CS 3100 – Models of Computation – Fall 2011
This assignment is worth 10% of the total points for assignments
100 points total

October 3, 2011

Assignment 5 Solution

1. **(25 points total) – Audio Automaton!** Go to `http://www.learnmorsecode.com/`. (If you are not familiar with Morse code or need a refresher, listen to `http://www.learnmorsecode.com/atozslow.mp3` and behave as a human FSM for a while, recognizing when you hear `a` through `z`.) Answer the following questions.em

   (a) **(2 points)** Suppose you are charged with building a finite-automaton that recognizes any sequence of *spacer-terminated letters*. (Note that I wrote my name above where each letter was followed by a spacer.) Then, would you prefer to build an NFA or a DFA, and why? (One sentence.)

   Answer: Let `IF` stand for a state name that is initial and final, and `F0, F1,..` be state names for final states. Let a state name `I` stand for an initial (but not final) state. States not starting with an `I` or an `F` are non-initial and non-final.

   Now let me show you how to draw the transitions for `E` and `S`.

   IF we allowed empty sequences, **THEN** it really does not matter: we can go as follows.

```
IF - 0 -> dot1
IF - 1 -> "decode similarly" (I'm just going to pursue . and ... in this illustration)

dot1 - 0 -> dot2
dot1 - 2 -> IF    ; going to accept an E

dot1 - 0 -> dot3
dot3 - 0 -> IF    ; going to accept an S
```

   BUT if we are not going to accept empty sequences of blank-terminated-letters, then a DFA like this won't do; instead, we would do something like this:

```
I - 0 -> dot1
I - 1 -> "decode similarly" (I'm just going to pursue . and ... in this illustration)

dot1 - 0 -> dot2
dot1 - 2 -> F   ; going to accept an E , but now transition to a new state F ..
                ; that will serve as a "pseudo initial" state now on
```

```
dot1 - 0 -> dot3
dot3 - 0 -> F   ; going to accept an S , and transition to F


F - 0 -> dot1
F - 1 -> "decode similarly" (I'm just going to pursue . and ... in this illustration)
```

In other words, we will be transitioning to a state `F` and from then only revisit `F` (never come back to I).

(b) **(15 points)** Let 0 model "Dit" (or `.`), 1 model "Dah" (or `-`), and 2 model the spacer between two letters. Draw, using JFLAP, a *FA corresponding to the Morse-code decoding tree in `http://www.learnmorsecode.com/`.
Please restrict your *FA to just A through Z. Do not encode anything else.
ANSWER: Finish drawing as above. I'll provide some more of this DFA below. I'm going to choose an NFA. I'll also illustrate A-E for you. This is the NFA description:

```
I - 0 -> S0
I - 1 -> S1

S0 - 2 -> F      ; recognize an E
S0 - 1 -> S01
S01 - 2 -> F     ; recognize an A

S1 - 0 -> S10

S10 - 0 -> S100
S10 - 1 -> S101

S100 - 0 -> S1000
S100 - 2 -> F     ; recognize D

S101 - 0 -> S1010

S1010 - 2 -> F    ; recognize C

S1000 - 2 -> F    ; recognize B

F - epsilon -> I
```

(c) **(1 point)** What is the alphabet of this *FA?
ANSWER: { 0, 1, 2}

(d) **(2 points)** What is the language of the above *FA, if you just considered A-E (we don't want to put through writing it for A-Z, so A-E will suffice)? Think carefully and write it down as a **regular expression.**

ANSWER: I'll use "A" to stand for `01` (note that "A" itself does not include the spacer). Then the language is `(A2 + B2 + C2 + D2 + E2)+`. Note that the regular expression `R+` is the same as `RR*`.

2. **(35 points total) – Help USPS Avoid Chapter-11!**

   (a) Let $L_1 = \{0^{3n} \mid n \geq 0\}$, and let each string s_1 in L_1 represent a stamp of value equal to the length of s_1.

   (b) Let $L_2 = \{0^{5n} \mid n \geq 0\}$, and let each string s_2 in L_2 represent a stamp of value equal to the length of s_2.

   (c) **(2 points)** Write L_1 as the Kleene-star of a language called l_1. What is l_1?
   ANSWER: $l_1 = \{000\}$
   Write l_1 down as a mathematical set.

