2

In order to prove that Maj-SAT (MSAT) is in NP-Complete, we must show that MSAT

(i.) MSAT $\in$ NP
AND
(ii.) MSAT $\in$ NP-Hard

We will prove (i) directly before reducing a know NP-Complete problem polynomially to MSAT. Specifically, we will define the reduction of satisfiability (SAT)

$$SAT \leq_p MSAT$$

lemma 2.1: MSAT $\in$ NP

Proof:

In order for MSAT to be in NP, it must have an efficient certifier. One such certifier, we will call it B, will take as input a list of clauses $C_1, \ldots, C_k$, a list of assignments $a_1, \ldots, a_n$. B will then assign value a_i to each x_j in each C_h where $i, j \in [1, n]$, $h \in [1, k]$. While doing so, B will count the true values of each C_h and return true ("yes") iff the strict majority of C_h contain a strict majority true values.

Lemma 2.2: $3SAT \leq_p MSAT$

SAT: $C = C_1 \wedge C_2 \wedge \dots \wedge C_k$, $X = \{x_1, \dots, x_n\}$, $L = \{l_1, \dots, l_k\}$

Proof: 3SAT is a special instanse of SAT in which each clause C_i of l_i disjunctions s.t. $1 \leq l_i \leq 3$. Recall that the clauses are comprised of disjunctive terms $X = \{x_1, \dots, x_n\}$.

We will reduce 3SAT to MSAT in polynomial time. In fact, our reduction will be of $O\left(\sum_{i=1}^{k} l_i(l_i-1) + k - 1\right)$, which is polynomial in the number of clauses and their lengths. Because the l_i are ≤ 3 by assumption, our reduction is linear in k*. Specifically, our reduction will comprise of 2 steps, after initializing $C' = \{C'_1, \dots, C'_k\}$ as a list/set of clauses, and each C'_i for $1 \leq i \leq k$ to be a list/set of ≤ 3 boolean terms in X. Of Course, each $X'_j \in C'_i$ corresponds to $x_j \in C_i$. Think of it as replacing each logical operator $\{\wedge, \vee\}$ with a ",". The 2 steps: (L remains the same)

1.) for each C'_i, append $(l_i - 1)$ true values (1) to C'_i

2.) append $k-1$ false values (0) to C'

Feeding this into MSAT will produce "yes" iff original C produces "yes" from 3SAT. This is true because the only way a clause will be true is if one of its boolean terms (position or negation, as defined in C_i + reflected in C'_i) evaluates true, thus giving that clause truth majority. Similarly, C' will only evaluate to true

---

* n is irrelevant because we don't mess w/ assignments in our reduction.

iff precisely k of its C_i's evaluate to true, else the 0's will have majority and it will evaluate false.

To make this concrete:

3SAT

$C = C_1 \wedge \cdots \wedge C_k$

for $i \in [1,k]$, $1 \leq l_i \leq 3$ of $x_j \in X$ in C_i
$j \in [1,n]$

$\leq_p$

MSAT

$C' = \{C'_1, \ldots, C'_k, \overbrace{0, \ldots, 0}^{k-1}\}$

for each C_i, $C_i = \{x_1, \ldots, x_h, \overbrace{1, \ldots, 1}^{l_i - 1}\}$
$i \in [1,k]$
$h \leq l_i$

- ~~note~~: the problem specifies that MSAT takes vars as input. above we give constants. this can easily be remedied by assigning all "1"'s to x_0 and all "0"'s to a clause consisting of only $\bar{x}_0$ for some new x_0 ∎

Thm 8.15: 3-Satisfiability is NP-Complete
(from Kleinberg & Tardos)

Thm 8.14: If $Y \in$ NP-Complete and $X \in$ NP where $Y \leq_p X$, then $X \in$ NP-Complete
(from K & T)

Thm 2.1: By Thm 8.15, Lemma 2.1, Lemma 2.2, and finally Thm 8.14, Maj-SAT is NP-Complete ∎
(from me)

iff precisely k of its C_is evaluate to true, else the 0's will have majority and it will evaluate false.

To make this concrete:

3SAT

$C = C_1 \wedge \cdots \wedge C_k$

for $i \in [1,k]$, $1 \leq l_i \leq 3$ of $x_j \in X$ in C_i, $j \in [1,n]$

$\leq_p$

MSAT

$C' = \{C'_1, \ldots, C'_k, \underbrace{0, \ldots, 0}_{k-1}\}$

for each C_i, $i \in [1,k]$, $C_i = \{x_1, \ldots, x_h, \underbrace{1, \ldots, 1}_{l_i - 1}\}$, $h \leq l_i$

- note: the problem specifies that MSAT takes vars as input. above we give constants. this can easily be remedied by assigning all "1"'s to x_0 and all "0"'s to a clause consisting of only $\bar{x}_0$ for some new x_0 ∎

Thm 8.15: 3-Satisfiability is NP-Complete
(from Kleinberg & Tardos)

Thm 8.14: If $Y \in$ NP-Complete and $X \in$ NP where $Y \leq_p X$, then $X \in$ NP-Complete
(from K & T)

Thm 2.1: By Thm 8.15, Lemma 2.1, Lemma 2.2, and finally Thm 8.14, Maj-SAT is NP-Complete ∎
(from me)