1. (a) Assume that we have a cheap table calculator with usual operators, with usual numbers (signs, decimal fractions, but no \( \text{e}'/\text{E}' \)), without operator priorities and without parentheses. Draw the NDFA for this calculator.

(b) Give a regular expression for Pascal-style strings. (As far as I know, Pascal strings have no escape sequences, but one can use double quotes.) 'this is a pascal string with a quote (\'\') in it'.

(c) Same for C-style comments, starting with /\* and ending with */.

(d) Same for C-style comments that start with // and last till the end of the line.

(e) Give a regular expression that accepts all words over \( \Sigma = \{a, b, c\} \) that contain each of the letters in \( \Sigma \) at least once. Could you describe a similar regular expression for the complete alphabet \( \Sigma' = \{a, b, c, \ldots, z\} \)?

2. (a) Using the translation scheme on the slides, translate the following regular expression into an NDFA

\[(a|b)^*abbc.\]

(b) and also:

\[(\epsilon|aa|bb)^*c.\]