Dangerous Knowledge:
Incompleteness – A Deeper Look at Gödel's Theorems
A great many different proofs of Gödel's theorem are now known, and the result is now considered easy to prove and almost obvious … no one loses sleep over it any more.

-- Gregory Chaitin, IBM, 1982
Paradoxes of self-reference

- The Cretan Paradox (Epimenides of Knossos, 7th Century BCE)
  … A Cretan (Epimenides) says, “All Cretans are liars.” (cf. Epistle to Titus 1:12)

- The Liar Paradox (Eubulides of Miletus, 4th Century BCE)
  … A man says, “What I am saying now is lies.” Is he telling the truth?

- “This sentence is false.” Is the sentence true?
Paradoxes of self-reference

- Gödel's statement (1931): “This theorem cannot be proved.” Is the theorem true or false?
  
  … If it cannot be proved, then it is true, yet you cannot prove it
  
  … If it can be proved, then it is false that it cannot be proved, which contradicts it
  
  … Shows that mathematics based on formal systems is either incomplete (there are truths that can't be proven) or inconsistent (you can prove contradictions)
What Is a Proof in a Formal System?

- A list of statements in the formal system’s language
- Starts with an axiom, definition or rule
- Ends with the desired theorem (“Q.E.D.”)
- Each statement is either:
  … Derived from the previous statement(s) by applying an axiom, definition, or rule
  … Itself another axiom, definition or rule
A Simple Formal System

- Symbols: ♠, ♥, ♣
- Variables: x

- Axiom: ♥♣

- Transformation Rules:
  ... (I)  x♠ → x♠♠
  ... (II)  ♥x → ♥xx
  ... (III)  ♣♣♣♣ → ♠
  ... (IV)  ♠♠ →

- Is ♥♠♣♣ a theorem of this system?
  ... ♥♣ (Axiom)
  ... ♥♣♣ (Rule II)
  ... ♥♣♣♣♣ (Rule II)
  ... ♥♠♣♣ Q.E.D. (Rule III)
Russell and Whitehead’s *Principia Mathematica* (1910-1913)

- Attempt to derive all mathematical truths from a formal system including axioms of set theory and arithmetic and rules of symbolic logic

- System includes:
  - **Variables** (p, q, etc.)
  - **Operations** (~, ∨, &, →, =, S)
  - **Punctuation** (“(“, “)”, “:”, “””, “{“, “}”)
  - **Quantifiers** (∀, ∃)
  - **Truth and Falsity**

- 200 pages to derive that $1 + 1 = 2$
Russell and Whitehead’s
*Principia Mathematica* (modern notation)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p, q, \ldots )</td>
<td>Variables</td>
</tr>
<tr>
<td>( \sim )</td>
<td>Not</td>
</tr>
<tr>
<td>( \lor )</td>
<td>Or</td>
</tr>
<tr>
<td>( \land )</td>
<td>And</td>
</tr>
<tr>
<td>( \rightarrow )</td>
<td>Implies (If \ldots \ then)</td>
</tr>
<tr>
<td>( S )</td>
<td>Successor of ((+1))</td>
</tr>
<tr>
<td>( = )</td>
<td>Equals (is the same as)</td>
</tr>
<tr>
<td>( \forall )</td>
<td>For All (instances of)</td>
</tr>
<tr>
<td>( \exists )</td>
<td>There Exists (at least one instance of)</td>
</tr>
<tr>
<td>( : )</td>
<td>Such That</td>
</tr>
<tr>
<td>( {} )</td>
<td>Set</td>
</tr>
<tr>
<td>( () )</td>
<td>Sub-statement</td>
</tr>
<tr>
<td>( ' )</td>
<td>Additional Variable (e.g., “( p, p', p'' ), \ldots”)</td>
</tr>
</tbody>
</table>
Example: The Law of Excluded Middle – “For all statements \( p \), either \( p \) is true, or \( p \) is not true”

\[ \forall p: (p \lor \sim p) \]

Proof:

\[ \forall p: \forall q: (p \rightarrow q) = (\sim p \lor q) \quad \text{(Definition of Implication)} \]

\[ q = p \quad \text{(Specification)} \]

\[ \forall p: (p \rightarrow p) = (\sim p \lor p) \quad \text{(Substitution)} \]

\[ \forall p: (\sim p \lor p) \rightarrow (p \lor \sim p) \quad \text{(Axiom of Commutativity)} \]

\[ \forall p: (p \lor \sim p) \quad \text{Q.E.D.} \quad \text{(Rule of Detachment)} \]
Recognizing Valid Proofs

- All proofs must be finite

- There is a procedure that is guaranteed to terminate for distinguishing valid proofs from invalid proofs
Recognizing Valid Proofs

Start with proof, desired theorem

Inspect next line of proof

Axiom, Rule, Definition?

Yes

Derived from a prior line?

No

Last line?

Yes

Desired theorem?

No

Invalid proof

Valid proof, QED
How Did Gödel Create His Statement?

- **Gödel numbering** – Getting mathematics to refer to itself
  … Ability to say “This theorem …”

- **Diagonalization** – Making a self-referential mathematical statement that says that it can not be proved
Gödel's Insight

- *Anything* can be a symbol in a formal system

- Why not use *numbers* as the symbols of the formal system’s *language*?
  
