CS 420, Spring 2019
Homework 9 Solutions

1. Let $F = \{\langle A\rangle | A$ is a DFA with input alphabet $\{0, 1\}$ and some string in $L(A)$ contains exactly three 1's$\}$. Prove that F is decidable.

   Solution: The set of all binary strings that contain exactly three 1's is a regular language, so there is a DFA B that recognizes this language. Then, $\langle A\rangle$ is in F if and only if $L(A) \cap L(B) \neq \emptyset$. Thus, we can define a Turing machine M that decides the language F by

   $M =$ "On input $\langle A\rangle$ where A is a DFA with tape alphabet $\{0, 1\}$,

   1. Produce a DFA C with $L(C) = L(A) \cap L(B)$, where B is a DFA that recognizes the set of binary strings that contain exactly three 1's.
   2. Run the TM T that decides E_{DFA} on $\langle C\rangle$.
   4. If T rejects, then *accept*. If T accepts, then *reject*."

2. Let $K = \{\langle A\rangle | A$ is a DFA with alphabet $\{a, b\}$ and $L(A)$ does not contain any string with exactly one more a than $b\}$ Show that K is decidable. (The solution to Problem 4.25 [4.23] is useful here.)

   Solution: The language of all strings in $\{a, b\}^*$ with exactly one more a than b is context-free. (You were given a context-free grammar for a similar language in the solutions to Homework 5.) Let P be a PDA that recognizes this language.

   A Turing machine T that decides K is given by:

   T="On input $\langle A\rangle$ where A is a DFA:

   1. Using the method of Problem 2.18a, construct a PDA S that recognizes $L(A) \cap L(P)$, where P is the PDA mentioned above.
   2. Convert S into a CFG G.
   3. Run the TM R that decides E_{CFG} on $\langle G\rangle$.
   4. If R accepts, *accept*. If R rejects, *reject*."

3. Let $L = \{\langle P\rangle | P$ is a PDA with input alphabet $\{0, 1\}$ and no string in $L(P)$ contains exactly three 1's$\}$. Prove that L is decidable.

   Solution: The set of binary strings that contain exactly three 1's is a regular language.

   A Turing machine T that decides L is given by:

   T="On input $\langle P\rangle$ where P is a PDA:

   1. Let E be a DFA that recognizes the set of binary strings that contain exactly three 1's.
   2. Using the method of Problem 2.18a, construct a PDA S that recognizes $L(E) \cap L(P)$.

3. Convert S into a CFG H.
4. Run the TM R that decides E_{CFG} on $\langle H \rangle$.
4. If R accepts, *accept.* If R rejects, *reject.*"

4. The set of all finite sequences of natural numbers is countable.

   Proof: Unlike the case of Σ^* where Σ is an alphabet, it is not possible to list all the finite sequences of natural numbers in order by their length, because there are infinitely many sequences of natural numbers with any given length other than 0, so this list would never get to any sequences of length 2, much less all the finite sequences.

   One method that does work is to list the finite sequences of natural numbers in order according to the sum of their entries. (Because we do not consider 0 to be a natural number, there are only finitely many finite sequences of natural numbers whose sum is any given number.) For a given sum, the sequences can be listed in lexicographic (ie, dictionary) order.

   The first sequence listed would be ε, then $\langle 1 \rangle$. Next would come the two sequences whose sum is 2: $\langle 1, 1 \rangle$ and $\langle 2 \rangle$. The sequences whose sum is 3 come next: $\langle 1, 1, 1 \rangle$, $\langle 1, 2 \rangle$, $\langle 2, 1 \rangle$ and $\langle 3 \rangle$. Next come the sequences with sum 4, and so on.

5. Let A be a countably infinite set, B be a set, and $f : A \to B$ be onto. Prove that B is countable.

   Solution:

   Since A is countably infinite, its elements cxan be listed without repetions as $a_1, a_2, a_3, \ldots$. Since f is onto, B is equal to the range of f, so the elements of B can be listed as $f(a_1), f(a_2), f(a_3), \ldots$. There may be repetitions in the list, but once repetitions are removed a finite or infinite list of the elements of B without repetition is obtained, so B is countable.

6. An infinite sequence of natural numbers $a(1)a(2)a(3)\cdots$ is called *strictly increasing* if $a(1) < a(2) < a(3) < \cdots$. Let B be the set of all strictly increasing sequences of natural numbers. Use diagonalization to prove that B is uncountable.

   Solution:

   Let $a_1, a_2, \ldots$ be a sequence elements of B. We will define a sequence d in B that is not on the list. This will show that B is uncountable, since no list of the form $a_1, a_2, \ldots$ can list every element of B.

   The sequence d is defined by $d(1) = a_1(1) + 1$ and $d(n+1) = a_{n+1}(n+1) + d(n)$. Then for all n, $d(n+1) = a_{n+1}(n+1) + d(n) > d(n)$ (since $a_{n+1}(n+1) > 0$), so d belongs to B, and for all m, $d(m) \neq a_m(m)$, so d is not on the list.

7. Let C be the set of infinite binary sequences $a(1)a(2)a(3)\cdots$ such that $a(1) = a(3) = a(5) = \cdots = 0$. In other words a sequence in C can

have either 0 or 1 in the even positions, but has to have 0 in the odd positions.Prove that C is uncountable.

Solution:

We give two proofs that C is uncountable. The first one uses diagonalization. Let $a_1, a_2, \cdots$ be a sequence of elements of C. We define a sequence $d = d(1)d(2)\cdots$ that is in C but is different from all the a_i's. We define $d(n) = 0$ for all odd numbers n, and we define $d(2n)$ to be 1 if $a_n(2n) = 0$ and to be 0 if $a_n(2n) = 1$. Then, d is in C and d is different from each a_n since $d(2n) \neq a_n(2n)$.

For a second proof, let B be the set of all infinite binary sequences. We showed in class that B is uncountable. We define a function $f : B \to C$ as follows. For every $a = a(1)a(2)\cdots$ in B, we define $f(a) = 0a(1)0a(2)0a(3)\cdots$. It is easy to see that f is a bijection from B to C, so B and C have the same size. Since B is uncountable, C is uncountable.