Homework 5
Due Feb 15, 2012

Prof. Amit Chakrabarti
Computer Science Department
Dartmouth College

Please think carefully about how you are going to organise your answers *before* you begin writing. Make sure your answers are complete, clean, concise and rigorous.

1. We proved in class that the language $L = \{ww : w \in \{0,1\}^*\}$ is not context-free. Its complement is

$$\overline{L} \;=\; \{x \in \{0,1\}^* : x \text{ is not of the form } ww \text{ for any } w \in \{0,1\}^*\}\,.$$

Prove that every even-length string $x \in \overline{L}$ can be decomposed as $x = uv$ where the middle symbol of u differs from the middle symbol of v. Using this property, design a context-free grammar that generates $\overline{L}$.

What can you conclude about the closure of context-free languages under complementation? [10 points]

2. Consider the following CFG:

$$S \longrightarrow 1S00 \mid 00S1 \mid SS \mid 0S1S0 \mid \varepsilon$$

2.1. Give a simple description of the language it generates, in the form $\{x \in \{0,1\}^* : \langle\text{some simple property of } x\rangle\}$. The simple property might be expressed as an equation or a sentence or some combination of the two. [5 points]

2.2. Prove the correctness of your answer. You may want to mimic the style of the proof we did in class for a closely related language. [10 points]

3. For each of the following languages, say whether or not it is a CFL and prove your answer, either by designing an appropriate CFG or PDA or by using closure properties and/or the pumping lemma. If designing a CFG/PDA, please explain your construction in brief so the grader can understand your design.

3.1. $\{a^n b^n c^m : n \le m \le 2n\}$. [8 points]

3.2. $\{a^n b^{n^2} : n \ge 0\}$. [8 points]

3.3. $\{x_1\#x_2\#\cdots\#x_k : k \ge 1$, each $x_i \in \{a,b\}^*$, and for some i and j (possibly equal), $x_i = x_j^{\mathcal{R}}\}$. Note that this language is over the alphabet $\{a,b,\#\}$. [10 points]

3.4. $\{b_i\#b_{i+1} : i \ge 1\}$, where b_i is the binary representation of the integer i with no leading 0's (e.g. $b_5 = 101, b_{18} = 10010$). Note that this language is over the alphabet $\{0,1,\#\}$. [12 points]

3.5. $(a \cup b)^* - \{(a^n b^n)^n : n \ge 1\}$. [12 points]

4. Define the *length* of a rule in a CFG to be the number of characters required to write down the rule. Thus, if $G = (V,\Sigma,R,S)$ is a CFG and "$A \to w$" is a rule in R with $w \in (V \cup \Sigma)^*$, then the length of this rule is $2 + |w|$ if $w \ne \varepsilon$ and 3 if $w = \varepsilon$ (because we do have to write one character to represent ε, even though $|\varepsilon| = 0$). Define the *complexity* of a CFG to be the sum of the lengths of all the rules in the CFG.

Note that things like "$A \to w \mid x \mid y$" are not rules; they are convenient notation for, in this case, three separate rules "$A \to w$", "$A \to x$", and "$A \to y$", so the lengths of these three rules must be figured separately and added up.

4.1. Suppose the PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ is in normal form as discussed in class (i.e., each move is either a one-character push or a pop, but not both; there is only one accept state; and the stack is empty upon acceptance). Suppose we use the procedure of Sipser's Lemma 2.27 to convert it into a CFG G. Give the best possible asymptotic upper bound on the complexity of G, in terms of $|Q|$. Assume $|\Sigma|$ and $|\Gamma|$ are constants.

By "asymptotic" I mean that you don't need to give an exact bound, and should use big-O notation to simplify, where possible. You should prove why your bound always holds. Make sure you assume nothing about M: your upper bound must hold even for the most outlandish of PDAs. [5 points]

4.2. We proved that every regular language is context-free as follows: a regular language is recognized by a DFA, which is automatically a PDA (that ignores its stack), which is equivalent to some CFG. However, having solved the previous problem, you know that if we start with a DFA with n states, the complexity of the CFG that results by following this proof may be ***huge*** (in terms of n, assuming a constant-sized alphabet).

Come to the rescue by proving that any n-state DFA (over a constant-sized alphabet) can be converted into an equivalent CFG whose complexity is only $O(n)$.

Hint: Prove that any DFA can be converted into a CFG where every rule is either of the form "$A \to aB$", or of the form "$A \to \varepsilon$", where A, B are variables and a is a terminal. What would be a natural choice for the set of variables? [10 points]

5. Do problem 2.27 from your textbook (Sipser). [10 points]

Challenge Problems

CP5: A deterministic pushdown automaton (DPDA) is just like a PDA, except that its transition function must be deterministic. Formally, it is a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$ that satisfies all of the conditions for a PDA and the following three additional conditions:

$$\begin{aligned} |\delta(q,a,s)| &\le 1, && \forall\, q \in Q, a \in \Sigma_\varepsilon, s \in \Gamma_\varepsilon\,. \\ |\delta(q,\varepsilon,s)| = 1 &\Rightarrow |\delta(q,a,s)| = 0, && \forall\, q \in Q, a \in \Sigma, s \in \Gamma_\varepsilon\,. \\ |\delta(q,a,\varepsilon)| = 1 &\Rightarrow |\delta(q,a,s)| = 0, && \forall\, q \in Q, a \in \Sigma_\varepsilon, s \in \Gamma\,. \end{aligned}$$

Prove that there does not exist a DPDA that recognizes the context-free language $\{0^n1^n : n \ge 0\} \cup \{0^n1^{2n} : n \ge 0\}$.

CP6: The *stack language* of a PDA is defined to be the set of all strings that can occur on its stack during a valid execution on some input, starting from the start state (recall how the contents of the stack are formally defined to be a string). Therefore, this is a language over the stack alphabet of the PDA.

Prove that, for every PDA, its stack language is regular.