
HW 8 – Regular Expression

CS 421 – Fall 2014

Revision 1.0

Assigned October 21, 2014

Due October 28, 2014, 23:59

Extension 48 hours (20% penalty)

1 Change Log

1.0 Initial Release.

2 Turn-In Procedure

Answer the problem below, save your work as a PDF (either scanned if handwritten or converted from a program), add the PDF to the subversion repository (`svn add hw8-submission.pdf`) and commit it (`svn commit -m ""`). Your file should be named `hw8-submission.pdf` and committed in your `assignments/hw8` directory.

3 Objectives and Background

The purpose of this HW is to test your understanding of regular expressions and remind you of their connection to finite state automata.

4 Problems

1. (20 points) For this problem, you will be asked to write a collection of regular expressions describing given languages, and to give a non-deterministic finite state automaton accepting the same language. For all languages below, the alphabet will be

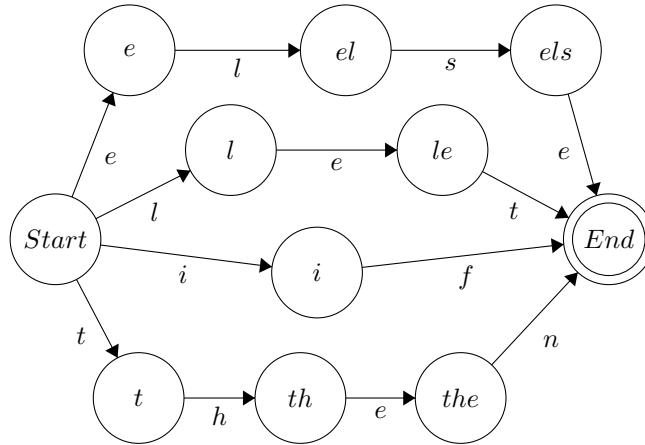
$$\Sigma = \{e, f, h, i, l, n, s, t, A, B, C, 0, 1, 8, 9, ' , . , - , \}$$

The regular expressions should be in standard mathematical notation for regular expressions as described in class, without the use of abbreviations, derived forms or syntactic sugar.

When giving a non-deterministic finite state automaton, you may label an edge with a set of characters from Σ to represent a set of edges, one for each character.

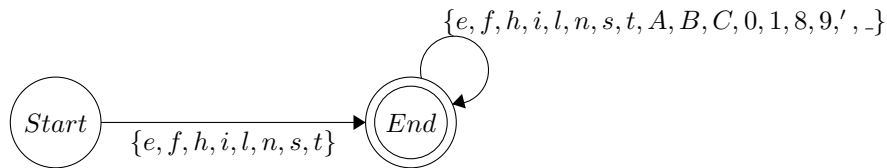
1. $L_1 = \{if, then, else, let\}$

Solution: $if \vee then \vee else \vee let$



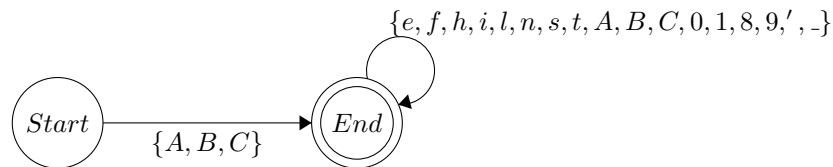
2. L_2 is the set of allowed ocaml variable names (over our reduced alphabet), where a variable name starts with a lowercase letter (in the usual sense) and then can be followed by any sequence of upper- or lowercase letters, digits, single quotes ('), or underscores (_).

Solution: $(e \vee f \vee h \vee i \vee l \vee n \vee s \vee t)(e \vee f \vee h \vee i \vee l \vee n \vee s \vee t \vee A \vee B \vee C \vee 0 \vee 1 \vee 8 \vee 9 \vee ' \vee _)*$



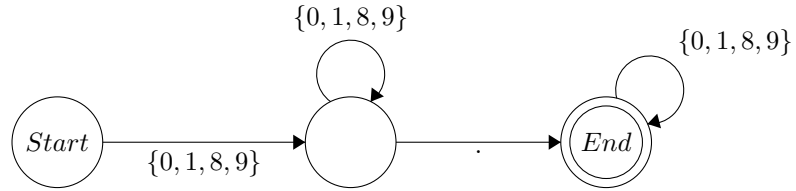
3. L_3 is the set of allowed ocaml data constructor names, where a data constructor name starts with an uppercase letter (in the usual sense) and then can be followed by any sequence of upper- or lowercase letters, digits, single quotes ('), or underscores (_).

Solution: $(A \vee B \vee C)(e \vee f \vee h \vee i \vee l \vee n \vee s \vee t \vee A \vee B \vee C \vee 0 \vee 1 \vee 8 \vee 9 \vee ' \vee _)*$



4. L_4 is the allowed set of strings representing floats in ocaml, where a float is represented by a non-empty sequence of digits, followed by a period (.), then optionally followed by a sequence of digits.

Solution: $(0 \vee 1 \vee 8 \vee 9)(0 \vee 1 \vee 8 \vee 9)^*(.) (0 \vee 1 \vee 8 \vee 9)^*$



5. L_5 is the set of representations of eight place binary numbers, with a prefix of 0B (eg 0B01110101).

Solution: $0B(0 \vee 1)(0 \vee 1)(0 \vee 1)(0 \vee 1)(0 \vee 1)(0 \vee 1)(0 \vee 1)(0 \vee 1)$

