Super Nodes

and

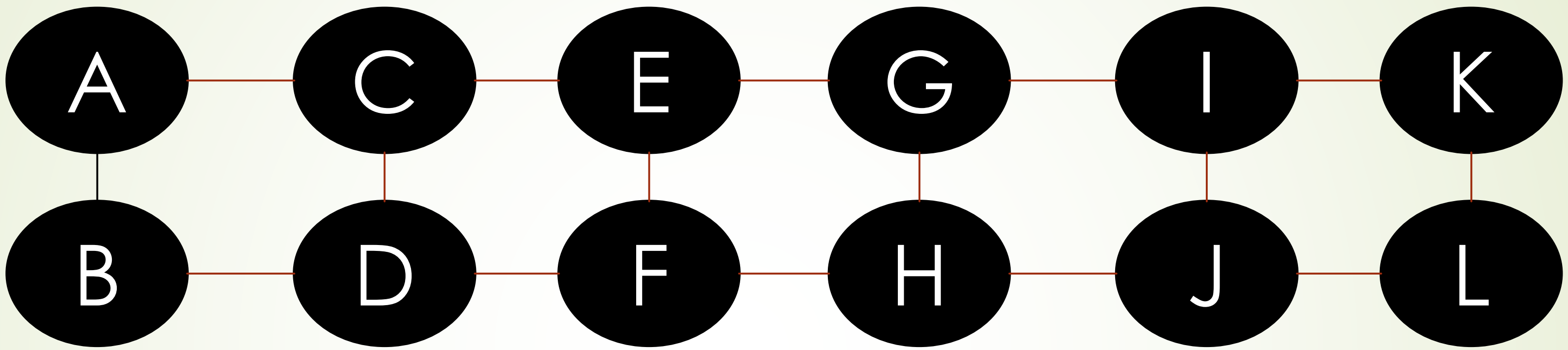
Junction Trees

Presented by: Mehran Kazemi

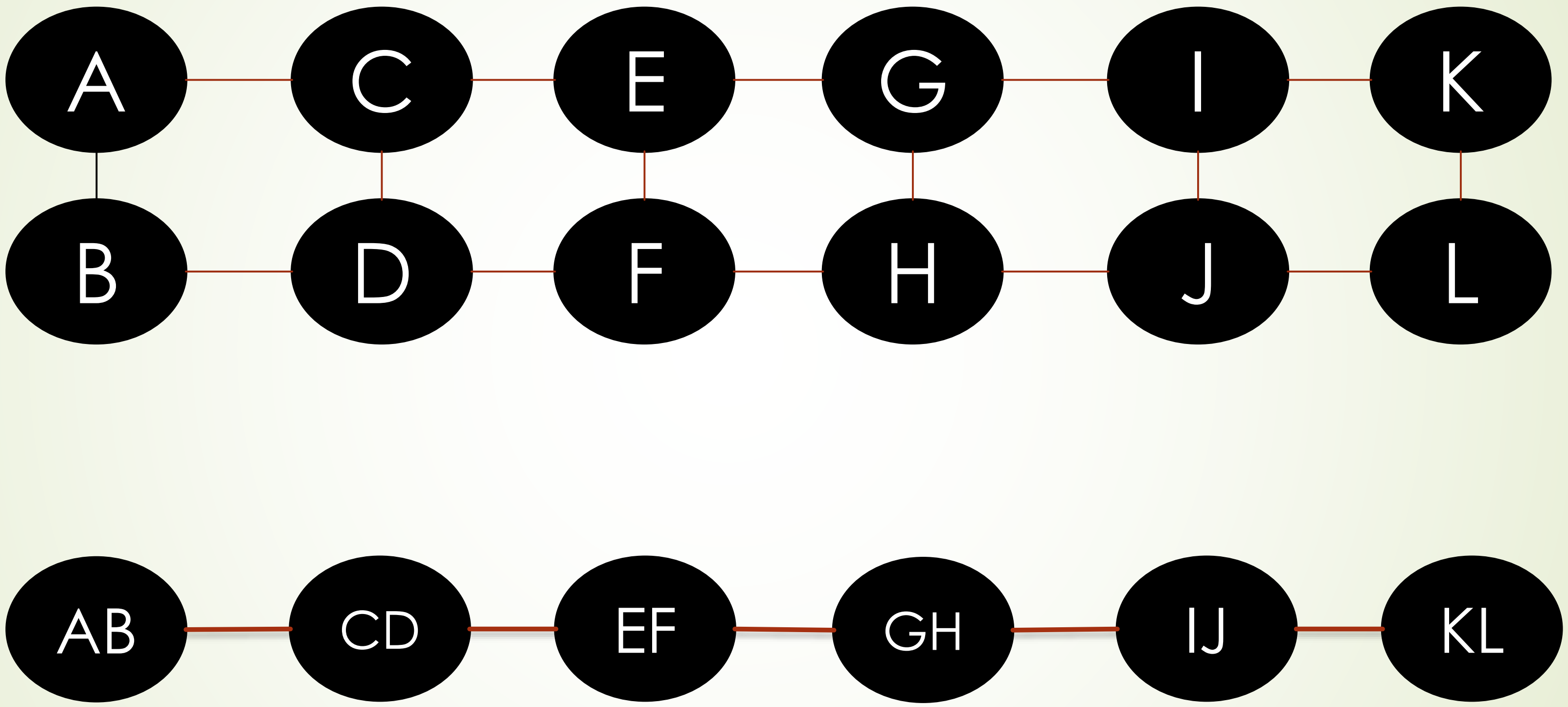
Outline

- Super Nodes
- Variable Elimination (VE)
- From VE to Junction Trees (Jtree)
- Calculating marginal probabilities in Jtrees

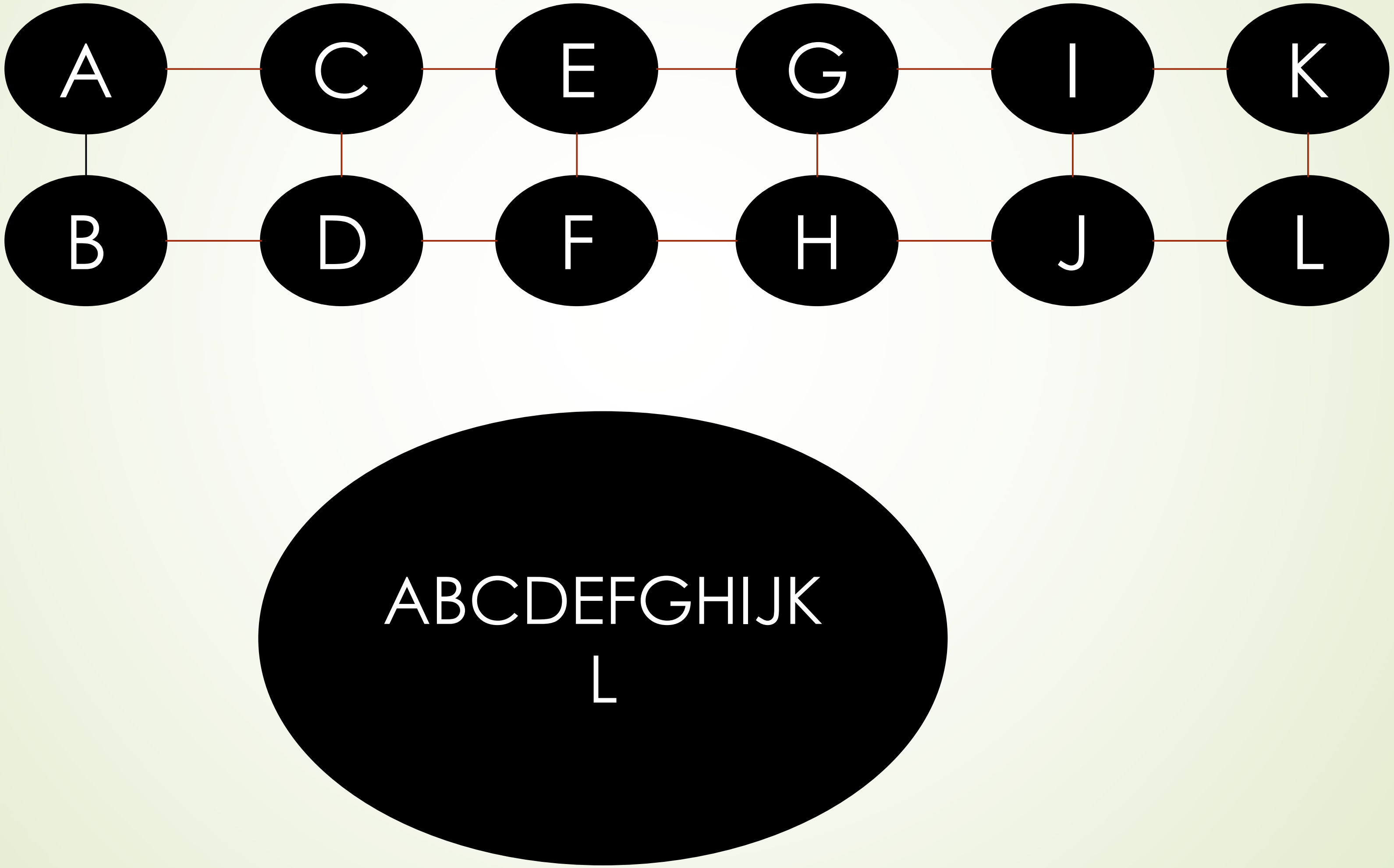
Super Nodes



Super Nodes



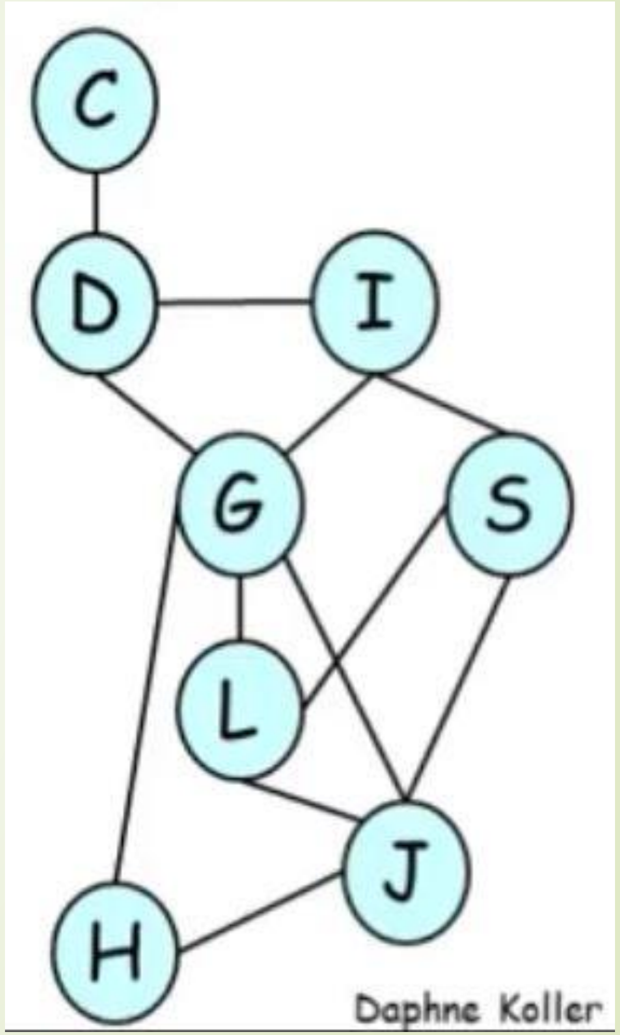
Super Nodes



Variable Elimination

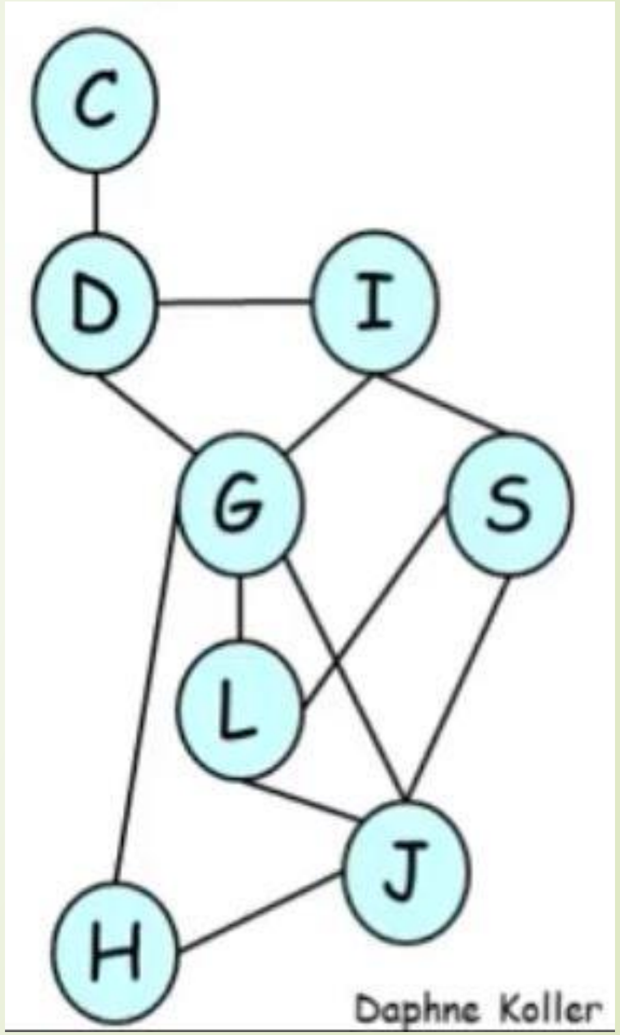
- We focus on calculating Z
- If we know how to calculate Z for a network, we can calculate all marginal probabilities.
- P(C=true | G=false)

$$= \frac{Z(Network|C=true,G=false)}{Z(Network|G=false)}$$



Variable Elimination

- $\phi_1(C)$
- $\phi_2(C,D)$
- $\phi_3(D,I,G)$
- $\phi_4(S,I)$
- $\phi_5(H,G,J)$
- $\phi_6(G,L)$
- $\phi_7(S,L,J)$

