Steeple #1. Part 2 of Rationalistic Genius: Gödel’s Completeness Theorem

Selmer Bringsjord

Are Humans Rational?
v of 12/1/16
RPI
Troy NY USA
Some Timeline Points

1906 Brünn, Austria-Hungary
1923 Vienna
Undergrad in seminar by Schlick
1929 Doctoral Dissertation: Proof of Completeness Theorem
1933 Hitler comes to power.
1940 Back to USA, for good.
1978 Princeton NJ USA.
Some Timeline Points

1906 Brünn, Austria-Hungary

1923 Vienna

1929 Doctoral Dissertation: Proof of Completeness Theorem
Undergrad in seminar by Schlick

1933 Hitler comes to power.

1940 Back to USA, for good.

1978 Princeton NJ USA.
R&W’s Axiomatization of the Propositional Calculus

A1  \((\phi \lor \phi) \rightarrow \phi\)
A2  \(\phi \rightarrow (\phi \lor \psi)\)
A3  \((\phi \lor \psi) \rightarrow (\psi \lor \phi)\)
A4  \((\psi \rightarrow \chi) \rightarrow ((\phi \lor \psi) \rightarrow (\phi \lor \chi))\)
R&W’s Axiomatization of the Propositional Calculus

A1 \((\phi \lor \phi) \rightarrow \phi\)
A2 \(\phi \rightarrow (\phi \lor \psi)\)
A3 \((\phi \lor \psi) \rightarrow (\psi \lor \phi)\)
A4 \((\psi \rightarrow \chi) \rightarrow ((\phi \lor \psi) \rightarrow (\phi \lor \chi))\)

All instances of these schemata are true no matter what the input (true or false). (Agreed?) And indeed every single formula in the propositional calculus that is true no matter what the permutation (as shown in a truth table), can be proved (somehow) from these four axioms (using the rules of inference given in AHR? 2016). This, Gödel knew, and could use.
Exercise 1:
Verify that these are true-no-matter what in a truth table; then prove using our rules for the prop. calc.

\[ (\phi \land \psi) \rightarrow (\psi \lor \chi) \]

\[ \phi \rightarrow (\psi \rightarrow \phi) \]
Exercise 1:
Verify that these are true-no-matter what in a truth table; then prove using our rules for the prop. calc.

\[(\phi \land \psi) \rightarrow (\psi \lor \chi)\]
Exercise 1:
Verify that these are true-no-matter what in a truth table; then prove using our rules for the prop. calc.

\[(\phi \land \psi) \rightarrow (\psi \lor \chi)\]

Truth Table showing this formula true no matter what the inputs.
Exercise 1:
Verify that these are true-no-matter what in a truth table; then prove using our rules for the prop. calc.

\[(\phi \land \psi) \to (\psi \lor \chi)\]

Truth Table showing this formula true no matter what the inputs.

Proof:
Exercise 1:
Verify that these are true-no-matter what in a truth table; then prove using our rules for the prop. calc.

\((\phi \land \psi) \rightarrow (\psi \lor \chi)\)

Truth Table showing this formula true no matter what the inputs.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(\phi \land \psi)</td>
<td>Supposition</td>
</tr>
<tr>
<td>2</td>
<td>(\psi)</td>
<td>1, Simplification</td>
</tr>
<tr>
<td>3</td>
<td>(\psi \lor \chi)</td>
<td>2, Addition</td>
</tr>
<tr>
<td>4</td>
<td>((\phi \lor \psi) \rightarrow (\psi \lor \chi))</td>
<td>1–3, Conditional Intro</td>
</tr>
</tbody>
</table>
### TABLE 1 Rules of Inference.

<table>
<thead>
<tr>
<th>Rule of Inference</th>
<th>Tautology</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p )</td>
<td>((p \land (p \rightarrow q)) \rightarrow q)</td>
<td>Modus ponens</td>
</tr>
<tr>
<td>( p \rightarrow q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \therefore q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \neg q )</td>
<td>((\neg q \land (p \rightarrow q)) \rightarrow \neg p)</td>
<td>Modus tollens</td>
</tr>
<tr>
<td>( p \rightarrow q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \therefore \neg p )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \rightarrow q )</td>
<td>((\rightarrow q \land (q \rightarrow r)) \rightarrow (p \rightarrow r))</td>
<td>Hypothetical syllogism</td>
</tr>
<tr>
<td>( q \rightarrow r )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \therefore p \rightarrow r )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \lor q )</td>
<td>(((p \lor q) \land \neg p) \rightarrow q)</td>
<td>Disjunctive syllogism</td>
</tr>
<tr>
<td>( \neg p )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \therefore q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p )</td>
<td>((p \lor q) \rightarrow (p \lor q))</td>
<td>Addition</td>
</tr>
<tr>
<td>( \therefore p \lor q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \land q )</td>
<td>((p \land q) \rightarrow p)</td>
<td>Simplification</td>
</tr>
<tr>
<td>( \therefore p )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \land q )</td>
<td>(((p \land (q)) \rightarrow (p \land q))</td>
<td>Conjunction</td>
</tr>
<tr>
<td>( q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \therefore p \land q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \lor q )</td>
<td>(((p \lor q) \land (\neg p \lor r)) \rightarrow (q \lor r))</td>
<td>Resolution</td>
</tr>
<tr>
<td>( \neg p \lor r )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \therefore q \lor r )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### EXAMPLE 3
State which rule of inference is the basis of the following argument: “It is below freezing. Therefore, it is either below freezing or raining now.”

**Solution:** Let \( p \) be the proposition “It is below freezing now” and \( q \) the proposition “It is raining now.”

- **Premises:**
  - \( p \)
  - \( q \)
- **Conclusion:** \( p \lor q \)

By the rule of **Addition**,

\( p \lor q \rightarrow (p \lor q) \)

Therefore, the rule of **Addition** is the basis of the argument.
**TABLE 1** Rules of Inference.

