CSE 355: Intro. to Theoretical Computer Science
HW #6: Decidability and Reducibility

- This assignment is worth 100 points and 5% of your final grade.
- It is due on **Friday, Apr. 23, 2021 at 11:59pm** Arizona time.
- Copying answers from any sources (online, book, last-semester's notes, ...) without proper citation is considered a violation of academic integrity and will be dealt with accordingly. • Answers can be provided in handwritten or typed form.
- Unreadable and unclear answers will be graded with 0 point.
- Scan and submit your homework on Canvas **as a single PDF file** (in case you use Apps. taking pictures of your answers, please make sure they are neat and readable)
- **No late submissions will be accepted!** Submission through emails will **NOT** be accepted!

1. [10 pts] Let $A_{\varepsilon-CFG}$ = {< G >| G is a CFG that generates ε}, show that $A_{\varepsilon-CFG}$ is decidable. '

First, convert G to an equivalent CFG.

Let G' = < V , Σ , R , S > Where S is the start state. If S → ε is a rule in G', then G' generates ε . So G will also generate ε because of L(G) = L(G'). The only way we can have ε $\in L(G')$ is if G' has the rule of S → ε in R. Thus if G' does not include rule S → ε then ε $not \in L(G')$.

2. [10 pts] Let B = {< R, S >| R and S are regular expressions and L(R) ⊆ L(S)}, show that B is decidable.

We know that R and S will generate regular languages. We need to show whether L(R) ⊆ L(S). Since regular language is closed under complement, intersection and L(R) - L(S) should be empty. The conversion should be RE → NFA → DFA. Now we check if L(C) = L(R) ∩ L(S)' is empty or not.

The following TM F decides B:

1. Construct DFA C as described
2. Run TM T
3. If T accepts, accept, If T rejects, reject.

3. [20 pts] Given a Turing machine M, we consider the problem of deciding whether this Turing machine has an equivalent finite automaton or not, formulate this problem as a language, then prove that it is undecidable.

Let EQ_{DFA} = {<M, A> | M and A are DFAs}. In order to prove M and A are equivalent DFA. It needs to satisfy L(M) - L(A) being empty L(C) = L(M) ∩ L(A)'. Where C is a new DFA construct from M and A. In order for it to be undecidable. L(M) - L(A) should not be empty, which means they have to recognize the same language and L(C) = L(M) ∩ L(A)' will accept nothing. And it will make it undecidable.

4. [20 pts] Let G_1 and G_2 be two CFGs, formulate the problem of deciding whether G_1 and G_2 generate the same set of strings as a language. Then prove the problem is undecidable.

Let EQ_{CFG} { <G_1, G_2> | G_1 and G_2 are CFGs and $L(G_1) = L(G_2)$}. In order for them to be equivalent, they need to be accepted by $L(C) = L(G_1) \cap L(G_2)'$. However, this is not possible because CFG is not closed under complementation and intersection.

5. [20 pts] Let $SUB_{CFG-DFA}$ = {< G,D >| G is a CFG and D is a DFA and $L(G) \subseteq L(D)$}. Is SUBCFG−DFA decidable? If your answer is yes, describe a TM that decides it; if your answer is no, prove it.

$SUB_{CFG-DFA}$ is decidable if $L(G) \cap L(D)'$ is empty. We know that the intersection of a regular language and a context-free language will result in context-free language. Which we could construct a CFG for it.

S = On input <G,w> : Convert G into grammar in CNF

If w accept, accept, if not, reject.

So $SUB_{CFG-DFA}$ is decidable.

6. [20 pts] Let $EQ_{\text{TM-MTM}}$ be the problem of determining whether the language of an ordinary Turing machine and the language of a multitape Turing machine are equivalent. Convert this problem into a language, and then show that $EQ_{\text{TM-MTM}}$ is undecidable.

Let M_1 = TM and M_2 = MTM

$EQ_{\text{TM-MTM}}$ {<M_1, M_2> | $L(M_1) = L(M_2)$}

Let TM R decides EQ_{TM} and construct TM S to decide E_{TM}

S = On input <M> where M is a TM

1. Run R on <M, M_1> Where M_1 is a TM that rejects all input.
2. If R accept, accept, if not, reject.

If R decides EQ_{TM} and S decides E_{TM} and E_{TM} is undecidable so EQ_{TM} will also be undecidable.