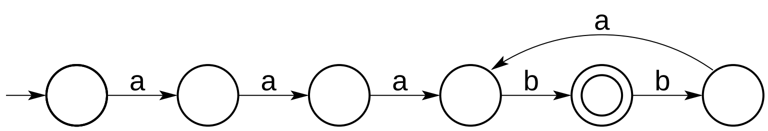
Do problem 0 and any three of problems 1-5.
If you attempt more than three of problems 1-5, please indicate which ones you want graded – otherwise, I'll make an arbitrary choice.

Graded on a scale of 100 points.
You can attempt from 89 to 110 points depending on which problems you choose. If you score over 100, you get to keep the extra credit.

0. (**5 points**) Your name: *Mark Greenstreet* Your student #: $1.602178480 \times 10^{-19}$

1. (**24 points**)

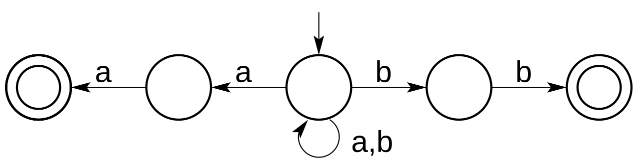
   (a) (**8** points) Let A_1 be the language recognized by the NFA below:



   Write a regular expression that generates A_1.

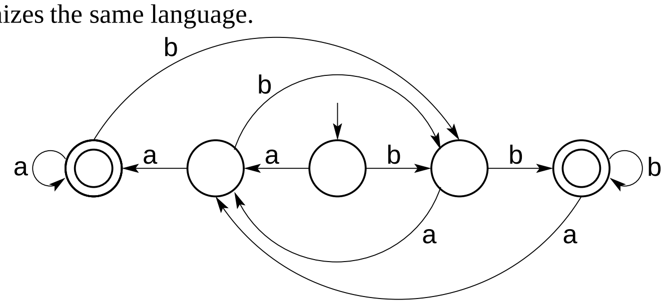
   Solution: $\mathtt{aaab}(\mathtt{bab})^*$

   (b) (**8** points) Let A_2 be the language recognized by the NFA below:



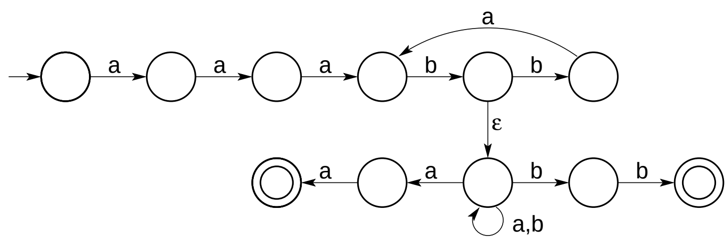
   Draw the state diagram for a DFA that recognizes A_2.

   Solution: The NFA recognizes all strings that end with two consecutives a's or two consecutive b's. The DFA below recognizes the same language.

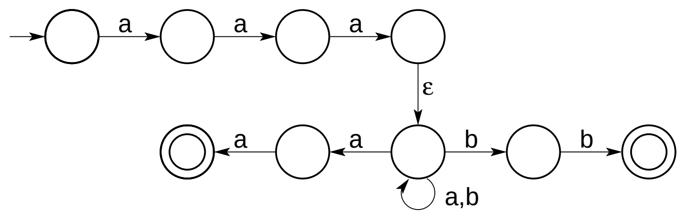


(c) (**8** points) Draw the state diagram for a NFA that recognizes $A_1 \cdot A_2$.

Solution 1: Just draw an ϵ-edge from the final state of the NFA for A_1 to the start state of the NFA for A_2:



Solution 2: Looking at the solution 1, we can see that the $(\mathsf{bab})^*$ loop at the accepting state of the NFA for A_1 is subsumed by the $(\mathsf{a} \cup \mathsf{b})^*$ loop at the start state of the NFA for A_2. Thus, we can drop the states and edges for the $(\mathsf{bab})^*$ loop without changing the language recognized by the NFA:



2. (**30 points**) Let $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$, and let $B = \{w \mid \exists i, j.\ w = \mathsf{a}^i \mathsf{b}^j \mathsf{c}^{i+j}\}$. Prove that B is not regular.

