Bradley White
CSCI 338
HW #6
4-8-2016

1.

High level description of a TM algorithm that decides $L = \{w|\ w$ does not contain twice as many 0s as 1s$\}$. Let M be the TM that decides L.

$M =$ "On input $\langle L \rangle$:

1. Scan for the first unmarked 1. If none found, go to step 5.
2. Scan for the first unmarked 0. If none found, accept.
3. Repeat step 2.
4. Repeat step 1.
5. Scan for any unmarked 0s. If none are found reject, otherwise accept.

2. 3.15d, p.189

Let M_1 be a Turing machine that decides a language *L* and construct a Turing machine M_2 that decides $\overline{L}$:

M_2 = "On input *w*:

1. Simulate M_1 on *w*.
2. Accept if M_1 rejects, reject if M_1 accepts."

Since M_1 decides *L* we know it halts on all inputs, therefore M_2 will also halt on all inputs. Additionally, M_2 will always produce the correct result because if $w \in L$, M_1 will accept and M_2 will reject, also if $w \notin L$, M_1 will reject, in which case M_2 will accept.

3. 4.2, p.211

$L = \{\langle A, R \rangle$ | where A is a DFA equivalent to the regular expression $R\}$
Let M_1 be a TM that decides *L*:

$M_1 =$ "On input $\langle A, R \rangle$:

1. Convert R into an equivalent DFA B
2. Run TM F, from Theorem 4.5 which decides EQ_{DFA}, on input $\langle A, B \rangle$
3. Accept if F accepts, and reject if F rejects."

Since EQ_{DFA} was proven to be a decidable language, L is therefore also decidable.

4. 4.3, p.211

Let M_1 be a TM that decides ALL_{DFA}:

$M_1 =$ "On input $\langle A \rangle$:

1. Construct a DFA B that recognizes $\overline{L(A)}$
2. Run TM T, from Theorem 4.4 which decides E_{DFA}, on input $\langle B \rangle$
3. Accept if T accepts, and reject if T rejects."

Since E_{DFA} was proven to be a decidable language, ALL_{DFA} is also decidable.

Bradley White
CSCI 338
HW #6
4-8-2016

5. 4.7, p.211

The proof is by contradiction, that is suppose that $\mathcal{B}$ is countable. Each element in $\mathcal{B}$ is an infinite sequence $(b_1, b_2, b_3, \dots)$ where $b_i \in \{0,1\}$. We can define a correspondence f between $\mathcal{B}$ and $\mathcal{N}$. Let $f(n) = (b_{n1}, b_{n2}, b_{n3}, \dots)$ where $n \in \mathcal{N}$ and b_{ni} is the ith bit in the nth sequence. For example:

n	$f(n)$
1	$(1,0,0,1,1, \dots)$
2	$(0,1,0,1,0, \dots)$
3	$(1,1,1,1,1, \dots)$
4	$(1,1,0,0,0, \dots)$
$\vdots$	$\vdots$

Define a sequence $s \in \mathcal{B}$, in which the ith bit in s is opposite the ith bit in the ith sequence. So for the example above $s = (0,0,0,1, \dots)$. Thus, s differs from each sequence by at least one bit and is not in $\mathcal{B}$. However s is an infinite sequence of bits and so must be contained in $\mathcal{B}$ by definition. From this contradiction we conclude that no such list can exist and $\mathcal{B}$ is not countable.

6. 4.8, p.211

We will construct a correspondence, f, which will map $\mathcal{T}$ to $\mathcal{N}$ by using the unique factorization theorem. The theorem states that every integer greater than 1 is prime itself or is the product of prime numbers and that this product is unique. So, we can arbitrarily choose primes $a \neq b \neq c$ and define the function $f(i,j,k) = a^i b^j c^k$. From the unique factorization theorem, any choice of i, j, k uniquely determines a value of f, therefore f is one-to-one and onto and $\mathcal{T}$ is countable.