“The Wager,” Overview of the Book, and Gödel’s Completeness Theorem

Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management & Technology
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Formal Logic (With AI)
4/6/2026

Rensselaer AI and Reasoning Lab

“The Wager,” Overview of the Book, and Gödel’s Completeness Theorem

Selmer Bringsjord

Note: This is a version of coverage of Gödel’s Completeness Theorem designed for those who’ve had at least one standard/ standard-paced university-level course in formal logic.

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management & Technology
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Formal Logic (With AI)
4/6/2026

Rensselaer AI and Reasoning Lab

HyperGrader releases …

AI vs. PI


PERSON OF THE YEAR
THE ARCHITECTS OF AI
TIME
$9.99US $10.99CAN
time.com

AI vs. PI

AI has brought forth a dizzying acronym kaleidoscope:

AI, Strong AI, Weak AI, Strong Weak AI, AGI, HI, HLAI, ASI, …
and SAI (Y. LeCun), and PI, HPI, API, SPI.

AI vs. PI

AI has brought forth a dizzying acronym kaleidoscope:

Selmer

AI, Strong AI, Weak AI, Strong Weak AI, AGI, HI, HLAI, ASI, …
and SAI (Y. LeCun), and PI, HPI, API, SPI.

Selmer

Background Context …

Could Artificial Intelligence Ever … Match Personal Intelligence?

Gödelian Essays on Minds vs. Machines

by Selmer Bringsjord

- Introduction ("The Wager")
- Brief Preliminaries (elementary discrete math, incl. ZOL, FOL)
- The Completeness Theorem
- The First Incompleteness Theorem
- The Second Incompleteness Theorem
- The Speedup Theorem
- The Continuum-Hypothesis Theorem
- The Time-Travel Theorem
- Gödel's "God Theorem"
- Could a Machine Match Gödel's Genius?

Could Artificial Intelligence Ever … Match Personal Intelligence?

Gödelian Essays on Minds vs. Machines

by Selmer Bringsjord

- Introduction ("The Wager")
- Brief Preliminaries (elementary discrete math, incl. ZOL, FOL)
- The Completeness Theorem
- The First Incompleteness Theorem
- The Second Incompleteness Theorem
- The Speedup Theorem
- The Continuum-Hypothesis Theorem
- The Time-Travel Theorem
- Gödel's "God Theorem"
- Could a Machine Match Gödel's Genius?

Could Artificial Intelligence Ever . . .
Match Personal Intelligence?
Gödelian Essays on Minds vs. Machines

by Selmer Bringsjord

- Introduction (“The Wager”) √
- Brief Preliminaries (elementary discrete math, incl. ZOL, FOL)
- The Completeness Theorem
- The First Incompleteness Theorem
- The Second Incompleteness Theorem
- The Speedup Theorem
- The Continuum-Hypothesis Theorem
- The Time-Travel Theorem
- Gödel’s “God Theorem”
- Could a Machine Match Gödel’s Genius?

Could Artificial Intelligence Ever …
Match Personal Intelligence?
Gödelian Essays on Minds vs. Machines
by Selmer Bringsjord

- Introduction ("The Wager")
- Brief Preliminaries (elementary discrete math, incl. ZOL, FOL)
- The Completeness Theorem
- The First Incompleteness Theorem
- The Second Incompleteness Theorem
- The Speedup Theorem
- The Continuum-Hypothesis Theorem
- The Time-Travel Theorem
- Gödel's "God Theorem"
- Could a Machine Match Gödel's Genius?

Some Timeline Points

Some Timeline Points

1906 Brünn, Austria-Hungary

Some Timeline Points

1923 Vienna

1906 Brünn, Austria-Hungary

Some Timeline Points

Undergrad in seminar by Schlick

1923 Vienna

1906 Brünn, Austria-Hungary

Some Timeline Points

1929 Doctoral Dissertation: Proof of Completeness Theorem

Undergrad in seminar by Schlick

1923 Vienna

1906 Brünn, Austria-Hungary

Some Timeline Points

1930 Announces (First) *In*completeness Theorem
1929 Doctoral Dissertation: Proof of Completeness Theorem
Undergrad in seminar by Schlick

1923 Vienna

1906 Brünn, Austria-Hungary

Some Timeline Points

1933 Hitler comes to power.

1930 Announces (First) *In*completeness Theorem

1929 Doctoral Dissertation: Proof of Completeness Theorem

Undergrad in seminar by Schlick

1923 Vienna

1906 Brünn, Austria-Hungary

Some Timeline Points

1940 Back to USA, for good.

1933 Hitler comes to power.

1930 Announces (First) *In*completeness Theorem

1929 Doctoral Dissertation: Proof of Completeness Theorem

Undergrad in seminar by Schlick

1923 Vienna

1906 Brünn, Austria-Hungary

Some Timeline Points

1940 Back to USA, for good.

1933 Hitler comes to power.

1930 Announces (First) *In*completeness Theorem

1929 Doctoral Dissertation: Proof of Completeness Theorem

Undergrad in seminar by Schlick

1923 Vienna

1906 Brünn, Austria-Hungary

Some Timeline Points

1978 Princeton NJ USA.

1940 Back to USA, for good.

1933 Hitler comes to power.

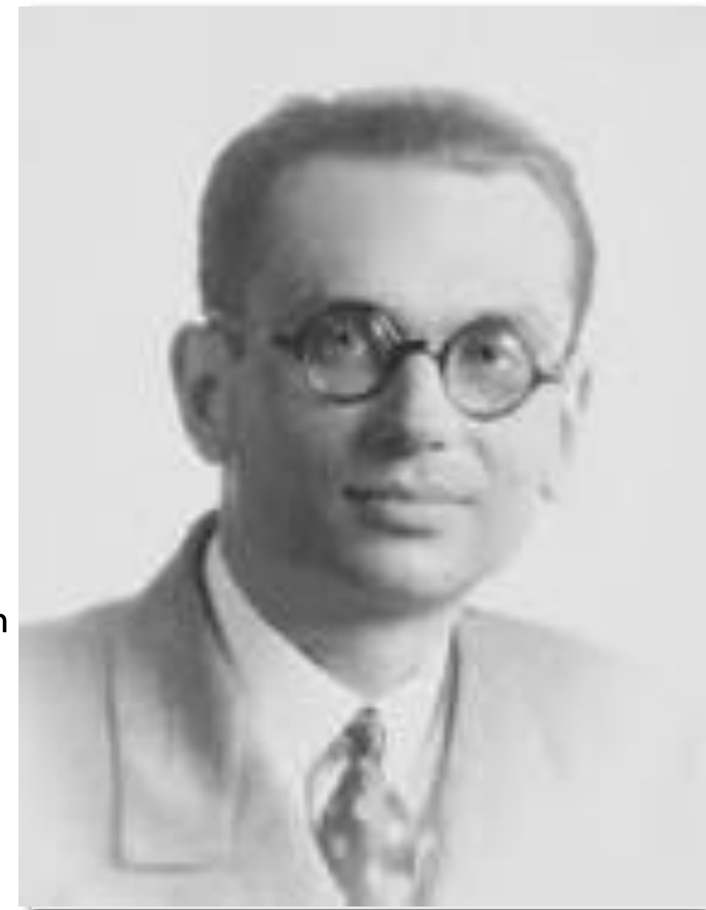
1930 Announces (First) *In*completeness Theorem

1929 Doctoral Dissertation: Proof of Completeness Theorem

Undergrad in seminar by Schlick

1923 Vienna

1906 Brünn, Austria-Hungary



Some Timeline Points

1978 Princeton NJ USA.

1940 Back to USA, for good.

1936 Schlick murdered; Austria annexed; advisor dies

1933 Hitler comes to power.

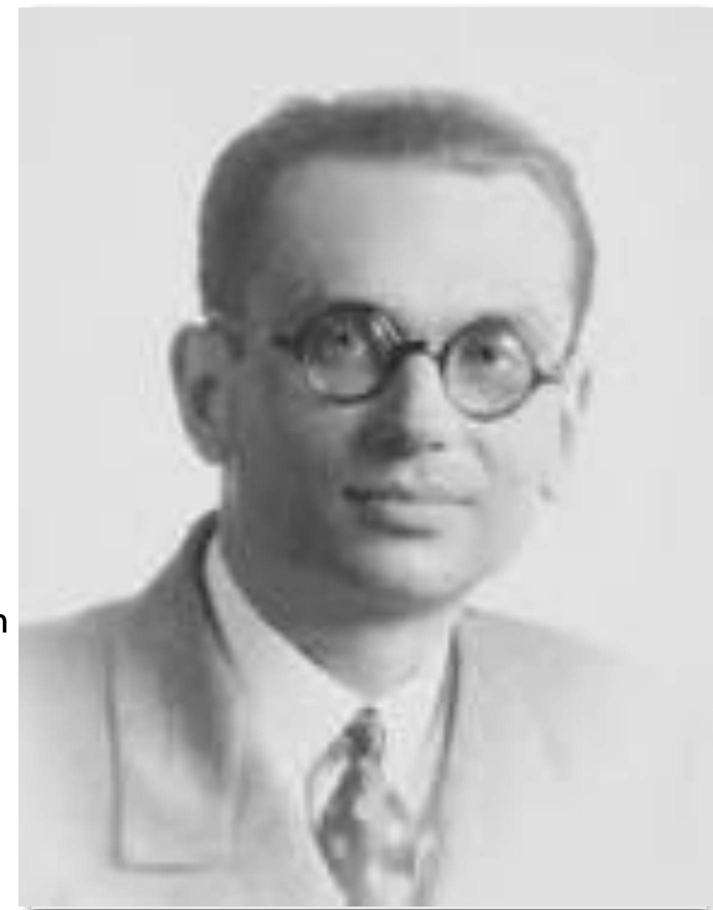
1930 Announces (First) *In*completeness Theorem

1929 Doctoral Dissertation: Proof of Completeness Theorem

Undergrad in seminar by Schlick

1923 Vienna

1906 Brünn, Austria-Hungary



Some Timeline Points

1978 Princeton NJ USA.

1940 Back to USA, for good.

1936 Schlick murdered; Austria annexed; advisor dies

1933 Hitler comes to power.

1930 Announces (First) *In*completeness Theorem

1929 Doctoral Dissertation: Proof of Completeness Theorem

Undergrad in seminar by Schlick

1923 Vienna

1906 Brünn, Austria-Hungary

Some Timeline Points

1978 Princeton NJ USA.

1940 Back to USA, for good.

1936 Schlick murdered; Austria annexed; advisor dies

1933 Hitler comes to power.

1930 Announces (First) *In*completeness Theorem

1929 Doctoral Dissertation: Proof of Completeness Theorem

Undergrad in seminar by Schlick

1923 Vienna

1906 Brünn, Austria-Hungary

Some Timeline Points

1978 Princeton NJ USA.

1940 Back to USA, for good.

1936 Schlick murdered; Austria annexed; advisor dies

1933 Hitler comes to power.

1930 Announces (First) *In*completeness Theorem

1929 Doctoral Dissertation: Proof of Completeness Theorem

Undergrad in seminar by Schlick

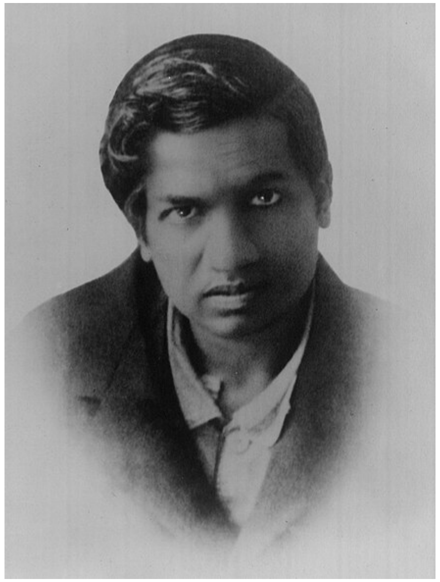
1923 Vienna

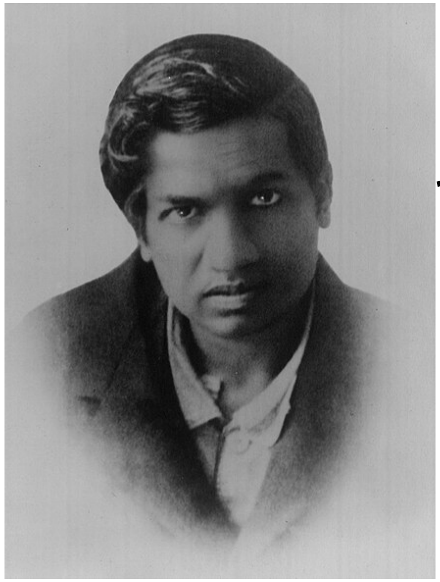
1906 Brünn, Austria-Hungary

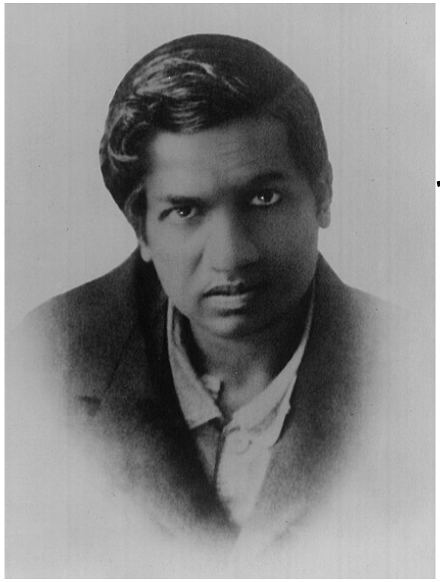
"I have proved that syntax and semantics are fundamentally the same."

