# CS 3100 - Models of Computation - Fall 2011 This assignment is worth $10 \%$ of the total points for assignments 100 points total 

September 25, 2011

## Assignment 5, Posted on: 9/22 9/29 Thursday 11:59pm

Guidelines:

- I have provided a section What to Submit at the end. Use that as a guideline.
- PLEASE don't handin tarballs. From now, you must handin exactly the files requested. If we don't file names matching the names we require, you will get $\mathbf{0}$ points for that question.
- Please look at the class webpage for assignment names you must use for late handins. For example, for asg5, use name asg5L1 for one-day late, asg5L4 for upto 4 days late, and asg5L5 for $>4$ days late (this set won't be graded except to perhaps push you up a notch if warranted, toward the end of the semester-entirely at my discretion, of course).
- All questions involve the use of only JFLAP - no Python-based coding necessary. All testing is to be conducted using JFLAP's step or suitable other commands. Your submissions may be single screen-shots or combined screen-shots; but please separate the submissions for each question. No tarballs please, as said above.
- Here are the file naming conventions (using the first question for detailed illustration): Submit all your answers in file AudioAutomaton.txt, numbering each answer suitably. Submit all your screen-shots in a single file AudioAutomaton.pdf or if you have multiple screen-shots then AudioAutomaton_1.pdf, AudioAutomaton_2.pdf, etc.
- For the USPS question, name files USPS.txt, USPS.pdf, etc.
- For the Leika space launch question, name the files SpaceLaunch.txt, SpaceLaunch.pdf, etc.

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) (5 points) Write your full name down in English, and then in Morse-code. Use blanks in lieu of spacers between letters. Thus, you should write dot-dash combinations for each letter of your name and put blanks separating those letters.
A spacer symbol is the longer-than-dash duration you hear following each letter.
Example: My first-name is what is written between the double-quotes: "--. .- -. . ... .... "
(b) (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.)
Call the finite automaton $* F A$ for later parts of this question.
(c) (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.
When this *FA is fed any sequence of spacer-terminated letters, it must accept. For example, it must accept A , A B , A B Z , etc.
(d) (1 point) What is the alphabet of this *FA?
(e) (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.
2. (35 points total) - Help USPS Avoid Chapter-11! USPS is going to do severe cost-cutting! No, they are not investing in carrier pigeon breeding or teaching people to transmit smoke signals. Their approach is this: instead of printing all kinds of stamp denominations, they've decided to print only 3cent and 5 -cent stamps! They claim though that barring postages below 6 cents, nobody needs to worry. (Of course you should never believe everything you hear.)
Unfortunately, their solution comes from an external contractor who has such a low opinion of US's automata-theoretic knowledge that they've decided to pull a fast one and not state things in plain English. Here is how the scheme is broached:
(a) Let $L_{1}=\left\{0^{3 n} \mid n \geq 0\right\}$, and let each string $s_{1}$ in $L_{1}$ represent a stamp of value equal to the length of $s_{1}$.
(b) Let $L_{2}=\left\{0^{5 n} \mid n \geq 0\right\}$, 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 $l_{1}$ ? Write $l_{1}$ down as a mathematical set.
(d) (1 points) Similarly write down $l_{2}$ corresponding to $L_{2}$.
(e) (6 points) Now define $L_{3}=\left(\left(L_{1}\right)^{*}\left(L_{2}\right)^{*}\right)^{*}$. Explain the contents of $L_{3}$ in one English sentence, and also draw an NFA for $L_{3}$ using JFLAP.
(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 0 s 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?
(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).
(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 .
(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 2 h and 2 i relate to each other? Write down a sentence describing the DFA complementation procedure.
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 in Morse-code ( $\mathrm{Y}=1011$ ). (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 Ys.
(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.
4. (20 EXTRA points - due same time as the rest): Draw a DFA for the above language. Explain your construction approach.

## What to Submit

These correspond to question numbers in the previous section. The thing in "quotes" below is what your answer file must contain.

1. (25 points total) - Audio Automaton!
(a) (5 points) '1a: Your name in dots and dashes." in AudioAutomaton.txt
(b) (2 points) '1b; NFA or DFA? Why (one sentence)." in AudioAutomaton.txt
(c) (15 points) "1c: JFLAP drawing" in AudioAutomaton 1 c . pdf
(d) (1 point) '1d; Alphabet" in AudioAutomaton.txt
(e) (2 point) '1e; RegExp" in AudioAutomaton.txt
2. (35 points total) - Help USPS Avoid Chapter-11!
(a) (2 points) '2a: $l_{1}$ as a set" in USPS.txt
(b) (2 points) '2b; $l_{2}$ as a set" in USPS.txt
(c) (6 points) '2c: $L_{3}$ NFA in JFLAP" in USPS_nfa_whole2c. pdf. English sentence in same JFLAP drawing.
(d) (4 points) '2d; Smallest $X$ and why" in USPS.txt
(e) (12 points) '2e; NFA of impossible strings in JFLAP" in USPS_nfa_impossible 2 e . pdf
(f) (5 points) '2f: DFA of possible strings in JFLAP" in USPS_dfa_possible $2 f$.pdf
(g) (5 points) '2g; DFA of impossible strings in JFLAP" in USPS_dfa_impossible 2 g . pdf
3. (40 points) - Help Leika Launch Successfully!
(a) (40 points) '3a; Single regular expression and clear explanation" in Leika.txt
4. (20 EXTRA points - due same time as the rest: Draw a DFA for the above language. Explain your construction approach.
(a) (20 points) "4a: Single JFLAP drawing, explanation in drawing" in DFALeike 4 a. pdf
