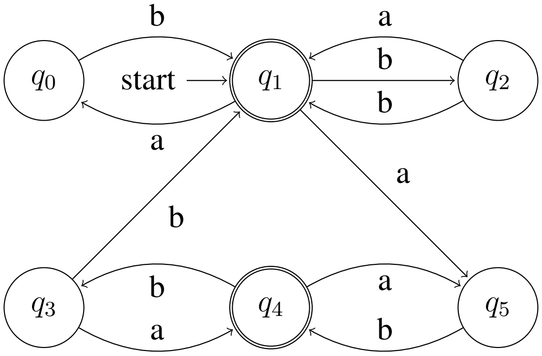
# Comp 330 (Fall 2024): Assignment 2

Answers must be submitted online by Oct 24 (11:59 pm), 2024.

## General instructions

- **Important:** All of the work you submit must be done by only you, and your work must not be submitted by someone else. Plagiarism is academic fraud and is taken very seriously.
- To some extent, collaborations are allowed. These collaborations should not go as far as sharing code or giving away the answer. **You must indicate on your assignments the names of the people with whom you collaborated or discussed your assignments (including members of the course staff). If you did not collaborate with anyone, write "No collaborators". If asked, you should be able to orally explain your solution to a member of the course staff.**
- It is your responsibility to guarantee that your assignment is submitted on time. We do not cover technical issues or unexpected difficulties you may encounter. Last minute submissions are at your own risk.
- Multiple submissions are allowed before the deadline. We will only grade the last submitted file. Therefore, we encourage you to submit as early as possible a preliminary version of your solution to avoid any last minute issue.
- Late submissions can be submitted for 24 hours after the deadline, and will receive a flat penalty of 20%. We will not accept any submission more than 24 hours after the deadline. The submission site will be closed, and there will be no exceptions, except medical.
- In exceptional circumstances, we can grant a small extension of the deadline (e.g. 24h) for medical reasons only. However, such request must be submitted before the deadline, justified and approved by the instructors.
- Violation of any of the rules above may result in penalties or even absence of grading. If anything is unclear, it is up to you to clarify it by asking either directly the course staff during office hours, by email at or on the discussion board on Ed. Please, note that we reserve the right to make specific/targeted announcements affecting/extending these rules in class and/or on the website. It is your responsibility to monitor Ed for announcements.
- The course staff will answer questions about the assignment during office hours or in the online forum. We urge you to ask your questions as early as possible. We cannot guarantee that questions asked less than 24h before the submission deadline will be answered in time. In particular, we will not answer individual emails about the assignment that are sent the day of the deadline.
- Unless specified, **you must show your work and all answers must be justified.**

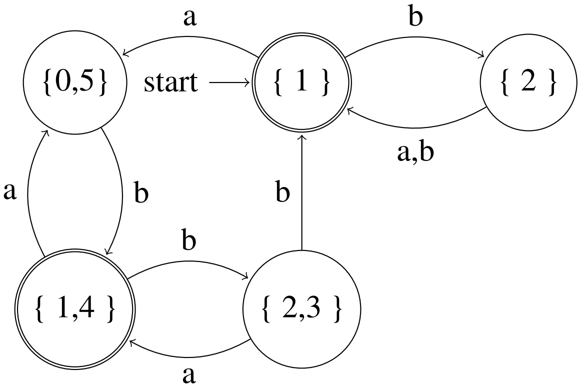
1. Let $\mathcal{A}$ be the automaton depicted below.



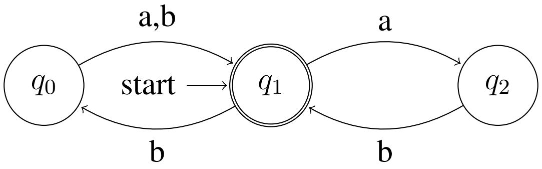
   (a) (10 points) Compute a minimal deterministic finite automata (DFA) from $\mathcal{A}$.

   (b) (10 points) Using the minimal DFA to determine a regular expression representing $L(\mathcal{A})$

**Solution:** The deterministic version of the automaton:



And the minimal version of this DFA is:



The regular expression is therefore: $(ab + b(a + b))^*$

2. (20 points) Let $\Sigma$ be a (non-empty) alphabet and let $w \in \Sigma^*$ be a string. We say that $x \in \Sigma^*$ is a prefix of the string $w$ if there exists a string $u \in \Sigma^*$ such that $w = xu$.
Consider the following language

$$L = \{w \in \{a, b\}^* | \text{for every prefix x of } w \; n_b(x) \geq n_a(x)\}$$

Prove that $L$ is a context-free language. Your proof should rely on mathematical induction.

