5DV037 Fundamentals of Computer Science Fall 2010
Obligatory Exercise 4
Due date: October 18, 2010 at 8am (0800)

1. Let $L = \{a^{k_1}b^{k_2}a^{k_3}b^{k_4} \mid k_1 < k_2 < k_3 < k_4\}$.

Under the assumption that this language is regular, let N be the constant guaranteed by the pumping lemma for regular languages, and let $\alpha = a^N b^{N+1} a^{N+2} b^{N+3}$. Clearly $\alpha \in L$.

(a) Identify all decompositions of α of the form $\alpha_1 \cdot \alpha_2 \cdot \alpha_3$ (or, alternatively, using the symbolism of the textbook, of the form $x \cdot y \cdot z$) which satisfy the conditions imposed by the pumping lemma for regular languages.

(b) Using the pumping lemma for regular languages and the decompositions found for part (a), show that L cannot be a regular language.

2. Again let $L = \{a^{k_1}b^{k_2}a^{k_3}b^{k_4} \mid k_1 < k_2 < k_3 < k_4\}$.

Under the assumption that this language is context free, let N be the constant guaranteed by the pumping lemma for context-free languages, and let $\alpha = a^N b^{N+1} a^{N+2} b^{N+3}$. Clearly $\alpha \in L$.

(a) Identify all decompositions of α of the form $\alpha_1 \cdot \alpha_2 \cdot \alpha_3 \cdot \alpha_4 \cdot \alpha_5$ (or, alternatively, using the symbolism of the textbook, of the form $u \cdot v \cdot x \cdot y \cdot z$) which satisfy the conditions imposed by the pumping lemma for context-free languages.
(Hint: There is total of seven main forms for such decompositions.)

(b) Using the pumping lemma for context-free languages and the decompositions found for part (a), show that L cannot be a context-free language.

Problem 3 is found on the next page.

5DV037, Obligatory Exercise 4, page 2

3. In the course slides, it is argued that C cannot be a context-free language. However, the argument rests upon the requirement that arbitrarily long identifiers be allowed. The purpose of this exercise is to show that this requirement is necessary.

For simplicity, consider only C programs which consist of a single declaration of integer variables followed by assignment of zero to these variables. Furthermore, assume that identifiers consist of a single lowercase letter. Examples include:

```
main(){int a,b;a=0;b=0}
main(){int a,b;b=0}
main(){int a,b;b=0;b=0;b=0}
main(){int a,b,c;a=0;c=0;a=0}
```

All identifiers which are used in an assignment statement must be declared, but declared identifiers need not be used in assignment statements. There is no restriction on how many times the same variable may be assigned a value, even the same value. The restriction is only that variables which are used in assignment statements must be declared.

(a) Show that this fragment of C is context free by sketching a context-free grammar which generates it.
(Hint: Take the union of 2^{26} grammars, one for each subset of possible identifiers.)

(b) Write out fully and formally the grammar obtained if only two identifiers, `a` and `b`, are allowed, instead of the full 26.

Further Notes:

1. As stipulated in the course syllabus, this exercise may be done either individually, in a group of two, or in a group of three. Remember that there are point penalties for late submission. See the course syllabus.
2. It is strongly recommended that you use a graphical tool to display your results. If you draw them by hand, they must be very neat. It is not allowed to copy the work of others. The submission must be the original work of the individual(s) in the working group. The grader reserves the right to interview members of the working group about the solution.
3. So that solutions may be discussed in a class meeting, students and/or groups that are very late in preparing a solution may be required to solve an alternate problem to receive credit for this exercise.