10.3. Message Passing: Belief Update

**Algorithm 10.3** Calibration using belief propagation in clique tree

```
Procedure CTree-BU-Calibrate (
    \Phi, // Set of factors
    \mathcal{T} // Clique tree over \Phi
)

1) Initialize-CTree
2) while exists an uninformed clique in \mathcal{T}
3) Select \((i-j) \in \mathcal{E}_\mathcal{T}\)
4) BU-Message\((i, j)\)
5) return \{\beta_i\}

Procedure Initialize-CTree ( )

1) for each clique \(C_i\)
2) \(\beta_i \leftarrow \prod_{\phi: \alpha(\phi)=i} \phi\)
3) for each edge \((i-j) \in \mathcal{E}_\mathcal{T}\)
4) \(\mu_{i,j} \leftarrow 1\)

Procedure BU-Message ( i, // sending clique
j // receiving clique )

1) \(\sigma_{i\rightarrow j} \leftarrow \sum_{C_i} s_{i,j} \beta_i\)
2) // marginalize the clique over the sepset
3) \(\beta_j \leftarrow \beta_j \cdot \frac{\sigma_{i\rightarrow j}}{\mu_{i,j}}\)
4) \(\mu_{i,j} \leftarrow \sigma_{i\rightarrow j}\)
```

The precise algorithm is shown in algorithm 10.3. Note that, as written, the message passing algorithm is underspecified: in line 3, we can select any pair of cliques \(C_i\) and \(C_j\) between which we will pass a message. Interestingly, we can make this choice arbitrarily, without damaging the correctness of the algorithm. For example, if \(C_i\) (for some reason) passes the same message to \(C_j\) a second time, the process of dividing out by the stored message reduces the message actually passed to 1, so that it has no influence. Furthermore, if \(C_i\) passes a message to \(C_j\) based on partial information (that is, without taking into consideration all of its incoming messages), and then resends a more updated message later on, the effect is identical to simply sending the updated message once. Moreover, at convergence, regardless of the message passing steps used, we necessarily have a calibrated clique tree. This property follows from the fact that, in order for all message updates to have no effect, we need to have