We can go through each function and predicate, writing down what we know in terms of the other symbols. For example, one's mother is one's female parent:

\[ \forall r, e \, \text{Mother}(e) = m \iff \text{Female}(m) \land \text{Parent}(m, e) \]

One's husband is one's male spouse:

\[ \forall h, w \, \text{Husband}(h, w) \land \text{Male}(h) \land \text{Spouse}(h, w) \]

Male and female are disjoint categories:

\[ \forall x \, \text{Male}(x) \land \neg \text{Female}(x) \]

Parent and child are inverse relations:

\[ \forall p, c \, \text{Parent}(p, c) \iff \text{Child}(p, c) \]

A grandparent is a parent of one's parent:

\[ \forall g, c \, \text{Grandparent}(g, c) \iff \exists p \, \text{Parent}(g, p) \land \text{Parent}(p, c) \]

A sibling is another child of one's parents:

\[ \forall x, y \, \text{Sibling}(x, y) \iff \exists p \, \text{Parent}(p, x) \land \text{Parent}(p, y) \]

We could go on for several more pages like this, and Exercise 8.14 asks you to do just that.

Each of these sentences can be viewed as an **axiom** of the kinship domain, as explained in Section 7A. Axioms are commonly associated with purely mathematical domains—we will see some axioms for numbers shortly—but they are needed in all domains. They provide the basic factual information from which useful conclusions can be derived. Our kinship axioms are also **definitions**; they have the form \( \forall x, y \, P(x, y) \). The axioms define the **Mother** function and the **Husband, Male, Parent, Grandparent, and Sibling** predicates in terms of other predicates. Our definitions "bottom out" at a basic set of predicates (**Child, Spouse, and Female**) in terms of which the others are ultimately defined. This is a natural way in which to build up the representation of a domain, and it is analogous to the way in which software packages are built up by successive definitions of subroutines from primitive library functions. Notice that there is not necessarily a unique set of primitive predicates; we could equally well have used **Parent, Spouse, and Male**. In some domains, as we show, there is no clearly identifiable basic set.

Not all logical sentences about a domain are axioms. Some are theorems—that is, they are entailed by the axioms. For example, consider the assertion that siblinghood is symmetric:

\[ \forall x, y \, \text{Sibling}(x, y) \iff \text{Sibling}(y, x) \]

Is this an axiom or a theorem? In fact, it is a theorem that follows logically from the axiom that defines siblinghood. If we ASK the knowledge base this sentence, it should return true.

From a purely logical point of view, a knowledge base need contain only axioms and no theorems, because the theorems do not increase the set of conclusions that follow from the knowledge base. From a practical point of view, theorems are essential to reduce the computational cost of deriving new sentences. Without them, a reasoning system has to start from first principles every time, rather like a physicist having to rederive the rules of calculus for every new problem.
Not all axioms are definitions. Some provide more general information about certain predicates without constituting a definition. Indeed, some predicates have no complete definition because we do not know enough to characterize them fully. For example, there is no obvious definitive way to complete the sentence

\[ \forall x \ Person(x) \]

\textbf{Fortunately,} first-order logic allows us to make use of the \textit{Person} predicate without completely defining it. Instead, we can write partial specifications of properties that every person has and properties that make something a person:

\[ \forall x \ Person(x) \]
\[ \forall x \ \ldots \ Person(x) \]

Axioms can also be "just plain facts," such as \textit{Male(Jim)} and \textit{Spouse(Jim, Laura)} Such facts form the descriptions of specific problem instances, enabling specific questions to be answered. The answers to these questions will then be theorems that follow from the axioms. Often, one finds that the expected answers are not forthcoming—for example, from \textit{Spouse (Jim, Laura)} one expects (under the laws of many countries) to be able to infer \textit{~Spouse (George, Laura)}; but this does not follow from the axioms given earlier—even after we add \textit{Jim George} as suggested in Section 8.2.8. This is a sign that an axiom is missing. Exercise 8.8 asks the reader to supply it.

\section*{8.3.3 Numbers, sets, and lists}

Numbers are perhaps the most vivid example of how a large theory can be built up from a tiny kernel of axioms. We describe here the theory of \textbf{natural numbers} or non-negative integers. We need a predicate \textit{NatNum} that will be true of natural numbers; we need one constant symbol, 0; and we need one function symbol, S (successor). The \textbf{Nano axioms} define natural numbers and addition. Natural numbers are defined recursively:

\begin{align*}
\text{NatNum}(0) \\
\forall n \text{ NatNum}(n) & \implies \text{NatNum}(S(n)).
\end{align*}

That is, 0 is a natural number, and for every object n, if n is a natural number, then \(S(n)\) is a natural number. So the natural numbers are 0, \(S(0), S(S(0)), \) and so on. (After reading Section 8.2.8, you will notice that these axioms allow for other natural numbers besides the usual ones; see Exercise 8.12.) We also need axioms to constrain the successor function:

\begin{align*}
\forall m, n \text{ S}(m) & \implies \text{S}(n). \\
\forall m, n \text{ S}(m) & \implies \text{S}(n).
\end{align*}

Now we can define addition in terms of the successor function:

\begin{align*}
\text{H of } \forall m, n \text{ NatNum}(m) & \implies S(m) = S(n) \\
\forall m, n \text{ NatNum}(m) & \implies S(m) = S(n).
\end{align*}

The first of these axioms says that adding 0 to any natural number m gives m itself. Notice the use of the binary function symbol "\(+\)" in the term \(+(m, 0)\); in ordinary mathematics, the term would be written \(m + 0\) using \textbf{infix} notation. (The notation we have used for first-order logic also include the principle of induction, which is a sentence of second-order logic rather than of first-order logic. The importance of this distinction is explained in Chapter 9.)