Theoretical Exercises 2
Lexical Analysis

Please submit solutions on Blackboard by Friday, 12.02.2021 14:00h

2.1 Regular languages

a. A *palindrome* is a word (or set of words, ignoring whitespace, punctuation characters and capitalization) that reads the same when read from left to right and from right to left.

Simple examples are "*rotor*" and "*madam*", but you can also come up with sophisticated palindromes such as the Finnish word "*saippuakuppinippukauppias*" and sentences like "*A man, a plan, a canal – Panama*" (however, this example only works when ignoring any whitespace and punctuation characters).

Are palindromes regular languages? Justify your answer (note that we don't expect some formal proof here, rather some sort of intuition).

b. In scientific notation, all numbers are written in the form $m \times 10^n$ or, as shorthand, mEn. The integer n is called the *order of magnitude* and the real number m is called the *significand* or *mantissa*.

In **normalized notation**, the exponent is chosen so that the absolute value (modulus) of the significand m is at least 1 but less than 10.

Examples for normalized notation are Avogadro's constant $= 6.022E23$ or the speed of light $c = 2.99792458E8\frac{m}{s}$.

Draw a deterministic finite automaton (DFA) which accepts numbers in **normalized notation** (no units required) using either the "E" or the "10^n" notation, the latter written as "`*10^n`".

c. Create a regular expression that generates the language over the alphabet `{a, b}` of all strings containing an even number of `b`s.

Examples of strings in the language:

```
ε
a
aa
aaa
abba
bb
bab
abbaa
aababbb
```

Examples of strings *not* in the language:

```
b
abbb
babab
abbbabab
bbaaba
```

Hint: make sure that your regular expression generates bs in pairs.

d. Create a regular expression that generates the language over the alphabet `{a, b}` of all strings in which

$n \bmod m = 2$

where n is the length of the string. In other words, all strings in the language have a length of 2, or 5, or 8, or 11, etc.

Examples of strings in the language:

```
ab
bb
baabb
aaaaa
abbbbbba
```

Examples of strings *not* in the language:

```
ε
a
abb
baaa
babbba
```

Hint: generate two initial letters, and expand the string by appending chunks of length 3.

2.2 NFAs and DFAs

a. Construct an NFA that accepts the following regular expression:

$$(\texttt{mars} \mid \texttt{mass} \mid \texttt{miss})$$

b. Convert your NFA to a DFA using the subset construction algorithm.

2.3 DFA minimization

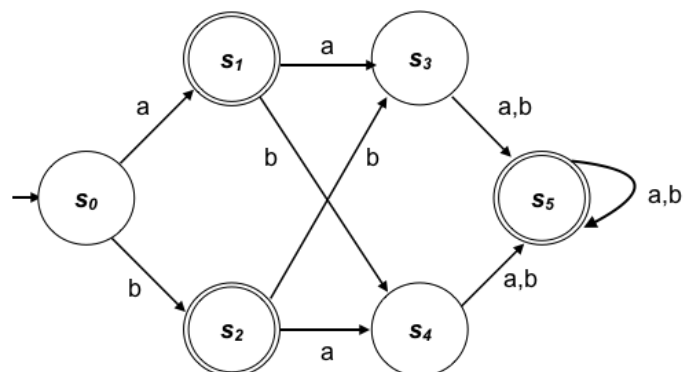Use the table method (Myhill-Nerode) to minimize the DFA given in Fig. 1. Show the steps performed, the related changes to the table and the final minimized DFA.

Figure 1: Non minimized DFA