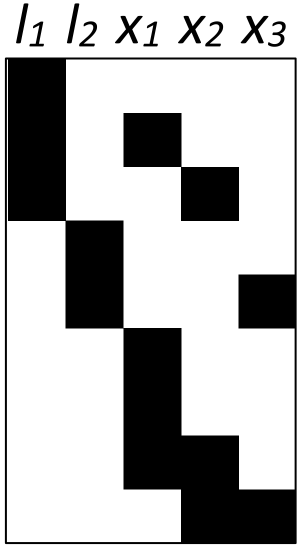



Courtesy of Daniel Kohlsdorf (Georgia Tech)

# Variable Elimination – Example

- Choose ordering: $l_1, l_2, x_1, x_2, x_3$
- Eliminate one node at a time



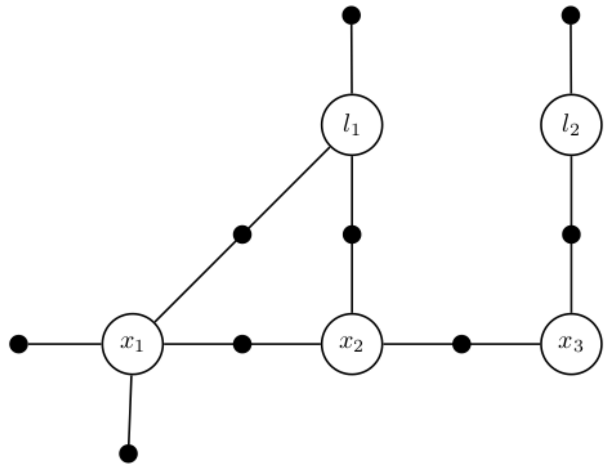




$$\psi(l_1,x_1,x_2) = \psi(l_1 | x_1,x_2)\ \psi(x_1,x_2)$$

# Variable Elimination

Factorization of the factors connected to $l_1$

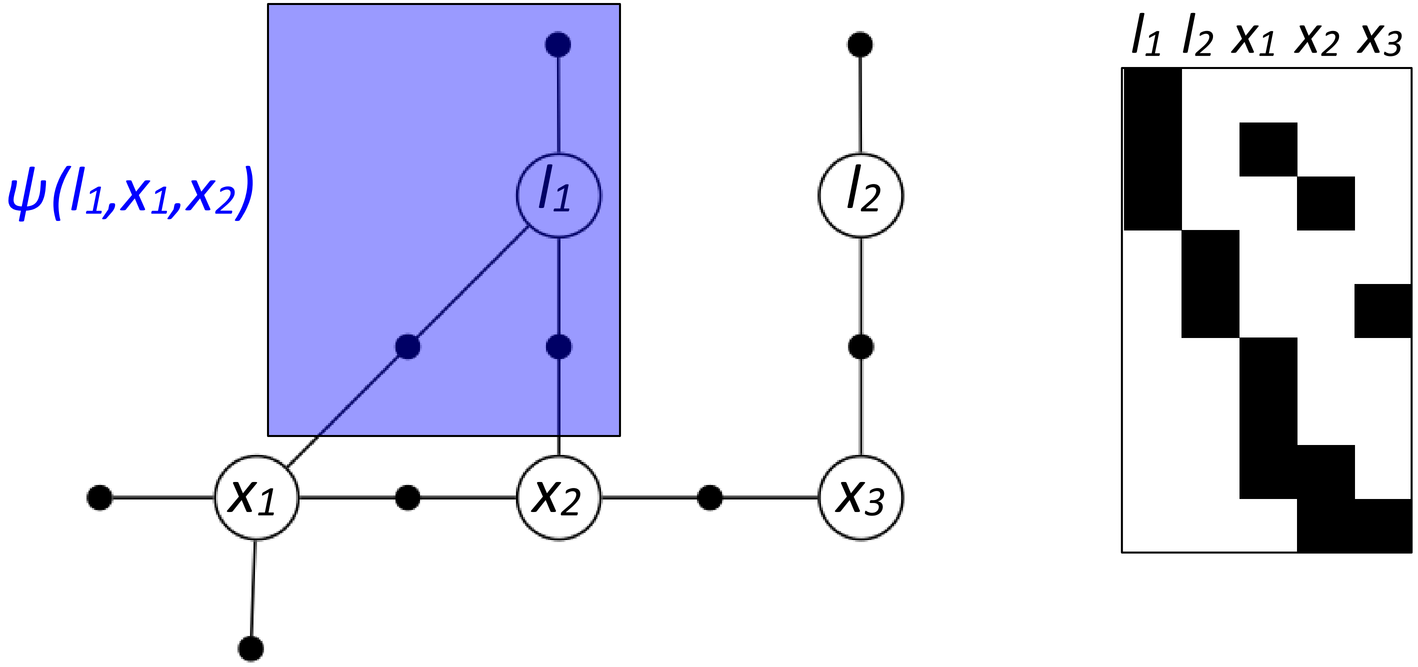
(on the board)




$$[A|b] = \begin{array}{c c} & \begin{array}{cccccc} \delta l_1 & \delta l_2 & \delta x_1 & \delta x_2 & \delta x_3 & rhs \end{array} \\ \begin{array}{c} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{array} & \left[\begin{array}{ccccc|c} & & A_{13} & & & b_1 \\ & & A_{23} & A_{24} & & b_2 \\ & & & A_{34} & A_{35} & b_3 \\ A_{41} & & & & & b_4 \\ & A_{52} & & & & b_5 \\ & & A_{63} & & & b_6 \\ A_{71} & & A_{73} & & & b_7 \\ A_{81} & & & A_{84} & & b_8 \\ & A_{92} & & & A_{95} & b_9 \end{array}\right] \end{array}$$

