Student ID: 805167986

Collaborators: David Tran, Dylan Gunn

CS181 Winter 2021 – Problem Set 4

Due Sunday, February 28, 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 LaTeX documents for free: `https://www.overleaf.com/`. The link also has tutorials to get you started. There are several other editors you can use. We have also posted a short LaTeX tutorial on CCLE under References.
- 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 6 to 12 hours. You are advised to start early.
- Submit your homework online on Gradescope. The Gradescope code is 5V7GW5.
- Homework points will be scaled according to the number of homework assignments. All assignments will be weighted equally. As per the syllabus, your homework assignments will together comprise 25% of your final grade.

Note: This homework set starts with an extra credit problem. The extra credit problem is optional, but is worth extra credit points. Problems 2 and 3 are *not* extra credit problems.

1. **(Optional: 10 Extra Credit Points aka 5 Tokens)** Let us define Fast-Rewind Turing Machines (FRTM). They are similar to Turing Machines, except that the head of a 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.

   Note that this requires you to show how to implement a "Move Left One Space" transition in a FRTM by using only "Move Right One Space" transitions and "Move All the Way Left" transitions.

   [answer to 1]

2. **(20 points)** Show that the set of three-dimensional coordinates $\{(x, y, z)|x, y, z \in \mathbb{Z}\}$ has size equal to $\mathbb{N}$.

Claim:

The set of three-dimensional coordinates $A = \{(x, y, z)|x, y, z \in \mathbb{Z}\}$ has size equal to $\mathbb{N}$.

Proof:

Let us create a map $f : A \to B$ where we are mapping the entire set of $A = \{(x, y, z)|x, y, z \in \mathbb{Z}\}$ to the co-domain $B \subset \mathbb{Q}$.

We can do this by mapping each coordinate to these 3 prime numbers raised to the power of each component of the point.

$$\forall (x, y, z) \in A \to 2^x 3^y 5^z$$

Then we can define B to be

$$B = \{2^x 3^y 5^z \mid x, y, z \in \mathbb{Z}\}$$

This mapping $f : A \to B$ is surjective and injective.

It is surjective because for every point (x, y, z) there is a mapping to a number $2^x 3^y 5^z$.

It is injective because for any two points $a = (x_a, y_a, z_a), b = (x_b, y_b, z_b)$

$$\forall a, b \in A, f(a) = f(b) \to a = b$$

This is true for all points a, b which satisfy the above because we picked 3 prime numbers.

Because we know it is surjective and injective this means that f is a bijection from $A \to B$.

And when there is a bijection from one set to another that means their cardinalities are equal.

$$|A| = |B|$$

We know that B can contain fractions which are not in the real numbers, $\mathbb{N}$.

$$\therefore |\mathbb{N}| \leq |B|$$

We know $B \subset \mathbb{Q}$ we can also argue for a surjection from $\mathbb{N}$ to B because $B \subset \mathbb{Q}$ and $|\mathbb{N}| = |\mathbb{Q}|$ as we saw in class.

$$\mathbb{N} \to B$$

$$\therefore |\mathbb{N}| \geq |B|$$

Because of these two statements we can say

$$|\mathbb{N}| = |B| = |A| = |\{(x, y, z)|x, y, z \in \mathbb{Z}\}|$$

$$\therefore |\mathbb{N}| = |\{(x, y, z)|x, y, z \in \mathbb{Z}\}|$$

$\square$

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 Σ^*. 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 $|\Sigma^*| \neq |\mathcal{L}|$.

To create another language L_2^{DIAG} which is also

$$L_2^{DIAG} \notin (L_1, L_2, L_3, \ldots)$$

We must let there be another Turing Machine A that recognizes L_1^{DIAG}.
We have an enumeration of Turing machines

$$(M_1, M_2, M_3, \ldots)$$

which accept elements in the enumeration of languages

$$(L_1, L_2, L_3, \ldots)$$

Let the enumeration of

$$\Sigma^* = (x_1, x_2, x_3, \ldots)$$

We can add our new TM A to the beginning of the list of languages

$$(A, L_1, L_2, L_3, \ldots)$$

Then we can use diagonalization (see Figure 1) to make L_2^{DIAG} where

$$L_2^{DIAG} \neq L_1^{DIAG}$$

$$L_2^{DIAG} = \{x_i \mid M_i(x_i) \text{ does not accept } \forall i \in (2, 3, 4, \ldots)\} \cup \{x_1 \mid A(x_1) \text{ does not accept}\}$$

From our construction you can see that $L_2^{DIAG} \neq L_1^{DIAG}$ and $L_2^{DIAG} \notin (L_1, L_2, L_3, \ldots)$.
$\therefore$ from L_2^{DIAG} we can also prove that $|\Sigma^*| \neq |\mathcal{L}|$.