<table>
<thead>
<tr>
<th>Rule of Inference</th>
<th>Tautology</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p )</td>
<td>((p \land (p \rightarrow q)) \rightarrow q)</td>
<td>Modus ponens</td>
</tr>
<tr>
<td>( p \rightarrow q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \therefore q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \neg q )</td>
<td>((\neg q \land (p \rightarrow q)) \rightarrow \neg p)</td>
<td>Modus tollens</td>
</tr>
<tr>
<td>( p \rightarrow q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \therefore \neg p )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \rightarrow q )</td>
<td>(((p \rightarrow q) \land (q \rightarrow r)) \rightarrow (p \rightarrow r))</td>
<td>Hypothetical syllogism</td>
</tr>
<tr>
<td>( q \rightarrow r )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \therefore p \rightarrow r )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \lor q )</td>
<td>(((p \lor q) \land \neg p) \rightarrow q)</td>
<td>Disjunctive syllogism</td>
</tr>
<tr>
<td>( \neg p )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \therefore q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \rightarrow (p \lor q) )</td>
<td></td>
<td>Addition</td>
</tr>
<tr>
<td>( \therefore p \lor q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \land q )</td>
<td>((p \land q) \rightarrow p)</td>
<td>Simplification</td>
</tr>
<tr>
<td>( \therefore p )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( (p \land (q)) \rightarrow (p \land q))</td>
<td></td>
<td>Conjunction</td>
</tr>
<tr>
<td>( p )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \therefore p \land q )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( (p \lor q) \land (\neg p \lor r)) \rightarrow (q \lor r))</td>
<td>Resolution</td>
<td></td>
</tr>
</tbody>
</table>

**EXAMPLE 3** State which rule of inference is the basis of the following argument: “It is below freezing. Therefore, it is either below freezing or raining now.”

**Solution:** Let \( p \) be the proposition “It is below freezing now” and \( q \) the proposition “It is raining now.” Therefore, the rule of inference used is **Addition**.
But what about the pure predicate calculus and the full predicate calculus?!? Does completeness hold here as well?? Is it true that whatever is true-no-matter-what can also be proved??
But what about the pure predicate calculus and the full predicate calculus?!? Does completeness hold here as well?? Is it true that whatever is true-no-matter-what can also be proved??

Gödel:
You give me a non-contradictory formula, and I’ll show you that there’s a Buzz-Lightyear train trip on which that formula is true — which* gives you an answer of “Yes!”.
But what about the pure predicate calculus and the full predicate calculus?!? Does completeness hold here as well?? Is it true that whatever is true-no-matter-what can also be proved??

Gödel:
You give me a non-contradictory formula, and I’ll show you that there’s a Buzz-Lightyear train trip on which that formula is true — which* gives you an answer of “Yes!”.

*For reasons we won’t go into.
From Last Time: The Grammar of the Pure Predicate Calculus

<table>
<thead>
<tr>
<th>Formula</th>
<th>$\Rightarrow$ AtomicFormula</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mid (\text{Formula Connective Formula})$</td>
</tr>
<tr>
<td></td>
<td>$\mid \neg \text{Formula}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AtomicFormula</th>
<th>$\Rightarrow$ $(\text{Predicate } \text{Term}_1 \ldots \text{Term}_k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mid (\text{Term} = \text{Term})$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Term</th>
<th>$\Rightarrow$ $(\text{Function } \text{Term}_1 \ldots \text{Term}_k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mid \text{Constant}$</td>
</tr>
</tbody>
</table>

| Connective      | $\Rightarrow$ $\land \mid \lor \mid \rightarrow \mid \leftrightarrow$ |

<table>
<thead>
<tr>
<th>Predicate</th>
<th>$\Rightarrow$ $P_1 \mid P_2 \mid P_3 \ldots$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>$\Rightarrow$ $c_1 \mid c_2 \mid c_3 \ldots$</td>
</tr>
<tr>
<td>Function</td>
<td>$\Rightarrow$ $f_1 \mid f_2 \mid f_3 \ldots$</td>
</tr>
</tbody>
</table>
Recall the Examples We Cited

Sally likes Bill.

\((\text{Likes sally bill})\)

Sally likes Bill and Bill likes Sally.

Sally likes Bill only if Bill’s mother is tall.

Matilda is Bill’s super-smart mother.

5 plus 5 equals the number 10.

…

Lexicon

Did you make sure you can simulate a machine that says “Yes that sentence is okay!” whenever it’s conforms to this grammar?
Recall the Examples We Cited

Sally likes Bill.
(Likes sally bill)

Sally likes Bill and Bill likes Sally.
Sally likes Bill’s mother.
Sally likes Bill only if Bill’s mother is tall.
Matilda is Bill’s super-smart mother.
5 plus 5 equals the number 10.

... Did you make sure you can simulate a machine that says “Yes that sentence is okay!” whenever it’s conforms to this grammar?
But Last Time We Noted These Examples

$$\begin{align*}
\text{Formula} & \Rightarrow \quad \text{AtomicFormula} \\
& | \quad (\text{Formula Connective Formula}) \\
& | \quad \neg \text{Formula} \\
\text{AtomicFormula} & \Rightarrow \quad (\text{Predicate Term}_1 \ldots \text{Term}_k) \\
& | \quad (\text{Term} = \text{Term}) \\
\text{Term} & \Rightarrow \quad (\text{Function Term}_1 \ldots \text{Term}_k) \\
& | \quad \text{Constant} \\
\text{Connective} & \Rightarrow \quad \land | \ \lor | \ \to | \ \leftrightarrow \\
\text{Predicate} & \Rightarrow \quad P_1 | P_2 | P_3 \ldots \\
\text{Constant} & \Rightarrow \quad c_1 | c_2 | c_3 \ldots \\
\text{Function} & \Rightarrow \quad f_1 | f_2 | f_3 \ldots
\end{align*}$$
But Last Time We Noted These Examples

If Sally likes Bill then Sally likes Bill.

