the information gain formula when some examples have unknown values for the attribute? These questions are addressed in Exercise 18.11.

- ◊ **Multivalued attributes**:When an attribute has a large number of possible values, the information gain measure gives an inappropriate indication of the attribute's usefulness. Consider the extreme case where every example has a different value for the attribute—for instance, if we were to use an attribute *RestaurantName* in the restaurant domain. In such a case, each subset of examples is a singleton and therefore has a unique classification, so the information gain measure would have its highest value for this attribute. However, the attribute may be irrelevant or useless. One possible solution is to use the **gain ratio**, as described in Exercise 18.12.
- ◊ **Continuous-valued attributes**: Attributes such as *Height* and *Weight* have a large or infinite set of possible values. They are therefore not well-suited for decision-tree learning in raw form. An obvious way to deal with this problem is to **discretize** the attribute. For example, the *Price* attribute for restaurants was discretized into $, $$, and $$$ values. Normally, such discrete ranges would be defined by hand. A better approach is to preprocess the raw attribute values during the tree-growing process in order to find out which ranges give the most useful information for classification purposes.

GAIN RATIO

A decision-tree learning system for real-world applications must be able to handle all of these problems. Handling continuous-valued variables is especially important, because both physical and financial processes provide numerical data. Several commercial packages have been built that meet these criteria, and they have been used to develop several hundred fielded systems.

18.5 LEARNING GENERAL LOGICAL DESCRIPTIONS

HYPOTHESIS SPACE

In this section, we examine ways in which more general kinds of logical representations can be learned. In the process, we will construct a general framework for understanding learning algorithms, centered around the idea that inductive learning can be viewed as a process of searching for a good hypothesis in a large space—the **hypothesis space**—defined by the representation language chosen for the task. We will also explain what is going on in logical terms: the logical connections among examples, hypotheses, and the goal. Although this may seem like a lot of extra work at first, it turns out to clarify many of the issues in learning. It enables us to go well beyond the simple capabilities of the decision tree learning algorithm, and allows us to use the full power of logical inference in the service of learning.

Hypotheses

CANDIDATE DEFINITION

The situation is usually this: we start out with a goal predicate, which we will generically call Q. (For example, in the restaurant domain, Q will be *WillWait*.) Q will be a unary predicate, and we are trying to find an equivalent logical expression that we can use to classify examples correctly. Each hypothesis proposes such an expression, which we call a **candidate definition** of the goal