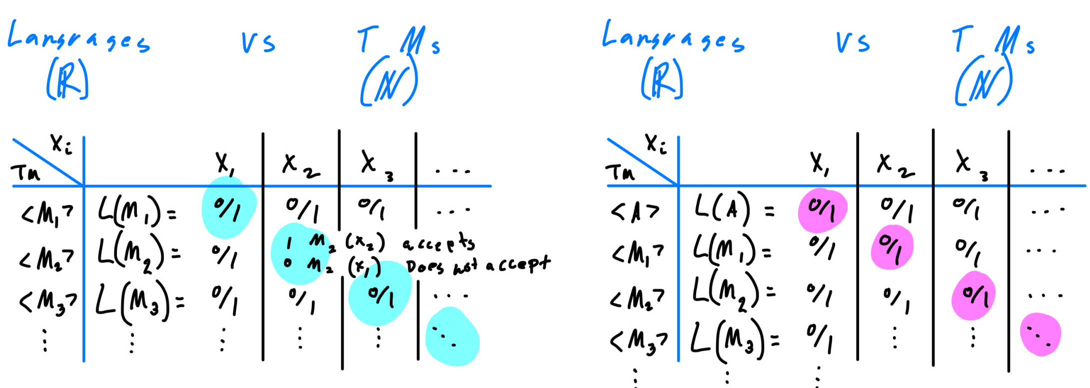

Figure 1: Diagonalization of L_2^{DIAG}

(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.

Let's create an enumeration of new Turing Machines

$$(A_1, A_2, A_3, \ldots)$$

which each respectively accept the languages

$$(L_1^{DIAG}, L_2^{DIAG}, L_3^{DIAG}, \ldots)$$

We also still have the enumeration of Turing Machines from part 3a.

$$(M_1, M_2, M_3, \ldots)$$

which each respectively accept the languages

$$(L_1, L_2, L_3, \ldots)$$

Now we can make L_i^{DIAG} such that

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

Now we can add the enumeration of TM's to the beginning of the other enumeration of Turing machines.

$$(A_i, A_{i-1}, A_{i-2}, \ldots, A_1, M_1, M_2, \ldots)$$

From here we can diagonalize as seen in Figure 2.
Now we can define L_i^{DIAG} to be the following

$$L_d = \{x_a \in \Sigma^* \mid A_{a-1}(x_a) \text{ does not accept } \forall a \in (2, 3, \ldots, i-1)\}$$

$$L_i^{DIAG} = \{x_b \in \Sigma^* \mid M_{b-i+1}(x_b) \text{ does not accept } \forall b = i, i+1, i+2, \ldots\} \cup L_d$$

Base Case:
$L_i^{DIAG}, \forall i \in \mathbb{N}$ satisfies

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

Inductive Case:
Let there be a language L_{i+1}^{DIAG}.
Because L_{i+1} is the diagonalization of Turing Machines

$$(A_i, A_{i-1}, A_{i-2}, \ldots, A_1, M_1, M_2, \ldots)$$

This means that L_{i+1} does not equal any of the orignal languages which we enumerated or any of the diagonal languages.
$\therefore$ there exists a set of diagonal languages $\mathcal{L}^{DIAG}$ which has an infinite cardinality.

