

Algorithm 9.2 Using Sum-Product-VE for computing conditional probabilities

```

Procedure Cond-Prob-VE (
     $\mathcal{K}$ , // A network over  $\mathcal{X}$ 
     $\mathbf{Y}$ , // Set of query variables
     $\mathbf{E} = \mathbf{e}$  // Evidence
)
1   $\Phi \leftarrow$  Factors parameterizing  $\mathcal{K}$ 
2  Replace each  $\phi \in \Phi$  by  $\phi[\mathbf{E} = \mathbf{e}]$ 
3  Select an elimination ordering  $\prec$ 
4   $\mathbf{Z} \leftarrow \mathcal{X} - \mathbf{Y} - \mathbf{E}$ 
5   $\phi^* \leftarrow$  Sum-Product-VE( $\Phi, \prec, \mathbf{Z}$ )
6   $\alpha \leftarrow \sum_{\mathbf{y} \in \text{val}(\mathbf{Y})} \phi^*(\mathbf{y})$ 
7  return  $\alpha, \phi^*$ 

```

Step	Variable eliminated	Factors used	Variables involved	New factor
1'	C	$\phi_C(C), \phi_D(D, C)$	C, D	$\tau'_1(D)$
2'	D	$\phi_G[I = i^1](G, D), \phi_I[I = i^1](), \tau'_1(D)$	G, D	$\tau'_2(G)$
5'	G	$\tau'_2(G), \phi_L(L, G), \phi_H[H = h^0](G, J)$	G, L, J	$\tau'_5(L, J)$
6'	S	$\phi_S[I = i^1](S), \phi_J(J, L, S)$	J, L, S	$\tau'_6(J, L)$
7'	L	$\tau'_6(J, L), \tau'_5(J, L)$	J, L	$\tau'_7(J)$

Table 9.3 A run of sum-product variable elimination for $P(J, i^1, h^0)$

elimination ordering that we used in table 9.1. The results are shown in table 9.3; the step numbers correspond to the steps in table 9.1. It is interesting to note the differences between the two runs of the algorithm. First, we notice that steps (3) and (4) disappear in the computation with evidence, since I and H do not need to be eliminated. More interestingly, by not eliminating I , we avoid the step that correlates G and S . In this execution, G and S never appear together in the same factor; they are both eliminated, and only their end results are combined. Intuitively, G and S are conditionally independent given I ; hence, observing I renders them independent, so that we do not have to consider their joint distribution explicitly. Finally, we notice that $\phi_I[I = i^1] = P(i^1)$ is a factor over an empty scope, which is simply a number. It can be multiplied into any factor at any point in the computation. We chose arbitrarily to incorporate it into step (2'). Note that if our goal is to compute a conditional probability given the evidence, and not the probability of the evidence itself, we can avoid multiplying in this factor entirely, since its effect will disappear in the renormalization step at the end.

network
polynomial

Box 9.A — Concept: The Network Polynomial. *The network polynomial provides an interesting and useful alternative view of variable elimination. We begin with describing the concept for the case of a Gibbs distribution parameterized via a set of full table factors Φ . The polynomial f_{Φ}*