CS 373: Theory of Computation Sariel Har-Peled and Madhusudan Parthasarathy

## Problem Set 5

Due: Thursday, March 12, 2009 at 12:30 in class (i.e., SC 1105)
Version: $\mathbf{1 . 0}$

Submission instructions: Submit each problem on a separate sheet of paper, put your name on each sheet, and write your discussion section time and day (e.g. Tuesday 10am) in the upper righthand corner. These details may sound picky, but they make the huge pile of homeworks much easier to grade quickly and more importantly, since we return them in the discussion sections, easier for you to get them back.

Also, write on each exercise the name/netid of your group members.
(Q1) What is in, what is out?
[Category: Understanding., Points: 10]
For each of the following grammars answer the following:
(a) Is abcd in the language of the grammar? If so give an accompanying derivation and parse tree.
(b) Is acaada in the language of the grammar, if so give an accompanying derivation and parse tree.
(c) What is the language generated by the grammar and explain your answer.
(Note: assume $\Sigma=\{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}\}$ and the start symbol is S for both grammars.)

$\mathcal{G}_{1}: \quad$| $\mathrm{S} \rightarrow \mathrm{aSd}\|\mathrm{A}\| \mathrm{C}$ |
| :--- |
| $\mathrm{A} \rightarrow \mathrm{aAc} \mid \mathrm{B}$ |
| $\mathrm{B} \rightarrow \mathrm{bBc} \mid \epsilon$ |
| $\mathrm{C} \rightarrow \mathrm{bCd} \mid \mathrm{B}$ |

$$
\begin{array}{ll} 
& \mathrm{S} \rightarrow \mathrm{~B} \mid \mathrm{AA} \\
\mathcal{G}_{2}: & \mathrm{A} \rightarrow \mathrm{cA} \mid \mathrm{dB} \\
& \mathrm{~B} \rightarrow \mathrm{aSa} \mid \epsilon .
\end{array}
$$

(Q2) Building grammars.
[Category: Construction., Points: 10]
Show that the following languages are context-free by giving a context-free grammar for each of them.
(a) $L=\left\{\mathrm{a}^{i} \mathrm{~b}^{j} \mid 2 i \leq j \leq 3 i, \quad i, j \in N\right\}$.
(Hint: Build a grammar for the case that $j=2 i$ and the case $j=3 i$, and think how to fuse these two grammars together to a single grammar.)
(b) $L^{\prime}=\left\{\begin{array}{l|l}w \in\{0,1\}^{*} & \begin{array}{l}\mathrm{w} \text { contains equal number of occurrences of } \\ \text { substring } 01 \text { and } 10\end{array}\end{array}\right\}$.

Thus 101 contains a single 01 and a single 10 and as such belongs to the language, while 1010 does not as it contains two 10 's and one 01.
(Q3) Prove this.
[Category: Proof, Points: 10]

Consider the following proof.

Lemma 0.1 If $L_{1}$ and $L_{2}$ are both context-free languages, then $L_{1} \cup L_{2}$ is a context-free language.

Proof: Let $\mathcal{G}_{1}=\left(\mathcal{V}_{1}, \Sigma, \mathcal{R}_{1}, \mathrm{~S}_{1}\right)$ and $\mathcal{G}_{2}=\left(\mathcal{V}_{2}, \Sigma, \mathcal{R}_{2}, \mathrm{~S}_{2}\right)$ be context free grammars for $L_{1}$ and $L_{2}$, respectively, where $\mathcal{V}_{1} \cap \mathcal{V}_{2}=\emptyset$. Create a new grammar

$$
\mathcal{G}=\left(\mathcal{V}_{1} \cup \mathcal{V}_{2}, \Sigma, \mathcal{R}, \mathrm{~S}\right)
$$

where $\mathrm{S} \notin \mathcal{V}_{1} \cup \mathcal{V}_{2}$ and $\mathcal{R}=\mathcal{R}_{1} \cup \mathcal{R}_{2} \cup\left\{\mathrm{~S} \rightarrow \mathrm{~S}_{1}, \mathrm{~S} \rightarrow \mathrm{~S}_{2}\right\}$.
We next prove that $L(\mathcal{G})=L_{1} \cup L_{2}$.
$\mathbf{L}(\mathcal{G}) \subseteq \mathbf{L}_{\mathbf{1}} \cup \mathbf{L}_{\mathbf{2}}:$
Consider any $w \in L(\mathcal{G})$, and any derivation of $w$ by $\mathcal{G}$. It must be of the following form:

$$
\mathrm{S} \rightarrow \mathrm{~S}_{i} \rightarrow X_{1} X_{2} \rightarrow \ldots \rightarrow w
$$

where $i$ is either 1 or 2 . Assume, without loss of generality, that $i=1$, and observe that if we remove the first step, this derivation becomes