\[ \text{Formula} \Rightarrow \begin{align*} & \text{AtomicFormula} \\ & (\text{Formula} \text{ Connective} \text{ Formula}) \\ & \neg \text{ Formula} \end{align*} \]

\[ \text{AtomicFormula} \Rightarrow (\text{Predicate} \ \text{Term}_1 \ldots \text{Term}_k) \]

\[ \Rightarrow (\text{Term} = \text{Term}) \]

\[ \text{Term} \Rightarrow (\text{Function} \ \text{Term}_1 \ldots \text{Term}_k) \]

\[ \Rightarrow \text{Constant} \]

\[ \text{Connective} \Rightarrow \land \ | \ \lor \ | \ \to \ | \ \leftrightarrow \]

\[ \text{Predicate} \Rightarrow P_1 \ | \ P_2 \ | \ P_3 \ldots \]

\[ \text{Constant} \Rightarrow c_1 \ | \ c_2 \ | \ c_3 \ldots \]

\[ \text{Function} \Rightarrow f_1 \ | \ f_2 \ | \ f_3 \ldots \]
But Last Time We Noted These Examples

<table>
<thead>
<tr>
<th>Formula</th>
<th>⇒</th>
<th>AtomicFormula</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(Formula Connective Formula)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>¬ Formula</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AtomicFormula</th>
<th>⇒</th>
<th>(Predicate Term₁ ... Termₖ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(Term = Term)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Term</th>
<th>⇒</th>
<th>(Function Term₁ ... Termₖ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Constant</td>
</tr>
</tbody>
</table>

| Connective         | ⇒ | ∧ | ∨ | → | ↔ |

<table>
<thead>
<tr>
<th>Predicate</th>
<th>⇒</th>
<th>P₁</th>
<th>P₂</th>
<th>P₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>⇒</td>
<td>c₁</td>
<td>c₂</td>
<td>c₃</td>
</tr>
<tr>
<td>Function</td>
<td>⇒</td>
<td>f₁</td>
<td>f₂</td>
<td>f₃</td>
</tr>
</tbody>
</table>
But Last Time We Noted These Examples

| Formula | ⇒ | AtomicFormula |
|         |   | (Formula Connective Formula) |
|         |   | ¬ Formula |

| AtomicFormula | ⇒ | (Predicate Term₁ ... Termₖ) |
|               |   | (Term = Term) |

| Term | ⇒ | (Function Term₁ ... Termₖ) |
|      |   | Constant |

| Connective | ⇒ | ∧ | ∨ | → | ↔ |

| Predicate | ⇒ | P₁ | P₂ | P₃ | ... |
| Constant  | ⇒ | c₁ | c₂ | c₃ | ... |
| Function  | ⇒ | f₁ | f₂ | f₃ | ... |
But Last Time We Noted *These* Examples

If Sally likes Bill then Sally likes Bill.

Sally likes Bill’s mother, or not.

Sally likes Bill and Bill likes Jane, only if Bill likes Jane.

Bill’s smart mother is a mother.

<table>
<thead>
<tr>
<th>Formula</th>
<th>$\Rightarrow$ AtomicFormula</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Formula Connective Formula)</td>
</tr>
<tr>
<td></td>
<td>$\neg$ Formula</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AtomicFormula</th>
<th>$\Rightarrow$ (Predicate Term$_1$ ... Term$_k$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Term $=$ Term)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Term</th>
<th>$\Rightarrow$ (Function Term$_1$ ... Term$_k$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Constant</td>
</tr>
</tbody>
</table>

| Connective | $\Rightarrow$ $\land$ | $\lor$ | $\rightarrow$ | $\leftrightarrow$ |

| Predicate | $\Rightarrow$ $P_1$ | $P_2$ | $P_3$ | ... |
|-----------|---------------------|
| Constant  | $\Rightarrow$ $c_1$ | $c_2$ | $c_3$ | ... |
| Function  | $\Rightarrow$ $f_1$ | $f_2$ | $f_3$ | ... |
But Last Time We Noted These Examples

\[
\begin{align*}
\text{Formula} & \Rightarrow \text{AtomicFormula} \\
& \quad \mid (\text{Formula } \text{Connective} \text{ Formula}) \\
& \quad \mid \neg \text{ Formula} \\
\text{AtomicFormula} & \Rightarrow (\text{Predicate } \text{Term}_1 \ldots \text{Term}_k) \\
& \quad \mid (\text{Term} = \text{Term}) \\
\text{Term} & \Rightarrow (\text{Function } \text{Term}_1 \ldots \text{Term}_k) \\
& \quad \mid \text{Constant} \\
\text{Connective} & \Rightarrow \land \mid \lor \mid \rightarrow \mid \leftrightarrow \\
\text{Predicate} & \Rightarrow P_1 \mid P_2 \mid P_3 \ldots \\
\text{Constant} & \Rightarrow c_1 \mid c_2 \mid c_3 \ldots \\
\text{Function} & \Rightarrow f_1 \mid f_2 \mid f_3 \ldots
\end{align*}
\]

If Sally likes Bill then Sally likes Bill.
Sally likes Bill’s mother, or not.
Sally likes Bill and Bill likes Jane, only if Bill likes Jane.
Bill’s smart mother is a mother.

\[\ldots\]
But Last Time We Noted These Examples

\[\begin{align*}
\text{Formula} & \Rightarrow \text{AtomicFormula} \\
& \quad | \quad (\text{Formula Connective Formula}) \\
& \quad | \quad \neg \text{Formula}
\end{align*}\]

\[\begin{align*}
\text{AtomicFormula} & \Rightarrow (\text{Predicate Term}_1 \ldots \text{Term}_k) \\
& \quad | \quad (\text{Term} = \text{Term})
\end{align*}\]

\[\begin{align*}
\text{Term} & \Rightarrow (\text{Function Term}_1 \ldots \text{Term}_k) \\
& \quad \quad | \quad \text{Constant}
\end{align*}\]

\[\begin{align*}
\text{Connective} & \Rightarrow \land \quad \lor \quad \rightarrow \quad \leftrightarrow
\end{align*}\]

\[\begin{align*}
\text{Predicate} & \Rightarrow P_1 \mid P_2 \mid P_3 \ldots \\
\text{Constant} & \Rightarrow c_1 \mid c_2 \mid c_3 \ldots \\
\text{Function} & \Rightarrow f_1 \mid f_2 \mid f_3 \ldots
\end{align*}\]

If Sally likes Bill then Sally likes Bill.

