

Compiler Construction 2019

Assignment 2: Lexical Analysis

1 NFA and DFA

Consider the following regular expression.

$$a(bb|c)^*b^*$$

Do the following tasks:

1. Draw a non-deterministic finite automaton (NFA) for the expression, and
2. transform it to a deterministic finite automaton (DFA).

2 Expressiveness

Explain why there are no regular expressions which can express the language $a^n b^n$, where n is any positive number.

3 Writing Regular Expressions (1)

Write a regular expression that matches *banana*, *bananas*, *ananas*, *ananana*, *anna*, but not *baanana*, *anannass*, *anaa*, *ans*.

Make the expression as short as possible.

4 Writing Regular Expressions (2)

Write a regular expression for non-negative integers with a time unit suffix. The possible suffixes are “ms” (milliseconds), “s” (seconds), “m” (minutes), “h” (hours), and “d” (days). Ensure that only integers that make sense are matched. For example, “25h” should not be possible, nor should “1200ms” be a match (instead, they should be written as “1d 1h” and “1s 200ms” respectively). Multiple units can be combined using spaces (but do not have to be in a specific order).

Write a single expression. Examples are as follows:

Matches 256ms, 5s, 32m, 16h, 4d, “1d 1h”, “2h 1d 3s”.

No matches 05ms, 10e, -2h, 100s, 25h.