Automata Theory: Solutions 5

```
                     X
                     X
number of         X  X
homeworks         X  X  X
                  X  X  X
            X     X  X  X     X
            X     X  X  X     X  X
            X     X  X  X     X  X
            X     X  X  X  X  X  X     X
            X     X  X  X  X  X  X     X
            X  X  X  X  X  X  X  X     X
            X  X  X  X  X  X  X  X  X  X
            -----------------------------
            1  2  3  4  5  6  7  8  9  10
                      grades
```

Problem 1
Demonstrate that, if L_1 is a regular language on the alphabet $\Sigma = \{a, b\}$, then the following subset of L_1 is also a regular language:

$$L_2 = \{w : \ w \in L_1 \text{ and the length of } w \text{ is even}\}.$$

Consider the language L_3 defined by the expression $((a + b)(a + b))^*$. Note that, since we describe L_3 by a regular expression, it is a regular language. This language includes *all even-length strings* on the alphabet $\{a, b\}$, and we may express L_2 as follows:

$$L_2 = L_1 \cap L_3.$$

Thus, L_2 is the intersection of two regular languages, which implies that it is also regular.

Problem 2
Describe a method for determining whether $L_1 \subseteq L_2$, for given regular languages L_1 and L_2.

The key observation is that $L_1 \subseteq L_2$ if and only if $L_1 \cap \overline{L_2} = \emptyset$. Given DFAs for L_1 and L_2, we can construct a DFA for $L_1 \cap \overline{L_2}$ and check whether the corresponding language is empty.

If the resulting automaton does *not* accept any strings, then $L_1 \cap \overline{L_2}$ is empty, which means that $L_1 \subseteq L_2$. If the automaton accepts some strings, then L_1 is *not* a subset of L_2.

Problem 3
Argue that the language $\{a^n b^{2n} : \; n \geq 0\}$ is not regular.

We suppose that the language is regular and derive a contradiction. If this language is regular, there is some m-state DFA that accepts it, where m is an unknown fixed number.

Suppose that we begin from the initial state of the DFA and trace a path for each of the following strings: $a^0, a^1, a^2, ..., a^m$. Since the total number of these strings is $m + 1$, two of them must lead to the same state; we denote these two strings a^i and a^j (see the picture).

Since the automaton accepts the string $a^i b^{2i}$, this string must lead from the initial state to a final state, as shown in the picture. Then, the string $a^j b^{2i}$ leads to the same final state, contradicting the fact that $a^j b^{2i}$ is *not* in the language.