Sally likes Bill’s mother, or not.

Sally likes Bill and Bill likes Jane, only if Bill likes Jane.

Bill’s smart mother is a mother.

…

These are all true, yes; but can they be proved?!
Add the Final Addition/Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus *simpliciter*
Add the Final Addition/Deeper Challenge:
Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus *simpliciter*

there exists at least one thing $x$ such that …
Add the Final Addition/Deeper Challenge:
Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus \textit{simpliciter}

there exists at least one thing $x$ such that …

for all $x$, it’s the case that …
Add the Final Addition/Deeper Challenge:
Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus *simpliciter*

$$\exists x \ldots \text{there exists at least one thing } x \text{ such that } \ldots$$

for all $x$, it’s the case that $\ldots$
Add the Final Addition/Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus \textit{simpliciter}

\[ \exists x \ldots \text{there exists at least one thing } x \text{ such that } \ldots \]

\[ \forall x \ldots \text{for all } x, \text{ it’s the case that } \ldots \]
Add the Final Addition/Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus \textit{simpliciter}

\forall x \ldots \text{ for all } x, \text{ it's the case that } \ldots

\exists x \ldots \text{ there exists at least one thing } x \text{ such that } \ldots

\forall \epsilon (\epsilon > 0 \rightarrow \exists \delta (\delta > 0 \land \forall x (d(x, a) < \delta \rightarrow d(f(x), b) < \epsilon)))
Add the Final Addition/Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus *simpliciter*

\[ \exists x \quad \text{there exists at least one thing } x \text{ such that } \ldots \]

\[ \forall x \quad \text{for all } x, \text{ it’s the case that } \ldots \]
Add the Final Addition/Deeper Challenge:
Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus *simpliciter*

\[ \exists x \ldots \text{there exists at least one thing } x \text{ such that } \ldots \]

\[ \forall x \ldots \text{for all } x, \text{ it’s the case that } \ldots \]

Every natural number is greater than or equal to zero.
Add the Final Addition/Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus *simpliciter*

$\exists x \ldots$ there exists at least one thing $x$ such that ...  

$\forall x \ldots$ for all $x$, it’s the case that ...  

Every natural number is greater than or equal to zero.  

$\forall x(x \geq 0)$
Add the Final Addition/Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus *simpliciter*

\[ \exists x \quad \text{there exists at least one thing } x \text{ such that } \ldots \]

\[ \forall x \quad \text{for all } x, \text{it's the case that } \ldots \]

Every natural number is greater than or equal to zero.

\[ \forall x (x \geq 0) \]

There's a positive integer greater than any positive integer.
Add the Final Addition/Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus *simpliciter*

\[ \exists x \ldots \text{there exists at least one thing } x \text{ such that } \ldots \]
\[ \forall x \ldots \text{for all } x, \text{it's the case that } \ldots \]

Every natural number is greater than or equal to zero.

\[ \forall x (x \geq 0) \]

There's a positive integer greater than any positive integer.

\[ \exists x \forall y (y < x) \]
Add the Final Addition/Deeper Challenge:
Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus *simpliciter*

\[ \exists x \ldots \text{ there exists at least one thing } x \text{ such that } \ldots \]

\[ \forall x \ldots \text{ for all } x, \text{ it’s the case that } \ldots \]

Every natural number is greater than or equal to zero.

\[ \forall x (x \geq 0) \]

\[ \exists x \forall y (y < x) \]
Add the Final Addition/Deeper Challenge:
Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus *simpliciter*

\[\exists x \ldots \text{there exists at least one thing } x \text{ such that } \ldots\]

\[\forall x \ldots \text{for all } x, \text{it’s the case that } \ldots\]

Every natural number is greater than or equal to zero.
\[\forall x (x \geq 0)\]
Add the Final Addition/Deeper Challenge:
Add Two Quantifiers to the Pure Predicate Calculus,
Which Yields the Predicate Calculus *simpliciter*

\[\exists x \ldots \text{there exists at least one thing } x \text{ such that } \ldots\]

\[\forall x \ldots \text{for all } x, \text{ it's the case that } \ldots\]

Every natural number is greater than or equal to zero.

\[\forall x (x \geq 0)\]

Every positive integer \(x\) is less-than-or-equal-to a positive integer \(y\).
Add the Final Addition/Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields the Predicate Calculus \textit{simpliciter}

\[ \exists x \ldots \text{there exists at least one thing } x \text{ such that } \ldots \]

\[ \forall x \ldots \text{for all } x, \text{ it’s the case that } \ldots \]

Every natural number is greater than or equal to zero.
\[ \forall x (x \geq 0) \]

Every positive integer \( x \) is less-than-or-equal-to a positive integer \( y \).
\[ \forall x \exists y (x \leq y) \quad \forall x \exists y (\leq (x, y)) \]
For every positive integer \( x \) is less-than-or-equal-to a positive integer \( y \).

\[
\forall x \exists y (\leq (x, y))
\]
For every positive integer \( x \) is less-than-or-equal-to a positive integer \( y \).

\[
\forall x \exists y (\leq (x, y))
\]

Gödel:

“You give me a non-contradictory formula, and I’ll show you that there’s a Buzz-Lightyear train trip on which that formula is true — which gives you an answer of ‘Yes!’

