Name ______________________ Student No. ______________________

No aids allowed. Answer all questions on test paper. Use backs of sheets for scratch work.

Total Marks: 50

[10] 1. Design a context-free grammar for the following language: $\{a^i b^j c^k \mid i \neq j \vee j \neq k\ \}$.

Solution: Break it up into cases: first into $i \neq j$ and $j \neq k$, so we have $S \longrightarrow S_1 | S_2$.

If $i \neq j$, then $i < j$ or $i > j$. Again, we break it up into these two cases: $S_1 \longrightarrow S_< | S_>$.

Now let $S_> \longrightarrow aS_> | aS_> b | \varepsilon$, because to generate more a's than b's, we generate all the a's we want, but for every b we must generate an a.

The other cases can be finished off in a similar manner.

2. Consider the context-free grammar $S \longrightarrow aS|aSbS|\varepsilon$.

[3] (a) Show that this grammar is ambiguous.

[7] (b) Prove that it generates all and only the strings of a's and b's such that every prefix has at least as many a'a as b's.

Solution: (a) Just observe that aab has two parse trees.

(b) We use the notation introduced in class: $n_a(\alpha)$ is the number of occurrences of symbol a in α.

The prefix property can be stated as follows: given $\alpha \in (V \cup T)^*$, for all $0 \leq i \leq |\alpha|$, $n_a(\alpha_1\alpha_2 \ldots \alpha_i) \geq n_b(\alpha_1\alpha_2 \ldots \alpha_i)$.

To see that only such strings are generated, use induction on the length of derivation. The BC is $S \Rightarrow \varepsilon$. The IS: suppose that the sentential form α, has the property, and $S \overset{*}{\Rightarrow} \alpha \Rightarrow \beta$. We show that β must have the property. Since a rule was applied, we know that $\alpha = \alpha_1 S \alpha_2$. We now consider all three rules, and see immediately that in each case the property is preserved.

Now suppose that w is a string with the property. Then, $w = ax$ (note that it cannot be $w = bx$!). Consider x. If it has the property, then $S \Rightarrow aS \overset{*}{\Rightarrow} ax$. If x does not have the property, then let $x_1 b$ be the shortest prefix for which it fails (we know that it must end with b, and we know that x_1 has the prefix property). Thus, $w = ax_1bx_2$. Note that x_2, must have the prefix property, for otherwise w wouldn't have it. Thus, $S \Rightarrow aSbS \overset{*}{\Rightarrow} ax_1bx_2 = w$.

[10] 3. Design a PDA to accept the following language: the set of all strings over $\{0, 1\}$ with twice as many 0's as 1's.

Solution: The idea is to keep the surplus of 0s or 1s on the stack. Then, when we see a 1 we pop two 0s and when we see two 0s we pop a 1. But some accounting has to be done to deal with "difficult" cases; for example, we have a single 0 on the stack, and we read a 1. In that case we write a $-0 \in Gamma$ on the stack, to record that we "borrowed" a zero. When -0 is on top, and we read a 0, we pop it. Similarly, when we read a 0, and we have a 1 on top of the stack, we replace it by $\frac{1}{2} \in \Gamma$, and when a zero comes and we have $\frac{1}{2}$ on top, we pop it. We accept, if by the time we finished reading the input the stack is empty.

[10] 4. Use the pumping lemma for context-free languages to show that $\{a^n b^n c^i | i \leq n\}$ is *not* a context-free language.

Solution: Let p be the pumping length, and consider $a^p b^p c^p = wvxyz$, which is in the language. Since $|vxy| \leq p$, and $vy \neq \varepsilon$, we have the following cases: (1) vy includes some c's and no a's; so pump it up, until we have more c's than a's. (2) vy includes only b's, so pump it up until we have more b's than a's. (3) vy includes a's, and no b's, so again pump it up. (4) vy includes a's and b's, in which case by pumping we get a string not in the right format.

[10] 5. Let 2PDA be a new model of computation, where we allow 2 stacks. Show that this model of computation is stronger than the usual one with just one stack. That is, show that every language accepted by a normal PDA is also accepted by a 2-stack PDA (this is an easy observation), but that there exists a language accepted by a 2PDA for which there is no normal PDA.

Solution: Every CFL can be decided by a 2PDA by simply ignoring the second stack. However, the language in Q4, which is not CF, can be decided on a 2PDA: push the a's on the first stack, the b's on the second stack, and then pop the two stacks while reading the c's, making sure that the two stacks have the same height n, and the c's end before we reach the bottom of the stacks.

[5] 6. BONUS QUESTION

Give a decision procedure for the problem of testing whether a given string w is in the language of a given context-free grammar G.

Solution: Convert G to CNF, and use the CYK algorithm, or simply examine all the derivations of length $2|w|-1$ to see if anyone of them generates w.