474 HW 15 problems (highlighted problems are the ones to turn in)

18.1a
(#1) 9

18.1b
(#2) 9

1. Church's Thesis makes the claim that all reasonable formal models of computation are equivalent. And we showed in, Section 17.4, a construction that proved that a simple accumulator/register machine can be implemented as a Turing machine. By extending that construction, we can show that any computer can be implemented as a Turing machine. So the existence of a decision procedure (stated in any notation that makes the algorithm clear) to answer a question means that the question is decidable by a Turing machine.

Now suppose that we take an arbitrary question for which a decision procedure exists. If the question can be reformulated as a language, then the language will be in D iff there exists a decision procedure to answer the question. For each of the following problems, your answers should be a precise description of an algorithm. It need not be the description of a Turing Machine:

a. Let $L = \{<M> : M$ is a DFSM that doesn't accept any string containing an odd number of 1's$\}$. Show that L is in D.

Problem #3 A TM M has tape alphabet { □, a, b} (this is the order used in the encoding <M>).

<M> = (q00,a00,q00,a00, →), (q00,a10,q01,a10,→), (q00,a01,y10,a10, ←), (q01,a01,q00,a10,→), (q01,a10,n11,a01, ←)

(a) (6) Provide a transition diagram or a transition table for the TM M.

(b) (3) For each of the following outcomes of running M, provide a short string of a's and b's that is
accepted by M,
rejected by M,
neither.

19.1
(#4) 6

19.2
(#5) 12

1. Consider the language $L = \{<M> :$ Turing machine M accepts at least two strings$\}$.

a. Describe in clear English a Turing machine M that semidecides L.

b. Now change the definition of L just a bit. Consider:

$L' = \{<M> :$ Turing machine M accepts *exactly* 2 strings>.

Can you tweak the Turing machine you described in part a to semidecide L'?

2. Consider the language $L = \{<M> :$ Turing machine M accepts the binary encodings of the first three prime numbers$\}$.

a. Describe in clear English a Turing machine M that semidecides L.

b. Suppose (contrary to fact, as established by Theorem 19.2) that there were a Turing machine *Oracle* that decided H. Using it, describe in clear English a Turing machine M that decides L.