  ... Every statement must have a *unique* number
Symbols: ♠ = 1, ♥ = 2, ♣ = 3
Variables: x = 4
Punctuation: 5
Axiom: ♥♣ = 235
Transformation Rules:
  ... (I)  x♣ = 43 → x♣♣ = 432
  ... (II) ♥x = 24 → ♥xx = 244
  ... (III) ♣♣♣ = 333 → ♠ = 1
  ... (IV) ♠♠ = 22 →

Is ♥♠♣ = 2135 a theorem of this system?
  ... ♥♠ = 235 (Axiom)
  ... ♥♠♣ = 2335 (Rule II)
  ... ♥♠♣♣♣ = 233335 (Rule II)
  ... ♥♠♣♣ = 2135 Q.E.D. (Rule III)
Gödel Numbering a Simple Formal System

- ♥♠♣ = 2135 is a theorem of this system – proof:
  ... ♥♣ = 235 (Axiom)
  ... ♥♠♣ = 2335 (Rule II)
  ... ♥♠♣♠ = 233335 (Rule II)
  ... ♥♠♣♣ = 2135 Q.E.D. (Rule III)

- We can string the whole proof together to make one big number: 23523352333352135

- 23523352333352135 is the proof of 2135

- 23523352333352135 is a mathematical function of 2135 (in fact 23523352333352135 = 1131(2135^3) + 3746)
Gödel Numbering *Principia Mathematica*

- Every symbol is replaced by a *unique* numeric code

\[ \forall \; p \; : \; ( \; p \; \lor \; \sim \; p \; ) \]

626,262,636,362,262,616,223,262,323,611

Key:

\[ \forall = 626 \quad \lor = 616 \]
\[ p = 262 \quad \sim = 223 \]
\[ q = 263 \quad \rightarrow = 633 \]
\[ : = 636 \quad = = 111 \]
\[ ( = 362 \quad S = 555 \]
\[ ) = 323 \quad 0 = 000 \]

611 represents the end of a statement
Gödel Numbering

- Turns every statement of mathematics into a (very big) unique number
- Turns every mathematical proof (a series of statements) into a (very very big) unique number

\[ \forall p : \forall q : (p \rightarrow q) = (\sim p \lor q) \]
\[ q = p \]

\[ \forall p : (p \rightarrow p) = (\sim p \lor p) \]

\[ \forall p : (\sim p \lor p) \rightarrow (p \lor \sim p) \]

\[ \forall p : (p \lor \sim p) \]
- Once mathematical statements are numbers, then mathematical statements can refer to themselves! … Suppose a statement could contain the Gödel number of that very same statement

\[ G = \text{Gödel number of whole statement} \]
Gödel Numbering

- Problem: \( G \) won’t fit inside its own statement!
Gödel Numbering

- Problem: \( G \) won’t fit inside its own statement!
- Solution: Put an \textit{exact logical description} of \( G \) inside the statement instead of \( G \)!

\[
\begin{align*}
\text{Mathematical statement} & \quad \text{Self-reference!} \\
\text{Exact description of } G & \\
\end{align*}
\]

\( G = \) Gödel number of whole statement
Substitution

- The replacement of a variable by a specific value
  ... Let \( G = 362,262,111,262,323,611 \)
    \[
    \begin{array}{c}
    \text{p} = \text{p} \\
    \end{array}
    \]
  ... Suppose we substitute the value of “1” for \( p \):
  ... Then we get \( q = 362,555,000,100,555,000,323,611 \)
    \[
    \begin{array}{c}
    \text{S} 0 = \text{S} 0 \\
    \end{array}
    \]
- Gödel represented this operation by \( \text{Sub}(G, p, n, q) \)
  ... \( G \) = the original statement’s Gödel number
  ... \( p \) = the variable being replaced in the original statement
  ... \( n \) = the specific value that replaces the variable
  ... \( q \) = the new statement’s Gödel number
- \( \text{Sub}(G, p, G, q) \) represents the substitution of a statement’s own Gödel number \( G \) into itself
Diagonalization

- Let $T$ be the unique Gödel number of some theorem
- Let $P$ be the (much bigger) unique Gödel number of the theorem’s proof
- Given that the procedure for determining whether the proof is valid is guaranteed to terminate
- Then there is a statement (a computable function) representing the mathematical relationship between $T$ and $P$
  … Write this equation as $\text{Math-function}(T) = P$
Gödel Numbering a Simple Formal System

- ♥♠♣ = 2135 is a theorem of this system – proof:
  ... ♥♣ = 235               (Axiom)
  ... ♥♣♣ = 2335           (Rule II)
  ... ♥♣♣♣ = 233335       (Rule II)
  ... ♥♠♣♣ = 2135      Q.E.D. (Rule III)

- We can string the whole proof together to make one big number: 23523352333352135

- 23523352333352135 is the proof of 2135

- 23523352333352135 is a mathematical function of 2135 (in fact 23523352333352135 = 1131(2135^3) + 3746)
Gödel Numbering a Simple Formal System

- ♥♣♣ = 2135 *is* a theorem of this system – proof:
  ... ♥♣ = 235 (Axiom)
  ... ♥♣♣ = 2335 (Rule II)
  ... ♥♣♣♣♣ = 233335 (Rule II)
  ... ♥♣♣♣ = 2135 Q.E.D. (Rule III)

- We can string the whole proof together to make one big number: 23523352333352135
  - 23523352333352135 is the *proof* of 2135

- 23523352333352135 is a *mathematical function* of 2135
  (in fact 23523352333352135 = 1131(2135^3) + 3746)
Gödel Numbering a Simple Formal System