**Solution:** To prove that $L$ is context-free, we first create a context-free grammar $G$ which generates it and then prove the correctness of $G$. Consider the grammar $G = (\{S\}, S, \{a, b\}, P)$ with production rules $P$:

$$S \to \ \epsilon \mid bSaS \mid bS$$

The base case $S \to \ \epsilon$ has been encoded. For the recursive case, if an $a$ is generated, it must be generated after a $b$. Otherwise, $b$'s can be generated without restriction. We now show that $L = L(G)$.

• $L \subseteq L(G)$.

We prove the claim $P(n)$ : if $w \in L$ and $|w| = n$ for some $n \in \mathbb{N}$, then $S \to^* w$. We do so by strong induction on $n$.

**Base case:** We show that $P(n)$ holds for $n = 0$ and $n = 1$. For $n = 0$, we have that $w = \epsilon$. Since $S \to \epsilon \in P$, we have that $S \to^* \epsilon$. For $n = 1$, it must be that $w = b$ for $w \in L$. Since $S \to bS \to b\epsilon = b$, we have that $S \to^* b$.

**Inductive hypothesis:** For some $n \in \mathbb{N}$, $n \geq 1$, we assume that $\forall k \in \mathbb{N}$, $0 \leq k \leq n$, $P(k)$ holds.

**Inductive step:** We show that $P(n + 1)$ holds. Suppose $w \in L$ and $|w| = n + 1$. It must be that the first letter of $w$ is a $b$. Thus $w = bx$. We split the inductive step into two cases based on the nature of $x$.

If $x \in L$, then since $|x| = n$, by the inductive hypothesis $S \to^* x$. This implies that we can generate $w$ since $S \to bS \to^* bx$.

If $x \notin L$, this must be because there was an $a$ at position $j$ in $x$ which caused $n_b(x_{1:j}) - n_a(x_{1:j}) < 0$ and, more specifically, $n_b(x_{1:j}) - n_a(x_{1:j}) = -1$. If we consider the first time this happened (i.e., where there was an extra $a$ in $x$ which caused a violation of the conditions of $L$), we can decompose $x$ as $x = u_1 a u_2$. By our choice of $a$, $u_1 \in L$. Furthermore $u_2 \in L$ because, if it was not, $w \notin L$ (Why?). Therefore, since $0 \leq |u_1| \leq n - 1$ and $0 \leq |u_2| \leq n - 1$, by the inductive hypothesis, $S \to^* u_1$ and $S \to^* u_2$. Therefore, we can generate $w$ since $S \to bSaS \to^* bu_1aS \to^* bu_1au_2 = w$.

• $L(G) \subseteq L$

We prove the claim $P(n)$: if $S \to^n w$ for some $n \in \mathbb{N}$ where $w \in \{a, b\}$, then $w \in L$. We do so by strong induction on $n$.

**Base case:** We show that $P(n)$ holds for $n = 1$ and $n = 2$. For $n = 1$, the only string of terminals that can be generated is $\epsilon$ through $S \to \epsilon$. Since $\epsilon \in L$, this verifies this first base case. For $n = 2$, the grammar can use two derivation steps to derive $b$ by $S \to bS \to b\epsilon = b$.

**Inductive hypothesis:** For some $n \in \mathbb{N}$, $n \geq 2$, we assume that $\forall k \in \mathbb{N}$, $1 \leq k \leq n$, $P(k)$ holds.

**Inductive step:** We show that $P(n{+}1)$ holds. If $S \to^n w \in \{a, b\}^*$, then the first derivation step would have been either $S \to bS$ or $S \to bSaS$. We consider both cases separately.

If the first derivation step used was $S \to bS$, then $S \to^n x \in \{a, b\}^*$ such that $w = bx$. By the inductive hypothesis, $x \in L$. Pre-pending a $b$ to $x$ does not break the conditions of $L$ thus $w = bx \in L$.

If the first derivation step used was $S \to bSaS$, then each $S$ would have $1 \leq t \leq n-1$ derivation steps to generate a string of terminals. By the inductive hypothesis, this means both would have generated strings $u_1, u_2 \in L$. Therefore $w = bu_1au_2$ which is in $L$ since the extra $a$ is offset by the starting $b$.

This completes the proof that $L = L(G)$. Thus $L$ is indeed context-free.

