Definition 1. A problem $L \subseteq \{0,1\}^*$ is *P-selective* if there exists a polynomial-time computable function $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ such that for each $(x, y) \in \{0,1\}^* \times \{0,1\}^*$ it holds that:

- $f(x, y) \in \{x, y\}$; and
- if $\{x, y\} \cap L \neq \emptyset$ then $f(x, y) \in L$.

Proposition 1 ([1])**.** If SAT is P-selective, then P = NP.

Proof. (Write down assumptions.)
Suppose that SAT is P-selective. Then there exists a polynomial-time computable function f such that for each two propositional formulas φ_1, φ_2 it holds that (i) $f(\varphi_1, \varphi_2) = \varphi_1$ or $f(\varphi_1, \varphi_2) = \varphi_2$ and (ii) if $\varphi_1 \vee \varphi_2$ is satisfiable, then $f(\varphi_1, \varphi_2)$ is satisfiable.

(What to prove?)
We describe a polynomial-time algorithm to solve 3SAT. Take an arbitrary 3CNF formula $\varphi = c_1 \wedge \cdots \wedge c_m$, where $c_i = (\ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3})$ for each $1 \leq i \leq m$. We decide whether φ is satisfiable by doing the following computation.

(Creative part.. which involves describing an algorithm with high-level descriptions.)
We can rewrite φ as the logically equivalent formula $\varphi_{1,1} \vee \varphi_{1,2} \vee \varphi_{1,3}$, where $\varphi_{1,1} = \ell_{1,1} \wedge c_2 \wedge \cdots \wedge c_m$, where $\varphi_{1,2} = \ell_{1,2} \wedge c_2 \wedge \cdots \wedge c_m$, and where $\varphi_{1,3} = \ell_{1,3} \wedge c_2 \wedge \cdots \wedge c_m$.

We use f (at most) twice to select one of $\varphi_{1,1}, \varphi_{1,2}, \varphi_{1,3}$, as follows. We will call the result φ_1. We will also compute a literal ℓ_1. Compute $f(\varphi_{1,1}, \varphi_{1,2} \vee \varphi_{1,3})$. If the result is $\varphi_{1,1}$, then let $\varphi_1 = \varphi_{1,1}$ and let $\ell_1 = \ell_{1,1}$. Otherwise, let $\varphi_1 = f(\varphi_{1,2}, \varphi_{1,3})$ and let ℓ_1 be the corresponding literal in $\{\ell_{1,2}, \ell_{1,3}\}$.

Let us show that φ_1 is satisfiable if and only if φ is satisfiable.

- Suppose that $\varphi \equiv \varphi_{1,1} \vee \varphi_{1,2} \vee \varphi_{1,3}$ is satisfiable. If $\varphi_1 = \varphi_{1,1}$, then $\varphi_{1,1}$ must be satisfiable, because $f(\varphi_{1,1}, \varphi_{1,2} \vee \varphi_{1,3}) = \varphi_{1,1}$. By a similar argument, if $\varphi_1 = \varphi_{1,2}$ or if $\varphi_1 = \varphi_{1,3}$, then φ_1 is satisfiable.
- Conversely, if φ_1 is satisfiable, because $\varphi_1 \models \varphi$, we know that φ is satisfiable too.

Next, we will rewrite φ_1 as the logically equivalent formula $\varphi_{2,1} \vee \varphi_{2,2} \vee \varphi_{2,3}$, where $\varphi_{2,1} = \ell_1 \wedge \ell_{2,1} \wedge c_3 \wedge \cdots \wedge c_m$, where $\varphi_{2,2} = \ell_1 \wedge \ell_{2,2} \wedge c_3 \wedge \cdots \wedge c_m$, and where $\varphi_{2,3} = \ell_1 \wedge \ell_{2,3} \wedge c_3 \wedge \cdots \wedge c_m$.

Again, we use f (at most) twice to select one of $\varphi_{2,1}, \varphi_{2,2}, \varphi_{2,3}$, as follows. We will call the result φ_2, and we compute a corresponding literal ℓ_2. Compute $f(\varphi_{2,1}, \varphi_{2,2} \vee \varphi_{2,3})$. If the result is $\varphi_{2,1}$, then let $\varphi_2 = \varphi_{2,1}$ and let $\ell_2 = \ell_{2,1}$. Otherwise, let $\varphi_2 = f(\varphi_{2,2}, \varphi_{2,3})$ and let ℓ_2 be the corresponding literal in $\{\ell_{2,2}, \ell_{2,3}\}$.

(No need to repeat (essentially) repeating arguments.)
By a similar argument as before, we get that φ_2 is satisfiable if and only if φ_1 is satisfiable—and thus if and only if φ is satisfiable.

We repeat this procedure for all subsequent clauses $c_3, \ldots, c_m$, each time defining the formula φ_i based on φ_{i-1}, using f to select one of $\varphi_{i,1}, \varphi_{i,2}, \varphi_{i,3}$ and letting ℓ_i be the corresponding literal among $\ell_{i,1}, \ell_{i,2}, \ell_{i,3}$.

The result is a formula $\varphi_m = \ell_1 \wedge \cdots \wedge \ell_m$ that is satisfiable if and only if φ is satisfiable.

(Simple claims need no extensive proof.)
We can check in polynomial time whether φ_m is satisfiable by checking if any pair of literals among $\ell_1, \ldots, \ell_m$ are each other's complement. If so, then φ_m is not satisfiable, and otherwise φ_m is satisfiable.

(Correctness.)
The algorithm returns "yes," if φ_m is satisfiable and "no," otherwise. Since φ_m is satisfiable if and only if φ is satisfiable, this means that the algorithm correctly decides 3SAT.

(Argue that it runs in polynomial time.)
This procedure uses m phases, one for constructing each formula φ_i. Each phase takes polynomial time—e.g., we call the function f at most twice. Moreover, after these phases, we check satisfiability of φ_m (and thus satisfiability of φ) in polynomial-time. Overall, this results in a polynomial running time of the algorithm deciding 3SAT.

(Concluding argument.)
Thus 3SAT $\in$ P, and since 3SAT is NP-complete, we get that P = NP. □

References

[1] Alan L. Selman. P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP. *Mathematical Systems Theory*, 13:55–65, 1979.