1

Definition. A *partially labeled graph* is a tuple (V, E, L, m), where (V, E) forms a graph (called the *underlying graph*), $L \subseteq V$, and $m : L \to \mathbb{N}$ is a function.

Definition. A *mine assignment* of a partially labeled graph $G = (V, E, L, m)$ is a subset of $V \setminus L$ such that for each $v \in L$, $m(v) = |N(v) \cap M|$.

Definition. Define MineConsistency to be the language

$$\{\langle G\rangle \mid G \text{ is a partially labeled graph and there exists a mine assignment on } G\}$$

where G is represented as lists of vertices and edges, and a list of pairs $\langle v, m(v)\rangle$.

Now first, we need to prove that MineConsistency $\in$ NP. We define the certificate of a mine-labeled graph to be the mine assignment, and clearly only graphs in MineConsistency have certificates. A verifier then need only check that $L \subseteq V \setminus M$, and that each $m(v) = |N(v) \cap M|$, so clearly there is a polynomial-time verifier.

Next, to show that MineConsistency is NP-hard, we show 3-SAT $\leq_p$ MineConsistency. For any CNF ϕ with variables $x_1, \cdots, x_n$ and clauses $c_1, \cdots, c_s$, we define the partially labeled graph $G_\phi = (V, E, L, m)$ with mine labeling $m_\phi : L \to \mathbb{N}$:

$$V = \{v_i, x_i, \neg x_i \mid i \leq n\} \sqcup \{c_j, c_j^1, c_j^2 \mid j \leq s\}$$
$$E = \{\{v_i, x_i\}, \{v_i, \neg x_i\} \mid i \leq n\} \cup \{\{c_j, c_j^1\}, \{c_j, c_j^2\} \mid j \leq s\} \cup \{\{x, c\} \mid x \text{ is a literal apearing in clause } c\}$$
$$L = \{v_i \mid i \leq n\} \cup \{c_j \mid j \leq s\}$$

where $m(v_i) = 1$ and $m_\phi(c_j) = 3$.

The number of vertices is $3n+3s$, the number of edges is $2n+5s$, and the number of labels is $n+s$, so $\langle G_\phi, m_\phi\rangle$ can be computed in polynomial time in the size of ϕ.

Claim. *If there is a satisfying assignment to ϕ, there is a mine assignment on G_ϕ.*

Proof. Consider the set M of true literals in the satisfying assignment. Clearly it has only one of $\{x_i, \neg x_i\}$ for each i, so each v_i neighbors exactly $1 = m(v_i)$ mines.
Moreover, each c_j neighbors at least one true literal and at most three. Thus we can add some amount of c_j^1, c_j^2 to M so that c_j borders exactly $3 = m(c_j)$ mines. □

Claim. *If there is a mine assignment on G_ϕ, there is a satisfying assignment to ϕ.*

Proof. Let M be the mine assignment, and consider $M \cap \{x_i, \neg x_i \mid i \leq n\}$. Since each v_i borders exactly one mine, this contains exactly one of each $\{x_i, \neg x_i\}$, so it is a valid assignment to ϕ. Since each c_j borders exactly 3 mines, and only 2 of its neighbors are not literals, it borders some literal which is a mine, and hence set true in the assignment. This is a satisfying assignment to ϕ. □

Hence $\langle G_\phi\rangle \in$ MineConsistency iff $\phi \in$ 3-SAT. Thus the polynomial-time computable function $\phi \mapsto \langle G_\phi\rangle$ is a reduction, and 3-SAT $\leq_p$ MineConsistency.

2

If the window was 2x2, we might not detect certain invalid configurations due to the head appearing more than once in the same row. For example, if a NTM had the choice of moving its head left or right when in a certain state, on reading a certain symbol then the next configuration must have one of these possibilities, but not both at once! However, if we used a 2x2 window, a row that contained two state symbols corresponding to each possible next move would not appear incorrect, as each window would be legal on its own. Furthermore, once two heads appeared in a single row, they could collaborate, as it were, to make the final row accepting, even though no actual accepting computation existed.

3

Suppose A is PSPACE-hard. Then for any $L \in \mathsf{NP} \subseteq \mathsf{NPSPACE} = \mathsf{PSPACE}$, by PSPACE-hardness $L \leq_p A$. Thus A is NP-hard.

4

We first give a useful fact:

Lemma 1. *Two DFAs A with a states and B with b states differ on some input iff they differ on some input of length at most ab.*

Proof. The reverse direction is trivial. In the forward direction, suppose A and B differ, and let w be a string of minimal length on which they differ. Suppose $|w| > ab$. There are only ab possible combinations of states w can take A and B to, so there must by pigeonhole be some $j > i$ for which both A and B are at the same states after i or j symbols of w. Then deleting symbols $i+1$ through j of w, we have a shorter string which takes A and B to the same states as w, and thus a shorter string on which they differ. This contradicts the minimality of w. □

The following corollary is obvious by considering the powerset DFAs.

Corollary 1. *Two NFAs A with a states and B with b states differ on some input iff they differ on some input of length at most 2^{a+b}.*

Using these, we have a straightforward nondeterministic algorithm to decide if two DFAs differ:

1. If the input does not encode two NFAs A and B, reject. Otherwise let $a = |A|, b = |B|$.
2. Store a list of states $S \subseteq Q_A$ ($O(a)$ space), a list of states $T \subseteq Q_B$ ($O(b)$ space), and a string length counter l ($a+b$ space, since we will bound its value by 2^{a+b}). This uses linear space.
3. Initialize S to be the ϵ-closure of the initial states of A, and T to be the same in B. Set $l = 0$.
4. For $l = 0$ to 2^{a+b}:

(a) If the states of A include a final state but the states of B do not, or vice versa, the symbols chosen so far construct a string on which A and B differ. Accept.

(b) Nondeterministically choose a symbol $\sigma \in \Sigma$.

(c) Update S to be (the ε-closure of) the union $\bigcup_{s \in S} \delta_A(s, \sigma)$. Do the same for T.

This accepts iff there is a string of length at most 2^{a+b} (i.e. if there is any string at all, by the previous corollary) on which the two NFAs differ.

This shows that the complement of EQ_{REX} lies in $\mathsf{NPSPACE} = \mathsf{PSPACE}$. But PSPACE is closed under complement: given a deterministic polynomial-space decider, you can switch the accept and reject states to get a decider for the complement. Thus $EQ_{REX} \in \mathsf{PSPACE}$.