EDAN65: Compilers
Computer Science
Lund University

August 28, 2016

E02: Regular expressions and scanning

E02-1: Write a regular expression describing the language of all natural numbers, 0, 1, 2, 3, ... Unnecessary initial zeros are allowed, like 00135.

E02-2: Write a regular expression describing the language of all natural binary numbers 0, 1, 10, 11, 100, ..., but where unnecessary initial zeros are *not* allowed.

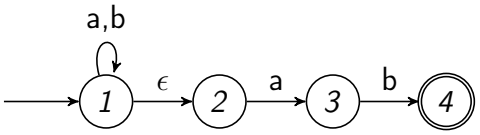
E02-3: Write a regular expression describing the language of all arithmetic expressions with natural numbers and the operators `+` and `*`, but without parentheses. Give some examples of expressions in the language.

E02-4: A *binary string* is a string over the binary alphabet 0, 1. A binary string may be the empty string, in contrast to binary numerals which will always have at least one digit. Write a regular expression describing the language of all binary strings that

a) contain the string `11`.
b) do not contain the string `11`.

E02-5: Construct
a) an NFA that accepts all binary strings that contain the string `11`. The automaton should not be deterministic.
b) a DFA that accepts all binary strings that contain the string `11`.

E02-6: Use simulation to construct a DFA that accepts the same language as the following NFA. Mark each state in the new automaton with the corresponding state numbers of the NFA.



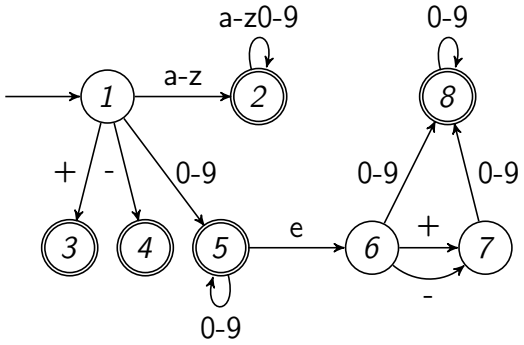
E02-7: Construct a DFA that accepts all binary strings that do not contain the string `11`.

E02-8: Construct a combined DFA recognizing binary integers and binary floating point numbers described by

```
BININT=[0-9]+
BNFLOAT=[0-9]+ "." [0-9]+
```

Make tables for a table-driven scanner.

E02-9: The following automaton describes a lexical analyzer. Give suitable names to the final states and write down regular expressions for them.



E02-10: Suppose that the lexical analyzer for the previous example always tries to do a longest match. How many characters past the end of a token might it have to examine before matching the token? Give an example where this lookahead is required.