Homework 9

1. Describe what is meant by *eliminating tail recursion*, when such an optimization can be applied and why it is a benefit?

2. It is true (I confirmed it) that

$$\text{if } \varnothing \text{ does not occur in } r \quad \text{then} \;\; L(r) \neq \{\}$$

holds, or equivalently

$$L(r) = \{\} \quad \text{implies} \quad \varnothing \text{ occurs in } r.$$

You can prove either version by induction on r. The best way to make more formal what is meant by '$\varnothing$ occurs in r', you can define the following function:

$$\begin{array}{lcl} occurs(\varnothing) & \stackrel{\text{def}}{=} & true \\ occurs(\epsilon) & \stackrel{\text{def}}{=} & false \\ occurs(c) & \stackrel{\text{def}}{=} & false \\ occurs(r_1 + r_2) & \stackrel{\text{def}}{=} & occurs(r_1) \vee occurs(r_2) \\ occurs(r_1 \cdot r_2) & \stackrel{\text{def}}{=} & occurs(r_1) \vee occurs(r_2) \\ occurs(r^*) & \stackrel{\text{def}}{=} & occurs(r) \end{array}$$

Now you can prove

$$L(r) = \{\} \quad \text{implies} \quad occurs(r).$$

The interesting cases are $r_1 + r_2$ and r^*. The other direction is not true, that is if $occurs(r)$ then $L(r) = \{\}$. A counter example is $\varnothing + a$: although $\varnothing$ occurs in this regular expression, the corresponding language is not empty. The obvious extension to include the not-regular expression, $\sim r$, also leads to an incorrect statement. Suppose we add the clause

$$occurs(\sim r) \stackrel{\text{def}}{=} occurs(r)$$

to the definition above, then it will not be true that

$$L(r) = \{\} \quad \text{implies} \quad occurs(r).$$

Assume the alphabet contains just a and b, find a counter example to this property.