Exercise 10.3. After your glorious app PikPok hit number 1 in the app store, you're preparing for version 2. Obviously, it needs to be great.

You've gathered a list of k features $F_1, \ldots, F_k$ that you could potentially add to version 2. However, there are complicated dependencies and requirements among them so you don't necessarily want to add all of them. There are 3 types of specifications defined over pairs of features F_i and F_j:

1. Requirements: Your app must include either F_i or F_j.
2. Conflicts: You cannot include both F_i and F_j.
3. Dependencies: If you include F_i, then you must include F_j.

Collectively, we call requirements, conflicts, and dependencies the *feature specifications*. The feature specifications are given in list form. The high-level task is to decide which of the features to implement, based on the given feature specifications. We have two versions of the problem. For each of the problems [below], either (a) design and analyze a polynomial time algorithm (the faster the better), or (b) prove that a polynomial time algorithm would imply a polynomial time algorithm for SAT.

Exercise 10.3.1. In the idealistic feature selection problem, the task is to decide if there is a subset of features that satisfies all the feature specifications.

Solution. We claim that the case of idealistic feature selection reduces to a 2-SAT problem and therefore has a polynomial time solution.

For each feature F_i, define a boolean variable f_i such that f_i true $\iff F_i$ implemented. Then, notice that the constraints for each feature specification can be translated into the language of 2-SAT.

1. Requirement: $(f_i \vee f_j)$
2. Conflict: $(\overline{f_i} \vee \overline{f_j})$
3. Dependency: $(\overline{f_i} \vee f_j)$

Thus, the conjunction of all of the feature specifications as 2-variable clauses gives us a 2-SAT problem. We can therefore determine satisfiability of the feature specifications in polynomial time by using the algorithm for 2-SAT.

Correctness. Suppose 2-SAT returns true. Then there exists some assignment $A : \{f_1, ..., f_k\} \to \{0, 1\}$ satisfying the conjunction of all the feature-spec clauses. Including each feature F_i if and only if $A(f_i) = 1$ gives us a subset of features that satisfies all feature specifications.

On the other hand, if 2-SAT returns false, then there is no assignment on $\{f_1, ..., f_k\}$ that satisfies the conjunction of all the feature-spec clauses. Hence, there is no subset of features that would satisfy all of the feature specifications. ■

□

Exercise 10.3.2. In the realistic feature selection problem, the task is to choose a subset of features that satisfies the maximum number of feature specifications

Solution. A realistic feature selection implies a polynomial time algorithm for SAT; we prove this creating a polynomial-time reduction from any Max 2-SAT problem to a realistic feature selection problem.

Consider a generic Max 2-SAT instance $f(x_1, ..., x_n)$ with m clauses.

If we let $F_1, ..., F_n$ be features, where each x_i is true if and only if its corresponding feature F_i is chosen, then Max 2-SAT on $f(x_1, ..., x_n)$ directly corresponds to a realistic feature selection problem on $F_1, ..., F_n$. In particular, each clause constructed with variables x_i and x_j directly corresponds to a type of feature dependency:

1. $(x_i \vee x_j)$: app must include either F_i or F_j (requirements).
2. $(\overline{x}_i \vee \overline{x}_j)$: cannot include both F_i and F_j (conflicts).
3. $(\overline{x}_i \vee x_j)$: if F_i is included, F_j must be included (dependencies).
4. $(x_i \vee \overline{x}_j)$: if F_j is included, F_i must be included (dependencies).

Hence, we can create a list of feature specifications on $F_1, ..., F_n$ from $f(x_1, ..., x_n)$ in polynomial time.

Therefore, if realistic feature selection on $F_1, ..., F_n$ has a polynomial-time solution, then so does Max 2-SAT on $f(x_1, ..., x_n)$

Correctness. Let $S \subseteq \{F_1, \ldots, F_n\}$ be a subset of features, and define 'F_i chosen' $\iff$ 'x_i is true'. Then, the clauses $\{C_j\}_{j=1}^k$ are satisfied $\iff$ S meets the corresponding feature requirements.

Suppose the Max 2-SAT instance $f(x_1, ..., x_n)$ has at most k satisfiable clauses, and label them $C_1, C_2, ..., C_k$. Assume ad absurdum that S satisfies more than k feature specifications. Then, the corresponding 2-SAT instance would satisfy more than k clauses, contradicting our assumption that the number of satisfiable clauses is bounded above by k. Hence, if no assignment satisfies more than k clauses, there can not be some subset of features S satisfying greater than than k feature specifications.

Now, suppose the Max 2-SAT instance $f(x_1, ..., x_n)$ has greater than k satisfiable clauses. Assume ad absurdum that S satisfies less than k feature specifications. Then, f would necessarily satisfy less than k clauses, which contradicts the assumption that f has greater than k satisfiable clauses. Thus, there can not be some subset of features S satisfying less than k feature specifications. ■

□