Ruminations on Srinivas Ramanujan

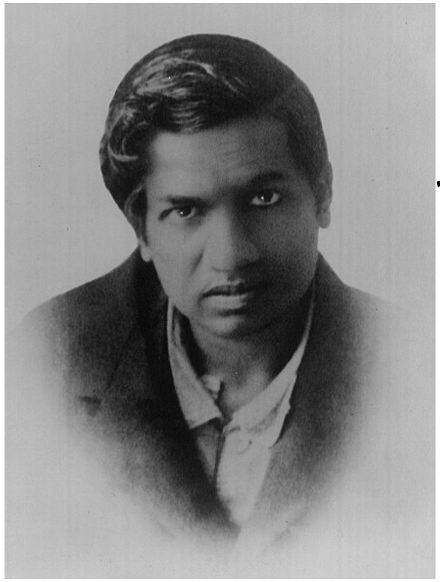
. . .





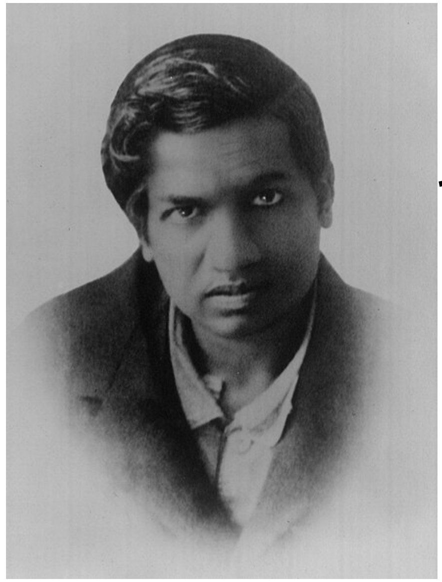


“ϕ is true!”



“ϕ is true!”

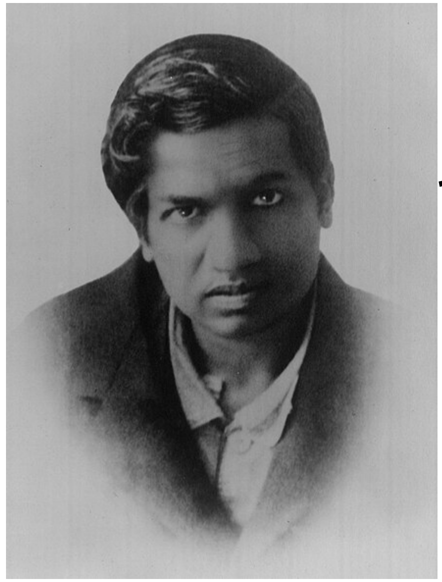
Should we trust R.?



"ϕ is true!"

Should we trust R.?

Should *everyone* trust R.?

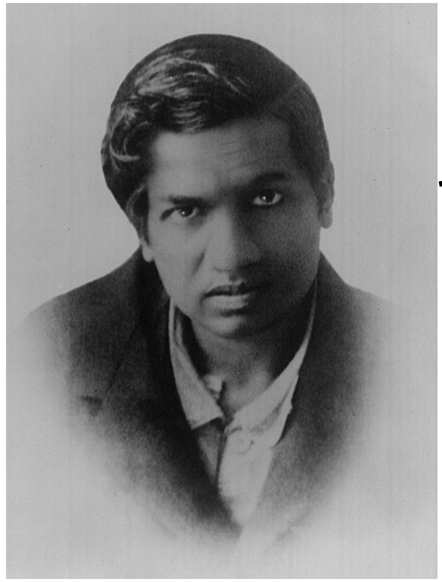


"ϕ is true!"

Should we trust R.?

Should *everyone* trust R.?

Let's assume the affirmative.



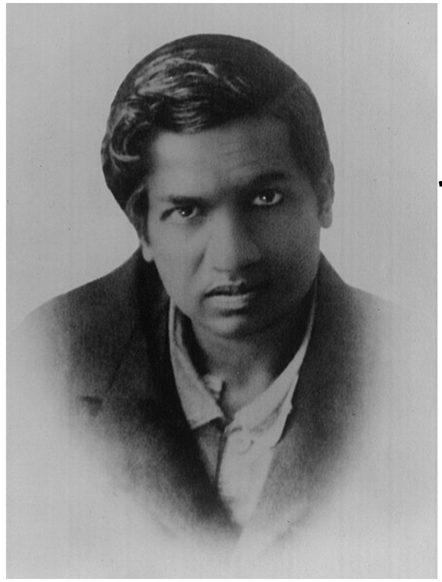
"ϕ is true!"

Should we trust R.?

Should *everyone* trust R.?

Let's assume the affirmative.

We still need … a proof!



"ϕ is true!"

Should we trust R.?

Should *everyone* trust R.?

Let's assume the affirmative.

We still need … a proof!

Or at least we need to know there *is* a proof!

Preliminaries: Propositional Calculus & First-Order Logic

. . .

Actually …

$$\mathscr{L}_0 < \mathscr{L}_1 < \mathscr{L}_2 < \mathscr{L}_3 \ldots$$

Actually …

Second-order logic.

Third-order logic, which Gödel used for his "God Theorem."

$$\mathscr{L}_0 < \mathscr{L}_1 < \mathscr{L}_2 < \mathscr{L}_3 \ldots$$

Zero-order logic; subsumes the propositional calculus.

First-order logic; this is what the Completeness Theorem is about: I.e., this logic is complete.

Actually …

Second-order logic.

Third-order logic, which Gödel used for his "God Theorem."

$$\mathscr{L}_0 < \mathscr{L}_1 < \mathscr{L}_2 < \mathscr{L}_3 \ldots$$

Zero-order logic; subsumes the propositional calculus.

First-order logic; this is what the Completeness Theorem is about: I.e., this logic is complete.

Actually …

This logic is *not* complete.**

Second-order logic.

Third-order logic, which Gödel used for his "God Theorem."

$$\mathscr{L}_0 < \mathscr{L}_1 < \mathscr{L}_2 < \mathscr{L}_3 \ldots$$

Zero-order logic; subsumes the propositional calculus.

First-order logic; this is what the Completeness Theorem is about: I.e., this logic is complete.

Actually …

This logic is *not* complete.**

Second-order logic.

Third-order logic, which Gödel used for his "God Theorem."

$$\mathscr{L}_0 < \mathscr{L}_1 < \mathscr{L}_2 < \mathscr{L}_3 \ldots$$

Zero-order logic; subsumes the propositional calculus.

First-order logic; this is what the Completeness Theorem is about: I.e., this logic is complete.

**At least in its "complete" form. There are complete *fragments* of $\mathscr{L}_2$.

R&W's Axiomatization of the Propositional Calculus

A1 $(\phi \vee \phi) \rightarrow \phi$

A2 $\phi \rightarrow (\phi \vee \psi)$

A3 $(\phi \vee \psi) \rightarrow (\psi \vee \phi)$

A4 $(\psi \rightarrow \chi) \rightarrow ((\phi \vee \psi) \rightarrow (\phi \vee \chi))$

R&W's Axiomatization of the Propositional Calculus

Some hyperlogical wizard want to throw these into a HOL workspace, and prove a substantive theorem from them?

A1 $(\phi \vee \phi) \rightarrow \phi$

A2 $\phi \rightarrow (\phi \vee \psi)$

A3 $(\phi \vee \psi) \rightarrow (\psi \vee \phi)$

A4 $(\psi \rightarrow \chi) \rightarrow ((\phi \vee \psi) \rightarrow (\phi \vee \chi))$

R&W's Axiomatization of the Propositional Calculus

Some hyperlogical wizard want to throw these into a HOL workspace, and prove a substantive theorem from them?

A1 $(\phi \vee \phi) \rightarrow \phi$

A2 $\phi \rightarrow (\phi \vee \psi)$

A3 $(\phi \vee \psi) \rightarrow (\psi \vee \phi)$

A4 $(\psi \rightarrow \chi) \rightarrow ((\phi \vee \psi) \rightarrow (\phi \vee \chi))$

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, e.g.), can be proved (somehow) from these four axioms (using any standard collection of inference schemata). This, Gödel (& later, Newell & Simon, when modern AI was born!) knew, and could use.

R&W's Axiomatization of the Propositional Calculus

Some hyperlogical wizard want to throw these into a HOL workspace, and prove a substantive theorem from them?

A1 $(\phi \vee \phi) \rightarrow \phi$

```
(if (or \phi \phi) \phi)
```

A2 $\phi \rightarrow (\phi \vee \psi)$

```
(if \phi (or \phi \psi))
```

A3 $(\phi \vee \psi) \rightarrow (\psi \vee \phi)$

```
(if (or \phi \psi) (or \psi phi))
```

