The Propositional Calculus via *Logical Journey of the Zoombinis*

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 Logic-based AI
9/14/2026





Logic-and-AI in the news

• • •

The part of Navier-Stokes no one is talking about

Posted on **9 September 2026** by **John**

Yesterday OpenAI announced a proof that settled a long-standing question about the Navier-Stokes equations from fluid dynamics. The announcement has created a lot of buzz, as one would expect. But there's an aspect of OpenAI's work that I haven't seen anyone talk about: they posted a Lean 4 formal proof at the same time as their conventional human-readable proof.

Quite a few other mathematical conjectures have been settled recently using AI, and these have also been accompanied with formal proofs, using Lean 4 in particular.

Until very recently, generating machine-verifiable formal proofs has been **excruciatingly tedious**. In 2005, Henk Barendregt and Freek Wiedijk wrote

> To give an indication of how much work is needed for formalisation, we estimate that it takes approximately one work-week (five work-days of eight work-hours) to formalise one page from an undergraduate mathematics textbook.

The Propositional Calculus …

CHAPTER

LOGICAL

In which we design a
of inference to derive
tions to deduce what

Knowledge-based agents
Reasoning
Representation

Humans, it seems, kno
based agents use a pr
decide what actions to

The problem-solvi
inflexible sense. They
specific action from a s
agent doesn't know tha
An 8-puzzle agent doe
they have is very usefu

The atomic repres
a partially observable
representing what it kn
give a human the goal
that to a problem-solv
the 16,000 or so towns

Chapter 6 introduc
assignments of values
the agent to work in a
In this chapter, we take
general class of repres
bine and recombine in
needs of the moment–
lates the Earth's life e
of explicitly described
new knowledge about
updating the relevant k

We begin in Secti
new environment, the
agent without going in
in Section 7.3 and the
a factored representati
the canonical structure

KB is true in the real world? (After all, *KB* is just '
philosophical question about which many, many b
A simple answer is that the agent's sensors create t
world agent has a smell sensor. The agent program
is a smell. Then, whenever that sentence is in the k
Thus, the meaning and truth of percept sentences a
sentence construction that produce them. What ab
as its belief that wumpuses cause smells in adjace
tation of a single percept, but a general rule—deri
but not identical to a statement of that experience
a sentence construction process called **learning**, w
fallible. It could be the case that wumpuses cause s
which is when they take their baths. Thus, *KB* m
good learning procedures, there is reason for optin

7.4 Propositional Logic: A Very S

We now present **propositional logic**. We describe
its semantics (the way in which the truth of senten
a simple, syntactic algorithm for logical inferenc
entailment. Everything takes place, of course, in th

7.4.1 Syntax

The **syntax** of propositional logic defines the all
consist of a single **proposition symbol**. Each suc
be true or false. We use symbols that start with a
letters or subscripts, for example: P, Q, R, $W_{1,3}$
but are often chosen to have some mnemonic value
that the wumpus is in [1,3]. (Remember that syn
and 3 are not meaningful parts of the symbol.)
fixed meanings: *True* is the always-true propositio
Complex sentences are constructed from simpler
called **logical connectives**. There are five connecti

- $\neg$ (not). A sentence such as $\neg W_{1,3}$ is called th
  atomic sentence (a **positive literal**) or a nega
- $\wedge$ (and). A sentence whose main connective is
  tion; its parts are the **conjuncts**. (The $\wedge$ look
- $\vee$ (or). A sentence whose main connective is $\vee$
  tion; its parts are **disjuncts**—in this example
- $\Rightarrow$ (implies). A sentence such as $(W_{1,3} \wedge P_{3,1}) =$
  ditional). Its **premise** or **antecedent** is $(W_{1,3}$
  is $\neg W_{2,2}$. Implications are also known as **rul**
  symbol is sometimes written in other books a
- $\Leftrightarrow$ (if and only if). The sentence $W_{1,3} \Leftrightarrow \neg W_{2,2}$

Sentence	$\rightarrow$	*AtomicSentence* | *ComplexSentence*
AtomicSentence	$\rightarrow$	*True* | *False* | *P* | *Q* | *R* | ...
ComplexSentence	$\rightarrow$	(*Sentence*)
	|	$\neg$ *Sentence*
	|	*Sentence* $\wedge$ *Sentence*
	|	*Sentence* $\vee$ *Sentence*
	|	*Sentence* $\Rightarrow$ *Sentence*
	|	*Sentence* $\Leftrightarrow$ *Sentence*
OPERATOR PRECEDENCE	:	$\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

Figure 7.7 A BNF (Backus–Naur Form) grammar of sentences in propositional logic, along with operator precedences, from highest to lowest.

