Name ____________________ Student No. ____________________

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

Total Marks: 30

[10] 1. Show that the language

$$\mathrm{INFINITE}_{\mathrm{DFA}} := \{\langle A\rangle : A \text{ is a DFA and } L(A) \text{ is an infinite set}\}$$

is a decidable language.

Solution: See solution to exercise 4.9 in Sipser, page 185.

[10] 2. Let C be a language. Prove that C is Turing-recognizable iff a decidable language D exists such that $C = \{x : \exists y[\langle x, y\rangle \in D]\}$.

Solution: $\Leftarrow$ Let C be recognized by M, where on input x, M simulates M_D, the decider for D on inputs $\langle x, y_i\rangle$, where $y_1, y_2, y_3, \ldots$ are all the strings in lexicographic order.

$\Rightarrow$ Suppose C is recognizable; let $L(M) = C$, i.e, M recognizes C, which means that for every $x \in C$, there exists an accepting computation of M on x. Let D be the language which on input $\langle x, y\rangle$ checks whether y encodes an accepting computation of $M(x)$. An example of such an encoding could be $y = c_0\texttt{\#}c_1\texttt{\#}\ldots\texttt{\#}c_n$ where the c_i's are configurations of M on x; c_0 the initial configuration, c_n an accepting configuration, and each c_i yields c_{i+1}. Clearly D is a decider.

[10] 3. Let $\Gamma = \{0, 1, \Box\}$ be the alphabet of all the TMs we consider in this problem. Define $\text{BB} : \mathbb{N} \longrightarrow \mathbb{N}$ to be the *Busy Beaver* function; for each k, consider all k-state TMs that halt when started on a blank tape. Let $\text{BB}(k)$ be the maximum number of 1s that remain on the tape among all of these machines. Show that BB is not computable.

Solution: Suppose that BB is computable; then we could use it to decide A_{TM} as follows: let N be a TM which in input $\langle M, w\rangle$, constructs the description of a new TM M_w, where, when M_w is started on an empty tape, it writes out w and simulates $M(w)$; if $M(w)$ ever halts, M_w halts as well, accepting or rejecting just like $M(w)$, but just before it halts, M_w overwrites all the squares it ever used with 1s.

Then N computes $l = \text{BB}(k)$ where $k = |Q_{M_w}|$, that is, k is the number of states of M_w. Now N marks the l-th square from the left on one of its tapes, and on that tape it simulates M_w on a blank tape. Either M_w stays within the first l squares, or it does not. If it does, it either halts (and accepts or rejects, in which case N does the same), or it enters an infinite loop; but since the space is bounded, this loop can be detected (when space is bounded, an infinite computation must start repeating configurations). If N detects that configurations are repeating, it rejects.

Finally, if M_w goes beyond the l-th square, N knows that the computation will never halt (if it did, then $\text{BB}(k) > l$, giving a contradiction, because if M_w would still halt after going beyond the l-square, it would have produced more 1s).