A4 $(\psi \rightarrow \chi) \rightarrow ((\phi \vee \psi) \rightarrow (\phi \vee \chi))$

```
(if (if \psi \chi) (if (or \phi \psi) (or \phi \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, e.g.), can be proved (somehow) from these four axioms (using any standard collection of inference schemata). This, Gödel (& later, Newell & Simon, when modern AI was born!) knew, and could use.

Exercise 1:

Verify that these are true-no-matter what in a truth tree in HyperSlate®; then prove using our rules for the prop. calc.; or perhaps better yet, have the oracle prove in HyperSlate®.

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

$$\phi \rightarrow (\psi \rightarrow \phi)$$

Exercise 1:

Verify that these are true-no-matter what in a truth tree; then prove using our rules for the prop. calc.

$$(\phi \wedge \psi) \to (\psi \vee \chi)$$

Exercise 1:

Verify that these are true-no-matter what in a truth tree; then prove using our rules for the prop. calc.

$$(\phi \wedge \psi) \to (\psi \vee \chi)$$

√

Build **truth tree** (**semantic hypergraph**, actually) showing this formula true no matter what the inputs.

Exercise 1:

Verify that these are true-no-matter what in a truth tree; then prove using our rules for the prop. calc.

$$(\phi \wedge \psi) \to (\psi \vee \chi)$$

√

Build **truth tree** (**semantic hypergraph**, actually) showing this formula true no matter what the inputs.

Proof:

Exercise 1:

Verify that these are true-no-matter what in a truth tree; then prove using our rules for the prop. calc.

$$(\phi \wedge \psi) \rightarrow (\psi \vee \chi)$$

√ Build **truth tree** (**semantic hypergraph**, actually) showing this formula true no matter what the inputs.

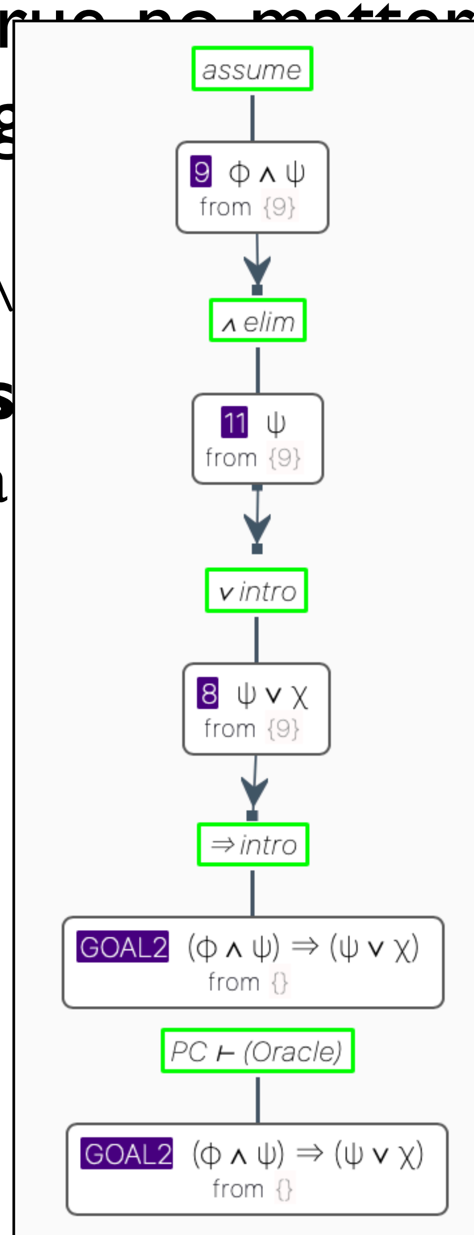
Proof:

Exercise 1:

Verify that these are tr what in a truth tree; then prove using the prop. calc.

$$(\phi \wedge \quad \chi)$$

Build **truth tree (s** **ergraph**, actually) showing this formula what the inputs.



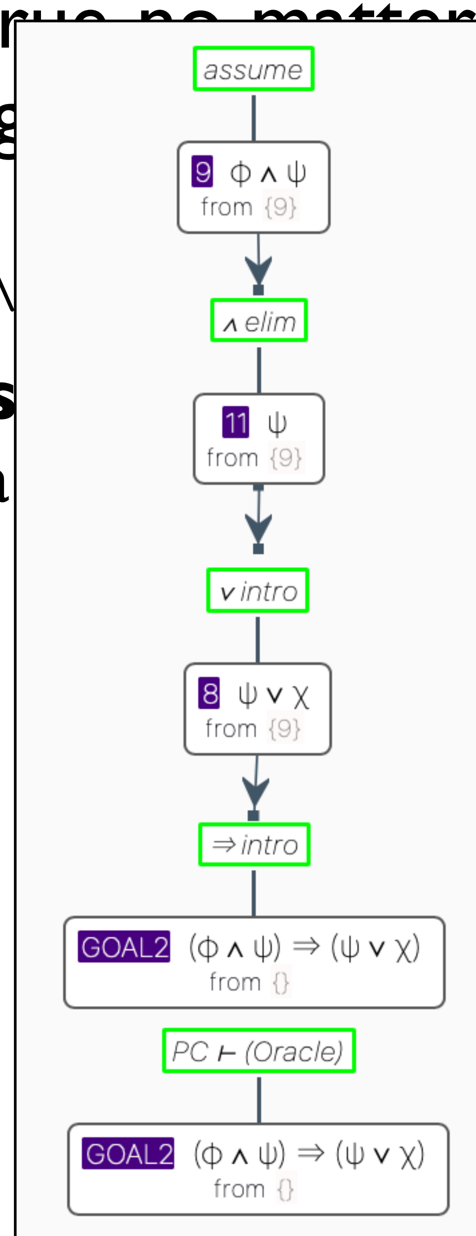
Proof:

Exercise 1:

Verify that these are tr… what in a truth tree; then prove using … the prop. calc.

$(\phi \wedge \ldots \chi)$

Build **truth tree (s… ergraph**, actually) showing this formula … what the inputs.



Proof:

resolution-based!

Exercise 1:

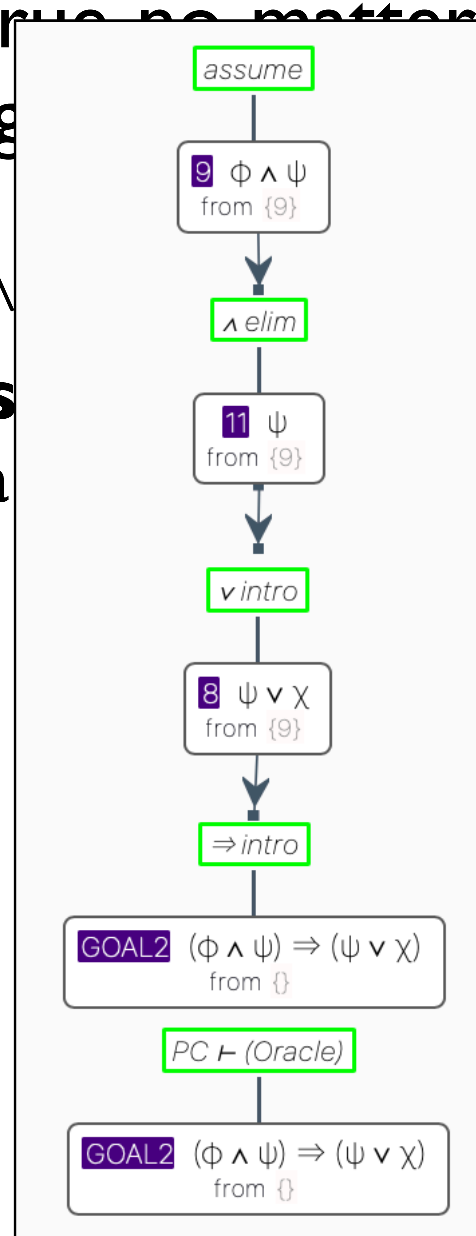
Verify that these are tr what in a truth tree;
then prove using the prop. calc.

$$(\phi \wedge \quad \chi)$$

√

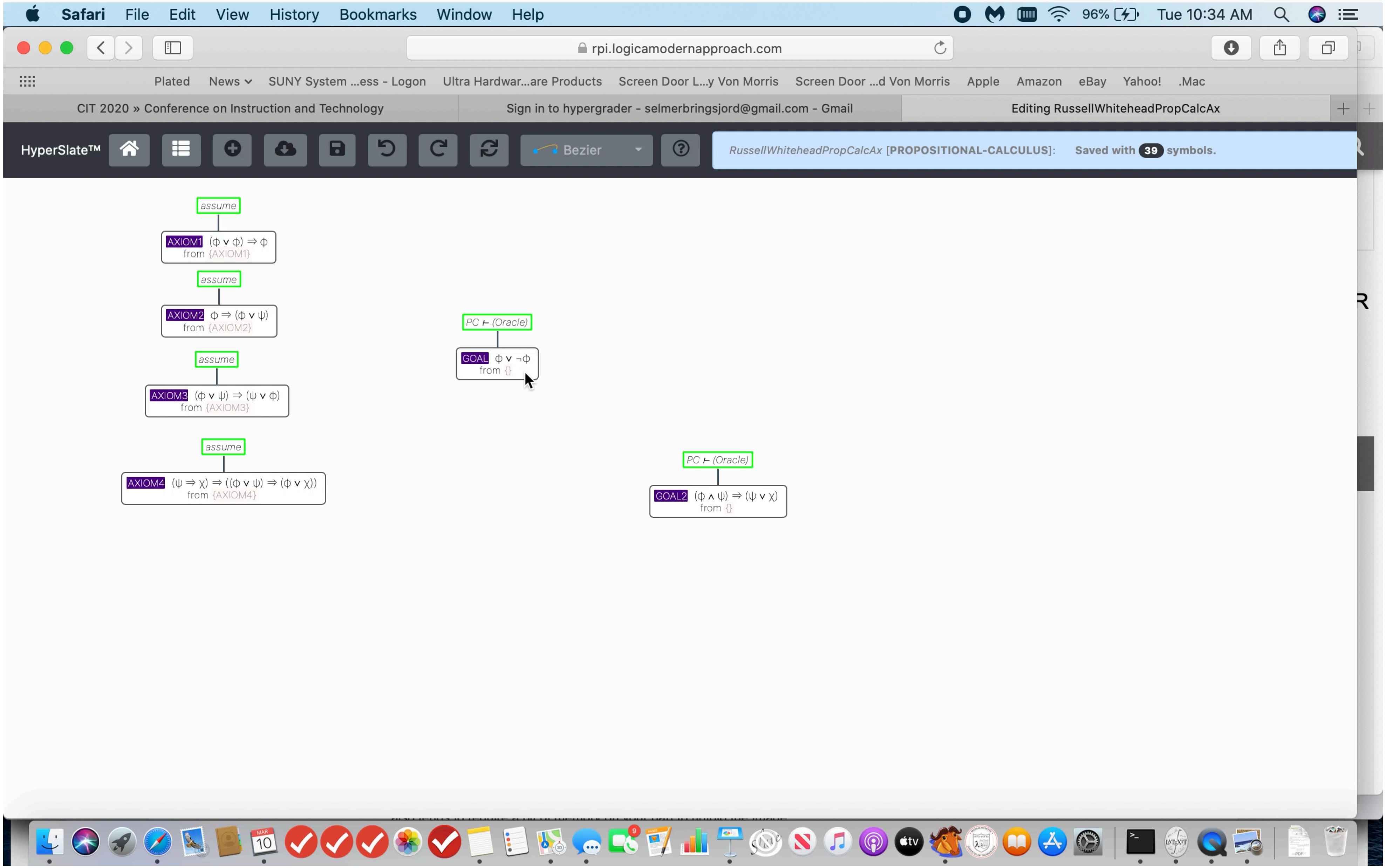
Build **truth tree (s** **ergraph**, actually)
showing this formula what the inputs.

Proof:

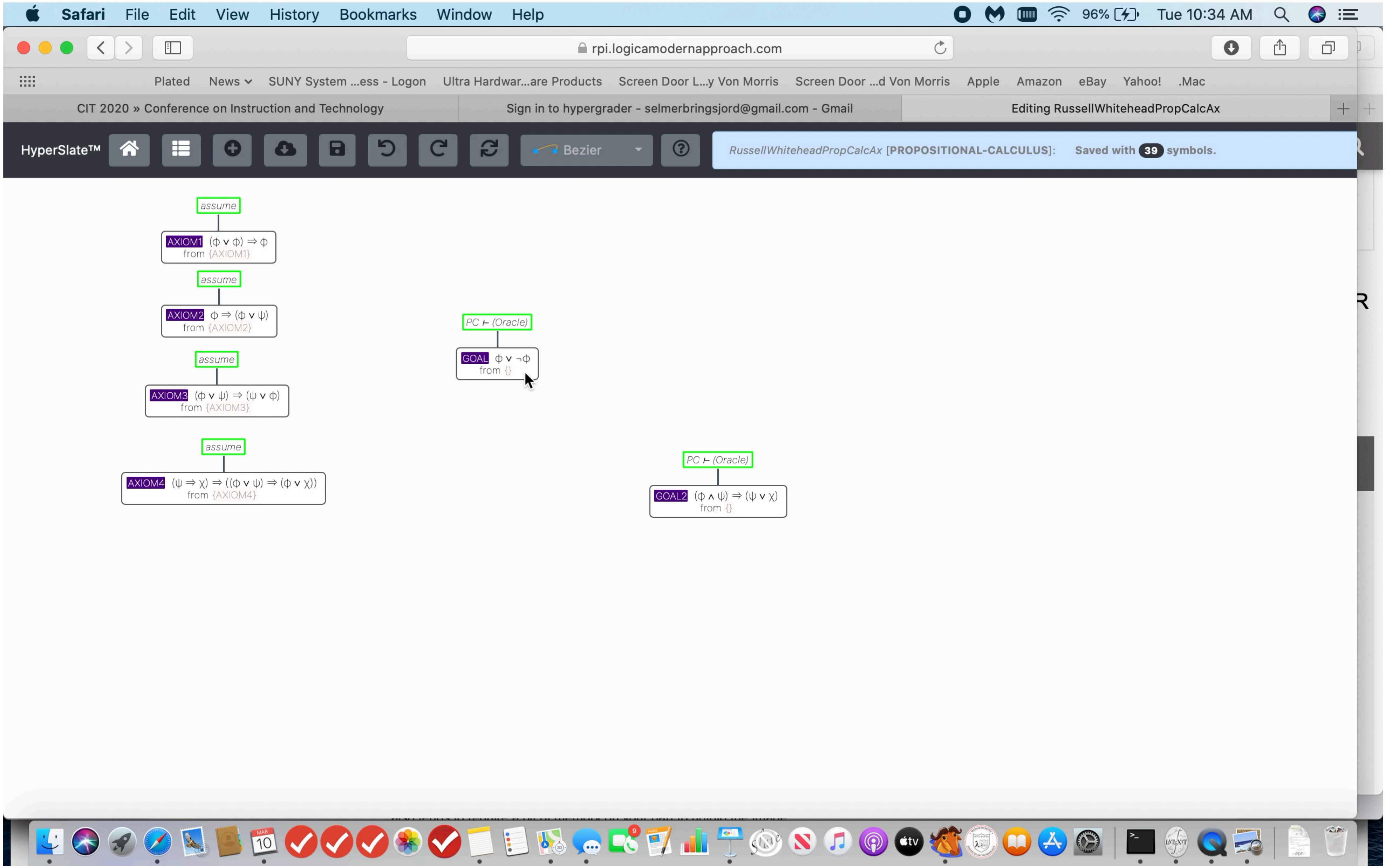


resolution-based!

As HyperSlate® Tutorial



As HyperSlate® Tutorial



The Grammar of $\mathscr{L}_0$ = the Pure Predicate Calculus

$$
\begin{array}{lcl}
\textit{Formula} & \Rightarrow & \textit{AtomicFormula} \\
 & | & (\textit{Formula}\ \ \textit{Connective}\ \ \textit{Formula}) \\
 & | & \neg\ \textit{Formula} \\
\hline
\textit{AtomicFormula} & \Rightarrow & (\textit{Predicate}\ \ \textit{Term}_1 \ldots \textit{Term}_k) \\
 & | & (\textit{Term}\ = \ \textit{Term}) \\
\hline
\textit{Term} & \Rightarrow & (\textit{Function}\ \ \textit{Term}_1\ \ \ldots\ \ \textit{Term}_k) \\
 & | & \textit{Constant} \\
\hline
\textit{Connective} & \Rightarrow & \wedge \mid \vee \mid \rightarrow \mid \leftrightarrow \\
\hline
\textit{Predicate} & \Rightarrow & P_1 \mid P_2 \mid P_3\ \ldots \\
\textit{Constant} & \Rightarrow & c_1 \mid c_2 \mid c_3\ \ldots \\
\textit{Function} & \Rightarrow & f_1 \mid f_2 \mid f_3\ \ldots
\end{array}
$$

Some Simple Examples

$$
\begin{array}{lcl}
\textit{Formula} & \Rightarrow & \textit{AtomicFormula} \\
 & \mid & (\textit{Formula Connective Formula}) \\
 & \mid & \neg\ \textit{Formula}
\end{array}
$$

$$
\begin{array}{lcl}
\textit{AtomicFormula} & \Rightarrow & (\textit{Predicate}\ \textit{Term}_1 \ldots \textit{Term}_k) \\
 & \mid & (\textit{Term} = \textit{Term})
\end{array}
$$

$$
\begin{array}{lcl}
\textit{Term} & \Rightarrow & (\textit{Function}\ \textit{Term}_1 \ \ldots\ \textit{Term}_k) \\
 & \mid & \textit{Constant}
\end{array}
$$

$$
\textit{Connective} \Rightarrow \wedge \mid \vee \mid \rightarrow \mid \leftrightarrow
$$

$$
\begin{array}{lcl}
\textit{Predicate} & \Rightarrow & P_1 \mid P_2 \mid P_3 \ \ldots \\
\textit{Constant} & \Rightarrow & c_1 \mid c_2 \mid c_3 \ \ldots \\
\textit{Function} & \Rightarrow & f_1 \mid f_2 \mid f_3 \ \ldots
\end{array}
$$

Lexicon

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.

...

Can Roger be counted upon to declare: "Yes that sentence is okay!" whenever it's conforms to this grammar?

Some Simple Examples

Formula	$\Rightarrow$	*AtomicFormula*
	$\mid$	(*Formula Connective Formula*)
	$\mid$	$\neg$ *Formula*
AtomicFormula	$\Rightarrow$	(*Predicate* $Term_1 \ldots Term_k$)
	$\mid$	(*Term* $=$ *Term*)
Term	$\Rightarrow$	(*Function* $Term_1 \ \ldots \ Term_k$)
	$\mid$	*Constant*
Connective	$\Rightarrow$	$\wedge \mid \vee \mid \rightarrow \mid \leftrightarrow$
`Likes` *Predicate*	$\Rightarrow$	$P_1 \mid P_2 \mid P_3 \ \ldots$
Constant	$\Rightarrow$	$c_1 \mid c_2 \mid c_3 \ \ldots$
Function	$\Rightarrow$	$f_1 \mid f_2 \mid f_3 \ \ldots$

Lexicon

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.

...

Can Roger be counted upon to declare: "Yes that sentence is okay!" whenever it's conforms to this grammar?

Slightly More Complicated Examples

$$
\begin{array}{lcl}
\textit{Formula} & \Rightarrow & \textit{AtomicFormula} \\
 & | & (\textit{Formula Connective Formula}) \\
 & | & \neg\ \textit{Formula}
\end{array}
$$

$$
\begin{array}{lcl}
\textit{AtomicFormula} & \Rightarrow & (\textit{Predicate}\ \textit{Term}_1 \ldots \textit{Term}_k) \\
 & | & (\textit{Term}\ =\ \textit{Term})
\end{array}
$$

$$
\begin{array}{lcl}
\textit{Term} & \Rightarrow & (\textit{Function}\ \textit{Term}_1\ \ldots\ \textit{Term}_k) \\
 & | & \textit{Constant}
\end{array}
$$

$$
\begin{array}{lcl}
\textit{Connective} & \Rightarrow & \wedge \mid \vee \mid \rightarrow \mid \leftrightarrow
\end{array}
$$

$$
\begin{array}{lcl}
\textit{Predicate} & \Rightarrow & P_1 \mid P_2 \mid P_3\ \ldots \\
\textit{Constant} & \Rightarrow & c_1 \mid c_2 \mid c_3\ \ldots \\
\textit{Function} & \Rightarrow & f_1 \mid f_2 \mid f_3\ \ldots
\end{array}
$$

Slightly More Complicated Examples

$$
\begin{array}{lcl}
\textit{Formula} & \Rightarrow & \textit{AtomicFormula} \\
 & | & (\textit{Formula Connective Formula}) \\
 & | & \neg\ \textit{Formula}
\end{array}
$$

$$
\begin{array}{lcl}
\textit{AtomicFormula} & \Rightarrow & (\textit{Predicate}\ \textit{Term}_1 \ldots \textit{Term}_k) \\
 & | & (\textit{Term} = \textit{Term})
\end{array}
$$

$$
\begin{array}{lcl}
\textit{Term} & \Rightarrow & (\textit{Function}\ \textit{Term}_1\ \ldots\ \textit{Term}_k) \\
 & | & \textit{Constant}
\end{array}
$$

$$
\textit{Connective} \Rightarrow \wedge \mid \vee \mid \rightarrow \mid \leftrightarrow
$$

$$
\begin{array}{lcl}
\textit{Predicate} & \Rightarrow & P_1 \mid P_2 \mid P_3\ \ldots \\
\textit{Constant} & \Rightarrow & c_1 \mid c_2 \mid c_3\ \ldots \\
\textit{Function} & \Rightarrow & f_1 \mid f_2 \mid f_3\ \ldots
\end{array}
$$

If Sally likes Bill then Sally likes Bill.