Figure 7.7 gives a formal grammar of propositional logic. (BNF notation is explained on page 1030.) The BNF grammar is augmented with an operator precedence list to remove ambiguity when multiple operators are used. The "not" operator ($\neg$) has the highest precedence, which means that in the sentence $\neg A \wedge B$ the $\neg$ binds most tightly, giving us the equivalent of $(\neg A) \wedge B$ rather than $\neg(A \wedge B)$. (The notation for ordinary arithmetic is the same: $-2+4$ is 2, not –6.) When appropriate, we also use parentheses and square brackets to clarify the intended sentence structure and improve readability.

7.4.2 Semantics

Having specified the syntax of propositional logic, we now specify its semantics. The semantics defines the rules for determining the truth of a sentence with respect to a particular model. In propositional logic, a model simply sets the **truth value**—*true* or *false*—for every proposition symbol. For example, if the sentences in the knowledge base make use of the proposition symbols $P_{1,2}$, $P_{2,2}$, and $P_{3,1}$, then one possible model is

Truth value

$$m_1 = \{P_{1,2}=false, P_{2,2}=false, P_{3,1}=true\}.$$

With three proposition symbols, there are $2^3=8$ possible models—exactly those depicted in Figure 7.5. Notice, however, that the models are purely mathematical objects with no necessary connection to wumpus worlds. $P_{1,2}$ is just a symbol; it might mean "there is a pit in [1,2]" or "I'm in Paris today and tomorrow."

The semantics for propositional logic must specify how to compute the truth value of *any* sentence, given a model. This is done recursively. All sentences are constructed from atomic sentences and the five connectives; therefore, we need to specify how to compute the truth of atomic sentences and how to compute the truth of sentences formed with each of the five connectives. Atomic sentences are easy:

- *True* is true in every model and *False* is false in every model.
- The truth value of every other proposition symbol must be specified directly in the model. For example, in the model m_1 given earlier, $P_{1,2}$ is false.

Zoombinis! …

Palette for Prop-Calc Binary Vectors



A Prop-Calc Binary Vector, Built



The Formal Language of the Propositional Calculus

Syntax	**Formula Type**	**Sample Representation**
$P, P_1, P_2, Q, Q_1, \ldots$	Atomic Formulas	"Larry is lucky." as L_l
$\neg\phi$	Negation	"Gary isn't lucky." as $\neg L_g$
$\phi_1 \wedge \ldots \wedge \phi_n$	Conjunction	"Both Larry and Carl are lucky." as $L_l \wedge L_c$
$\phi_1 \vee \ldots \vee \phi_n$	Disjunction	"Either Billy is lucky or Alvin is." as $L_b \vee L_a$
$\phi \rightarrow \psi$	Conditional (Implication)	"If Ron is lucky, so is Frank." as $L_r \rightarrow L_f$
$\phi \leftrightarrow \psi$	Biconditional (Coimplication)	"Tim is lucky if and only if Kim is." as $L_t \leftrightarrow L_k$

Table 2.1: Syntax of the Propositional Calculus. Note that ϕ, ψ, and ϕ_i stand for arbitrary formulas.

The Formal Language

(presented as formal grammar)

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

$$
\mathit{AtomicFormula} \quad \Rightarrow \quad \mathtt{P}_1 \mid \mathtt{P}_2 \mid \mathtt{P}_3 \mid \ldots
$$

