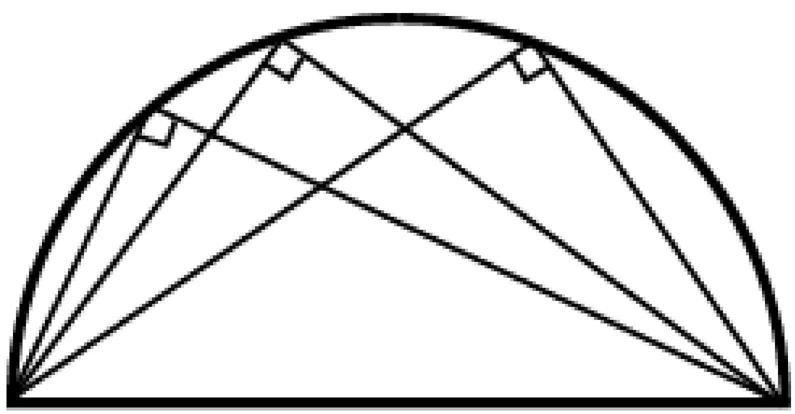
Great Theoretical Ideas In Computer Science		
Steven Rudich		CS 15-251 Spring 2005
Lecture 28	April 26, 2005	Carnegie Mellon University

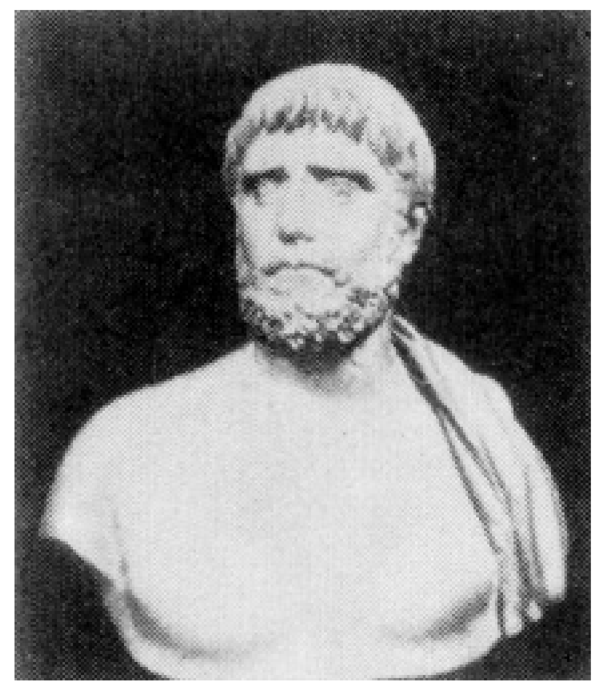
Gödel's Legacy: The Limits Of Logics

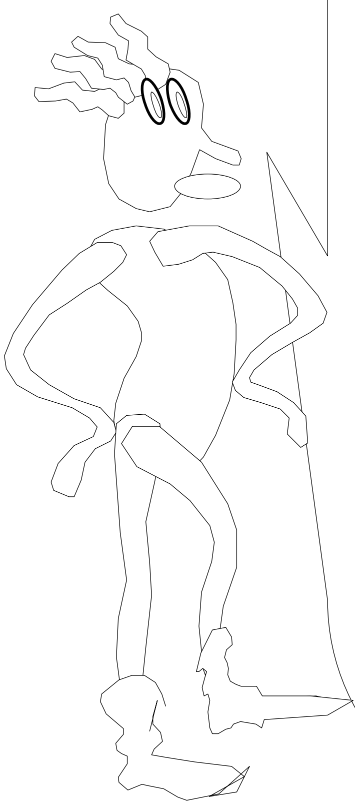
Thales Of Miletus (600 BC)
Insisted on Proofs!

"first mathematician"

Most of the starting theorems of geometry. SSS, SAS, ASA, angle sum equals 180, . . .







GENERAL PICTURE:

A **decidable** set of
“SYNRACTICALLY VALID
STATEMENTS S.

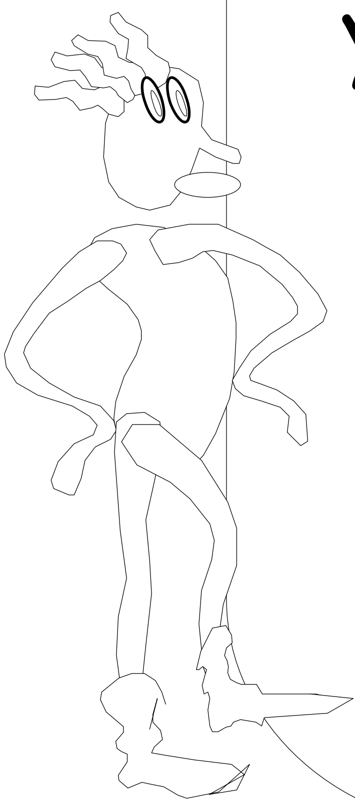
A **possibly incomputable**
subset of S called TRUTH_S
I.e. TRUE STATEMENTS
OF S

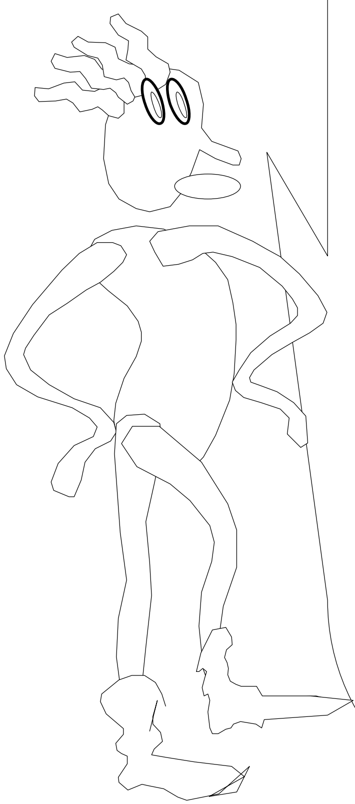
GENERAL PICTURE:

A computable LOGIC_S function
$\text{Logic}_S(x,y)$ =
y does/doesn't follow from x.

