Computability

DO NOT TURN THIS PAGE UNTIL YOU HAVE BEEN ASKED TO.

Subset sum

We consider a set of elements A. Each of these elements is characterized by a integer, for example its weight or its size. In the following, we will use the word "weight". Possibly, several objects can be associated with the same number (they can have the same weight...). The question is to find a subset of this elements that have exactly a given weight (e.g to fill a truck...). This problem is called **SUBSET SUM**.

SUBSET SUM
INSTANCE : An integer $t \in \mathbb{N}$, and a finite set of elements A, each of which $a \in A$ are associated with a number $s(a)$
QUESTION : Is there a subset $A' \subseteq A$ such that the total weight of these elements is t, i.e. such that $\sum_{a \in A'} s(a) = t$?

This is the formal definition of `SUBSET SUM`. For the sake of simplicity, in the following we will consider that A is the set of numbers, and the question is to find a subset of these numbers which elements have a sum of exactly t. We can write $\sum_{a \in A'} a = t$, but take care : A is a set in which a number may be repeated ! We want to show that this problem is NP-complete. We are building a polynomial transformation from `3-SAT`. An instance of `3-SAT` is composed of k clauses $C_1, \ldots, C_k$ using n variables $x_1, \ldots, x_n$. We assume that there is no clause where a variable and its negation both appear. We also assume that every variable appear in at least one clause.

Question 1 *Give the formal definition of `3-SAT`. Show that the assumptions we have made do not modify the generality of the instances of `3-SAT` we consider.*

Question 2 *Let a instance of `SUBSET SUM` be defined by A={1, 2, 3, 5, 10, 13, 23, 34, 36, 45, 107, 207, 407, 666, 667, 806, 896, 733, 1000, 1008, 1969, 1998, 2000, 16634, 17567, 17563, 19000 } and* $t = 19252$. *Is it a positive instance ? Show that* `SUBSET SUM` *is in* `NP`.

We now proceed to the transformation as follows. The set A is initially empty. For every variable x_i we add two integers v_i and v'_i in A. For every clause C_j, we add two integers s_j and s'_j. Theses integers are encoded with $n+k$ digits, in the usual decimal encoding. We build a table where each of the n first columns (from the left to the right) correspond to (are indexed by) one variable $\{x_1, \ldots, x_n\}$, and each of the k other columns correspond to (are indexed by) one clause $C_1, \ldots, C_k$. The $2n$ first rows will define the value of the integers v_i and v'_i, the $2k$ following rows will define the value of the integers s_j and s'_j. The last row will define the value of the number t. In this table, the values of the digits are determined as follows :

— for every $i \in \{1, \ldots, n\}$, the digits of v_i and v'_i in the column corresponding to x_i is set to 1, and the others $n-1$ most significant digits (those of the columns not indexed by x_i) are set to 0 (these digits correpond to variables $x_j, j \neq i$). Concerning the k least significant digits, the digit of v_i correponding to a clause C_j is set to one if x_i appears in the clause C_j. If $\neg x_i$ appears in the clause C_j, then the digit of v'_i correponsding to the clause C_j is set to one. Note that a variable may appear in different

clauses, and several digit may have a value 1 (at most, k). All the others k least significant digits are set to 0.
- for all $j \in \{1, \ldots, k\}$, the digits of s_i and s'_i of the n most significant digits are set to 0. The digit of s_j in the column corresponding to C_j is set to 1, and the digit of s'_j in the column corresponding to C_j is set to 2. All the others $k-1$ least significant digits of these numbers are set to 0.

Finally, t is constructed as follows : the n most significant digits are set to 1, and the k least significant digits are set to 4.

Question 3 *Give the values of the elements of A and the value of t when the initial instance of* ***`3-SAT`*** *is*

$$\phi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_1 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$

You will define these numbers by writing the table build as above. Show that the construction of this instance of ***`SUBSET SUM`*** *is polynomial. Show that the sum of the digits in one column can not exceed 6. Thus there can not be any carry from one column to the others, and the column can be considered separately.*

Question 4 *Show that if ϕ is satisfiable, then we can define A' by selecting the following numbers :*
- *v_i (respectively v'_i) if x_i has a value* ***`TRUE`*** *in the truth assignment (respectively if $\neg x_i$ has a value* ***`TRUE`*** *in the truth assignment) ;*
- *at least one of the two numbers (s_j, s'_j) for al $j \in \{1, \ldots, k\}$.*

Question 5 *Show that if the instance of* ***`SUBSET SUM`*** *is positive, (thus there exists $A' \subseteq A$ such that the elements of A' have a sum of t), then Φ is satisfiable (indication : consider the numbers $v_i \in A'$).*

Question 6 *Conclude about the NP-completeness of* ***`SUBSET SUM`****?*

Partition

We have proved that `SUBSET SUM` is NP-complete. We can build a polynomial transformation from `SUBSET SUM` to `PARTITION` as follows. Let $S = \sum_{a \in A} a$. We define a instance of `PARTITION` by adding to numbers to the set A. These numbers are : $t_1 = 2S - t$ and $t_2 = S + t$. Call $W = A \cup \{t_1, t_2\}$

Question 7 *Define formally the problem of* ***`PARTITION`****. Show that partition is NP-complete (indication : notice that if we can partition W into two subsets W_1 and W_2, then t_1 and t_2 can not be in the same subset).*

	x_1	x_2	x_3	C_1	C_2	C_3	C_4
v_1							
v_1'							
v_2							
v_2'							
v_3							
v_3'							
s_1							
s_1'							
s_2							
s_2'							
s_3							
s_3'							
s_4							
s_4'							
t							