Student ID: 004757414

Collaborators: Terry Ye

# CS181 Winter 2019 – Problem Set 4

## Due Tuesday, February 26, 11:59 PM

- Please write your student ID **and the names of anyone you collaborated with** in the spaces provided and attach this sheet to the front of your solutions. **Please do not include your name anywhere since the homework will be blind graded.**
- An extra credit of **5%** will be granted to solutions written using LaTeX. Here is one place where you can create LaTeXdocuments for free: `https://www.overleaf.com/`. The link also has tutorials to get you started. There are several other editors you can use.
- If you are writing solutions by hand, please write your answers in a neat and readable hand-writing.
- Always explain your answers. When a proof is requested, you should provide a rigorous proof.
- If you don't know the answer, write "I don't know" along with a clear explanation of what you tried. For example: "I couldn't figure this out. I think the following is a start, that is correct, but I couldn't figure out what to do next. [[Write down a start to the answer that you are sure makes sense.]] Also, I had the following vague idea, but I couldn't figure out how to make it work. [[Write down vague ideas.]]" At least 20% will be given for such an answer.

  Note that if you write things that do not make any sense, no points will be given.
- The homework is expected to take anywhere between 8 to 14 hours. You are advised to start early.
- Submit your homework online on Gradescope.

1. **(20 points)** Show that the set of three-dimensional coordinates $\{(x, y, z) | x, y, z \in \mathbb{Z}\}$ has size equal to $\mathbb{N}$.

2. **(30 points)** Let us define Fast-Rewind Turing Machines (FRTM). They are similar to Turing Machines, except that the head of an FRTM is not allowed to move left one cell at a time. Instead, the head of the FRTM *must* move left only all the way to the left-hand end of the tape (i.e. the first cell). The head of the FRTM can move right one step at a time, just like the usual Turing Machines. The transition function for the FRTM is of the form
$\delta : Q \times \Gamma \to Q \times \Gamma \times \{R, FIRST\}$.

   Explain how to convert any Turing Machine into an FRTM. A full proof is not needed, you only need to give an explanation of the intuition behind why your transformation works.

3. **(50 points).** In class, we showed the existence of two kinds of infinities. Let $\Sigma = \{0, 1\}$ and $\mathcal{L} = \{L \mid L \subseteq \Sigma^*\}$. We showed that $|\Sigma^*|$ is not the same as $|\mathcal{L}|$.

   We briefly describe the proof below and establish some notation. The proof proceeds by contradiction. Let $(\epsilon, 0, 1, 00, 01, 10, 11, \ldots)$ be a given enumeration of strings in $\Sigma^*$. We assume for the sake of contradiction that there exists some enumeration $(L_1, L_2, L_3, \ldots)$ of languages in $\mathcal{L}$. Then we proceed by constructing a language $L^{DIAG} \in \mathcal{L}$ such that $\forall i, L^{DIAG} \neq L_i$ via Cantor's Diagonalization to establish a contradiction.

   (a) **(10 points)** Call $L_1^{DIAG} = L^{DIAG}$. Construct another language $L_2^{DIAG} \neq L_1^{DIAG}$ via diagonalization which is also not present in the enumeration $(L_1, L_2, L_3, \ldots)$. Thus $L_2^{DIAG}$ would have also proved to us that $|\Sigma^*| \neq |\mathcal{L}|$.

   (b) **(15 points)** Construct an infinite set of languages $\mathcal{L}^{DIAG} = \{L_1^{DIAG}, L_2^{DIAG}, \ldots, L_i^{DIAG}, \ldots\}$ via diagonalization by providing a description of $L_i^{DIAG}$ such that

   - $\forall j, j \neq i, L_i^{DIAG} \neq L_j^{DIAG}$.
   - $\forall j, L_i^{DIAG} \neq L_j$.

   Formally prove your construction by induction.