Slightly More Complicated Examples

$$
\begin{array}{lcl}
\textit{Formula} & \Rightarrow & \textit{AtomicFormula} \\
 & \mid & (\textit{Formula Connective Formula}) \\
 & \mid & \neg\ \textit{Formula}
\end{array}
$$

$$
\begin{array}{lcl}
\textit{AtomicFormula} & \Rightarrow & (\textit{Predicate}\ \textit{Term}_1 \ldots \textit{Term}_k) \\
 & \mid & (\textit{Term}\ =\ \textit{Term})
\end{array}
$$

$$
\begin{array}{lcl}
\textit{Term} & \Rightarrow & (\textit{Function}\ \textit{Term}_1\ \ldots\ \textit{Term}_k) \\
 & \mid & \textit{Constant}
\end{array}
$$

$$
\textit{Connective} \quad \Rightarrow \quad \wedge \mid \vee \mid \rightarrow \mid \leftrightarrow
$$

$$
\begin{array}{lcl}
\textit{Predicate} & \Rightarrow & P_1 \mid P_2 \mid P_3\ \ldots \\
\textit{Constant} & \Rightarrow & c_1 \mid c_2 \mid c_3\ \ldots \\
\textit{Function} & \Rightarrow & f_1 \mid f_2 \mid f_3\ \ldots
\end{array}
$$

If Sally likes Bill then Sally likes Bill.

Sally likes Bill's mother, or not.

Slightly More Complicated Examples

$$
\begin{array}{lcl}
\mathit{Formula} & \Rightarrow & \mathit{AtomicFormula} \\
 & | & (\mathit{Formula}\ \ \mathit{Connective}\ \ \mathit{Formula}) \\
 & | & \neg\ \mathit{Formula}
\end{array}
$$

$$
\begin{array}{lcl}
\mathit{AtomicFormula} & \Rightarrow & (\mathit{Predicate}\ \ \mathit{Term}_1 \ldots \mathit{Term}_k) \\
 & | & (\mathit{Term}\ \ =\ \ \mathit{Term})
\end{array}
$$

$$
\begin{array}{lcl}
\mathit{Term} & \Rightarrow & (\mathit{Function}\ \ \mathit{Term}_1\ \ \ldots\ \ \mathit{Term}_k) \\
 & | & \mathit{Constant}
\end{array}
$$

$$
\begin{array}{lcl}
\mathit{Connective} & \Rightarrow & \wedge \mid \vee \mid \rightarrow \mid \leftrightarrow
\end{array}
$$

$$
\begin{array}{lcl}
\mathit{Predicate} & \Rightarrow & P_1 \mid P_2 \mid P_3\ \ldots \\
\mathit{Constant} & \Rightarrow & c_1 \mid c_2 \mid c_3\ \ldots \\
\mathit{Function} & \Rightarrow & f_1 \mid f_2 \mid f_3\ \ldots
\end{array}
$$

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.

Slightly More Complicated Examples

$$
\begin{array}{lcl}
\textit{Formula} & \Rightarrow & \textit{AtomicFormula} \\
 & | & (\textit{Formula Connective Formula}) \\
 & | & \neg\ \textit{Formula} \\
\hline
\textit{AtomicFormula} & \Rightarrow & (\textit{Predicate } \textit{Term}_1 \ldots \textit{Term}_k) \\
 & | & (\textit{Term} = \textit{Term}) \\
\hline
\textit{Term} & \Rightarrow & (\textit{Function } \textit{Term}_1 \ \ldots \ \textit{Term}_k) \\
 & | & \textit{Constant} \\
\hline
\textit{Connective} & \Rightarrow & \wedge \mid \vee \mid \rightarrow \mid \leftrightarrow \\
\hline
\textit{Predicate} & \Rightarrow & P_1 \mid P_2 \mid P_3 \ \ldots \\
\textit{Constant} & \Rightarrow & c_1 \mid c_2 \mid c_3 \ \ldots \\
\textit{Function} & \Rightarrow & f_1 \mid f_2 \mid f_3 \ \ldots
\end{array}
$$

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.

Slightly More Complicated Examples

$$
\begin{array}{lcl}
\textit{Formula} & \Rightarrow & \textit{AtomicFormula} \\
 & | & (\textit{Formula}\ \ \textit{Connective}\ \ \textit{Formula}) \\
 & | & \neg\ \textit{Formula}
\end{array}
$$

$$
\begin{array}{lcl}
\textit{AtomicFormula} & \Rightarrow & (\textit{Predicate}\ \ \textit{Term}_1 \ldots \textit{Term}_k) \\
 & | & (\textit{Term}\ = \ \textit{Term})
\end{array}
$$

$$
\begin{array}{lcl}
\textit{Term} & \Rightarrow & (\textit{Function}\ \ \textit{Term}_1\ \ldots\ \textit{Term}_k) \\
 & | & \textit{Constant}
\end{array}
$$

$$
\textit{Connective} \Rightarrow \wedge \mid \vee \mid \rightarrow \mid \leftrightarrow
$$

$$
\begin{array}{lcl}
\textit{Predicate} & \Rightarrow & P_1 \mid P_2 \mid P_3\ \ldots \\
\textit{Constant} & \Rightarrow & c_1 \mid c_2 \mid c_3\ \ldots \\
\textit{Function} & \Rightarrow & f_1 \mid f_2 \mid f_3\ \ldots
\end{array}
$$

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.

...

Slightly More Complicated Examples

$$
\begin{array}{lll}
\textit{Formula} & \Rightarrow & \textit{AtomicFormula} \\
 & | & (\textit{Formula Connective Formula}) \\
 & | & \neg\ \textit{Formula}
\end{array}
$$

$$
\begin{array}{lll}
\textit{AtomicFormula} & \Rightarrow & (\textit{Predicate } \textit{Term}_1 \ldots \textit{Term}_k) \\
 & | & (\textit{Term} = \textit{Term})
\end{array}
$$

$$
\begin{array}{lll}
\textit{Term} & \Rightarrow & (\textit{Function } \textit{Term}_1 \ \ldots \ \textit{Term}_k) \\
 & | & \textit{Constant}
\end{array}
$$

$$
\textit{Connective} \Rightarrow \wedge \mid \vee \mid \rightarrow \mid \leftrightarrow
$$

$$
\begin{array}{lll}
\textit{Predicate} & \Rightarrow & P_1 \mid P_2 \mid P_3 \ \ldots \\
\textit{Constant} & \Rightarrow & c_1 \mid c_2 \mid c_3 \ \ldots \\
\textit{Function} & \Rightarrow & f_1 \mid f_2 \mid f_3 \ \ldots
\end{array}
$$

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?!

Slightly More Complicated Examples

$$
\begin{array}{lcl}
\textit{Formula} & \Rightarrow & \textit{AtomicFormula} \\
 & | & (\textit{Formula Connective Formula}) \\
 & | & \neg\ \textit{Formula} \\
\hline
\textit{AtomicFormula} & \Rightarrow & (\textit{Predicate}\ \textit{Term}_1 \ldots \textit{Term}_k) \\
 & | & (\textit{Term} = \textit{Term}) \\
\hline
\textit{Term} & \Rightarrow & (\textit{Function}\ \textit{Term}_1 \ \ldots\ \textit{Term}_k) \\
 & | & \textit{Constant} \\
\hline
\textit{Connective} & \Rightarrow & \wedge \mid \vee \mid \rightarrow \mid \leftrightarrow \\
\hline
\textit{Predicate} & \Rightarrow & P_1 \mid P_2 \mid P_3 \ \ldots \\
\textit{Constant} & \Rightarrow & c_1 \mid c_2 \mid c_3 \ \ldots \\
\textit{Function} & \Rightarrow & f_1 \mid f_2 \mid f_3 \ \ldots
\end{array}
$$

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?!

Yes!

But Now the Deeper Challenge:
Add Two Quantifiers to the Pure Predicate Calculus,
Which Yields $\mathscr{L}_1$ First-order Logic = Predicate Calculus *simpliciter*

But Now the Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields $\mathscr{L}_1$ First-order Logic = Predicate Calculus *simpliciter*

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

But Now the Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields $\mathscr{L}_1$ First-order Logic = Predicate Calculus *simpliciter*

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

for all *x*, it's the case that …

But Now the Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields $\mathscr{L}_1$ First-order Logic = Predicate Calculus *simpliciter*

$\exists x \quad \ldots$ there exists at least one thing *x* such that …

for all *x*, it's the case that …

But Now the Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields $\mathscr{L}_1$ First-order Logic = 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 …

But Now the Deeper Challenge:
Add Two Quantifiers to the Pure Predicate Calculus, Which Yields $\mathscr{L}_1$ First-order Logic = 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 …

$$\lim_{x \to a} f(x) = L$$

iff

$$\forall \epsilon (\epsilon > 0 \rightarrow \exists \delta (\delta > 0 \wedge \forall x (d(x, a) < \delta \rightarrow d(f(x), L) < \epsilon)))$$

But Now the Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields $\mathscr{L}_1$ First-order Logic = 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 …

$$\lim_{x \to a} f(x) = L$$

