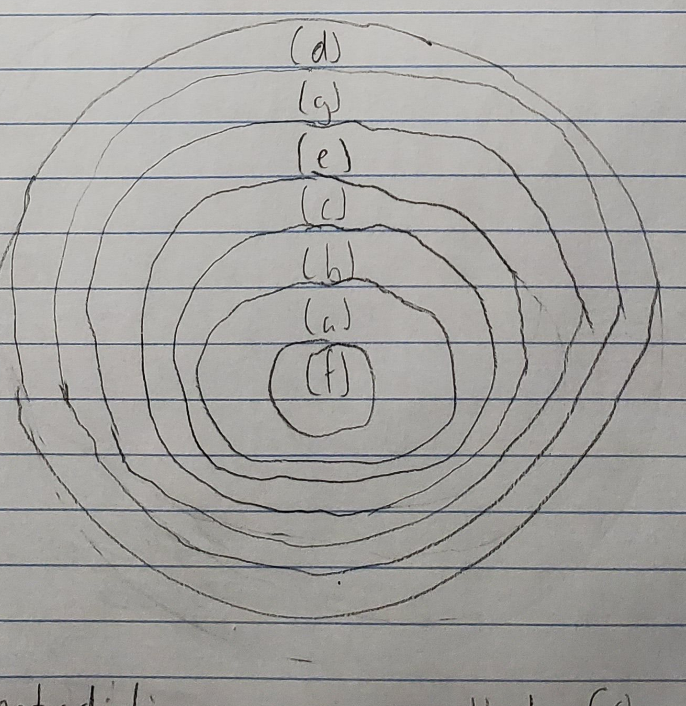
Alex Jacobs
Prof. Cao

HW#11
11/22/20

1) a) We cannot use the same way because a turing recognizable language must constantly halt step-by-step instead of reviewing the entire string.

b) We run the same procedure as 3.15(a), except we only accept after both machines accept and reject if either rejects

2) a) P
b) The CFL's
c) Decidable Languages
d) Languages whose complement is Turing-Recognizable
e) " " is decidable
f) Regular Languages
g) Turing recognizable languages



3) For contradiction, we can say that EQ is decidable. We construct CFG G'' such that $L(G'') = \Sigma^*$. $G'' = S \Rightarrow xS \mid \varepsilon,\ \forall x \in \Sigma$.
If we submit $\langle G, G'' \rangle$ to the Turing machine, we get a decider.
This is a contradiction, which means that EQ is undecidable.

4) 1) Undecidable: If T1 accepts {a, aa, aaa} and T2 accepts a^*, then T2 satisfies the property and T1 doesn't.

2) Undecidable: Let S = set of Turing recognizable and decidable languages. L(T1) is any decidable language and L(T2) is any recognizable, but not decidable language. L(T1) is in the set, but L(T2) is not.

3) Undecidable: Not a language property. Rice's theorem cannot be applied.

4) Undecidable: Let S = set of turing-recognizable languages that have 100 strings. L1 = a^*, L2 = {a, aa, aaa, ... (100 a's)}. L2 is in the set, but L1 is not.

5) Decidable: No language can have a negative amount of strings.

6) Undecidable: Let S = set of languages that satisfy {L(M) : w^R if contains w}. L1 = {aa, bb, aba, bab, babab}, L2 = {ab, ba, bba}. L1 is in S, but L2 is not.