The Theory of Computation
Bernard M.E. Moret
Addison-Wesley-Longman, 1998

Solution Manual
Chapter 7, Additional Problem 3

Membership in NP is clear. We prove this problem NP-complete by transforming *3SAT* to it.

Given an instance of *3SAT* with variables $V = \{x_1, \ldots, x_n\}$ and clauses $C = \{\{\hat{x}_i, \hat{y}_i, \hat{z}_i\}, 1 \leq i \leq r\}$ (where $\hat{x}$ is used to denote one of x or $\overline{x}$), we construct an instance S, $C \subseteq 2^S$ of *SS* by letting $S = \{x, \overline{x} | x \in V\} \cup \{u\}$ and $C = \{\{x, \overline{x}\} | x \in V\} \cup \{\{\hat{x}, \hat{y}, \hat{z}, u\} | \{\hat{x} + \hat{y} + \hat{z}\}$ is a clause$\}$. The extra variable u is to allow the splitting of sets that correspond to clauses for which an assignment sets all literals true; it also determines which side of the solution (partition) to *SS* corresponds to the assignment true in *3SAT*. Our transformation is obviously polynomial-time.

Given a solution to *3SAT*, we get a solution to *SS* by putting in one set all literals assigned false plus the extra variable, and on the other side all the literals assigned true. Since each clause is true, at least one literal in it is assigned true and thus goes in the 'true' partition side; the extra variable u always goes on the other side; this guarantees that all subsets derived from the clauses get split. The other sets (the truth setting sets) are obviously split by any truth assignment. Conversely, a solution to *SS* must split those last sets, thereby guaranteeing that no variable can be on the same side as its complement. We shall assign false to these literals that are on the same side as u, true to those on the other side. Since all clause-derived subsets are split, at least one literal in them is not on u's side and thus is assigned true, thereby satisfying the clause. Hence we are done.

From *NAE3SAT*, there is no need for the extra u and a much simpler proof results: just make the literals the set elements (thus $2n$ set elements for n variables), make each clause into a 3-set, and a a 2-set $\{x, \overline{x}\}$ for each variable x.

From *Positive NAE3SAT*, we do not even need a transformation: we can restrict *SS* to *Positive NAE3SAT* by allowing only instances where the subsets have three elements each. Splitting a 3-set is exactly equivalent to satisfying a 3-clause in NAE manner.