$$
\mathrm{S}_{1} \rightarrow X_{1} X_{2} \rightarrow \ldots \rightarrow w
$$

Namely, $\mathrm{S}_{1} \stackrel{*}{\Longrightarrow} w$ using grammar rules only from $\mathcal{R}_{1}$. We conclude that $w \in L\left(\mathcal{G}_{1}\right)=$ $L_{1}$, as $\mathrm{S}_{1}$ is the start symbol of $\mathcal{G}_{1}$.
The case $i=2$ is handled in a similar fashion.
Thus, we conclude that $w \in L_{1} \cup L_{2}$, implying that $L(\mathcal{G}) \subseteq L_{1} \cup L_{2}$.
$\mathbf{L}_{\mathbf{1}} \cup \mathbf{L}_{\mathbf{2}} \subseteq \mathbf{L}(\mathcal{G}) \mathbf{:}$
Consider any word $w \in L_{1} \cup L_{2}$, and assume without limiting generality, that $w \in L_{1}$. As such, we have that $\mathrm{S}_{1} \underset{\mathcal{G}_{1}}{*} w$. But $\mathrm{S} \rightarrow \mathrm{S}_{1}$ is a rule in $\mathcal{G}$, and as such we have that

$$
\mathrm{S} \rightarrow \mathrm{~S}_{1} \xrightarrow[\mathcal{G}_{1}]{\stackrel{*}{\Longrightarrow}} w .
$$

Namely, $\mathrm{S} \underset{\mathcal{G}}{*} w$, since all the rules of $\mathcal{G}_{1}$ are in $\mathcal{G}$. We conclude that $w \in L(\mathcal{G})$.
Putting the above together, implies the lemma.
Provide a detailed formal proof to the following claim, similar in spirit and structure to the above proof.

Lemma 0.2 If $L_{!}$and $L_{2}$ are both context-free languages, then $L_{1} L_{2}$ is a context-free language.
(Q4) Edit with some mistakes.
[Category: Construction., Points: 10]

The edit distance between two strings $w$ and $w^{\prime}$, is the minimal number of edit operations one has to do to modify $w$ into $w^{\prime}$. We denote this distance between two strings $x$ and $y$ by EditDist $(x, y)$. We allow the following edit operations: (i) insert a character, (ii) delete a character, and (iii) replace a character by a different character.

For example, the edit distance between shalom and halo is 2 . The edit distance between har-peled and sharp_eyed is 4:

$$
\text { har-peled } \Longrightarrow \text { shar-peled } \Longrightarrow \text { sharpeled } \Longrightarrow \text { sharp_eled } \Longrightarrow \text { sharp_eyed. }
$$

For the sake of simplicity, assume that $\Sigma=\{\mathrm{a}, \mathrm{b}, \$\}$. For a parameter $k$, describe a CFG for the language

$$
L_{k}=\left\{x \$ y^{R} \mid x, y \in\{\mathrm{a}, \mathrm{~b}\}^{*}, \operatorname{EditDist}(x, y) \leq k\right\}
$$

For example, since EditDist(aba, bab) $=2$, we have that aba $\$ \mathrm{bab} \in L_{2}$, but aba $\$ \mathrm{bab} \notin L_{1}$. Similarly, EditDist(aaaa, abb) $=3$, and as such aaaa $\$$ bba $\in L_{3}$, but aaaa $\$$ bba $\notin L_{2}$.
(Hint: What is the language $L_{0}$ ? Try to give a grammar to $L_{1}$ before solving the general case.)

Provide a short argument why your CFG works.
(Q5) Speedup theorem for CFGs.
[Category: Proof., Points: 10]
Assume you are given a CFG $\mathcal{G}=(\mathcal{V}, \Sigma, \mathcal{R}, \mathrm{S})$, such that any word $w \in L(\mathcal{G})$ has a derivation of $w$ with $f(n)$ steps, where $n=|w|$. Here $f(n)$ is some function.
Prove the following claim.
Claim 0.3 There exists a grammar $\mathcal{G}^{\prime}$ such that $L(\mathcal{G})=L\left(\mathcal{G}^{\prime}\right)$, and furthermore, any derivation of a word $w \in L(\mathcal{G})$ of length $n$ requires at most $\lceil f(n) / 2\rceil$ derivation steps.

For example, consider the grammar $\mathrm{S} \rightarrow \mathrm{aSb} \mid \epsilon$.
A word $\mathrm{a}^{n} \mathrm{~b}^{n}$ can be derived using $n+1$ steps. Its speeded up version is

$$
\mathrm{S}^{\prime} \rightarrow \mathrm{aaS}^{\prime} \mathrm{bb}|\mathrm{ab}| \epsilon .
$$

Now, $\mathrm{a}^{n} \mathrm{~b}^{n}$ has derivation of length $\lceil(n+1) / 2\rceil$ by the grammar of $\mathrm{S}^{\prime}$.
(Hint: Consider a parse tree of height $h$ in the grammar, and think how to modify the grammar, so that you get a parse tree for the same word with height only 「h/2ך.)