Equivalently, the predicate calculus is complete: every formula in it that must-be-true, can in fact be proved!”
For every positive integer $x$ is less-than-or-equal-to a positive integer $y$.

\[ \forall x \exists y (\leq (x, y)) \]

First, I’ll break this down into a list of conjunctions and drop the quantifiers.
For every positive integer $x$ is less-than-or-equal-to a positive integer $y$.
\[ \forall x \exists y (\leq (x, y)) \]

First, I’ll break this down into a list of conjunctions and drop the quantifiers.

\[
\begin{align*}
\leq x_1 x_2 \\
\leq x_1 x_2 \land \leq x_2 x_3 \\
\leq x_1 x_2 \land \leq x_2 x_3 \land \leq x_3 x_4 \\
\leq x_1 x_2 \land \leq x_2 x_3 \land \leq x_3 x_4 \land \leq x_4 x_5 \\
\vdots \\
\leq x_1 x_2 \ldots \leq x_{99} x_{100}
\end{align*}
\]
For every positive integer $x$ is less-than-or-equal-to a positive integer $y$.

\[ \forall x \exists y \left( \leq (x, y) \right) \]

First, I’ll break this down into a list of conjunctions and drop the quantifiers.

\[
\begin{align*}
\leq x_1 & x_2 \\
\leq x_1 x_2 & \land \leq x_2 x_3 \\
\leq x_1 x_2 & \land \leq x_2 x_3 & \land \leq x_3 x_4 \\
\leq x_1 x_2 & \land \leq x_2 x_3 & \land \leq x_3 x_4 & \land \leq x_4 x_5 \\
\vdots \\
\leq x_1 & x_2 \ldots \leq x_{99} x_{100}
\end{align*}
\]

Next, I’ll take you on a Buzz-Lightyear train trip on which this entire list is true …
Now, recall König’s Lemma
Toward König’s Lemma as Train Travel
“To infinity and beyond!”
König’s Lemma (train-travel version)

In a one-way train-travel map with finitely many options leading from each station, if there are partial paths forward of every finite length, there is an infinite path (= a path “to infinity”).
Exercise 2:
Is there an algorithm for traveling this way?
Exercise 2:
Is there an algorithm for traveling this way?

No. This strategy for travel is beyond the reach of standard computation.
Exercise 2:
Is there an algorithm for traveling this way?

No. This strategy for travel is beyond the reach of standard computation.

(To anticipate our final class & Steeple #3: Does it not then follow, assuming that humans can find and “use” a provably correct strategy for this travel, that humans can’t be fundamentally computing machines?)
Proving the Lemma
(that there is an infinite branch)

Proof: We are seeking to prove that there is an infinite path (= that you can keep going forward forever = that the number of your stops forward are the size of $\mathbb{Z}^+$).

To begin, assume the antecedent of the theorem (i.e. that, (1), there are finitely many options leading from each station, and that, (2), in the map there are partial paths forward of every finite size).

Now, you are standing at Penn Station ($S_1$), facing $k$ options. At least one of these options must lead to partial paths of arbitrary size (the size of any $m$ in $\mathbb{Z}^+$). (Sub-Proof: Suppose otherwise for indirect proof. Then there is some positive integer $n$ that places a ceiling on the size of partial paths that can be reached. But this violates (2) — contradiction.) Proceed to choose one of these options that lead to partial paths of arbitrary size. You are now standing at a new station ($S_2$), one stop after Penn Station. At least one of these options must lead to partial parts of arbitrary size (the size of any $m$ in $\mathbb{Z}^+$). (Sub-Proof: Suppose otherwise for indirect proof …)

Since you can iterate this forever, you’ll be on an infinite trip to infinity! Buzz will be happy.
Gödel’s Buzz-Lightyear Branch

Every single $x_m x_{m+1}$ is true on $B_\infty$
for $\leq x_m x_{m+1}$

$B_\infty :=$
Gödel's infinite branch

 Lexicographic (essentially dictionary) ordering
§1. Introduction. This paper deals with aspects of my doctoral dissertation\(^1\) which contributed to the early development of model theory. What was of use to later workers was less the results of my thesis, than the method by which I proved the completeness of first-order logic—a result established by Kurt Gödel in his doctoral thesis 18 years before.\(^2\)

The ideas that fed my discovery of this proof were mostly those I found in the teachings and writings of Alonzo Church. This may seem curious, as his work in logic, and his teaching, gave great emphasis to the constructive character of mathematical logic, while the model theory to which I contributed is filled with theorems about very large classes of mathematical structures, whose proofs often bypass constructive methods.

Another curious thing about my discovery of a new proof of Gödel’s completeness theorem, is that it arrived in the midst of my efforts to prove an entirely different result. Such “accidental” discoveries arise in many parts of scientific work. Perhaps there are regularities in the conditions under which such “accidents” occur which would interest some historians, so I shall try to describe in some detail the accident which befell me.
§1. Introduction. This paper deals with aspects of my doctoral dissertation¹ which contributed to the early development of model theory. What was of use to later workers was less the results of my thesis, than the method by which I proved the completeness of first-order logic—a result established by Kurt Gödel in his doctoral thesis 18 years before.²

The ideas that fed my discovery of this proof were mostly those I found in the teachings and writings of Alonzo Church. This may seem curious, as his work in logic, and his teaching, gave great emphasis to the constructive character of mathematical logic, while the model theory to which I contributed is filled with theorems about very large classes of mathematical structures, whose proofs often by-pass constructive methods.

Another curious thing about my discovery of a new proof of Gödel’s completeness theorem, is that it arrived in the midst of my efforts to prove an entirely different result. Such “accidental” discoveries arise in many parts of scientific work. Perhaps there are regularities in the conditions under which such “accidents” occur which would interest some historians, so I shall try to describe in some detail the accident which befell me.

---

Received November 17, 1995, and in revised form, January 4, 1996.
THE DISCOVERY OF MY COMPLETENESS PROOFS

LEON HENKIN

Dedicated to my teacher, Alonzo Church, in his 91st year.

§1. Introduction. This paper deals with aspects of my doctoral dissertation\(^1\) which contributed to the early development of model theory. What was of use to later workers was less the results of my thesis, than the method by which I proved the completeness of first-order logic—a result established by Kurt Gödel in his doctoral thesis 18 years before\(^2\).