```
(forall (\epsilon) (if (> \epsilon 0)
                       (exists (\delta) (and (> \delta 0)
                                             (forall (x) (if (< (dist x a) \delta)
                                                             (< (dist (f x) L) \epsilon)))))))
```

$$\forall \epsilon (\epsilon > 0 \rightarrow \exists \delta (\delta > 0 \wedge \forall x (d(x,a) < \delta \rightarrow d(f(x), L) < \epsilon)))$$

But Now the Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields $\mathscr{L}_1$ First-order Logic = 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 …

$$\lim_{x \to a} f(x) = L$$

```
(forall (\epsilon) (if (> \epsilon 0)
                      (exists (\delta) (and (> \delta 0)
                                             (forall (x) (if (< (dist x a) \delta)
                                                             (< (dist (f x) L) \epsilon)))))))
```

$$\forall \epsilon (\epsilon > 0 \rightarrow \exists \delta (\delta > 0 \land \forall x (d(x,a) < \delta \rightarrow d(f(x), L) < \epsilon)))$$

Every natural number is greater than or equal to zero.

But Now the Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields $\mathscr{L}_1$ First-order Logic = 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 …

$$\lim_{x \to a} f(x) = L$$

```
(forall (\epsilon) (if (> \epsilon 0)
                       (exists (\delta) (and (> \delta 0)
                                             (forall (x) (if (< (dist x a) \delta)
                                                             (< (dist (f x) L) \epsilon)))))))
```

$$\forall \epsilon (\epsilon > 0 \rightarrow \exists \delta (\delta > 0 \wedge \forall x (d(x,a) < \delta \rightarrow d(f(x), L) < \epsilon)))$$

Every natural number is greater than or equal to zero.

$$\forall x (x \geq 0)$$

But Now the Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields $\mathscr{L}_1$ First-order Logic = 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 …

$$\lim_{x \to a} f(x) = L$$

```
(forall (\epsilon) (if (> \epsilon 0)
                       (exists (\delta) (and (> \delta 0)
                                             (forall (x) (if (< (dist x a) \delta)
                                                             (< (dist (f x) L) \epsilon)))))))
```

$$\forall \epsilon (\epsilon > 0 \rightarrow \exists \delta (\delta > 0 \wedge \forall x (d(x,a) < \delta \rightarrow d(f(x), L) < \epsilon)))$$

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.

But Now the Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields $\mathscr{L}_1$ First-order Logic = Predicate Calculus *simpliciter*

$\exists x \quad \ldots$ there exists at least one thing *x* such that …

$\forall x \quad \ldots$ for all *x*, it's the case that …

$$\lim_{x \to a} f(x) = L$$

```
(forall (\epsilon) (if (> \epsilon 0)
                       (exists (\delta) (and (> \delta 0)
                                             (forall (x) (if (< (dist x a) \delta)
                                                             (< (dist (f x) L) \epsilon)))))))
```

$$\forall \epsilon (\epsilon > 0 \rightarrow \exists \delta (\delta > 0 \wedge \forall x (d(x,a) < \delta \rightarrow d(f(x), L) < \epsilon)))$$

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.

$$\exists x \forall y (y < x)$$

But Now the Deeper Challenge:
Add Two Quantifiers to the Pure Predicate Calculus,
Which Yields $\mathscr{L}_1$ First-order Logic = Predicate Calculus *simpliciter*

$\exists x \quad \ldots$ there exists at least one thing *x* such that …

$\forall x \quad \ldots$ for all *x*, it's the case that …

$$\lim_{x \to a} f(x) = L$$

```
(forall (\epsilon) (if (> \epsilon 0)
                       (exists (\delta) (and (> \delta 0)
                                             (forall (x) (if (< (dist x a) \delta)
                                                             (< (dist (f x) L) \epsilon)))))))
```

$$\forall \epsilon (\epsilon > 0 \rightarrow \exists \delta (\delta > 0 \wedge \forall x (d(x, a) < \delta \rightarrow d(f(x), L) < \epsilon)))$$

Every natural number is greater than or equal to zero.

$$\forall x (x \geq 0)$$

$$\exists x \forall y (y < x)$$

But Now the Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields $\mathscr{L}_1$ First-order Logic = 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 …

$$\lim_{x \to a} f(x) = L$$

```
(forall (\epsilon) (if (> \epsilon 0)
                      (exists (\delta) (and (> \delta 0)
                                            (forall (x) (if (< (dist x a) \delta)
                                                            (< (dist (f x) L) \epsilon)))))))
```

$$\forall \epsilon (\epsilon > 0 \rightarrow \exists \delta (\delta > 0 \wedge \forall x (d(x,a) < \delta \rightarrow d(f(x), L) < \epsilon)))$$

Every natural number is greater than or equal to zero.

$$\forall x (x \geq 0)$$

But Now the Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields $\mathscr{L}_1$ First-order Logic = Predicate Calculus *simpliciter*

$\exists x \quad \ldots$ there exists at least one thing *x* such that …

$\forall x \quad \ldots$ for all *x*, it's the case that …

$$\lim_{x \to a} f(x) = L$$

```
(forall (\epsilon) (if (> \epsilon 0)
                       (exists (\delta) (and (> \delta 0)
                                             (forall (x) (if (< (dist x a) \delta)
                                                             (< (dist (f x) L) \epsilon)))))))
```

$$\forall \epsilon (\epsilon > 0 \rightarrow \exists \delta (\delta > 0 \land \forall x (d(x, a) < \delta \rightarrow d(f(x), L) < \epsilon)))$$

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*.

But Now the Deeper Challenge: Add Two Quantifiers to the Pure Predicate Calculus, Which Yields $\mathscr{L}_1$ First-order Logic = 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 …

$$\lim_{x \to a} f(x) = L$$