# Variable Elimination – Example

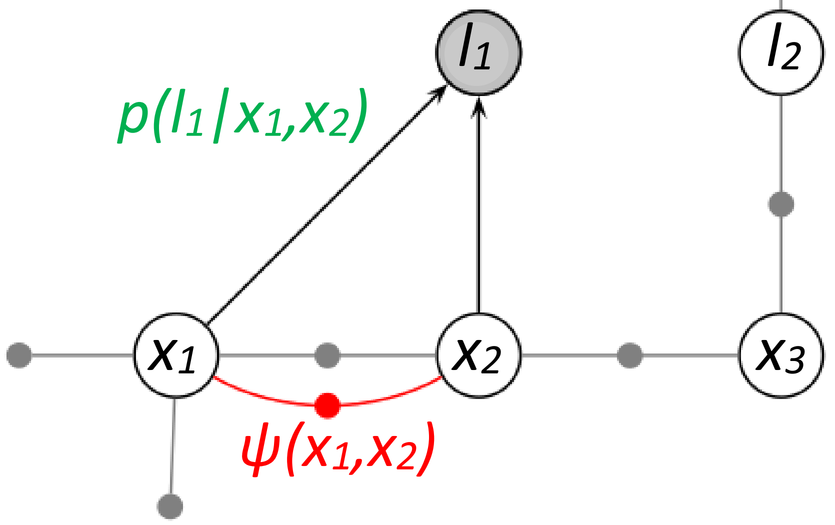
- Choose ordering: $l_1, l_2, x_1, x_2, x_3$
- Eliminate one node at a time

