Homework 4
Mike Gao
260915701
Prof. Prakash Panangaden

Question 1

This is decidable for a given input. $T = (Q, \Sigma, \Gamma, \delta, q0)$ be the turing machine in question, and s the given input. There are only $598 * |Q| * |\Gamma|^{598}$ configurations that T can be in that the machine has never use more than 598 cells. So we run the machine for $598 * |Q| * |\Gamma|^{598} + 1$ steps, Using the pigenhole principle, we know that if a configuration is repeated, the machine never use more than 598 cells, otherwise, the machine will use more than 598 cells.

Question 2

We can run membership testing algorithm in $L(G_1)$ and $L(G_2)$ on every word, and such tester is sure to terminate. If we find a word both in $L(G_1)$ and $L(G_2)$ we know the statement is false. However, the case is undecidable when we are determining whether the statement is true, because $VALCOMPS_2(M, w)$ is empty only if turing machine M does not accept w. Thus we have a reduction to $\neg A_{TM}$ which is undecidable.

Question 3

The parameters to determine the motion of the submarine is its starting location (x, y), its velocity v and direction.

At each step n we make a guess of those parameters. For example, if we guess direction 'up', we will get $(x, y + n * v)$. We simply try every possible combination.

We will use the idea of dovetailing. Since (x, y) is an integer pair, and v is a natual number, we can map it to a single natual number n using a variation of cantor's paring function $f : \mathbb{Z} \times \mathbb{Z} \times \mathbb{N} \times \{0, 1, 2, 3\} \to \mathbb{N}$. Where 0 represents up, 1 represents right, 2 represents left, 3 represents down.

Question 4

4.1

Suppose a DFA $D = (Q, \Sigma, \delta, F, q_0)$ accepts L. We define a new DFA $D' = (Q, \Sigma, \delta, F', q_0)$ which has the same start state, transition and alphabet as D, the only difference is the final state, where $F' = \{q \in Q | \delta(q, w) \in F\}$. This DFA recognizes L/w so it is regular.

4.2

Consider the language $L = N\#\Sigma^* \cup \Sigma^*\#L(G)$. We know Σ^* is regular. Since N is context free, we have that $N\#\Sigma^*$ and $\Sigma^*\#L(G)$ are also context free. Since the union of two context free languages are context free, we get that L is also context free.

Now we have two cases:

$L(G) = \Sigma^*$, the language L is just $\Sigma^*\#\Sigma^*$ which is regular

$L(G) \neq \Sigma^*$, and assume there exist $s \in \Sigma^* \wedge s \notin L(G)$. Consider $L/\#s = N$. Assume N is not regular, so $L/\#s$ and hence L is not regular. So we know that L is regular if and only if $L(G) = \Sigma^*$ But this is known to be undecidable.

Question 5

5.1

$R \subseteq L$ is undecidable.

It is a well known problem that $L = \Sigma^*$ is undecidable. Since Σ^* is a regular language, it is undecidable whether $R \subseteq L$

5.2

$L \subseteq R$ is decidable.

$L \subseteq R$ iff $L \cap \overline{R} = nil$, So $L \cap \overline{R}$ is context free, and therefore $L \subseteq R$ is decidable.

Question 6

Your favorite language, OCaml.

```
(fun s -> Printf.printf s (string_of_format s))
    "(fun s -> Printf.printf s (string_of_format s)) %S;;";;
```