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

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:

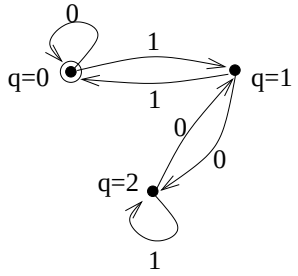
```
q = 0
loop until no more characters
    get(c)                    % c ∈ {0,1}
    q = (2 · q + c) mod 3     % q ∈ {0,1,2}
end loop
if q = 0 then
    return("accept")
else
    return("reject")
end if
```

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

   - Answer: $L = \{\alpha \in \{0,1\}^* \mid x_\alpha$ is divisible by three, where x_α is the integer obtained when thinking of α in binary. $\}$.

   (b) Convert the program into a DFA.

   - Answer:



   (c) Convert the DFA into a regular expression.

   - Answer: Add an ϵ transition from a new start state to $q=0$ and another from the accept state $q=0$ to the new accept state.
   Ripping out state $q=2$ gives a self loop on state $q=1$ labeled 01^*0.
   Ripping out state $q=1$ gives a self loop on state $q=0$ labeled $0 \cup 1(01^*0)^*1$.
   Ripping out state $q=0$ gives the final answer $[0 \cup 1(01^*0)^*1]^*$.

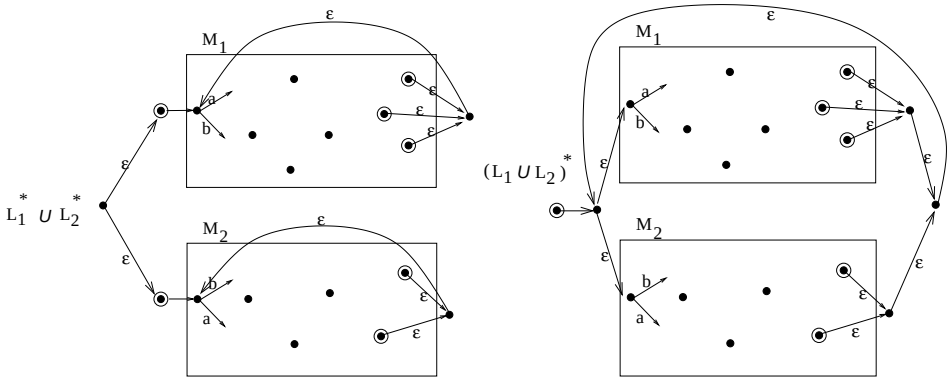
2. **NFA:**
Let L_1 and L_2 be arbitrary languages and let M_1 and M_2 below be diagrams representing the NFA for them.

   (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.
(Do not do it for simply $L_1 = \{a\}$ and $L_1 = \{b\}$.)
(c) Draw an NFA for the language $(L_1 \cup L_2)^*$.

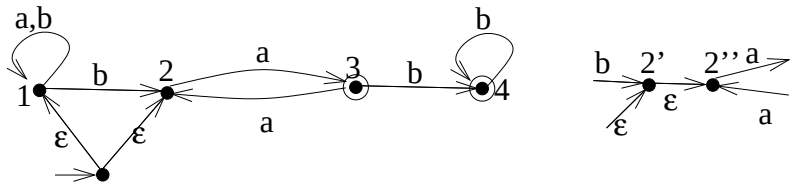
- Answer:



(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".

