Proofs can be Arbitrarily Long

We measure the length of a mathematical proposition, or a proof, to be its number of symbols, when written in a formal language $L$ in such a way that a computer could verify correctness of the proof. We let $\Sigma$ be the alphabet of $L$.

**Theorem 1** If $F$ is a recursive function from integers to integers, then there is some mathematical statement $S$ which has a proof, such that any proof of $S$ has length greater than $F(|S|)$.

**Proof:** Let us suppose that Theorem 1 is false. Then there is a recursive function $F$ such that, for any $n$ and for any provable proposition $S$ of length $n$, $S$ has a proof of length at most $F(n)$.

Let $V$ be a proof verification machine. If $S$ is any statement and $P$ is any string, $V$ decides whether $P$ is a proof of $S$.

We now show that the halting problem is decidable. Pick a string $x$. Then “$x \in H$” is a mathematical statement, and can be expressed formally as a string $S \in \Sigma^*$. Since $H$ is recursively enumerable, every member of $H$ can be proved to be a member of $H$, that is, if $S$ is true it has a proof. Let $n = |S|$, the length of the statement $S$. By our hypothesis, either $x \notin H$, or there is some string $y$ which is a proof of $S$ and which has length at most $F(n)$.

Calculate $F(n)$. Let $y_1, y_2, \ldots, y_N$, (for some large $N$) be the list of all strings of length at most $F(n)$ over $\Sigma$. For each $y_i$, run $V$ with input $(S, y_i)$. By our hypothesis, $x \notin H$ if and only if there is some proof of $S$ of length at most $F(n)$, that is, if $V$ accepts the pair $(S, y_i)$ for some $i \leq N$. Since there are only finitely many $y_i$, this is a finite task; if we fail to find a proof, then $x \notin H$.

Thus, we can decide whether $x \in H$, contradicting the known fact that $H$ is undecidable. \qed