

- (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 ***FA** for later parts of this question.

- (c) **(15 points)** Let 0 model “Dit” (or \cdot), 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 3-cent 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 $L_1 = \{0^{3n} \mid n \geq 0\}$, 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 ? 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 $L_3 = ((L_1)^* (L_2)^*)^*$. 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 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?
- (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 2h and 2i 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 Ys 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 $YYY..Y?YY..Y?YYY\dots$ (**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 have one Y corrupted and then another, but that's all!
- (b) If you get $YYY..Y!YYY\dots$ (**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 $YYY\dots$ (**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 AudioAutomaton1c.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_impossible2e.pdf
 - (f) (5 points) “2f: DFA of possible strings in JFLAP” in USPS_dfa_possible2f.pdf
 - (g) (5 points) “2g: DFA of impossible strings in JFLAP” in USPS_dfa_impossible2g.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 DFALeika4a.pdf