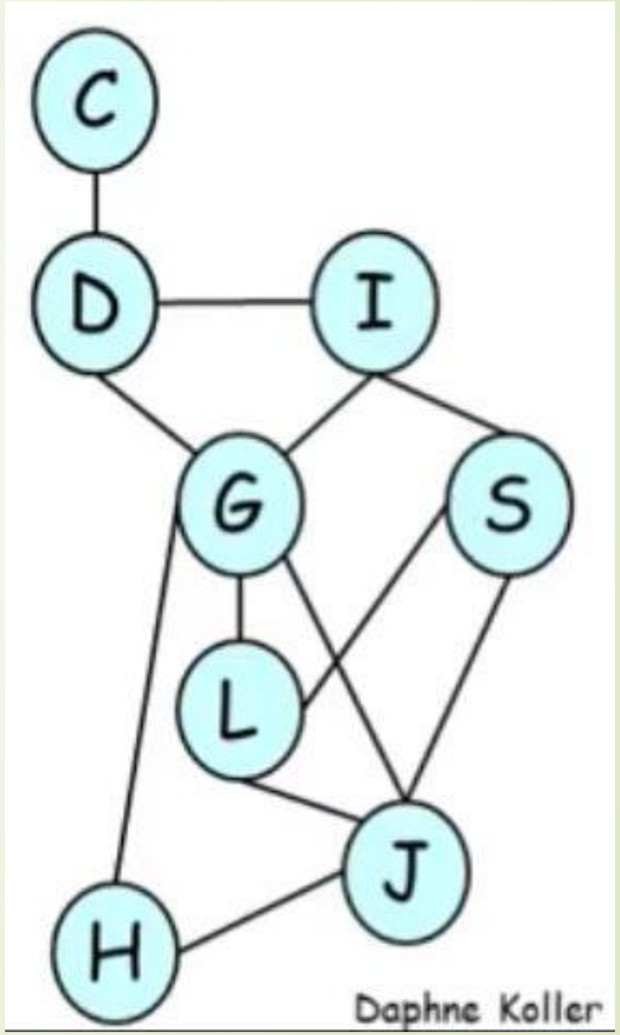


Variable Elimination(inference)

- $Z = \sum_J \sum_L \sum_S \sum_G \sum_H \sum_I \sum_D \sum_C$
  $\phi_1(C)\phi_2(C,D)\phi_3(D,I,G)\phi_4(S,I)$
  $\phi_5(H,G,J)\phi_6(G,L)\phi_7(S,L,J)$

- Elimination Order:
  $\psi = \langle$C, D, I, H, G, S, L, J$\rangle$

- $Z =$
  $\sum_J \sum_L \sum_S \phi_7(S,L,J) \sum_G \phi_6(G,L) \sum_H \phi_5(H,G,J)$
  $\sum_I \phi_4(S,I) \sum_D \phi_3(D,I,G) \sum_C \phi_1(C)\phi_2(C,D)$



Variable Elimination

- $Z = \sum_J \sum_L \sum_S \phi_7(S,L,J) \sum_G \phi_6(G,L) \sum_H \phi_5(H,G,J) \sum_I \phi_4(S,I) \sum_D \phi_3(D,I,G) \sum_C \phi_1(C)\phi_2(C,D)$
- $\phi_1(C)$ $\phi_2(C,D)$ $\lambda_1(C,D) = \phi_1(C)\phi_2(C,D)$ $\tau_1(D) = \sum_D \lambda(C,D)$

$\phi_1(C)$:

C	Value
T	2
F	1.2

$\phi_2(C,D)$:

C	D	Value
T	T	0.5
T	F	1
F	T	1
F	F	2

$\lambda_1(C,D)$:

C	D	Value
T	T	1
T	F	2
F	T	1.2
F	F	2.4

$\tau_1(D)$:

D	Value
T	2.2
F	4.4

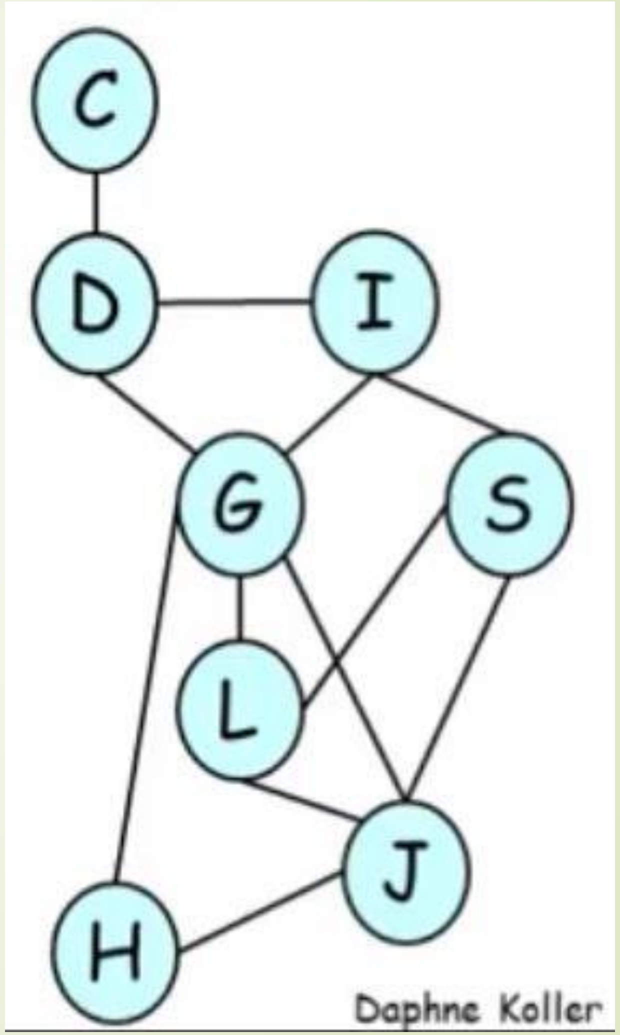
- $Z = \sum_J \sum_L \sum_S \phi_7(S,L,J) \sum_G \phi_6(G,L) \sum_H \phi_5(H,G,J) \sum_I \phi_4(S,I) \sum_D \phi_3(D,I,G)\tau_1(D)$

Variable Elimination(inference)

- $Z = \sum_J \sum_L \sum_S \phi_7(S,L,J) \sum_G \phi_6(G,L) \sum_H \phi_5(H,G,J)$
  $\sum_I \phi_4(S,I) \sum_D \phi_3(D,I,G)\tau_1(D)$
- $Z = \sum_J \sum_L \sum_S \phi_7(S,L,J) \sum_G \phi_6(G,L) \sum_H \phi_5(H,G,J)$
  $\sum_I \phi_4(S,I)\tau_2(G,I)$
- $Z = \sum_J \sum_L \sum_S \phi_7(S,L,J) \sum_G \phi_6(G,L)\tau_3(S,G) \sum_H \phi_5(H,G,J)$
- $Z = \sum_J \sum_L \sum_S \phi_7(S,L,J) \sum_G \phi_6(G,L)\tau_3(S,G)\tau_4(G,J)$
- …
- $Z = \sum_J \tau_{n-1}(J)$

Variable Elimination(decoding)

- $Argmax_J Argmax_L \dots Argmax_C$
  $\phi_1(C)\phi_2(C,D)\phi_3(D,I,G)\phi_4(S,I)$
  $\phi_5(H,G,J)\phi_6(G,L)\phi_7(S,L,J)$
- Elimination Order:
  $\psi = \text{<C, D, I, H, G, S, L, J>}$
- $Argmax_J \dots Argmax_C \phi_1(C)\phi_2(C,D)$



Variable Elimination(decoding)

- $Argmax_J \ldots Argmax_C \phi_1(C)\phi_2(C,D)$
- $\phi_1(C)$ $\phi_2(C,D)$ $\lambda_1(C,D) = \phi_1(C)\phi_2(C,D)$ $\tau_1(D) = \max_C \lambda(C,D)$

C	Value
T	2
F	1.2

C	D	Value
T	T	0.5
T	F	1
F	T	1
F	F	2

C	D	Value
T	T	1
T	F	2
F	T	1.2
F	F	2.4

D	Value
T	1.2
F	2.4

D	Val(C)
T	F
F	F

- $Argmax_J \ldots \tau_1(D)$

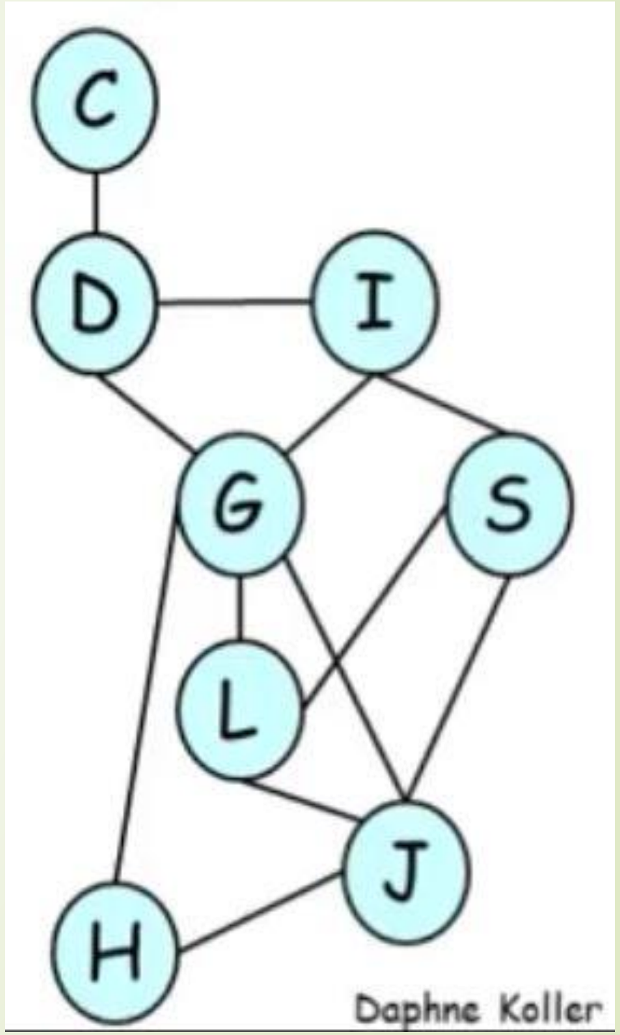
Time Complexity(Assuming binary variables)

- Let $\phi_1, \phi_2, \dots, \phi_m$ be potentials containing a variable V.
- Let $\tau(C_1, C_2, \dots, C_w) = \sum_V \phi_1 \phi_2 \dots \phi_m$
- $\tau(C_1, C_2, \dots, C_w)$ can be calculated in $O(2^w)$
- Let $w_1, w_2, \dots w_n$ correspond to the number of variables in $\tau_1, \tau_2, \dots \tau_n$ given a specific elimination order ψ.
- Let $\omega = \max(w_1, w_2, \dots, w_n)$
- Variable elimination with elimination order ψ is $O(n2^\omega)$
- ω is called the width of ψ.

ω depends on the elimination order

- $Z = \sum_C \sum_D \sum_I \sum_G \sum_S \sum_L \sum_J \sum_H \phi_1(C)\phi_2(C,D)\phi_3(D,I,G)\phi_4(S,I)\phi_5(H,G,J)\phi_6(G,L)\phi_7(S,L,J)$
- Elimination Order:

  $\psi = \langle G, \ldots \rangle$
- $Z = \sum \ldots \sum_G \phi_3(D,I,G)\phi_5(H,G,J)\phi_6(G,L)$
- $Z = \sum \ldots \tau(D,I,H,J,L)$



Time Complexity

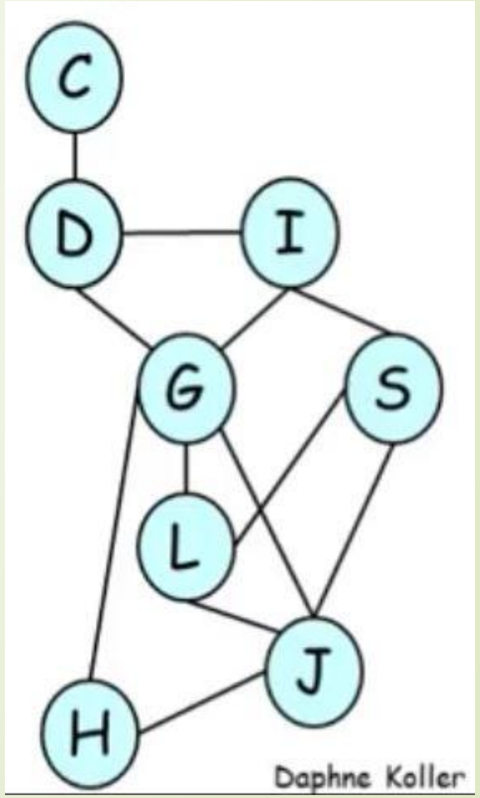
- Let $\{\psi_1, \psi_2, \dots, \psi_t\}$ represent all possible elimination orders, and $\{\omega(\psi_1), \omega(\psi_2), \dots, \omega(\psi_t)\}$ represent the widths of these elimination orders.
- Define $\text{treewidth} = \min_{\psi \in \{\psi_1, \psi_2, \dots, \psi_t\}} \omega(\psi)$
- Variable elimination is then $O(n2^{treewidth})$
- Finding a ψ with $\omega(\psi)$=treewidth is NP-Hard.

From VE to Junction Trees (Jtrees)

- Variable Elimination is query sensitive: we must specify the query variable in advance. Each time we run a new query, we must re-run the entire algorithm.
- The junction tree algorithms generalizes VE to avoid this; they compile the UGM into a data structure which supports simultaneous execution of queries.

