HW11

Shane Drafahl

16 October ,2017

1.
$L = \{\ p(M)p(w)$: M uses a finite number of tape cells when running on input w }
This language is acceptable but not decidable. It is not decidable because the size of the tape is unknown. We can create a turing machine that simply gives M the input w and L and check that it used a finite number of tape cells. Essentialy $p(M)p(w)$ is the same as the problem $\{\ P(M)p(w) : M \searrow w\}$ which we know is not decidable.

$L = \{\ p(M)p(w)01^n0$: M uses at most n tape cells when running on input w }

This language is decidable because we know the number n of tape cells that you can check. If it loops we can check if the head goes beyond n cells. If it loops over the same cells we can count the number of times it goes over the cells.

2.

$\{\ p(M) : |L(M)| \leq 10\}$

This language is not turing acceptable. The turing machine might go forever to comfirm that there are 10 strings or less in the machine.

$\{\ p(M) : |L(M)| \geq 10\}$

This language is acceptable because if there is at least 10 words in the language it can be accepted. This is not decidable because its complement is not turing acceptable.

$\{\ p(M)p(w) : M \searrow w$ in 10 steps or less $\}$

This is turing decidable because it has a finite number of steps so we know for sure it halts. If in 10 steps it has not printed a Y then it would print a N by the last step.

$R = \{\ p(M)p(w) : M \searrow w$ in 10 steps or more $\}$

This language halts so we know it is acceptable. We will use reduction to prove it is not decidable. We know from the class notes that $\{\ p(M)p(w) : M \searrow w\}$. If we build a machine M'. This machine first takes input w and checks that it is at least 10 steps. If not it adds more steps to the tape. This tape is then given to R. If R returns a Y or a N then M' returns a Y or a N otherwise it loops. So therefore R reduces to $\{\ p(M)p(w) : M \searrow w\}$ so the language is only accepting by not decidable.

3. Use reduction to prove that the language is not decidable.

$L = \{\ p(M_1)p(M_2) : L(M_1) \subseteq L(M_2)\}$

Using proof by contradiction we will assume that L is decidable. We know that it is not turing decidable for a turing machine to determine if two sets are equal. We will say that a machine that checks for set equality is R. Set equality for A and B for example would need to check if $A \subseteq B$ and $B \subseteq A$. Therefore we would know that checking $L(M_1) = L(M_2)$ is not decidable. This would mean either checking $L(M_1) \subseteq L(M_2)$ or $L(M_2) \subseteq$

$L(M_1)$ is undecidable. Our assumption is that $L(M_1) \subseteq L(M_2)$ is decidable so we would have to assume that $L(M_2) \subseteq L(M_1)$ is not decidable but this is a contradiction because if this was not decidable then $L(M_1) \subseteq L(M_2)$ cant be decidable so therefore L is undecidable.