page. For example, if the page contains a string of the form "<a ... add to cart ... </a>" then it is an offer. This could be represented in first-order logic, but it is more straightforward to encode it into program code. We show how to do more sophisticated information extraction in Section 22.4.

12.7.1 Following links

The strategy is to start at the home page of an online store and consider all pages that can be reached by following relevant links. The agent will have knowledge of a number of stores, for example:

\[
\text{Amazon E OnlineStores A Homepage(Amazon, "amazon.com")}
\]

\[
\text{Ebay E OnlineStores A Homepage(Ebay, "ebay.com")}
\]

\[
\text{ExampleStore E OnlineStores A Homepage(ExampleStore, "example.com")}
\]

These stores classify their goods into product categories, and provide links to the major categories from their home page. Minor categories can be reached through a chain of relevant links, and eventually we will reach offers. In other words, a page is relevant to the query if it can be reached by a chain of zero or more relevant category links from a store's home page, and then from one more link to the product offer. We can define relevance:

\[
\text{Relevant(page, query) } \iff \\
\text{store, home store } \epsilon \text{ OnlineStores A Homepage(store, home)} \\
\text{A } \exists url, url2 \text{ RelevantChain(home, url2, query) A Link(url2, url)} \\
\text{A page } \neg \text{ Contents(url)}
\]

Here the predicate \text{Link(from, to)} means that there is a hyperlink from the \text{from} URL to the \text{to} URL. To define what counts as a \text{RelevantCategoryName}, we need to follow not just any old hyperlinks, but only those links whose associated anchor text indicates that the link is relevant to the product query. For this, we use \text{LinkText(from, to, text)} to mean that there is a link between \text{from} and \text{to} with \text{text} as the anchor text. A chain of links between two URLs, \text{start} and \text{end}, is relevant to a description \text{d} if the anchor text of each link is a relevant category name for \text{d}. The existence of the chain itself is determined by a recursive definition, with the empty chain (\text{start = end}) as the base case:

\[
\text{RelevantChain(start, end, query) } \iff (\text{start = end)} \\
\text{V } u, text \text{ RelevantChain(start, u, text) A Relevant CategoryName(query, text)} \\
\text{A Relevant CategoryName(cont, end, query))}
\]

Now we must define what it means for \text{tea} to be a \text{Relevant CategoryName} for \text{query}. First, we need to relate strings to the categories they name. This is done using the predicate \text{Name(s, e)}, which says that string \text{s} is a name for category \text{c}—for example, we might assert that \text{Name("laptops", "Laptop Computers")}. Some more examples of the \text{Name} predicate appear in Figure 12.9(b). Next, we define relevance. Suppose that \text{query} is "laptops." Then \text{Relevant CategoryName(query, text)} is true when one of the following holds:

- The text and query name the same category—e.g., "notebooks" and "laptops."

An alternative to the link-following strategy is to use an Internet search engine; the technology behind Internet search, information retrieval, will be covered in Section 22.3.
The text names a supercategory such as "computers."

• The text names a subcategory such as "ultralight notebooks."

The logical definition of RelevantCategoryName is as follows:

\[
\text{RelevantCategoryName(query, text)} \iff c_1, c_2 \ \text{Name(query, c_1)} \land \text{Name(text, c_2)} \land \left( c_1 \subset C \lor c_2 \subset C \right) .
\] (12.1)

Otherwise, the anchor text is irrelevant because it names a category outside this line, such as "clothes" or "lawn & garden."

To follow relevant links, then, it is essential to have a rich hierarchy of product categories. The top part of this hierarchy might look like Figure 12.9(a). It will not be feasible to list all possible shopping categories, because a buyer could always come up with some new desire and manufacturers will always come out with new products to satisfy them (electric kneecap warmers?). Nonetheless, an ontology of about a thousand categories will serve as a very useful tool for most buyers.

In addition to the product hierarchy itself, we also need to have a rich vocabulary of names for categories. Life would be much easier if there were a one-to-one correspondence between categories and the character strings that name them. We have already seen the problem of synonymy—two names for the same category, such as "laptop computers" and "laptops." There is also the problem of ambiguity—one name for two or more different categories. For example, if we add the sentence

\[
\text{Name("CDs", CertificatesOfDeposit)}
\]

to the knowledge base in Figure 12.9(b), then "CDs" will name two different categories.

Synonymy and ambiguity can cause a significant increase in the number of paths that the agent has to follow, and can sometimes make it difficult to determine whether a given page is indeed relevant. A much more serious problem is the very broad range of descriptions that a user can type and category names that a store can use. For example, the link might say "laptop" when the knowledge base has only "laptops" or the user might ask for "a computer..."