From VE to Junction Trees (Jtrees)

- $Z = \sum \ldots \sum_D \phi_3(D, I, G) \sum_C \phi_1(C) \phi_2(C, D)$



From VE to Junction Trees (Jtrees)

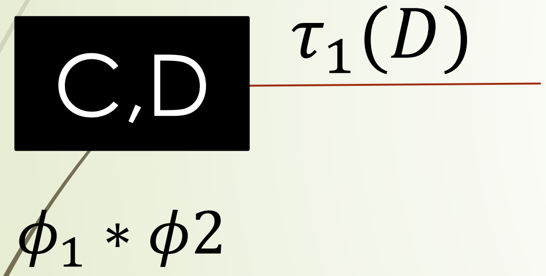
- $Z = \sum \ldots \sum_D \phi_3(D, I, G) \sum_C \phi_1(C)\phi_2(C, D)$

C,D

$\phi_1 * \phi 2$

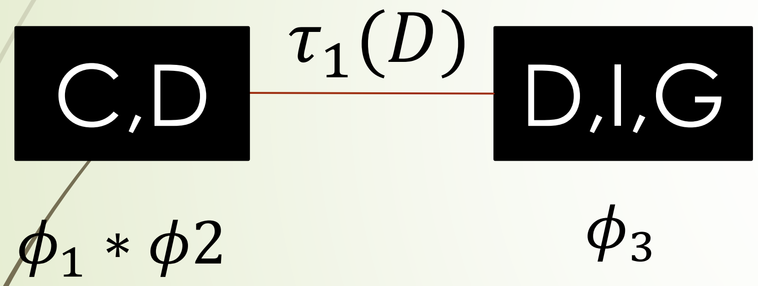
From VE to Junction Trees (Jtrees)

- $Z = \sum \ldots \sum_D \phi_3(D, I, G)\, \tau_1(D)$



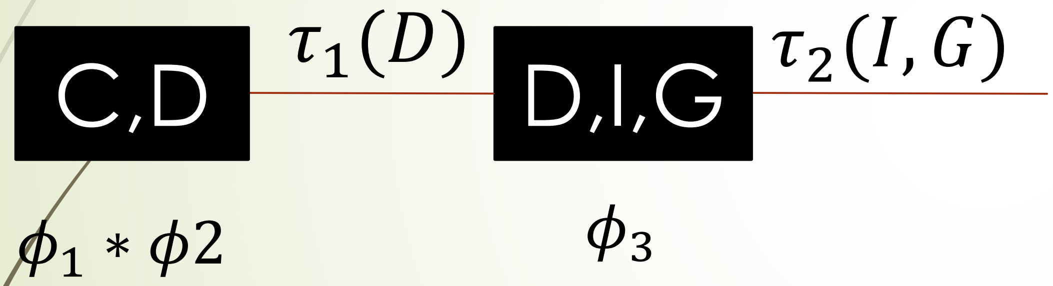
From VE to Junction Trees (Jtrees)

- $Z = \sum \ldots \sum_D \phi_3(D, I, G)\, \tau_1(D)$



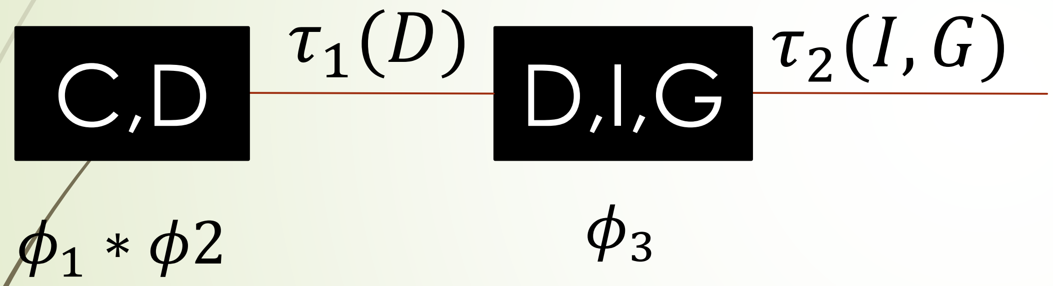
From VE to Junction Trees (Jtrees)

- $Z = \sum \ldots \tau_2(I, G)$



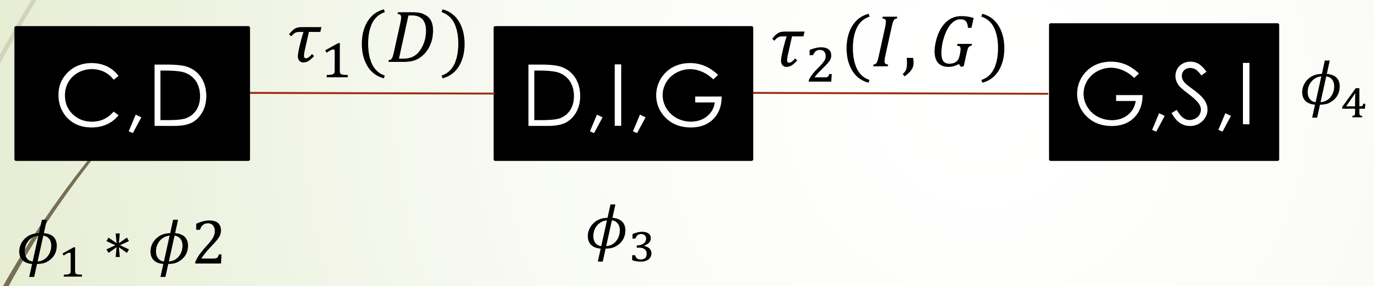
From VE to Junction Trees (Jtrees)

- $Z = \sum_J \sum_L \sum_S \phi_7(S,L,J) \sum_G \phi_6(G,L) \sum_H \phi_5(H,G,J) \sum_I \phi_4(S,I) \tau_2(G,I)$



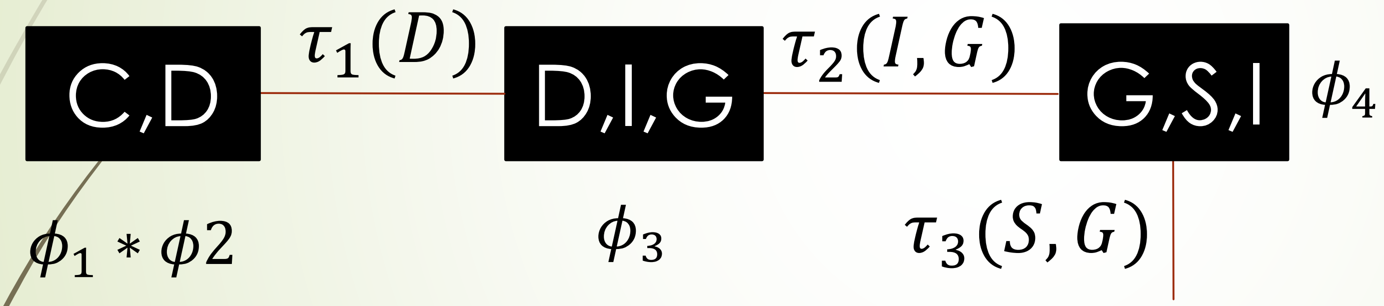
From VE to Junction Trees (Jtrees)

- $Z = \sum_J \sum_L \sum_S \phi_7(S,L,J) \sum_G \phi_6(G,L) \sum_H \phi_5(H,G,J) \sum_I \phi_4(S,I)\tau_2(G,I)$



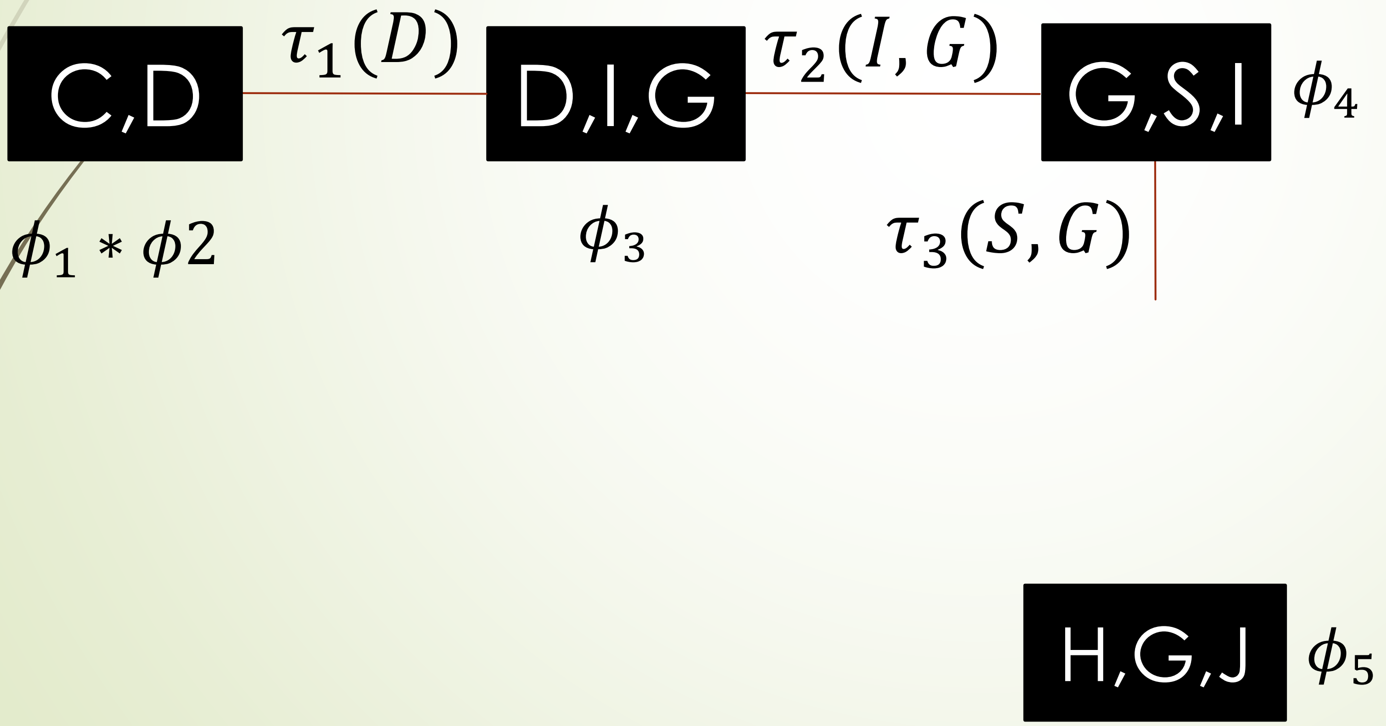
From VE to Junction Trees (Jtrees)

- $Z = \sum_J \sum_L \sum_S \phi_7(S,L,J) \sum_G \phi_6(G,L)\tau_3(S,G) \sum_H \phi_5(H,G,J)$



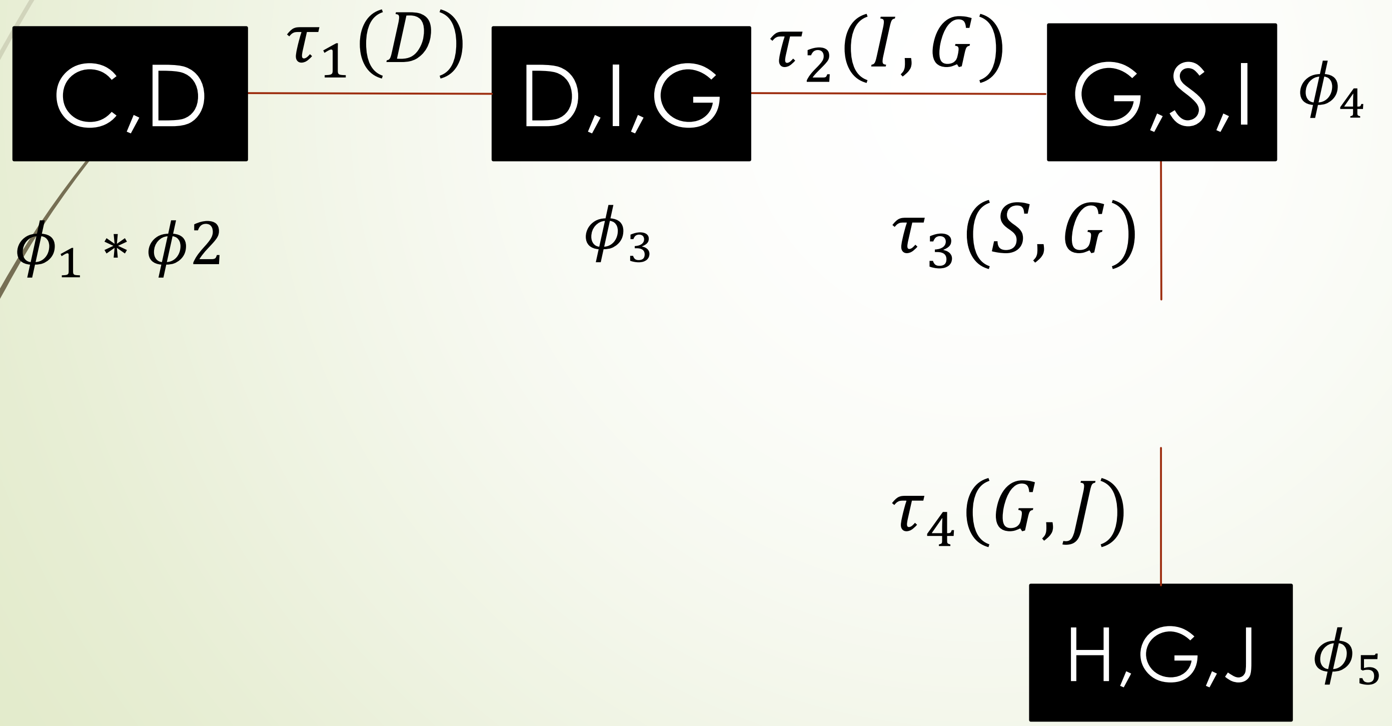
From VE to Junction Trees (Jtrees)

- $Z = \sum_J \sum_L \sum_S \phi_7(S, L, J) \sum_G \phi_6(G, L) \tau_3(S, G) \sum_H \phi_5(H, G, J)$



