# Assignment #5

Gustavo Estrela de Matos
CSCE 433: Formal Languages and Automata

Friday 8th April, 2016

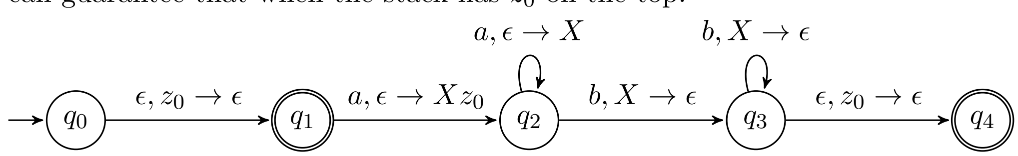
**Question 1. For each of the following languages, construct a PDA to accept $L$.**
**(a)** $L = \{a^n b^{2n} \mid n \geq 0\}$

**(b)** $L = \{a^i b^j c^k \mid i, j, k \geq 0$ **and** $i + k = j\}$

**(c)** $L = \{w \mid w$ **has twice as many** $0$**'s as** $1$**'s**$\}$

(a)

The idea for this language PDA is to have a state where the $a$'s are read and for each of them we push 2 elements in the stack; and a also a state where the $b$'s are read and for each of them an element is poped from the stack. Since we pushed 2 elements for every $a$ and poped one for every $b$ we can guarantee that when the stack has $z_0$ on the top.

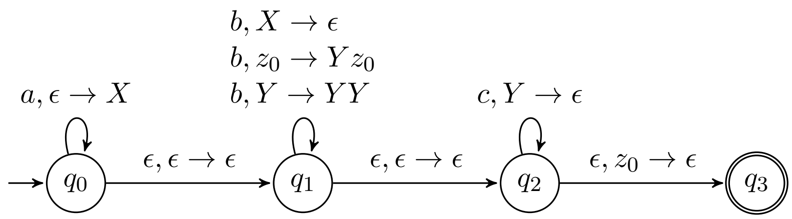



This PDA is deterministic.

(b)

Saying that $i + k = j$ is the same as saying that $k = j - i$, therefore we need to calculate $j - i$ while reading $a$s and $b$s. To do that we are going to push an element $X$ for every $a$ and pop them for every $b$ read. Once the stack have $z_0$ on the top we can start pushing $Y$s for every $b$ read.

The number of $Y$s in the stack after reading all $b$s will be the number of $c$s needed to the input to be in $L$. If after reading all the $b$s we have a $X$ on the top of the stack we know we already know that the string is not in $L$ because having a $X$ on the top means that $j < i$ thus there's no positive $k$ such that $i + k = j$.




This PDA is nondeterministic.

(c)

To build this DFA we are going to use two symbols:

- $Y$: the number of $Y$s in the stack will represent the number of 0s needed for the string to be in $L$;

- $X$: the number of $X$s in the stack will represent the number of 1s needed for the string to be in $L$;
  and the stack will keep the stack with only $z_0$ and either $X$s or $Y$s.

Our DFA will have at least two states: one to represent that the string so far read is in $L$ and other state if there are symbols needed to be read for the string to be in $L$. If we are in the second state we need to have a feew rules:

- $1, Y \to YYY$: if needed 0s and read a 1 we need two more 0.

- $0, Y \to \epsilon$: if needed 0s and read a 0 we need one less 0.

- $1, X \to \epsilon$: if needed 1s and read a 1 we need one less 1.

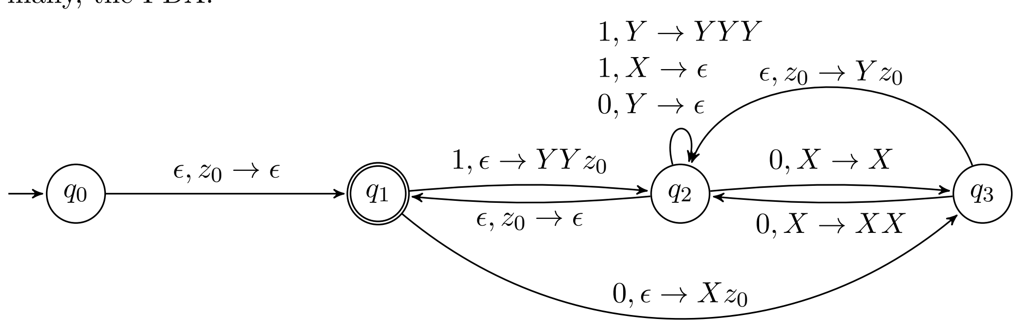
But what if we read one more 0 when needing 1s? We would need to read one more 1 and one more 0. Since we don't want to have mixed $X$s and $Y$s in our stack we will create another state that will represent our need for an amount of 1s and one more 0. From this state three things can happen:

