CS 4510 Automata and Complexity

Exam 2A: Context-Free Languages

Date: 3:30-4:45pm Tuesday, October 24 *Prof. Jaeger*

Name: ______________________ GTID: ______________________

- **When handing in this test, please go to a TA with your BuzzCard to verify the name on the test.**
- Do not open this exam until you are directed to do so. Read all the instructions first.
- **Write your name and GTID# on the top of every page.** Your GTID# can be found on your BuzzCard.
- By submitting this exam, you agree that your actions during this exam conform with the Georgia Tech Honor Code.
- This exam is out of 100 points (with one extra credit point for writing your name). A curve may be applied.
- Write your solutions in the space provided. If you run out of space, continue your answer on the back of the last page, and make a notation on the front of the sheet.
- There is a page of scratch paper at the end of the exam. We can provide additional scratch paper if you need it. The only outside material you may use on the exam is one (1) double-sided, hand-written, 8.5"x11" page of notes.
- Calculators are NOT permitted.
- You may use any of the theorems/facts/lemmas from the lecture notes, homeworks, or textbook without re-proving them unless explicitly stated otherwise.
- If you have a question, you may ask a TA. You should not communicate with anyone other than teaching staff during the exam.
- Do not spend too much time on any one problem! If you get stuck, move on and come back to that one later.
- Each problem is labeled with the number of points it is expected to be worth. These values may be shifted if any problems end up being more difficult than expected.
- Good luck!

Name: ______________________ GTID: ______________________

0. **Following Instructions [1 point extra credit]**

   Please write your name and GTID# at the top of EACH page of this exam. Your GTID# can be found on your BuzzCard.

1. **Select All [16 points]** Select *all* that apply. Correct choices are bolded.

   (a) Which of the following models can recognize the language $\{ww \mid w \in \{0,1\}^*\}$?.

   ◯ **A "PDA" with two stacks**

   ◯ A PDA

   ◯ An NFA

   ◯ **A Turing machine**

   (b) L_1 and L_2 are context-free languages. Which of the following MUST be true?

   ◯ $L_1 \cap L_2$ is context free.

   ◯ **There exists a grammar in CNF that generates $L_1 \cup L_2$.**

   ◯ There exists an unambiguous grammar that generates $L_1 \cup L_2$.

   ◯ There exists a PDA that recognizes $\overline{L_1}$ (the complement of L_1).

2. **The Empty Language [8 points]** Give three different grammars which generate the empty language. This means that your languages must have fundamentally different sets of rules - changing the names of variables is not sufficient.

   Solution: *Empty* is just a adjective. So the empty language, is the language that does not include anything (i.e., it is a language and it is empty). Note that the language $\{\varepsilon\}$ in NOT empty. It contains something, namely the empty string ε.[1]

   We were generous to students who gave grammars for $\{\varepsilon\}$ instead of the empty language $\emptyset = \{\}$. We will not be generous about this on the final. Here are example grammars.

The empty grammar.	$S \to S$ No terminals.	$S \to AS$ $A \to \texttt{a}$ No way to get rid of S.

[1] Again, here empty is an adjective. The empty string is a string and it is empty.

Name: ______________________ GTID: ______________________

3. **Ambiguity [8 points]** Give an unambiguous grammar that generates the same language as the following ambiguous grammar:

$$A \rightarrow AA|\mathtt{a}|\varepsilon$$

Solution: Note that this is a grammar for the language described by the regular language a^*. It is ambiguous, for example, because $\mathtt{a}$ can be derived by $A \Rightarrow AA \Rightarrow \varepsilon A \Rightarrow \mathtt{a}$ or by $A \Rightarrow AA \Rightarrow \mathtt{a}A \Rightarrow \mathtt{a}$ The following is an unambiguous grammar for that language.

$$A \rightarrow Aa|\varepsilon$$

4. **Chomsky-Normal Form [8 points]** Using the method shown in class or any other way, give a grammar in CNF form that generates the same language as the grammar given below. You can show intermediate steps but your final grammar must be clearly marked.

$$S \rightarrow A$$
$$A \rightarrow 0A0|1A1|\varepsilon$$

Solution: All of the rules in the grammar are disallowed by CNF. We will first get rid of the rule $A \rightarrow \varepsilon$. Then the rule $S \rightarrow A$. Then we will add intermediate variables V_{0A}, V_{1A}, V_0, V_1 to clean up the remaining rules.

$S \rightarrow A\|\varepsilon$ $A \rightarrow 0A0\|1A1\|00\|11$	$S \rightarrow 0A0\|1A1\|00\|11\|\varepsilon$ $A \rightarrow 0A0\|1A1\|00\|11$	$S \rightarrow V_{0A}V_0\|V_{1A}V_1\|V_0V_0\|V_1V_1\|\varepsilon$ $A \rightarrow V_{0A}V_0\|V_{1A}V_1\|V_0V_0\|V_1V_1$ $V_{0A} \rightarrow V_0A$ $V_{1A} \rightarrow V_1A$ $V_0 \rightarrow 0$ $V_1 \rightarrow 1$

Name: ____________________ GTID: ____________________

5. **Context-Free Grammars [15 points]** The following three languages are all context free. Choose TWO of them and write CFGs that generate these languages. Clearly indicate which languages you choose.

