Proof of NP-Completeness

of the

Subset Sum Language

Guillaume Denoncourt 002251327

Maxime Foltz 002268095

Nicholas Lafleur 002242625

Submitted to Stefan Bruda for CS467/567, February 8, 2020

Introduction:

The decision problem of subset sum is as follows:

Given a set of integers, say S, and a target number t (in our case t is hard set to 0), does there exist $S' \subseteq S$ such that:

$$\sum_{x_i \in S'}^{x} i = 0$$

[1]

Equivalent writing of the problem:

Given a set S of integers, is there a subset $S' \subseteq S$ such that $\sum_{i \in S'} i = t$, t being an integer.

Section I: Proving Membership in NP

In order to prove that the decision problem of Subset Sum is in the class of NP, we need to prove that the problem is verifiable within polynomial time. Here is the pseudo code of an algorithm that can verify a given certificate S' of the language:

1. Int sumSoFar = 0
2. Iterate through each element of S'
3. sumSoFar = sumSoFar + $x_i \in S'$
4. return sumSoFar

Proof: let n = $|S|$, since $S' \subseteq S$, S' contains at most n elements. Therefore, 2. will execute at most n times. Furthermore, since 3. takes constant time, we get O(n) time complexity, which is a polynomial of degree 1.

Hence, Subset Sub $\in$ NP.

Section II: Proving membership in NPC using a polynomial reduction from 3-SAT

We show that 3-SAT $\leq_p$ Subset Sum:

Reduction: we have a formula ϕ with n variables xi and m clauses cj (containing 3 variables)

We will have 2 numbers in set S for each variable and for each clause, the numbers have n+m digits, and we have a number representing the integer t that also has n+m digits.

The first n digits represent the variables and then the next m digits represent the clauses [2].

- vi and v'i are the numbers for the xi variable, among the first n digits the i-th digit is 1 and the others are 0.
  Then the (n+j)-th digit is equal to 1, if xi is in cj for vi and, if $\bar{x}_i$ is in cj for v'i. Other digits are 0.
- sj and s'j are the numbers for the cj clause, the first n digits are all 0.
  Then the (n+j)-th digit is equal to 1 for sj and 2 for s'j. Other digits are 0.
- The first n digits of t are 1 and then it is 4.

The maximum sum of digits from a same column is 6. Therefore, by using a base 10, we won't have carries when adding the numbers.

We can do this reduction in polynomial time:

- The set S contains 2n+2m numbers, each one of these numbers have n+m digits.
- The time to produce a digit of a number in S is polynomial in (n+m).
- The integer t has n+m digits.
- The time to produce a digit of t is constant.

Now we will show that: ϕ is satisfiable ⇔ there exists a subset $S' \subseteq S$ whose sum is t.

⇒ : we suppose that ϕ is satisfiable

For i=1 to n, if xi is true then vi is in S', else if xi is false then v'i is in S'.

Since ϕ is satisfiable, each clause contains at least one true and at most 3 true. So if we add the m last columns of vi and v'i numbers in S', we will have either 1, 2 or 3 [2].

And using these sums we will pick the sj and s'j to include in S'. If for the (n+j)-th column we have 1 we will add sj and s'j in S' to get 4 (1+1+2). If we have 2, we will add s'j in S' to get 4 (2+2). If we have 3, we will add sj in S' to get 4 (3+1).

For the sum of the digits of the n first columns of numbers in S', we will have 1 since we have either vi or v'i in S' and for the sj and s'j there are only 0.

Then the values in S' sum to t.

⇐ : we suppose that there exists a subset $S' \subseteq S$ whose sum is t

The subset S' must include either vi or v'i for each i= 1 to n. If vi is in S' we take xi as True, else if v'i is in S' then we take xi as false.

Let's prove that every clause cj is satisfied.

To have a sum of 4 in the (n+j)-th column representing clause cj, we need at least one vi or v'i value that contains a 1 in this column, since the sj and s'j sum to at most 3.

If S' includes a vi that has a 1 in the (n+j)-th column then there is xi in cj. Since vi is in S', xi is True so cj is satisfied.

If S' includes a v'i that has a 1 in the (n+j)-th column then there is $\bar{x}_i$ in cj. Since v'i is in S', xi is False so cj is satisfied.

Therefore, all clauses are satisfied and ϕ is satisfiable [2],[3].

2.2 Example:

Let us reduce down the following instance of a 3-SAT problem to a Subset Sum Problem:

Let ϕ = $(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)$

Create a table with the following rules: let vi be true if and only if x_i is true, otherwise let v'i be true.

Table 1. [3]

		x_1	x_2	x_3	C_1	C_2	C_3	C_4
V1	=	1	0	0	1	0	0	1
V'1	=	1	0	0	0	1	1	0
V2	=	0	1	0	1	1	0	0
V'2	=	0	1	0	0	0	1	1
V3	=	0	0	1	0	1	1	1
V'3	=	0	0	1	1	0	0	0
S1	=	0	0	0	1	0	0	0
S'1	=	0	0	0	2	0	0	0
S2	=	0	0	0	0	1	0	0
S'2	=	0	0	0	0	2	0	0
S3	=	0	0	0	0	0	1	0
S'3	=	0	0	0	0	0	2	0
S4	=	0	0	0	0	0	0	1
S'4	=	0	0	0	0	0	0	2
t	=	1	1	1	4	4	4	4

[
Taking each row as a decimal number, we get:
S = {1001001,1000110,101100,100011,10111,11000,1000,2000,100,200,10,20,1,2}, with our target t = 1114444. We will use any of S_i or S'_i in order for the columns to sum up to the arbitrary value of 4 for this example. Now, given a certificate of the 3-SAT problem, say $\{x_1 = True, x_2 = False, x_3 = True\}$, 3-SAT is true *if and only if* there exists $S' \subseteq S$ such that

$$\sum_{x_i \in S'}^{x} {}_i = t$$

In this instance, φ is satisfied, and there is indeed a subset S'(given by the highlighted rows in Table 1) = {1001001,0100011,10111,1000,2000,100,200,20,1} whose sum adds up to t [3].

Section III: Optimization version of the problem

Optimization problem of subset sum:

We have a set S of positive integers and a positive integer b.

We want to find an optimal solution sol such that $sol \leq b$ and that $\sum_{i \in S'} i = sol$ with $S' \subseteq S$ [4].

Example:

$S = \{2; 5; 10; 4\}, b = 18$

The optimal solution would be: $17\ with\ S' = \{2; 5; 10\}$ where we have $2 + 5 + 10 = 17$.

References:

1. P. Crescenzi & V. Kann, The NP-completeness of Subset Sum[Course Presentation], https://www.cs.mcgill.ca/~lyepre/pdf/assignment2-solutions/subsetSumNPCompleteness.pdf
2. M. Sipser, Introduction to the Theory of Computation, Second Edition, http://fuuu.be/polytech/INFOF408/Introduction-To-The-Theory-Of-Computation-Michael-Sipser.pdf
3. B. Arad, The subset-sum problem,https://www.cs.bgu.ac.il/~algo132/wiki.files/3sat-susu.pdf
4. C. Parry, Reduction of Subset-Sum-Optimization to Subset-Sum-Decision, https://courses.cs.washington.edu/courses/csep521/05sp/lectures/sss.pdf