- ♥♠♣ = 2135 is a theorem of this system – proof:
  ... ♥♣ = 235 (Axiom)
  ... ♥♠♣ = 2335 (Rule II)
  ... ♥♠♠♣ = 233335 (Rule II)
  ... ♥♠♣♣ = 2135 Q.E.D. (Rule III)

- We can string the whole proof together to make one big number: 23523352333352135
- 23523352333352135 is the proof of 2135
- 23523352333352135 is a mathematical function of 2135 (in fact 23523352333352135 = 1131(2135^3) + 3746)

Math-function (T) = P
Diagonalization

- There is also a statement $\exists P: \text{Proof-Pair}(T, P)$

  … “There exists a number $P$ which is the Gödel number of the proof of the theorem whose Gödel number is $T$”

  … True if and only if $\text{Math-function}(T) = P$

  … $T$ and $P$ must be positive integers: a Diophantine equation
Gödel Numbering a Simple Formal System

- ♥♠♠ = 2135 is a theorem of this system – proof:
  ... ♥♠ = 235 (Axiom)
  ... ♥♠♠ = 2335 (Rule II)
  ... ♥♠♠♠♠ = 233335 (Rule II)
  ... ♥♠♠ = 2135 Q.E.D. (Rule III)

- We can string the whole proof together to make one big number: 23523352333352135
  - 23523352333352135 is the proof of 2135
  - 23523352333352135 is a mathematical function of 2135 (in fact 23523352333352135 = 1131(2135³) + 3746)

Math-function (T) = P

Proof-Pair (T, P) = “P = 1131T³ + 3746” [a Diophantine equation]
Diagonalization

- Now suppose $T$ is equal to $G$, the Gödel number of the entire statement in which the Proof-Pair function appears!
- Now suppose $T$ is equal to $G$, the Gödel number of the entire \textit{statement} in which the \textit{Proof-Pair} function appears!

- Assert that there is \textit{no} number that forms a \textit{Proof-Pair} with $G$:

$$\sim \exists P: \exists q: (\text{Proof-Pair}(T, P) \& \text{Sub}(G,T,G,q))$$

$G =$ Gödel number of entire statement
Diagonalization

- Now suppose $T$ is equal to $G$, the Gödel number of the entire statement in which the Proof-Pair function appears!

- Assert that there is no number that forms a Proof-Pair with $G$:

  $\neg \exists P \exists q : (\text{Proof-Pair}(T, P) \land \text{Sub}(G, T, G, q))$

  $G = \text{Gödel number of entire statement}$

- This statement says, “There is no proof for the theorem $(T)$ with a Gödel number equal to $G$”
  … But since $G$ is the statement itself, this is equivalent to saying, “This theorem can not be proved”
Implications of Gödel's Theorems

- **Mathematics**: Does mathematics exist independent of the mind, or does the mind create it?
- **Physics**: Can there be a true “theory of everything” if mathematics is essentially incomplete?
- **Philosophy**: The nature of truth
- **Computers**: The birth of computing theory
- **Artificial intelligence**: Can a computer or machine do everything a mind can do? Or does the power of the mind exceed that of any conceivable mechanism or computational object?
Gödel's Theorems and Mathematics

... I think that your result has solved negatively the foundational question: there is no rigorous justification for classical mathematics.

-- Letter from John Von Neumann to Kurt Gödel, Fall 1930
Gödel's Theorems and Mathematics

- All mathematics is not captured by a single all-embracing system of logic
  … Some truths are only accessible through intuition

- Purely finite methods of reasoning and proof will not guarantee the soundness of the foundations of mathematics

- Mathematical Platonism is a rational philosophical standpoint
  … There are mathematical facts that are independent of empirical justification
  … Mathematical truth is discovered, not created
Gödel's Theorems and Mathematics

... The existence of unsolvable problems would seem to disprove the view that mathematics is our own creation; for a creator necessarily knows all properties of his creatures ... [although] we build machines and still cannot predict their behavior in every detail ... [that objection] is very poor ... For we don’t create machines out of nothing, but build them out of some given material.

-- Kurt Gödel
Gödel's Theorems and Physics

… it seems that on the strength of Gödel's theorem … there are limits to the precision of certainty, that even in the pure thinking of theoretical physics there is a boundary …

-- Stanley Jaki, 1966

One may speculate that undecidability is common in all but the most trivial physical theories. Even simply formulated problems in theoretical physics may be found to be provably insoluble.

-- Stephen Wolfram, 1994
Gödel's Theorems and Physics

Gödel proved that … the world of pure mathematics is inexhaustible … I hope that an analogous situation exists in the physical world. If my view … is correct, it means that the world of physics and astronomy is also inexhaustible … there will always be new things happening, new information coming in, new worlds to explore, a constantly expanding domain of life, consciousness, and memory.

-- Freeman Dyson, 2004
Well, one day I was at the Institute for Advanced Study, and I went to Gödel's office, and there was Gödel. It was winter and Gödel had an electric heater and had his legs wrapped in a blanket. I said, “Professor Gödel, what connection do you see between your incompleteness theorem and Heisenberg’s uncertainty principle?” And Gödel got angry and threw me out of his office!

