Section Handout 6

Problem One: Programming Turing Machines

In Problem Set 7, you're asked to augment the **WB6** language up to a language called **WB8**, which supports a finite number of unbounded counters. The new commands you'll be adding are

- `C++`, which increments counter *C*;
- `C--`, which decrements counter *C*;
- `If Zero(C), go to L`, which goes to line *L* if counter *C* is zero;
- C_1 `:=` C_2, which sets counter C_1 equal to counter C_2;
- C_1 `:=` C_2 `+` C_3, which sets counter C_1 equal to $C_2 + C_3$; and
- `If` C_1 `=` C_2`, go to L`, which goes to line *L* if counters C_1 and C_2 have the same value.

Now, suppose that you want to augment **WB** one more time by creating the language **WB9**, which is **WB8** with the addition of one extra command:

- `v := Tape[C]`, which sets variable *v* to hold the *C*th cell of Track 1, Tape 1; and

Describe how to convert an arbitrary **WB9** program into a **WB8** program. Your description should address the following questions:

1. In converting from **WB9** to **WB8**, will you need extra stacks, tracks, variables, tape symbols, tapes, or counters? If so, how many of each will you need and why?
2. In converting from **WB9** to **WB8**, will you need to set up the tapes, stacks, tracks, variables, or counters in any way before beginning the program? If so, how and why?
3. How will you translate `v := Tape[C]` into equivalent **WB8** commands? You should give the specific commands with which you will replace this command.

Problem Two: Nondeterministic Algorithms

Prove that the **RE** languages are closed under homomorphism. That is, if $L \subseteq \Sigma_1^*$ is an **RE** language and $h^* : \Sigma_1^* \to \Sigma_2^*$ is a homomorphism, then $h^*(L) = \{w \in \Sigma_2^* \mid \exists x \in L.\ h^*(x) = w\}$ is an **RE** language. As a strong hint, you might want to use a nondeterministic Turing machine.

Problem Three: Unsolvable Problems

i. Consider the language $ENTER = \{ \langle M, w, q \rangle \mid$ TM M enters state q when run on string $w \}$. Prove that $ENTER \notin$ **R** by showing if $ENTER \in$ **R**, then $A_{TM} \in$ **R**.

ii. Consider the language $NOENTER = \{ \langle M, w, q \rangle \mid$ TM M does not enter state q when run on string $w \}$. Prove that $NOENTER \notin$ **RE** by showing if $ENTER \in$ **RE**, then $L_D \in$ **RE**.