AC32008 Theory of Computation

Tutorial Sheet 5 - Turing Machines. Partially Decidable and Totally Decidable Languages.

1. Consider the Turing machine described below:

   Input alphabet $\Sigma = \{0, 1\}$
   Tape alphabet $\Gamma = \{0, 1, B\}$
   States: $Q = \{q_1, q_2, q_3, q_4, q_5, q_6\}$
   Initial state: q_1
   Final/accepting states: $F = \{q_6\}$

   Transition function δ:

δ	0	1	B
q_1	$(q_3, 0, R)$	$(q_2, 1, R)$	$-$
q_2	$(q_3, 0, R)$	$(q_2, 1, R)$	(q_5, B, L)
q_3	$(q_3, 0, R)$	$(q_4, 1, R)$	(q_5, B, L)
q_4	(q_4, B, R)	(q_4, B, R)	(q_5, B, L)
q_5	$-$	$-$	(q_5, B, L)
q_6	$-$	$-$	$-$

   Carry out the computation of this machine on input 110010001.

2. Construct a Turing Machine (i.e. give the states, tape and input alphabets and transition function δ) which accepts the language $\{0^n \mid n \geq 0, n \text{ not divisible by } 2\}$.

   Note that this language is actually regular, with regular expression $\mathbf{0}(\mathbf{00})^*$.

3. Construct a Turing Machine that will repeat its (binary) input, so that if the input is a string w, the output is ww.

   Thus if the original input is, say, 101, then at the end of the computation the tape should just have the string 101101.

   Note: It is possible to do this just using binary, but you will probably find it easy to use some extra tape symbols. Typically the extra symbols will be in pairs, with one of each pair representing 0 and the other 1.

4. What does it mean to say that a language L is not totally decidable?