-- John Wheeler
Gödel's Theorems and the Nature of Truth

- Truth and proof are different
  ... Not everything true can be proven; truth is stronger than proof

- False and “not true” are different
  ... Things might be undecidable: not true, but also not false
Alfred Tarski

- Polish mathematician and philosopher; joined Berkeley math department in 1942 and became US citizen 1945

- Defined truth in formal systems

- Proved that no formal system can define its own truths

Alfred Tarski (1901-1983)
Truth in Formal Systems

- In any given formal system, truth can only be defined by reference to a *meta-system* outside that formal system
Truth in Formal Systems

- In any given formal system, truth can only be defined by reference to a *meta-system* outside that formal system.

**Formal system**
- Axioms, rules, etc.
- Theorems, Proofs
Truth in Formal Systems

- In any given formal system, truth can only be defined by reference to a *meta-system* outside that formal system.

<table>
<thead>
<tr>
<th>Meta-system</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Additional information (more axioms, rules, etc.)</td>
</tr>
<tr>
<td>- Truths</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Formal system</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Axioms, rules, etc.</td>
</tr>
<tr>
<td>- Theorems, Proofs</td>
</tr>
</tbody>
</table>
Truth in Formal Systems

- In any given formal system, truth can only be defined by reference to a *meta-system* outside that formal system.

**Meta-system**
- Additional information (more axioms, rules, etc.)
- Truths

**Formal system**
- Axioms, rules, etc.
- Theorems, Proofs
Tarski’s Material Adequacy Conditions

- “P” is **true** in a formal system if and only if P **actually obtains in the meta-system**
  
  ... Example: “Snow ∈ {white things}” if and only if snow is white

<table>
<thead>
<tr>
<th>Meta-system (&quot;real world&quot;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Snow is white</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Formal system (logic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Snow ∈ {white things}”</td>
</tr>
</tbody>
</table>
Gödel's Theorems and the Birth of Computing Theory

- *Decidability (proof)* and *computation (mechanical process)* are intimately related

- Gödel numbering makes it possible to precisely define what a *computable function* is

- The existence of *undecidable* mathematical statements implies that some computations may never come to a stop (i.e., may never produce a result)
David Hilbert

- Leading German mathematician of late 19th and early 20th century
- Trained, at Gottingen, many of 20th century’s foremost mathematicians
- In 1900, proposed 23 problems that set the course of 20th century mathematical research
- Believed it was possible to develop a finite logical procedure for determining the truth or falsity of any mathematical statement (Hilbert’s *Entscheidungsproblem*, or “decision problem”)
Alan Turing
1912-1954

- Developed theory of binary computation and programmability ("Turing machine")
- Built on work of Gödel
- Developed computing devices used in breaking Germany’s Enigma code during WW II
Turing’s “Bombe” at Bletchley Park
Alonzo Church

- American mathematician; taught at Princeton and UCLA; Turing’s Ph.D. thesis advisor

- With Turing, developed the Church-Turing Thesis, which states that anything that can be computed can be computed by a mechanical process

- Indicated in 1936 that Hilbert’s Entscheidungsproblem is undecidable for one model of computation (the lambda calculus)
“The greatest improvement was made possible through the precise definition of the concept of finite procedure . . . This concept . . . is equivalent to the concept of a “computable function of integers” . . . The most satisfactory way, in my opinion, is that of reducing the concept of finite procedure to that of a machine with a finite number of parts, as has been done by the British mathematician Turing.”

-- Kurt Gödel, Gibbs lecture, 1951
Schematic of "Turing Machine"
The “Turing Machine”

- Endless tape divided into “squares”
  … Stores in-coming “program” and data, coded in “symbols”
    (or “marks”, i.e., 1’s, and “blanks,” i.e., 0’s)
  … Serves as memory

- “Read/write head” with internal operating rules (or “states”)

- Performs any of the following functions
  … Move tape forward one square and read the symbol
  … Move tape backward one square and read the symbol
  … Write a mark (i.e. a “1”) in the current square
  … Erase a mark (i.e. write a blank, or “0”) in the current square
  … Change the internal state

- Can do anything any computer can (universal computation)
  by simulating any target machine
A Lego Turing Machine
Turing’s Insight

- Computations are **codes**, like Gödel numbers
  … Given a Turing machine, the instructions (or **program**) for any computation can be encoded as **binary numbers** on the tape

- Enables Turing machines to simulate other Turing machines
Turing’s Formulation of the *Entscheidungsproblem*

- Is there a way of telling whether a computation (mechanical process) that is trying to decide if a given mathematical statement is provable or not will come to a stop?
Enumerability
A Countable List of All Theorem-Proof Programs

Binary Code for Program

0 (0)
1 (1)
2 (10)
3 (11)
4 (100)
5 (101)
6 (110)
...
G (1101001110011100010110111000001010010 ... 01)
...
Recognizing Valid Proofs

Start with proof, desired theorem

Inspect next line of proof

Axiom, Rule, Definition?

Yes

Last line?

No

Yes

Derived from a prior line?

No

Yes

Desired theorem?

No

Invalid proof

Valid proof, QED

(by means of axioms, definitions, or rules)
Recognizing Valid Proofs

1. Start with proof, desired theorem
2. Encoded in binary form on Turing machine tape
3. Inspect next line of proof
4. Axiom, Rule, Definition?
   - Yes
   - No
5. Derived from a prior line?
   - Yes
   - No
6. Last line?
   - Yes
   - No
7. Desired theorem?
   - Yes
   - No
8. Invalid proof
9. Valid proof, QED

(by means of axioms, definitions, or rules)
Recognizing Valid Proofs

Start with proof 0, desired theorem

Encoded in binary form on Turing machine tape

Valid proof, QED

Go to next proof
"Hilbert’s Dream Machine"

Start with proof 0, desired theorem

Encoded in binary form on Turing machine tape

(by means of axioms, definitions, or rules)

Go to next proof

Valid proof, QED

Invalid proof

Desired theorem?

