CSE 355 Test 2, Fall 2016

28 October 2016, 8:35-9:25 a.m., LSA 191

Last Name	`SAMPLE`	ASU ID	`1357924680`
First Name(s)	`Ima`		

Regrading of Midterms

If you believe that your grade has not been added up correctly, return the entire paper to the instructor with a short note indicating what you believe to be the error.

Other than for that reason, test grades are almost never changed. If you believe that you did not receive the proper credit, first **read these sample solutions carefully** to see if you can understand the answer to your concern. If that does not resolve it, write a clear explanation of why you believe the grade is in error and submit that, along with the entire test paper, to the instructor. Please do not discuss in your explanation how your solution is like that of another student, as FERPA legislation makes it impossible for me to discuss one student's work with another. Please take into account that more than 350 papers were graded, and it is quite unfair to change the grade on one paper without giving every other student the same opportunity. If you nevertheless want the paper regraded, be advised that the entire paper will be regraded and the grade may go up, stay the same, or go down. The new grade will be final.

It is a violation of the Academic Integrity Policy to request a grade change simply because you need or want a higher grade.

If you require a clarification of the sample solutions (not a grade change or review as discussed above), ask in recitations, or in office hours of the TA or instructor. You will be asked whether you have read the sample solution and to indicate what precisely is unclear to you about it, so **read these sample solutions carefully** first. Note that under no circumstances can anyone change a grade other than the instructor, so do not ask the TAs to do so – they are not able to.

Grade change requests, whether submitted as described above or not, will not be considered if received after 24 November 2016.

Instructions

Do not open the exam until you are instructed to do so. There are five sheets of paper, two-sided, containing "Multiple Choice Questions", "Answers to Multiple Choice", and three "Long Answer" questions. *You* **must** *write your name and student number on each and every sheet indicated; failure to do so may result in your test not being properly graded.* Write legibly – we must be able to read your name and number. You must turn in **all sheets** including the multiple choice questions. You have 50 minutes to complete the exam. No books, notes, electronic devices, or other aids are permitted. Turn off all wireless devices and place them away from your work space. Write all answers on the examination paper itself. BUDGET YOUR TIME WELL! SHOW ALL WORK!

Answers to Multiple Choice [14 marks in total]

Enter each response (one of `a, b, c, d, e`) for the questions on the "Multiple Choice" pages. Giving 0, or 2 or more, responses to a question is incorrect. Illegible or blank responses are incorrect.

1 D	2 E	3 B	4 C	5 D	6 E	7 A
8 A	9 C	10 B	11 B	12 A	13 B	14 A
TOTAL						

Multiple Choice [14 marks in total] Select the most appropriate answer for each, and enter each response (one of `a, b, c, d, e`) on the *"Answers to Multiple Choice"* page.

1. A class of languages is closed under subsets if whenever L is in the class and $L' \subseteq L$, L' is also in the class. Among the context-free, regular, and finite languages, the classes that are closed under subsets are:
   d only finite.
2. A derivation of a string w of length n in a context-free grammar
   e can involve any positive integer number of rules.
3. A derivation of a string w of length n in a context-free grammar in Chomsky normal form
   b must involve exactly $2n - 1$ applications of rules, except possibly when $n \leq 5$.
4. A context-free grammar G is *ambiguous* if
   c some string $w \in L(G)$ has at least two different parse trees.
5. To show that a language is context-free, one could give a PDA for it. One could also
   d give a context-free grammar for it or use closure properties
6. To show that a language is **not** context-free, one could
   e use the pumping lemma for context-free languages or use closure properties
7. Whenever each transition of a PDA M does not pop a symbol, the language of M
   a must be regular but need not be finite.
8. Whenever each transition of a PDA M that pushes also pops, the language of M
   a must be regular but need not be finite.
9. Whenever each transition of a PDA M pops a symbol, the language of M
   c must be finite but need not be empty.
10. Whenever each transition of PDA M either pops or pushes, but not both, the language of M
    b must be context-free but need not be regular.
11. In a CFG in CNF with start variable S, which rule could **not** arise?
    b $B \to Bc$.
12. In converting a regular grammar to Chomsky normal form, which step is **not** required?
    a Break up long right hand sides.
13. Context-free languages are closed under
    b union, star, and concatenation but not intersection or complementation.
14. Suppose that $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ is a PDA. Which of the following must be **false**?
    a Q is empty.

Question 1. [12 marks] The *square* of a language L is $\texttt{Sq}(L) = \{ww : w \in L\}$; the *double* of a language is $\texttt{Do}(L) = \{wx : w, x \in L\}$.

(a) [4 marks] If L is regular, must $\texttt{Do}(L)$ be context-free? Justify your answer carefully.

Yes, $Do(L)$ must be context-free. $Do(L) = \{wx : w, x \in L\}$ is just LL. Since L is regular, and regular languages are closed under concatenation, LL is also regular. But since every regular language is context-free, $Do(L)$ is context-free.

(b) [8 marks] If L is regular, must $\texttt{Sq}(L)$ be context-free? Justify your answer carefully.