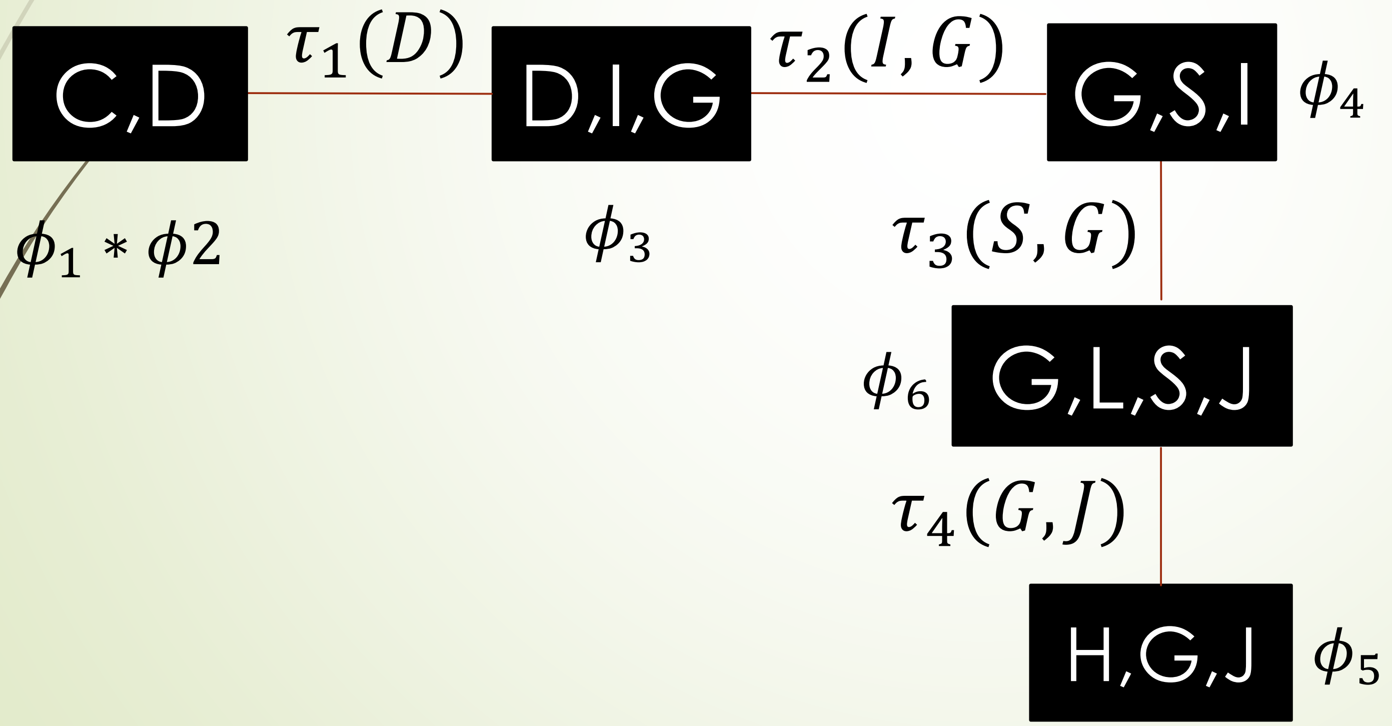
From VE to Junction Trees (Jtrees)

- $Z = \sum_J \sum_L \sum_S \phi_7(S,L,J) \sum_G \phi_6(G,L)\tau_3(S,G)\tau_4(G,J)$



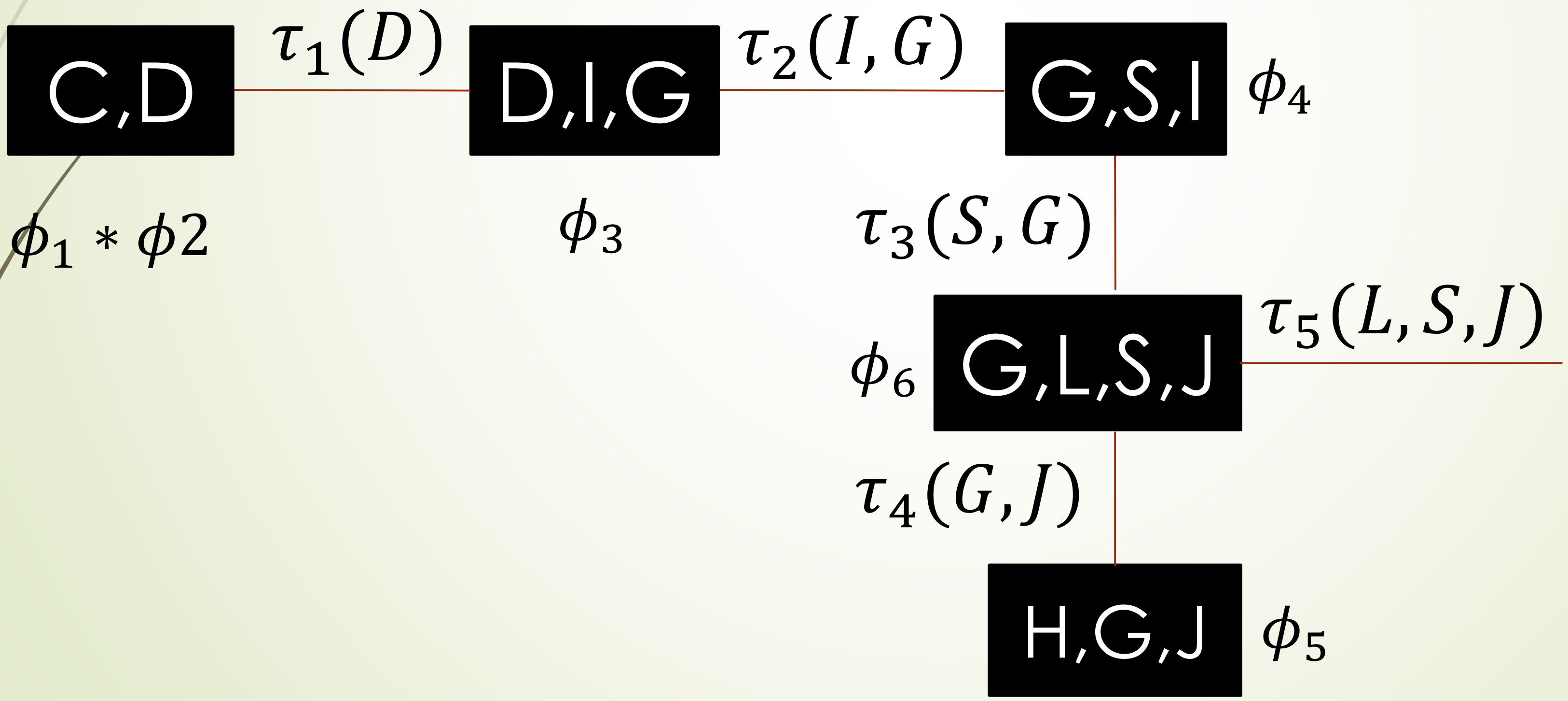
From VE to Junction Trees (Jtrees)

- $Z = \sum_J \sum_L \sum_S \phi_7(S,L,J) \sum_G \phi_6(G,L)\tau_3(S,G)\tau_4(G,J)$



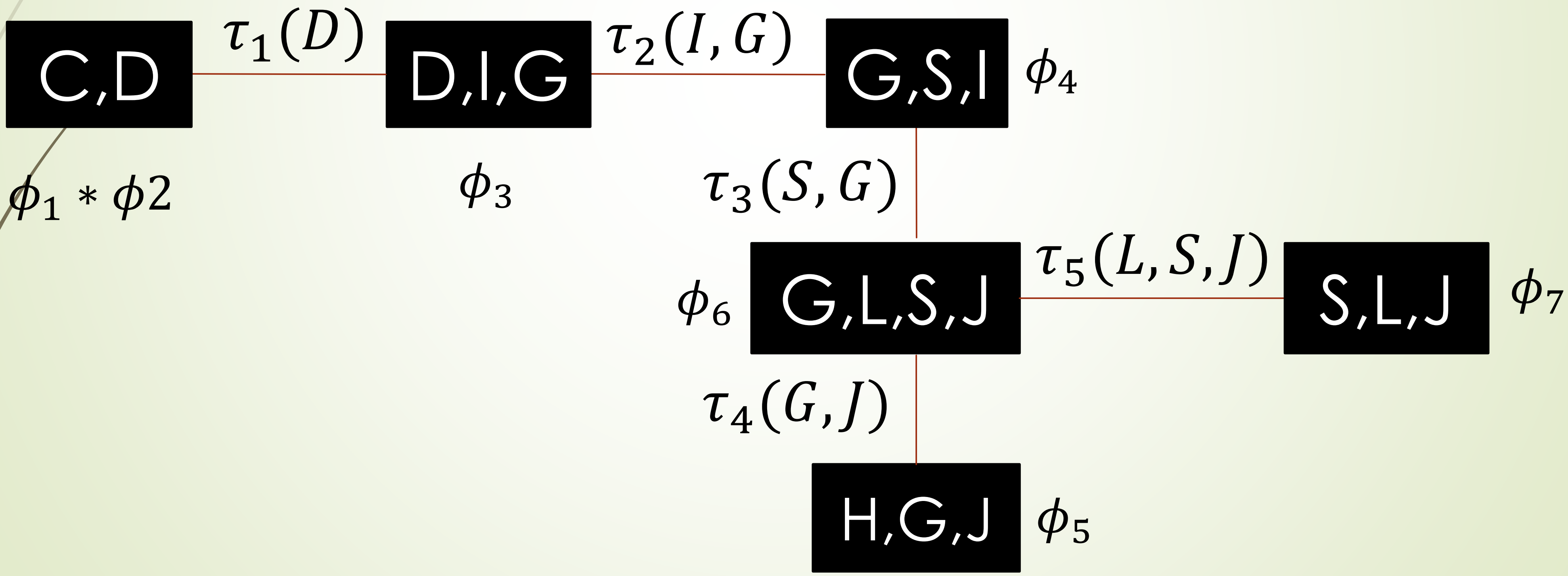
From VE to Junction Trees (Jtrees)

- $Z = \sum_J \sum_L \sum_S \phi_7(S,L,J)\, \tau_5(L,S,J)$



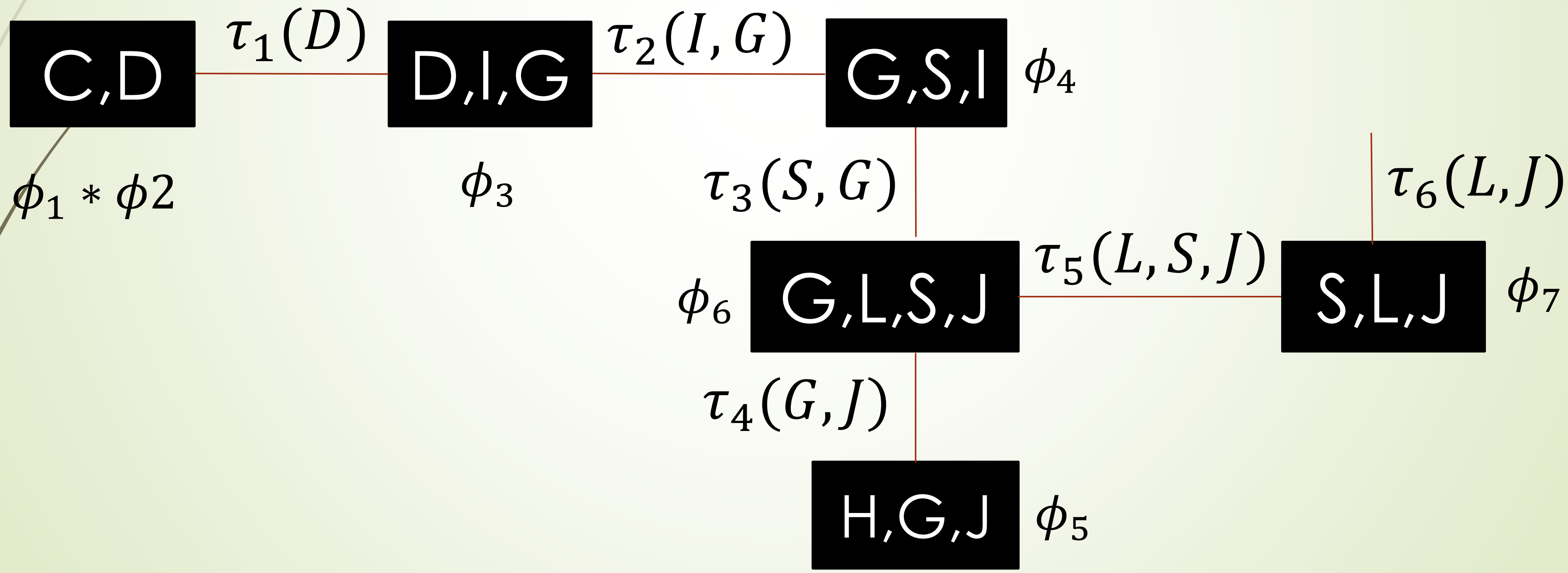
From VE to Junction Trees (Jtrees)

- $Z = \sum_J \sum_L \sum_S \phi_7(S, L, J)\, \tau_5(L, S, J)$



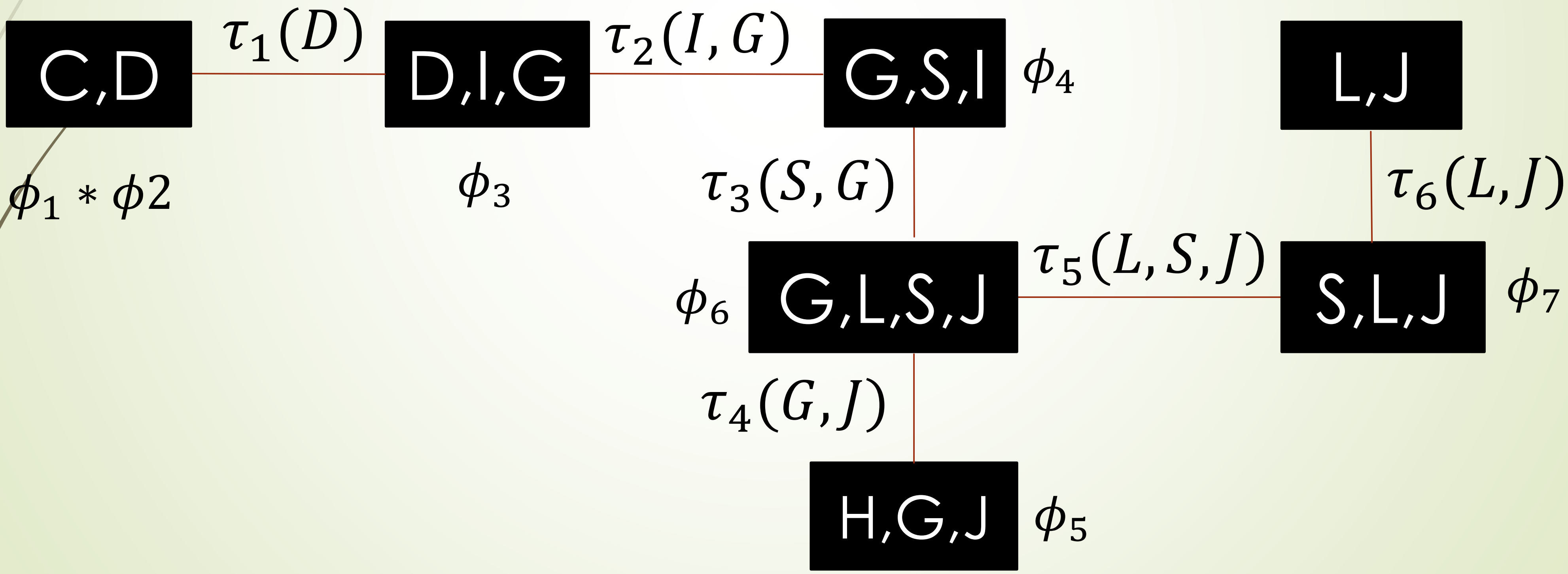
From VE to Junction Trees (Jtrees)