3. (10 points) Compute the Chomsky Normal Form of the following context-free grammar (CFG).

$$\begin{aligned}
&S \to aAa|bBb|\epsilon \\
&A \to C|a \\
&B \to C|b \\
&C \to CDE|\epsilon \\
&D \to A|B|ab
\end{aligned}$$

**Solution:** First, we remove $\epsilon$-productions.

$$\begin{aligned}
S &\to aAa \mid bBb \mid aa \mid \epsilon \\
A &\to C \mid a \\
B &\to C \mid b \\
C &\to CDE \mid DE \mid CE \mid E \\
D &\to A \mid B \mid ab
\end{aligned}$$

Then, we remove unit productions. We first identify the unit pairs: $\{(S,S),(A,A),(B,B),(C,C),$ $(D,D),(E,E),(A,C),(B,C),(D,A),(D,B),(C,E),(A,E),(B,E),(D,E)\}$ and derive from them the productions (below the right-hand sides).

| $Pair$ | $Productions$ |
|---|---|
| $(S,S)$ | $aAa \mid bBb \mid aa \mid bb$ |
| $(A,A)$ | $a$ |
| $(B,B)$ | $b$ |
| $(C,C)$ | $CDE \mid DE \mid CE$ |
| $(D,D)$ | $ab$ |
| $(E,E)$ | $\emptyset$ |
| $(A,C)$ | $CDE \mid DE \mid CE$ |
| $(B,C)$ | $CDE \mid DE \mid CE$ |
| $(D,A)$ | $CDE \mid DE \mid CE \mid a$ |
| $(D,B)$ | $CDE \mid DE \mid CE \mid b$ |
| $(C,E)$ | $\emptyset$ |
| $(A,E)$ | $\emptyset$ |
| $(B,E)$ | $\emptyset$ |
| $(D,E)$ | $\emptyset$ |

We combine all productions found after identifyin the unit pairs:

$$\begin{aligned} S &\rightarrow aAa \mid bBb \mid aa \mid bb \\ A &\rightarrow CDE \mid DE \mid CE \mid a \\ B &\rightarrow CDE \mid DE \mid CE \mid b \\ C &\rightarrow CDE \mid DE \mid CE \\ D &\rightarrow CDE \mid DE \mid CE \mid a \mid b \mid ab \end{aligned}$$

We eliminating variables that derive no terminal string.

$$\begin{aligned} S &\rightarrow aAa \mid bBb \mid aa \mid bb \\ A &\rightarrow a \\ B &\rightarrow b \\ D &\rightarrow a \mid b \mid ab \end{aligned}$$

We eliminating variables that are not accessible from the start symbol.

$$\begin{aligned} S &\rightarrow aAa \mid bBb \mid aa \mid bb \\ A &\rightarrow a \\ B &\rightarrow b \end{aligned}$$

Finally, we derive the CNF.

$$\begin{aligned} S &\rightarrow XA \mid YB \mid AA \mid BB \\ A &\rightarrow a \\ B &\rightarrow b \\ X &\rightarrow AA \\ Y &\rightarrow BB \end{aligned}$$

4. (20 points) State whether the following claim is true or false and prove your answer.

   **Claim:** For any (non-empty) alphabet $\Sigma$, there is no language $L \subseteq \Sigma^*$ which is both regular and inherently ambiguous.

Hint: Use a context-free grammar recognizing $L$ to show a contradiction if it were ambiguous.

**Solution:** True. If a language $L$ is regular, then there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that $L = L(M)$ that accepts it. We can convert this DFA to a right-linear grammar $G = (V, S, T, P)$

such that $L(G) = L$ where:

$$\begin{aligned} V =& Q \\ S =& q_0 \\ T =& \Sigma \\ P :& \\ & q \to \sigma p \iff \exists \sigma \in \Sigma, p, q \in Q, \delta(q, \sigma) = p \\ & q \to \epsilon \iff q \in F \end{aligned}$$

We claim that this grammar $G$ is not ambiguous. Suppose it were. There there would exist a $w \in L$ such that at least two distinct leftmost derivations existed such that $S \to_{lm}^{*1} w$ and $S \to_{lm}^{*2} w$.

