CS 205: Introduction to Discrete Structures I Homework 4

Sami Kamal, Sam Vedi

April 2023

Problem 1. [9 points]

Question:

A palindrome is a string that reads the same backward as it does forward. For example,*abaaaba* is a palindrome. Suppose that we need to define a language that generates palindromes.

Part (a):

Define a context-free grammar that generates the set of all palindromes over the alphabet $\{a, b\}$ clearly describing the recursive rules that generates palindromes. Use the notation Symbol $\rightarrow$ rule. The empty set is denoted by λ.

Solution:

$S \rightarrow \lambda|a|b|aSa|bSb$

Part (b):

Show the set of terminal symbols and the set of non-terminal symbols in your grammar.

Solution:

Terminal symbols = $\{\lambda, a, b\}$

Non-terminal symbols = $\{S\}$

Part (c):

Show that the palindrome *abaaaba* can be recognized by your grammar. To show this, show all steps of parsing the expression *abaaaba* using the rules you defined above.

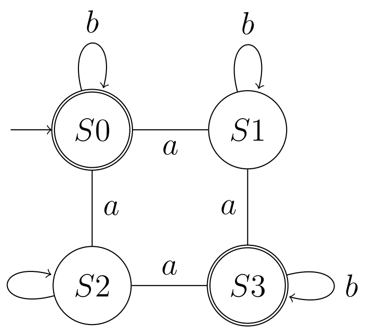
Solution:

$$\begin{aligned}
S &\rightarrow aSa && \text{(Use rule } S \rightarrow aSa\text{)}\\
S &\rightarrow abSba && \text{(Use rule } S \rightarrow b\text{)}\\
S &\rightarrow abaSaba && \text{(Use rule } S \rightarrow aSa\text{)}\\
S &\rightarrow abaaaba && \text{(Use rule } S \rightarrow a\text{)}
\end{aligned}$$

Problem 2. [15 points]

Question:

Consider the following FSA over the alphabet $\{a, b\}$ with starting state $s0$. The accept states are marked with double circles.



Part (a):

Describe the language recognized by this FSA?

Solution:

When b is in a self loop, the final state has been reached. When ab^*a, b^*, and $(ab^*a)^*$ final state has been reached too, we end up with the following language for this FSA:

$$(b^*(ab^*a)^*)^*$$

Part (b):

Show the transition table for this FSA.

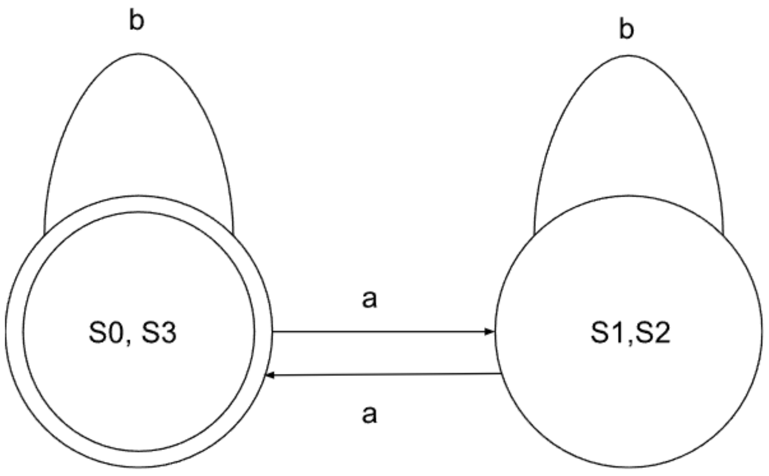
Solution:

Transition	a	b
$S0$	$S1$	$S0$
$S1$	$S3$	$S1$
$S2$	$S0$	$S2$
$S3$	$S2$	$S3$

Part (c):

Build an equivalent FSA with reduced states.

Solution:



Part (d):

State how one may prove that the FSA in (c) is indeed equivalent to the one given in the question.