The ideas that fed my discovery of this proof were mostly those I found in the teachings and writings of Alonzo Church. This may seem curious, as his work in logic, and his teaching, gave great emphasis to the constructive character of mathematical logic, while the model theory to which I contributed is filled with theorems about very large classes of mathematical structures, whose proofs often by-pass constructive methods.

Another curious thing about my discovery of a new proof of Gödel’s completeness theorem, is that it arrived in the midst of my efforts to prove an entirely different result. Such “accidental” discoveries arise in many parts of scientific work. Perhaps there are regularities in the conditions under which such “accidents” occur which would interest some historians, so I shall try to describe in some detail the accident which befell me.

Received November 17, 1995, and in revised form, January 4, 1996.
THE DISCOVERY OF MY COMPLETENESS PROOFS

LEON HENKIN

Dedicated to my teacher, Alonzo Church, in his 91st year.

§1. Introduction. This paper deals with aspects of my doctoral dissertation ¹ which contributed to the early development of model theory. What was of use to later workers was less the results of my thesis, than the method by which I proved the completeness of first-order logic—a result established by Kurt Gödel in his doctoral thesis 18 years before ².

The ideas that fed my discovery of this proof were mostly those I found in the teachings and writings of Alonzo Church. This may seem curious, as his work in logic, and his teaching, gave great emphasis to the constructive character of mathematical logic, while the model theory to which I contributed is filled with theorems about very large classes of mathematical structures, whose proofs often by-pass constructive methods.

Another curious thing about my discovery of a new proof of Gödel’s completeness theorem, is that it arrived in the midst of my efforts to prove an entirely different result. Such “accidental” discoveries arise in many parts of scientific work. Perhaps there are regularities in the conditions under which such “accidents” occur which would interest some historians, so I shall try to describe in some detail the accident which befell me.

Received November 17, 1995, and in revised form, January 4, 1996.
THE DISCOVERY OF MY COMPLETENESS PROOFS

LEON HENKIN

Dedicated to my teacher, Alonzo Church, in his 91st year.

§1. Introduction. This paper deals with aspects of my doctoral dissertation\(^1\) which contributed to the early development of model theory. What was of use to later workers was less the results of my thesis, than the method by which I proved the completeness of first-order logic—a result established by Kurt Gödel in his doctoral thesis 18 years before.\(^2\)

The ideas that fed my discovery of this proof were mostly those I found in the teachings and writings of Alonzo Church. This may seem curious, as his work in logic, and his teaching, gave great emphasis to the constructive character of mathematical logic, while the model theory to which I contributed is filled with theorems about very large classes of mathematical structures, whose proofs often by-pass constructive methods.

Another curious thing about my discovery of a new proof of Gödel’s completeness theorem, is that it arrived in the midst of my efforts to prove an entirely different result. Such “accidental” discoveries arise in many parts of scientific work. Perhaps there are regularities in the conditions under which such “accidents” occur which would interest some historians, so I shall try to describe in some detail the accident which befell me.

Received November 17, 1995, and in revised form, January 4, 1996.
THE DISCOVERY OF MY COMPLETENESS PROOFS

LEON HENKIN

Dedicated to my teacher, Alonzo Church, in his 91st year.

§1. Introduction. This paper deals with aspects of my doctoral dissertation which contributed to the early development of model theory. What was of use to later workers was less the results of my thesis, than the method by which I proved the completeness of first-order logic—a result established by Kurt Gödel in his doctoral thesis 18 years before.

The ideas that fed my discovery of this proof were mostly those I found in the teachings and writings of Alonzo Church. This may seem curious, as his work in logic, and his teaching, gave great emphasis to the constructive character of mathematical logic, while the model theory to which I contributed is filled with theorems about very large classes of mathematical structures, whose proofs often by-pass constructive methods.

Another curious thing about my discovery of a new proof of Gödel's completeness theorem, is that it arrived in the midst of my efforts to prove an entirely different result. Such “accidental” discoveries arise in many parts of scientific work. Perhaps there are regularities in the conditions under which such “accidents” occur which would interest some historians, so I shall try to describe in some detail the accident which befell me.

Received November 17, 1995, and in revised form, January 4, 1996.
Next Time
(Final Content-Delivery Class Mtg)

• Steeple #2: The Immortal Incompleteness Result
• Steeple #3: Could a machine ever be this rational and creative?
slutten
Appendices ...
So what would be a specific g*? A truth of arithmetic that you can’t move from the axioms of arithmetic?!?
So what would be a specific g*? A truth of arithmetic that you can’t move from the axioms of arithmetic?!?

Here you go:
So what would be a specific $g^*$? A truth of arithmetic that you can’t move from the axioms of arithmetic?!?

Here you go:

That the Goodstein Sequence eventually reaches zero!
Goodstein Sequence;
Goodstein’s Theorem ...
Pure base $n$ representation of a number $r$

- Represent $r$ as only sum of powers of $n$ in which the exponents are also powers of $n$ etc

$$266 = 2^{2^{(2^0 + 2^0)}} + 2^{2^0} + 2^0$$
Grow Function

\[ \text{Grow}_k(n) : \]

1. Take the pure base \( k \) representation of \( n \)
2. Replace all \( k \) by \( k + 1 \). Compute the number obtained.
3. Subtract one from the number
Example of Grow

\[ \text{Grow}_2(19) \]

\[ 19 = 2^22^0 + 2^2^0 + 2^0 \]

\[ 3^33^3^0 + 3^3^0 + 3^0 \]

\[ 3^{3^3^3^0} + 3^{3^0} + 3^0 - 1 \]

7625597484990
Goodstein Sequence

- For any natural number $m$

  \[
  m \\
  \text{Grow}_2(m) \\
  \text{Grow}_3(\text{Grow}_2(m)) \\
  \text{Grow}_4(\text{Grow}_3(\text{Grow}_2(m))), \\
  \ldots
  \]
Sample Values
Sample Values
Sample Values
<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>m</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Sample Values

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

- $m$
## Sample Values

<table>
<thead>
<tr>
<th>$m$</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>26</td>
<td>41</td>
<td>60</td>
<td>83</td>
</tr>
</tbody>
</table>

...
### Sample Values

<table>
<thead>
<tr>
<th>m</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>0</th>
<th></th>
<th></th>
<th></th>
<th>11327 (96th term)</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>26</td>
<td>41</td>
<td>60</td>
<td>83</td>
<td>109</td>
<td>139</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>~10^{13}</td>
<td>~10^{155}</td>
<td>~10^{2185}</td>
<td>~10^{36306}</td>
<td>10^{695975}</td>
<td>10^{15151337}</td>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Yet, The Theorems!!
Yet, The Theorems!!

