CS331 Autumn 2002 End-term Examination

Time: 3 hrs Total marks: 100

- ***The exam is open book and notes.***
- ***Results/proofs covered in class/problem sessions/assignments may simply be cited, unless specifically asked for.***
- ***Unnecessarily lengthy solutions will be penalized.***
- ***If you need to make any assumptions, state them clearly.***

1. *[10 marks]* Consider the alphabet $\Sigma = \{a_1, \ldots, a_N, b_1, \ldots, b_N\}$ and the language $L = \bigcup_{i=1}^{N} \{a_i^j b_i^j \mid j \geq 1\}$. Clearly L is a context-free language.

   Show that it is not possible to recognize L using an NPDA with k states and k stack symbols, if $N > k^3$.

2. *[10 marks]* For a string $w \in (\mathbf{a} + \mathbf{b} + \mathbf{c})^*$, let $N_a(w)$, $N_b(w)$ and $N_c(w)$ denote the number of a's, b's and c's respectively in w. Consider the language $L = \{w \mid w \in (\mathbf{a} + \mathbf{b} + \mathbf{c})^+,\ N_a(w) = N_b(w) + N_c(w)\}$. This language is known to be context-free.

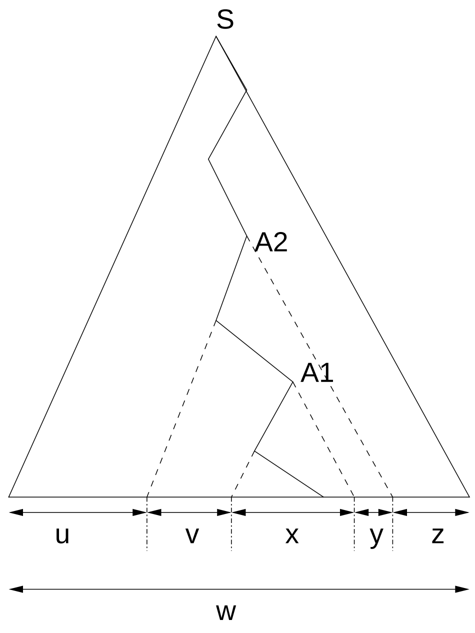

Figure 1:

Let G be a CNF grammer for L with n non-terminals. Consider the string $w = a^{2^{n+3}} b^{2^{n+2}} c^{2^{n+2}} \in L$. Let the parse tree for w be as shown in Fig. 1. Suppose we mark the c's in w (as in Ogden's Lemma). Using an argument similar to that used in Ogden's Lemma, we can find a path Π containing at least $n+1$ non-terminals from the root to a marked c leaf. As in Ogden's Lemma, suppose we now start moving up from the leaf of Π towards the root, stopping when we first see a non-terminal repeated twice. Since the path has at least $n+1$ non-terminals, we must find such a node within $n+1$ steps going up. Let the non-terminal repeated twice be A (shown as A_1 and A_2 in Fig. 1) We now divide the string w into segments u, v, x, y, z as shown in the figure. Since the grammar is CNF and A_2 is within $n+1$ steps from the marked leaf, $|vxy| \leq 2^{n+1}$. Given that $N_b(w) = N_c(w) = 2^{n+2}$, and that vxy contains at least a marked c, it follows that vxy does not contain any a. By applying a reasoning similar to Pumping Lemma, we can now claim that uv^ixy^iz must be in L for all $i \geq 0$. However, since v and y contain only b's and c's, we can choose an i large enough so that the number of b's and/or c's in w exceeds number of a's. Unfortunately, such strings cannot be in L from the definition of L.

Indicate which step in the above argument is flawed, and give justifications for your answer.

3. *[15 marks]* Consider the class of primitive recursive functions that take two arguments. We will restrict ourselves to functions from natural numbers to natural numbers. Each such function can be expressed syntactically in terms of the initial functions (zero, successor, projection) and the composition and primitive recursion operations. Therefore, it is possible to encode all primitive recursive functions by enforcing syntactic structure on strings. It follows that each primitive recursive function can be encoded as integers in binary. However, some integers, when expressed in binary, may not represent primitive recursive functions if their syntactic structure isn't right.

   We wish to know if there exists an algorithm that takes two integers m and n as inputs and returns YES if m encodes a primitive recursive function that returns n for *some* value of its argument. Otherwise, the algorithm returns NO.

   Show that such an algorithm does not exist.

4. *[10 marks]* Given context-free languages L_1, L_2, L_3 and L_4 show that it is undecidable whether $L_1 \cap L_2 \subset L_3 \cap L_4$.