Solution:

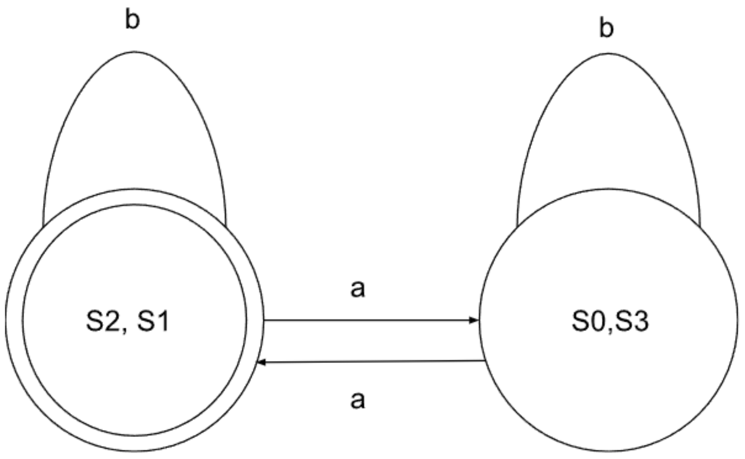
The way to prove the FSA in (c) is if the language is the same.

Part (e):

What language is recognized by the complement of the reduced FSA?

Solution:

If we swap the initial state and the final state from the reduced FSA from part (c), it will give us the complement:



Only the name of the states were interchanged, therefore the FSA is still the same so the language recognized by the complement is the same as part (a) which was $b^*(ab^*a)^*)^*$.

Problem 3. [9 points]

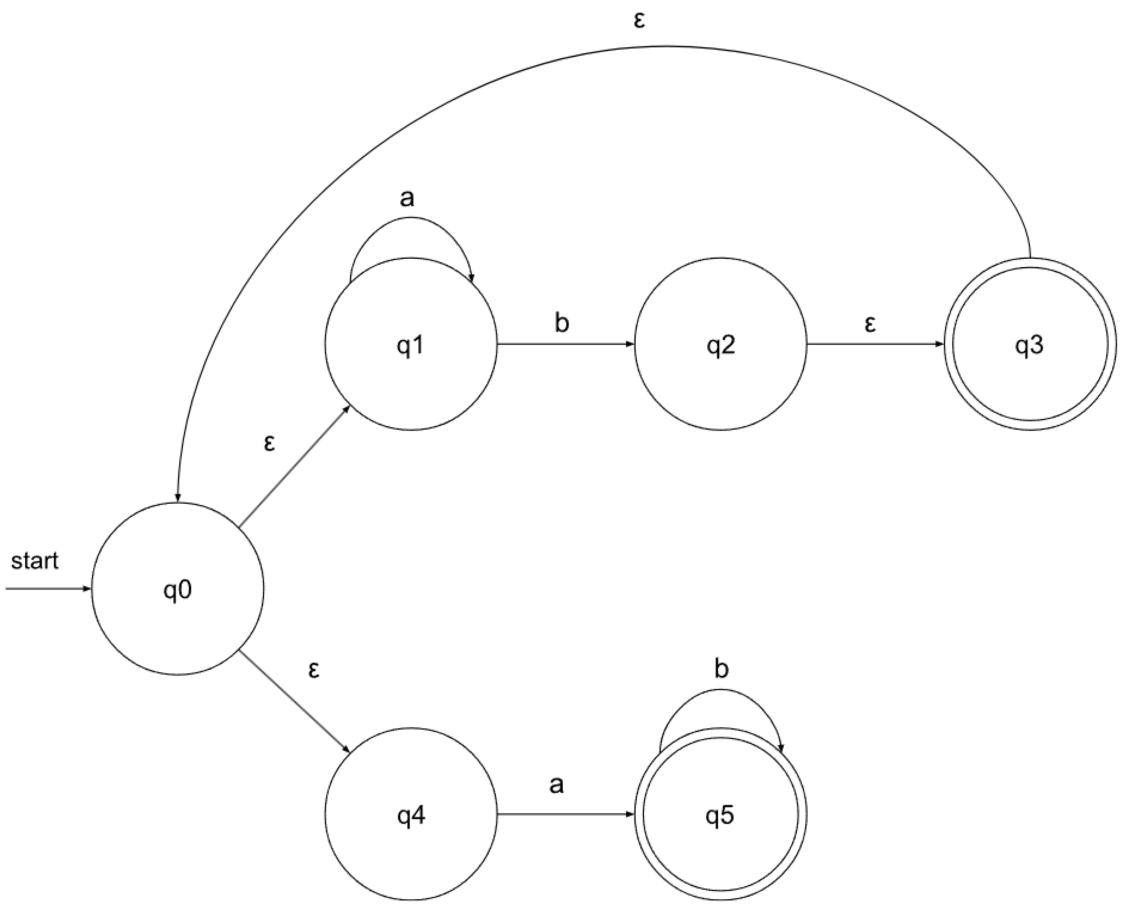
Question:

Recall that any Regular Expression can be simulated by an NFA.