   (c) **(15 points)** Construct one more language $L^{SUPERDIAG}$ such that

   - $\forall j, L^{SUPERDIAG} \neq L_j^{DIAG}$ and
   - $\forall j, L^{SUPERDIAG} \neq L_j$

   Briefly explain why your language satisfies the above properties.

   (d) **(10 points)** Construct yet another language $L_2^{SUPERDIAG}$ that is different from $L^{SUPERDIAG}$ satisfying the same properties as part (c). Briefly explain your answer.

# 1

There is a function f from $\{(x, y, z)|x, y, z \in \mathbb{Z}\}$ to z by line x,y,z as $x_1, x_2, ..., y_1, y_2, ..., z_1, z_2, ...$ And this can be transformed to a line with $(x_1, y_1, z_1), (x_1, y_1, z_2)....$ So this is injective because they if f(a) = f(b) $\in Z$, a = b because they must consist of the same variable in x,y,z as it follows the order. F is also surjective because every x $\in Z$ will be reached at last.

# 2

An FRTM is only different from Turing machine when a TM needs to move its head to one step left. When a TM needs to move its head to left, it can be achieved in an FRTM by changing the current cell to its corresponding mark like a to a'. Then FRTM can move to left end and copy all tape cells to one right cell on a new tape. It keeps the marker by change the cell that is left to the current marker to its corresponding marker. Then move to the left again, start reading to the right and stops when it reads the marker. This is the left one of the cell before and we can work on the new tape as moving to left.

# 3

(a)
We can construct $L_2^{DIAG}$ from the sequence $\{L_1^{DIAG}, L_1, L_2, ...\}$. We can apply Cantor's Diagonalization to it by having all x $\in \Sigma^*$ in columns and $\{L_1^{DIAG}, L_1, L_2, ...\}$ on the rows. The matrix M is constructed by having 1 in the entry if the col is in row, 0 otherwise. Then $L_2^{DIAG}$ is constructed by complementing the diagnol of M as include the column if the matching diagnol entry is 0. $L_2^{DIAG}$ is not in the sequence because it takes the complement of diagnol. So this can also be used to prove that $|\Sigma^*| \neq |\mathcal{L}|$.

(b))
Construct $L_i^{DIAG}$ as the Cantor Diagonal of the sequence $\{L_1^{DIAG}, L_1, L_2^{DIAG}, L_2, ...., L_{i-1}^{DIAG}, L_{i-1}, L_i...\}$.
The base case is when i = 1, $L_1^{DIAG}$ satisfies the first requirement as every $L_i^{DIAG}$ is constructed by taking the Cantor Diagnol so they are all different from $L_1^{DIAG}$ in first element. It satisfies the second requirement because we already proved before that $L_1^{DIAG}$ is different from all L.
Suppose $L_n^{DIAG}$ satisfies the two requirements, $L_{n+1}^{DIAG}$ satisfies the first requirement as it is different from all previous $L_i^{DIAG}$ $(i \leq n)$ as it is a Cantor Diagnol of the set containing it. It is also different from all $L_i^{DIAG}$ (i ¿ n+1) because they are constructed via Cantor Diagnolization of a set containing $L_{n+1}^{DIAG}$.
It satisfies second requirement the same reason as $L_1^{DIAG}$ as it is constructed by Cantor Diagnolization of all $L_j$.
So by induction, the infinite set satisfies the requirements.

(c)
We can construct $L^{SUPERDIAG}$ via Cantor Diagonalization of the set $\{L_1^{DIAG}, L_1, L_2^{DIAG}, L_2, ...., L_{i-1}^{DIAG}, L_{i-1}, L_i...\}$. Then it satisfies the first and second requirement because it is different from all elements in the set as it takes the complement of diagnol.

(d)
We can just simply add $L^{SUPERDIAG}$ to the set in (c) and construct $L_2^{SUPERDIAG}$ via Cantor Diagonalization of the set. It satisfies the two requirements for the same reason for $L^{SUPERDIAG}$ and it is different from $L^{SUPERDIAG}$ because that is in the set as well.