CSCI 338: Assignment 4 (6 points)

Elliott Pryor

April 3, 2020

This assignment is due on **Friday, April 3, 11:30pm**. It is strongly encouraged that you use Latex to generate a single pdf file and upload it under *Assignment 3* on D2L. But there will NOT be a penalty for not using Latex (to finish the assignment). This is **not** a group-assignment, so you must finish the assignment by yourself.

Problem 1

Let $\mathcal{B}$ be the set of all infinite sequences over $\{a, b\}$. Show that $\mathcal{B}$ is uncountable, using a proof by diagonalization.

PROOF. By Contradiction.

Suppose $\mathcal{B}$ is countable. Then its elements could be ordered $b_1, b_2, b_3, ...$

$$\begin{aligned} b_1 &= \quad \textbf{a}\text{abbaa...} \\ b_2 &= \quad \text{a}\textbf{b}\text{abab...} \\ b_3 &= \quad \text{bb}\textbf{b}\text{aaa...} \\ &... \end{aligned}$$

We construct a new element b' by taking the elements on the diagonal and complementing them (if they were an 'a' make it a 'b' and if they were a 'b' make it an 'a'). Clearly $b' \in \mathcal{B}$. So then some $b_i = b'$, but by our construction b' differs from b_i at the ith spot. So $b' \neq b_i$. A contradiction. So $\mathcal{B}$ is not countable. $\square$

Problem 2

Let $T = \{(i, j, k) | i, j, k \in \mathbf{N}\}$. Show that T is countable.

Proof.

We need to show that there is some $f(x)$ that is a correspondence between T and $\mathbf{N}$. We construct this $f(x)$.

$\mathbf{N}$	$f(x)$
1	(1, 1, 1)
2	(2, 1, 1)
3	(2, 2, 1)
4	(2, 1, 2)
5	(2, 2, 2)
6	(3, 1, 1)
7	(3, 2, 1)
...	...

This mapping enumerates all possible values of T. For a given value of i it enumerates all (j, k) pairs in order s.t $j + k$ is minimized and $j, k \leq i$. Once it enumerates to $i = j = k$ it increases i by 1 and starts again. This is clearly one-to-one as it never lists the same point twice, and is clearly onto as it explores all possible values for T. Therefore we have constructed a correspondence between T and $\mathbf{N}$, then T must be countable.

$\square$

Problem 3

Let $INFINITE_{PDA} = \{< M > | M$ is a PDA and $L(M)$ is an infinite language$\}$. Show that $INFINITE_{PDA}$ is decidable.

Proof.

We construct a TM S for $INFINITE_{PDA}$. By Theorems 2.9 and 2.20 we can convert the PDA into a CFG G in Chompsky Normal Form. We then accept if there is some derivation $D \xrightarrow{*} xDy$ where x, y consist only of terminals. We reject otherwise. If $L(G)$ is finite, then there certainly is a finite number of derivations. If $L(G)$ is infinite then by the pumping lemma we will find a derivation $D \xrightarrow{*} xDy$ in a finite number of steps. So we can enumerate all the possible derivations.

$\square$

Problem 4

Let $\Sigma = \{a, b\}$. Define the following language ODD_{TM}:

$ODD_{TM} = \{< M > | M$ is a TM and $L(M)$ contains only strings of odd length $\}$.

Prove that ODD_{TM} is undecidable.

Proof. By Contradiction.

We assume that R decides ODD_{TM}. Then we construct a TM S that decides A_{TM}. We let $< M', w >$ be the input into A_{TM}.

First we construct a TM H on input x as follows:

1. If M' accepts w accept
2. if $x = aaa$ then accept, otherwise reject

So now we can construct S on $< M', w >$:

1. Construct H as above
2. Run R on $< H >$.
3. If R accepts: reject
   If R rejects: accept.

Then S decides A_{TM} a contradiction of theorem 4.11. Therefore ODD_{TM} is undecidable. $\square$

Problem 5

Show that EQ_{CFG} is undecidable.

Proof. By Contradiction

Assume that Turing Machine R decides EQ_{CFG}. We construct a TM S that decides ALL_{CFG}.

S: $< G >$

1. Construct CFG H such that $L(H) = \sum^*$
2. Run R on $< G, H >$
3. If R accepts $< G, H >$ then accept, if R rejects $< G, H >$ then reject

Then S decides ALL_{CFG} which is a contradiction of Theorem 5.13. Therefore EQ_{CFG} is undecidable.

$\square$

Problem 6

Show that EQ_{CFG} is co-Turing-recognizable.

Proof.

We construct a TM S that recognizes $\overline{EQ_{CFG}}$.

S: $< G, H >$ where G, H are CFGs

1. We generate a unique string $w \in \sum^*$.
2. We check if $A_{CFG} < G, w > \neq A_{CFG} < H, w >$. If true, then we accept
3. If false we generate a new string w and try again

We know by Theorem 4.7 A_{CFG} is decidable. So if $L(G) \neq L(H)$ S will eventually find some w that satisfies the condition. Therfore, S returns true on a yes instance as required for it to recognize $\overline{EQ_{CFG}}$.

$\square$

Problem 7

Problem 5.3 (page 239—third edition of Sipser).

If we number the cards 1-4. We have a sequence of 4, 4, 2, 1. This generates

$$\frac{aa|aa|b|ab}{a|a|a|abab} = \frac{aaaabab}{aaaabab}$$