Homework 6: Context Doesn't Always Matter

CSCI 161

Due 9pm, Thursday, October 29, 2020

Ben Dreyer

Overview. This week's homework concerns the equivalence of pushdown automata and context-free grammars. It also touches on parse trees (as generated by the CYK algorithm when checking whether a string is in a grammar) and context-sensitivity (using the CFL pumping lemma to prove a language is not a CFL). You are welcome to use JFLAP to generate your diagrams, but take screenshots of these diagrams and embed them in your pdf or other main submission rather than submitting jff files. For 5% extra credit, edit this tex file with your solutions and upload your source code and your compiled pdf, which should include all diagrams.

Question 1. The following CFG in Chomsky normal form generates strings of balanced parentheses:

$$V_1 \to V_2V_2 \mid V_4V_3 \mid V_4V_5 \mid \epsilon$$
$$V_2 \to V_2V_2 \mid V_4V_3 \mid V_4V_5$$
$$V_3 \to V_2V_5$$
$$V_4 \to ($$
$$V_5 \to)$$

1. Recall that CYK running on a grammar with r variables and an input string w of length n populates an $n \times n \times r$ array such that $P[\ell, s, i]$ is true if and only if variable R_i derives $w_s \dots w_{s+\ell-1}$, the input substring of length ℓ starting at index s. For the grammar above and input $w = (())(()())$, give a true false answer describing the contents of each of the following cells:

   $P[1,4,5] =$ True

   $P[1,5,3] =$ False

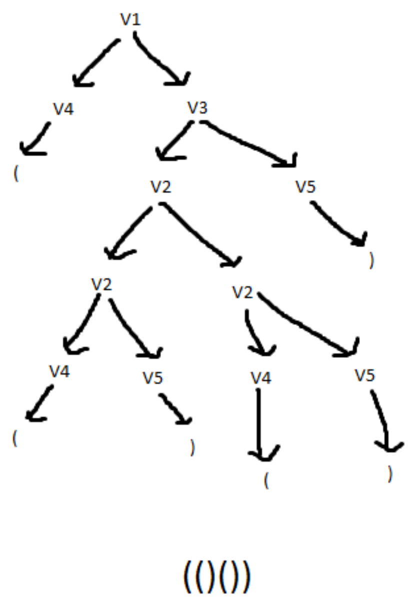
   $P[5,2,1] =$ False

   $P[9,2,3] =$ False

   Hint: Note that you do *not* need to run the whole algorithm to infer the subproblem answers.

2. Draw the parse tree and corresponding leftmost derivation for $(()())$.

   Hint: The unambiguous derivation is 11 steps.



Solution $V1 \to V4V3 \to (V3 \to (V2V5 \to (V2V2V5 \to (V4V5V2V5 \to ((V5V2V5 \to (()V2V5 \to (()V4V5V5 \to (()(V5V5 \to (()()V5 \to (()())$

Question 2. This question explores the context-free language of valid unary addition equations. Specifically, it concerns the following language over the alphabet $\Sigma = \{1, +, =\}$:

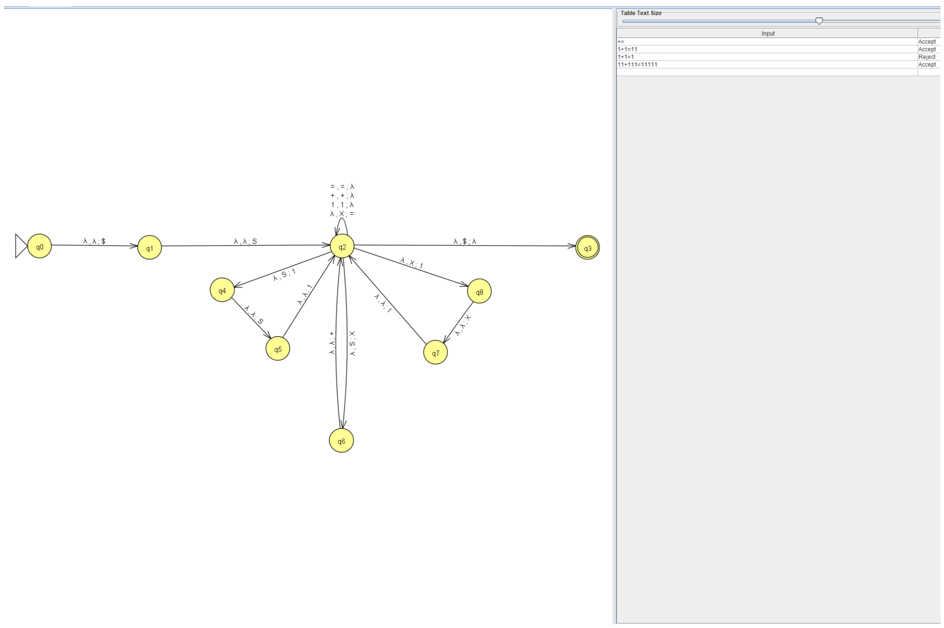
$$L = \{a + b = c \mid a, b, c \in 1^*, |a| + |b| = |c|\}.$$

Note that $|a|$ means the length of a, and since a is all 1s, this is just the numerical value of a. For example, the string $111 + 11111 = 11111111$ is in L, whereas $1 + 11 = 11$ is not. The value zero is represented by the empty string, since $|\epsilon| = 0$. So $+ =$, $11+ = 11$, and $+1 = 1$ are all valid strings in the language.

a) Give a CFG for L. Part b) will be easier if you have a small grammar.

