CS 415 Compilers: Problem Set 3

Spring 2022

Due date: Monday, March 7, 11:59pm

Problem 1 – Regular Expressions

You are asked to provide different definitions for identifiers for potential use in your new programing language, each with different required structures. The alphabet, i.e., finite set of symbols over which to define your "identifier language" ID is

$$\Sigma = \{\text{a, b, c, d, e, A, B, C, D, E, 0, 1, 2, 3, 4, \$, \&, \_}\} \text{ with } ID \subset \Sigma^*$$

You may use the following regular expressions for your definitions of ID:
letter_lower := (a | b | c | d | e)
letter_upper := (A | B | C | D | E)
digit := (0 | 1 | 2 | 3 | 4)
special := ($ | & | _)

Give regular expressions for ID that satisfy the following conditions. Note: You may want to define your own regular expressions to help you defince ID. All identifiers need to have at least one symbol. **You must use the regular expression syntax as presented on slide 10 in lecture 9. However, you may use the "+" operatur, i.e., positive Kleene closure**.

1. All identifiers that start with a lower-case letter and end with an upper case letter, and are not more than 6 symbols long.
2. All identifiers that do not have more than 3 digits and have exactly one special symbol.
3. All Identifiers that have an odd number of upper case letters and end with a special symbol if they start with a special symbol.

Problem 2 – DFA

Give a DFA for the following languages over the aphabet {a, b}:

1. The set of all strings with an even number of “a”s and an odd number of “b”s . The empty string is not accepted.

2. The set of all strings that do not have three consecutive “b”s and end in an “a”.

Problem 3 – NFA and DFA

1. Construct a nondeterministic finite automaton (NFA) for the regular expression $a|b(cd)^*$ using Thompson’s Construction Algorithm.

2. Convert your NFA with ϵ transitions into a DFA using the subset construction. Show your work.

3. Is the DFA minimal? If not, give the minimal DFA. Show your work.