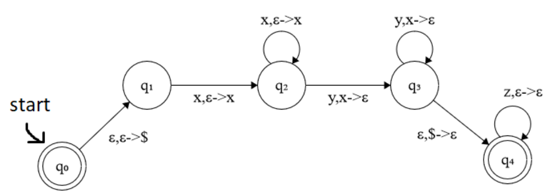
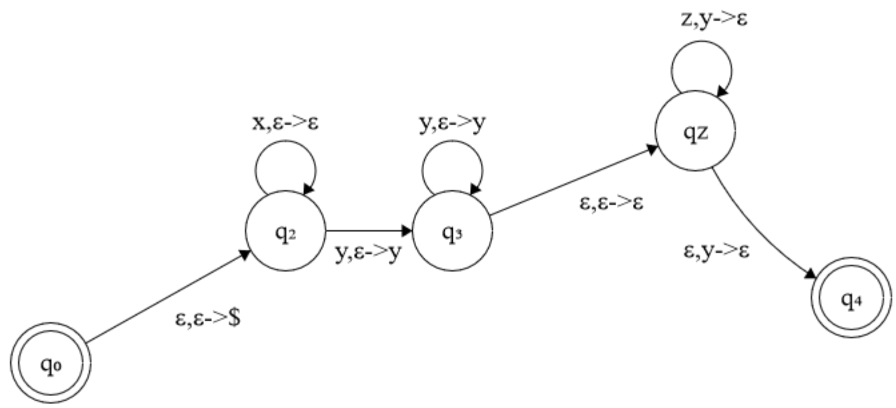
(1) A language is a CFL if a PDA can be constructed. The PDA for the language is the union of PDAs, which must be a CFL, for $p = q, q < r$, or both. We first start by reading a string of x's and putting it on the stack. We do the same with y's elimnating x's to ensure equivalence. When we reach the end of the stack, we'll write however many z's. This is the case where $p = q$. For $q < r$, We can go through x and y regularly but then when we are done writing z's to the stack, only accept when we read a y after writing a string of z's. This ensures the number of y's is less than the number of z's. See figures below:





note: $q_0$is the initial state for the second PDA.s

(2) A wants to cover two conditions. One where p is greater than q and q is greater than or equal to zero. The CFG equivalent of this would be appending a x every time and y is seen. In the CFG we'd have to terminate a string of character in the language. That is, when we need to terminate the variable that generates the languages conditions for x and y as well as a way to generate a string of $z's$. The CFG for A can be seen below:

$$S \rightarrow XZ$$

$$X \rightarrow xXy|xX|x$$

$$Z \rightarrow zZ|\epsilon$$

To handle cases for $q > r \geq 0$, let's use a production with the variable Y and use a different X variable to generate the x's. The CFG B:

$$S \rightarrow XY$$

$$X \rightarrow xX|\epsilon$$

$$Y \to yYz|yY|y$$

The intersection of A and B would be the combination of the conditions for p, q, r: $p > q > r \geq 0 \therefore L = \{x^p y^q z^r | p > q > r \geq 0\}$. Let us use the pumping lemma to show that this intersection is irregular:

Let $p \geq 1, w = x^{p+2} y^{p+1} z^p$ where $|w| \geq p$ and could abstracted to $w = uvxyz, |vy| > 0, |vxy| \leq p$. This last condition ensures only two distinct characters to be adjacent. Since v or y must be non-empty, let's consider the case where vxy is adjacent to just one symbol x. Trying to pump the word, we can see: $\tilde{w} = uv^0xy^0z$. However, this string results in $|x^{p+2}| \leq |y^{p+1}| \Rightarrow p+2 \leq p+1 = 2 \leq 1$ which is false. If vxy is adjacent to two symbols. We'd get a $\tilde{w}$ resulting in the omission of some of $x's$ and $y's$ which violates $|vy| > 0$.

(b) The union of A and B would generate the appropriate language $\bar{X} = \{x^p y^q z^r | p \leq q \leq r\}$ The pumping lemma can be used to establish its regularity: Let $p \geq 1, w = x^{p+2} y^{p+1} z^p$ where $|w| \geq p$ and could abstracted to $w = uvxyz, |vy| > 0, |vxy| \leq p$. This last condition ensures only two distinct characters. Similar to part (a), pumping the string when vxy are neighbors to only one symbol we'll see for the string $\tilde{w} = uv^0xy^0z$ that the number of y characters decreases by atleast 1. In the other case of two adjacent symbols, we'll see a decrease in y and z characters which violates the languages condition of $q \leq r$. This means that $\bar{X}$ cannot be a CFL as it contradicts the pumping lemma.

(3)

$L = \{a^m b^n | m = n^2, m, n \geq 1\}$

$L = \{a^{n^2} b^n | n \geq 1\}$

Let $p \geq 1$ where $|w| \geq p$ and could abstracted to $w = uvxyz, |vy| > 0, |vxy| \leq p$

Assume L is context free:

Consider we have:

$\tilde{w} = uv^0xy^0z$

Case 1: vxy is adjacent to one symbol, consider an a character. Pumping $\tilde{w}$ we'd remove at least one a characters which result in the language be modified to $L = \{a^{m-k} b^n | k > 0\}$. However $m = n^2$ and $n^2 - k \neq n$ for any $k > 0$. Therefore modifying how many a's there are without changing the $b's$ would ensure $\tilde{w} \notin L$.

Case 2: vxy is adjacent to two symbols, consider an a, b characters. In the uv^ixy^iz composition, if we apply i=0, this means a, b are removed in a way that preserves $m = n^2$. This is not possible since the only way to preserve that relationship would be decreasing a's more than p however $|vxy|$ is strictly less than p.

Consider this when n=2:

$uv^0xy^0z \to uxz \to aab$ or a^2b. Here, $m = 2$ and $n = 1$. However this contradicts the $L's$ constraint $m = n^2$since $2 \neq 1^2$.

Considering in both cases$\tilde{w} \notin L \therefore L$ is not regular.

(4) (a) $q_0 0011, xq_1 011, x0q_1 11, xq_2 0y1, q_2 x0y1, xq_0 0y1, xxq_1 y1,$

$xxyq_1 1, xxq_2yy, xq_2xyy, xxq_0yy, xxyq_3y, xxyyq_3, xxyq_4y$(accept).
(b)$q_0 00011, xq_1 0011, x0q_1 011, x00q_1 11, x0q_2 0y1, xq_2 00y1, q_2 x00y1,$
$xq_0 00y1, xxq_1 0y1, xx0q_1 y1, xx0yq_1 1,$
$xx0q_2yy, xxq_2 0yy, xq_2 x0yy, xxq_0 0yy, xxxq_1yy, xxxyq_1y, xxxyyq_1$(accept).

(5) TM starts at the initial state and looks for a 0 replacing it with an x if it is unable to find one. This repeats if there are multiple zeroes on tape. If it finds a 1 moving left replacing it with a y. If 0 or y is read switch to head to right. If looking for an x, the head moves right moving to the left only if 0 or y is read. When we run out of 0's and 1's, keep head to right until a blank is found. If an empty character is read, accept.