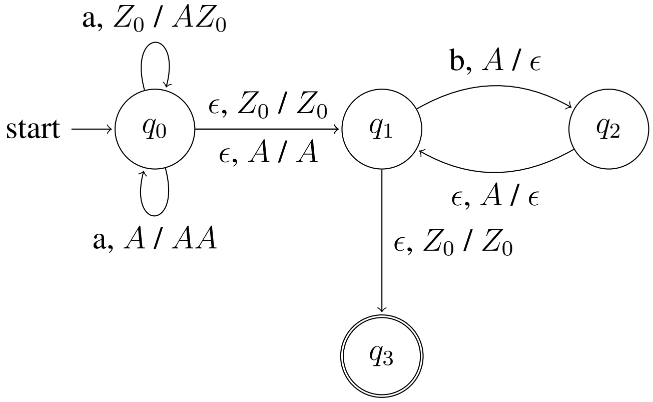
We represent $w$ explicitely as $w = \sigma_1 \sigma_2 \cdots \sigma_{|w|}$ and consider the first derivation step where $S \to_{lm}^{*1} w$ and $S \to_{lm}^{*2} w$ differ. That is for $\sigma, \gamma \in \Sigma^*$, and $p, q, r \in Q$ we have that:

$$S \to_{lm}^{*} xq \to_{lm} x\sigma p \to_{lm}^{*} w \text{ , for } S \to_{lm}^{*1} w$$
$$S \to_{lm}^{*} xq \to_{lm} x\gamma r \to_{lm}^{*} w \text{ , for } S \to_{lm}^{*2} w$$

If $\sigma = \gamma$ then $p \neq r$, but then, by construction, $\delta(q, \sigma) = p$ and $\delta(q, \sigma) = r$ where $r \neq q$, a contradiction since M was deterministic. Thus $\sigma \neq \gamma$. But since all derivations were the same up until $xq$, we have that $w = x\sigma u_1$ and $w = x\gamma u_2$ for $u_1, u_2 \in \Sigma^*$. Thus the same string $w$ has differing letters at the same position $|x| + 1$, a contradiction. Thus, it cannot be that $G$ was ambiguous and thus there is an unambiguous grammar which generates every regular language.

5. (20 points) Build a Pushdown Automaton (PDA) recognizing the language $L = \{a^i b^j | i = 2 \cdot j\}$. The PDA must recognize the words by an *empty stack*. Briefly explain how your PDA works. Then, describe an execution of your PDA on $aaaabb$ and show it accepts it.

**Solution:**



Execution on $aaaabb$:
$(q_0, aaaabb, Z_0) \vdash (q_0, aaabb, AZ_0) \vdash (q_0, aabb, AAZ_0) \vdash (q_0, abb, AAAZ_0) \vdash (q_0, bb, AAAAZ_0)$
$\vdash (q_1, bb, AAAAZ_0) \vdash (q_2, b, AAAZ_0) \vdash (q_1, b, AAZ_0) \vdash (q_2, \epsilon, AZ_0) \vdash (q_1, \epsilon, Z_0) \vdash (q_3, \epsilon, Z_0)$

6. (10 points) Use the pumping lemma to show that the language $L = \{a^n b^n a^n b^n | n \geq 0\}$ is not context-free.

**Solution:** We assume $L$ is context-free.
Let $n \geq 0$ be the pumping lemma constant and consider the string $z = a^n b^n a^n b^n$. $z \in L$ obviously. Since $|z| > n$, under the pumping lemma, we can write $z = uvwxy$ with $|vwx| \leq n$ and $|vx| \geq 1$. We now discuss where $vwx$ occurs in $z$. There are two cases:

- $vwx$ has no $b$'s. Then, it is entirely contained within the first or second block of $a$'s. Since $|vx| \geq 1$, $v$ or $x$ constains at least one $a$. Using the pumping lemma, we have $uv^0wx^0y \in L$. But $uv^0wx^0y$ has a block of $a$'s (those where $vwx$ belongs to) with at least one less $a$. Contradiction. The same argument works if $vwx$ has no $a$'s.
- $vwx$ has a mix of $a$'s and $b$'s. Because $|vwx| \leq n$, there are only two cases. Either we have a string of $a$'s followed by a string of $b$'s, or the opposite. Assume the former. Then, $vwx$ straddles the first blocks of $a$'s and $b$'s or the second ones. Thus, in $uv^0wx^0y$, the combined length of these two blocks will decrease while the length of the two other will not change. Therefore, $uv^0wx^0y$ cannot belong to $L$. Contradiction.

| Question: | 1 | 2 | 3 | 4 | 5 | 6 | Total |
|---|---|---|---|---|---|---|---|
| Points: | 20 | 20 | 10 | 20 | 20 | 10 | 100 |
| Score: | | | | | | | |