- $Z = \sum_J \sum_L \tau_6(L, J)$



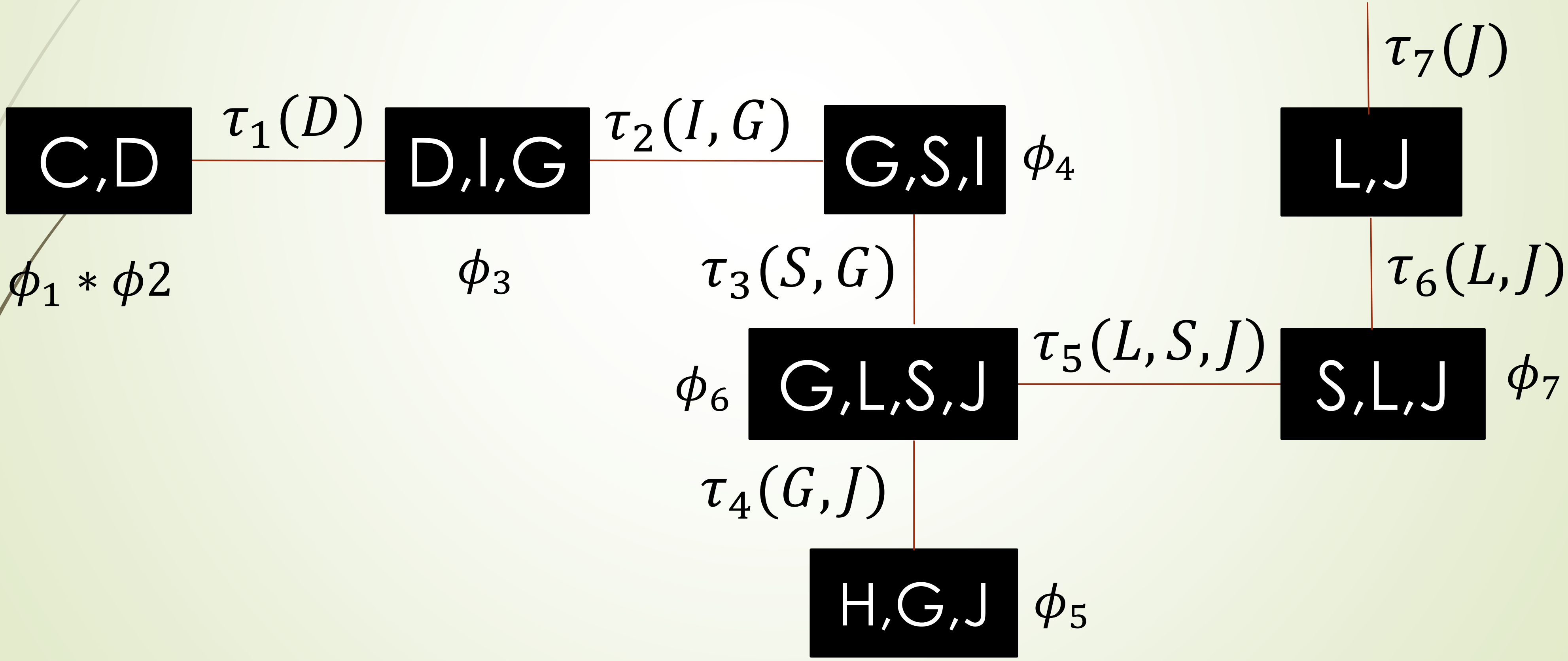
From VE to Junction Trees (Jtrees)

- $Z = \sum_J \sum_L \tau_6(L, J)$



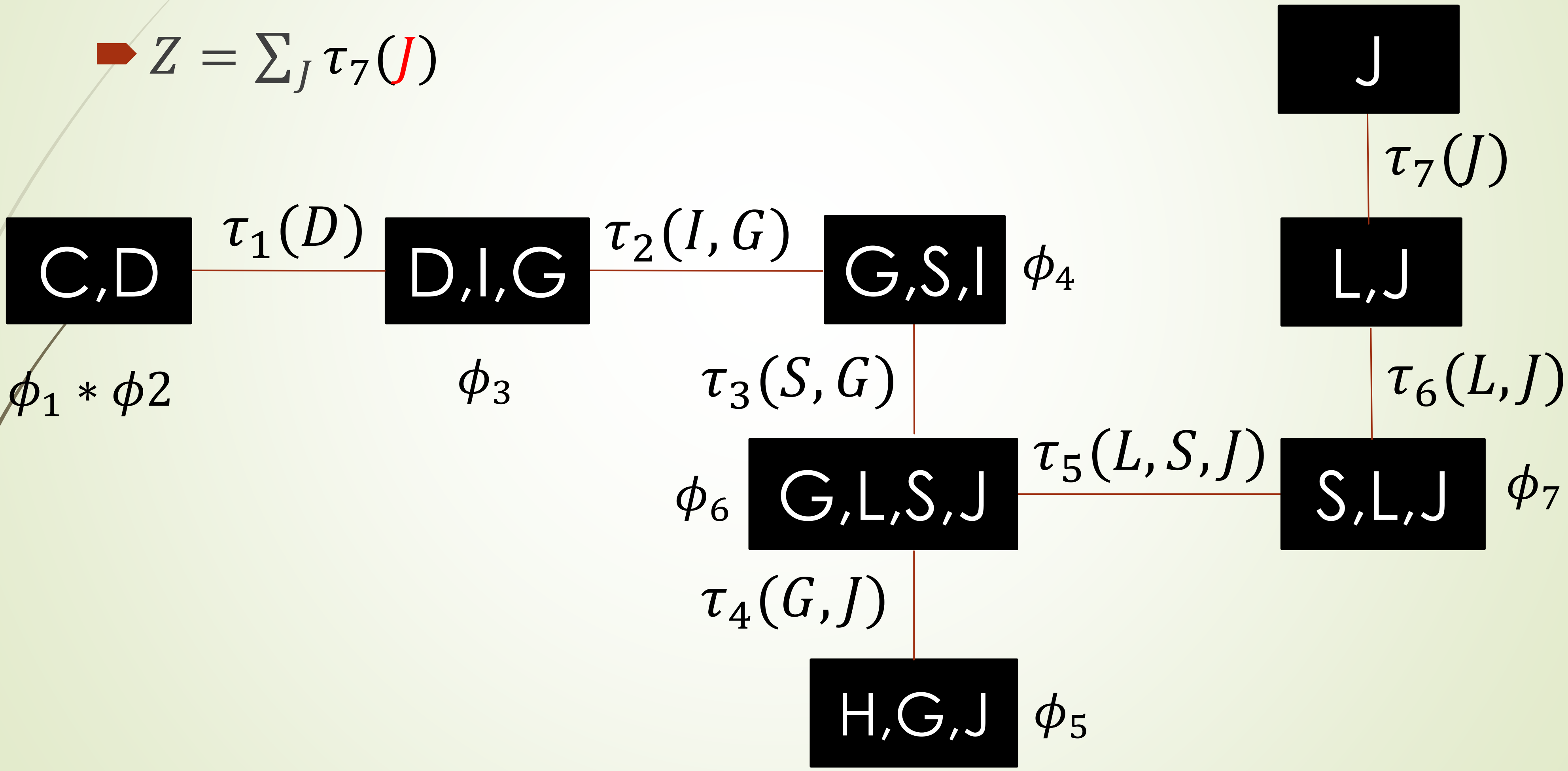
From VE to Junction Trees (Jtrees)

- $Z = \sum_J \tau_7(J)$



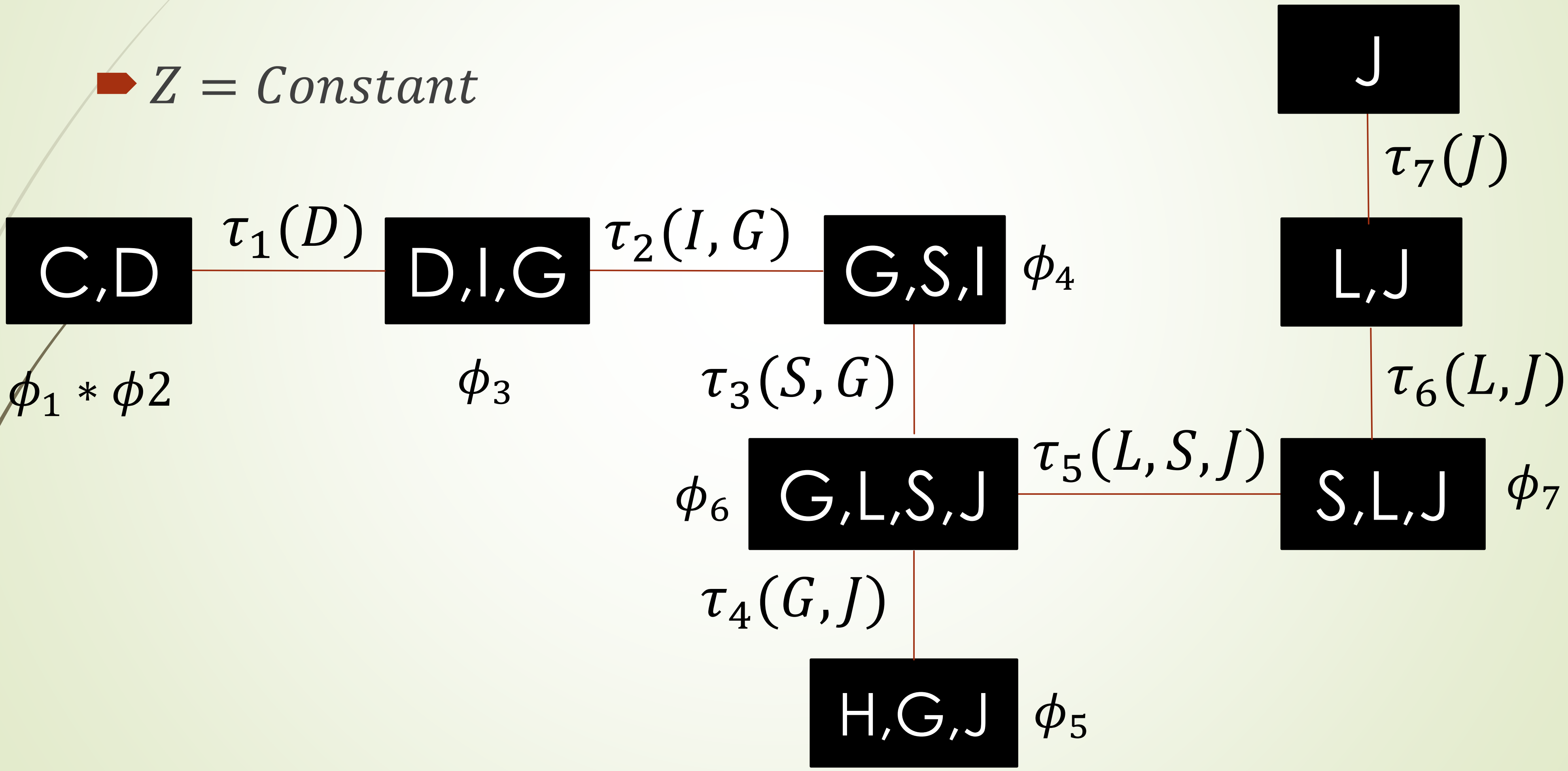
From VE to Junction Trees (Jtrees)