$\square$

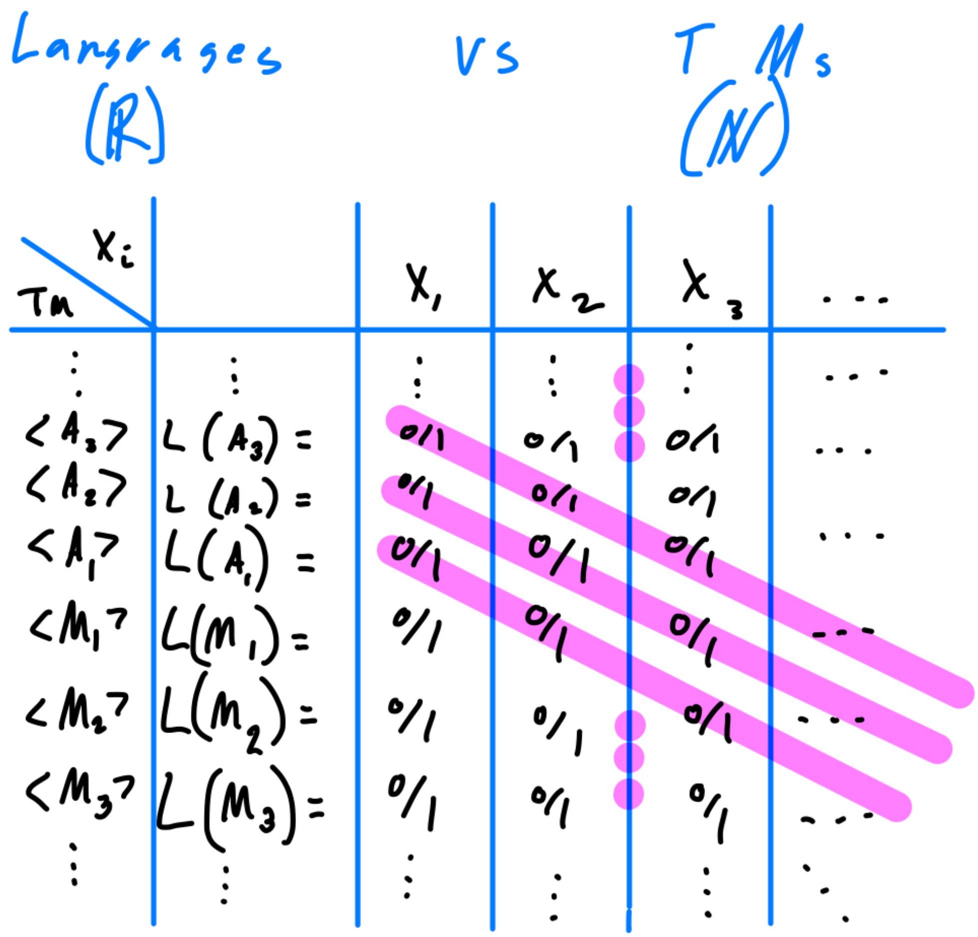

Figure 2: Diagonalization of L_i^{DIAG}

(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.

We have an enumeration of new Turing Machines

$$(A_1, A_2, A_3, \ldots)$$

which each respectively accept the languages

$$(L_1^{DIAG}, L_2^{DIAG}, L_3^{DIAG}, \ldots)$$

We also still have the enumeration of Turing Machines from part 3a.

$$(M_1, M_2, M_3, \ldots)$$

which each respectively accept the languages

$$(L_1, L_2, L_3, \ldots)$$

Now we can make a new enumeration of these machines like the following

$$(M_1, A_1, M_2, A_2, \ldots)$$

which interleaves the machines from 3a and 3b.
Then we can diagonalize this enumeration of TM's as seen in Figure 3.
We can define $L^{SUPERDIAG}$ as follows

$$L_{SD1} = \{x_{2a-1} \in \Sigma^* \mid M_a(x_{2a-1}) \text{ does not accept } \forall a \in \mathbb{N}\}$$
$$L_{SD2} = \{x_{2b} \in \Sigma^* \mid A_b(x_{2b}) \text{ does not accept } \forall b \in \mathbb{N}\}$$
$$L^{SUPERDIAG} = L_{SD1} \cup L_{SD2}$$

From this $L^{SUPERDIAG}$ satisfies

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

By this construction of $L^{SUPERDIAG}$ through the diagonalization of the new enumeration of TM's

$$(M_1, A_1, M_2, A_2, \ldots)$$

Where this new enumeration of TM's above solves the languages

$$(L_1, L_2, L_3, \ldots)$$

and

$$(L_1^{DIAG}, L_2^{DIAG}, L_3^{DIAG}, \ldots)$$

$\therefore L^{SUPERDIAG}$ satisfies those properties by diagonalization.