Part (a):

Construct the NFA for the RegEx $((a * b) * |ab*)$.

Solution:



Part (b):

Show that the expression *aaaa* will NOT be accepted by this machine. To show this, simulate the machine after reading each symbol, including the empty symbol.

Solution:

Steps when simulating the string *aaaa*:

$$\begin{aligned}
\text{Initial State} &= q0 && \text{(Remaining input: } aaaa\text{)}\\
\text{Input} &= \epsilon && \\
\text{State} &= q1 && \text{(Remaining input: } aaaa\text{)}\\
\text{Input} &= a && \\
\text{State} &= q1 && \text{(Remaining input: } aaa\text{)}\\
\text{Input} &= a && \\
\text{State} &= q1 && \text{(Remaining input: } aa\text{)}\\
\text{Input} &= a && \\
\text{State} &= q1 && \text{(Remaining input: } a\text{)}\\
\text{Input} &= a && \\
\text{State} &= q1 && \text{(Input not accepted)}
\end{aligned}$$

Since state $q1$ is not an accepting state and there is no remaining input left, the string *aaaa* has not been accepted by the NFA.

Part (c):

Show that the expression *abab* will be accepted by this machine. To show this, simulate the machine after reading each symbol, including the empty symbol.

Solution:

Steps when simulating the string *abab*:

Initial State = $q0$	(Remaining input: $abab$)
Input = ϵ	
State = $q1$	(Remaining input: $abab$)
Input = a	
State = $q1$	(Remaining input: bab)
Input = b	
State = $q2$	(Remaining input: ab)
Input = ϵ	
State = $q3$	(Remaining input: ab)
Input = ϵ	
State = $q0$	(Remaining input: ab)
Input = ϵ	
State = $q1$	(Remaining input: ab)
Input = a	
State = $q1$	(Remaining input: b)
Input = b	
State = $q2$	(Remaining input: ϵ (empty string))
Input = ϵ	
State = $q3$	(Input accepted)

Since state $q3$ is an accepting state and there is no remaining input left, the string $abab$ has been accepted by the NFA.

Problem 4. [9 points]

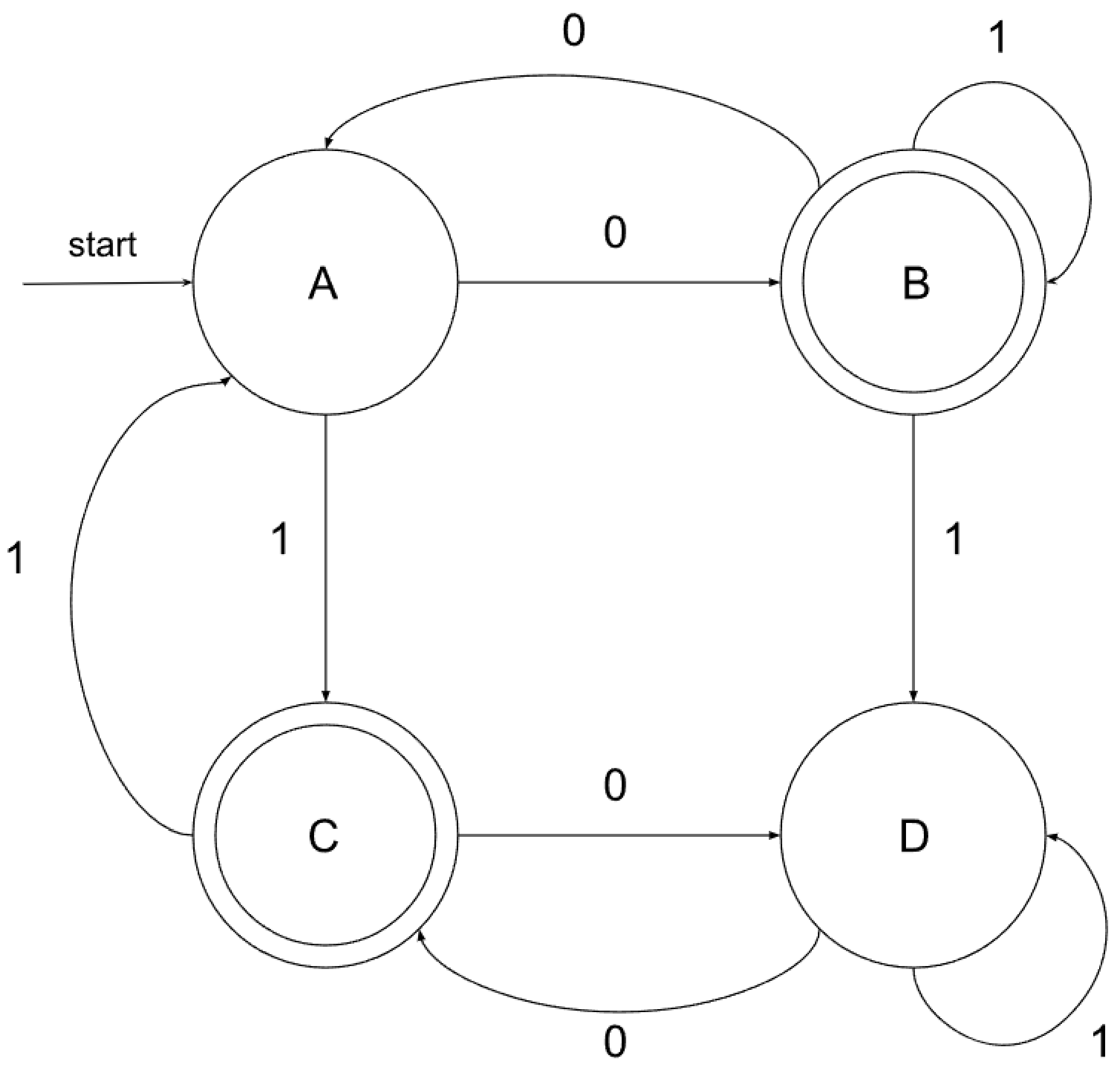
Question:

Consider an NFA to recognize the set of all binary strings that have either odd number of 0s', or that the number of 1s' is not a multiple of 3, but not both.

Part (a):

Draw the NFA with clear markings of all states including start and acceptance state(s).

Solution:



Part (b):

Simulate the NFA to show that string 010011 will be accepted by the NFA.

Solution:

Steps when simulating 010011:

$$
\begin{aligned}
\text{Initial State} &= A && \text{(Remaining input: 010011)}\\
\text{Input} &= 0 &&\\
\text{State} &= B && \text{(Remaining input: 10011)}\\
\text{Input} &= 1 &&\\
\text{State} &= B && \text{(Remaining input: 0011)}\\
\text{Input} &= 0 &&\\
\text{State} &= A && \text{(Remaining input: 011)}\\
\text{Input} &= 0 &&\\
\text{State} &= B && \text{(Remaining input: 11)}\\
\text{Input} &= 1 &&\\
\text{State} &= B && \text{(Remaining input: 1)}\\
\text{Input} &= 1 &&\\
\text{State} &= B && \text{(Input accepted)}
\end{aligned}
$$

Since state B is an accepting state and there is no remaining input left, the string 010011 has been accepted by the NFA.

Part (c):

Simulate the NFA to show that string 01010 will not be accepted by the NFA.

Solution:

Steps when simulating 01010:

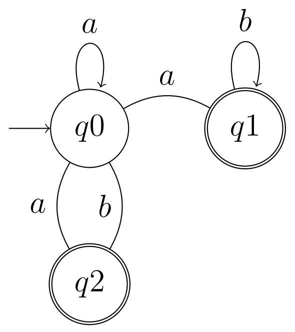
$$\begin{aligned}
\text{Initial State} &= A && \text{(Remaining input: 01010)}\\
\text{Input} &= 0 &&\\
\text{State} &= B && \text{(Remaining input: 1010)}\\
\text{Input} &= 1 &&\\
\text{State} &= B && \text{(Remaining input: 010)}\\
\text{Input} &= 0 &&\\
\text{State} &= A && \text{(Remaining input: 10)}\\
\text{Input} &= 1 &&\\
\text{State} &= C && \text{(Remaining input: 0)}\\
\text{Input} &= 0 &&\\
\text{State} &= D && \text{(Input not accepted)}
\end{aligned}$$

Since state D is not an accepting state and there is no remaining input left, the string 01010 has not been accepted by the NFA.

