CSCI 338: Assignment 4

Nathan Stouffer

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: We prove that $\mathcal{B}$ is uncountable by contradiction, that is, we assume that $\mathcal{B}$ is countable. Since $\mathcal{B}$ is countable, we know there exists a bijective map
$f : \mathcal{B} \longrightarrow \mathcal{N}$. We now construct an element $b \in \mathcal{B}$ that violates the surjective property of f, that is, there exists no $n \in \mathcal{N}$ such that $f(n) = b$.

For $i \in \mathcal{N}$, let $f(i) = a$. We then choose $b_i \neq a_i$ (where $*_i$ means the i^{th} element of $*$). To be explicit, if $a_i = a$, then $b_i = b$. Alternatively, if $a_i = b$, then $b_i = a$.

We have now constructed $b \in \mathcal{B}$ (the Codomain) such that there exists no $n \in \mathcal{N}$ (the Domain) where $f(n) = b$. This means that the map f is no longer surjective. Thus, in assuming that $\mathcal{B}$ is countable, we have encountered a contradiction. So it must be the case that $\mathcal{B}$ is uncountable.

□

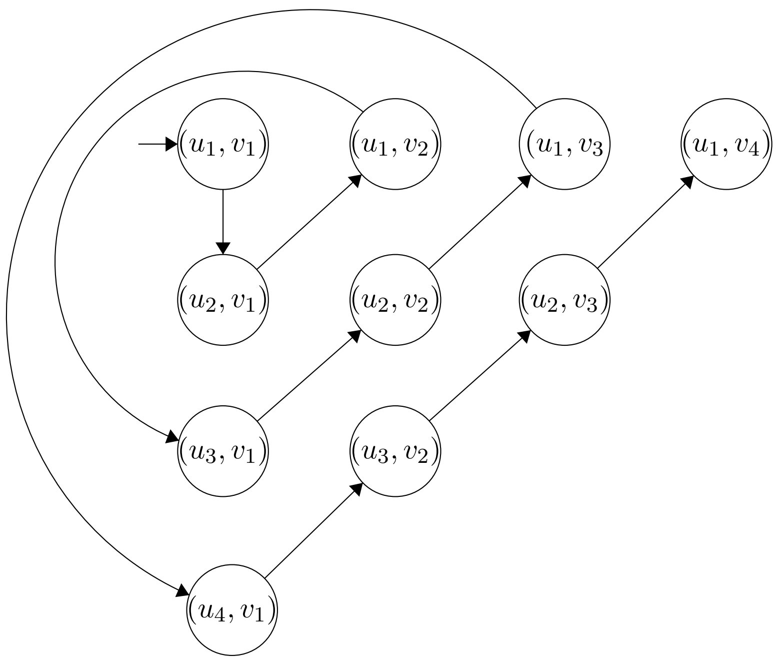
Problem 2

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

Proof: To show that T is countable, we must that there exists a bijection $f : \mathcal{N} \longrightarrow T$.

To show this, we first present a more general lemma: for U and V, two countably infinite sets, the cartesian product $U \times V$ is countable.

Since both U and V are countably infinite, there exists two bijective maps
$u : \mathcal{N} \longrightarrow U$ and $v : \mathcal{N} \longrightarrow V$ and all the elements of $U \times V$ can be listed as



In the same way that we showed that the set of rational numbers is countable, we can also show that $U \times V$ is also countable. So the lemma is shown to be true.

Using this lemma, we now show that T is countable. To do this, we first note that $T = \mathcal{N} \times \mathcal{N} \times \mathcal{N}$. Let $\mathcal{M} = \mathcal{N} \times \mathcal{N}$, then $\mathcal{M}$ is certainly countable (by the above lemma) since $\mathcal{M}$ is the product of two countable infinite sets. Additionally, $T = \mathcal{M} \times \mathcal{N}$. We can then apply the lemma again to say that T must also be countably infinite. A countably infinite set is countable so T is countable.

$\square$

Problem 3

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

Proof: We show that $INFINITE_{PDA}$ is decidable by giving a Turing Machine that decides it. Before doing so, we give some background knowledge.

By the pumping lemma, all infinite Context Free Languages have a derivation have a sufficiently long string $s = uvxyz$ such that uv^ixy^iz remains in the language for all $i \geq 0$. So there must exist a derivation step for s that looks like $V \overset{*}{\Longrightarrow} uVx$. Also recall that the pumping length (the minimum length for s) is finite. Now consider the TM A.

$A =$ "On input P a PDA:

1. Convert P to an equivalent CFG G in Chomsky Normal Form
2. Repeat for each natural number (referenced as n)
3. Compute all derivations of G with length n
4. If there are no derivations of length n, *reject*
5. If there is a derivation step of the form $V \overset{*}{\Longrightarrow} uVx$, *accept*"

We know that A will terminate since a finite CFL will have a finite number of derivations and all infinite CFL satisfy the pumping lemma. So, one of steps **4** and **5** must be true for some n. So A decides $INFINITE_{PDA}$.