$\square$

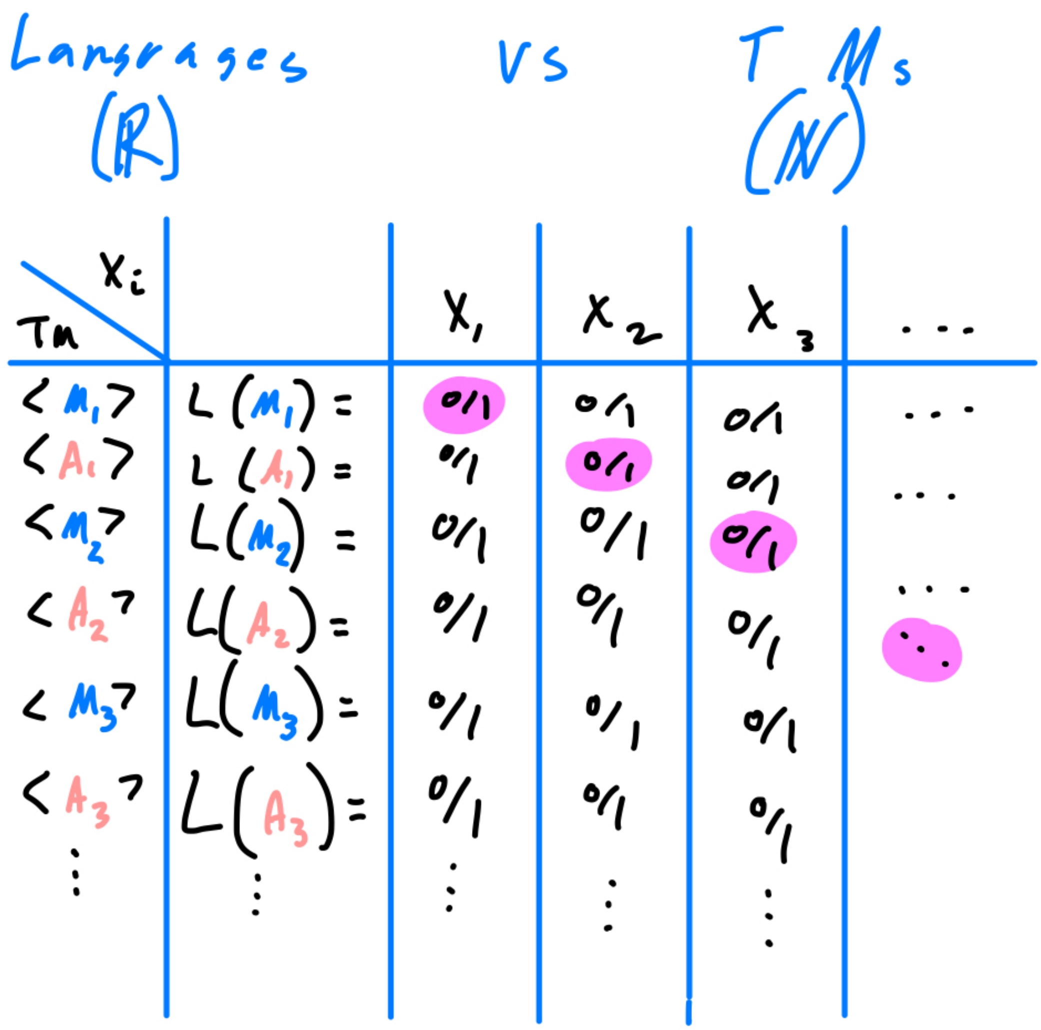

Figure 3: Diagonalization of $L^{SUPERDIAG}$

(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.

Let B be the TM which recognizes $L^{SUPERDIAG}$.
We will put this TM at the beginning of the enumeration from 3c.

$$(B, M_1, A_1, M_2, A_2, \ldots)$$

which interweaves the previous TM's from 3b with the TM's from 3a.
We can make $L_2^{SUPERDIAG}$ by diagonalizing this new enumeration of TM's as seen in Figure 4.
We can define it as the following

$$L_{S1} = \{x_{2b-1} \in \Sigma^* \mid A_b(x_{2b-1}) \text{ does not accept } \forall b \in (2, 3, 4, \ldots)\}$$
$$L_{S2} = \{x_1 \mid B(x_1) \text{ does not accept}\}$$
$$L_{S3} = \{x_{2a} \in \Sigma^* \mid M_a(x_{2a}) \text{ does not accept } \forall a \in (2, 3, 4, \ldots)\}$$
$$L_2^{SUPERDIAG} = L_{S1} \cup L_{S2} \cup L_{S3}$$