Yes

Derived from a prior line?

Yes

Axiom, Rule, Definition?

Yes

Last line?

Yes

Inspect next line of proof

No

No

No

No
“Hilbert’s Dream Machine”

- This machine is not guaranteed to stop!
  … Could enter an “infinite loop” if desired theorem has no proof

- Can we tell whether the machine is looping infinitely? Turing showed that in general we can not!
Turing’s Infinite Loop Tester

Input = coded program

Simulate input program

Will program stop?

Yes

Write “1”

No

Write “0”
Turing’s Modified Infinite Loop Tester

Input = coded program

Simulate input program

Will program stop?

Yes → Infinite loop

No → Stop
Turing’s Modified Infinite Loop Tester

Input = modified infinite loop tester

Simulate input program

Will program stop?

Yes → Infinite loop

No → Stop
Gödel's Theorems and Artificial Intelligence (AI)

- Is the mind a formal system?
  ... If so, the Church-Turing Thesis implies that cognition = computation

- If it is, then what if we simulate it on a Turing machine (or computer)?

- If it is not, then why not?
The Fundamental Premise of AI

Mind (Symbols, Models, Consciousness)

The Brain and Central Nervous System

Neural Substrate (Biochemistry)
The Fundamental Premise of AI

- Mind (Symbols, Models, Consciousness)
- The Brain and Central Nervous System
- Neural Substrate (Biochemistry)
- "Ultimate" Substrate (Physics)
- Macroscopic World
- Microscopic World

Knowledge representation
The Fundamental Premise of AI

- **AI Program** (Symbols, Models, Self-Reference)
- **Lower Software Levels** ("Operating System")
  - **Electronic Substrate**
- **Mind** (Symbols, Models, Consciousness)
  - **The Brain and Central Nervous System**
  - **Neural Substrate** (Biochemistry)
- **Macroscopic World**
- **Microscopic World**
  - "Ultimate" Substrate (Physics)

Functional equivalence (based on Church-Turing Thesis)
The Fundamental Premise of AI

- **Macroscopic World**
  - Knowledge representation
  - Mind (Symbols, Models, Consciousness)

- **Microscopic World**
  - "Ultimate" Substrate (Physics)

- **Formal Systems!**
  - Functional equivalence (based on Church-Turing Thesis)
  - AI Program (Symbols, Models, Self-Reference)

- **Lower Software Levels** ("Operating System")
  - Electronic Substrate

- **The Brain and Central Nervous System**
  - Neural Substrate (Biochemistry)

- **Macrosopic World**
  - Macroscopic World
The Fundamental Premise of AI

- Macroscopic World
  - Mind (Symbols, Models, Consciousness)
  - The Brain and Central Nervous System
  - Electronic Substrate
  - Neural Substrate (Biochemistry)
- Microscopic World
  - "Ultimate" Substrate (Physics)
  - Lower Software Levels ("Operating System")
- AI Program (Symbols, Models, Self-Reference)

Functional equivalence (based on Church-Turing Thesis)

Formal Systems?
The Fundamental Premise of AI

Not a Formal System!

AI Program (Symbols, Models, Self-Reference)

Macroscopic World

Mind (Symbols, Models, Consciousness)

Knowledge representation

Lower Software Levels (“Operating System”)

The Brain and Central Nervous System

Neural Substrate (Biochemistry)

Microscopic World

Electronic Substrate

“Ultimate” Substrate (Physics)

Functional equivalence (based on Church-Turing Thesis)
- Outputs activated if total of all inputs exceeds activation threshold
  … Some inputs inhibitory (-), some excitatory (+)
A Synapse

- Amount of neurotransmitter in synapse determines “weight” or “strength” of signal going from axon to dendrite
- “Weight” changes depending on activation frequency
McCullogh and Pitts
Artificial Neurons (1943)

Warren Sturgis McCullogh (1898-1969)

Walter Pitts (1923-1969)
- Each input ($x_j$) is multiplied by a synaptic weight ($w_{kj}$)
- The weighted inputs are added up ($\Sigma \rightarrow v_k$)
- The final sum ($v_k$) is multiplied by an activation function ($\varphi$) and the threshold ($\theta_k$) is subtracted to create the output ($y_k$)
Frank Rosenblatt – the Perceptron (1957-1960)

Frank Rosenblatt (1928-1971)  
Rosenblatt with the Mark 1 Perceptron at Cornell Aeronautical Laboratory
A Perceptron Neural Network

Stimulus (from sensors, etc.)

Classifies stimuli into groups or categories
Marvin Minsky

- American cognitive scientist and computer scientist; considered one of the fathers of artificial intelligence research

- Co-founded the MIT Artificial Intelligence laboratory in 1959

- In 1969, co-authored (with Seymour Papert) the book *Perceptrons*, which attacked neural networks as vehicles for achieving artificial intelligence, showing that they could not perform certain logic and memory functions
John Hopfield

- American physicist and molecular biologist at Berkeley, Caltech and Princeton; Dirac medal, 2002

- In 1982, invented a neural network with feedback, today known as a *Hopfield network*
An Artificial Neural Network with Feedback
Brains vs. Minds

- Even if the brain/central nervous system is a formal system, does this actually imply that the mind is a formal system?

