Theorem 1. *3SAT is $\mathcal{NP}$-complete.*

Proof sketch. It's easy to see that 3SAT $\in \mathcal{NP}$. We reduce $3CSP$ to 3SAT: Given a list $\mathcal{L}$ of 3-constraints over formal variables $x_1, ..., x_n$, we construct a 3CNF Φ over the same set of formal variables $x_1, ..., x_n$ such that for every assignment x to the variables, the assignment satisfies $\mathcal{L}$ if and only if it satisfies Φ.

We will map $\mathcal{L}$ to Φ constraint-by-constraint. The main part of the proof is the following (we will use it with $v = 3$):

Fact 2. *Every function $f\colon \{0,1\}^v \to \{0,1\}$ can be computed by an* AND *of at most* 2^v OR*s over the v input variables*

We map each 3-constraint $c \in \mathcal{L}$ to an AND of at most eight ORs of three variables, and the 3CNF is an AND of all these ANDs of ORs (which is itself an AND of ORs). Note that this reduction runs in polynomial time.[1]

As for correctness, for every fixed assignment x and every constraint c, the constraint is satisfied by x if and only if the corresponding AND of at most eight ORs is satisfied by x. Hence, an assignment x satisfies all constraints in $\mathcal{L}$ if and only if it satisfies all the ORs, i.e. satisfies the 3CNF. ■

[1] For example, you can implement it in time $\tilde{O}(N)$ on a RAM machine (where N is the description length of $\mathcal{L}$), hence in polynomial time on a single-tape machine.