



Assignment 1

Please submit solutions on Blackboard by Monday, 03.02.2020 20:00h

1.1 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*”.

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. Filenames in MS-DOS are restricted to the following format:

Up to eight characters for the name followed by a dot (“.”) and up to three characters for the extension.

Legal characters for DOS filenames include the following¹:

- Upper case letters A–Z
- Numbers 0–9
- Space

Examples for valid names are CONFIG.SYS, WORD.EXE, but also A FILE.123

Build a regular expression that accepts MS-DOS filenames (no automaton required).

1.2 NFAs and DFAs

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

(mars | mass | miss)

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

¹In reality, some more characters are in fact allowed, see https://en.wikipedia.org/wiki/8.3_filename – but we didn’t want the regexp to become too complex.

1.3 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.

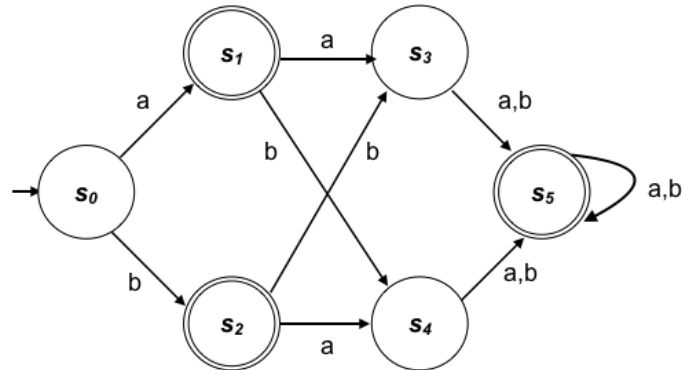


Figure 1: Non minimized DFA

1.4 Hands-on with scanner generators

- a. The Unix tool `wc` (word count) outputs the following information about a given text file:
 - Number of lines in the file
 - Number of words in the file (separated by whitespace or punctuation)
 - Number of characters in the file (including whitespace, punctuation etc.)

Implement a version of `wc` using a lex scanner that outputs these three values for a given input.

Hint:

You don't need to implement file handling, running your scanner from standard input like this:

```
$ mywc < textfile.txt
```

is perfectly fine.

- b. Extend your `wc` tool to also count the number of strings (delimited by double quotes) in the file and output the average string length. You may assume that a string never extends beyond the end of a line.
- c. Run your code against the example test cases provided on Blackboard and submit your output.

A skeleton lex source file, make file and test cases are available in the file `compilers-pl.tar.gz`.