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

As S-expressions

$$
\begin{array}{lcl}
\mathit{Formula} & \Rightarrow & \mathit{AtomicFormula} \\
 & | & (\mathit{Formula}\ \mathit{Connective}\ \mathit{Formula}) \\
 & | & \neg\ \mathit{Formula} \\
\hline
\mathit{AtomicFormula} & \Rightarrow & \mathtt{P}_1 \mid \mathtt{P}_2 \mid \mathtt{P}_3 \mid \ldots \\
\hline
\mathit{Connective} & \Rightarrow & \wedge \mid \vee \mid \rightarrow \mid \leftrightarrow
\end{array}
$$

```
p   P   bradywillbeback  P26   ...

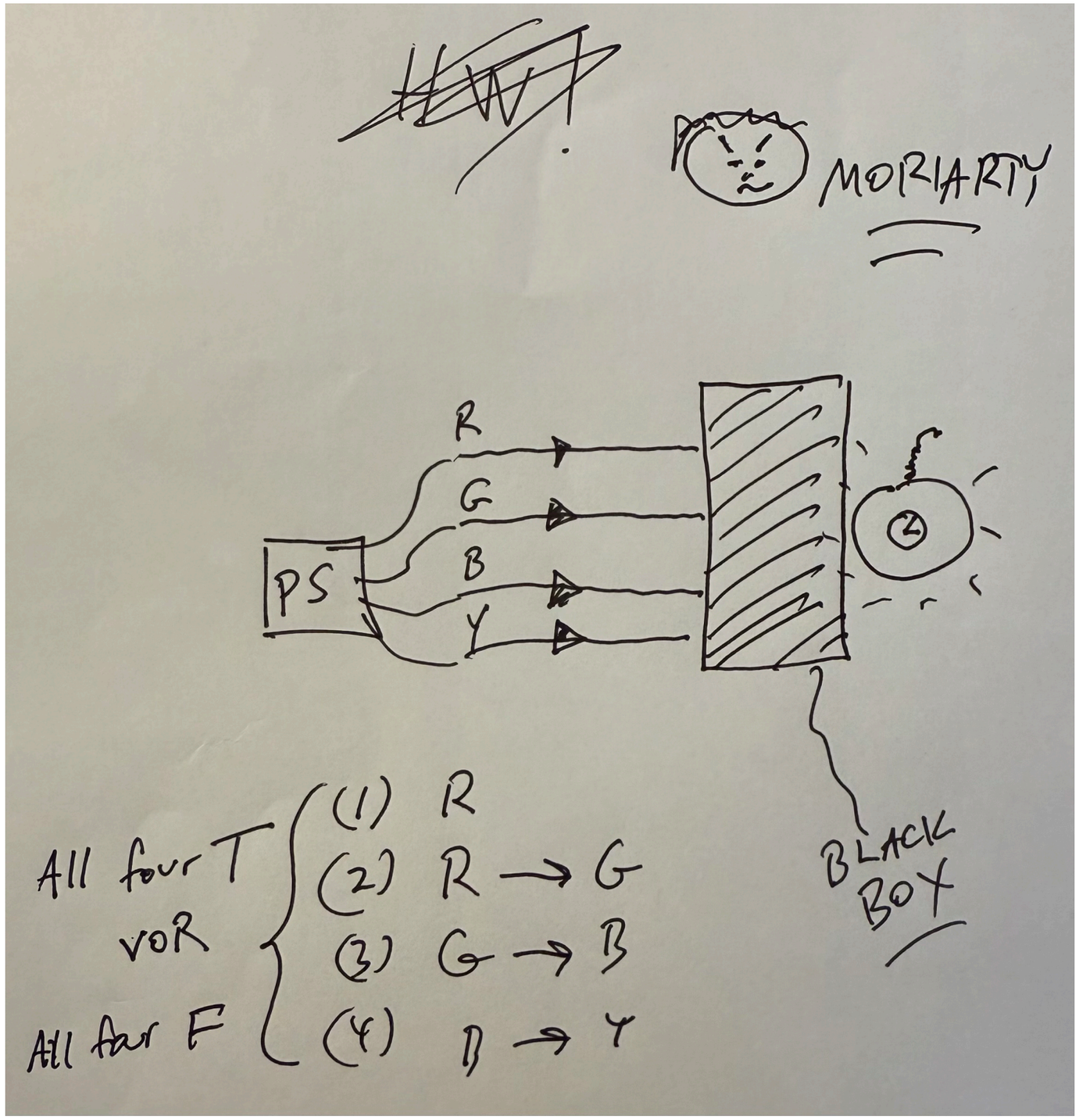
(not p)  (not P)  (not P26)    ...

(and P Q)  (or P Q) (if P Q) (iff P Q)
```