$\text{PROVABLE}_{S,L}$ =

All $Q \in S$ for which there is a valid proof of Q in logic L

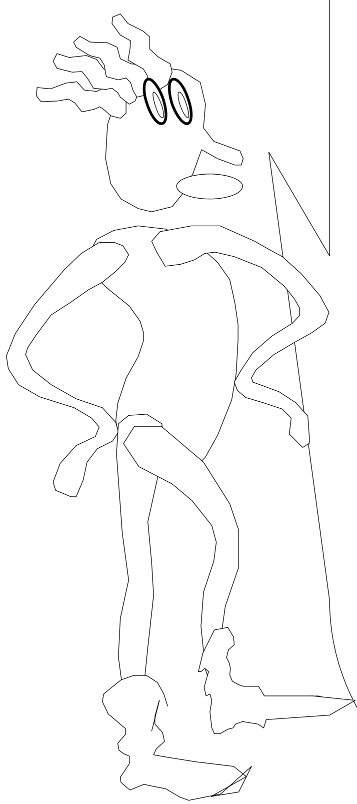




PROOF IN $LOGIC_S$.

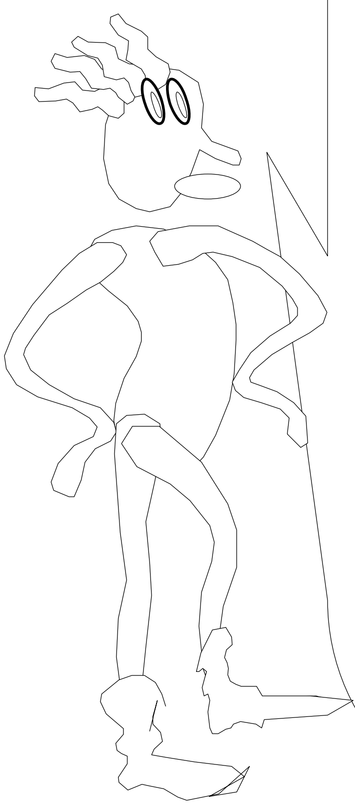
If LOGIC_S(x,y) = y
follows x in one step.

Then we say that statement x implies statement y.



We add a “start statement” Δ to the input set of our LOGIC function.

$\text{Logic}_S(\Delta, S) = \text{Yes}$
will mean that our logic views S as an axiom.

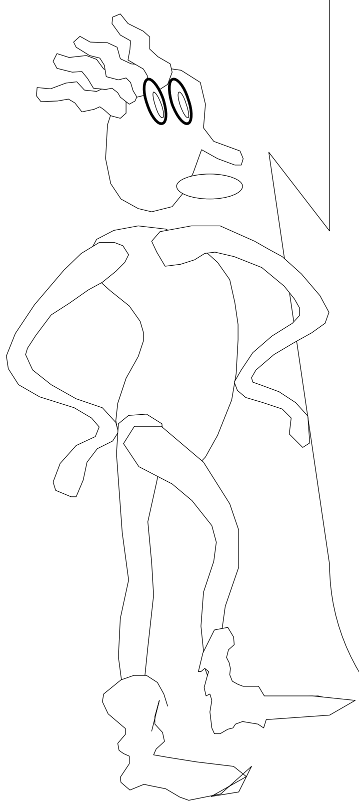


A sequence of statements $s_1, s_2, \ldots, s_n$ is a VALID PROOF of statement Q in LOGIC_S iff

