Fabien Bessez

COMP 321: Design of Programming Languages

Homework 2: Regular Expressions

Written problems (5 points) Write regular expressions for each of the following, where you may use ∈ , concatenation, disjunction, Kleene-∗, character class notation [a – b] and fixed repetition r{n}, where if r is a regular expression, r{n} is r . . . r (n times):

Problem 1. Strings over {a, b, c} where the first a preceeds the first b (note that a string with no as or with not bs satisfies this criterion).

(a|c)*(ab(a|b|c)*)|(a|c)*

Problem 2. Strings over {a, b, c} with an even number of as.

(b|c)*(a(b|c)*a(b|c)*)*

Problem 3. Strings over {0, 1} that represent numbers divisible by 4 in binary (assume mostsignificant bit first and no leading 0s).

1(1|0)*00

Problem 4. Strings over {a, b, c} that do not contain the sequence ba.

((a|c)*|(b|c)*)(b|ca*)*

Problem 5. The language of non-negative octal and decimal integer literals in C.
These consist of:
• The digit 0;
• The octal (base-8) numerals, which start with the digit 0 followed by one or more base-8 digits, where the first such is not 0.
• The decimal (base-10) numerals, which consist of one or more base-10 digits, where the first such is not 0.

0|0[1-8][0-8]*|[1-9][0-9]*

Problem 6. The language of non-negative integers written in groups of three separated by commas as appropriate. Example words in the language are 0; 12; 762; 9, 652; and 92, 100, 542.

0|[1-9]{1,3}(,[0-9]{3})*