- Philosophy’s *mind-body problem*
  … *Monism*: Yes – mind and body are not separable
  … *Dualism*: No – mind is distinct from body
J. R. Lucas

- British philosopher, emeritus professor at Oxford

- Has written on the philosophy of science and mathematics, the mind, free will and determinism

- Best known for his 1961 paper, “Minds, Machines and Gödel”
Gödel's theorem seems to me to prove that Mechanism is false, that is, that minds cannot be explained as machines.

- J. R. Lucas, “Minds, Machines and Gödel”, 1961
I have come to cherish incompleteness for the support it lends to mechanism ... and to Turing’s thesis in particular.

-- Judson Webb, Associate Professor of Philosophy, Boston University: “Mechanism, Mentalism and Metamathematics”, 1980
Lucas’s Argument

- Machines are formal systems
- If a machine is complex enough to simulate a mind, it must be complex enough to represent mathematics
- Then the machine will have a Gödel statement $G$, which it cannot prove to be true, but which a mind other than the machine can see is true
- Therefore there is something a mind can do that a machine cannot
Counter Arguments to Lucas

- Not all conceivable machines are formal systems
  … Example: Quantum computers

- If a machine is complex enough, perhaps even a mind might not be able to formulate its Gödel statement

- Like formal systems, minds are also inconsistent or incomplete
  … Example: People may prefer A to B, and B to C, yet prefer C to A
  … Turing: “If a machine is expected to be infallible, it cannot also be intelligent”
Turing ... gives an argument which is supposed to show that mental procedures cannot go beyond mechanical procedures ... What Turing disregards completely is the fact that mind, in its use, is not static, but constantly developing ... there may exist systematic means of actualizing this development, which could form part of the procedures. Therefore, although at each stage the number and precision of the abstract terms at our disposal may be finite, both (and therefore also Turing’s number of distinguishable states of mind) may converge toward infinity in the course of the application of the procedure.

-- Kurt Gödel, “A Philosophical Error in Turing’s Work”, 1972 (italics original)
Roger Penrose

- English mathematician and philosopher; Rouse Ball professor of mathematics at Oxford
- Collaborated with Stephen Hawking
- Argues that Gödel's results imply that new “non-computational” theories of physics are needed in order to explain human intelligence
- Believes “non-computational” physics would provide a usable theory of quantum gravity
What is “Non-computational” Physics?

- Large-scale quantum coherence phenomena that decohere (undergo wavefunction collapse) under the influence of quantum gravity

- In the brain, this would happen inside microtubules that are part of the sub-cellular structure of neurons

- Implies that even the brain might not be a formal system
Quantum Coherence

- Example: 2 electrons interact
  … Each electron’s spin is either up (↑) or down (↓)
  … Spins must be opposite after the interaction

A ↑ B ↓ or A ↓ B ↑
Quantum Coherence

- Example: 2 electrons interact
  ... Each electron’s spin is either up (↑) or down (↓)
  ... Spins must be opposite after the interaction

A \(\uparrow\) B \(\downarrow\) or A \(\downarrow\) B \(\uparrow\)
Quantum Coherence

- Example: 2 electrons interact
  … Each electron’s spin is either up (\(\uparrow\)) or down (\(\downarrow\))
  … Spins must be opposite after the interaction

\[
\begin{align*}
A \uparrow & \quad B \downarrow \quad \text{or} \quad A \downarrow & \quad B \uparrow
\end{align*}
\]

\[
\text{Entangled}
\]

A and B are \textit{coherent}
Quantum Coherence

- Example: 2 electrons interact
  … Each electron’s spin is either up (↑) or down (↓)
  … Spins must be opposite after the interaction

\[
\text{A} \uparrow \text{B} \downarrow \quad \text{or} \quad \text{A} \downarrow \text{B} \uparrow
\]

A and B are **coherent**
Quantum Decoherence

- Example: 2 electrons interact
  … Each electron’s spin is either up (↑) or down (↓)
  … Spins must be opposite after the interaction

\[
\begin{array}{cc}
A & \uparrow \quad B & \downarrow \\
\text{or } A & \downarrow & B & \uparrow
\end{array}
\]
Quantum Decoherence

- Example: 2 electrons interact
  ... Each electron’s spin is either up (↑) or down (↓)
  ... Spins must be opposite after the interaction

A ↑ B ↓ or A ↓ B ↑

Spin of A *instantaneously* set regardless of distance between A & B

2 electrons are now **decoherent**
Coherence and Computability

- Example: 2 electrons interact
  … Each electron’s spin is either up (↑) or down (↓)
  … Spins must be opposite after the interaction

A  
\[ \text{↑} \quad \text{↓} \quad \text{or} \quad \text{↓} \quad \text{↑} \]

A and B are coherent

What is happening to the individual electrons in the coherent state can’t be computed!
Coherence and Computability

- Example: 2 electrons interact
  … Each electron’s spin is either up (↑) or down (↓)
  … Spins must be opposite after the interaction

$$A \uparrow B \downarrow \quad \text{or} \quad A \downarrow B \uparrow$$

The best we can do is assign **probabilities** to the alternatives we might find if we **observe** the electrons.
Microtubules

Electron micrograph of microtubules in neuronal cell cytoplasm

25 nm
“Either...the human mind (even within the realm of pure mathematics) infinitely surpasses the powers of any finite machine, or else there exist absolutely unsolvable Diophantine problems.”