   (a) $\{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k\mathtt{d}^l \mid j = i + 2k + 3l \text{ and } i, j, k, l \in \mathbb{Z}_{\geq 0}\}$.
   Solution:

$$
\begin{aligned}
S &\to AD \\
A &\to \mathtt{a}A\mathtt{b} | \varepsilon \\
D &\to \mathtt{bbb}D\mathtt{d} | C \\
C &\to \mathtt{bb}C\mathtt{c} | \varepsilon
\end{aligned}
$$

   (b) $\{w\#w^{\mathcal{R}}\#\mathtt{1}^n\mathtt{0}^n \mid w \in \{0,1\}^* \text{ and } n \in \mathbb{Z}_{\geq 0}\}$. (Recall that $w^{\mathcal{R}}$ is the reverse of w.)
   Solution:

$$
\begin{aligned}
S &\to A\#B \\
A &\to 1A1 | 0A0 | \# \\
B &\to 1B0 | \varepsilon
\end{aligned}
$$

   (c) $\{1^n w 1^n \mid w \in \{0,1,2,3,4,5,6,7,8,9\}^* \text{ and } n \in \mathbb{Z}_{\geq 0}\}$.
   Solution:

$$
\begin{aligned}
S &\to 1S1 | A \\
A &\to AA \\
A &\to 0|1|2|3|4|5|6|7|8|9|\varepsilon
\end{aligned}
$$

   Alternatively, one can note that any beginning or ending 1's can be incorporated into w and so this language is just $\{0,1,2,3,4,5,6,7,8,9\}^*$, giving the following grammar.

$$
S \to SS|0|1|2|3|4|5|6|7|8|9|\varepsilon
$$

Name: ______________________ GTID: ______________________

6. **Push-down Automata [15 points]** The following three languages are all context free. Choose TWO of them and draw PDAs that accept these languages. Clearly indicate which languages you choose.

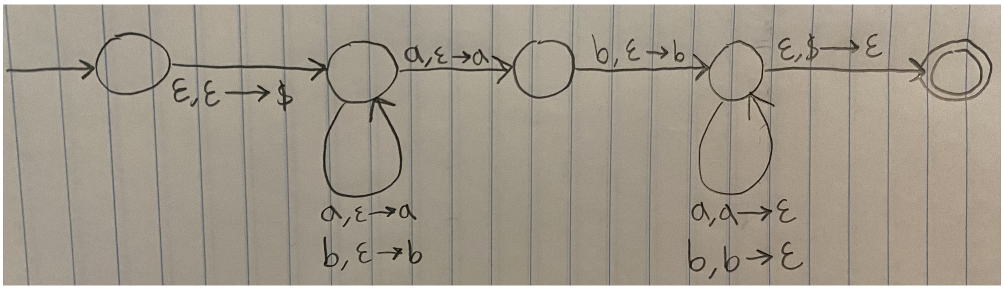
   (a) $\{w \mid w \in \{\mathtt{a}, \mathtt{b}\}^*$ is an even length palindrome with `abba` as the middle characters$\}$

   (b) $\{w \mid w \in \{0, 1\}^*$ has twice as many `0`s as `1`s$\}$

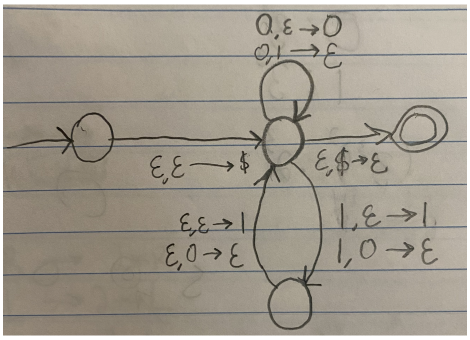
   (c) $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^m\mathtt{d}^m \mid n, m \in \mathbb{Z}_{\geq 0}\}$

   Solutions

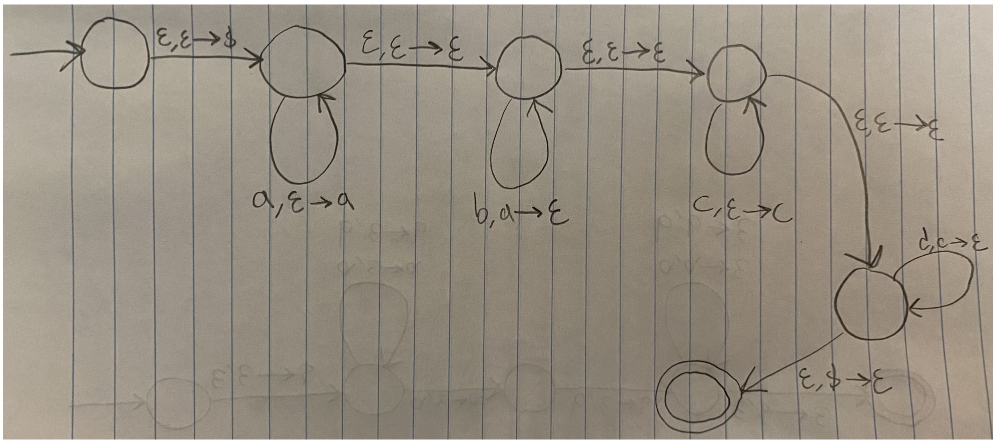
   (a) The following PDA pushes input letters on the stack, then guesses when it has gotten close to the middle where it requires that the input have an `a` and a for it to push on the stack. Then it checks if the rest of the input matches what is on the stack. Additional answers that were correct included accounting for all four letters of "abba" manually, or popping "ba" after the midpoint.