- $Z = \sum_J \tau_7(J)$

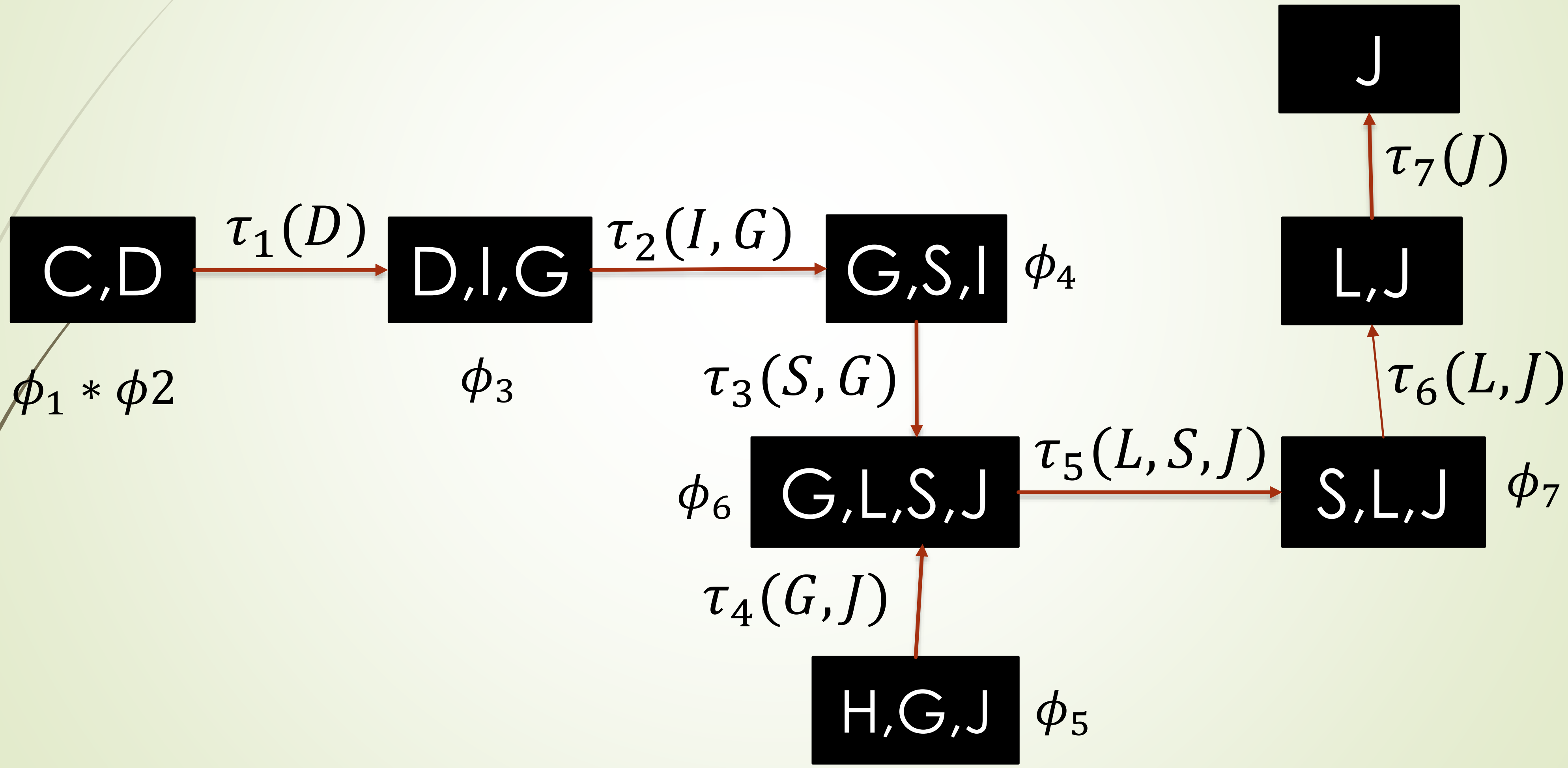


From VE to Junction Trees (Jtrees)

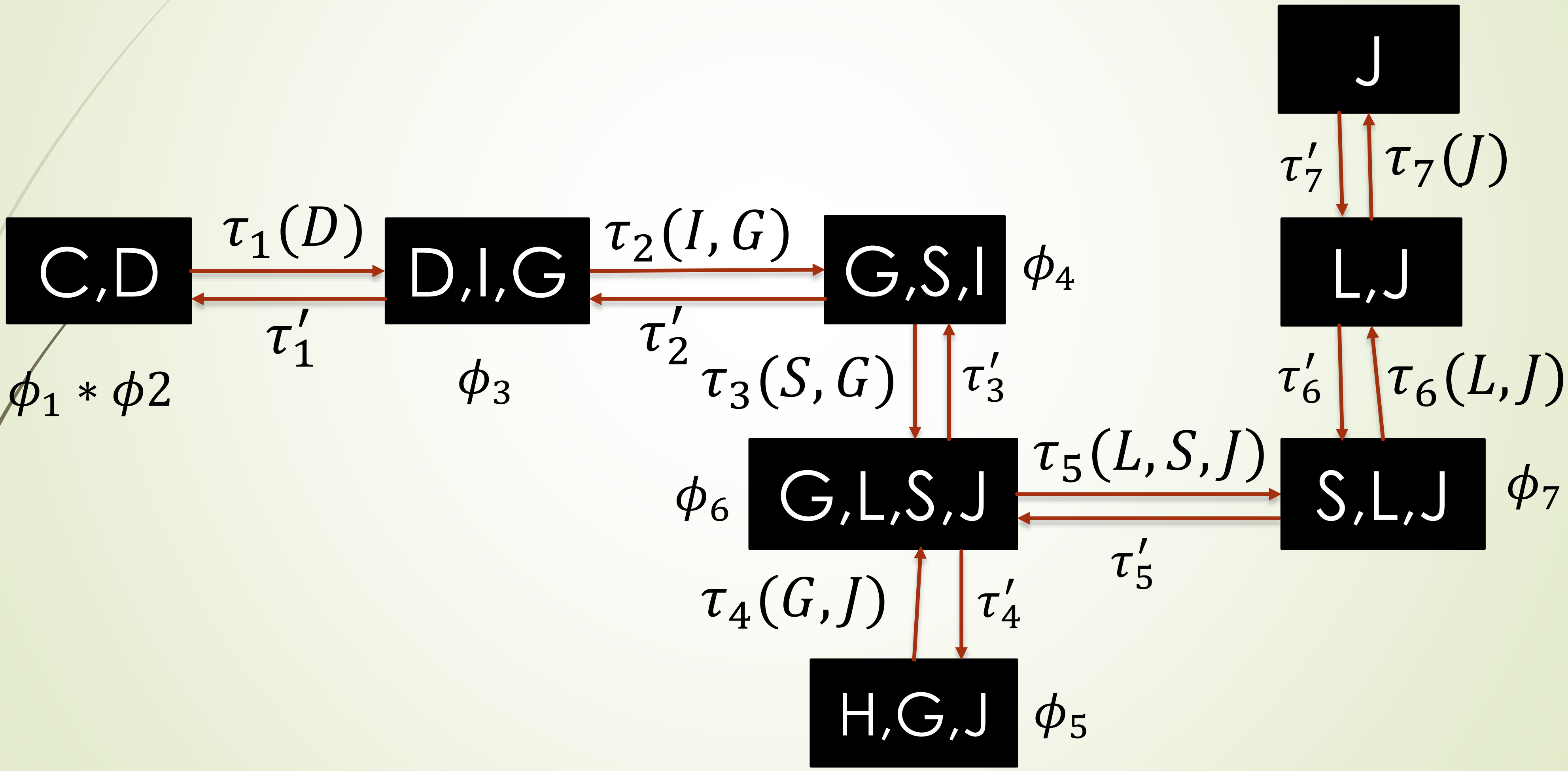
- $Z = Constant$



Inference in Jtrees

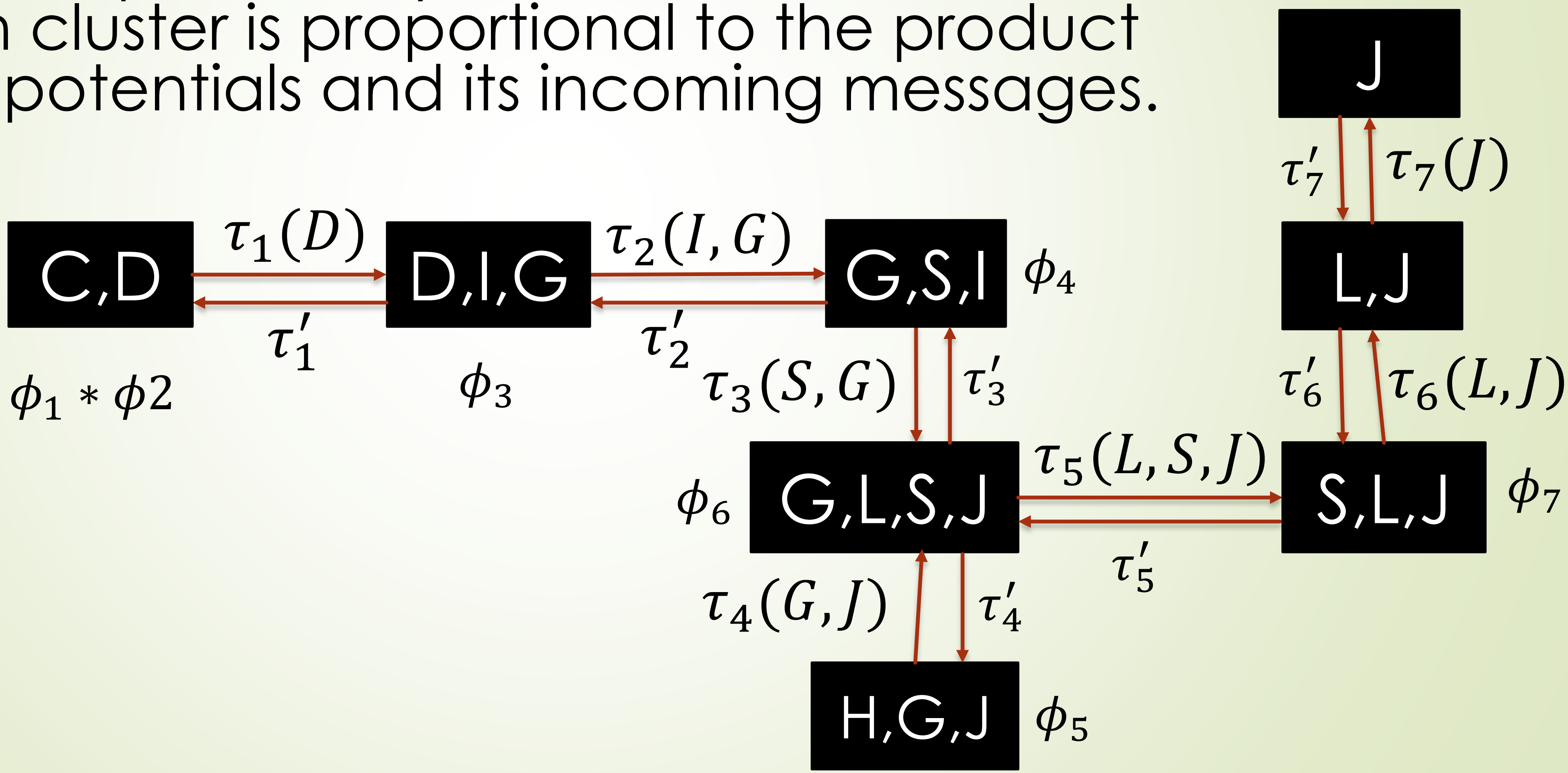


Inference in Jtrees



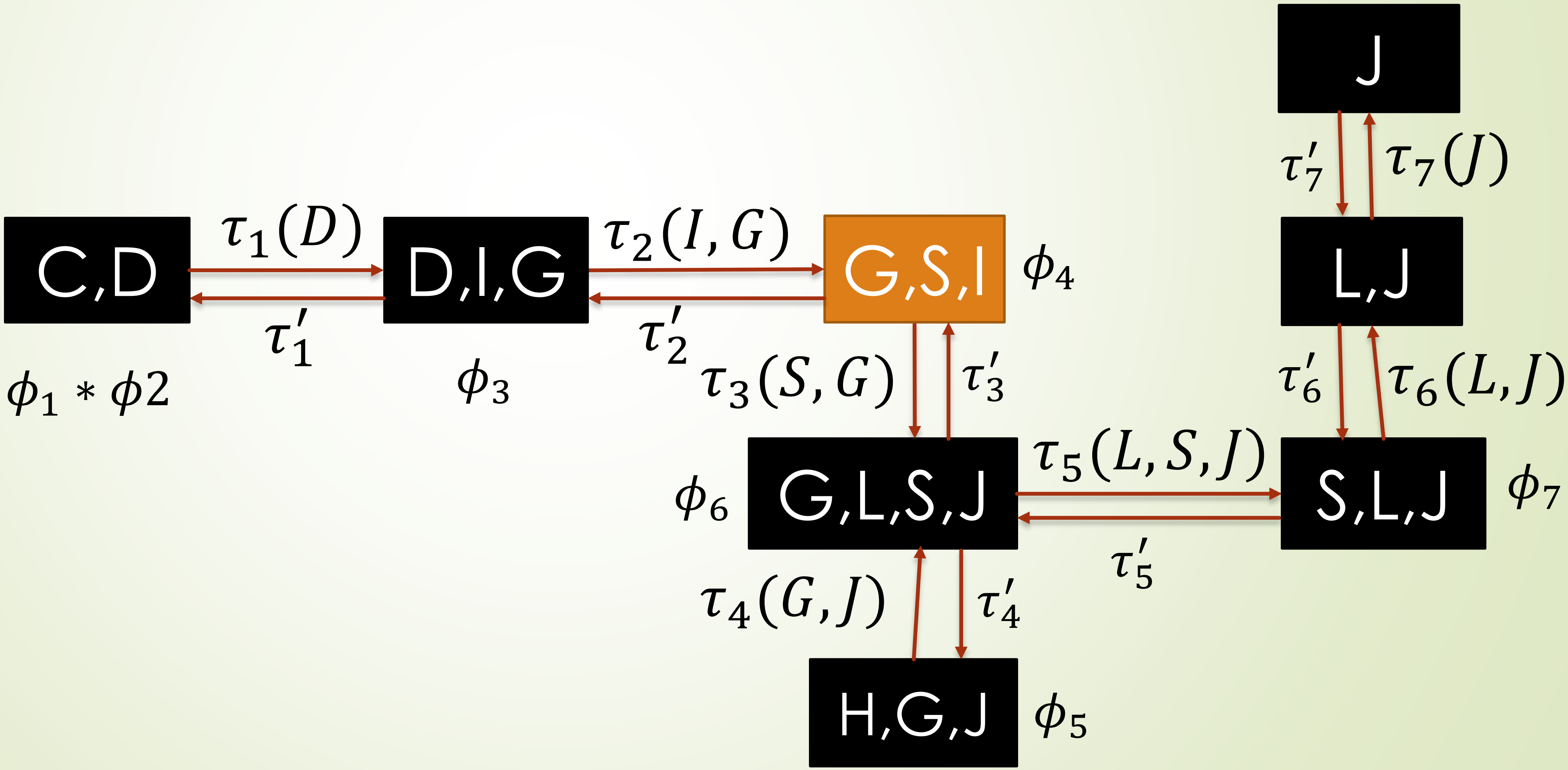
Inference in Jtrees

- The joint probability of the variables in each cluster is proportional to the product of its potentials and its incoming messages.