By diagonalizing the formation of $L_2^{SUPERDIAG}$ over the new enumeration of TM's which solve
$$(L^{SUPERDIAG}, L_1, L_1^{DIAG}, L_2, L_2^{DIAG}, \ldots)$$
respectively.
This ensures that $L_2^{SUPERDIAG} \neq L, \forall L \in (L_1, L_1^{DIAG}, L_2, L_2^{DIAG}, \ldots)$
and
$L_2^{SUPERDIAG} \neq L^{SUPERDIAG}$
and it satisfies

- $\forall j, L_2^{SUPERDIAG} \neq L_j^{DIAG}$ and
- $\forall j, L_2^{SUPERDIAG} \neq L_j$

$\therefore$ our construction of $L_2^{SUPERDIAG}$ is valid.

$\square$

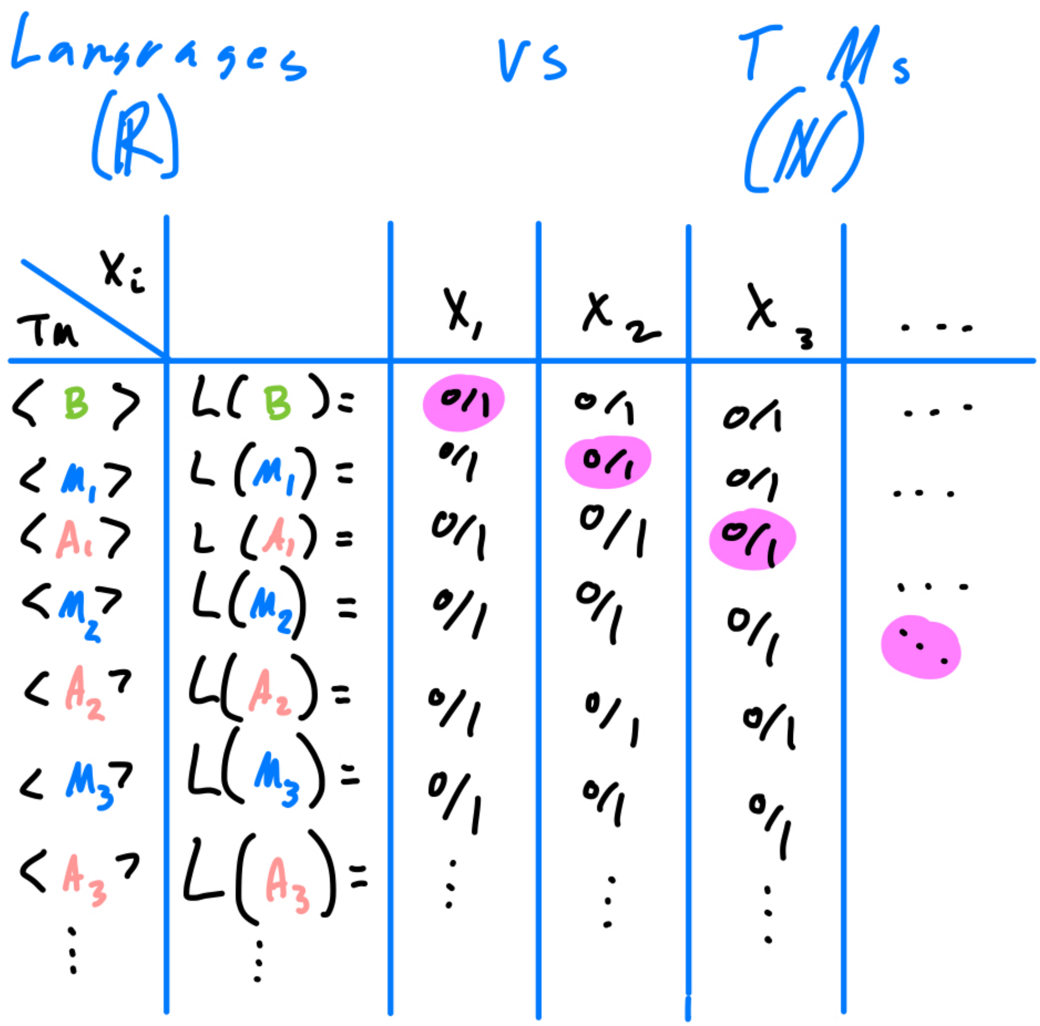

Figure 4: Diagonalization of $L_2^{SUPERDIAG}$