CS476: Automata Theory and Formal Languages

Homework 1

Assigned: 15/10/2012
Due: 2/11/2012 17.00 (Questions 1-5, hard copy)
3/11/2012 23:59 (Question 6, email attachment)

Questions

1. (20pts) State whether the following statements are true or not. You must give a BRIEF explanation or show a counter example to receive full credit.

   (a) (5pts) Any subset of a regular language is also regular.

   (b) (5pts) Let $R_1 = aa^*b$ and $R_2 = ab^+$ be two regular expressions then $R_1 \cup R_2 = a(a+b)^*b$.

   (c) (5pts) The number of states in a minimal NFA (the one with minimum number of states) is always smaller than the number of states in the minimized DFA for the same language.

   (d) (5pts) With pumping lemma, we can prove that $L = \{w : w \text{ contains } 100 \text{ as a substring}\}$ is a regular language.

2. (a) (10pts) The following table shows a non-associative operation closed under set $A = \{a, b\}$. The left-to-right and right-to-left evaluation values of a string w, when interpreted as an expression, are represented as w_{lr} and w_{rl}, respectively. As an example if $w = abb$ then $w_{lr} = (ab)b = bb = b$ and $w_{rl} = a(bb) = ab = b$. Give a DFA for all strings w such that $w_{lr} = w_{rl}$ where the first input to the DFA will be the left-most letter of the string.

	a	b
a	b	b
b	a	b

   (b) (10pts) Design an NFA with 4 states for the language $L = \{a^n \mid n \geq 0\} \cup \{b^n a \mid n \geq 1\}$.

3. (20pts) Give a regular expression for each of the following languages. Make your regular expressions as compact as possible:

   (a) (10pts) All strings that contains no more than 3 as defined over the alphabet $\{a, b, c\}$.

   (b) (10pts) $L = \{w \in \{a, b\}^* \mid n_a(w) \text{ and } n_b(w) \text{ are even}\}$, where $n_x(w)$ denotes number of times the symbol x appears in the string w.

4. (20pts) Are the following languages regular? Prove your answer.

   (a) (10pts) $L = \{w \mid w \in \{0, 1\}^* \text{ and } w \text{ is not a palindrome }\}$.

   (b) (10pts) $L = \{0^m 1^n \mid m \neq n\}$.

5. (20pts) Show that the set of regular languages are closed under the following operations.

   (a) (10pts) AllSuffixes$(L) = \{v : uv \in L \text{ for some } u \in \sum^*\}$.

   (b) (10pts) Reflect$(L) = \{u : uu^R \in L\}$.

6. (25pts) *perl* is a language which has a lot of scripting capabilities. In this exercise, you will write regular expression based find-replace scripts. Please first read the submission guidelines given at the end of the homework.

   (a) In this part, you will use the regular expression capabilities of *perl*. You will write a script such that given a file the script displays some information about the strings in the file such that

      i. The number of strings in the file that do not contain 011.

      ii. The number of strings with equal number of 0s and 1s such that no prefix contains two more 0s than 1s and two more 1s than 0s. E.g., 001 should not be counted since 00 is a prefix.

      iii. The number of strings that start with 0s and ends with 01.

      iv. The number of strings that contain at least one 0 and one 1.

   The alphabet is $\Sigma = \{0, 1\}$ hence the strings are binary strings. The strings can be separated by any number and any kind of whitespaces, i.e., tab, space, newline etc.

   (b) In a certain typesetting program used in a galaxy far far away, we have the `{\sc TEXT}` command to convert the `TEXT` between the brackets to small-caps format. As a simple example `{\sc Small-Caps    Format}` is printed as SMALL-CAPS FORMAT. You will write a script that finds the `{\sc TEXT}` patterns in a given file where the `TEXT` is formed by one or more words separated by whitespaces or `-` symbol. Note that, in the pattern `{`, `\sc`, `TEXT`, and `}` can be separated by any number and any kind of whitespaces, i.e., tab, space, newline etc. For each such pattern, you will delete the `-` symbols in `TEXT`. Your program should delete all `-` symbols at one iteration and print the output to the screen (you can do multiple iterations in your script). A simple input-output pair is given below:

```
{\sc OneWord} {\sc Pattern-Two} {\sc
This-Is-The-Third} {\sc The-4th-One
} This is Not-A-Pattern but this is {
\sc A-Pattern}. This is not-a-pattern
again. Also you need to consider
a text with whitespaces like {\sc this
is Also-A-Pattern}. But whitespaces
cannot divide the \sc command like
{\s c Not-One).
```

```
{\sc OneWord} {\sc PatternTwo} {\sc
ThisIsTheThird} {\sc The4thOne
} This is Not-A-Pattern but this is {
\sc APattern}. This is not-a-pattern
again. Also you need to consider
a text with whitespaces like {\sc this
is AlsoAPattern}. But whitespaces
cannot divide the \sc command like
{\s c Not-One).
```

Caveat: Answers for questions 1-5 should be returned in a **stapled** hard copy and the answers for question 6 should be an attached to an e-mail that will be sent to your TA (**mehmet.karahan@cs.bilkent.edu.tr**), with the subject line "**cs476hw1**". Do not send your homework to any other TA or do not use any other subject line.

Please use the following procedure to submit your homework. Note that the procedure is strict, if your file/folder names are incorrect we may ignore your submission.

1. Create a folder with name "**SurnameName**" using English letters, e.g., "YildizIbrahim". (This is the only folder you will create, do not put any other folder in it.)

2. Copy your source files "**q6part1.pl**", "**q6part2.pl**" into it. Please do not copy any input file to your folder. Your source files will need an input file and the filename must be the first argument your program reads from the command line.

3. Prepare a "**readme.txt**" which includes the commands and parameters to execute your programs. Copy it into the folder.

4. Compress the folder and create "**SurnameName.zip**".

5. Email it to the TA **with the subject line "cs476hw1**".

All of the outputs must be written to **stdout**. The deadline for submission is 5:00pm of the due date.