Solution 1:

Let p be a proposed pumping lemma constant for B.
Let $w = \mathsf{a}^p \mathsf{b}^p \mathsf{c}^{2p} \in B$.
Let x, y and z be three strings such that $xyz = w$, $|xy| \leq p$ and $|y| > 1$. Note that $xy \in L(\mathsf{a}^*)$.
Thus, $xy^0z = \mathsf{a}^{p-|y|}\mathsf{b}^p\mathsf{c}^{2p}$ which is not in B because $(p - |y|) + p = 2p - |y| \neq 2p$.

Solution 2:

Let $B' = B \cap \mathsf{a}^*\mathsf{c}^* = \mathsf{a}^n\mathsf{c}^n$. Because the regular languages are closed under intersection and $\mathsf{a}^*\mathsf{c}^*$ is regular, B' would be regular if B were regular. It was shown in class (and in *Sipser*) that $\mathsf{a}^n\mathsf{b}^n$ (and thus $\mathsf{a}^n\mathsf{c}^n$) is not regular; therefore B is not regular either.

3. (**30 points**) Let C_1 be a language with alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}\}$. Let

$$C_2 \quad = \quad \{w \in \Sigma^* \mid \mathsf{a}^{|w|} \in C_1\}$$

Show that if C_1 is regular, then C_2 is regular as well.
It is sufficient, for example, to describe how to construct an NFA for C_2 given a NFA or DFA (you choose) for C_1. You don't have to give all of the formal details, just describe enough that it is clear that you could write the formulas if you had sufficient time. For example, my solution consists of four English sentences.

Solution 1:

Consider the state diagram for a DFA that recognizes C_1. Discard all edges except for those labeled a. Replace the edges labeled a with edges labeled Σ. This produces a NFA that recognizes C_2.

Solution 2:

Let $N = (Q, \Sigma, \delta, q_0, F)$ be a (Sipser-style) NFA that recognizes $C_1 \cap \mathsf{a}^*$. For all $q \in Q$ and $c \in \Sigma$ define $\delta'(q, c) = \delta(q, \mathsf{a})$. Then, $N' = (Q, \Sigma, \delta', q_0, F)$ is a NFA that recognizes C_2.

Explanatory comment (not needed to get full credit): N' works by replacing every edge labeled with a with an edge labeled with Σ. Thus, N' accepts any string of length n iff N accepts a^n.

4. (**35 points**) Let D_1 be a language with alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}\}$. Let

$$D_2 \quad = \quad \{w \in \Sigma^* \mid \exists x \in D_1.\ x = \mathsf{a}^n \mathsf{b}^n \text{ with } n = |w|\}$$

Show that if D_1 is regular, then D_2 is regular as well.
It is sufficient to describe how to construct an NFA for D_2 given a NFA or DFA for D_1. You don't have to give all of the formal details, just describe enough that it is clear that you could write the formulas if you had sufficient time.

Solution (long): Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA that recognizes language D_1 with $Q = \{q_0, q_1, \ldots q_{m-1}\}$ where m is the number of states of M. I'll now construct languages $C_0 \ldots C_{m-1}$ that are each regular, and whose union is D_2. Let $M_{a,i} = (Q, \Sigma, \delta, q_0, \{q_i\})$. Clearly, $L(M_{a,i})$ is regular, and by question 3, the language

$$A_i \quad = \quad \{w \mid a^{|w|} \in L(M_{a,i})\}$$

is regular as well. Note that A_i is the set of all strings w such that machine M reaches state q_i after reading $\mathsf{a}^{|w|}$.

Likewise, we can let $M_{b,i} = (Q, \Sigma, \delta, q_i, F)$ and

$$B_i \quad = \quad \{w \mid b^{|w|} \in L(M_{b,i})\}$$

is regular – it is the set of all strings w such that M can reach a state in F by starting in state q_i and reading $\mathsf{b}^{|w|}$.