- Answer: In both machines there is a state that has one epsilon edge to the start of one M_1 and another to that of M_2. Any clone on this state must choose which machine to enter. However, in the machine for $L_1^* \cup L_2^*$ this is a bigger commitment than it is in the machine for $(L_1 \cup L_2)^*$. In the second, the clone can later switch to the other machine; however, in the first, the clone must stick forever with the machine first chosen.
  In both machines there are epsilon edges from all the accept states of M_1 and of M_2 back the beginning so that the machine can be traversed again. (In my figure, to make the figure easier, I had these edges collect together before going back. This is not necessary.) In the machine for $L_1^* \cup L_2^*$, these go back to the start state of the same M_1 or M_2 so that the same one must be traversed again. In contrast, in the machine for $(L_1 \cup L_2)^*$, these edges go back before the choice between M_1 and M_2 so that the clone again has a choice of whether to enter M_1 or M_2.
  In the L^* NFA construction, the start state must be an accept state so that the NFA accepts the empty string. However, we don't want to make the first state of M_1 an accept state because that might change the workings of this machine. To accomplish this, the construction adds a new start state.

3. Consider the following NFA.



Our goal is to explain in words the language $L(M)$ accepted by this NFA
and then to prove by loop invariants (induction) that $L(M) = L$.

Goal: Prove $L(M) = L$ by proving $\forall q_i, M(q_i) = L(q_i)$.

The Machine and its Labeled Paths: $M(q_i) = \{\alpha$ with path to state $q_i\}$.
i.e. it is just the definition of $L(M_i)$ for the NFA M_i where q_i is the only accept state.

The Language and Properties of its Strings: $L(q_i) = \{\alpha$ with some property$\}$.

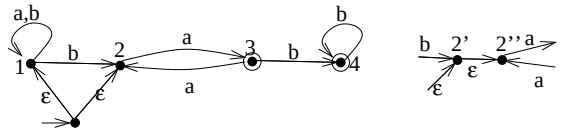
Suppose the DFA machine has read in the substring "abbaaa" so far. What do you want it to be remembering about this substring? What does Pooh write on his black board? What is the common property of the strings (including this one) that you want to arrive at this state? We will denote the set of strings with this property as $L(q_3)$. Effectively, what is the "meaningful name" you are giving to this state?

(a) For each state q_i, we *guessed* what that set $M(q_i)$ would be with a *Name* $L(q_i)$.
Hint: Split state q_2 as done into two states q_2' and q_2'' and note that this does not change the language.
Hint: Also define the languages $L(q_2''$ to $q_2'')$ and $L(q_2''$ to $q_3)$ in which the start state is q_2'' and accept state is either q_2'' or q_3.

- Answer:
  - $L(q_0)$ = "empty string".
  - $L(q_1)$ = "any string".
  - $L(q_2')$ = "ends in a b or is empty".
  - $L(q_2''$ to $q_2'')$ = a block of a's of (zero or) even length".
  - $L(q_2''$ to $q_3)$ = a block of a's of odd length".
  - $L(q_2)$ = "ends in a block of a's of (zero or) even length".
  - $L(q_3)$ = "ends in a block of a's of odd length".
  - $L(q_4)$ = "ends in a non-zero block of b's and the previous block of a's has odd length".

(b) Define the language $L(M)$ accepted by machine M in terms of these state languages $M(q_i)$.
In doing so, determine in words what language is accepted by this NFA.

- Answer:
  $L(M) = \bigcup_{accept\ states\ q_i} L(q_i)$
  $= L(q_3) \cup L(q_4)$
  = "ends in a block of a's of odd length"
  $\cup$ "ends in a non-zero block of b's and the previous block of a's has odd length"
  = "The last non-empty block of a's (it might have b's after it) must have odd length."



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

- Answer: $(\{a,b\}^* b \cup \epsilon)\ \ a(aa)^*\ b^*$

(d) Our goal is to prove that $\forall i, M(q_i) = L(q_i)$.
Our loop invariant after having read t characters will be
LI_t = "$\forall i, \forall \alpha$ of length t, $\alpha \in M(q_i)$ iff $\alpha \in L(q_i)$".
Assume LI_t. Consider an arbitrary string αc with length $t{+}1$.
You must prove $\alpha c \in M(q_2)$ iff $\alpha c \in L(q_2)$

