Assignment sheet 8

Exercise 1

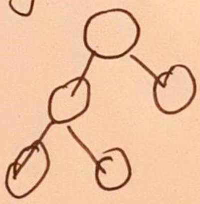
$w \in L(G)$, G be in Chomsky normal form.
with $|w| = n \geq 1$.

We know that chomsky normal form is described ~~as~~ from the rules:-

$A \to BC$
$A \to a$
$S \to \varepsilon$

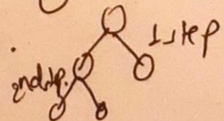
where $A, B, C \in V$ and, $a \in \Sigma$, S is starting variable.

Chomsky normal form only contains the ~~rules below~~ aforementioned rules. So, to create a word w with $|w| = n$, we can see it as a binary tree



(Here we have created a word with $n = 3$).

We see that converting the starting root ~~to other additional 4 nodes required~~ to a tree of 5 nodes, where, there are 3 children, we needed 2 steps.



This can be seen where we have n children, we need $(n-1)$ steps, (since every time we add a newer element to a leaf when we branch out).

n children can be seen as the n variables where we apply transformation, $A \to a$. so, further n steps is required.

In total, we require $(n-1+n)$ = $2n-1$ steps.

Exercise 2

$ALL_{DFA} = \{\langle A\rangle \mid A$ is a DFA and $L(A) = \Sigma^*\}$. Show that ALL_{DFA} is decidable.

let us first see a ~~tur~~ language,

$ALL_{DFAacc} = \{\langle A\rangle \mid A$ is a DFA that accepts $w \in \Sigma^*\}$.

For, ALL_{DFAacc} let us construct a turing machine M_{DFAacc}

The algorithm for this turing machine would be as follows:

1) Initialize the tapes.
2) If the symbol under the head of T1 is ⊔, compare current state on T2 to list of accepting states on T2. If it matches, accepts, otherwise reject.
3) Read symbol under the head of T1 and move the head to the right.
4) Find for current state on T2 and current input symbol the transition in T3.
5) Store new state ~~on~~ to T2.
6) Go to 1. We can see that M_{DFAcc} is a decider.

$ALL_{DFArej} = \{\langle A\rangle \mid A$ is a DFA that rejects $w \in \Sigma^*\}$

let us suppose the turing Machine to be M_{DFArej}

We can see that, applying the same algorithm as before but with line 2 as

"If the symbol under the head of T_1 is ⊔, compare current state on T2 to list of accepting states on T2. If it matches, rejects otherwise accept"

We can effectively create a turing machine that accepts all FAs that rejects a string w.

Combining the two deciders, $M_{DFA_{acc}}$ and $M_{DFA_{rej}}$ into a single turing machine M_{DFA}, we can ~~see~~ formulate an algorithm :-

1) Accept if either of the turing machine accept.
2) Reject if both reject.

~~We~~ ~~Hence~~ M_{DFA} is a decider for the language $\langle A \rangle$ because no ~~loop is present~~ infinite loop is present in either of the algorithm due to the finite number of symbols.

Exercise 3

$A_{\varepsilon}CFG = \{\langle G\rangle \mid G$ is a CFG that generates $\varepsilon\}$. Show that $A_{\varepsilon}CFG$ is decidable.

To show that $A_{\varepsilon}CFG$ is decidable, we require a turing machine that accepts such CFG ~~and~~ halts when ~~acce~~ the CFG accepts or rejects.

We describe the turing machine with the following algorithm:
let us say the turing machine is M_{CFG}

1) Store G appropriately on tapes.

2) Mark all εs on the tape

3) IF none exist, reject

4) Mark all symbols A, such that in the rule $A \to \ldots a \ldots$, , $a \in \Sigma$, ~~and~~ or, $a \in V$ were previously marked.

5) IF any symbol was marked go to 4.

6) IF the startsymbol 'S' was marked, accept; otherwise reject.

Showing M_{CFG} is decider,

⇒ Possible looping occurs in step 4 of the given algorithm. Since we have finite number of variables and terminals, infinite looping is not possible

Hence M_{CFG} is a decider.