CSE 105: Homework Set 7

Joshua Wheeler

May 20, 2014

1. Turing Machine Recognizability

   This statement is True. It is possible to build a TM that is able to accept or loop, but never reject. We can construct such a machine with a slight modification to the universal machine as it was defined in Theorem 4.11 as follows

   $U_{noreject}$ = "On input $< M, w >$, where M is a recognizable TM and w is a string:
   1. Simulate M on input w
   2. If M ever enters its accept state, accept; if M ever enters its reject state, move the head to the right, transition on whichever symbol is now being read and do not reject"

   This machine defines any Turing recognizable machine that instead of rejecting, will always keep looping indefinitely by moving its tape head to the right if it reaches a reject state and thus forcing an extra transition. Therefore, it is true that all Turing recognizable languages can accept, halt and never reject.

2. Turing Machine Decidability

   Theorem 4.22 proves that a language is decidable iff it is Turing-recognizable and co-Turing-recognizable. In order to show that ALL_{DFA} is decidable we must show it holds both of the above properties. Note that since these languages can be recognized by a DFA then it automatically holds that they can also be recognized by a TM because the class of regular languages is a subset of Turing Recognizable languages.

   ALL_{DFA} is Turing-recognizable since by definition a DFA has a finite amount of states, and therefore will always reach an accepting or rejecting state for any input $w \in \Sigma^*$ after some finite amount of transitions.

ALL_{DFA} is co-Turing-recognizable since the complement of the language $L(A) = \Sigma^*$ is $L(\overline{A}) = \emptyset$ which can be defined by a DFA with one initial state, no transitions and no accept states.

3. Proving Countability

   In order to prove that $N \times N$ is countable we must create a function f: $N \to N \times N$ that is a both one to one and onto. To construct such a function we can take advantage of diagonalization in a similar way to figure 4.16 in the textbook. Such a table can be created in the following way:

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	...
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	
...					...

   To count through this table we count the diagonals in exactly the same way figure 4.16 counts them in the textbook. By counting the elements in this way it is possible to iterate through all of the elements in N, and thus form a function $f : N \to N \times N$ that is both one to one and onto.

4. Proving Uncountability

   This proof is similar to the proof to show that the set of real numbers is uncountable by diagonalizing the set and finding some string x that is not in f(n) for any n.

   We can begin by inserting the set into a countably infinite table and count the elements as we insert them in a diagonal fashion to construct a table in the order shown:

0	10	101	1001	...
1	100	1000	...	
11	111	...		
110	...			
1010				
...				...

Counting the diagonals it seems that this set is countable but suppose we construct a function f(n) such that n is a pair of integers representing the position in the table and f(n) is a unique binary string in the table with a one to one and onto correspondence with n. n was proved in problem 3 to be countable. By flipping a bit in some unique nth position of the binary string we can find a string w that is not in f(n) for any n and hence a contradiction. Therefore the set of infinite binary strings is uncountable.