- $1, X \to \epsilon$: if we read a 1 we would need one less 1;

- $\epsilon, z_0 \to Xz_0$: if we read all 1s we needed, then we only need to read one more 0;

- $0, X \to XX$: if we read a 0 then we only need to read one more 1 (for the 0 we read when entering and getting out of this state).

Finally, the PDA:



This PDA is deterministic.

**Question 2. Consider the following context-free grammar $G = (\{S, S_1, S_2, X\}, \{a, b\}, R, S)$, where $R$ consists of the following rules:**

$$\begin{array}{lcl} S & \rightarrow & S_1 | S_2 \\ S_1 & \rightarrow & aS_1a | bS_1b | \epsilon \\ S_2 & \rightarrow & XXS_2 | X \\ X & \rightarrow & a | b \end{array}$$

**(a) Give a succinct English description of $L(G)$.**

**(b) Is the context-free grammar $G$ ambiguous? Justify your answer.**

**(c) Give a one-state PDA $M$ that accepts $L(G)$.**

**(d) For your PDA $M$, show an accepting computation path for the string $abba$.**

**(e) For your PDA $M$, show an accepting computation path for the string $bbb$.**

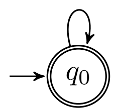
(a) This CFL represents strings of $a$s and $b$s that are palindromo or with odd length.

(b) Considering the CFLs with starting symbol $S_1$ and $S_2$ we can see that both are not ambiguous.

Since $S$ is the union of these two CFLs and they don't have any string in common (because one has only strings of odd length and other only with even length) we can say that $S$ is also not ambiguous.

(c)

$a, a \to \epsilon$
$b, b \to \epsilon$
$\epsilon, S \to S_1$
$\epsilon, S \to S_2$
$\epsilon, S_1 \to aS_1a$
$\epsilon, S_1 \to bS_1b$
$\epsilon, S_1 \to \epsilon$
$\epsilon, S_2 \to XXS_2$
$\epsilon, S_2 \to X$
$\epsilon, X \to a$
$\epsilon, X \to b$



(d) $(q, abba, S) \vdash (q, abba, S_1) \vdash (q, abba, aS_1a) \vdash (q, bba, S_1a) \vdash (q, bba, bS_1ba) \vdash (q, ba, S_1ba) \vdash (q, ba, ba) \vdash (q, a, a) \vdash (q, \epsilon, \epsilon)$.

(e) $(q, bbb, S) \vdash (q, bbb, S_2) \vdash (q, bbb, XXS_2) \vdash (q, bbb, bXS_2) \vdash (q, bb, XS_2) \vdash (q, bb, bS_2) \vdash (q, b, S_2) \vdash (q, b, X) \vdash (q, b, b) \vdash (q, \epsilon, \epsilon)$

**Question 3. In each case, use the pumping lemma to show that the given language is not a CFL.**
**(a)** $L = \{a^i b^j c^k \mid k = max(i, j)\}$

**(b)** $L = \{w \in \{a, b, c\}^* \mid \eta_a(w) < \eta_b(w) < eta_c(w)\}$**. The number of** $a$**'s,** $b$**'s, and** $c$**'s in the string** $w$ **are represented by** $\eta_a(w)$**,** $\eta_b(w)$ **and** $\eta_c(w)\}$ **respectively.**

(a)

Assume that $L$ is a CFL. Let $p$ be the pumping constant. Choose the string $s$ to be $a^p b^p c^p$. Since the $|s| \geq p$, then $s$ can be broken up into 5 pieces $u, v, x, y$ and $z$, such that:

- $|vy| \geq 1$,
- $|vxy| \leq p$, and
- for all $i \geq 0$, $uv^ixy^iz \in L$.

