SE4I03 Test 2 Fall 2001

Name ______________________ Student No. ______________________

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

Total Marks: 40

[10] 1. Consider the following fact:

Fact: The language $L = \{a^n b^n c^n | n \geq 1\}$ is *not* context free.

Consider now the following modification of Pushdown Automata: *instead of one stack, we allow two stacks.*

Prove (using, among other things, the above fact) that the new PDAs are more powerful (they recognize more languages) than the old, single stack, PDAs.

Solution: Since single-stack PDAs recognize exactly the set of context-free languages, it follows by the fact that no single-stack PDA can recognize L.

On the other hand, there is a two-stack PDA D that recognizes L: on input w, D pushes a^n onto the first stack, then it pushes b^m onto the second stack, and then for each c in c^k it pops one a and one b. D accepts if both stacks become empty exactly when it reaches the end of the c's (i.e., when $n = m = k$).

Since it is obvious that two-stack PDAs can simulate one-stack PDAs, it follows that the new PDAs are more powerful.

[10] 2. Give a Context Free Grammar G for the language L, over the alphabet $\{a, b\}$, consisting of all strings with the same number of a's as b's.

$$G = (\{S\}, \{a, b\}, \{S \rightarrow aSbS | bSaS | \varepsilon\}, S)$$

[10] 3. Let L_{PAL} be the language of palindromes over the alphabet $\{0,1\}$; that is,

$$L_{\text{PAL}} = \{w | w \in \{0,1\}, w = w^R\}$$

Give a *clear* but informal description of a Turing Machine M that recognizes the language L. Emphasize the algorithm that the Turing Machine is implementing.

Solution: The idea behind the algorithm is the following: M moves back and forth, comparing the first and last symbols, and deleting them, until they all vanish, in which case M accepts, or until it finds a mismatch, in which case M rejects.

In more detail: on input $x = x_1 x_2 \dots x_n$, M reads x_1. If $x_1 = B$, then the input string is empty, so M accepts.

If $x_1 = i \neq B$, then M enters the state p_i (here $i \in \{0,1\}$), and replaces x_1 by B.

Then M goes to the end of the string, and checks x_n. If the values of x_n is i (which M "remembers" as it is in state p_i), M replaces x_n by B, and moves on top of x_{n-1}.

If $x_n \neq i$, then M rejects (if the first and last symbol are not the same, the string cannot be a palindrome).

Now M enters the state r_i, where $i = x_{n-1}$, and moves the head to the beginning of the string, which now is x_2. If x_2 and x_{n-1} match, it continues, if not, it rejects.

4. In class we proved the following theorem:

Theorem: The language $A_{\text{TM}} = \{\langle M, w\rangle |\ M$ is a TM that accepts the input string $w\}$ is undecidable.

Consider now the following TM: $M = (\{q_{\text{accept}}, q_{\text{reject}}, q_0\}, \{0,1\}, \{0,1,B\}, \delta)$ where δ is given as follows:

state	symbol	action
q_0	0	q_{reject}
q_0	1	q_{accept}
q_0	B	(q_0, B, R)

[5] (a) Prove that M accepts the string 100011100101011100011001110010 1.

Solution: By convention, M starts in state q_0 reading the first symbol, which in this case is 1. By definition of δ, it enters the state q_{accept}, and halts.

[5] (b) In part (a) you showed that M accepts a particular string. In fact, it seems that for any $w \in \{0,1\}^*$, one is able to decide if M accepts w. Explain why this does *not* contradict the **Theorem** above.

Solution: The theorem above says that there is no algorithm that would work on <u>all</u> Turing Machines and inputs.
Thus, even though we are able to decide for this particular TM whether it accepts or not any given string, this does *not* contradict the fact that we have no procedure that would work on all TMs.