$S \to 1S1| + X$
$x \to 1X1| =$

b) Give a PDA for L with the structure of Figure 2.24 illustrating the general CFG→PDA conversion.



Extra practice (not extra credit): Design a PDA for the language directly, thinking about how to use the stack to keep track of the number of 1s seen so far with state transitions to keep track of which special characters have been seen. An intuitive solution will be given for your reference.

Question 3. Consider the language over $\Sigma = \{a, b, c\}$ that is the concatenation of arbitrary palindromes of length at least 3 and occurrences of the substring abc:

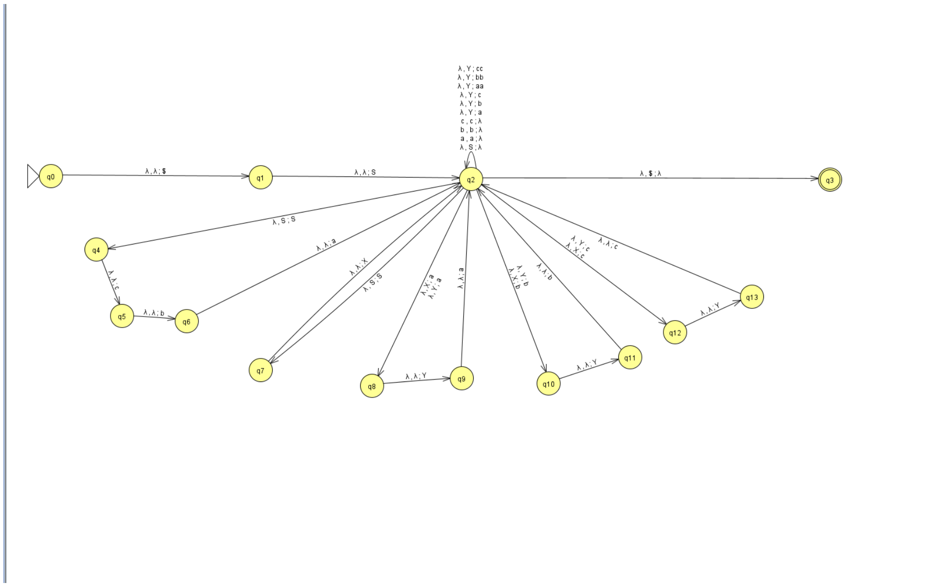
$$L = \left\{x_1 x_2 \ldots x_k \mid k \geq 0, x_i = abc \text{ or } x_i = x_i^{\mathcal{R}}, |x_i| \geq 3 \text{ for } i = 1, \ldots, k\right\}.$$

a) Give a CFG for L.

S $\rightarrow abcS \mid XS \mid \epsilon$
X $\rightarrow aYa \mid bYb \mid cYc \mid \epsilon$
Y $\rightarrow aYa \mid bYb \mid cYc \mid a \mid b \mid c \mid aa \mid bb \mid cc$

b) Give a PDA for L. Although the generic CFG$\rightarrow$PDA conversion will give you a correct answer, for full credit you should design a PDA directly using no more than 12 states. (My solution uses 8.)

(I couldn't figure out how to get it down to 12 states)



Question 4. Consider the following language over $\Sigma = \{a, b, c\}$:

$$L = \left\{a^{2n}b^{n}c^{2n} \mid n \geq 0\right\}.$$

The following PDA P does *not* recognize L:

a) What is $L(P)$ instead? Give a precise description.

   Hint: It accepts every string in L but also some strings not in L.

   Solution

   $$L(P) = \left\{a^{n}b^{m}c^{2(n-m)} \mid n \geq 0, m \geq 0, m \leq n\right\}.$$

   The language of a's b'c and c's with any amount of a's, the same or less amount of b's and the number of c's is equal to double the amount of the number a's minus the number of b's

b) Use the CFL pumping lemma to prove that no PDA recognizes L.

   Suppose L is a context-free with a pumping length p.
   Then $s = a^{p}b^{p/2}c^{p}$
   For any decomposition s = uvxyz with length (vxy) $\leq$ p , length (v) ≥ 0 and length (y) ≥ 0
   Case 1: vxy contains only a's and b's. If you pump down, you would have less a's than c's, which would cause s to be out of the language.
   Case 2: vxy contains only b's. If you pump up, you would have more b's than half of the number of a's and c's, which would cause s to be out of the language.

   Case 3: vxy contains only b's and c's. If you pump down, you would have less c's than a's, which would cause s to be out of the language.

   These are the only possible cases for any decomposition of s, therefore s is not in the language and L is context-sensitive.