HW4

Due: October 6, 2016

Attach a fully filled-in cover sheet to the front of your printed homework. Your name should not appear anywhere except on the cover sheet; each individual page of the homework should include your Banner ID instead.

While collaboration is encouraged in this class, please remember not to take away notes from any collaboration sessions. Also please list the names and logins of any of your collaborators at the beginning of each homework.

Please monitor Piazza, as we will post clarifications of questions there. You should hand in your solutions by 12:55 to the CSCI 1010 bin on the second floor of the CIT. Late homeworks are not accepted.

The following is a warmup question, which will not be graded:

For each of the languages below, give a CFG that generates the language.

a. \{xy \mid x, y \text{ are palindromes over the alphabet } \Sigma = \{0, 1\}\}
   For the purposes of this problem, \(\varepsilon\) is a palindrome.

b. \{01^{n+1}0^n1^m \mid m, n \geq 0\}

c. \{a^i b^j c^k \mid i = j \text{ or } j = k \text{ where } i, j, k \geq 0\}

The following questions are lab problems.

Lab Problem 1

Rudolph and Ginger are taking a break from competitive farming to vote in Myami’s upcoming midterm elections. This year, an ambitious turmeric farmer is trying to upset the socio-agricultural arena by running under the campaign slogan, “Make Turmerica Great Again”. His name is Walter Chestnutt.

Walter knows that parsnip farmers are some of the most important swing voters, and they are notoriously picky about proper syntax. In an effort to pander to the parse-tree-loving parsnip farmers, he needs to ensure that all of his pamphlets are syntactically correct and preferably unambiguous.
Let $\Sigma = \{a, +, \times, (, )\}$. A string $w \in \Sigma^*$ is called \textit{syntactically correct} if it forms a valid arithmetic expression. Walter comes up with the following CFG $G$ that recognizes the language $L$:

$$S \rightarrow S + S \mid S \times S \mid (S) \mid a$$

a. Prove that $L$ is not regular.

b. Show that $G$ is ambiguous, i.e. that the same string can yield two distinct parse trees.

c. Walter comes up with another CFG $H$ that is also ambiguous:

$$S \rightarrow \text{if } C \text{ then } S \mid \text{if } C \text{ then } S \text{ else } S \mid s$$
$$C \rightarrow c1 \mid c2 \mid c3$$

Help Walter come up with an unambiguous grammar $H'$ such that $L(H) = L(H')$.

d. Sketch a proof that $H'$ is unambiguous.

\textbf{Lab Problem 2}

In this problem, we will show that the language $L = \{a^ib^jc^k \mid i = j \text{ or } j = k\}$ is inherently ambiguous. Although we don’t use the pumping lemma for context-free languages directly, its proof is helpful for doing this problem.

a. Show that $L$ is context-free.

b. Let $G$ be any context-free grammar for $L$, and let $n$ be the number of variables in $G$, and let $d$ be the maximum number of symbols on the right hand side of a rule in $G$. Let $p = d^n + 1$. Let $\tau$ be a minimal parse tree for the string $s = a^pb^pc^m$ for $m = p! + p$. Argue that there is a leaf of the parse tree, corresponding to position $i$ in the string, where $1 \leq i \leq p$, such that the depth of this leaf is at least $n + 1$. That is, there is a leaf labeled ‘a’ with a path of length at least $n + 1$ to the root. The root is a \textit{parsnip}, so called because it is the root of a parse tree. Refer to the picture at the end of this problem as a guide.
c. Argue that it follows from part (b) that there must exist a variable $A$ in $G$ such that $A \Rightarrow^* vAy$ for $v = a^j$, $y = b^j$ for some $j > 0$, and that this variable $A$ must appear somewhere in $\tau$. (Recall that the notation $x \Rightarrow^* y$ notation means that either $x = y$ or there is a sequence $x_1, \ldots, x_k$ such that $x \Rightarrow x_1 \Rightarrow x_2 \ldots \Rightarrow x_k \Rightarrow y$.)

d. Using $\tau$ and the fact that it can be “pumped” using $A \Rightarrow^* vAy$, prove that there exists a parse tree $\sigma$ for the string $a^mb^mc^m$ that contains the variable $A$.

e. Let $\tau'$ be a minimal parse tree for the string $s' = a^{m'b^pc^p}$. Analogously to parts (b), (c) and (d), show that it follows that there must exist a variable $C$ in $G$ such that $C \Rightarrow^* uCw$ for $u = b^\ell$, $w = c^\ell$ for some $\ell > 0$, and that this variable $C$ must appear somewhere in $\tau'$. Using $\tau'$ and the fact that it can be “pumped” using $C \Rightarrow^* uCw$, prove that there exists a parse tree $\sigma'$ for the string $a^mb^mc^m$ that contains the variable $C$.

f. Show that $\sigma$ and $\sigma'$ are distinct parse trees for the same string. Hint: Can $\tau$ contain $C$?

---

Lab Problem 3

Walter Chestnutt’s “Turmerican Revolution” is unexpectedly taking root in Myami’s electoral landscape. Though he is new to campaigning, Walter has
always had political aspirations. Working at Yammany Hall—the pinnacle of Myami politics—has been his dream for as long as he can remember. As the votes roll in for the midterm election, he wants to determine whether they have been correctly counted. Unfortunately, even though Walter has heavily studied CFGs, Myami’s votes are traditionally counted in unary.

A positive integer $n$ can be encoded using the unary alphabet $\{1\}$, by the string $1^n$. For example, 1 represents one, and 111 represents three.

The language of true addition equalities $L$ is a string over the alphabet $\Sigma = \{1, =, +\}$ containing exactly one ‘$=$’ symbol such that the sum of the integers to the left of ‘$=$’ is equal to that to the right. For example, the following are true addition equalities, and represent correct vote counts:

- $11 + 111 = 11111$
- $11111 = 11 + 11$
- $11 + 11 + 1111 = 111 + 111 + 11$
- $11 = 11$

The following are not true addition equalities, and may be signs of voter fraud:

- $11 + 111 = 111111$ (because $2 + 3 \neq 6$)
- $11 + 111 = 11111 = 1 + 1111$, because there must be only one $=$ symbol.
- $11111$, because there must be at least one $=$ symbol.
- $1 + 0 = 1$, because 0 is not in the alphabet.
- $1+ = 1$, because the expression $1+$ does not have a well-defined value.
- $=,$, because there must be expressions evaluating to numbers on both sides of the $=$.

In this problem, you will give a CFG for the language of true addition equalities, so that Walter can verify the correctness of the midterm vote counts.

a. Prove that $L$ is not regular.

b. Design a CFG for $L.$