Moriarty Bomb

Challenge #2 …

Do in HyperSlate® …



More *Expressive* Formal Language: Pure Predicate Calculus …

$\mathscr{L}_0$



Better Formal Language: Pure Predicate Calculus (presented via formal grammar)

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

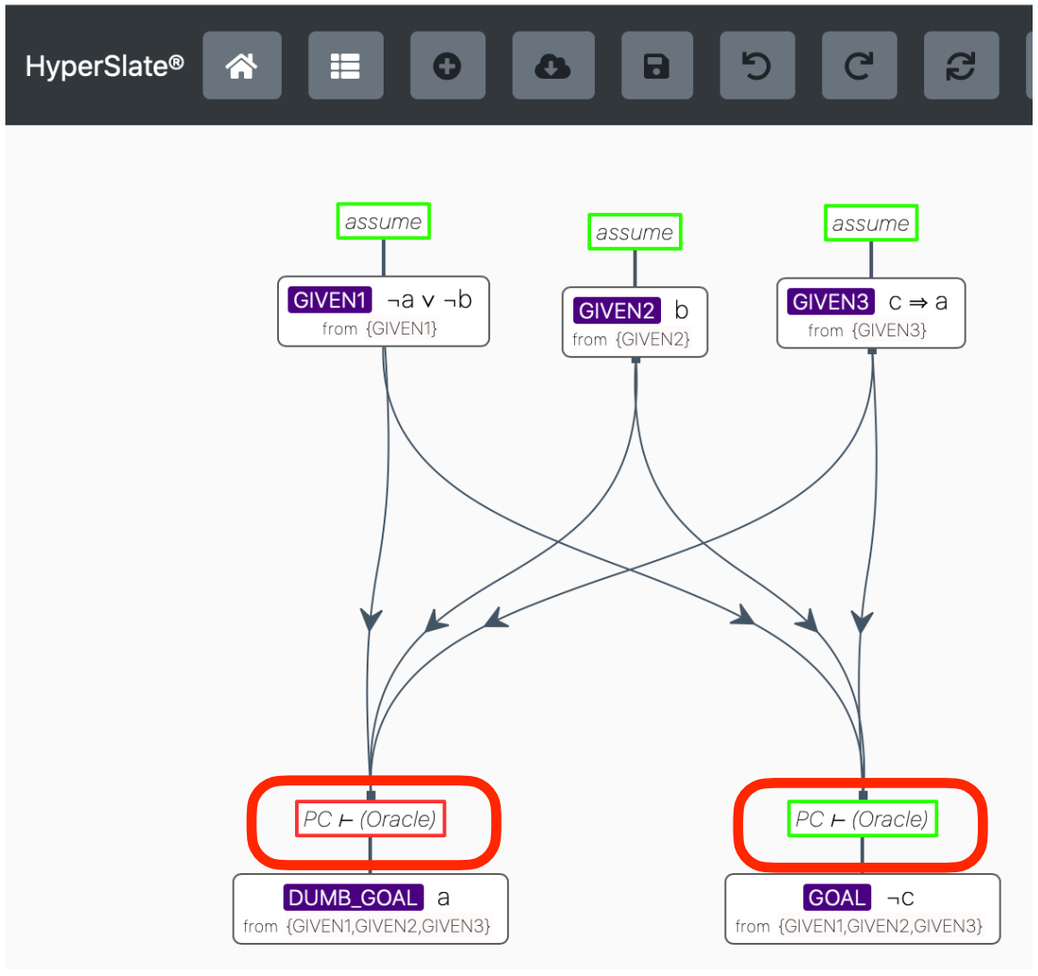
$$
\mathit{AtomicFormula} \quad \Rightarrow \quad (\mathit{Predicate}\ \mathit{Term}_1 \ldots \mathit{Term}_k)
$$