□

Problem 4

Let $\Sigma = \{a, b\}$. Define the following language *ODD*$_{TM}$:
ODD$_{TM}$ $= \{\langle M \rangle \mid M$ is a TM and $L(M)$ contains only strings of odd length $\}$.

Prove that *ODD*$_{TM}$ is undecidable.

Proof: As is typical in proving undecidability, we use reduction in this proof. For the sake of contradiction, we assume that a TM R decides *ODD*$_{TM}$. Now consider the following Turing Machine (where M and w reference $< M, w >$ from A_{TM}).

$M_1 =$ "On input x:

1. If $|x|$ is odd, *accept*

2. If $|x|$ is even, run M on w and *accept* if M accepts w"

Then $L(M_1)$ will always contain at least all strings of odd length. However, $L(M_1)$ will also contain strings of even length if M accepts w. That is, the membership of M_1 in ODD_{TM} depends entirely on whether M accepts w. We will now construct a TM to decide A_{TM}:

$S =$ "On input $\langle M, w \rangle$:

1. Construct M_1 as above

2. Run R on input $\langle M_1 \rangle$

3. If R accepts, *reject*

4. If R rejects, *accept*"

So in assuming that R (a decider for *ODD*$_{TM}$) existed, we found that we could decide A_{TM}. Yet, this cannot be the case since we know A_{TM} to be undecidable. Therefore, no such R can exist and *ODD*$_{TM}$ must be undecidable as well.

$\square$

Problem 5

Show that EQ_{CFG} is undecidable.

Proof: First note that

$$EQ_{CFG} := \{\langle G_1, G_2 \rangle \mid G_1 \text{ and } G_2 \text{ are CFGs and } L(G_1) = L(G_2)\}$$

Now recall from Theorem 5.13 that $ALL_{CFG} := \{\langle G \rangle \mid G \text{ is a CFG and } L(G) = \Sigma^*\}$ is undecidable. Now, for the sake of contradiction, we assume that EQ_{CFG} is decidable by a TM R. Let us now give a TM S that decides ALL_{CFG}.

$S =$ "On input $\langle G \rangle$ a CFG:

1. Let H be a CFG where $L(H) = \Sigma^*$
2. Run TM R on input $\langle H, G \rangle$
3. If R accepts, *accept*. If R rejects, *reject*.

Under the assumption that a TM R that decides EQ_{CFG} existed, we were able to show that ALL_{CFG} was decidable. However, we know ALL_{CFG} to be undecidable so we have reached a contradiction. Thus, EQ_{CFG} must be undecidable.

$\square$

Problem 6

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

Proof: First recall that a language is co-Turing-recognizable if it is the complement of a Turing-recognizable language. Also recall that

$$EQ_{CFG} := \{\langle G_1, G_2\rangle \mid G_1 \text{ and } G_2 \text{ are CFGs and } L(G_1) = L(G_2)\}$$

So our task is then to show that $\overline{EQ}_{CFG}$ is Turing recognizable. But what is $\overline{EQ}_{CFG}$? It is shown below

$$\overline{EQ}_{CFG} = \{\langle G_1, G_2\rangle \mid G_1 \text{ and } G_2 \text{ are CFGs and } L(G_1) \neq L(G_2).$$

Now note that the alphabet Σ^* is countably infinite (there is a correspondence between the natural numbers and Σ^*). Also note that A_{CFG} (from Theorem 4.7) decides whether a given CFG accepts a given string. With this in mind, we now give a Turing machine that recognizes $\overline{EQ}_{CFG}$. Consider the following Turing Machine A.

A = "On input G_1 and G_2 (both Context Free Grammars):

1. Repeat the steps below

2. Take w, the next member of Σ^* and compute A_{CFG} for $\langle G_1, w\rangle$ and $\langle G_2, w\rangle$

3. If above results differ, *accept*"

The TM A recognizes $\overline{EQ}_{CFG}$, so EQ_{CFG} is co-Turing-recognizable.

$\square$

Problem 7

Find a match in the following instance of the Post Correspondence Problem.

$$\left\{ \left[\frac{ab}{abab}\right], \left[\frac{b}{a}\right], \left[\frac{aba}{b}\right], \left[\frac{aa}{a}\right] \right\}$$

Proof: A match in the Post Correspondence Problem is a list of dominos (allowing repetitions) such that reading the top gives the same result as reading the bottom. With this definition, the following is a match

$$\left[\frac{aa}{a}\right] \left[\frac{aa}{a}\right] \left[\frac{b}{a}\right] \left[\frac{ab}{abab}\right] = \frac{aaaabab}{aaaabab}$$

$\square$