$\text{LOGIC}(\Delta, s_1) = \text{True}$

And for n+1> i>1
$\text{LOGIC}(s_{i-1}, s_i) = \text{True}$

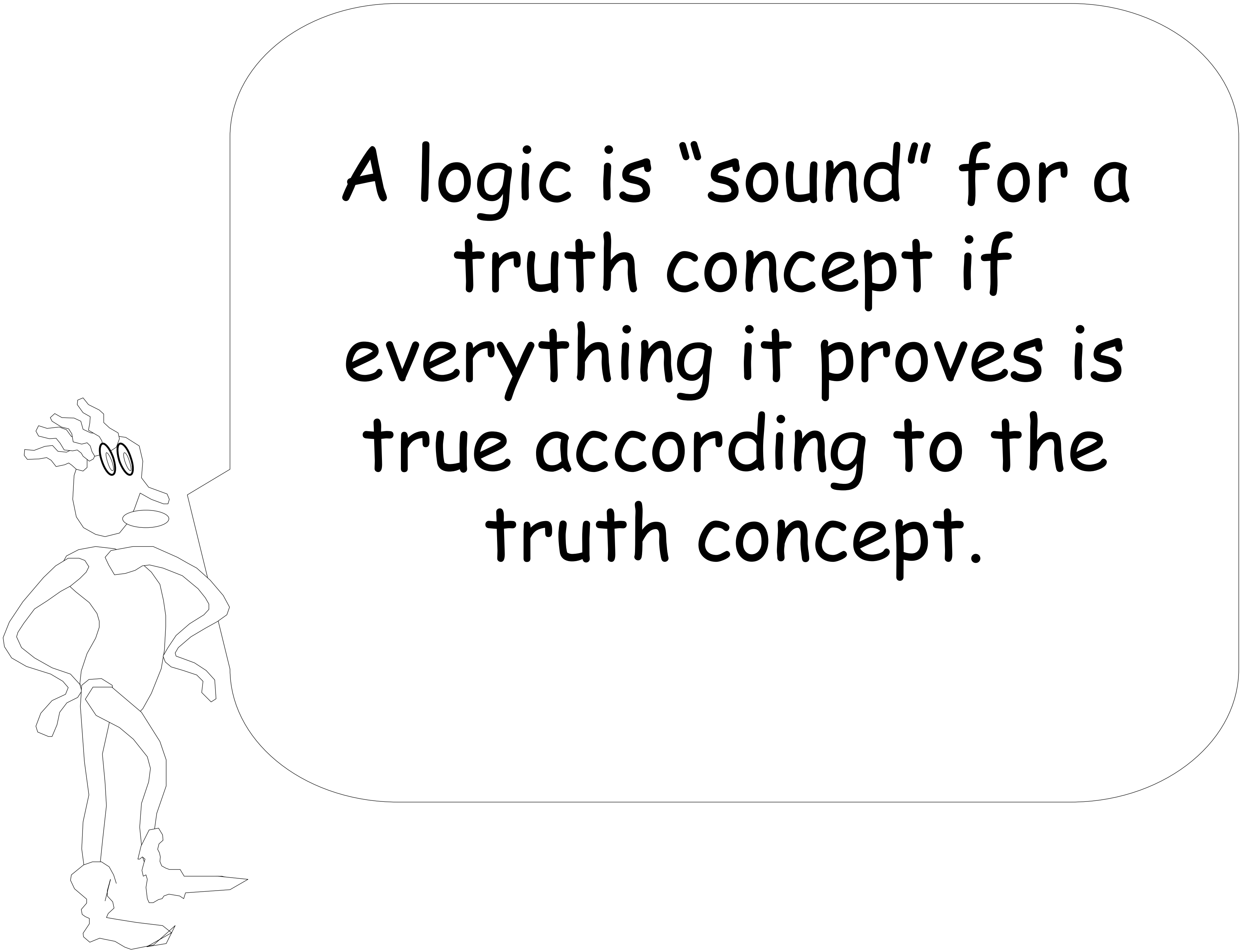
s_n = Q

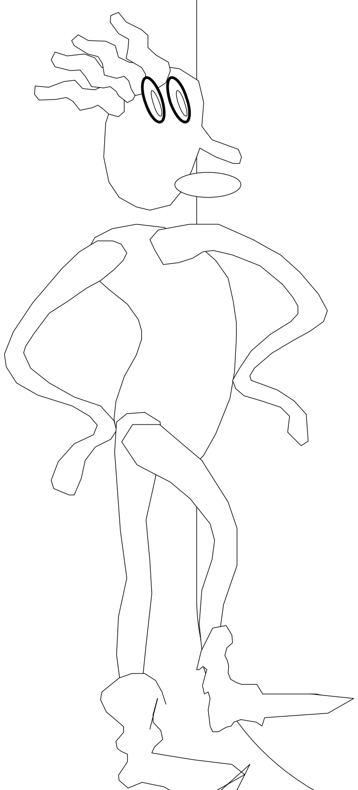


Let S be a set of statements. Let L be a logic function.

$\text{PROVABLE}_{S,L}$ =

All $Q \in S$ for which there is a valid proof of Q in logic L


A logic is "sound" for a truth concept if everything it proves is true according to the truth concept.