- Answer:
  $\alpha c \in M(q_2)$ (by def^n of $M(q_2)$)
  iff αc has path of length $t{+}1$ to q_2 (by edges in M into q_2)
  iff α has path of length t (to q_1 and $c = b$) or (to q_3 and $c = a$) (by def^n of $M(q_i)$)
  iff ($\alpha \in M(q_1)$ and $c = b$) or ($\alpha \in M(q_3)$ and $c = a$) (by LI$_t$)
  iff ($\alpha \in L(q_1)$ and $c = b$) or ($\alpha \in L(q_3)$ and $c = a$) (by def^n of $L(q_i)$)
  iff (α is "any string" and $c = b$) or (α "ends in a block of a's of odd length" and $c = a$)
  iff αc is "any string ending in a b" or "ends in a non-zero block of a's of even length"
  iff αc "ends in a zero block of a's" or "ends in a non-zero block of a's of even length"
  iff αc "ends in a block of a's of (zero or) even length" (by def^n of $L(q_2)$)
  iff $\alpha c \in L(q_2)$

(e) State the general *required connections* that must be true about these guessed sets $L(q_i)$?
For each state, state the required connection with respect to $L(q_i)$.
Argue that this condition is in fact true.

- Answer:
The general required connections on $L(q_i)$ are
$\forall i, L(q_i) = \bigcup_{\langle q_j, q_i, c\rangle} L(q_j) \cdot c$.

q_0: Consider state q_0. It does not have any actual edges into it. But you can sort of imagine that being the start state is kind of like having an edge into it labeled ϵ from the universe. The required condition is that
$L(q_1) = L(q_{universe}) \cdot \epsilon \cup \bigcup_{no\ edges} = \{\epsilon\}$.

q_1: Consider state q_1 with an edge labeled ϵ from q_0 and an edge labeled a and b from q_1. The required condition is that
$L(q_1) = L(q_0) \cdot \epsilon \cup L(q_1) \cdot \{a, b\}$.
Proof that this is true:

$$\begin{aligned} L(q_1) &= \text{“any string”} \\ &= \{\epsilon\} \cup \text{“any string”} \cdot \{a, b\} \\ &= L(q_0) \cdot \epsilon \cup L(q_1) \cdot \{a, b\}. \end{aligned}$$

q_2: Consider state q_2 with an edge labeled ϵ from q_0, an edge labeled b from q_1, and an edge labeled a from q_3. The required condition is that
$L(q_1) = L(q_0) \cdot \epsilon \cup L(q_1) \cdot b \cup L(q_3) \cdot a$.
Proof that this is true:

$$\begin{aligned} L(q_1) &= \text{“ends in a block of } a\text{'s of (zero or) even length”} \\ &= \text{“does not end in } a\text{”} \cup \text{“ends in a nonzero block of } a\text{'s of even length”} \\ &= (\text{“empty string”} \cup \text{“any string”} \cdot b) \cup \text{“ends in a block of } a\text{'s of odd length”} \cdot a \\ &= L(q_0) \cdot \epsilon \cup L(q_1) \cdot b \cup L(q_3) \cdot a \end{aligned}$$

q_3: Consider state q_3 with an edge labeled a from q_2. The required condition is that
$L(q_1) = L(q_2) \cdot a$.
Proof that this is true:

$$\begin{aligned} L(q_1) &= \text{“ends in a block of } a\text{'s of odd length”} \\ &= \text{“ends in a block of } a\text{'s of (zero or) even length”} \cdot a \\ &= L(q_2) \cdot a \end{aligned}$$

q_4: Consider state q_4 with an edge labeled b from q_3 and from q_4. The required condition is that
$L(q_4) = L(q_3) \cdot b \cup L(q_4) \cdot b$.
Proof that this is true:

$$\begin{aligned} L(q_4) &= \text{“ends in a non-zero block of } b\text{'s and the previous block of } a\text{'s has odd length”} \\ &= \text{“ends in a block of } a\text{'s of odd length”} \cdot b \\ &\quad \cup \text{“ends in a non-zero block of } b\text{'s and the previous block of } a\text{'s has odd length”} \cdot b \\ &= L(q_3) \cdot b \cup L(q_4) \cdot b \end{aligned}$$

