York University
CSE 2001 Fall 2017 – Assignment 2 of 4
Instructor: Jeff Edmonds

Family Name: __________ Given Name: __________

Student #: __________ Email: __________

Family Name: __________ Given Name: __________

Student #: __________ Email: __________

1) Program to DFA	$12 + 12$	
2) Build NFA	$0 + 6 + 6 + 12$	
3) NFA to DFA	$12 + 12 + 12$	
4) NFA into a Reg	16	
0) Art	2	
Total	102 marks	

This exam is designed to be completed in an hour.
Keep your answers short and clear.

0) (2 marks) Art therapy question: When half done the exam, draw a picture of how you are feeling.

1. **Program to DFA:**
   Note in binary if $x = 101_2 = 5$ and $y = 1011_2 = 11$ then $y = 2 \cdot x + 1$.
   Remember $x \; mod \; 3 = 2$ is the remainder when you divide x by 3.
   Consider the following program:

   ```
   q = 0
   loop until no more characters
         get(c)                      % c ∈ {0,1}
         q = (2 · q + c) mod 3       % q ∈ {0,1,2}
   end loop
   if q = 0 then
         return("accept")
   else
         return("reject")
   end if
   ```

   (a) Describe this language in one easy to understand English sentence.
   Hint: Look at examples in slides.

   (b) Convert the program into a DFA.

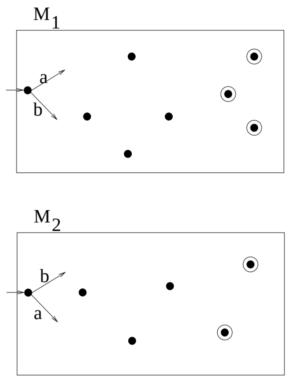
   (c) Convert the DFA into a regular expression.

2. **NFA:**
   Let L_1 and L_2 be arbitrary languages and let M_1 and M_2 below be diagrams representing the NFA for them.

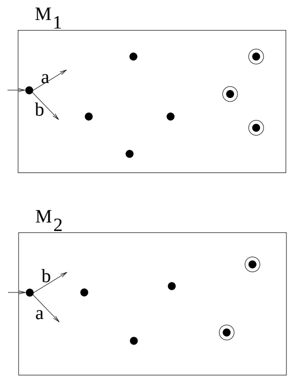
   (a) Explain (as if to a 1030 student) the key differences between the languages $L_1^* \cup L_2^*$ and $(L_1 \cup L_2)^*$. Give an example of a string that is in one but not in the other and vice versa.

   - Answer in Notes: Let $L_1 = \{a\}$ and $L_1 = \{b\}$. $(L_1^* \cup L_2^*)$ contains strings that either only contain a's or only contain b's. On the other hand, $(L_1 \cup L_2)^*$ contains strings that contains only a's and b's. Let $\omega = ab$. It is in the second and but not the first. Everything in the first is in the second.

(b) Draw an NFA for the language $L_1^* \cup L_2^*$ for this generic L_1 and L_2.

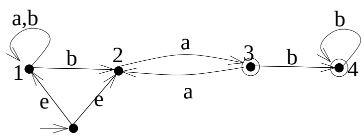


(c) Draw an NFA for the language $(L_1 \cup L_2)^*$.



(d) Explain (as if to a 1030 student) the key differences between the structures of your $L_1^* \cup L_2^*$ and $(L_1 \cup L_2)^*$ NFAs and why these differences cause the difference in the languages accepted. Hint: Describe how clones can travel through the machines. Use the word "commit".

3. Consider the following NFA.
Here e is the empty string.



(a) Explain in words what language is accepted by this NFA.
Hint: Break the string α into non-empty blocks of a's and b's. What is the requirement on the lengths of these blocks?

(b) Formally prove that this NFA computes your stated language as follows:
We want to prove that $\forall \alpha \in \{a,b\}^*,\ M(\alpha) = L(\alpha)$.
Proof: Let α be an arbitrary string in $\{a,b\}^*$.
We prove $M(\alpha) = L(\alpha)$ using two cases:

$L(\alpha) = yes$	iff	$\alpha \in L$
	iff	α has stated property
	iff	α has decomposition property
	$\Rightarrow_1 \Leftarrow_2$	$\exists$ constructed path labeled α
	$\Rightarrow_3 \Leftarrow_4$	$\exists$ accepted path labeled α
	iff	$M(\alpha) = yes$

i. Given that the string $\alpha \in L$ has the property stated in the definition L, what *decomposition property* does it have?

Hint: Decompose the string into three substrings.
What properties does each of these substrings have?

ii. Define what *constructed paths* through NFA M look like.
Hint: Deconstruct the path into three stages.
What are the key land marks between these stages?

iii. For each of the four implications $\Rightarrow_1$, $\Leftarrow_2$, $\Rightarrow_3$, $\Leftarrow_4$ given above, state the property required of your contructed paths and prove that they have it.

(c) Do not do any long conversion. Write an extended regular expression that expresses the same language.

(d) Without doing the conversion, design a DFA for this language. Label the states with meaningful names.
Hint: The loop invariant states that what is remembered about the prefix read so far is:
- whether we are working on a block of a's or a block of b's.
- whether the last block of a's has even or odd length.
This implies there are four states.
You don't need to, but my DFA collapses two of these states into one.

(e) Do the steps with the table to convert this NFA into a DFA.

4. Do one step of converting this NFA into a regular expression by ripping out state 2.