With two customers and two books, the Bayes net looks like the one in Figure 14.17(b). For larger numbers of books and customers, it becomes completely impractical to specify the network by hand.

Fortunately, the network has a lot of repeated structure. Each Recommendation variable has as its parents the variables Honest, Kindness, and Quality. Moreover, the CPTs for all the Recommendation variables are identical, as are those for all the Honest variables, and so on. The situation seems tailor-made for a first-order language. We would like to say something like

\[ \text{Recommendation}(c, b) \sim \text{RecCPT}(\text{Honest}(c), \text{Kindness}(c), \text{Quality}(b)) \]

with the intended meaning that a customer’s recommendation for a book depends on the customer’s honesty and kindness and the book’s quality according to some fixed CPT. This section develops a language that lets us say exactly this, and a lot more besides.

### 14.6.1 Possible worlds

Recall from Chapter 13 that a probability model defines a set of possible worlds with a probability \( P(\omega) \) for each world \( \omega \). For Bayesian networks, the possible worlds are assignments of values to variables; for the Boolean case in particular, the possible worlds are identical to those of propositional logic. For a first-order probability model, then, it seems we need the possible worlds to be those of first-order logic—that is, a set of objects with relations among them and an interpretation that maps constant symbols to objects, predicate symbols to relations, and function symbols to functions on those objects. (See Section 8.2.) The model also needs to define a probability for each such possible world, just as a Bayesian network defines a probability for each assignment of values to variables.

Let us suppose, for a moment, that we have figured out how to do this. Then, as usual (see page 485), we can obtain the probability of any first-order logical sentence as a sum over the possible worlds where it is true

\[ P(\phi) = \sum_{\omega \in \Omega, \phi \text{ is true in } \omega} P(\omega) \]  

(14.13)

Conditional probabilities \( P(\phi | e) \) can be obtained similarly, so we can, in principle, ask any question we want of our model—e.g., “Which books are most likely to be recommended highly by dishonest customers?”—and get an answer. So far, so good.

There is, however, a problem: the set of first-order models is infinite. We saw this explicitly in Figure 8.4 on page 293, which we show again in Figure 14.18 (top). This means that (1) the summation in Equation (14.13) could be infeasible, and (2) specifying a complete, consistent distribution over an infinite set of worlds could be very difficult.

Section 14.6.2 explores one approach to dealing with this problem. The idea is to borrow not from the standard semantics of first-order logic but from the database semantics defined in Section 8.2.8 (page 299). The database semantics makes the unique names assumption—here, we adopt it for the constant symbols. It also assumes domain closure—there are no nonexistent objects than those that are named. We can then guarantee a finite set of possible worlds by making the set of objects in each world be exactly the set of constant
symbols that are used; as shown in Figure 14.18 (bottom), there is no uncertainty about the mapping from symbols to objects or about the objects that exist. We will call models defined in this way relational probability models, or RPMs. The most significant difference between the semantics of RPMs and the database semantics introduced in Section 8.2.8 is that RPMs do not make the closed-world assumption—obviously, assuming that every unknown fact is false doesn’t make sense in a probabilistic reasoning system!

When the underlying assumptions of database semantics fail to hold, RPMs won’t work well. For example, a book retailer might use an ISBN (International Standard Book Number) as a constant symbol to name each book, even though a given "logical" book (e.g., "Gone With the Wind") may have several ISBNs. It would make sense to aggregate recommendations across multiple ISBNs, but the retailer may not know for sure which ISBNs are really the same book. (Note that we are not reifying the individual copies of the book, which might be necessary for used-book sales, car sales, and so on.) Worse still, each customer is identified by a login ID, but a dishonest customer may have thousands of IDs! In the computer security field, these multiple IDs are called sibyls and their use to confound a reputation system is called a sibyl attack. Thus, even a simple application in a relatively well-defined, online domain involves both existence uncertainty (what are the real books and customers underlying the observed data) and identity uncertainty (which symbol really refer to the same object). We need to bite the bullet and define probability models based on the standard semantics of first-order logic, for which the possible worlds vary in the objects they contain and in the mappings from symbols to objects. Section 14.6.3 shows how to do this.

*The name relational probability model was given by Pfeffer (2000) to a slightly different representation, but the underlying ideas are the same.*