Let $C_i = A_i \cap B_i$. C_i is regular because A_i and B_i are regular and the regular languages are closed under intersection. Language C_i is the set of all strings w such that M can start in state q_0, read $\mathsf{a}^{|w|}$ to reach state q_i, then read $\mathsf{b}^{|w|}$ to reach a state in F. Thus, if $w \in L(C_i)$ then $w \in D_2$. Finally, note that if $w \in D_2$, then there is some state q_i such that $\delta(q_0, \mathsf{a}^{|w|}) = q_i$ and $\delta(q_i, \mathsf{b}^{|w|}) \in F$. Thus, $w \in C_i$. We conclude that

$$D_2 \quad = \quad \bigcup_{i=0}^{m-1} C_i$$

and D_2 is regular because the regular languages are closed under union.

Solution (short): Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA that recognizes language D_1 with $Q = \{q_0, q_1, \ldots q_{m-1}\}$. Let A_i be the set of all strings w, such that machine M reaches state q_i after reading $\mathsf{a}^{|w|}$. Let B_i be the set of all strings w such that machine M reaches a state in F after starting in state q_i and reading $\mathsf{b}^{|w|}$. A_i and B_i are both regular by the solution to question 3. String w is in D_2 iff there is some state q_i such that $\delta(q_0, \mathsf{a}^{|w|}) = q_i$ and $\delta(q_i, \mathsf{b}^{|w|}) \in F$. Thus,

$$D_2 \quad = \quad \bigcup_{i=0}^{m-1} A_i \cap B_i$$

and D_2 is regular because the regular languages are closed under intersection and union.

Solution (alternate): Construct two NFAs: the first NFA starts in state q_0 and moves forward on a-edges of M. The second NFA starts in any state in F and moves backward along b-edges of M. A string is in D_2 iff these two NFAs can reach the same state after $|w|$ moves. This is basically building a product machine that recognizes D_2. We've never discussed product NFA in class (only product DFAs), but a solution that notes that extrapolation will get full credit.

5. (**40 points**) Let $\Sigma = \{\texttt{0}, \texttt{1}\}$, and let E_1 and E_2 be the languages defined below:

$$\begin{aligned} E_1 &= \{w \mid \exists k \in \mathbb{Z}, x \in \Sigma^*. \ (|w| = 10k) \wedge (|x| = k) \wedge (w = x^{10})\} \\ E_2 &= \{w \mid \exists k \in \mathbb{Z}, x \in \Sigma^*. \ (|w| = 10k) \wedge (|x| = 10) \wedge (w = x^{k})\} \end{aligned}$$

One of these languages is regular and the other is not. Identify which language is which.

(a) (**2**0 points) Describe a DFA, NFA or regular expression for the language that is regular – you **don't** need to draw a complete state diagram or write the entire expression; just describe how to construct it.

Solution: E_2 is regular.

Because $|x| = 10$ and $|\Sigma| = 2$, there are 2^{10} possible values for x in the definition of E_2. Let α_i be a regular expresion that matches the i^{th} such string. For example, we could define:

$$\begin{array}{rcl} \alpha_0 &=& \texttt{0000000000} \\ \alpha_1 &=& \texttt{0000000001} \\ \alpha_2 &=& \texttt{0000000010} \\ \alpha_3 &=& \texttt{0000000011} \\ \alpha_4 &=& \texttt{0000000100} \\ \vdots & \vdots & \vdots \\ \alpha_{1023} &=& \texttt{1111111111} \end{array}$$

Now, note that E_2 is generated by the regular expression

$$\beta = \bigcup_{i=0}^{1023} \alpha_i^*$$

This is a finite union. Thus, β is a regular expression, and therefore E_2 is regular.

(b) (**2**0 points) For the other language, prove that it is not regular.

Solution: E_1 is not regular.

Let p be a proposed pumping lemma constant for E_1.

Let $w = (\texttt{0}^p\texttt{1})^{10} \in E_1$.

Let x, y and z be any three strings with $xyz = w$, $|xy| \le p$ and $|y| \ge 1$.

Note that xy is a prefix of the first p $\texttt{0}$'s of w.

Thus, $xy^2z = \texttt{0}^{p+|y|}\texttt{1}(\texttt{0}^p\texttt{1})^9$ which is not in E_1. Therefore, E_1 is not regular.