```
(forall (\epsilon) (if (> \epsilon 0)
                       (exists (\delta) (and (> \delta 0)
                                             (forall (x) (if (< (dist x a) \delta)
                                                             (< (dist (f x) L) \epsilon)))))))
```

$$\forall \epsilon (\epsilon > 0 \rightarrow \exists \delta (\delta > 0 \land \forall x (d(x, a) < \delta \rightarrow d(f(x), L) < \epsilon)))$$

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*.

$$\forall x \exists y (x \leq y) \qquad \forall x \exists y (\leq (x, y))$$

The Shoulders Available to Gödel for Standing Upon

. . .

Completeness Theorem for The Propositional Calculus

Let Γ be a set $\{\phi_1, \phi_2, \ldots\}$ of formulae in the the propositional calculus. Then either all of Γ are satisfiable, or the conjunction up to and including the point k (i.e. $\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_k$) of failure is refutable.

Completeness Theorem for The Propositional Calculus

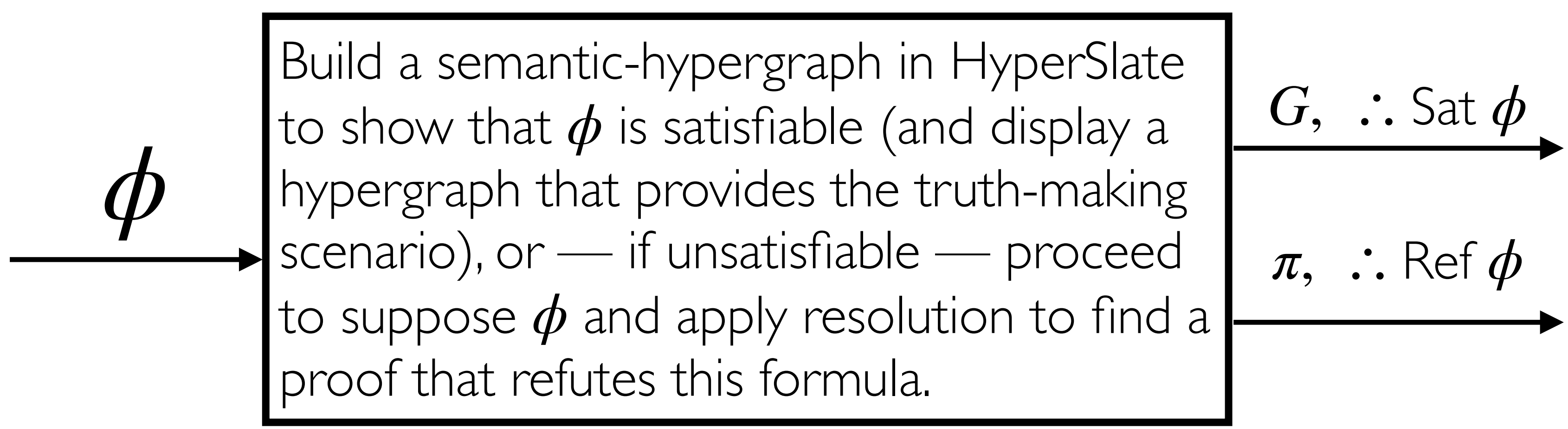
Let Γ be a set $\{\phi_1, \phi_2, \ldots\}$ of formulae in the the propositional calculus. Then either all of Γ are satisfiable, or the conjunction up to and including the point k (i.e. $\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_k$) of failure is refutable.

Completeness Theorem for The Propositional Calculus

Let Γ be a set $\{\phi_1, \phi_2, \ldots\}$ of formulae in the the propositional calculus. Then either all of Γ are satisfiable, or the conjunction up to and including the point k (i.e. $\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_k$) of failure is refutable.

Let Γ be a set $\{\phi_1, \phi_2, \ldots\}$ of formulae in the the propositional calculus. Then either all of Γ can be simultaneously true in some scenario, or the conjunction up to and including the point k (i.e. $\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_k$) of failure is **refutable** (i.e. $\vdash \neg(\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_k)$).

Completeness Theorem (prop calc):
Either ϕ is satisfiable, or ϕ is refutable.



What does the F-O Completeness Theorem say?

. . .

Completeness Theorem as an Equation

In first-order logic: NECESSARY TRUTH = PROVABILITY.

Completeness Theorem, More Precisely Put

For every first-order statement ϕ: if ϕ is a necessary or absolute truth (i.e. true in any scenario whatsoever), then ϕ is provable.

And the version Gödel targeted, and proved:

For every first-order statement ϕ: Either ϕ is true in some scenario, or ϕ is refutable (= it's negation $\neg\phi$ can be proved).

GCT

The Proof-Sketch

