the structure of this factor set (see figure 11.14), we see that this computation can be done using standard exact inference on the factor set. In this particular case, the factor set is a tree network, and so inference is cheap. Similarly, we can compute the marginals over the variables in $\delta_{2\to3}$. Thus, we can use the properties of M-projections and exact inference to compute the resulting projected message without ever explicitly enumerating the joint over the four variables in $\delta_{2\to3}$. In an $n \times n$ grid, for example, we can perform these operations in time that is linear in n, whereas the explicit computation would have constructed a factor of exponential size. As we discussed, the more complex approximation of figure 11.13b also has a bounded tree-width of 2, and therefore also allows linear time inference.

Algorithm 11.4 Projecting a factor set to produce a set of marginals over a given set of scopes

Procedure Factored-Project (
  $\vec{\phi}$, // an input factor set
  $\mathcal{Y}$, // A set of desired output scopes
)
  Build an inference data structure $\mathcal{U}$
  Initialize $\mathcal{U}$ with factors in $\vec{\phi}$
  Perform inference in $\mathcal{U}$
  $\vec{\phi}^o \leftarrow \emptyset$
  for all $\boldsymbol{Y}_j \in \mathcal{Y}$
    Find a clique $\boldsymbol{C}_j$ in $\mathcal{U}$ so that $\boldsymbol{Y}_j \subseteq \boldsymbol{C}_j$
    $\psi_j \leftarrow \beta_{\mathcal{U}}(\boldsymbol{Y}_j)$ // marginal of $\boldsymbol{Y}_j$ in $\mathcal{U}$
    $\vec{\phi}^o \leftarrow \vec{\phi}^o \cup \{\psi_j\}$
  return $\vec{\phi}^o$

The overall structure of the algorithm is shown in algorithm 11.4. At a high level, we can view exact inference in the factor set as a "black box" and not concern ourselves with the exact implementation. However, it is also useful to consider how this type of projected message may be computed efficiently. Most often, the inference data structure $\mathcal{U}$ is a cluster graph or a clique tree, into which the initial factors $\vec{\phi}$ can easily be incorporated, and from which the target factors, of the appropriate scopes, can be easily extracted. To allow for that, we typically design the cluster graph to be family-preserving with respect to both sets of factors. Under that assumption, we can extract a factor ψ_j over the scope $\boldsymbol{Y}_j$ by identifying a cluster $\boldsymbol{C}_j$ in $\mathcal{U}$ whose scope contains $\boldsymbol{Y}_j$, and marginalizing it over $\boldsymbol{Y}_j$. As an alternative approach, we can construct an unconstrained clique tree, and use the out-of-clique inference algorithm of section 10.3.3.2 to extract from the graph the joint marginals of subsets of variables that are not together in a clique. (We note that out-of-clique inference is more challenging in the context of cluster graphs, since the path used to relate the clusters containing the query variables can affect the outcome of the computation; see exercise 11.22.)

If our representation of a message is simply a product of marginals over disjoint subsets of variables, this algorithm suffices to produce the output message: We produce a factor over each (disjoint) scope $\boldsymbol{Y}_j$; the product of these factors is the M-projection of the distribution. But for