```
function FOIL(examples, target) returns a set of Horn clauses
  inputs: examples, set of examples
          target, a literal for the goal predicate
  local variables: clauses, set of clauses, initially empty

  while examples contains positive examples do
     clause ← NEW-CLAUSE(examples, target)
     remove examples covered by clause from examples
     add clause to clauses
  return clauses
```

```
function NEW-CLAUSE(examples, target) returns a Horn clause
  local variables: clause, a clause with target as head and an empty body
                   l, a literal to be added to the clause
                   extended-examples, a set of examples with values for new variables

  extended-examples ← examples
  while extended-examples contains negative examples do
     l ← CHOOSE-LITERAL(NEW-LITERALS(clause), extended-examples)
     append l to the body of clause
     extended-examples ← set of examples created by applying EXTEND-EXAMPLE
         to each example in extended-examples
  return clause
```

```
function EXTEND-EXAMPLE(example, literal) returns
  if example satisfies literal
     then return the set of examples created by extending example with
         each possible constant value for each new variable in literal
  else return the empty set
```

Figure 21.8 Sketch of the FOIL algorithm for learning sets of first-order Horn clauses from examples. NEW-LITERAL and CHOOSE-LITERAL are explained in the text.

NEW-LITERALS takes a clause and constructs all possible “useful” literals that could be added to the clause. Let us use as an example the clause

$$Father(x,z) \;\Rightarrow\; Grandfather(x,y)$$

There are three kinds of literals that can be added.

1. Literals using predicates: the literal can be negated or unnegated, any existing predicate (including the goal predicate) can be used, and the arguments must all be variables. Any variables can be used for any argument of the predicate, with one restriction: each literal must include *at least one* variable from an earlier literal or from the head of the clause. Literals such as $Mother(z,u)$, $Married(z,z)$, $\neg Male(y)$, and $Grandfather(v,x)$ are allowed, whereas $Married(u,v)$ is not. Notice that the use of the predicate from the head of the clause allows FOIL to learn *recursive* definitions.