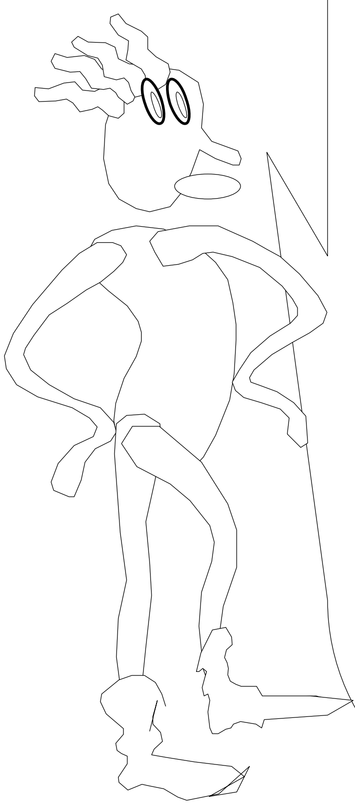
LOGIC_S is SOUND for TRUTH_S if

$$\text{LOGIC}(\Delta, A) = \text{true}$$

$$\Rightarrow A \in \text{TRUTH}_S$$

$$\text{LOGIC}(B,C)=\text{true}$$
$$\text{and } B \in \text{TRUTH}_S$$

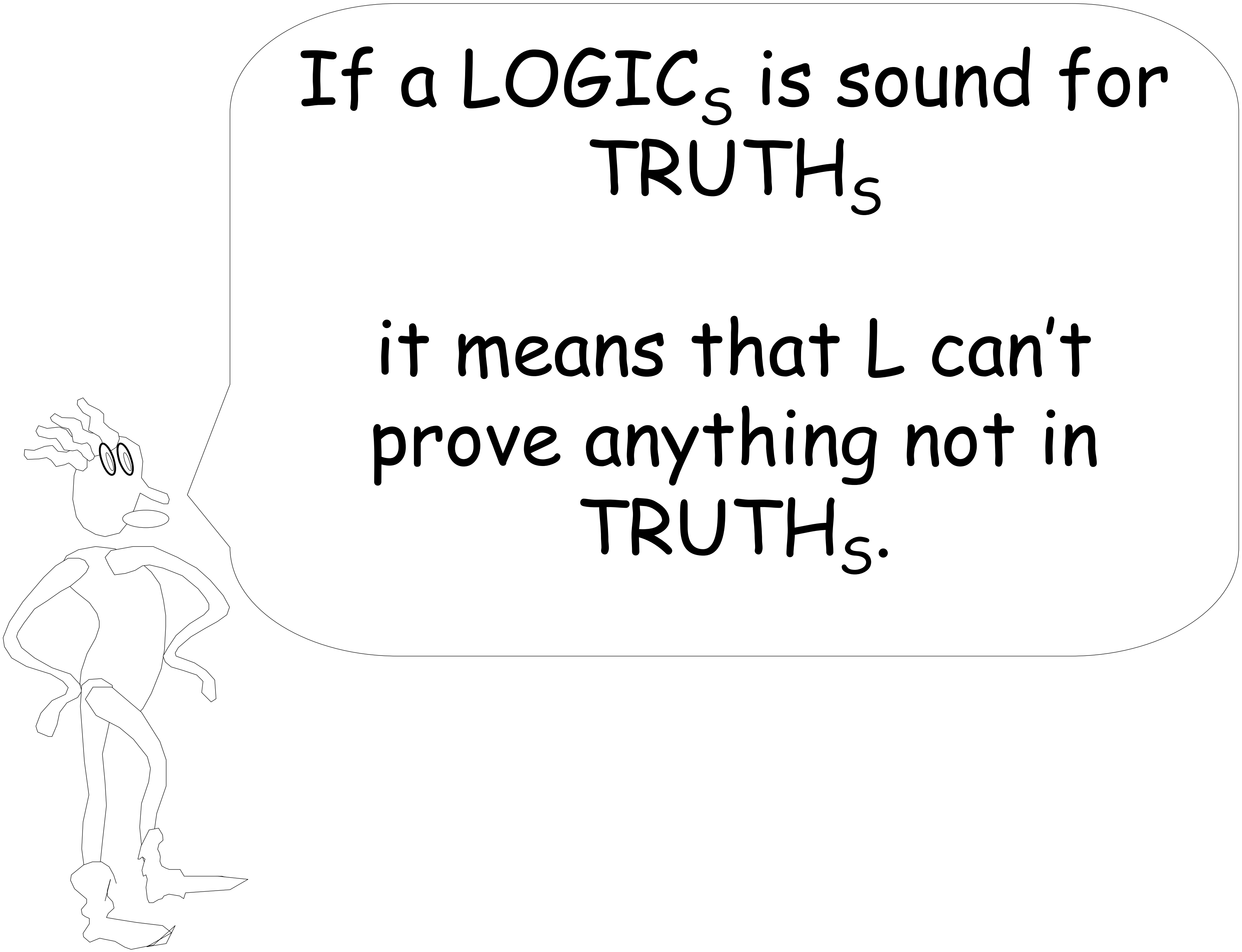
$$\Rightarrow \text{TRUTH}(C)=\text{True}$$