-- Kurt Gödel, Gibbs Lecture, Yale University, 1951
Gödel's Incompleteness Theorems

- Either the formal system of mathematics is inconsistent (some untruths or contradictions are theorems) or incomplete (some truths are not theorems)

- Why?
  … Gödel showed that it is possible to create the following mathematical statement \( G \): \( G = \text{“This theorem can not be proved”} \)
  … If \( G \) is a theorem, then the system is inconsistent
  … If \( G \) is not a theorem, then the system is incomplete

- The same paradox arises in any formal system that can represent addition, multiplication and comparison of natural numbers
  … No such formal system can prove its own consistency
Gödel Numbering

- Turns every statement of mathematics into a (very big) number
- Turns every mathematical proof (a series of statements) into a (very very big) number

\[ \forall p : \forall q : ( p \rightarrow q ) = ( \neg p \vee q ) \]  
(Definition of Implication)

\[ q = p \]  
(Substitution)

\[ \forall p : ( p \rightarrow p ) = ( \neg p \vee p ) \]  
(Substitution)

\[ \forall p : ( \neg p \vee p ) \rightarrow ( p \vee \neg p ) \]  
(Axiom of Commutativity)

\[ \forall p : ( p \vee \neg p ) \]  
(Rule of Detachment),

Q.E.D
- Using the $\text{Sub}(G, p, G, q)$ function Gödel was able to make an exact logical description of inserting $G$ into itself

$G = \text{“p can not be proven”}$

$\text{Sub}(G, p, G, q)$

$q = \text{“Can not be proven” can not be proven}$
## Diagonalization

### Is A Mathematical Statement Provable?

<table>
<thead>
<tr>
<th>Gödel Number of $\exists P: \text{Proof-Pair}(T,P)$</th>
<th>Gödel Number of Theorem (T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>\vdots</td>
<td></td>
</tr>
<tr>
<td>n</td>
<td></td>
</tr>
<tr>
<td>n+1</td>
<td></td>
</tr>
<tr>
<td>\vdots</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
</tr>
<tr>
<td>\vdots</td>
<td></td>
</tr>
</tbody>
</table>
## Diagonalization

### Is A Mathematical Statement Provable?

<table>
<thead>
<tr>
<th>Gödel Number of $\exists P: Proof-Pair(T,P)$</th>
<th>Gödel Number of Theorem (T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 1 2 3 … n n+1 … G …</td>
</tr>
<tr>
<td>0</td>
<td>No No No No No No No No No</td>
</tr>
<tr>
<td>1</td>
<td>No No No No No No No No No</td>
</tr>
<tr>
<td>2</td>
<td>No No No No No No No No No</td>
</tr>
<tr>
<td>3</td>
<td>No No No No No No No No No</td>
</tr>
<tr>
<td>…</td>
<td>… … … … … … … … … … … …</td>
</tr>
<tr>
<td>n</td>
<td>No No No No No No No No No</td>
</tr>
<tr>
<td>n+1</td>
<td>No No No No No Yes No No No</td>
</tr>
<tr>
<td>…</td>
<td>… … … … … … … … … … … …</td>
</tr>
<tr>
<td>G</td>
<td>No No No No No No No No No</td>
</tr>
<tr>
<td>…</td>
<td>… … … … … … … … … … … …</td>
</tr>
</tbody>
</table>
Halting
Will Hilbert’s Dream Machine stop?

Turing Code for Proof of Theorem (P) 0 1 2 3 … T … P … N … G …

0
1
2
3
…
T
…
P
…
N
…
G
**Halting**

(Yes = invalid)

Will Hilbert’s Dream Machine stop? *(Yes = valid)*

<table>
<thead>
<tr>
<th>Turing Code for Proof of Theorem (P)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>T</th>
<th>P</th>
<th>N</th>
<th>...</th>
<th>G</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>?</td>
</tr>
</tbody>
</table>
Diagonalization

\[ \forall p : \forall q : (p \rightarrow q) = (\neg p \lor q) \]

\[ q = p \]

\[ \forall p : (p \rightarrow p) = (\neg p \lor p) \]

\[ \forall p : (\neg p \lor p) \rightarrow (p \lor \neg p) \]

\[ T = \forall p : (p \lor \neg p) \]
Diagonalization

\[ T = \forall p : \forall q : (p \rightarrow q) = (\neg p \lor q) \]

\[ 263,111,262,611, \]

\[ q = p \]

\[ (p \rightarrow p) = (\neg p \lor p) \]

\[ (\neg p \lor p) \rightarrow (p \lor \neg p) \]

\[ \forall p : (p \lor \neg p) \]

\[ 626,262,636,362,223,262,323,611 \]
Diagonalization

\[ P = \]

\[ 626,262,636,626,263,636,362,262,633,263,323,111,362,223,262,616,262,323,611, \]
\[ \forall p : \forall q : ( p \rightarrow q ) = ( \sim p \vee q ) \]
\[ 263,111,262,611, \]
\[ q = p \]
\[ 626,262,636,362,262,633,262,323,111,362,223,262,616,262,323,611, \]
\[ \forall p : ( p \rightarrow p ) = ( \sim p \vee p ) \]
\[ \forall p : ( \sim p \vee p ) \rightarrow ( p \vee \sim p ) \]

\[ T = 626,262,636,362,262,616,223,262,323,611 \]
\[ \forall p : ( p \vee \sim p ) \]

\[ N = \text{Godel number of } \sim \exists P : \exists T : \text{Proof-Pair}(T, P) \]
# Diagonalization

Can a Mathematical Statement Be Proved?