No, $Sq(L)$ need not be context-free. Let $L = 0^\star 1^\star$, so L is regular. Assume to the contrary that $Sq(L)$ is context-free, and let p be its pumping length p. Choose $s = 0^p1^p0^p1^p$; then $s \in Sq(L)$ and $|s| \geq p$. For each way to write $s = uvxyz$ with $|vxy| \leq p$ and $|vy| \geq 1$, consider $uv^0xy^0z = 0^{p-a}1^{p-b}0^{p-c}1^{p-d}$ for some $a, b, c, d \geq 0$. Because $1 \leq |vxy| \leq p$ we have $1 \leq a + b + c + d \leq p$.

Case 1. $a \neq c$ **or** $b \neq d$**:** Then $0^{p-a}1^{p-b}0^{p-c}1^{p-d}$ is not of the form ww and so is not in $Sq(L)$.

Case 2. $a > 0$**:** Because $|vxy| \leq p$ we have $c = d = 0$ and hence a contradiction because $a = c$ is necessary (see Case 1).

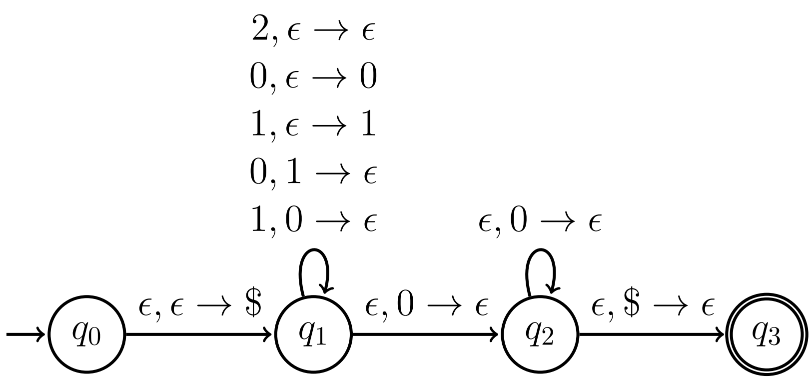
Case 3. $d > 0$**:** Because $|vxy| \leq p$ we have $a = b = 0$ and hence a contradiction because $b = d$ is necessary. (see Case 1).

No matter how s is written as $uvxyz$, $uv^0xy^0z \notin Sq(L)$, and hence $Sq(L)$ is not context-free.

For a different solution see Question 1 in Quiz 9 sample solutions.

Question 2. [12 marks] Let $L_1 = \{w \in \{0, 1, 2\}^\star : w$ contains more 0s than 1s$\}$. Let $L_2 = \{w \in \{0, 1, 2\}^\star : w$ has the same number of 1s and 2s$\}$.

(a) [3 marks] Give a transition diagram for a PDA to recognize L_1.



Explanation: On reading a 0, we can push a 0 or pop a 1, while on reading a 1, we can push a 1 or pop a 0. This results in the number of 0's minus the number of 1's on the stack matching the number of 0's minus the number of 1's in the input. We want to have at least one more 0 than 1, so we can only transition to the final state if there is a 0 on the top of the stack. If, on seeing 0, we always push 0 when there is a 0 on top of the stack and pop 1 when there is a 1 (and vice versa when seeing 1), we will end up with the stack containing only 0's or only 1's. So if there are more 0's than 1's, we can remove all the 0's on the stack and reach its bottom. Using other transitions may make fail to accept a valid string but will not accept an invalid string. So the PDA correctly decides L_1.

(b) [3 marks] Give a CFG to generate L_2.

$$S \to \epsilon \mid S0S \mid S1S2S \mid S2S1S$$

The empty string is in L_2. We can insert a 0 in between two valid strings to obtain another valid string. The last two rules are valid because in each, a 1 is paired with a 2, and the other parts have a balanced number of 1's and 2's by induction.

Explanation: To show that all strings in L_2 are covered by this grammar, consider such a string w. If w consists entirely of 0's, it is covered by the first two rules. Otherwise, there is some first 1 or 2 in w. Suppose it is a 1; the case for it being a 2 can be obtained by exchanging 1 and 2 in the following. We search forward until we find a 2 after having passed an equal number of intervening 1's and 2's, and then apply the $S \to S1S2S$ rule, after which we have three pieces with an equal number of 1's and 2's, so can use the three S's on the right-hand side for them. It is always possible to find a matching 2: Otherwise, we reach the end of the string with more 1's than 2's, contradicting the assumption that $w \in L_2$.

(c) [6 marks] State whether or not $L_1 \cap L_2$ is context-free. Justify your answer carefully.