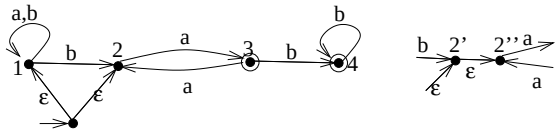
(f) Use loop invariants (induction) and these verified required conditions to prove the machine M computes the stated language, i.e. $M(L) = L$.

- Answer:
  i. **Loop Invariant:**
  Our goal is to prove that $\forall i, M(q_i) = L(q_i)$.
  Our loop invariant after having read t characters will be
  $LI_t =$ “$\forall i, \forall \alpha$ of length t, $\alpha \in M(q_i)$ iff $\alpha \in L(q_i)$”.
  ii. **Establishing the Loop Invariant:** Prove LI_0.
  Namely, we must prove that “$\forall i M(q_i) = L(q_i)$ is correct for the empty string ϵ.

For the start state q_0, both $M(q_0)$ and $L(q_0)$ contain the empty string ϵ.
But for the other states, they do not.

iii. **Maintain the Loop Invariant:** Assume LI_t and prove LI_{t+1}
Namely, we must prove that "$\forall i, \forall \alpha c$ of length $t+1$, $\alpha c \in M(q_i)$ iff $\alpha c \in L(q_i)$".
To do this, consider some state q_i and an arbitrary string αc with length $t+1$.
You must prove $\alpha c \in M(q_i)$ iff $\alpha c \in L(q_i)$
$\alpha c \in M(q_i)$
iff αc has path of length $t+1$ to q_i
iff α has path of length t to q_j for some edge $\langle q_j, q_i, c\rangle$ (from q_j to q_i labeled c).
(by LI_t) α has path to q_j iff $\alpha \in L(q_j)$.
Unioning these possibilities gives
iff $\alpha \in \bigcup_{\langle q_j, q_i, c\rangle} L(q_j)$.
iff $\alpha c \in \bigcup_{\langle q_j, q_i, c\rangle} L(q_j) \cdot c = L(q_i)$.

iv. **Obtaining the Post Condition:**
Define the language $L(M)$ accepted by machine M in terms of these state languages $M(q_i)$. In so doing prove that $L(M) = L$.
Done above.



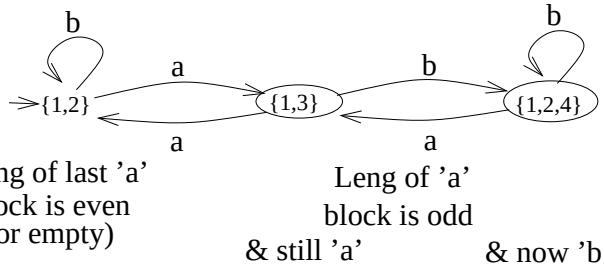
(g) Without doing the conversion, design a DFA for this language. Label the states with meaningful names.
Hint: The loop invariant states that what is remembered about the prefix read so far is:
- whether we are working on a block of a's or a block of b's.
- whether the last block of a's has even or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

- Answer: See answer below.

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

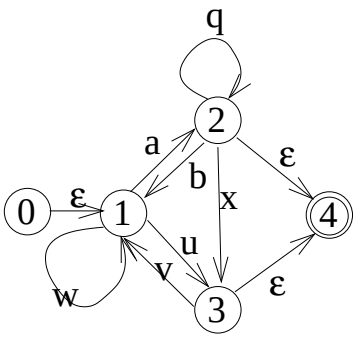
- Answer:

	a	b
1	{1}	{1,2}
2	{3}	{}
3	{2}	{4}
4	{}	{4}



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

Answer:



Rip 2