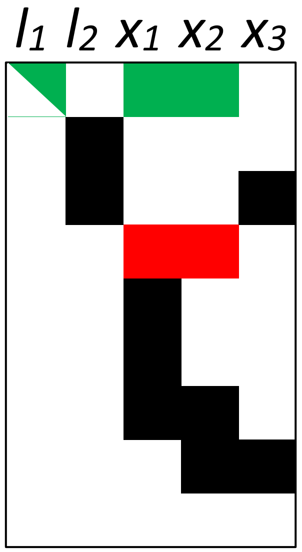



$$\psi(l_1,x_1,x_2) = \psi(l_1|x_1,x_2)\ \psi(x_1,x_2)$$

# Variable Elimination – Example

- Choose ordering: $l_1, l_2, x_1, x_2, x_3$
- Eliminate one node at a time

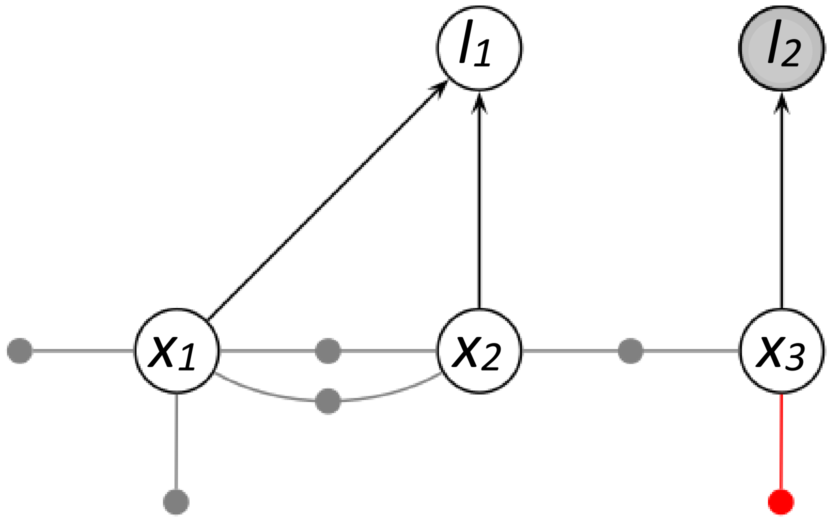


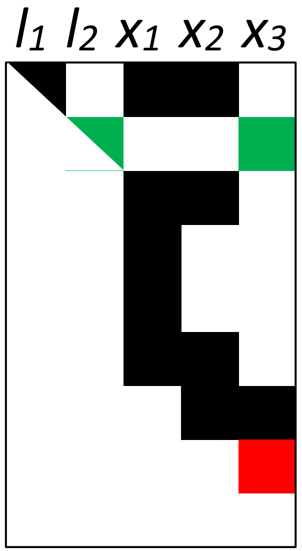




$\psi(l_1,x_1,x_2) = \psi(l_1|x_1,x_2)\ \psi(x_1,x_2)$

# Variable Elimination – Example

- Choose ordering: $l_1, l_2, x_1, x_2, x_3$
- Eliminate one node at a time

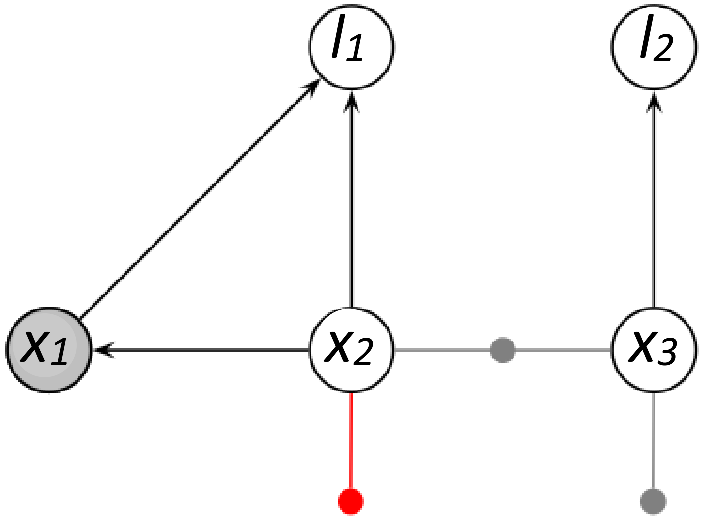


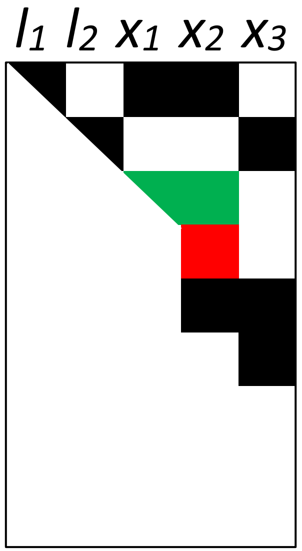




$$\psi(l_2,x_3) = \psi(l_2|x_3)\ \psi(x_3)$$

# Variable Elimination – Example

- Choose ordering: $l_1, l_2, x_1, x_2, x_3$
- Eliminate one node at a time



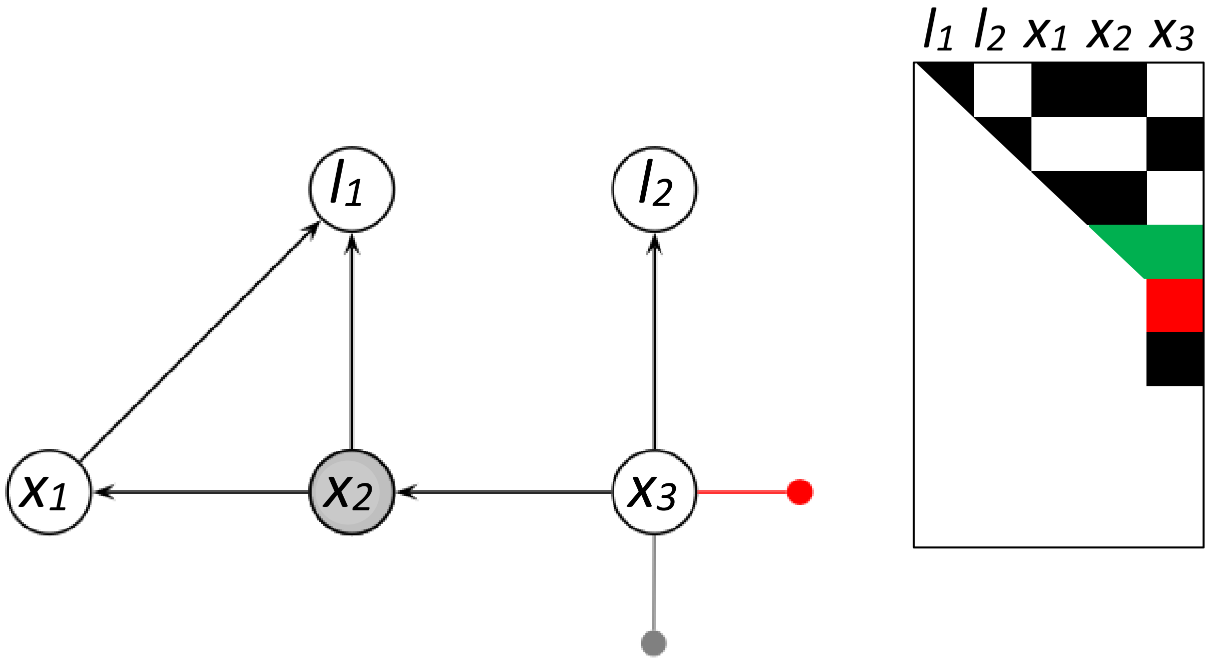




$\psi(x_1,x_2) = \psi(x_1|x_2)\ \psi(x_2)$

[Kaess et al., IJRR 12]

# Variable Elimination – Example

- Choose ordering: $l_1, l_2, x_1, x_2, x_3$
- Eliminate one node at a time

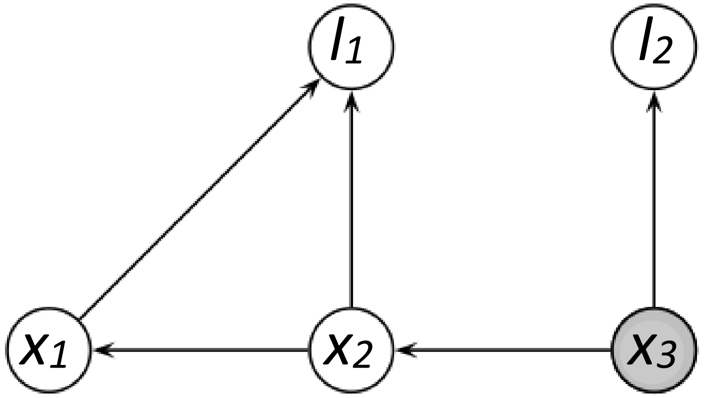


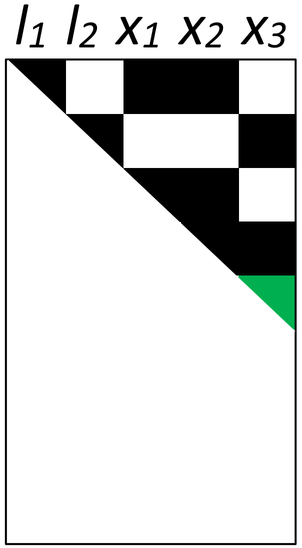

$\psi(x_2,x_3) = \psi(x_2|x_3)\ \psi(x_3)$

[Kaess et al., IJRR 12]

# Variable Elimination – Example

- Choose ordering: $l_1, l_2, x_1, x_2, x_3$
- Eliminate one node at a time




$\psi(x_3)$

# Variable Elimination – Algorithm

**Algorithm 3.1** The Variable Elimination Algorithm

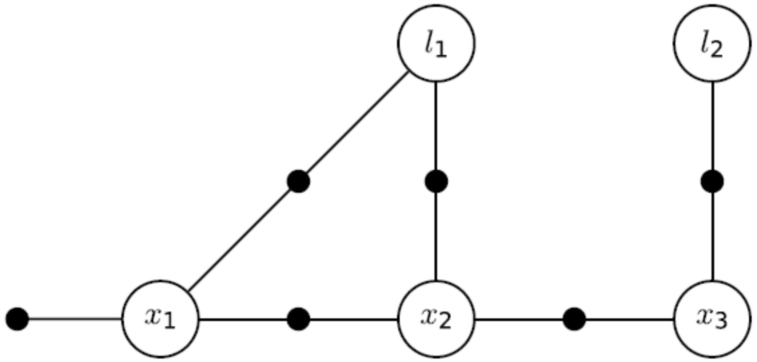
1: **function** ELIMINATE($\Phi_{1:n}$) ▷ given a factor graph on $n$ variables
2: **for** $j = 1...n$ **do** ▷ for all variables
3: $p(x_j|S_j), \Phi_{j+1:n} \leftarrow \text{EliminateOne}(\Phi_{j:n}, x_j)$ ▷ eliminate $x_j$
4: **return** $p(x_1|S_1)p(x_2|S_2)\dots p(x_n)$ ▷ return Bayes net