<table>
<thead>
<tr>
<th>Gödel Number of Math-Function(T)</th>
<th>Gödel Number of Theorem (T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 1 2 3 ... T ... P ... N ... G ...</td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

Math-Function(G)?
Diagonalization
Can a Mathematical Statement Be Proved?

<table>
<thead>
<tr>
<th>Gödel Number of Math-Function(T)</th>
<th>Gödel Number of Theorem (T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>P</td>
<td>No</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>N</td>
<td>No</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Math-Function(G) ?</td>
<td>No</td>
</tr>
</tbody>
</table>
## Diagonalization

### Can a Mathematical Statement Be Proved?

<table>
<thead>
<tr>
<th>Gödel Number of Math-Function(T)</th>
<th>Gödel Number of Theorem (T)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>…</th>
<th>T</th>
<th>…</th>
<th>P</th>
<th>…</th>
<th>N</th>
<th>…</th>
<th>G</th>
<th>…</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>1</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>T</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>P</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>N</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Math-Function(G)?</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

### Gödel Number of Math Function:

1. The Gödel number of a mathematical statement is a unique number assigned to the statement.
2. Different statements have different Gödel numbers.
3. The Gödel number of a mathematical statement can be represented by a sequence of symbols from a well-defined alphabet.

### Gödel Number of Theorem (T):

1. The Gödel number of a theorem is a unique number assigned to the theorem.
2. The Gödel number of a theorem can be represented by a sequence of symbols from a well-defined alphabet.
3. The Gödel number of a theorem is the Gödel number of the mathematical function that represents the theorem.

### Mathematical Function:

1. A mathematical function is a rule that assigns to each element of a set a unique element of another set.
2. The mathematical function can be represented by a sequence of symbols from a well-defined alphabet.
3. The mathematical function is the Gödel number of the theorem.

### Diagonalization:

1. Diagonalization is a proof technique used in mathematical logic to show that a certain statement cannot be proved within a given formal system.
2. The technique involves constructing a sequence of statements, each of which makes a diagonal claim about the truth value of the statement that precedes it.
3. The sequence of statements is constructed in such a way that it eventually leads to a contradiction, thereby proving that the statement cannot be proved within the given formal system.
Diagonalization

Is A Mathematical Statement True?

<table>
<thead>
<tr>
<th>Gödel Number of $\sim \exists P: \exists T$: $Proof-Pair(T,P)$</th>
<th>Gödel Number of Theorem (T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 ...</td>
<td>T ... P ... N ... G ...</td>
</tr>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
</tr>
</tbody>
</table>
## Diagonalization

### Is A Mathematical Statement True?

<table>
<thead>
<tr>
<th>Gödel Number of $\sim \exists P: \exists T$: <strong>Proof-Pair</strong>(T,P)</th>
<th>Gödel Number of Theorem (T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 1 2 3 … T … P … N … G …</td>
</tr>
<tr>
<td>No</td>
<td>No No No No No No No No No</td>
</tr>
<tr>
<td>No</td>
<td>No No No No No No No No No</td>
</tr>
<tr>
<td>No</td>
<td>No No No No No No No No No</td>
</tr>
<tr>
<td>No</td>
<td>No No No No No No No No No</td>
</tr>
<tr>
<td>…</td>
<td>… … … … … … … … … … …</td>
</tr>
<tr>
<td>T</td>
<td>No No No No No No No No No</td>
</tr>
<tr>
<td>…</td>
<td>… … … … … … … … … … …</td>
</tr>
<tr>
<td>P</td>
<td>No No No No No No No No No</td>
</tr>
<tr>
<td>…</td>
<td>… … … … … … … … … … …</td>
</tr>
<tr>
<td>N</td>
<td>No No No No No Yes No No No</td>
</tr>
<tr>
<td>…</td>
<td>… … … … … … … … … … …</td>
</tr>
<tr>
<td>G</td>
<td>No No No No No No No No No</td>
</tr>
<tr>
<td>…</td>
<td>… … … … … … … … … … …</td>
</tr>
</tbody>
</table>
# Diagonalization

Is A Mathematical Statement True?

<table>
<thead>
<tr>
<th>Gödel Number of ( \sim \exists P: \exists T ): Proof-Pair ( T, P )</th>
<th>Gödel Number of Theorem (T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

...
Tarski’s Material Adequacy Conditions

- A statement in a formal system “$S(p, q, ...)$” is **true relative to an assignment of values** to its variables $p, q, ...$ if and only if the corresponding values of the variables **actually stand in the meta-system in the relation expressed by $S$**

... Example: “$(q = 2p) \& (q \in \{\text{quarts}\} \& p \in \{\text{pints}\})$” if and only if two pints make a quart

Meta-system ("real world")
- Two pints make a quart

**Formal system (logic): $S=$**

“$(q = 2p) \& (q \in \{\text{quarts}\} \& p \in \{\text{pints}\})$”
Schematic of “Turing Machine”

Read/write head with internal operating rules ("states")

Endless tape
An Artificial Neuron

- Each input is multiplied by a synaptic weight ($w_{kj}x_j$)
- The weighted inputs are added up ($\Sigma_k$)
- The final sum is multiplied by a transfer function ($\phi$) to create the output ($y_k$)
A Hopfield Neural Network