**Theorem 1 (Goodstein’s Theorem).** For all natural numbers, the Goodstein sequence reaches zero after a finite number of steps.

**Theorem 2 (Unprovability of Goodstein’s Theorem).** Goodstein’s theorem is not provable in Peano Arithmetic (PA) (or any equivalent theory of arithmetic).
Yet, The Theorems!!

Theorem 1 (Goodstein’s Theorem). For all natural numbers, the Goodstein sequence reaches zero after a finite number of steps.

Theorem 2 (Unprovability of Goodstein’s Theorem). Goodstein’s theorem is not provable in Peano Arithmetic (PA) (or any equivalent theory of arithmetic).

So, Gödel was right, empirically!

We have in GT a truth of elementary arithmetic that we can’t prove from elementary arithmetic!
Could a computing machine get this?? ...
Small Steps Toward Hypercomputation via Infinitary Machine Proof Verification and Proof Generation

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord
Department of Computer Science
Department of Cognitive Science
Rensselaer AI & Reasoning Laboratory
govian@rpi.edu • licaj@rpi.edu • selmer@rpi.edu
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180 USA

Abstract. After setting a context based on two general points (that humans appear to reason in infinitary fashion, and that actual hypercomputers aren’t currently available to directly model and replicate such infinitary reasoning), we set a humble engineering goal of taking initial steps toward a computing machine that can reason in infinitary fashion. The initial steps consist in our outline of automated proof-verification and proof-discovery techniques for theorems independent of PA that seem to require an understanding and use of infinitary concepts. We specifically focus on proof-discovery techniques that make use of a marriage of analogical and deductive reasoning (which we call analogico-deductive reasoning).

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

Bringsjord has repeatedly pointed out the obvious fact that the behavior of formal scientists, taken at face value, involves various infinitary structures and reasoning. (We say “at face value” to simply indicate we don’t presuppose some view that denies the reality of infinite entities routinely involved in the formal sciences.) For example, in (Bringsjord & van Heijenoort 2003), Bringsjord himself operates as such a scientist in presenting an infinitary paradox which so far knowledge has yet to be solved. And he has argued that apparently infinitary behavior constitutes a grave challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkoudas 2006, Bringsjord & Zemans 2003). More generally, Bringsjord conjectures that every human-produced proof of a theorem independent of Peano Arithmetic (PA) will make use of infinitary structures and reasoning, when those structures are taken at face value. We have ourselves designed logic-computational logics for handling infinitary reasoning (e.g., see the treatment of the infinitized wise-man puzzle: Arkoudas & Bringsjord 2005), but this work simply falls back on the human ability to carry out induction on the natural numbers. It doesn’t dissect and explain this ability. Finally, it must be admitted by all that there is simply no systematic, comprehensive model or framework anywhere in the formal/computational approach to understanding human knowledge and intelligence that provides a theory about how humans are able to engage with infinitary structures. This is revealed perhaps most clearly when one studies the fruit produced by the part of formal AI devoted to producing discovery systems: such fruit is embarrassingly infinitary (e.g., see Shilliday 2009).

Given this context, we are interested in exploring how one might give a machine the ability to reason in infinitary fashion. We are not saying that we in fact have figured out how to give such ability to a computing machine. Our objective here is much more humble and limited: it is to push forward in the attempt to engineer a computing machine that has the ability to reason in infinitary fashion. Ultimately, if such an attempt is to succeed, the computing machine in question will presumably be capable of outright hypercomputation. But the fact is that from an engineering perspective, we don’t know how to create and harness a hypercomputer. So what we must first do, as explained in (Bringsjord & Zemans 2003), is pursue engineering that initiates the attempt to engineer a hypercomputer, and takes the first few steps. In the present paper, the engineering is aimed specifically at giving a computing machine the ability to, in a limited but well-defined sense, reason in infinitary fashion. Even more specifically, our engineering is aimed at building a machine capable of at least providing a strong case for a result which, in the human sphere, has hitherto required use of infinitary techniques.

1 A weaker conjecture along the same line has been ventured by Isaacson, and is elegantly discussed by Smith (2007).
Needs Understanding of Ordinal Numbers …
Needs Understanding of Ordinal Numbers …

\[ 19_1 = 2^{2^{2^0}} + 2^{2^0} + 2^0 < \omega^{\omega^0} + \omega^0 + \omega^0 \]
Needs Understanding of Ordinal Numbers …

\[ 19_1 = 2^{2^{2^0}} + 2^0 + 2^0 < \omega^{\omega^{\omega^0}} + \omega^0 + \omega^0 \]

\[ 19_2 = 3^{3^{3^0}} + 3^0 + 3^0 - 1 < \omega^{\omega^{\omega^0}} + \omega^0 + \omega^0 - 1 \]
Needs Understanding of Ordinal Numbers …

$$19_1 = 2^{2^{2^0}} + 2^0 + 2^0 < \omega^{\omega^{\omega^0}} + \omega^0 + \omega^0$$

$$19_2 = 3^{3^{3^0}} + 3^0 + 3^0 - 1 < \omega^{\omega^{\omega^0}} + \omega^0 + \omega^0 - 1$$