Gödel’s Version Equivalent

Gödel's Version Equivalent

$\models \phi$ implies $\vdash \phi$

iff

Either ϕ is true in some scenario (i.e. Sat ϕ),
or ϕ is refutable (i.e. $\vdash \neg\phi$).

Gödel's Version Equivalent

$\models \phi$ implies $\vdash \phi$

iff

Either ϕ is true in some scenario (i.e. Sat ϕ),
or ϕ is refutable (i.e. $\vdash \neg\phi$).

Prove the equivalence.

Gödel's Version Equivalent

$\models \phi$ implies $\vdash \phi$

iff

Either ϕ is true in some scenario (i.e. Sat ϕ),
or ϕ is refutable (i.e. $\vdash \neg\phi$).

Prove the equivalence.

Can you prove Gödel's version for the propositional calculus in HyperSlate?

The Proof-Sketch

To prove the theorem in the case of first-order logic ($= \mathscr{L}_1$), we need to show that given any set Γ of formulae in first-order logic, either there's a scenario on which every member of this set is true; otherwise, there is a refutation of the set, i.e. a proof from the set to an outright contradiction $\phi \wedge \neg\phi$. We can accomplish this by finding a procedure $\mathscr{P}$ that first takes the set in question and goes hunting for a scenario that does the trick. If the scenario is found, we're done. But, if such a scenario *can*'*t* be found, then our procedure moves on to find a proof of a contradiction from Γ!

How?! The procedure $\mathscr{P}$ is the building out of a semantic hypergraph! If all the branches in the graph close, then the finding of a proof of a contradiction uses **resolution**, and the **resolution guarantee**. The guarantee is that if you have a set of formulae that can't be true in any scenario, resolution applied to the set finds a contradiction $\perp = \zeta \wedge \neg\zeta = \{\}$. **QED**

The Proof-Sketch

To prove the theorem in the case of first-order logic ($= \mathscr{L}_1$), we need to show that given any set Γ of formulae in first-order logic, either there's a scenario on which every member of this set is true; otherwise, there is a refutation of the set, i.e. a proof from the set to an outright contradiction $\phi \wedge \neg\phi$. We can accomplish this by finding a procedure $\mathscr{P}$ that first takes the set in question and goes hunting for a scenario that does the trick.
If the scenario … nario *can't*
be found, then … of of a
contradiction …

See LAMA-BDLAHGHS.

How?! The procedure $\mathscr{P}$ is the building out of a semantic hypergraph! If all the branches in the graph close, then the finding of a proof of a contradiction uses **resolution**, and the **resolution guarantee**. The guarantee is that if you have a set of formulae that can't be true in any scenario, resolution applied to the set finds a contradiction $\bot = \zeta \wedge \neg\zeta = \{\}$. **QED**