**Algorithm 3.2** Eliminate variable $x_j$ from a factor graph $\Phi_{j:n}$.

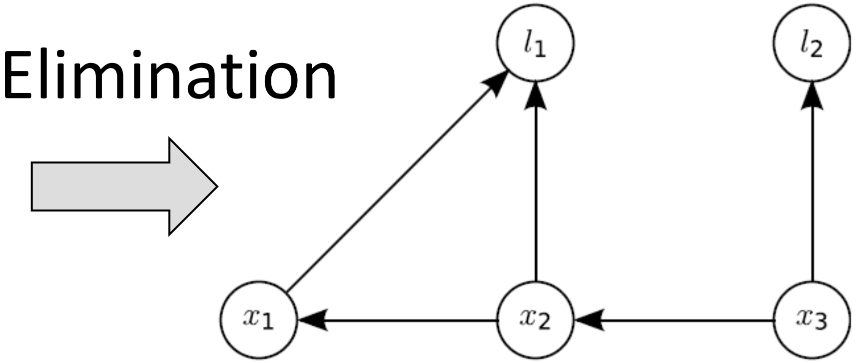
1: **function** ELIMINATEONE($\Phi_{j:n}, x_j$) ▷ given reduced graph $\Phi_{j:n}$
2: Remove all factors $\phi_i(X_i)$ that are adjacent to $x_j$
3: $\mathcal{S}(x_j) \leftarrow$ all variables involved excluding $x_j$ ▷ the separator
4: $\psi(x_j, S_j) \leftarrow \prod_i \phi_i(X_i)$ ▷ create the product factor $\psi$
5: $p(x_j|S_j)\tau(S_j) \leftarrow \psi(x_j, S_j)$ ▷ factorize the product $\psi$
6: Add the new factor $\tau(S_j)$ back into the graph
7: **return** $p(x_j|S_j), \Phi_{j+1:n}$ ▷ Conditional and reduced graph

[Kaess et al., IJRR 12]

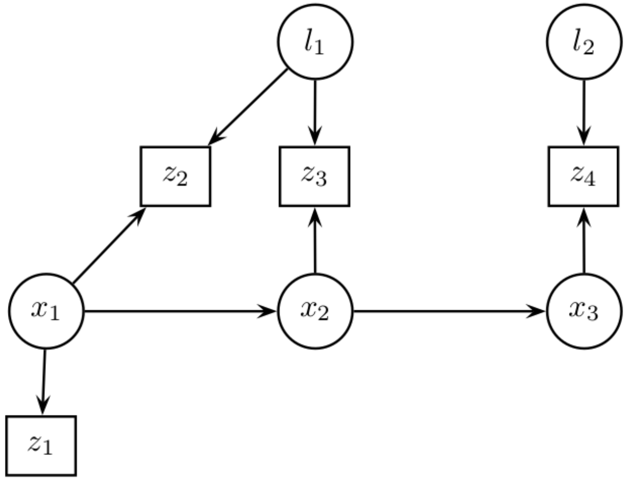
# Bayes Tree Data Structure



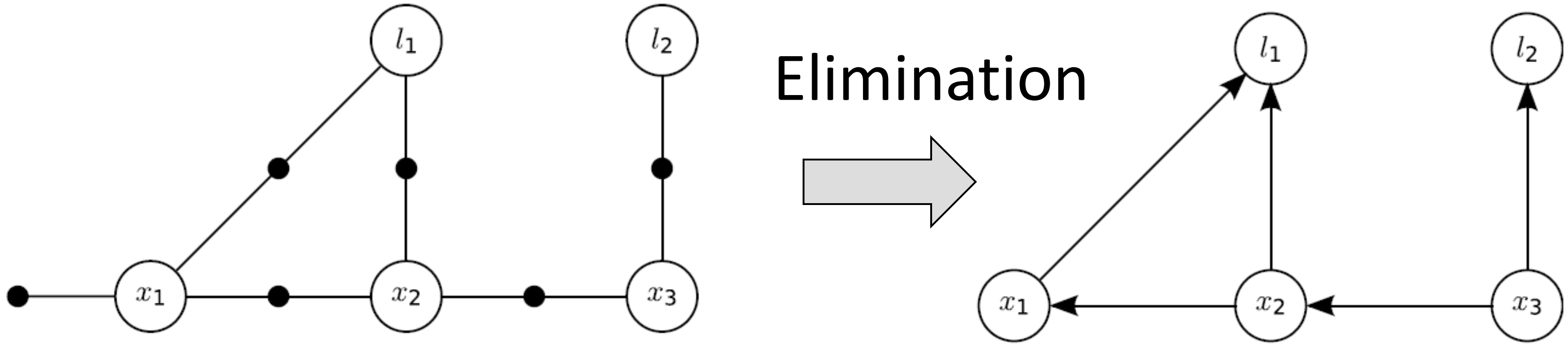

Elimination




Not the same as the original Bayes net!



