

Formal Language Selected Homework Chapter 4.2

1. Show that there exists an algorithm to determine whether or not $w \in L_1 - L_2$, for any given w and any regular languages L_1 and L_2 .
2. Show that there exists an algorithm for determining if $L_1 \subseteq L_2$, for any regular languages L_1 and L_2 .
5. A language is said to be a *palindrome* language if $L = L^R$. Find an algorithm for determining if a given regular language is a palindrome language.
12. Let L be any regular language on $\Sigma = \{a, b\}$. Show that an algorithm exists for determining if L contains any strings of even length.

Sol:

1. Since by Example 4.1 $L_1 - L_2$ is regular, there exists a membership algorithm for it.
2. If $L_1 \subseteq L_2$, then $L_1 \cup L_2 = L_2$. Since $L_1 \cup L_2$ is regular and we have an algorithm for set equality, we also have an algorithm for set inclusion.
5. From the dfa for L , construct the dfa for L^R , using the construction suggested in Theorem 4.2. Then use the equality algorithm in Theorem 4.7.
12. Here you need a little trick. If L contains no even-length strings, then

$$L \cap L((aa + ab + ba + bb)^*) = \emptyset.$$

The left side is regular, so we can use Theorem 4.6.

