Exercise 9.9. Consider the following special case of SAT, which we will call *k-occurrence-SAT* for a fixed parameter $k \in \mathbb{N}$. The input consists of a SAT formula $f(x_1, \ldots, x_n)$ in CNF such that every variable x_i appears (as is, or negated) in at most k clauses. The problem is to decide whether there is a satisfying assignment. For $k = 3$, either (a) design and analyze a polynomial time algorithm, or (b) show that a polynomial time algorithm for k-occurrence-SAT implies a polynomial time algorithm for (CNF-)SAT.[1]

[1] As a warmup, it might be helpful to first consider the case $k = 5$. If you figure out 5-occurrence SAT, but don't figure out 3-occurrence SAT, we will give partial credit for a solution to 5-occurrence SAT.

Solution. We claim that for $k = 3$, the existance of a polynomial-time algorithm for k-occurrence-SAT implies a polynomial-time algorithm for SAT. To see this, we propose a polynomial-time reduction from any CNF-SAT formula $f(x_1, ..., x_n)$ to an corresponding 3-occurrence-SAT formula.

Reduction. For each variable x_i in f, let $\hat{k}$ be the number of occurrences of x_i in f.

If x_i has $\hat{k} \leq 3$ occurrences, then f already meets the constraints of 3-occurrence-SAT.

In the case that there exists $x_i \in f$ with $\hat{k} > 3$ occurrences, we split x_i into multiple variables. We chose to split x_i into k equivalent variables $x_{i1} = x_{i2} = ... = x_{ik}$ such that j^{th} occurrence of x_i in f can be replaced by a new variable x_{ij}, $1 \leq j \leq \hat{k}$.

We also have to enforce equality between all the new variables x_{ij}. This can be done by simply appending the clauses

$$(\overline{x}_{i1} \vee x_{i2}) \ \wedge \ (\overline{x}_{i2} \vee x_{i3}) \ \wedge \ ... \ \wedge \ (\overline{x}_{i\hat{k}} \vee x_{i1})$$

to the boolean formula. The addition of these $\hat{k}$ clauses adds 2 more occurrences of each x_{ij}, for a total of 3 occurrences.

When we perform the above substitution for all x_i with more than 3 occurrences, notice that the size of the new formula, call it f', is linearly proportional to the size of the original formula f, whence the reduction is polynomial-time. ■

Correctness. Suppose we apply the reduction above to a CNF boolean formula f to get a 3-occurrence boolean formula f'. We will prove that f satisfiable $\iff$ f' satisfiable.

($\implies$) If f is satisfiable, then by definition there exists some SAT assignment $A : \{x_i\}_{i=0}^n \to \{0, 1\}$ satisfying $f(x_1, ..., x_n)$. By our construction above, the corresponding 3-occurrence formula f' is also satisfied by assigning each "duplicate" variable x_{ij} to the same value as the original variable x_i.

($\impliedby$) If f' is satisfiable, then by definition there exists some SAT assignment $A : \{x_i\}_{i=0}^n \to \{0, 1\}$ satisfying f'. We also know if we have "duplicate" variables $x_{i1}, ...x_{ik}$ for some x_i, then $A(x_{i1}) = \ ... \ = A(x_{ik})$ is enforced by the equality clauses in f'. Hence, if we set the corresponding variable x_i in f to have the same assignment, f is also satisfied. ■

□