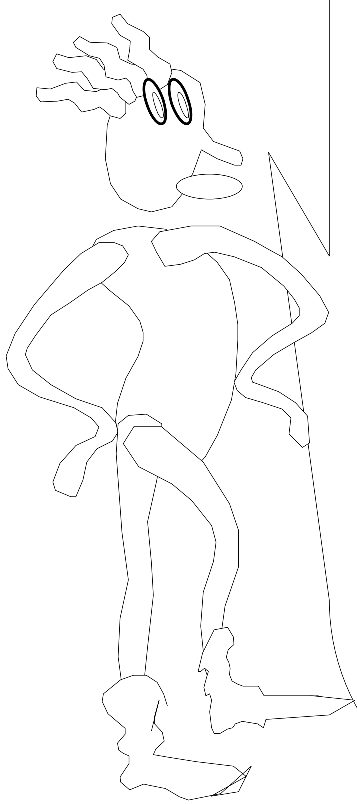


If $LOGIC_S$ is sound for $TRUTH_S$

Then

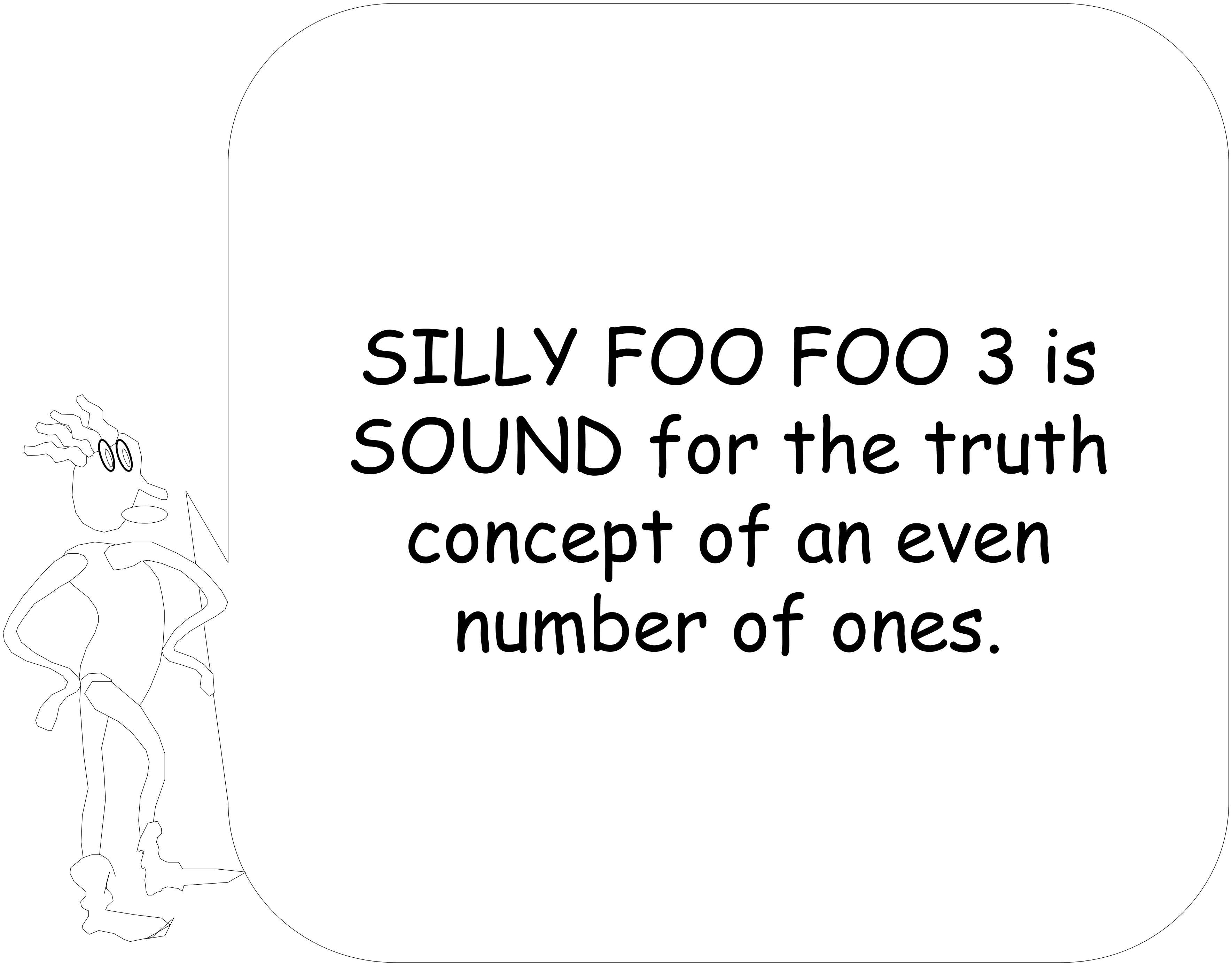
$LOGIC_S$ proves C
$\Rightarrow C \in TRUTH_S$

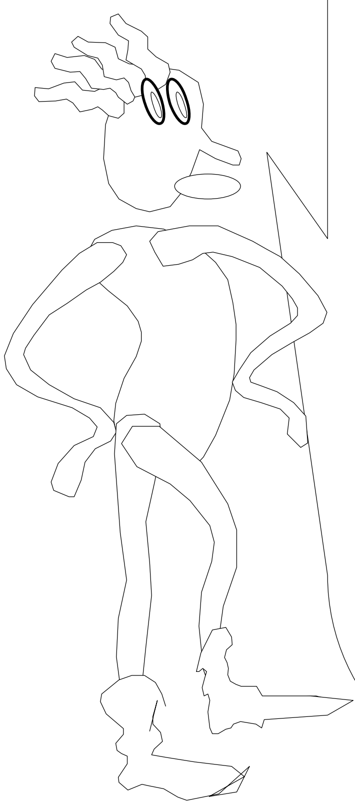

If a LOGIC_S is sound for TRUTH_S
it means that L can't prove anything not in TRUTH_S.



Boolean algebra is SOUND for the truth concept of propositional tautology.

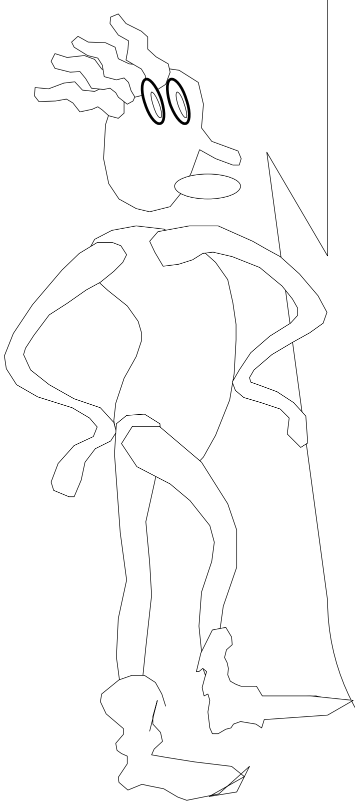
High school algebra is SOUND for the truth concept of algebraic equivalence.


SILLY FOO FOO 3 is SOUND for the truth concept of an even number of ones.



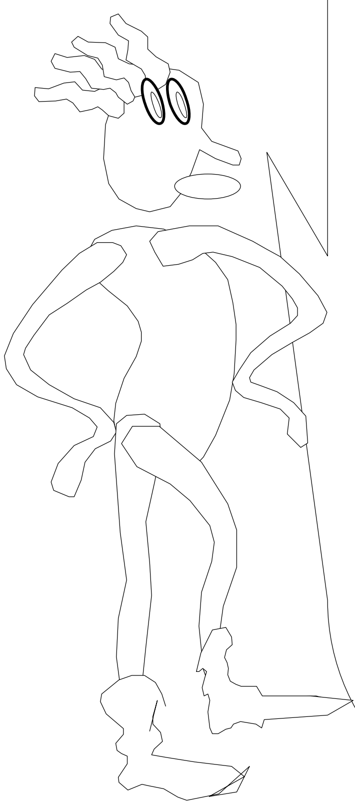
Euclidean Geometry is SOUND for the truth concept of facts about points and lines in the Euclidean plane.