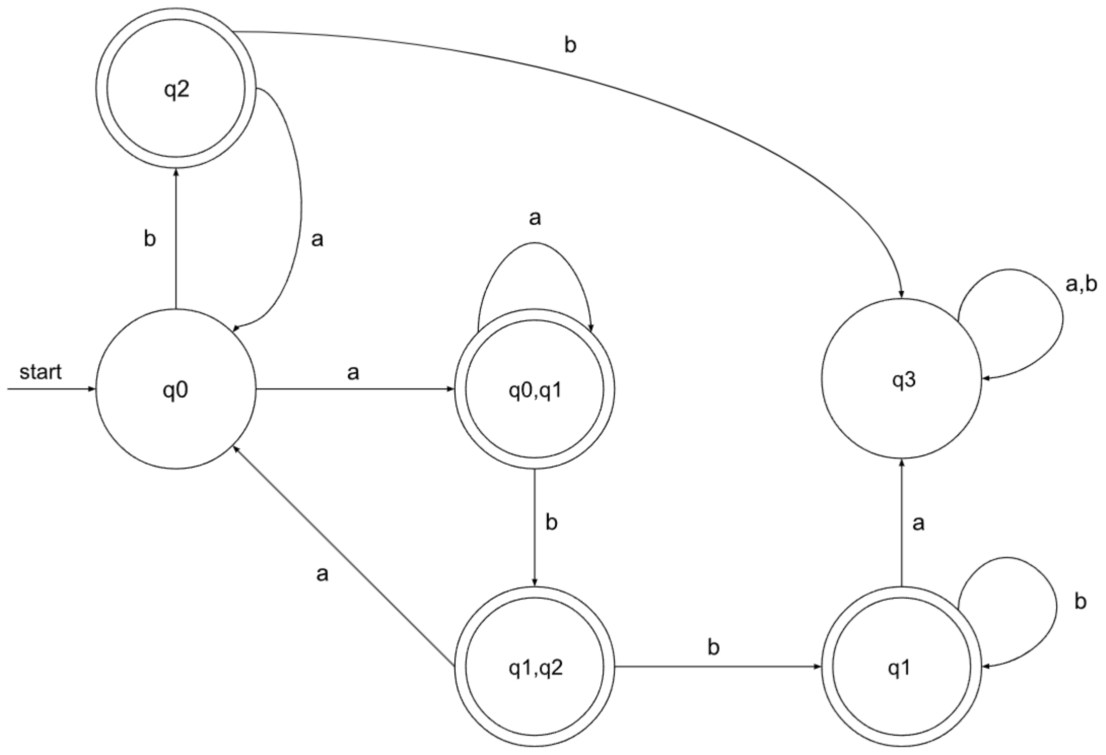
Problem 5. [4 points]

Question:

Find a DFA equivalent to the following NFA.



Solution:



Problem 6. [5 points]

Question:

Construct a Turing machine that recognizes the set of all bit strings that contain at least two 1s'. You need to define the state transition table in the format used in lectures. That is, $f(s_i, a) = (s_j, A, R)$ where s_i is the current state, s_j is the next state, a is the input symbol, A is the output symbol written on tape, and L indicates moving left or R indicates moving right (and $\perp$ indicates not moving).

Solution:

$Q : \{q1, q2, q3, q4\}$

$\Sigma : \{0, 1\}$

$\Gamma : \{0, 1, B\}$

$s : \{q1\}$

$F : \{q4\}$

State Transition Table:

$$f(q1, 0) = (q1, 0, R)$$

$$f(q1, 1) = (q2, 1, R)$$

$$f(q2, 0) = (q2, 0, R)$$

$$f(q2, 1) = (q3, 1, R)$$

$$f(q3, 0) = (q3, 0, R)$$

$$f(q3, 1) = (q3, 1, R)$$

$$f(q3, B) = (q4, B, L)$$

Problem 7.[12 points]

Question:

Recall that $a|b$(a divides b), if there exists an integer c, such that $b = a \cdot c$.

Part (a):

If $a|b$ and $a|c$, show that $a^2|(b \cdot c)$.

Solution:

There exists integers x and y where $b = a \cdot x$ and $c = a \cdot y$. To show $a^2|(b \cdot c)$, we can say there exists an integer z, where $b \cdot c = a^2 \cdot z$.

$$\begin{aligned} b \cdot c &= (a \cdot x)(a \cdot y) && \text{(substitute } b = a \cdot x \text{ and } c = a \cdot y\text{)} \\ &= a^2 \cdot xy \end{aligned}$$

Because xy is an integer, this tells us that a^2 divides $b \cdot c$ and says $a^2|(b \cdot c)$, thus proved.