   (b) This PDA pushes a 0 on the stack or pops a 1 off the stack whenever it sees a 0. When it sees a 1, it performs two operations of either pushing a 1 or popping a 0.



   (c) This PDA pushes the `a`'s it seems, then pops them to see it the input has the same number of `b`'s. It repeats this with the `c`'s and `d`'s.



Name: ______________________ GTID: ______________________

7. **Pumping Lemma [15 points]** The following two languages are not context-free. Choose ONE of the following languages and use the context-free pumping lemma to prove that it is not context-free. Make sure to clearly state which language you are choosing.

   Choose one:

   (a) $\{\mathtt{1}^n \# w \# \mathtt{0}^n \mid w \in \{0,1\}^* \text{ is a palindrome and } |w| = n\}$

   Solution: We use the contrapositive of the pumping lemma. Let $p \in \mathbb{N}$ be given and define

   $$s = 1^p \# 1^p \# 0^p.$$

   Note that $|s| = 3p + 2 > p$ and s is in the language because 1^p is a palindrome of the appropriate length. We will argue that any choice of split $uvxyz = s$ is not pumpable. In particular, if $|vy| > 0$ and $|vxy| \leq p$, then $\psi = uv^2xy^2z$ will not be in the language.

   If vy contains an octothorpe, then ψ will contain more than two octothorpes and so is not in the language. Suppose vy does not contain an octothorpe. Then $\psi = 1^i \# 1^j \# 1^k$ for some i, j, k. This is only in the language if $i = j = k$. As $|vxy| \leq p$, vy cannot contain characters from before the first octothorpe and after the last one. So either i or k must still equal p. However, $i + j + k = 3p + |vy| > 3p$ so at least one of i, j, k is not p. Hence ψ is not in the language.

   (b) $\{\mathtt{a}^n \mathtt{b}^{n+m} \mathtt{c}^m \mathtt{a}^n \mid m, n \in \mathbb{Z}_{\geq 0}\}$

   Solution: We use the contrapositive of the pumping lemma. Let $p \in \mathbb{N}$ be given and define

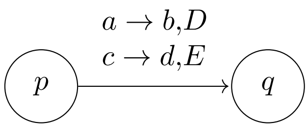
   $$s = \mathtt{a}^p \mathtt{b}^p \mathtt{a}^p$$

   Note that $|s| = 3p > p$ and s is in the language from $n = p, m = 0$.[2] We will argue that any choice of split $uvxyz = s$ is not pumpable. In particular, if $|vy| > 0$ and $|vxy| \leq p$, then $\psi = uv^0xy^0z$ will not be in the language.

   Note that $\psi = \mathtt{a}^i \mathtt{b}^j \mathtt{a}^k$ for some i, j, k where $i + j + l = 3p - |vy| < 3p$. So (at least) one of i, j, k must be less than p. However, i and l cannot both be less than p as that would require $|vxy| \geq p + 2$. So $i = j = k$ cannot hold, meaning ψ is not in the language.

[2] If we chose any string with $m > 0$, then it would have been pumpable by choosing $vy = \mathtt{bc}$, thereby foiling our proof.

8. **Turing Machines [15 points]** Give the state diagram for a Turing Machine that recognizes the following language. To make things simpler, assume it a two-tape, nondeterministic Turing Machine, whose tapes are infinite in both directions. So for input $w \in \{0, 1, \#\}^*$, the first tape is initialized as having infinitely many $\sqcup$'s followed by $w_0 w_1 \dots w_{|w|-1}$ followed by infinitely many $\sqcup$'s. Its tapehead is initialized pointing at w_0. The second tape is initialized to consist entirely of $\sqcup$'s with its tapehead pointing at one of them. For the state diagram, write transitions that look like the following.



Here a, b, c, d are in the tape alphabet Γ and D, E are in $\{L, S, R\}$. The transition correspond to reading an a on the first tape and a c on the second tape, replacing them with a b and a d respectively, and moving the tapeheads in the directions indicated by D and E. (L means left, R means right, and S means stay put.)

(a) $L = \{0^n 1^m 0^n 1^m \mid n, m \in \mathbb{Z}_{>0}\}$. There is a nice answer with 5 states. Note the use of "> 0" so that you do not have to accept the empty string.

The following is a 5 state solution. It first write the leading zeros onto the second tape. Then write the following ones onto the second tape. Then it rewinds the second tape head back to the beginning, before comparing if the rest of the input string matches what was written on the second tape.