(Gödel/Turing incompleteness) Prove that we cannot consistently and completely define a total, deterministic interpreter. An interpreter is a predicate $\mathcal{I}$ that applies to texts; when applied to a text representing a binary expression, its result is equal to the represented expression. For example,

$$\mathcal{I} \forall s: [*\text{char}]: #s \geq 0 = \forall s: [*\text{char}]: #s \geq 0$$

§ Let $Q = \neg \mathcal{I}Q$. Now

$$\mathcal{I}Q$$  

replace $Q$ with its equal

$$\mathcal{I}\neg \mathcal{I}Q$$  

If $\mathcal{I}$ is a complete interpreter as described in the question, then

$$\neg \mathcal{I}Q$$

If $\mathcal{I}$ is a complete incomplete, we have an inconsistency. To save ourselves we can leave the interpreter incomplete. In particular,

$$\mathcal{I}\neg \mathcal{I}Q = \neg \mathcal{I}Q$$

must not be a theorem. If it is an antitheorem, then $\mathcal{I}$ is not an interpreter. So leave it unclassified. Alternatively, we could let $\mathcal{I}$ be partial so that $\mathcal{I}Q = \text{null}$, or nondeterministic so that $\mathcal{I}Q = \text{bin}$. Then $\mathcal{I}Q = \neg \mathcal{I}Q$ is a theorem, but we cannot use the Completion Rule to prove it is an antitheorem because $\mathcal{I}Q$ is not elementary. So we do not have an inconsistency, but we also do not have a total, deterministic interpreter. As any programmer can see, applying $\mathcal{I}$ to $\neg \mathcal{I}Q$ will cause an infinite execution, and produce no answer.

Although the question does not ask for this, here is how you define an interpreter. Start with

$$\mathcal{I}\top = \top$$
$$\mathcal{I}\bot = \bot$$

Now, for texts that represent negations, we want to say something like

$$\mathcal{I}(\neg; s) = \neg \mathcal{I} s$$

It says: to apply $\mathcal{I}$ to a text that starts with $\neg$, just apply $\mathcal{I}$ to the text after the $\neg$, and then negate the result. For texts that represent conjunctions, we want to say something like

$$\mathcal{I}(s; \land; t) = \mathcal{I}s \land \mathcal{I}t$$

And so on for all operators of the theory we are interpreting. The trouble is precedence. For example, the expression

$$\neg \top \land \bot$$

starts with $\neg$, but it's not negating $\top \land \bot$. One solution is to insist that all expressions be fully parenthesized. Another solution is to use Polish prefix notation (see Subsection 3.2.2 on page 31.)