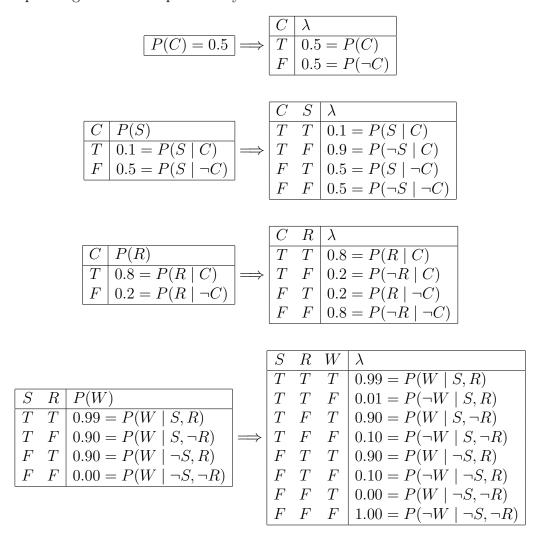
A Variable Elimination Algorithm for Belief Networks

This handout describes a general algorithm for computing $P(X \mid E)$ in a belief network, where X is a node and E is the evidence (those nodes known to be true or false). The example is computing $P(R \mid W)$ for the network in Figure 14.11, p. 510. The algorithm (named elim-bel) comes from Rina Dechter, Bucket elimination: A unifying framework for probabilistic inference, *Proceedings 1996 Conference on Uncertainty in Artificial Intelligence*.

Conversion of Conditional Probability Tables to λ Tables

Each conditional probability table is converted to a λ (lambda) table. Each possible value assignment to the node and its parents are associated with a λ value, which is initialized to the corresponding conditional probability.



Eliminating Evidence Variables

It is given that W is true. In each λ table that contains W, we delete all rows that have W as false, and eliminate the W column. [If W was given as false, then we would delete the rows that had W as true.]

S	R	\overline{W}	λ				
T	T	T	0.99				
T	T	F	0.01		S	R	λ
T	F	T	0.90		T	T	0.99
T	F	F	0.10	\Longrightarrow	T	F	0.90
F	T	T	0.90		F	T	0.90
F	T	F	0.10		F	F	0.00
F	F	T	0.00				
F	F	F	1.00				

Eliminating Other Variables

Now we want to eliminate S and C, leaving only R, the variable that we are interested in. First, we eliminate S. [Order of elimination does not affect correctness.]

- 1. First find all the λ tables that reference S. This includes the S, R λ table just created by evidence elimination, and the C, S λ table generated earlier.
- 2. These tables also reference C and R. The new λ table is for these two variables.
- 3. To compute an entry in the new C, R λ table (say C false and R true), we add two values. One value is the product of the λ values that are consistent with C false, R true, and S true. The other value is the product of the λ values that are consistent with C false, R true, and S false.

To eliminate C, there is one C λ table and two C, R λ tables to combine.

Now, we have one λ table for R alone, and we can compute $P(R \mid W)$ by:

$$P(R \mid W) = \frac{0.4581}{0.4581 + 0.1890} \approx 0.7079$$

If we ended up with more than one λ table for R, we would compute one value p by multiplying all the λ values for R as true, compute a second value q by multiplying all the λ values for R as false, and then perform p/(p+q).

Pseudocode for Elim-Bel

In this pseudocode, I assume the existence of the following subroutines. Variables (a) returns the set of variables in a, which might be a belief network, a value assignment, or a lambda table. Parents(x) returns the set of the parents of the variable x. Assignments(S) returns all possible value assignments to a set of variables S; if S has n variables, then Assignments(S) returns 2^n assignments. $P(x \mid v)$ looks up the probability that x is true given value assignment v in the probability table for x. Lambda(v, ltable) returns the value in the lambda table ltable consistent with value assignment v; any variables in v not mentioned by ltable are ignored.

Elim-Bel determines the probability that x is true given value assignment v. v does not assign a value to x. v can be a partial value assignment.

function Elim-Bel(x, v)

```
/* Create lambda tables. */
ltables \leftarrow \emptyset
for each y in Variables (belief net) do
   S \leftarrow \text{Parents}(y)
   ltable \leftarrow new lambda table
   for each u in Assignments(S)
       Lambda(u \cup \{y = true\}, ltable) \leftarrow P(y \mid u)
      Lambda(u \cup \{y = false\}, ltable) \leftarrow 1 - P(y \mid u)
   end for
   Add ltable to ltables
end for
/* Eliminate evidence variables. */
for each ltable in ltables do
   Remove all LAMBDA(u, ltable) values where u is inconsistent with v
   Remove Variables (v) from ltable
end for
```

```
/* Eliminate other variables. */
for each y in Variables (ltables) - \{x\} do
   ytables \leftarrow \text{subset of } ltables \text{ that refer to } y
   S \leftarrow \text{Variables}(ytables) - \{y\}
   ltable \leftarrow \text{new lambda table}
   for each u in Assignments(S) do
       ytrue \leftarrow 1
       yfalse \leftarrow 1
       for each ytable in ytables do
           ytrue \leftarrow ytrue * Lambda(u \cup \{y = true\}, ytable)
           yfalse \leftarrow yfalse * Lambda(u \cup \{y = false\}, ytable)
       end for
       Lambda(u, ltable) \leftarrow ytrue + yfalse
   end for
   Remove ytables from ltables
   Add ltable to ltables
end for
/* Calculate conditional probability. */
xtrue \leftarrow 1
xfalse \leftarrow 1
for each ltable in ltables do
   xtrue \leftarrow xtrue * Lambda(\{x = true\}, ltable)
   xfalse \leftarrow xfalse * Lambda(\{x = false\}, ltable)
end for
return xtrue/(xtrue + xfalse)
```