No, it is not context-free. Using the notation $|w|_x$ for "the number of x's appearing in w,"

$$L_1 \cap L_2 = \{w \in \{0,1,2\}^* \mid |w|_0 > |w|_1 = |w|_2\}$$

Assume (to the contrary) that $L_1 \cap L_2$ is context-free, and has pumping length p. Choose $w = 0^{p+1}1^p2^p$ to pump. There are two cases to consider for $w = uvxyz$.

vy **contains no 2:** Then vy contains at least one 0 or 1 because $|vy| \geq 1$. But $uv^0xy^0z = uxz$ has at most as many 0's or fewer 1's (or both) than 2's, so the result is not in $L_1 \cap L_2$.

vy **does contain a 2:** Then it cannot contain a 0, because $|vxy| \leq p$. So uv^2xy^2z has at least as many 2's as 0's, again yielding a string not in $L_1 \cap L_2$.

So the string w cannot be pumped, a contradiction. So $L_1 \cap L_2$ is not context-free.

Because context-free languages are not closed under intersection, we have to check the specific language $L_1 \cap L_2$. It may happen that L, L', and $L \cap L'$ are all CFLs, so you cannot just say that because we do *not* have closure under intersection, we do not need to check!

Question 3. [12 marks] Consider the context-free grammar G with variables $\{S, A, B\}$, terminals $\{a, b\}$, and rules

1. $S \to A \mid B$
2. $A \to aAa \mid Aa \mid B \mid \varepsilon$
3. $B \to aBb \mid Bb \mid B \mid \varepsilon$

(a) [6 marks] Using grammar G *and using the methods from class*, form an equivalent grammar G' in which (1) the start variable does appear on the RHS of a rule, (2) there are no unit rules, and (3) the only ε-rule has the start variable on the LHS.

I assume that S is the start variable, **not** on the RHS when we start. Eliminate ε-rules:

1. $S \to A \mid B \mid \varepsilon$
2. $A \to aa \mid a \mid aAa \mid Aa \mid B$
3. $B \to ab \mid b \mid aBb \mid Bb \mid B$

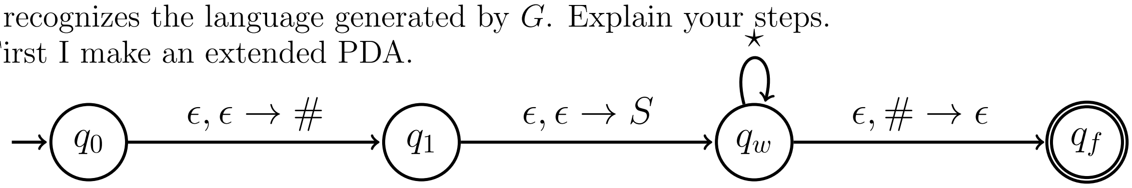
Now eliminate unit rules:

1. $S \to aa \mid a \mid ab \mid b \mid aAa \mid Aa \mid aBb \mid Bb \mid \varepsilon$
2. $A \to aa \mid a \mid ab \mid b \mid aAa \mid Aa \mid aBb \mid Bb$
3. $B \to ab \mid b \mid aBb \mid Bb$

Now **if you <u>state</u> the assumption that the question meant to say "does *not* appear" instead of "does appear"**, we are done. To do the question as stated, add a variable T and the rule $S \to TS$.

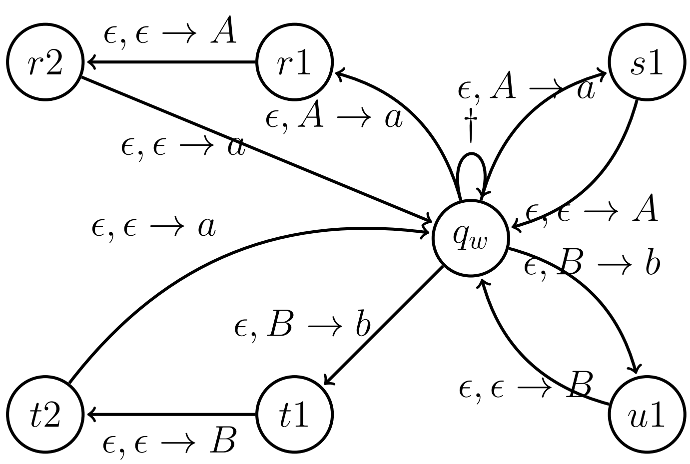
(b) [6 marks] Using the method described in class, show a transition diagram for a PDA that recognizes the language generated by G. Explain your steps.

First I make an extended PDA.



There are loops at $\star$ labeled $a, a \to \epsilon$; $b, b \to \epsilon$; $\epsilon, S \to A$; $\epsilon, S \to B$; $\epsilon, A \to B$; $\epsilon, A \to Aa$; $\epsilon, A \to aAa$; ; $\epsilon, A \to \epsilon$; $\epsilon, B \to \epsilon$; $\epsilon, B \to B$; $\epsilon, B \to Bb$; and $\epsilon, B \to aBb$.

Now I break up multiple pushes, replacing the loop transitions at q_w with



There are loops at $\dagger$ labeled $a, a \to \epsilon$; $b, b \to \epsilon$; $\epsilon, S \to A$; $\epsilon, S \to B$; $\epsilon, A \to B$; $\epsilon, A \to \epsilon$; $\epsilon, B \to \epsilon$; and $\epsilon, B \to B$.