2.4.3 A DFA to Recognize a Set of Keywords

We can apply the subset construction to any NFA. However, when we apply that construction to an NFA that was designed from a set of keywords, according to the strategy of Section 2.4.2, we find that the number of states of the DFA is never greater than the number of states of the NFA. Since in the worst case the number of states exponentiates as we go to the DFA, this observation is good news and explains why the method of designing an NFA for keywords and then constructing a DFA from it is used frequently. The rule for constructing the set of DFA states is as follows:

a) If \( q_0 \) is the start state of the NFA, then \([q_0]\) is one of the states of the DFA.

b) Suppose \( p \) is one of the NFA states, and it is reached from the start state along a path whose symbols are \( a_1a_2\cdots a_m \). Then one of the DFA states is the set of NFA states consisting of:

1. \( q_0 \)
2. \( p \)
3. Every other state of the NFA that is reachable from \( q_0 \) by following a path whose labels are a suffix of \( a_1a_2\cdots a_m \), that is, any sequence of symbols of the form \( a_{j}a_{j+1}\cdots a_m \).

Note that in general, there will be one DFA state for each NFA state \( p \). However, in step (b), two states may actually yield the same set of NFA states, and thus become one state of the DFA. For example, if two of the keywords begin with the same letter, say \( a \), then the two NFA states that are reached from \( q_0 \) by an

2.4. AN APPLICATION: TEXT SEARCH

are labeled \( a \) will yield the same set of NFA states and thus get merged in the DFA.

Example 2.15: The construction of a DFA from the NFA of Fig. 2.16 is shown in Fig. 2.17. Each of the states of the DFA is located in the same position as the state \( p \) from which it is derived using rule (b) above. For example, consider the state 135, which is our shorthand for \( \{1,3,5\} \). This state was constructed from state 2. It includes the start state 1, because every set of the DFA states does. It also includes state 3 because that state is reached from state 1 by a suffix \( a \) of the string \( w \) that reaches state 3 in Fig. 2.16.

The transitions for each of the DFA states may be calculated according to the subset construction. However, the rule is simple. From any set of states that includes the start state \( q_0 \) and some other states \( \{p_1,p_2,\ldots,p_r\} \), determines, for each symbol \( a \), where the \( p_i \)'s go in the NFA, and let this DFA state have a transition labeled \( a \) to the DFA state consisting of \( q_0 \) and all the targets of the