Peano Arithmetic is SOUND for the truth concept of (first order) number facts about Natural numbers.



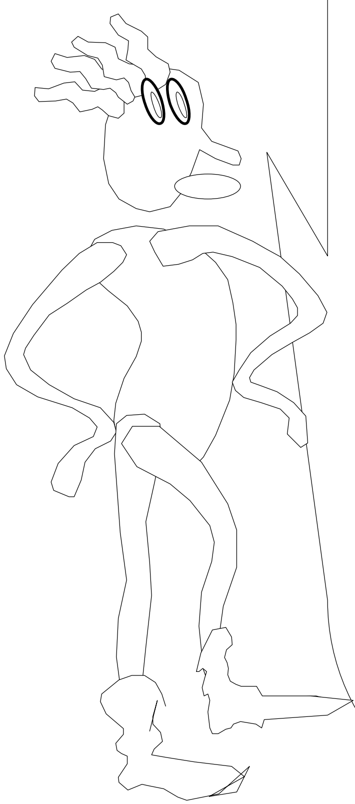
A logic may be SOUND but it still might not be complete.

A logic is “COMPLETE” if it can prove every statement that is True in the truth concept.



SOUND:
$PROVABLE_{S,L} \subset TRUTH_S$

COMPLETE:
$TRUTH_S \subset PROVABLE_{S,L}$



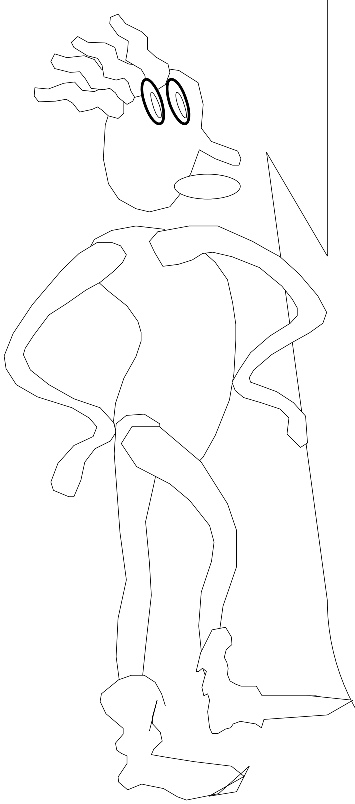
SOUND:

$\mathrm{PROVABLE}_{S,L} \subset \mathrm{TRUTH}_S$

COMPLETE:

$\mathrm{TRUTH}_S \subset \mathrm{PROVABLE}_{S,L}$

Ex: Axioms of Euclidean Geometry are known to be sound and complete for the truths of line and point in the plane.



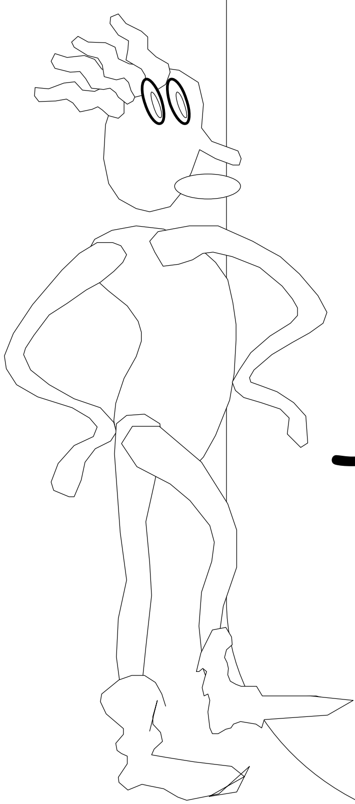
SOUND:

$\text{PROVABLE}_{S,L} \subset \text{TRUTH}_S$

COMPLETE:

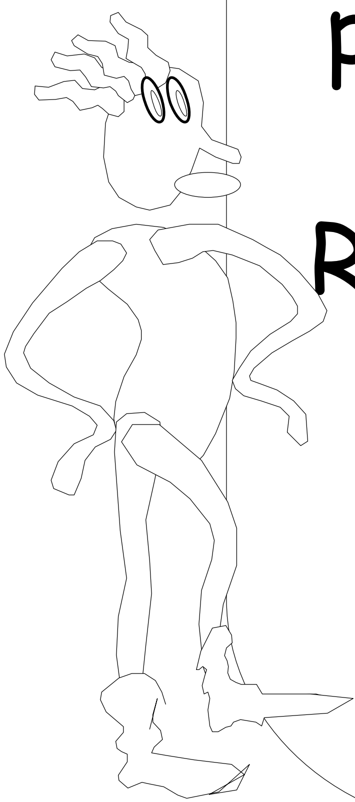
$\text{TRUTH}_S \subset \text{PROVABLE}_{S,L}$

SILLY FOO FOO 3 is sound and complete for the truth concept of strings having an even number of 1s.

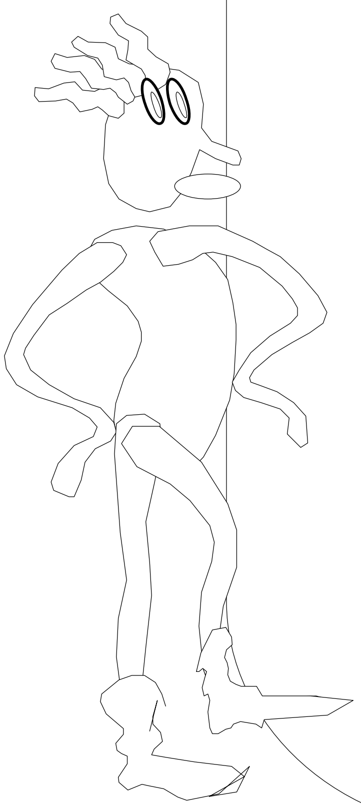


GENRALLY SPEAKING A
LOGIC WILL NOT BE ABLE
TO KEEP UP WITH TRUTH!