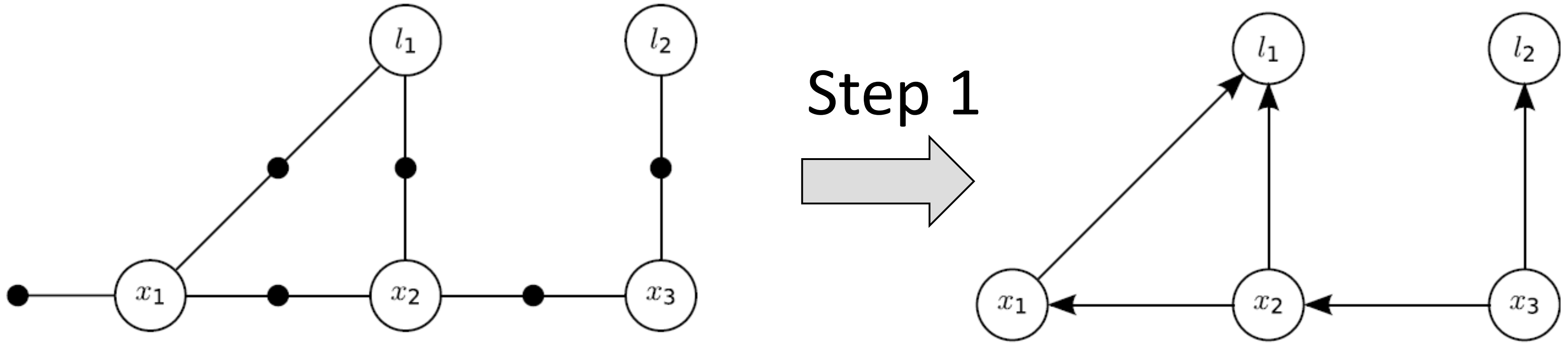
# Bayes Tree Data Structure




The Bayes net has a special property: its undirected equivalent is chordal by construction

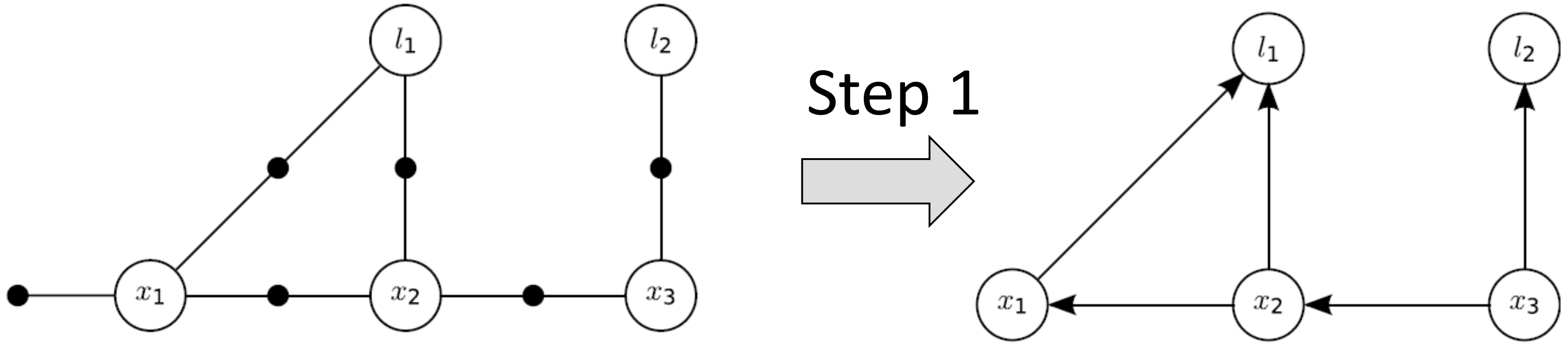
Chordal: There is no cycle greater than 3 that has no shortcut

# Bayes Tree Data Structure

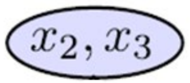



Step 2: Find cliques in reverse elimination order:

# Bayes Tree Data Structure

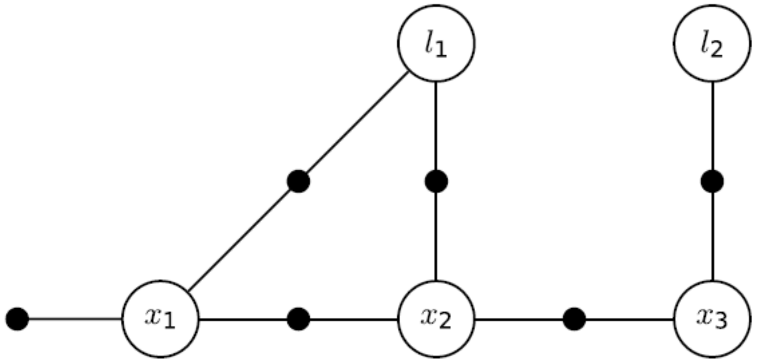



Step 2: Find cliques in reverse elimination order:

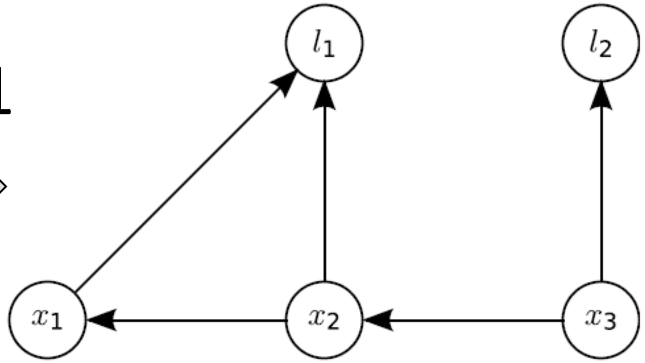


[Kaess et al., IJRR 12]

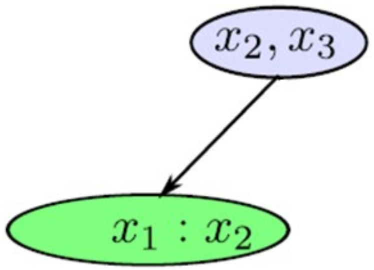
# Bayes Tree Data Structure




Step 1

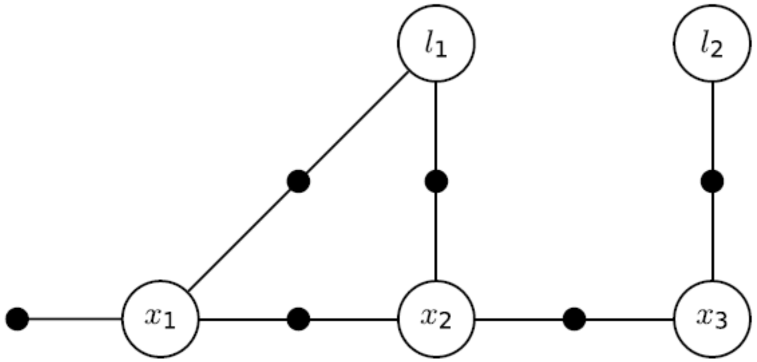



Step 2: Find cliques in reverse elimination order:

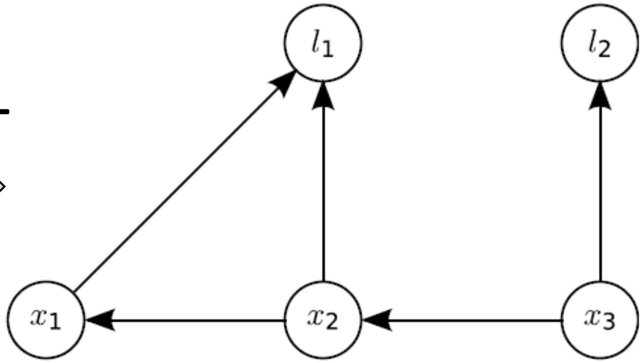


[Kaess et al., IJRR 12]

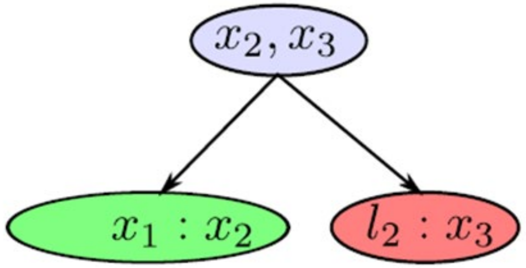
# Bayes Tree Data Structure




Step 1



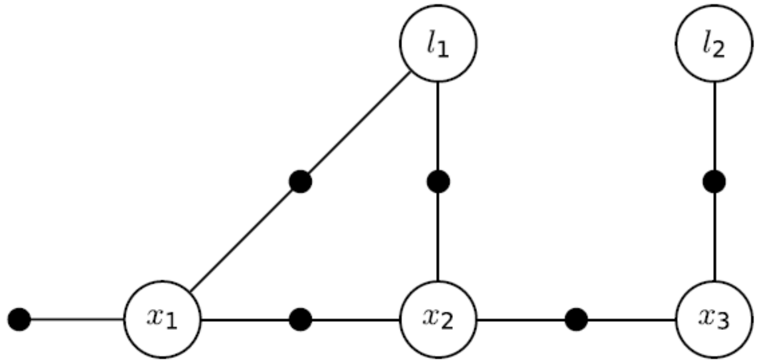

Step 2: Find cliques in reverse elimination order:



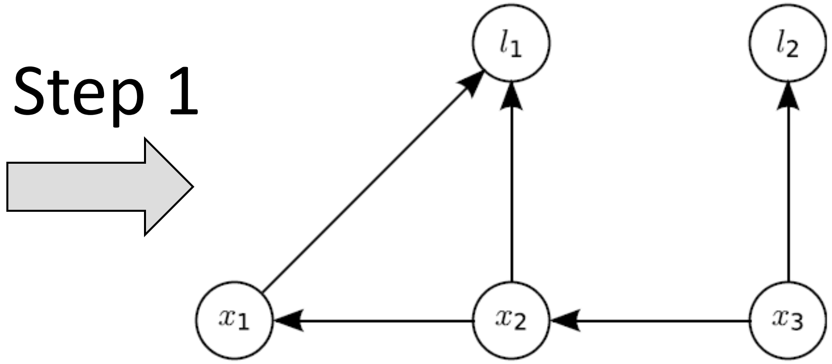


Carnegie Mellon
THE ROBOTICS INSTITUTE

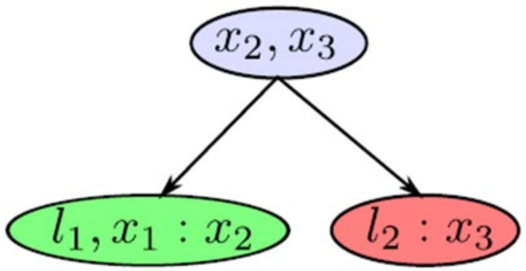
# Bayes Tree Data Structure




Step 1

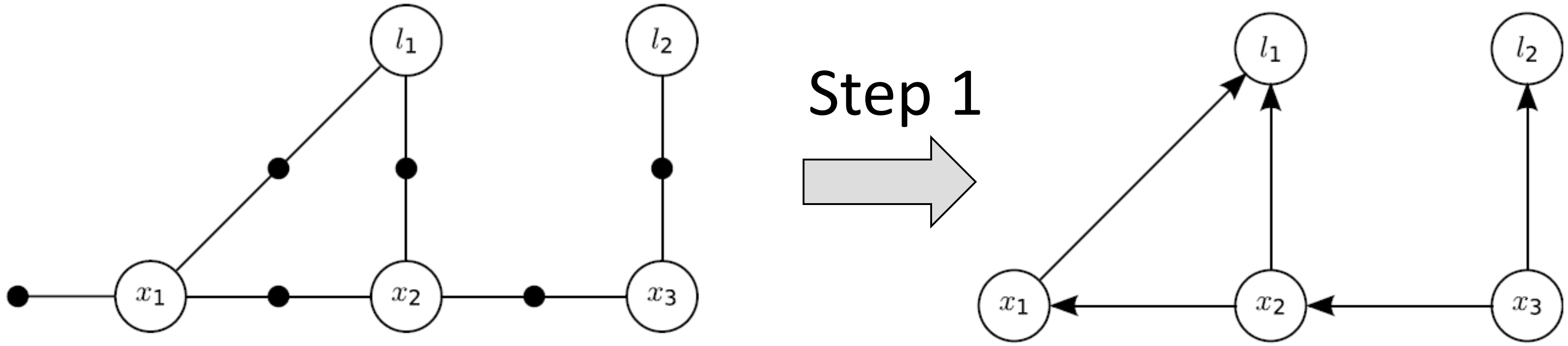



Step 2: Find cliques in reverse elimination order:

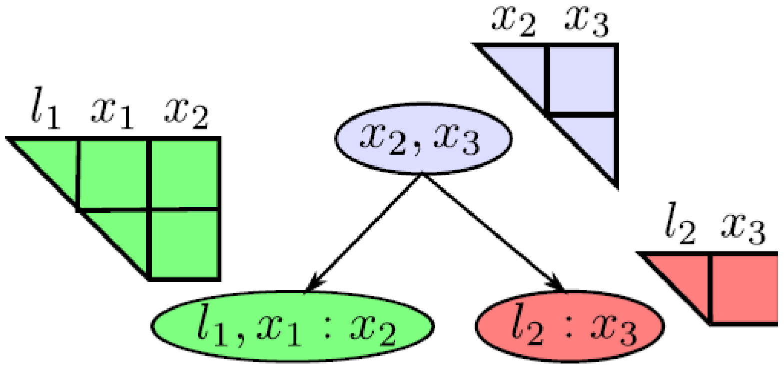


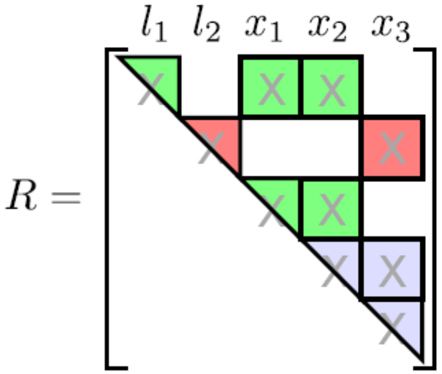
[Kaess et al., IJRR 12]

# Bayes Tree Data Structure




Step 2: Find cliques in reverse elimination order:




$$P(x_j|S_j) \propto \exp\left\{-\frac{1}{2\sigma^2}\left(x_j + rS_j - d\right)^2\right\}$$

# Bayes Tree – Algorithm

**Alg. 3** Creating a Bayes tree from the chordal Bayes net resulting from elimination (Alg. 2).

For each conditional density $P(\theta_j|S_j)$ of the Bayes net, in *reverse* elimination order:
If no parent ($S_j = \{\}$)
    start a new root clique $F_r$ containing $\theta_j$
else
    identify parent clique $C_p$ that contains the first eliminated variable of $S_j$ as a frontal variable
        if nodes $F_p \cup S_p$ of parent clique $C_p$ are equal to separator nodes $S_j$ of conditional
            insert conditional into clique $C_p$
    else
            start new clique $C'$ as child of $C_p$ containing $\theta_j$

# Bayes Tree Example

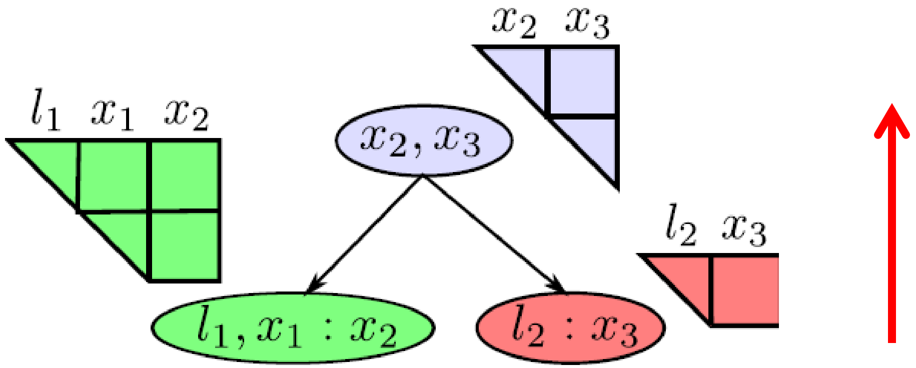
- On the board:
  - Example with 4 nodes, 3 different orderings

# Question

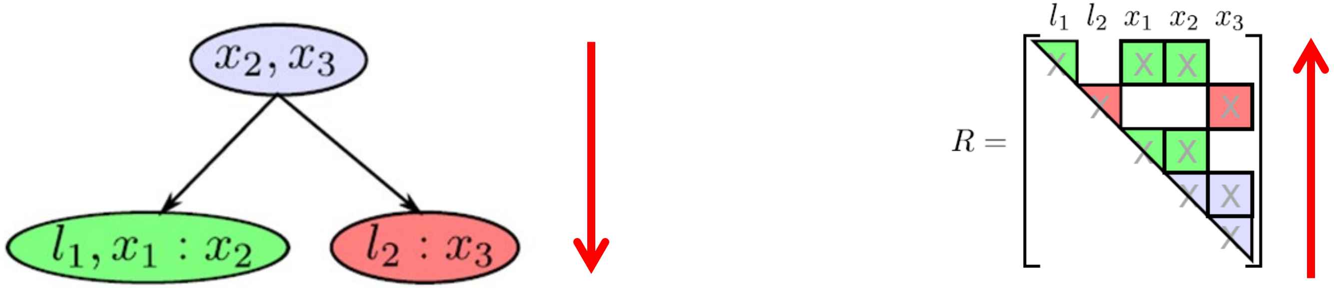
- How to do backsubstitution in the graph?

# Backsubstitution in the Graph

- Inference is a two-step process:
  - Elimination starts at leaves and proceeds to the root




  - Solving starts at root and proceeds to the leaves



# Question

- In the matrix factorization, which entries correspond to the root in the Bayes tree?