   (d) **(1 points)** Similarly write down l_2 corresponding to L_2.
   ANSWER: $l_1 = \{00000\}$

   (e) **(6 points)** Now define $L_3 = ((L_1)^* \ (L_2)^*)^*$. Explain the contents of L_3 in one English sentence, and also draw an NFA for L_3 using JFLAP.
   ANSWER: Apply the star construction to the above language. This is all strings formable by concatenating blocks of three 0 and blocks of five 0.

   (f) **(4 points)** The contractor claims that L_3 models all possible postage rates people might ever want to create (assuming that stamps are micro-miniaturized so that you can put a lot of them on an envelope). This means that each string in L_3 has as many 0s as the postage value one may want to affix to an envelope. Argue that L_3 has all strings of length $\geq$ some $X \geq 0$. What is the smallest value of X, and why?
   ANSWER: X = 8. The contractor is wrong. Strings of length 1, 2, 4, and 7 can't be realized.

   (g) **(12 points)** Show that the contractor is lying, by drawing an NFA for all stamp values that *cannot* be created using only 3-cent and 5-cent stamps (*e.g.*, if a 4-cent stamp cannot be created, your NFA must accept a string of four 0s). Assume of course that you **can** create a 0-cent stamp by forgetting to put any stamp at all! Present this NFA as a JFLAP drawing. We want the NFAs to be not bloated (a few extra states are OK – nothing in excess).
   ANSWER: Draw an NFA that accepts sequences of 0s of lengths 1, 2, 4, and 7.

   (h) **(5 points)** Draw, using JFLAP, a DFA for all stamp denominations that *can* be made. *E.g.*, if 6-cents can be made, the DFA must accept a string of length 6.
   ANSWER: Draw a DFA that accepts sequences of 0s of lengths 0, 3, 5, 6, 8, and all higher than 8.

   (i) **(5 points)** Draw, using JFLAP, a DFA for all stamp denominations that *cannot* be made. *E.g.*, if 4-cents cannot be made, the DFA must accept a string of length 4. How do these two DFAs in parts 2h and 2i relate to each other? Write down a sentence describing the DFA complementation procedure.
   ANSWER: Complement the above DFA.

3. **(40 points) – Help Leika Launch Successfully!** Leika dog is awaiting space launch, and your job as the designer of the launch-control finite automaton is to keep Leika safe. Everything is going well if the automaton keeps receiving `Y`s in Morse-code (`Y=1011`). (*No need to consider a spacer, for simplicity.*) Unfortunately, the launch controller gets so much static that you must be willing to tolerate up to two bit errors (if you crash the launch controller upon the first error, Leika may be in danger). This means:

(a) If you get `YYY..Y?YY..Y?YYY...(only Ys now)` where `?` is a four-bit sequence with a one-bit error somewhere, the automaton must accept. For example the first `?` may be `1111` and the second `?` may be `0011`. Thus, we can bave one `Y` corrupted and then another, but that's all!

(b) If you get `YYY..Y!YYY...(only Ys now)` where `!` is a four-bit sequence with a two-bit error somewhere, the automaton must accept. For example the `!` may be `0111`, but after that, you must have only `Y`s.

(c) The best case of course is `YYY...(only Ys here)`, the automaton must accept.

Write down a regular expression encompassing Case 3c, Case 3b, and Case 3a above.

(a) There must be one sub-regular expression that deals with Case 3c.

(b) There must be one sub-regular expression that deals with Case 3b.

(c) There must be one sub-regular expression that deals with Case 3a.

(d) The whole regular expression (RE) must be obtained by using the RE union operation on the above three sub-REs.

(e) The repetition inside each sub-RE must be achieved using the RE Kleene-star operation. Thus, for example, to repeat `Y`, you must Kleene-star the `1011` pattern.

(f) Even the different cases of errors must be accomplished using the RE union construction.

ANSWER: I'll use `Y` as an abbreviation as introduced above. I'll further use `?0`, `?1`, `?2`, and `?3` for the four 1-bit error cases, and `!` to represent the single two-bit error case. Then the desired RE is

```
Y* + Y* (?0+?1+?2+?3) Y* (?0+?1+?2+?3) Y* + Y* !  Y*
```

4. **(20 EXTRA points – due same time as the rest):** Draw a DFA for the above language. Explain your construction approach.

   ANSWER: Will draw in class.