```
if examples is empty then return PLURALITY-VALUE(parent_examples)
else if all examples have the same classification then return the classification
else if attributes is empty then return PLURALITY-VALUE(examples)
else
    A ← argmax_{a ∈ attributes} IMPORTANCE(a, examples)
    tree ← a new decision tree with root test A
    for each value v_k of A do
        exs ← {e : e ∈ examples and e.A = v_k}
        subtree ← DECISION-TREE-LEARNING(exs, attributes − A, examples)
        add a branch to tree with label (A = v_k) and subtree subtree
    return tree
```

Figure 18.5 The decision-tree learning algorithm. The function IMPORTANCE is d scribed in Section 18.3.4. The function PLURALITY-VALUE selects the most common outp value among a set of examples, breaking ties randomly.

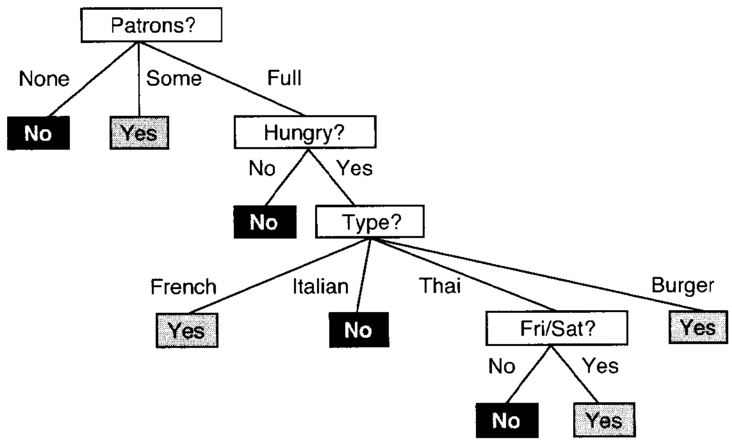

Figure 18.6 The decision tree induced from the 12-example training set.

In that case it says not to wait when *Hungry* is false, but I (SR) would certainly wait. more training examples the learning program could correct this mistake.

We note there is a danger of over-interpreting the tree that the algorithm selects. W there are several variables of similar importance, the choice between them is somewhat trary: with slightly different input examples, a different variable would be chosen to spli first, and the whole tree would look completely different. The function computed by the would still be similar, but the structure of the tree can vary widely.

LEARNING CURVE

We can evaluate the accuracy of a learning algorithm with a **learning curve**, as sh in Figure 18.7. We have 100 examples at our disposal, which we split into a training set