Gödel as Giant: Some Evidence

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.

Gödel as Giant: Some Evidence

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.

But How'd He Handle All of $\mathscr{L}_1$?

But How'd He Handle All of $\mathscr{L}_1$?

Gödel proves the lemma that if the GCT holds for formula of degree *j*, GCT holds of formulae of degree *j*+1. So the challenge reduces to formulae of degree 1:

But How'd He Handle All of $\mathscr{L}_1$?

Gödel proves the lemma that if the GCT holds for formula of degree *j*, GCT holds of formulae of degree *j*+1. So the challenge reduces to formulae of degree 1:

$$\forall x_1, x_2, \ldots, x_k \exists y_1, y_2, \ldots, y_m \phi(x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_m)$$

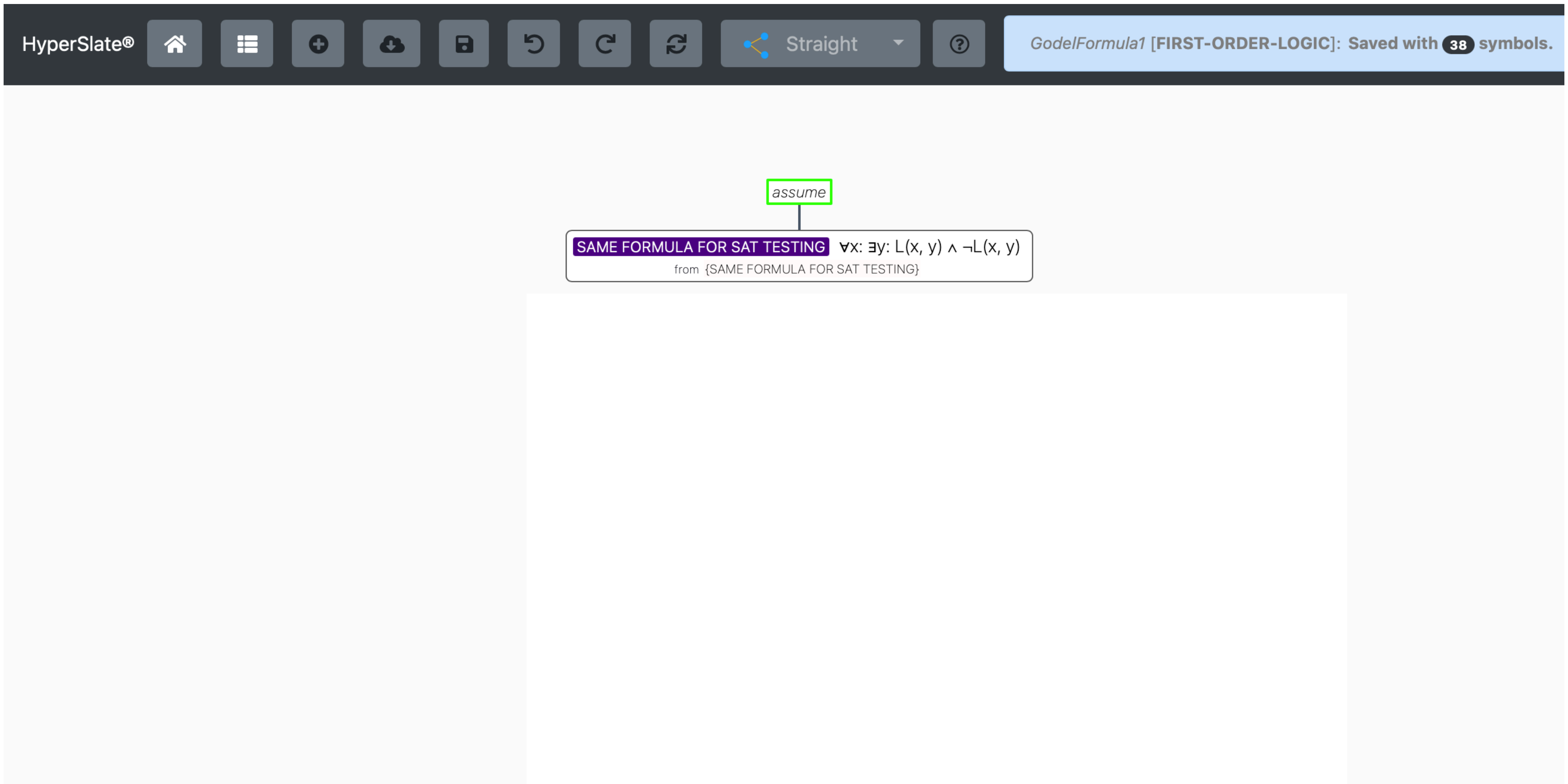
But How'd He Handle All of $\mathscr{L}_1$?

Gödel proves the lemma that if the GCT holds for formula of degree *j*, GCT holds of formulae of degree *j*+1. So the challenge reduces to formulae of degree 1:

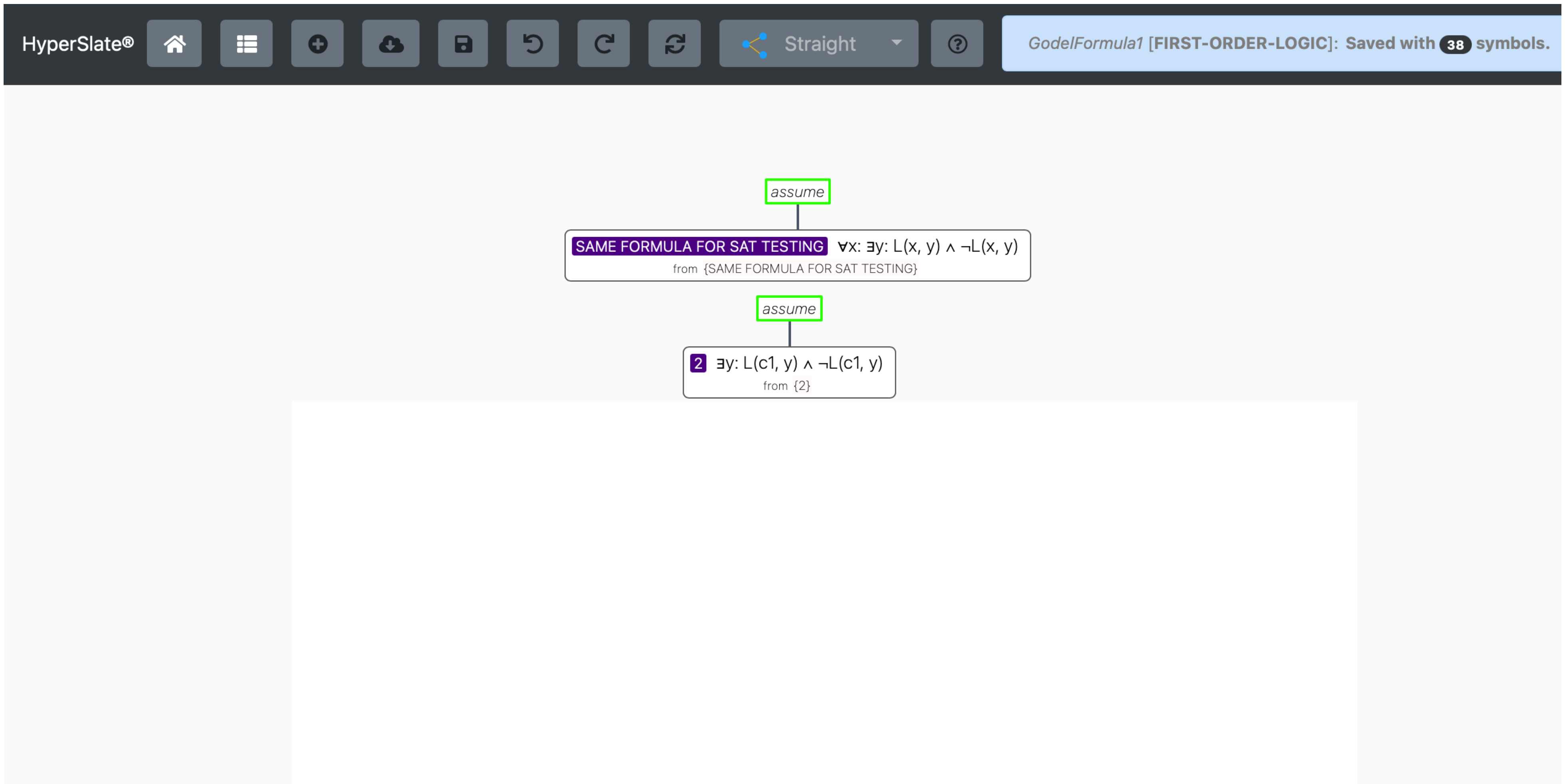
$$\forall x_1, x_2, \ldots, x_k \exists y_1, y_2, \ldots, y_m \phi(x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_m)$$

How? By ingenious tree-building, which starts by creating an enumeration of new constants $\mathfrak{c} = c_1, c_2, \ldots$ that becomes our "universe of discourse"/"domain of quantification." Note that from $\mathfrak{c}$ we can algorithmically generate an enumeration of tuples $\mathfrak{c}^t = \langle c \rangle_1, \langle c \rangle_2, \ldots$ of any finite size. (Those of size *k* will work for the *x*-variables, and those of size *n* will work for the *y*-variables.) And now we can build a BIG tree at the level of the pure predicate calculus, looking for either a scenario that makes our formula true by traveling with Buzz to infinity, or getting all branches closed, in which case we turn back to the resolution guarantee! Let's make sense of this by hand on paper …

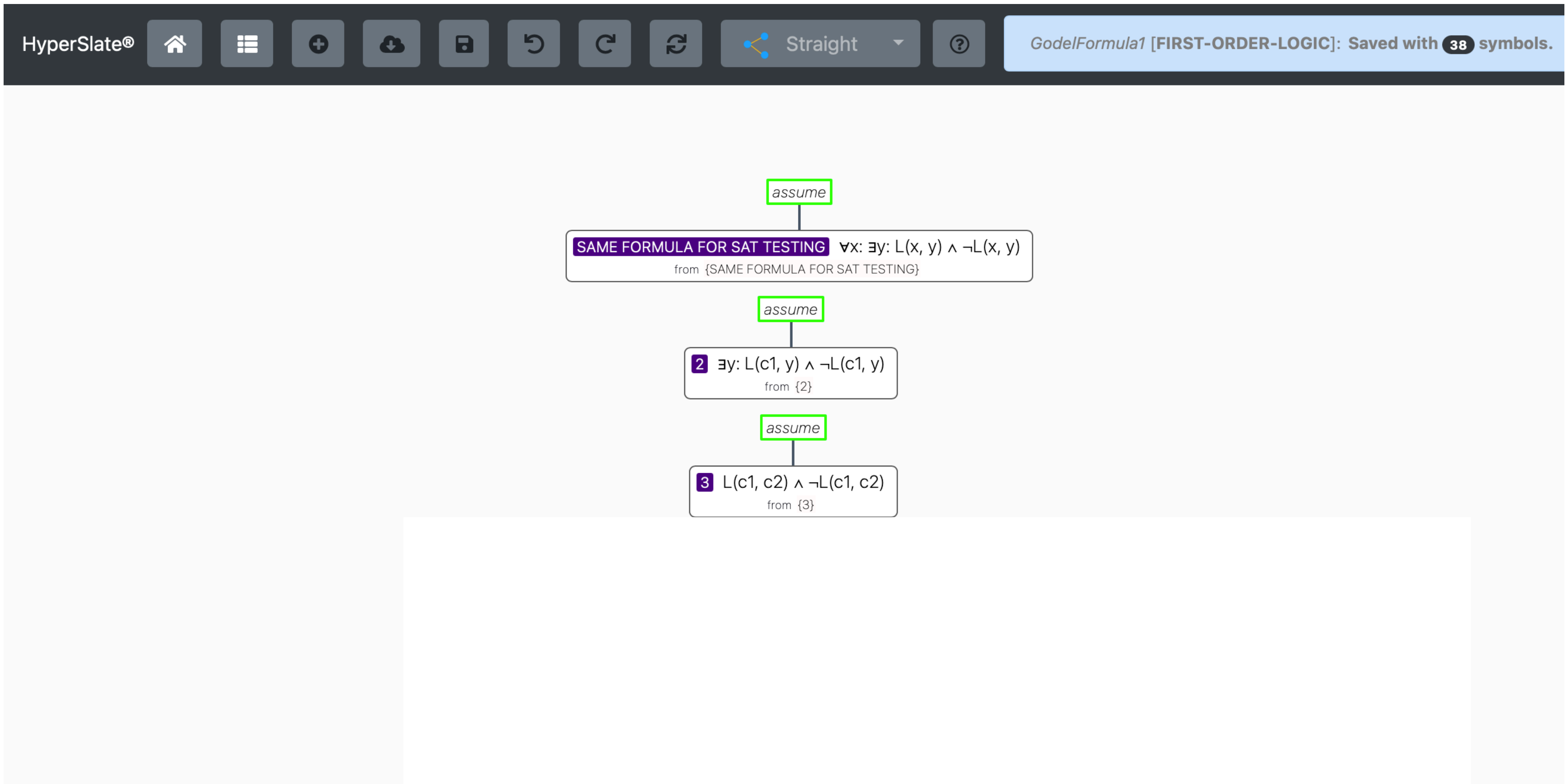
Making this Concrete Courtesy of HyperSlate®



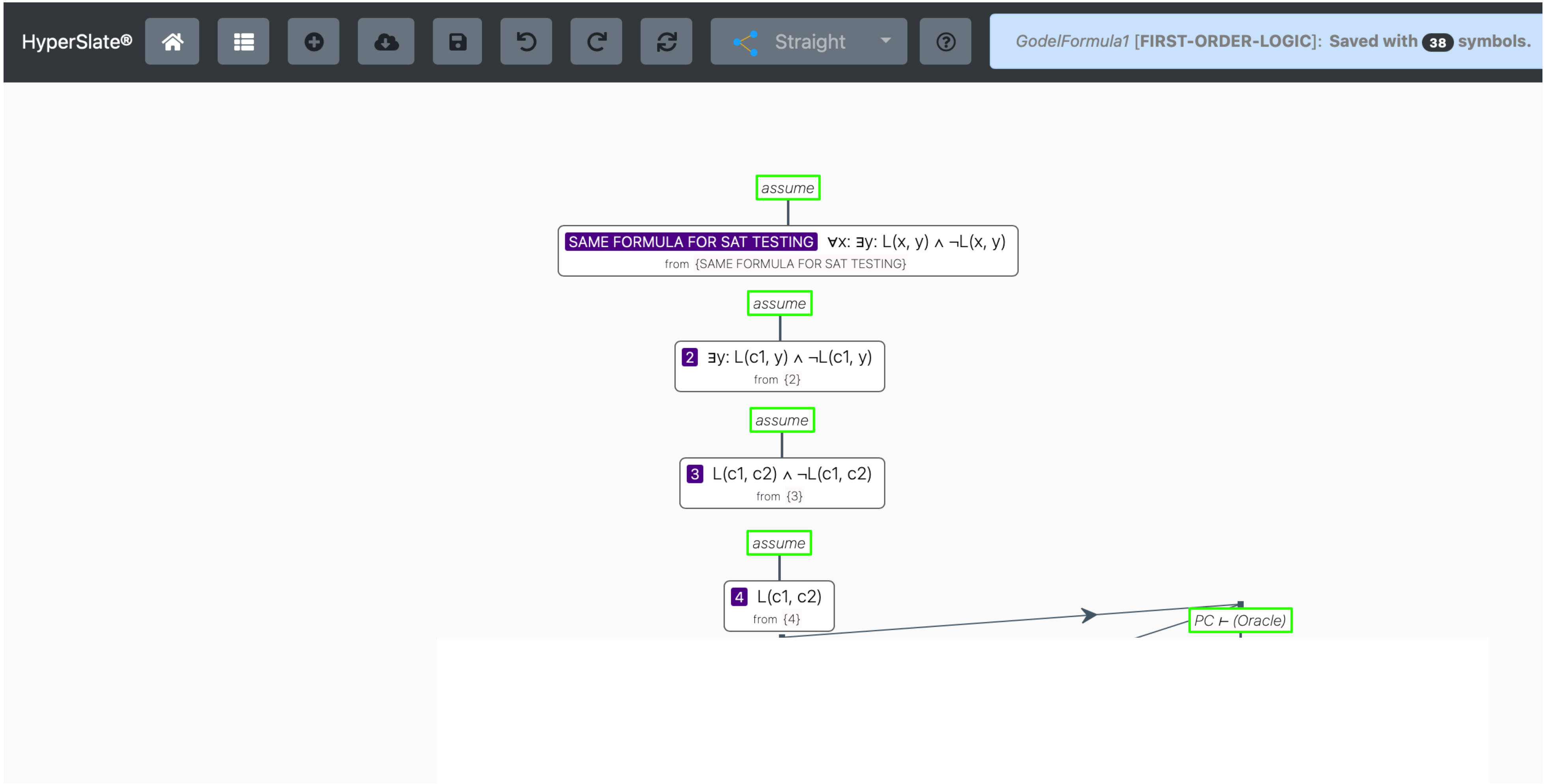
Making this Concrete Courtesy of HyperSlate®



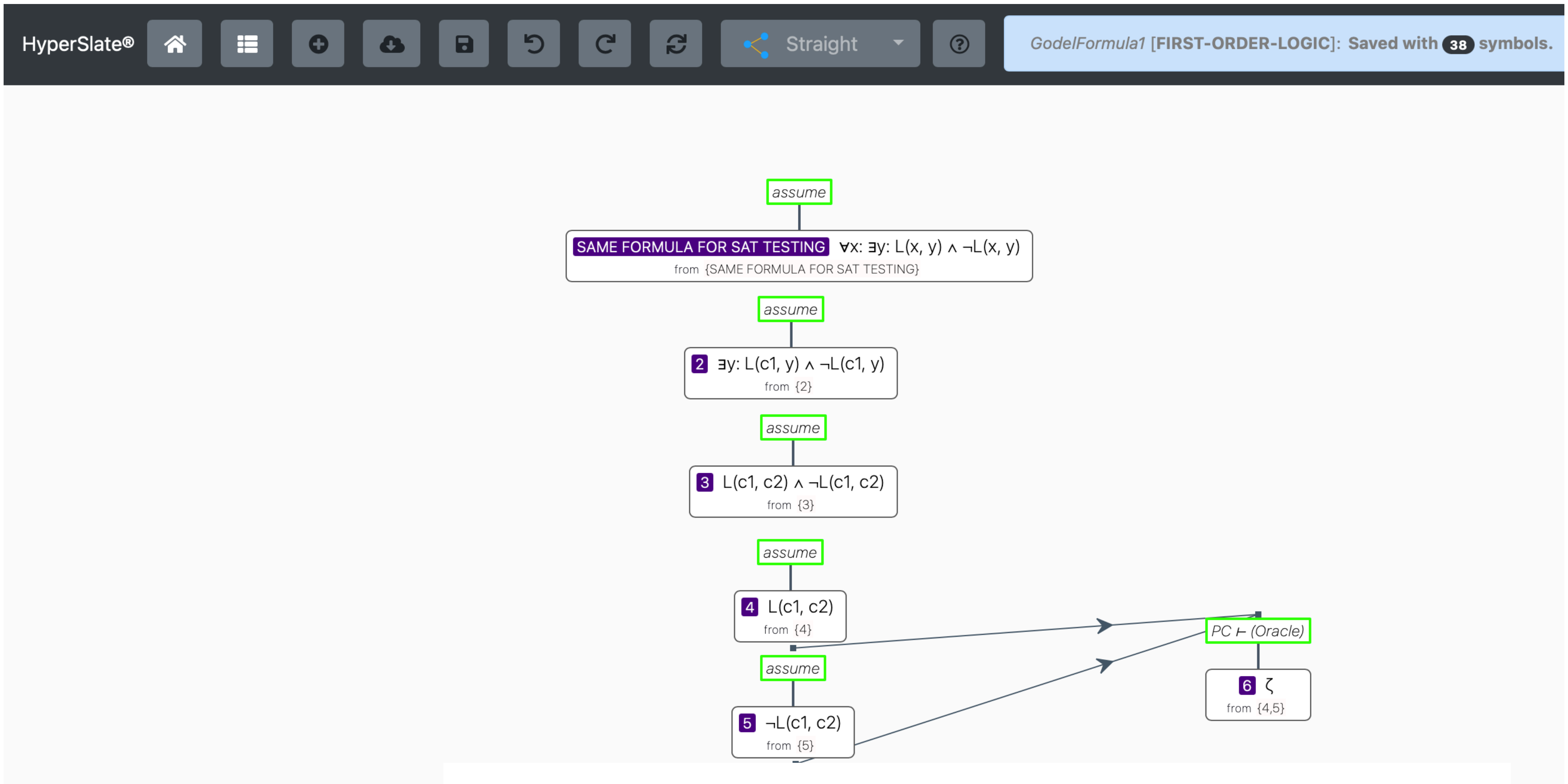
Making this Concrete Courtesy of HyperSlate®



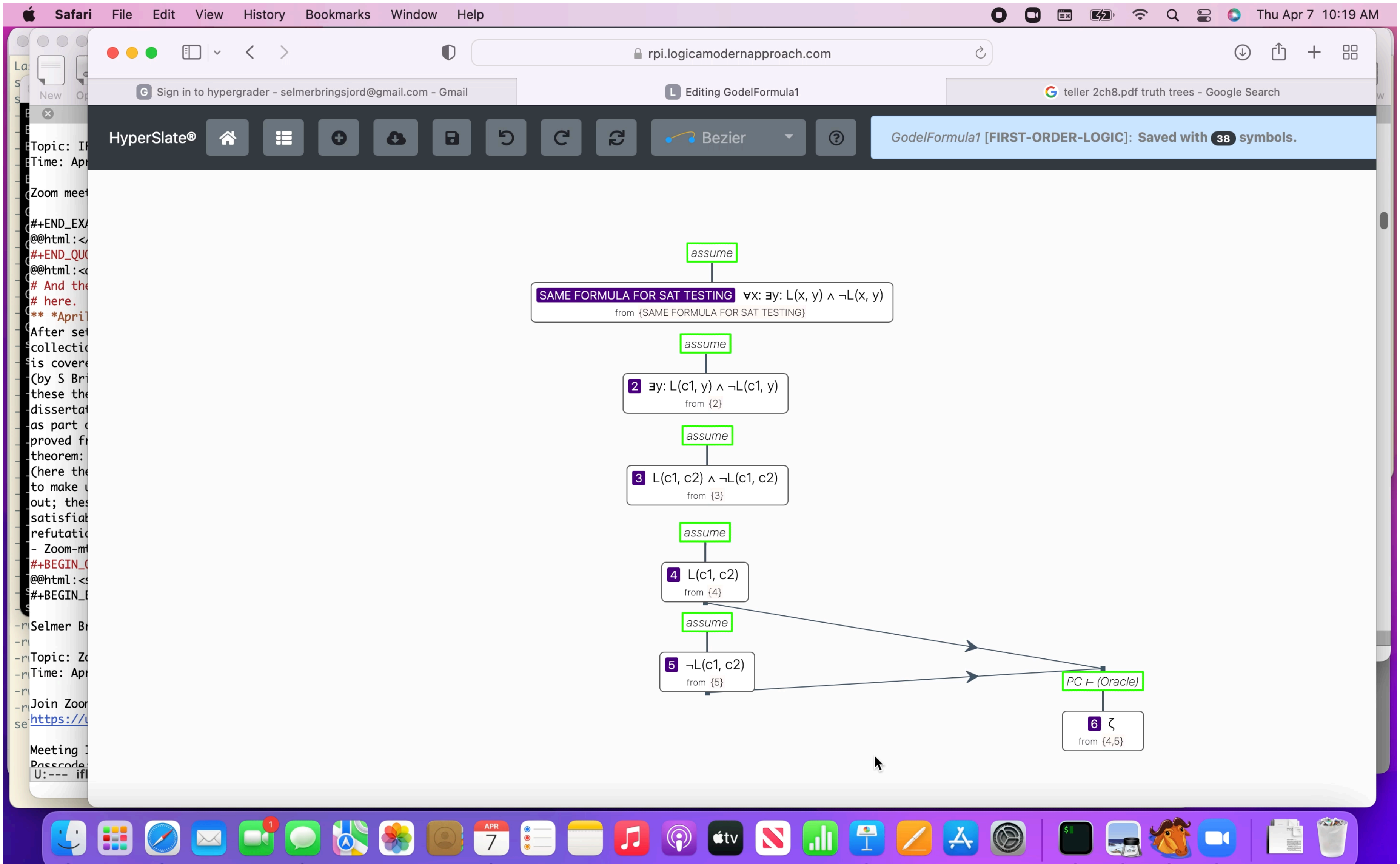
Making this Concrete Courtesy of HyperSlate®



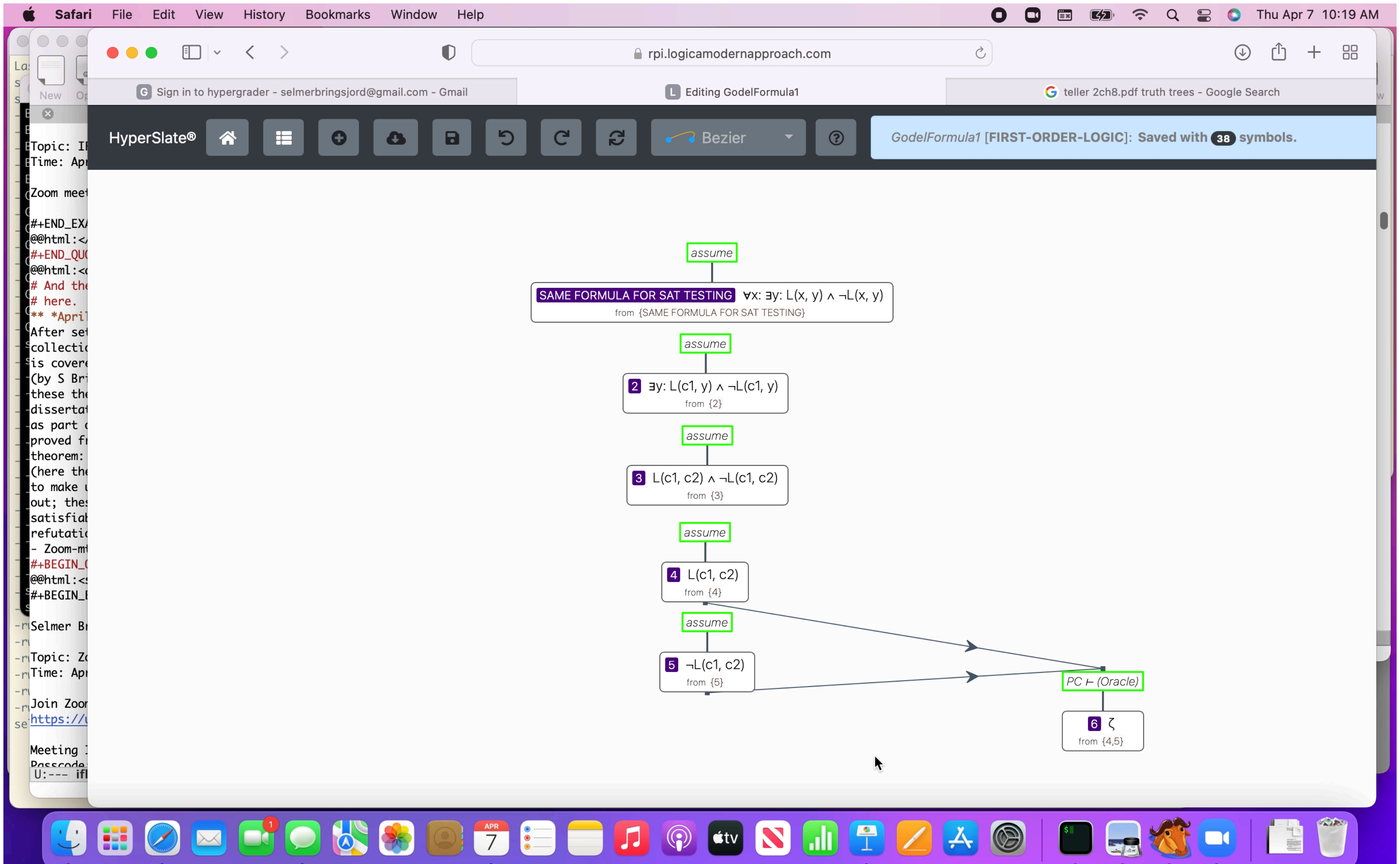
Making this Concrete Courtesy of HyperSlate®



Making this Concrete Courtesy of HyperSlate®



Making this Concrete Courtesy of HyperSlate®



slutten