HW10

Shane Drafahl

16 October ,2017

1. For union for L_1 and L_2 I have machines M_1 and M_2. Given an input $x \in L_1 \cup L_2$ I give x to M_1 and M_2 and if its accepted by either machine that means they are accepted. Similar to intersection but both machines must be accepted. I would build this turring machine by getting two tape and with input x and then dove tailing the two tape into each relative turring machine. For example below is a representation of the tape where $b_1 = a_1$ and so on and so forth.

So for union

```
input x in Sigma*
Run M1 on x and then run M2 on x
Accept if M1 or M2 both accept
```

intersection

```
input x in Sigma*
Run M1 on x and then run M2 on x
Accept if M1 and M2 both accept
```

Reversal

```
input x in Sigma*
Assign y:= reverse(x)
Run M1 on x and then run M2 on y
Accept if M1 and M2 both accept
```

For reversal I would simply copy the input from one side of the tape to the other in reverse order. I would then have the same turring machine dove tail both peices of tape and if both are accepted then the whole thing is accepted.

Reversal

```
input x in Sigma*
Assign y:= reverse(x)
Run M1 on x and then run M2 on y
Accept if M1 and M2 both accept
```


2. Given a string that z = xy and you have M_1, M_2. You create every possible combination of ways to divided up z onto multiple peices of tape. For example if x = ab and y = cd then you have
$T[a]_1...T[bcd]^2$
$T[ab]_3...T[cd]_4$
$T[abc]_5...T[d]_6$

Where $T[]_n$ is a peice of tape. M_1 dove tails for every odd indexed tape and M_2 does every even. If both of the machines have at least one accepting string then the whole thing is accepted.

concatination

```
input x in Sigma*
Assign t1[], t2[]:= sub(x) // returns every possible sub string of x to arrays o
Run M1 for all t1 and run M2 for all t2
Accept if M1 and M2 have at least Accept one string each.
```

3. a

For this turring machine it would depend on if it goes left,right, or does nothing
$\delta(a_n, a_n) = (a_{n+1}, R)$ and push (GO RIGHT)
$\delta(a_n, a_n) = (a_{n-1}, L)$ (GO LEFT)
$\delta(a_n, a_n) = (a_n, a_n)$ (Do Nothing)
b.

To prove that all languages accepted by PA's can be accepted by M we will prove that the accepted languages of $PA \subset M$. We can prove this because we can simulate a PA with M by starting the head on the left most data on the read only tape and only using Γ_1 and reading from left to right.

We can prove that it is equivalant to a turring machine because $M \subset T$ and $T \subset M$. In this case we can prove $M \subset T$ because a Turring machine can use two peices of tape to simulate the two stacks and a third as the read only tape. We can also prove $T \subset M$. We can simulate a turring machine because we can put all the contents left of the head in Γ_1. and to the right Γ_2. If we want to move right we pop Γ_2 and left Γ_1. Therfore they are equal. We also know that all languages accepted by any type of PA including NPDA or determinsitic ones can also be accepted by a turring machine so that is another reason M can accept the same languages as a class of automata.