Comp 36111 Coursework B

16th November, 2019

Ian Pratt-Hartmann

Q1 a) The following non-deterministic procedure recognizes 2-NO-MAJORITY, and obviously runs in polynomial time.

```
begin 2noMaj (S)
    set  a := 0
    set  b := 0
    for each s in S do
        either  set  a := a + s   } non-deterministic choice
        or      set  b := b + s   }
    end for each
    if  a = b
        output Y
    end if
    output N
```

b) If S_1^* contains both $(2\Sigma S - b)$ and $(\Sigma S + b)$, then $\Sigma S_1^* \geq 3\Sigma S$ and $\Sigma S_2^* < \Sigma S$. Since S must be nonempty, we have $\Sigma S_1^* > \Sigma S_2^*$. If S_1^* contains neither number, a symmetric argument shows $\Sigma S_1^* < \Sigma S_2^*$

c) $\Sigma S_1^* = \Sigma S' + (2\Sigma S - b)$

$\Sigma S_2^* = \Sigma(S \setminus S') + (\Sigma S + b)$

$= 2\Sigma S - \Sigma S' + b$

Hence, if $\Sigma S_1^* = \Sigma S_2^*$, then

$\Sigma S' + (2\Sigma S - b) = 2\Sigma S - \Sigma S' + b$

$\therefore 2\Sigma S' = 2b$

$\therefore \Sigma S' = b$

d) (Q1 contd)

Define $S_1^* = S'' \cup \{ 2\Sigma S - b \}$

$S_2^* = S \setminus S'' \cup \{ \Sigma S + b \}$

Thus, S_1^* and S_2^* partition S^*. Since $\Sigma S'' = b$, we have

$\Sigma S_1^* = \Sigma S'' + 2\Sigma S - b = 2\Sigma S$

$\Sigma S_2^* = \Sigma S - \Sigma S'' + (\Sigma S + b) = 2\Sigma S$

Hence $\Sigma S_1^* = \Sigma S_2^*$ as required.

e) If S^* is a positive instance of 2-NO-MAJORITY, then, by part b), S^* can be partitioned into S_1^* and S_2^* with $\Sigma S_1^* = \Sigma S_2^*$ and S_1^* containing $2\Sigma S - b$, but not $\Sigma S + b$. Now define S' as in part c) so that $\Sigma S' = b$, whence (S, b) is a positive instance of SUBSETSUM.

If (S, b) is a positive instance of SUBSETSUM, then S^* is a positive instance of 2-NO-MAJORITY* by d).

It is obvious that this reduction can be effected in logarithmic space, since we need only store carries in computing sums and differences.

Hence, $(S, b) \mapsto S^*$ is a reduction. Since SUBSETSUM is known to be NP-hard, and SUBSETSUM $\leq_m^{\log}$ 2-NO-MAJORITY, 2-NO-MAJORITY is NP-hard.

Q2. a) Let $S = \{s_1, \dots, s_p\}$ Note that, by assumption $s_1 \leq s_2 + \dots + s_p$. So let i be the largest number $(1 \leq i < p)$ s.t.

$$\sum_{i \leq j \leq i} s_j \leq \sum_{i < j \leq p} s_j \qquad (1)$$

Now let $S_1 = \{s_1, \dots, s_i\}$

$S_2 = \{s_{i+1}\}$

$S_3 = \{s_{i+2}, \dots, s_p\}$

By construction, $\sum S_1 \leq \sum S_2 + \sum S_3$, since this is just the statement (1). By assumption, $\sum S_2 \leq \sum S_1 + \sum S_3$, since this is just the statement $s_{i+1} \leq (s_1 + \dots + s_i) + (s_{i+2} + \dots + s_p)$. We need only show $\sum S_3 \leq \sum S_1 + \sum S_2$.

Suppose $i = p-1$. Then by assumption $\sum S_2 = s_p \leq s_1 + \dots + s_{p-1}$ and $\sum S_3 = 0$ and there is nothing to show. On the other hand, suppose $i < p-1$. Then, by maximality of i in (1), we have

$$s_1 + \dots + s_{i+1} > s_{i+2} + \dots + s_p .$$

But this is just the statement $\sum S_3 < \sum S_1 + \sum S_2$, so we certainly have $\sum S_3 \leq \sum S_1 + \sum S_2$ as required.

Q2 contd b)

Suppose the multiset S is written in blocks on the input tape, with the integers (in binary) separated by #. Write $S = \{s_1, \ldots, s_p\}$

The following subroutine computes the jth bit and carry of the sum of all the $s_{i'}$ except for s_i:

bit+Carry(j, i)

let $c' := \begin{cases} 0 & \text{if } j = 1 \\ c'' & \text{where bit+Carry}(j-1, i) = (b'', c'') \end{cases}$

for each $i' = 1$ to m

let b' be the jth bit of $s_{i'}$

if $i \neq i'$ then

let $c' = c' + b'$

end if

end for

let b be the least significant bit of c' $(= c' \% 2)$

let $c = \lfloor c'/2 \rfloor$ (i.e. shift c' right)

return (b, c).

This clearly takes only logarithmic space, since the number of bits required for c is $\leq \lceil \log p \rceil$

Note that the ~~recursive~~ call space used by the recursive call can be recovered; we need only store the index j and the carry c''.

26 contd

Now, to test if S is a positive instance of n-NO-majority, just look for a number in S greater than the sum of all the others. If one is found, reject; otherwise, accept

nNmaj (S)

Let $S = s_1 \cdots s_p$. Let b be one greater than the maximum number of bits in any s_i plus p.

for $i = 1$ to p

for $j = b$ to 1 (i.e. j decreases)

let b be jth bit of s_i

let $(b', c') =$ bit + Carry (i, j)

IF $b > b'$ answer N ($s_i >$ sum of others)

IF $b' < b$ set $j = 0$ ($s_i <$ sum of others, so break out of inner loop)

end for j

end for i

answer Y (No s_i is greater than sum of all others)