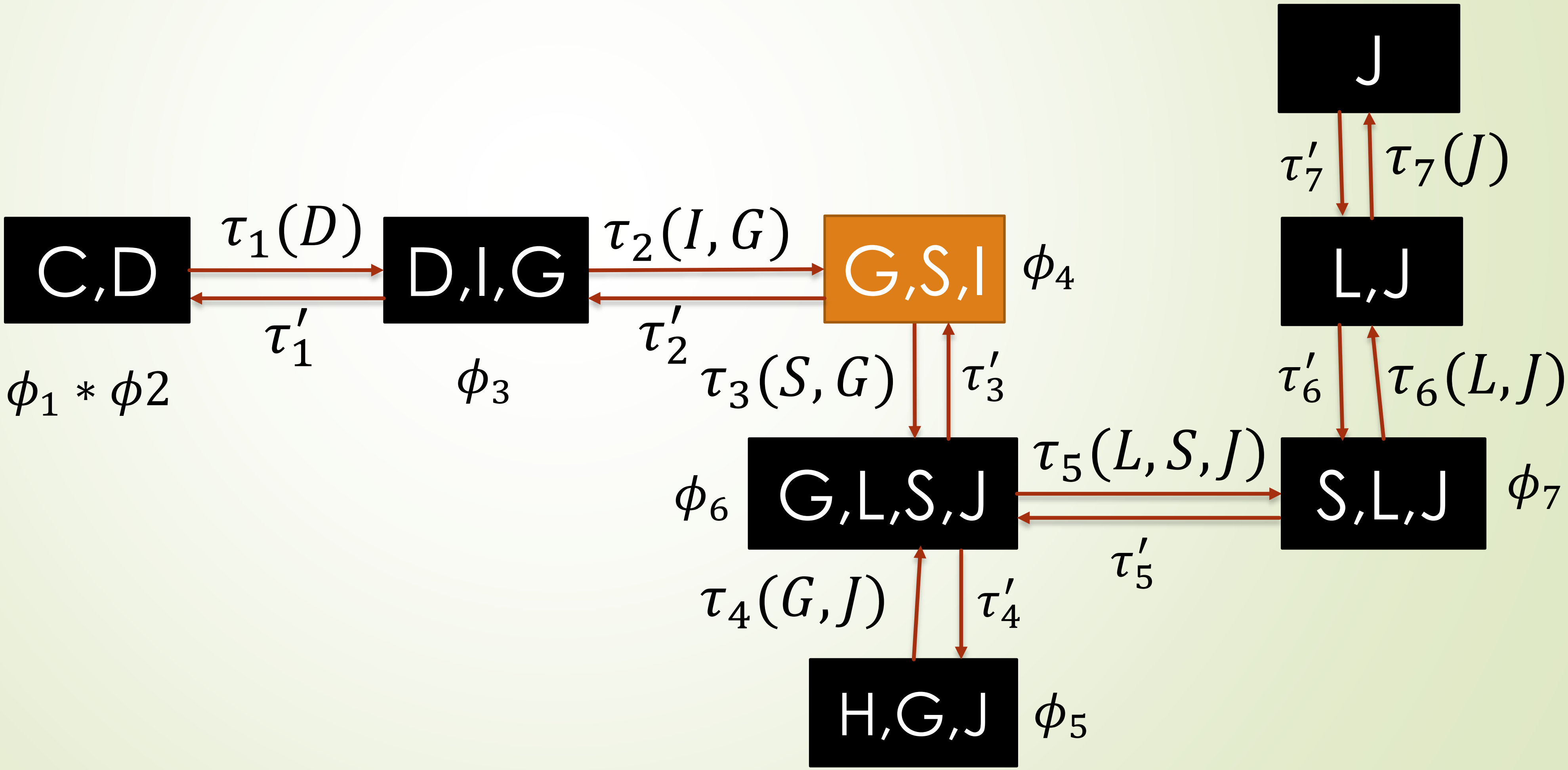
Inference in Jtrees

- $P(G,S,I) \propto \ \phi_4 \tau_2(I,G)\tau_3'(S,G)$



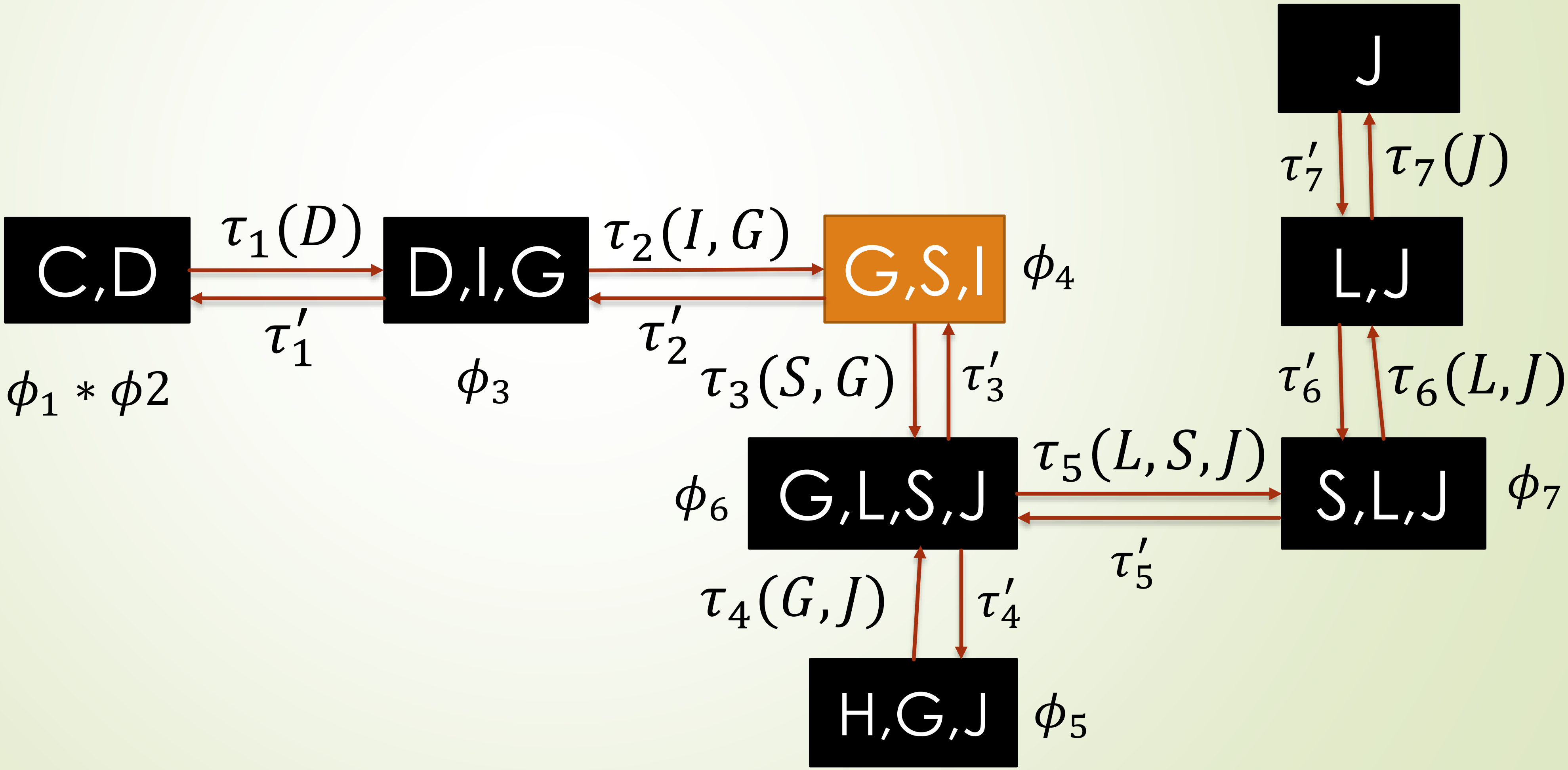
Inference in Jtrees

- $P(G,S,I) \propto \phi_4(\sum_D \phi_3\, \tau_1(D))\tau'_3(S,G)$



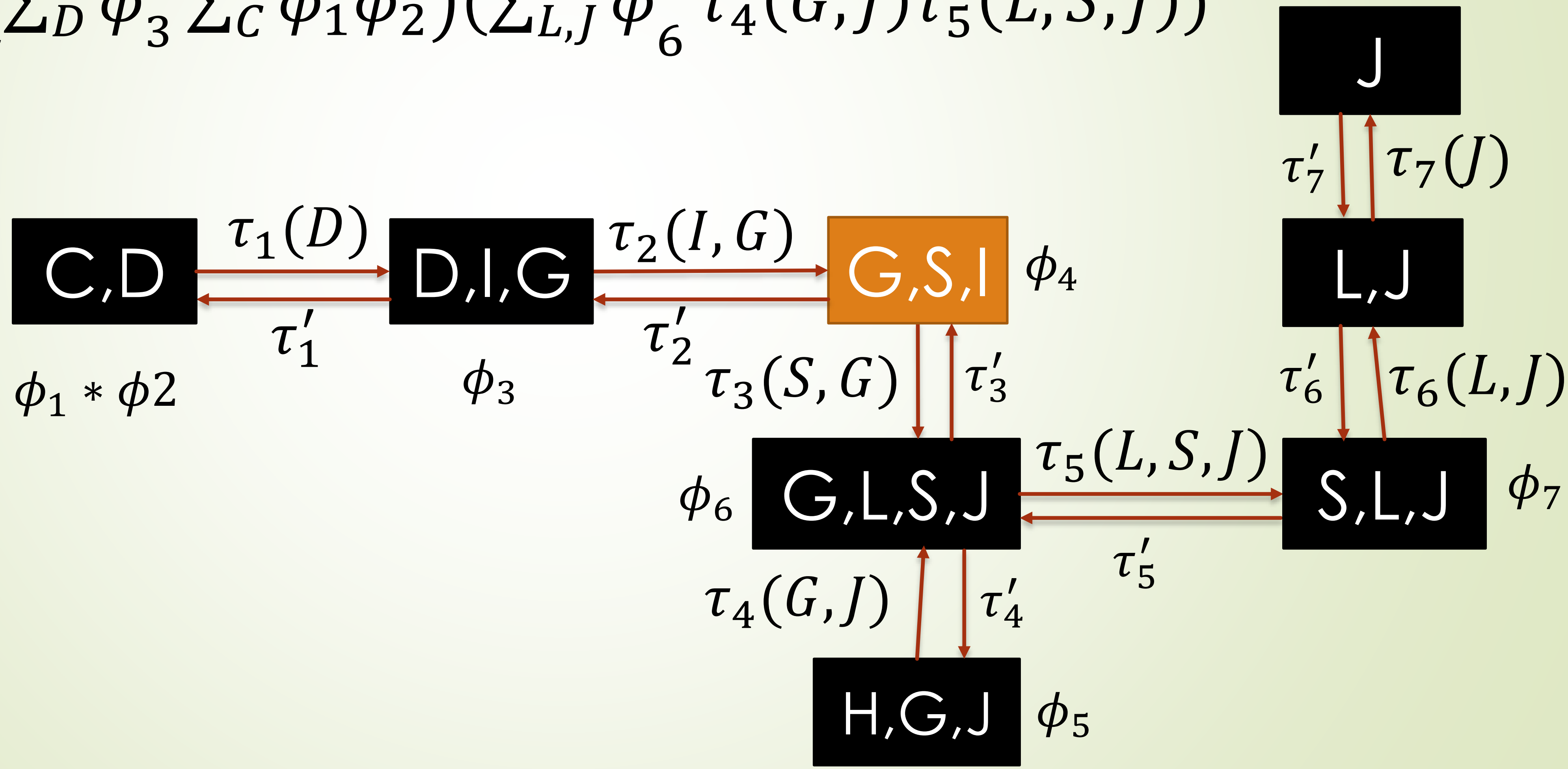
Inference in Jtrees

- $P(G,S,I) \propto \phi_4(\sum_D \phi_3 \sum_C \phi_1\phi_2)\tau'_3(S,G)$



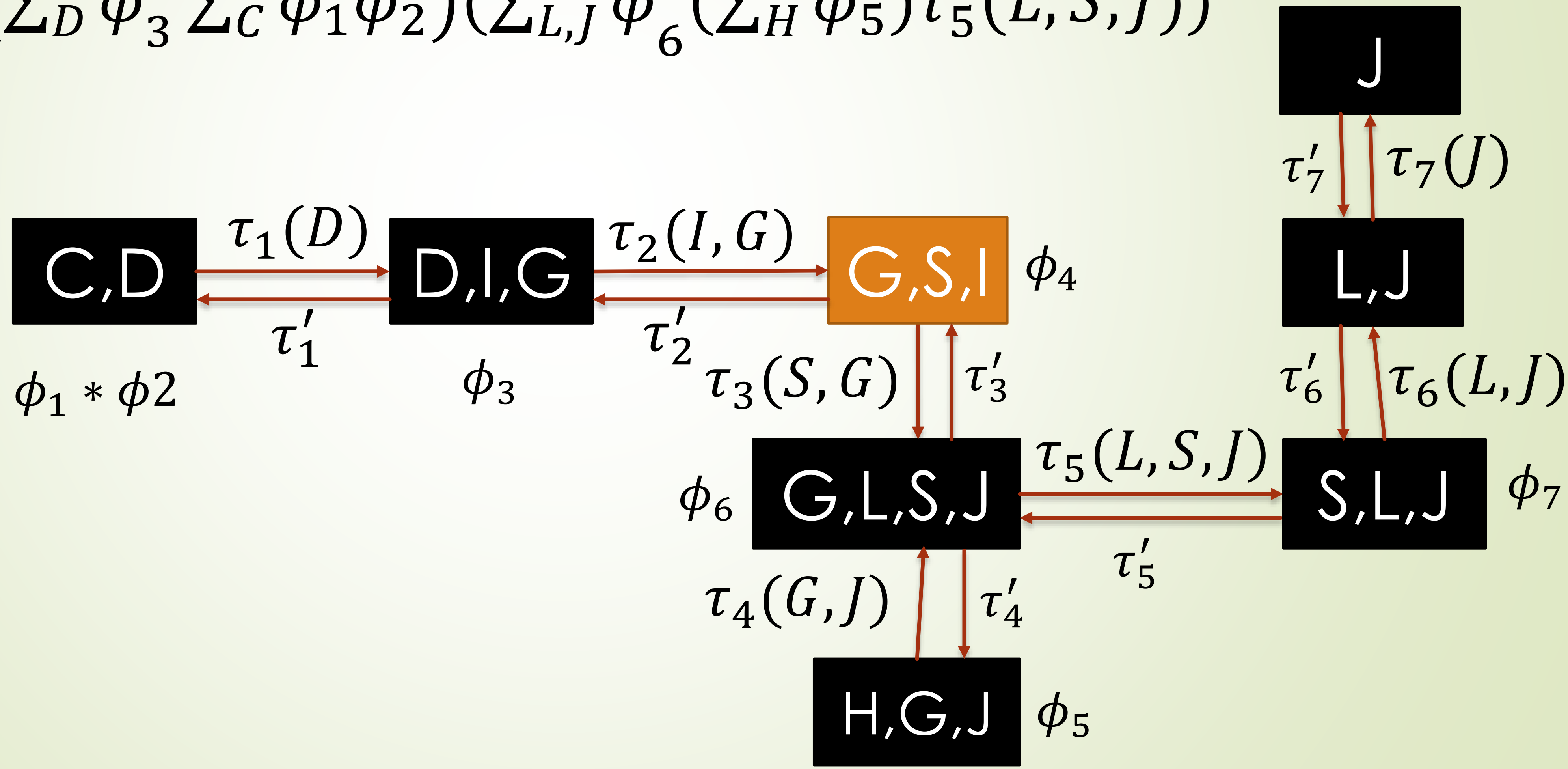
Inference in Jtrees

- $P(G,S,I) \propto \phi_4\left(\sum_D \phi_3 \sum_C \phi_1\phi_2\right)\left(\sum_{L,J} \phi_6\, \tau_4(G,J)\tau_5'(L,S,J)\right)$