THE PROVABLE
CONSEQUENCES OF ANY
LOGIC ARE RECURSIVELY
ENUMERABLE. THE SET OF
TRUE STATEMENTS ABOUT
HALT/NON-HALTING
PROGRAMS IS NOT.



We have seen that the set of programs which do not halt on themselves – IS NOT RECURSIVELY ENUMERABLE.



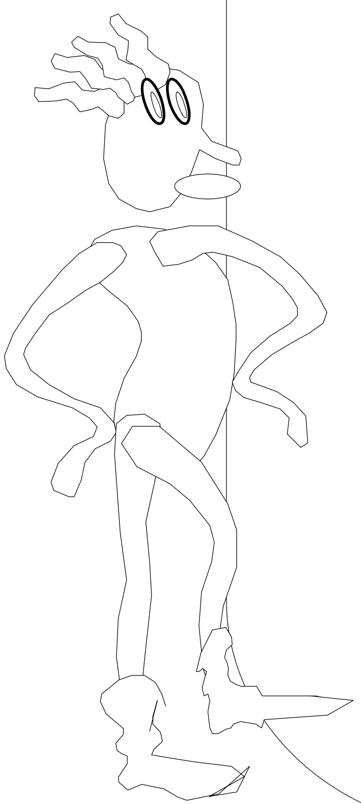
Given any LOGIC, we can enumerate all of its provable consequences.

Listing PROVABLE$_{LOGIC}$

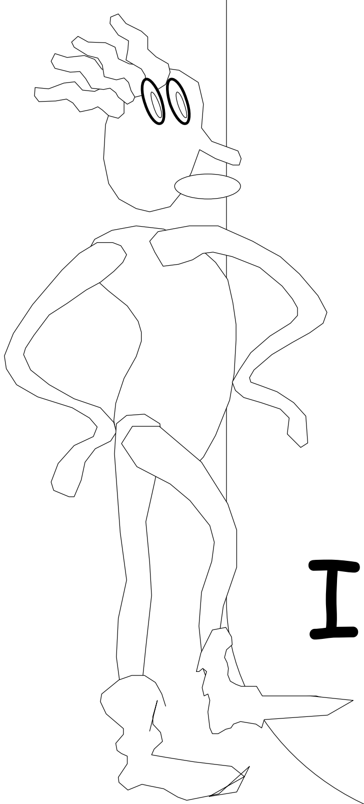
```
k;=0;

For sum = 0 to forever do

{Let PROOF loop through all strings of length k do
        {Let STATEMENT loop through strings of length <k do
         If proofcheck(STATEMENT, PROOF) = valid, output STATEMENT
        k++
         }
}
```

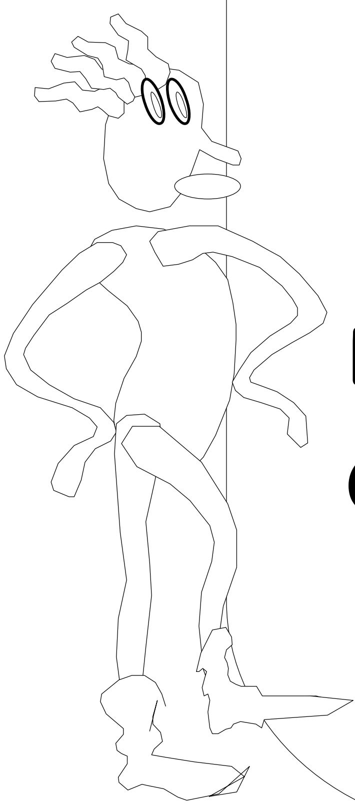


Let S be a language and TRUTH_S be a truth concept. We say that "TRUTH_S EXPRESSES THE HALTING PROBLEM" iff there exists a *computable* function r such that $r(x) \in \text{TRUTH}_S$ exactly when $x \in K$.



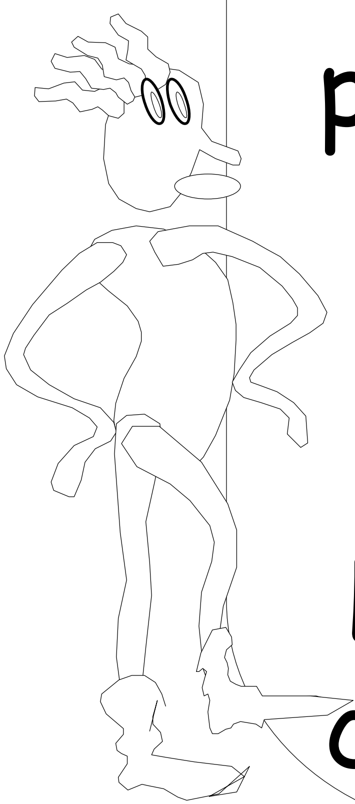
Let S be a language, L be a logic, and $TRUTH_S$ be a truth concept that expresses the halting problem.

THEOREM: If L is sound for $TRUTH_S$, then L is INCOMPLETE for $TRUTH_S$.



THEOREM: If L is sound for TRUTH_S, then L is INCOMPLETE for TRUTH_S.