Part (b):

Let a be an integer and d is a positive integer. Then there are unique integers q and r such that $0 \leq r < d$ such that $a = dq + r$. Given $a = -11, d = 3$, find q and r. Show your work.

Solution:

Since the equation above is a division algorithm we can do the following:

$\frac{-11}{3} = -3$ with a remainder of 1

This shows us that -11 can be written as 3 times some integer q with a remainder of 1 and so we can write the following:

$-11 = 3(-3) + 1$

Based of this equation this shows us that $q = -3$ and $r = 1$ which matches the conditions of division algorithm.

Part (c):

Write at least 3 positive integers and 3 negative integers in the equivalence class [12], given $m = 7$.

Solution:

Finding 3 positive integers:

We can compute $7 + 12 = 19$, then compute $19 + 7 = 26$, and finally compute $26 + 7 = 33$. So our three positive integers in the equivalence class [12] are $19, 27, 33$.

Finding 3 negative integers:

We can compute $12 - 7 = 5$, then compute $5 - 7 = -2$, and finally compute $-2 - 7 = -9$. So our three negative integers in the equivalence class [12] are $5, -2, -9$

Part (d):

Find 12^8 *mod* 7 using equivalence classes. Show work to receive credit.

Solution:

In order to use equivalence classes, we need to find what the remainder is for $\frac{12^8}{7}$, so we can look at the remainder for each power of 12 when we divide it by 7. We also know that 12 is congruent to 5 *mod* 7 since $12 - 7 = 5$, so we can do the following:

$$\begin{aligned}
12^1 &= 5^1 = 5 \, mod \, 7 \\
12^2 &= 5^2 = 4 \, mod \, 7 \\
12^3 &= 5^3 = 6 \, mod \, 7 \\
12^4 &= 5^4 = 2 \, mod \, 7 \\
12^5 &= 5^5 = 3 \, mod \, 7 \\
12^6 &= 5^7 = 1 \, mod \, 7
\end{aligned}$$

We can see that $12^6 = 1 \, mod \, 7$ which tells us that any power of 12 multiplied by 6 is congruent to $1 \, mod \, 7$. Since 8 is not a multiple of 6, we can use this to reduce the exponent and do the following:

$$\begin{aligned}
12^8 &= 12^6 \cdot 12^2 \\
&= 1 \cdot 4 \\
&= 4 \, mod \, 7
\end{aligned}$$

This shows that 12^8 is congruent to $4 \, mod \, 7$ and that the remainder if you compute $\frac{12^8}{7}$ is 4.

Problem 8. [12 points]

Question:

Recall the RSA encryption/decryption system. The following questions are based on RSA. Suppose $n(= 15)$ is the product of the two prime numbers 3 and 5.

Part (a):

Find an encryption key e for for the pair (e, n).

Solution:

We will choose $p = 3$ and $q = 5$, so $n = 15$ and compute the following:

$$\phi = (p - 1)(q - 1) = 2 \cdot 4 = 8$$

In order to find e for the pair (e, n), the product of the keys must be equal to $1 mod \phi$. We can calculate the following values of $(1 \; mod \; \phi)$:

$$9, 17, 25, 33, 41, 49, 57, 65...$$

We will choose the smallest number, which is 9 and $9 = 3 \cdot 3 = e \cdot d$. So $e = 3$ and the encryption key pair for (e, n) is $(3, 15)$.

Part (b):

Find a decryption key d for for the pair (d, n).

Solution:

To calculate d we need $de = 1 \; mod \; \phi$, so we can choose $d = 3$ because $33 = 1 \; mod \; 8$, so the decryption key pair (d, n) is $(3, 15)$.

Part (c):

Given the plaintext message $x = 3$, find the ciphertext $y = x^e$ (where x^e is the message x encoded with encryption key e).

Solution:

To find the ciphertext we will do the following:

$$\begin{aligned} y &= 3^3 \; mod \; 15 \\ &= 27 \; mod \; 15 \\ &= 12 \end{aligned}$$

Hence $y = 12$ is the ciphertext.

Part (d):

Given the ciphertext message y (which you found in previous part), Show that the original message $x = 3$ can be recovered using (d, n).

Solution:

Since $y = 12$, we need show we can revert it to our original message by doing the following:

$$\begin{aligned} x &= (12)^3 \, mod \, 15 \\ &= 1728 \, mod \, 15 \\ &= 3 \end{aligned}$$

As shown we were able to get $x = 3$ and recover the original message.