We must consider the following ways of partitioning the string $s$ into $u, v, x, y$, and $z$. We list partitions based on the contents of $v$ and $y$ since they affect the pumped string $s'$.

| | $v$ | $y$ | $i$ | $s' = uv^ixw^iy$ | Why is $s' \notin L$? |
|---|---|---|---|---|---|
| | | | | $v$ and $y$ contains a single distinct symbol | |
| 1. | $a^l$ | $a^m$ | 2 | $a^{p+l+m)}b^pc^p$ | Number of $a$'s is greater than the number of $c$'s. |
| 2. | $b^l$ | $b^m$ | 2 | $a^pb^{p+l+m}c^p$ | Number of $b$'s is greater than the number of $c$'s. |
| 3. | $c^l$ | $c^m$ | 0 | $a^pb^pc^{p-(l+m)}$ | Number of $c$'s is less than $p = max(p,p)$. |
| | | | | $v$ and $y$ contain two different distinct symbols | |
| 4. | $a^l$ | $b^m$ | 2 | $a^{p+l}b^{p+m}c^p$ | Since $l$ and $m$ are greater than zero (otherwise, case 1 or 2)we have that $p$ can't be $max(p+l, p+m)$. |
| 5. | $b^l$ | $c^m$ | 0 | $a^pb^{p-l}c^{p-m}$ | Since $m$ is greater than zero (otherwise, case 2) we have that $p - m$ can't be $max(p, p-l)$. |
| | | | | $v$ or $y$ contain 2 distinct symbols | |
| 6. | $a^l$ | $a^mb^n$ | 0 | $a^{p-(l+m)}b^{p-n}c^p$ | Since $l$, $m$ and $n$ are greater than zero (othewise, case 1, 2, 4) we have that the number of $c$'s is greater than $max(p-(l+m), p-n)$ |
| 7. | $a^lb^m$ | $b^n$ | 0 | $a^{p-l}b^{p-(m+n)}c^p$ | Since $l$, $m$ and $n$ are greater than zero (othewise, case 1, 2, 4) we have that the number of $c$'s is greater than $max(p-l, p-(n+m))$. |
| 8. | $b^l$ | $b^mc^n$ | 0 | $a^pb^{p-(l+m)}c^{p-n}$ | Since $n$ is greater than zero (otherwise, case 2) we have that the number of $c$'s is less than $max(p, p-(l+m))$ |
| 9. | $b^lc^m$ | $c^n$ | 2 | $a^pb^{p-l}c^{p-(m+n)}$ | Since $n$ and $m$ are greater than zero (otherwise, case 2 or 5) we have that the number of $c$'s is less than $max(p, p-l)$ |

The above cases show that no matter how we partition the string $s$ into $u, v, x, y$, and $z$, the resulting pumped string $s'$ will not be in the language $L$. Thus, our assumption that $L$ is context-free is false.

(b)

Assume that $L$ is a CFL. Let $p$ be the pumping constant. Choose the string $s$ to be $a^pb^{p+1}c^{p+2}$. Since the $|s| \geq p$, then $s$ can be broken up into 5 pieces $u, v, x, y$ and $z$, such that:

- $|vy| \geq 1$,
- $|vxy| \leq p$, and
- for all $i \geq 0$, $uv^ixy^iz \in L$.