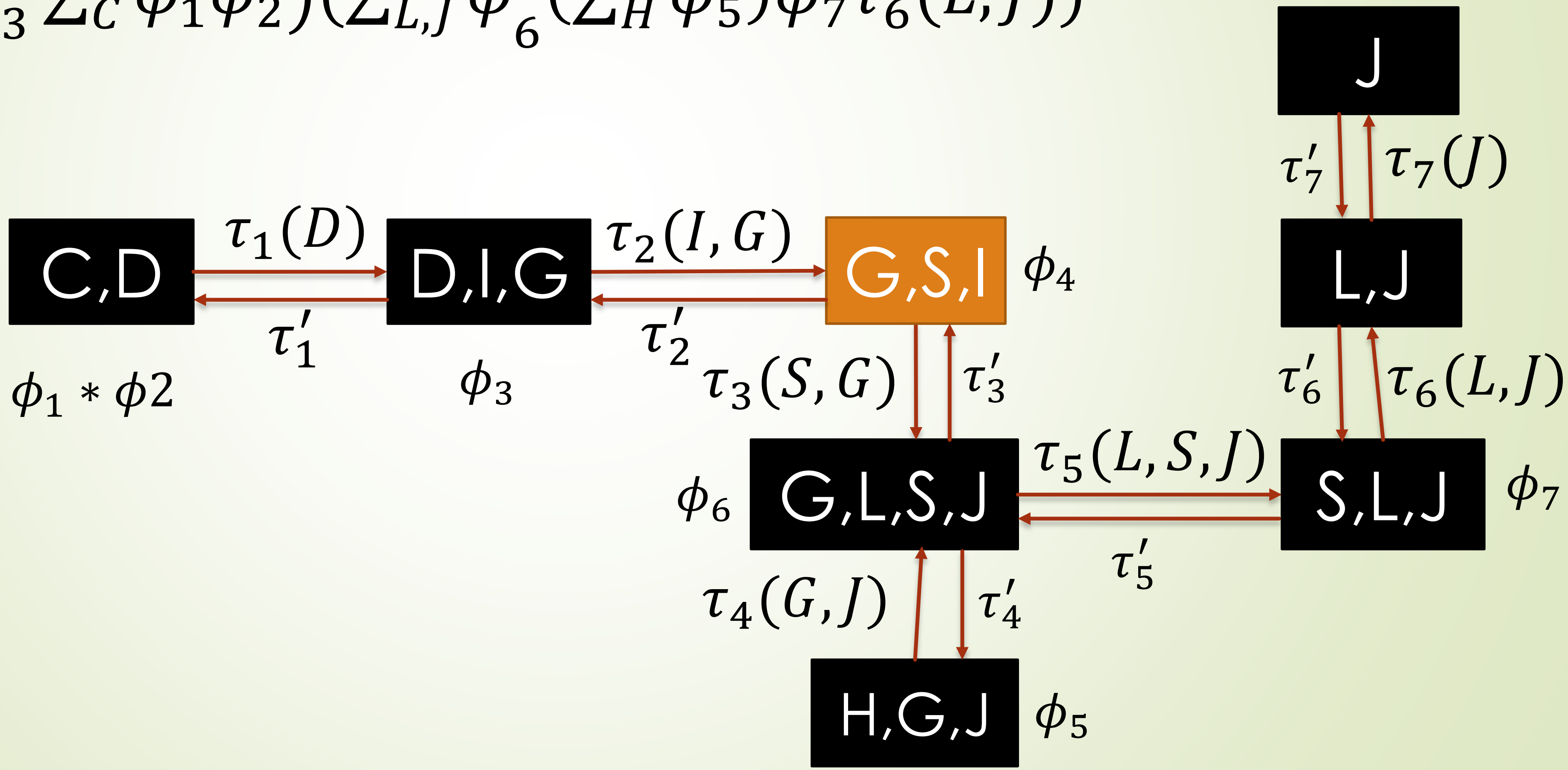
Inference in Jtrees

- $P(G,S,I) \propto \phi_4\left(\sum_D \phi_3 \sum_C \phi_1\phi_2\right)\left(\sum_{L,J} \phi_6\left(\sum_H \phi_5\right)\tau_5'(L,S,J)\right)$



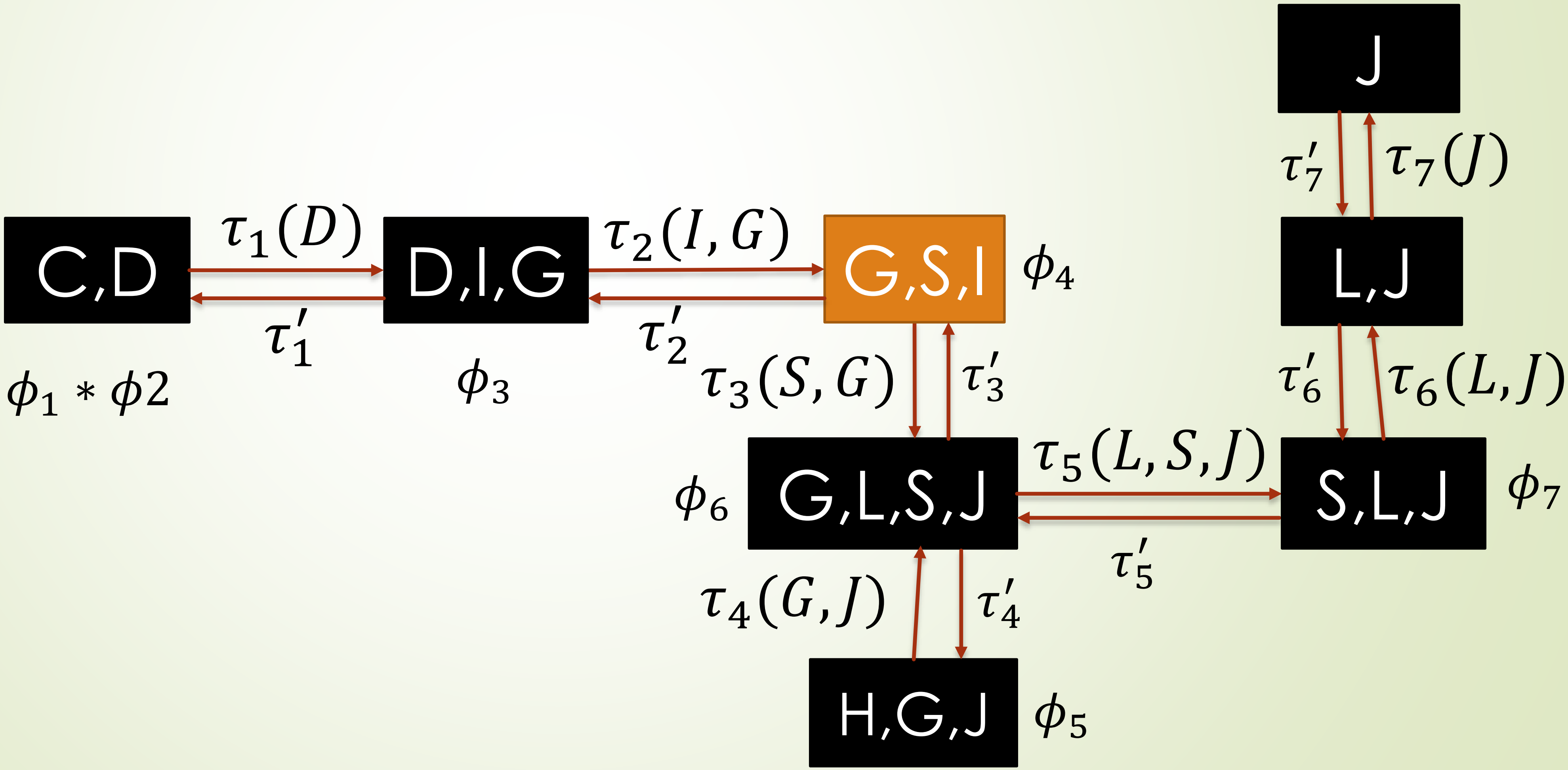
Inference in Jtrees

- $P(G,S,I) \propto$
$\phi_4\left(\sum_D \phi_3 \sum_C \phi_1\phi_2\right)\left(\sum_{L,J} \phi_6\left(\sum_H \phi_5\right)\phi_7\tau'_6(L,J)\right)$



Inference in Jtrees

- $P(G,S,I) \propto \phi_4\left(\sum_D \phi_3 \sum_C \phi_1\phi_2\right)\left(\sum_{L,J} \phi_6\left(\sum_H \phi_5\right)\phi_7\right)$

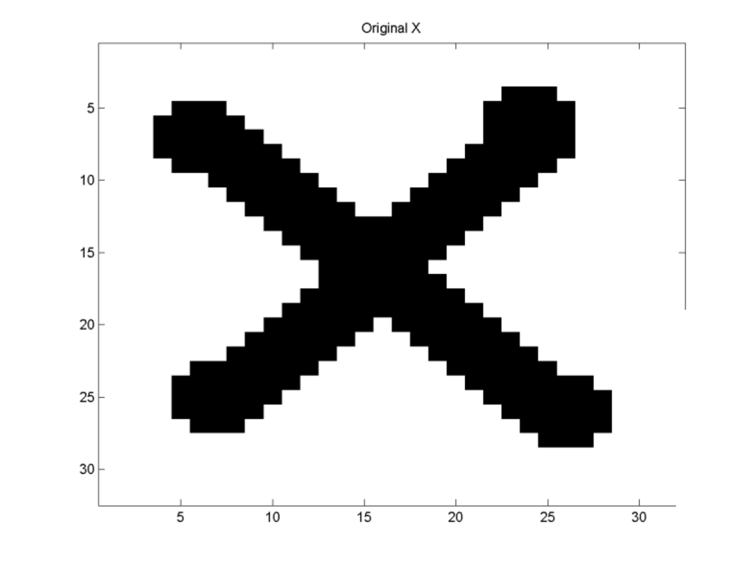


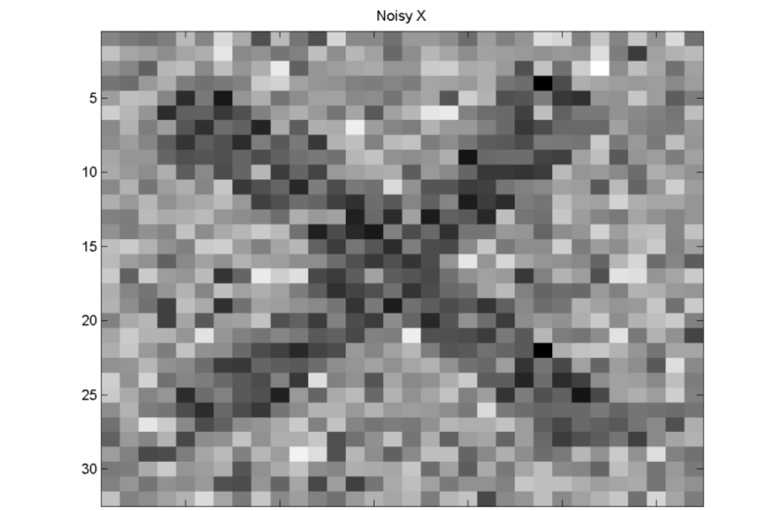
Inference in Jtrees

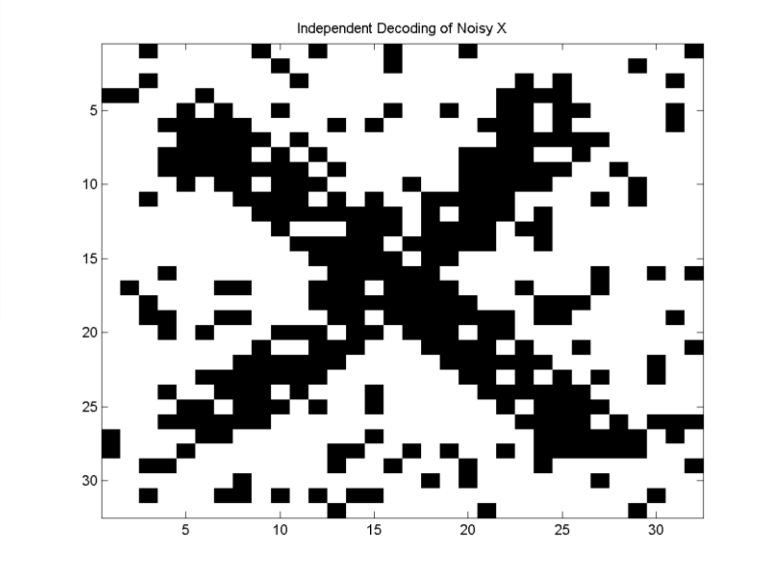
- For every elimination order ψ, we will get a different Jtree
- The time complexity of sending messages in each direction in a Jtree generated with elimination order ψ is $O(n2^{\omega(\psi)})$.
- With spending twice the time of VE, we can have the probabilities for all random variables.

Demos

- Go though the GraphCuts demo








THANK
YOU!