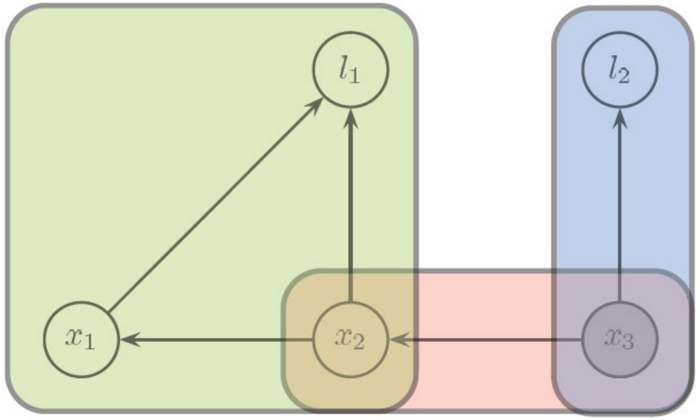
# Question

- In the matrix factorization, which entries correspond to the root in the Bayes tree?
  - The bottom-/right-most entries correspond to the root – they are not conditioned on any other variables

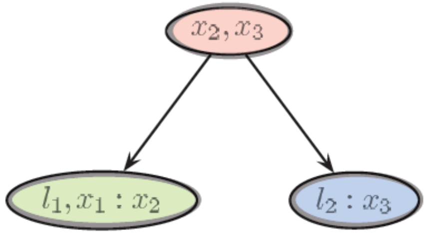
# Bayes Tree vs. Junction Tree/Clique Tree

BT = direct(ed) result from elimination

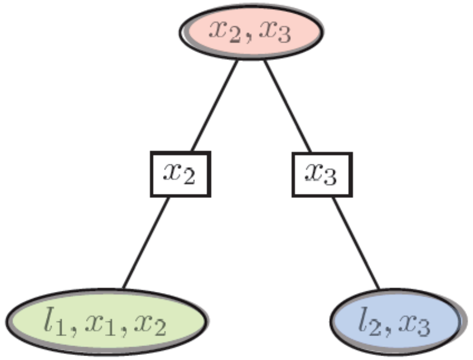
More intuitive, directly encodes square root inform. factor, but also less general: reflects an ordering




Chordal Bayes Net and cliques

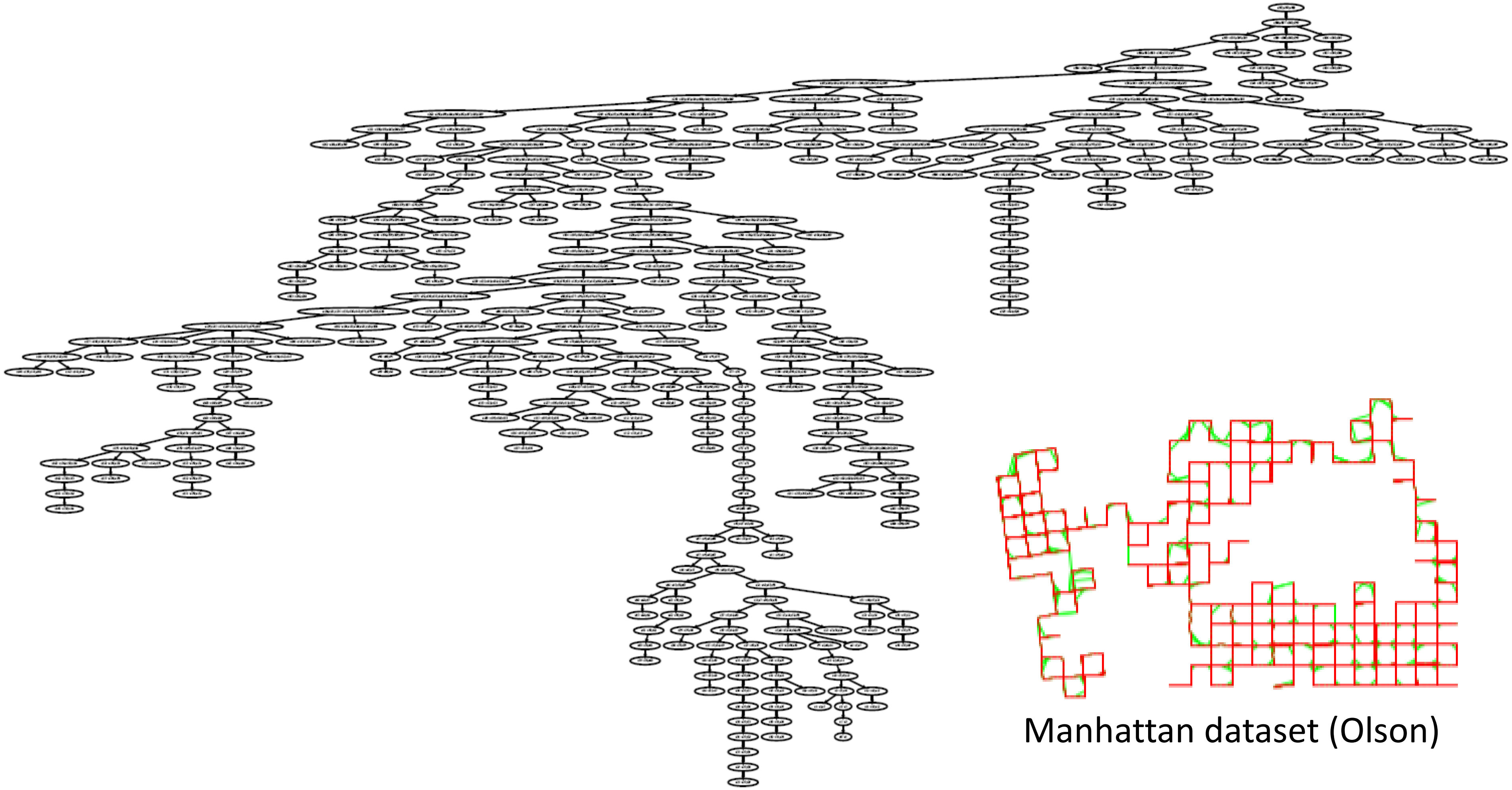



Bayes Tree




Junction Tree

# iSAM2: Bayes Tree Example



Manhattan dataset (Olson)

Complexity depends on the size of the largest clique