We must consider the following ways of partitioning the string $s$ into $u, v, x, y$, and $z$. We list partitions based on the contents of $v$ and $y$ since they affect the pumped string $s'$.

| | $v$ | $y$ | $i$ | $s' = uv^ixw^iy$ | Why is $s' \notin L$? |
|---|---|---|---|---|---|
| | | | | $v$ and $y$ contains a single distinct symbol | |
| 1. | $a^l$ | $a^m$ | 2 | $a^{p+l+m}b^{p+1}c^{p+2}$ | Since $l+m > 0$, the number of $a$'s is greater or equal than the number of $b$'s. |
| 2. | $b^l$ | $b^m$ | 2 | $a^pb^{p+l+m}c^p$ | Since $l+m > 0$, the number of $b$'s is greater or equal than the number of $c$'s. |
| 3. | $c^l$ | $c^m$ | 0 | $a^pb^{p+1}c^{p+2-(l+m)}$ | Since $l+m > 0$, the number of $c$'s is less or equal than the number of $b$'s. |
| | | | | $v$ and $y$ contain two different distinct symbols | |
| 4. | $a^l$ | $b^m$ | 2 | $a^{p+l}b^{p+1+m}c^{p+2}$ | Since $m$ is greater than zero (otherwise, case 1) we have that the number of $b$'s is greater or equal to the number of $c$'s. |
| 5. | $b^l$ | $c^m$ | 0 | $a^pb^{p-l+1}c^{p-m+2}$ | Since $l$ is greater than zero (otherwise, case 3) we have that the number of $b$'s is less or equal than the number of $a$'s. |
| | | | | $v$ or $y$ contain 2 distinct symbols | |
| 6. | $a^l$ | $a^mb^n$ | 2 | $a^{p+l+m}b^{p+n+1}c^{p+2}$ | Since $n$ is greater than zero (othewise, case 1) we have that the number of $b$'s is greater or equal to the number of $c$'s |
| 7. | $a^lb^m$ | $b^n$ | 2 | $a^{p+l}b^{p+m+n+1}c^{p+2}$ | Since $m$ and $n$ are greater than zero (otherwise, case 1, 4) we have that the number of $b$'s is greater or equal than the number of $c$'s. |
| 8. | $b^l$ | $b^mc^n$ | 0 | $a^pb^{p-(l+m)+1}c^{p+2-n}$ | Since $l$ and $m$ are greater than zero (otherwise, case 3 or 5) we have that the number of $b$'s is less than the number of $a$'s |
| 9. | $b^lc^m$ | $c^n$ | 0 | $a^pb^{p-l+1}c^{p-(m+n)+2}$ | Since $n$ and $m$ are greater than zero (otherwise, case 3 or 5) we have that the number of $c$'s is less than the number of $b$'s. |

The above cases show that no matter how we partition the string $s$ into $u, v, x, y$, and $z$, the resulting pumped string $s'$ will not be in the language $L$. Thus, our assumption that $L$ is context-free is false.

**Question 4. For each of the following languages, determine whether it is context-free or not, and a proof for your answer.**
**(a)** $L = \{xayb \mid x, y \in \{a, b\}^*$ **and** $|x| = |y|\}$

**(b)** $L = \{\alpha\beta\alpha \mid \alpha, \beta \in \{a, b\}^*$ **and** $|\alpha| \geq 1\}$ **(changed variable names to avoid conflict with pumping lemma variables)**
**(c)** $L = \{(ab^n)^n \mid n \geq 0\}$

(a) This language is a CFL and to prove this we are going to build a CFG $G = (V, \Sigma, R, S)$ that represents this language.

- $V = \{S, A, T\}$
- $\Sigma = \{a, b\}$
- $R = \{S \to Ab, A \to TAT, A \to a, T \to a, T \to b\}$

(b)

Assume that $L$ is a CFL. Let $p$ be the pumping constant. Choose the string $s$ to be $a^p b^p c^p$. Since the $|s| \geq p$, then $s$ can be broken up into 5 pieces $u, v, x, y$ and $z$, such that:

- $|vy| \geq 1$,
- $|vxy| \leq p$, and
- for all $i \geq 0$, $uv^ixy^iz \in L$.

We must consider the following ways of partitioning the string $s$ into $u, v, x, y$, and $z$. We list partitions based on the contents of $v$ and $y$ since they affect the pumped string $s'$.

