From Bucket-Elimination To Bucket Trees

Bucket E: \( P(E|B,C) \)
Bucket D: \( P(D|A,B) \)
Bucket C: \( P(C|A) \)
Bucket B: \( P(B|A) \)
Bucket A: \( P(A) \)

\( T = \) \[
\begin{array}{c}
\text{E} \\
\text{D} \\
\text{C} \\
\text{B} \\
\text{A}
\end{array}
\]

**Definition:** \( T \) is a bucket tree.

**Theorem:** \( T \) is an i-map of \( G \).
- Variable-elimination can be viewed as message-passing (elimination) using a rooted bucket tree.
- Any variable (bucket) can be the root.
Generalization: Eliminate (sum over) Variables Not in Separators

- Multiply all incoming messages, and $P_i$'s in the bucket and sum over $B_1 \cap B_2$.
- $\lambda^{B_2}_{B_1}(s) = \sum_{B_1-s} (\prod \lambda_i) \cdot (\prod P_i)$
- Given a rooted bucket tree, $T$, every node can be the “root” of the variables-elimination computation.
- If $B_3$ is the root, bucket $B_2$ and then Bucket $B_1$ should be processed; $\pi$-messages sent from $B_2$ to $B_1$ and from $B_1$ To $B_3$
Bucket Propagation Algorithm

- **Input:** A bucket tree \( B_1 \ldots B_n \)

- **Output:** For Each \( B_i \) and parent \( B_j \), functions \( \lambda_i^j(S_{ij}) \) and \( \pi_i^j(S_{ij}) \) are exchanged.

\[
B_i \cap B_j = S_{ij} \quad \lambda \quad B_j
\]

**Top Down:**
- Let \( s \lambda_1 \ldots \lambda_k \) messages from child nodes of \( B_i \), \( P_1 \ldots P_r \) in \( B_i \) original functions.

\[
\lambda_i^j(S_{ij}) = \sum_{B_i-B_j} \Pi_i \lambda_i \cdot \Pi_j P_j
\]

**Bottom Up:**
- Let \( \pi_i^j \) be received from \( B_j \).

\[
\pi_i^k(S_{ki}) = \sum_{B_k-B_i} (\Pi_j P_j) \cdot \pi_i^j \cdot \Pi_{i \neq k} \lambda_i
\]
• The belief of $B_i$

• $P(B_i) = \prod_{i} P_j \cdot \prod_{i} \lambda_i \cdot \pi_j$

• if $x$ index Bucket $i$
  get $\text{Bel}(x)$ by summing out $\text{Bel}(x) = \alpha \sum_{S_{ij}} P(B_i)$
Propagation in a Bucket Tree

Definitions:
- Let $G$ be a Bayesian network, $d$, an ordering and $B_1 \ldots B_n$ the final bucket created processing along $d = x_1 \ldots x_n$.
- Let $B_i$ be the set of variables appearing in bucket $i$ when it is processed.

Bucket Tree:
- A bucket tree has each $B_i$ cluster as a node and there is an arc from $B_i$ to $B_j$ if the function created at $B_i$ was placed in $B_j$.

Graph-Based Definition:
- Let $G_d$ be the induced graph along $d$. Each variable $x$ and it’s earlier neighbors in a node, $B_x$. There is an arc from $B_x$ to $B_y$ if $y$ is the closest parent of $x$. 

Upwards Messages On The Bucket Tree

\[ \Pi(A) = P(A) \]
\[ \Pi_B^P(A, B) = P(B, A) \bullet \lambda_C^B(A, B) \]
\[ \Pi_B^C(A, B) = P(B, A) \bullet \Pi(A) \bullet \lambda_D^B(A, B) \]
\[ \Pi_C^E(B, C) = \sum_A P(C, A) \bullet \Pi_B^C(A, B) \]