CS 4820, Spring 2019 **Homework 8, Problem 1**
Name: Kevin Klaben NetID: kek228

(1) *(15 points)* Define $(\leq k)$-CNF to be a CNF with each clause containing at most k literals, and a k-CNF to be a CNF with each clause containing exactly k literals.

(1a) *(8 points)* Consider a variant of 3SAT, which we call $(2,3)$-LSAT, defined in the following way: An input instance of $(2,3)$-LSAT is a (≤ 3)-CNF on n variables, with the additional restriction that each literal appears at most 2 times, and the output is 'Yes' if the given instance is satisfiable and 'No' otherwise. Either (i) design an efficient algorithm that finds a satisfying assignment to an input instance of $(2,3)$-LSAT or outputs 'Impossible' in the case that the input instance is unsatisfiable, or (ii) prove that it is NP-complete.

(1b) *(7 points)* Consider another variant of 3SAT, which we call $(3,3)$-VSAT, defined in the following way: An input instance of $(3,3)$-VSAT is a 3-CNF on n variables, with the additional restriction that each variable appears at most 3 times, and the output is 'Yes' if the given instance is satisfiable and 'No' otherwise. Either (i) design an efficient algorithm that finds a satisfying assignment to an input instance of $(3,3)$-VSAT or outputs 'Impossible' in the case that the input instance is unsatisfiable, or (ii) prove that it is NP-complete.

(1a)
It can be shown that each an instance of 3SAT can be reduced to an instance of (2,3)-LSAT. The key restriction comes in restricting each literal to appear at most 2 times. An instance of 3SAT suppose there is some variable x in $x_1 ... x_n$ which appears ¿2 times in the instance of 3SAT. We can iterate over each appearance from appearance 1 to appearance k and replace each instance with a new variable $x_{n+1} ... x_{n+k}$ within the original clauses. To ensure that these new variables are restricted to take on the same value, additional clauses of
$(\bar{x_{n+1}} \bigvee x_{n+2}) \bigwedge (\bar{x_{n+2}} \bigvee x_{n+3}) \bigwedge ... \bigwedge (\bar{x_{n+k-1}} \bigvee x_{n+k}) \bigwedge (\bar{x_{n+k}} \bigvee x_{n+1})$
With these additional clauses we ensure that the literals can only take on one possible value as at least one of these clauses will not be satisfied if any of these new symbols do not take on the same value. As this is now an instance of (2,3)-LSAT that contains the same information as an instance of 3SAT, we can conclude that (2,3)-LSAT is NP-complete as 3SAT is NP-complete.

(1b) The extension for part a can similarly be applied to 3SAT to satisfy this restriction, we can again conclude that $(3,3)$-VSAT is NP-complete. $(3,3)$-VSAT is less resticted than (2,3)-LSAT as it can include up to three instances of a literal. However, our extension in the proof from part a ensures each literal only occurs twice and thus, satisfies both $(3,3)$-VSAT and (2,3)-LSAT thus $(3,3)$-VSAT is NP-complete.