Junction Tree Algorithm

- completely solve the inference problem ⇒ find marginals of all cliques in one algorithm

- very similar to BP algorithm

BP: works on tree, works on “clique tree”

Does not work on graphs with loops in its corresponding clique graph

Edges on clique tree do not encode any probability information
Question: Can we get rid of some clique graph edges to form a tree so that we can run BP?

Two examples:

The right one is more sensible because the marginalization at each node (for creating message) involves only "locally-controlled" node.

* Important property: to incorporate dependencies of clique nodes, we want the SUB-GRAPH of the clique induced by ANY RANDOM VARIABLE to be CONNECTED.

⇒ "Junction Tree".

* Not all clique graphs have a JT.
What should we do? Add Edges!

How many edges we need to add? Where should we add them?

**JT algorithm solves all that:**

1. If you start with an directed graph, moralize it first.
2. Choose an elimination order. ⇒ reconstituted graph (add edge to neighbors of eliminated node)
3. Start with the reconstituted graph $G_0$
4. Draw the corresponding clique graph $CG(G_0)$
5. $T :=$ maximum spanning tree of $CG(G_0)$
6. Messages passed up from leaves:

   \[
   \phi^{(3)}(S_{3k}) = \sum_{S_{3k}} \phi^{(3)}(X_{3})
   \]

\[\xrightarrow{\text{Step 2}}\]

\[\phi^{(i)}(X_{C_3}) = \phi^{(i)}(X_{C_3}) \phi^{(0)}(X_{S_{3,3}}) \phi^{(i)}(X_{S_{3,3}})\]

8. Messages passed down from root:

   JT property (as we'll see) implies
   \[
   \phi^{(1)}(X_c) = \mathbb{P}(X_c)
   \]
   \[
   \phi^{(2)}(X_c) = \mathbb{P}(X_c) !\!\!\!
   \]