**Homework 4: Regular Expressions and NFAs**

---

**This assignment is a solo effort, i.e., NO TEAMS.**

**Fair Warning:** Variations of Problems 1, 2, and 3 are likely to show up on quizzes. So you should practice answering such questions “by-hand”.

---

**Part I: Problems on Paper**

The languages (i.e., set of strings) considered in Problems 1, 2, and 3 are all over the alphabet \( \{ a, b \} \).

**Problem 1: (40 points).**

Use the rules on page 5 of the Lexical Analysis slides to give a formal derivation of each of the following. (Except for part (i), these are short.)

(a) \( (a|(b|c)) \downarrow a \)  
(b) \( (a|(b|c)) \downarrow b \)  
(c) \( (a|(b|c)) \downarrow c \)  
(d) \( (a(bc)) \downarrow abc \)  
(e) \( ((ab)c) \downarrow abc \)

**Definition.** \( \#_c(w) \) = the number of times character \( c \) occurs in string \( w \).

**Example:** \( \#_a(abaabba) = 4 \) and \( \#_b(abaabba) = 3 \).

**Problem 2: (16 points).**

(a) **Background.** Let \( L_1 = \{ w \in \{ a, b \}^* : (\#_a(w) \mod 3) = 0 \} \), that is, \( w \in L_1 \iff \) the number of \( a \)'s in \( w \) is a multiple of 3 (and there can be any number of \( b \)'s). A regular expression for this language is: \( b^* (ab*ab a b^* a b^*)^* \) and an NFA is \( M_1 = (\{ 0, 1, 2 \}, \text{Moves}_1, 0, \{ 0 \} ) \) where

\[
\text{Moves}_1 = \{ 0 \xrightarrow{a} 0, 0 \xrightarrow{b} 1, 1 \xrightarrow{b} 1, 1 \xrightarrow{a} 2, 2 \xrightarrow{b} 2, 2 \xrightarrow{a} 0 \}
\]

or in diagram form:

---

(b) **Background.** Let \( L_2 = \{ w \in \{ a, b \}^* : \#_a(w) \geq 2 \text{ or } \#_b(w) = 2 \} \).

A regular expression for \( L_2 \) is: \( (((a|b)*ab^*a(a|b)*) | (a^*ba^*ba^*)) \) and an NFA is \( M_2 = (\{ 1, \ldots, 6 \}, \text{Moves}_2, 1, \{ 4, 6 \} ) \) where

\[
\text{Moves}_2 = \{ 1 \xrightarrow{a} 1, 2 \xrightarrow{a} 2, 3 \xrightarrow{a} 4, 4 \xrightarrow{a} 4, 5 \xrightarrow{a} 5, 6 \xrightarrow{a} 6, 1 \xrightarrow{b} 5, 2 \xrightarrow{b} 2, 3 \xrightarrow{b} 3, 4 \xrightarrow{b} 4, 5 \xrightarrow{b} 6, 1 \xrightarrow{\varepsilon} 2, 2 \xrightarrow{a} 3 \}
\]

or in diagram form:

---

**Your Problem: (4 points)** Give an \( M_1 \)-accepting path for \( aabbaabaa \). (See pages 18 and 19 of the Lexical slides.)

---

**Your Problem: (12 points)** Give four distinct \( M_2 \)-accepting paths for \( abaab \). (See pages 18 and 19 of the Lexical slides.)
Problem 3: (16 points).

For each of the following languages over \{a, b\}, give both (i) a regular expression and (ii) a NFA that precisely captures it. (Hint: It is often easier to start with the NFA and then use the NFA to help figure out the regular expression.)

(a) Those strings that contain aaa as a substring.
(b) Those strings that fail to contain aaa as a substring.

Part II: Programming Problems

Problem 4: (12 points).

Do Problem 2.15 in Mogensen and program your answer in Haskell using Simon Thompson’s modules. Design and run some tests for your code. (Hint: You’ll need a “\|” in the equation for nonempty(st).)

Problem 5: (16 points).

BACKGROUND. On page 13 Mogensen defines the shorthands

\[ r? = \text{def} \ r|\epsilon \quad r^+ = \text{def} \ r(r^*) \]

A start at modifying Thompson’s library to handle these two new forms can be found in:

http://www.cis.syr.edu/courses/cis352/code/RegExp2/

Your Problems.

(a) In Matches2.hs the function matches does not have cases for Opt or Plus expressions. Add the missing cases to matches. Test your solution by running (quickCheck prop_equivA). Also come up with some tests of your own.

(b) In BuildNFA2.hs the function build is missing cases for Opt or Plus expressions. Add the missing cases to build. Test your solution by running (quickCheck prop_equivB). Also come up with some tests of your own.

Obvious hint for both parts (a) and (b): The Opt-case should be a variation on the Or-case and the Plus-case should be a variation on the Star-case.

Administrivia

This assignment is a solo effort, i.e., NO TEAMS.

➢ If you trade ideas with another student, document it on the cover sheet.
➢ For Part I, hand written answers are fine. However, if we can’t read it, we won’t grade it.
➢ For Part II, use Top.hs as your starter file.
➢ Let me know if any of my QuickCheck tests seem dodgy.
➢ To turn in Part I: Drop the papers in the CIS 352 bin on the 4th floor of SciTech (next to SciTech 4-226 and next to the CIS 252 box). Include a paper copy of your cover sheet.
➢ To turn in Part II: Do a Blackboard submission of (i) the files you modified from the Reg2 directory, (ii) the transcripts of test runs, and (iii) your coversheet.

Grading criteria

➢ The homework is out of 100 points.
➢ Each programming problem is, roughly, 70% correctness and 30% testing.
➢ You get 5 points taken off for omitting your name on your papers.