$$19_3 = 4^{4^{4^0}} + 4^0 - 1 < \omega^{\omega^{\omega^0}} + \omega^0 - 1$$
Needs Understanding of Ordinal Numbers …

\[ 19_1 = 2^{2^{2^0}} + 2^0 + 2^0 < \omega^{\omega^{\omega^0}} + \omega^0 + \omega^0 \]

\[ 19_2 = 3^{3^{3^0}} + 3^0 + 3^0 - 1 < \omega^{\omega^{\omega^0}} + \omega^0 + \omega^0 - 1 \]

\[ 19_3 = 4^{4^{4^0}} + 4^0 - 1 < \omega^{\omega^{\omega^0}} + \omega^0 - 1 \]

\[ 19_4 = 5^{5^{5^0}} + 5^0 + 5^0 - 1 < \omega^{\omega^{\omega^0}} + \omega^0 + \omega^0 - 1 \]
Needs Understanding of Ordinal Numbers ...

\[ 19_1 = 2^{2^{2^2}} + 2^0 + 2^0 < \omega^\omega^\omega^0 + \omega^0 + \omega^0 \]

\[ 19_2 = 3^{3^{3^3}} + 3^0 + 3^0 - 1 < \omega^\omega^\omega^0 + \omega^0 + \omega^0 - 1 \]

\[ 19_3 = 4^{4^{4^4}} + 4^0 - 1 < \omega^\omega^\omega^0 + \omega^0 - 1 \]

\[ 19_4 = 5^{5^{5^5}} + 5^0 + 5^0 - 1 < \omega^\omega^\omega^0 + \omega^0 + \omega^0 - 1 \]

\[ 19_5 = 6^{6^{6^6}} + 6^0 < \omega^\omega^\omega^0 + \omega^0 \]
Needs Understanding of Ordinal Numbers …

\[
19_1 = 2^{2^{2^{2^0}}} + 2^0 + 2^0 < \omega^{\omega^{\omega^0}} + \omega^0 + \omega^0 \\
19_2 = 3^{3^{3^0}} + 3^0 + 3^0 - 1 < \omega^{\omega^{\omega^0}} + \omega^0 + \omega^0 - 1 \\
19_3 = 4^{4^{4^0}} + 4^0 - 1 < \omega^{\omega^{\omega^0}} + \omega^0 - 1 \\
19_4 = 5^{5^{5^0}} + 5^0 + 5^0 - 1 < \omega^{\omega^{\omega^0}} + \omega^0 + \omega^0 - 1 \\
19_5 = 6^{6^{6^0}} + 6^0 < \omega^{\omega^{\omega^0}} + \omega^0 \\
\vdots
\]
Needs Understanding of Ordinal Numbers ...

\[
19_1 = 2^{2^{2^{2^0}}} + 2^0 + 2^0 < \omega^{\omega^0} + \omega^0 + \omega^0
\]

\[
19_2 = 3^{3^{3^{3^0}}} + 3^0 + 3^0 - 1 < \omega^{\omega^0} + \omega^0 + \omega^0 - 1
\]

\[
19_3 = 4^{4^{4^{4^0}}} + 4^0 - 1 < \omega^{\omega^0} + \omega^0 - 1
\]

\[
19_4 = 5^{5^{5^{5^0}}} + 5^0 + 5^0 - 1 < \omega^{\omega^0} + \omega^0 + \omega^0 - 1
\]

\[
19_5 = 6^{6^{6^{6^0}}} + 6^0 < \omega^{\omega^0} + \omega^0
\]

...
Needs Understanding of Ordinal Numbers …

\[ 19_1 = 2^{2^{2^0}} + 2^0 + 2^0 < \omega_0 + \omega_0 + \omega_0 \]

\[ 19_2 = 3^{3^{3^0}} + 3^0 + 3^0 - 1 < \omega_0 + \omega_0 + \omega_0 - 1 \]

\[ 19_3 = 4^{4^{4^0}} + 4^0 - 1 < \omega_0 + \omega_0 - 1 \]

\[ 19_4 = 5^{5^{5^0}} + 5^0 + 5^0 - 1 < \omega_0 + \omega_0 + \omega_0 - 1 \]

\[ 19_5 = 6^{6^{6^0}} + 6^0 < \omega_0 + \omega_0 \]

… strictly decreasing
\[
19_1 = 2^{2^{2^0}} + 2^0 + 2^0 < \omega^{\omega^0} + \omega^0 + \omega^0
\]
\[
19_2 = 3^{3^{3^0}} + 3^0 + 3^0 - 1 < \omega^{\omega^{\omega^0}} + \omega^0 + \omega^0 - 1
\]
\[
19_3 = 4^{4^{4^0}} + 4^0 - 1 < \omega^{\omega^{\omega^0}} + \omega^0 - 1
\]
\[
19_4 = 5^{5^{5^0}} + 5^0 + 5^0 - 1 < \omega^{\omega^{\omega^0}} + \omega^0 + \omega^0 - 1
\]
\[
19_5 = 6^{6^{6^0}} + 6^0 < \omega^{\omega^{\omega^0}} + \omega^0
\]
\[
\vdots
\]
strictly decreasing
Ordinal Numbers ...
Yet, Conjecture (C) 
(see “Isaacson’s Conjecture”)
Yet, Conjecture (C)  
(see “Isaacson’s Conjecture”)

In order to produce a rationally compelling proof of any true sentence $S$ formed from the symbol set of the language of arithmetic, but independent of PA, it’s necessary to deploy concepts and structures of an irreducibly infinitary nature.
Yet, Conjecture (C)
(see “Isaacson’s Conjecture”)

In order to produce a rationally compelling proof of any true sentence $S$ formed from the symbol set of the language of arithmetic, but independent of PA, it’s necessary to deploy concepts and structures of an irreducibly infinitary nature.

If this is right, and computing machines can’t use irreducibly infinitary techniques, they’re in trouble — or: there won’t be a Singularity.