Note that if we pump $s$ and get $s'$ and if $s'$ can be written as $a_1^i b_1^j a_2^k b_2^l$ the $\beta$ part of the string can't do more than subtract $b_1$'s or $a_2's$, therefore, if the pumped string has more $b_2$'s than $b_1$'s or more $a_1$'s than $a_2$'s we can say that $s'$ is not in L.

| | $v$ | $y$ | $i$ | $s' = uv^ixw^iy$ | Why is $s' \notin L$? |
|---|---|---|---|---|---|
| $v$ and $y$ contains a single distinct symbol | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 1. | $a_1^i$ | $a_1^j$ | 2 | $a^{p+i+j}b^pa^pb^p$ | More $a_1$'s than $a_2$'s. |
| 2. | $b_1^i$ | $b_1^j$ | 0 | $a^pb^{p-(i+j)}a^pb^p$ | More $b_2$'s than $b_1$'s. |
| 3. | $a_2^i$ | $a_2^j$ | 0 | $a^pb^pa^{p-(i+j)}b^p$ | More $a_1$'s than $a_2$'s. |
| 4. | $b_2^i$ | $b_2^j$ | 2 | $a^pb^pa^pb^{p+i+j}$ | More $b_2$'s than $b_1$'s. |
| $v$ and $y$ contain two different distinct symbols | | | | | |
| 5. | $a_1^i$ | $b_1^j$ | 0 | $a^{p-i}b^{p-j}a^pb^p$ | More $b_2$'s than $b_1$'s. |
| 6. | $b_1^i$ | $a_2^j$ | 0 | $a^pb^{p-i}a^{p-j}b^p$ | More $a_1$'s than $a_2$'s. |
| 7. | $a_2^i$ | $b_2^j$ | 0 | $a^pb^pa^{p-i}b^{p-j}$ | More $a_1$'s than $a_2$'s. |
| $v$ or $y$ contain 2 distinct symbols | | | | | |
| 8. | $a_1^i$ | $a_1^jb_1^k$ | 0 | $a^{p-(i+j)}b^{p-k}a^pb^p$ | More $b_2$'s than $b_1$'s. |
| 9. | $b_1^i$ | $b_1^ja_2^k$ | 0 | $a^pb^{p-(i+j)}a^{p-k}b^p$ | More $b_2$'s than $b_1$'s. |
| 10. | $a_2^i$ | $a_2^jb_2^k$ | 0 | $a^pb^pa^{p-(i+j)}b^{p-k}$ | More $a_1$'s than $a_2$'s. |
| 11. | $a_1^ib_1^j$ | $b_1^k$ | 0 | $a^{p-i}b^{p-(j+k)}a^pb^p$ | More $b_2$'s than $b_1$'s. |
| 12. | $b_1^ia_2^j$ | $a_2^k$ | 0 | $a^pb^{p-i}a^{p-(j+k)}b^p$ | More $b_2$'s than $b_1$'s. |
| 13. | $a_2^ib_2^j$ | $b_2^k$ | 0 | $a^pb^pa^{p-i}b^{p-(j+k)}$ | More $a_1$'s than $a_2$'s. |

The above cases show that no matter how we partition the string $s$ into $u, v, x, y$, and $z$, the resulting pumped string $s'$ will not be in the language $L$. Thus, our assumption that $L$ is context-free is false.

(c) This language is not a CFL. To prove this we are going to use the pumping lemma.

Assume that $L$ is a CFL. Let $p$ be the pumping constant. Choose the string $s$ to be $(ab^p)^p$. Since the $|s| \geq p$, then $s$ can be broken up into 5 pieces $u, v, x, y$ and $z$, such that:

- $|vy| \geq 1$,
- $|vxy| \leq p$, and
- for all $i \geq 0$, $uv^ixy^iz \in L$.

According to the pumping lemma, $s' = uxz \in L$, but note that $|s'| = |s| - (|v| + |x|)$ and since $|vxy| \leq p$ and $|vy| \geq 1$ we have that $|s| = (1+p)p = \underline{p^2 + p > |s'| \geq p^2} = |s| - p$. Also, the biggest string of $L$ that is smaller than $|s|$ is $(ab^{n-1})^{n-1}$, which has length $((p-1)+1)(p-1) = p^2 - p$. Therefore, there is no string in $L$ with length in the interval $(p^2 + p, p^2]$ and $|s'|$ is not in $L$, a contradiction caused by the assumption that this language is a CFL. Then, $L$ is not a CFL.