L proves $r(x) \leftrightarrow x(x)$ doesn't halt. Thus, we can run x(x) and list theorems of L – one of them will tell us if x(x) halts.



FACT: Truth's of first order number theory (for every natural, for all naturals, plus, times, propositional logic) express the halting problem.

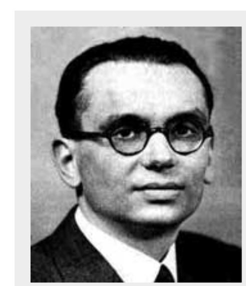
INCOMPLETENESS: No LOGIC for number theory can be sound and complete.

Hilbert's Question [1900]

Is there a foundation for mathematics that would, in principle, allow us to decide the truth of any mathematical proposition? Such a foundation would have to *give* us a clear procedure (algorithm) for making the decision.

GÖDEL'S INCOMPLETENESS THEOREM

In 1931, Gödel stunned the world by proving that for any consistent axioms F there is a true statement of first order number theory that is not provable or disprovable by F. I.e., a true statement that can be made using 0, 1, plus, times, for every, there exists, AND, OR, NOT, parentheses, and variables that refer to natural numbers.



GÖDEL'S INCOMPLETENESS THEOREM

Commit to any sound LOGIC F for first order number theory. Construct a *true* statement $GODEL_F$ that is not provable in your logic F.

YOU WILL EVEN BE ABLE TO FOLLOW THE CONTRUCTION AND ADMIT THAT $GODEL_F$ is a true statement that is missing from the consequences of F.

CONFUSE$_F$(P)

Loop though all sequences of symbols S

If S is a valid F-proof of “P halts”, then LOOP_FOR_EVER

If S is a valid F-proof of “P never halts”, then HALT

$GODEL_F$

$GODEL_F =$
$AUTO\_CANNIBAL\_MAKER(CONFUSE_F)$

Thus, when we run $GODEL_F$ it will do the same thing as:

$$CONFUSE_F(GODEL_F)$$

$GODEL_F$

Can F prove $GODEL_F$ halts?

Yes -> $CONFUSE_F(GODEL_F)$ does not halt

Contradiction

Can F prove $GODEL_F$ does not halt?

Yes -> $CONFUSE_F(GODEL_F)$ halts

Contradiction

$GODEL_F$

F can't prove or disprove that $GODEL_F$ halts.

$GODEL_F = CONFUSE_F(GODEL_F)$
Loop though all sequences of symbols S

If S is a valid F-proof of "$GODEL_F$ halts",
then LOOP_FOR_EVER

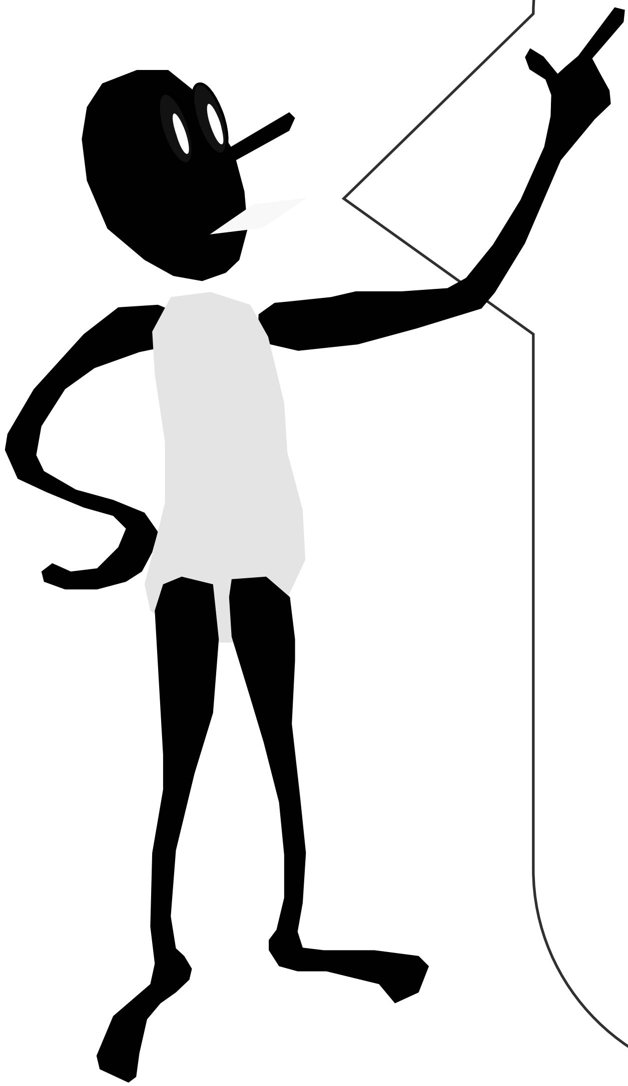
If S is a valid F-proof of "$GODEL_F$ never halts", then HALT

$GODEL_F$

F can't prove or disprove that $GODEL_F$ halts.

Thus $CONFUSE_F(GODEL_F) = GODEL_F$ will not halt. Thus, we have just proved what F can't.

F can't prove something that we know is true. It is not a complete foundation for mathematics.



In any logic that can express statements about programs and their halting behavior – can also express a Gödel sentence G that asserts its own improvability!

So what is mathematics?

THE DEFINING INGREDIENT OF MATHEMATICS IS HAVING A SOUND LOGIC - self-consistent for some notion of truth.

ENDNOTE

You might think that Gödel's theorem proves that people are mathematically capable in ways that computers are not. This would show that the Church-Turing Thesis is wrong.

Gödel's theorem proves no such thing!

We can talk about this over coffee.