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

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

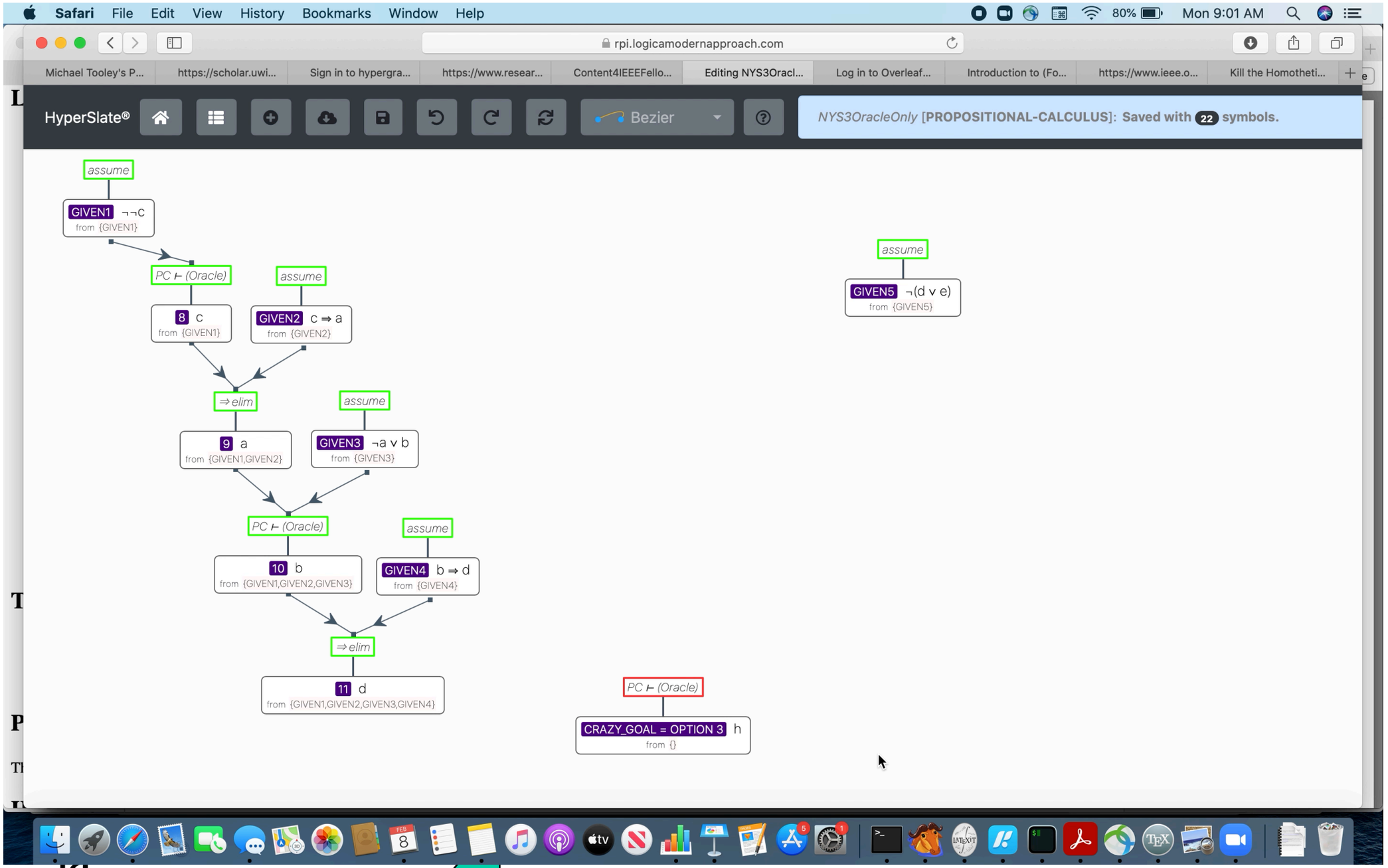
$$
\begin{array}{ll}
\mathit{Predicate} & \Rightarrow\ P_1 \mid P_2 \mid P_3\ \ldots \\
\mathit{Constant} & \Rightarrow\ c_1 \mid c_2 \mid c_3\ \ldots \\
\mathit{Variable} & \Rightarrow\ v_1 \mid v_2 \mid v_3\ \ldots \\
\mathit{Function} & \Rightarrow\ f_1 \mid f_2 \mid f_3\ \ldots
\end{array}
$$

The PC (Provability) Oracle



Ok, to HyperSlate® for some exploration …

The Infamous “NYS 3”

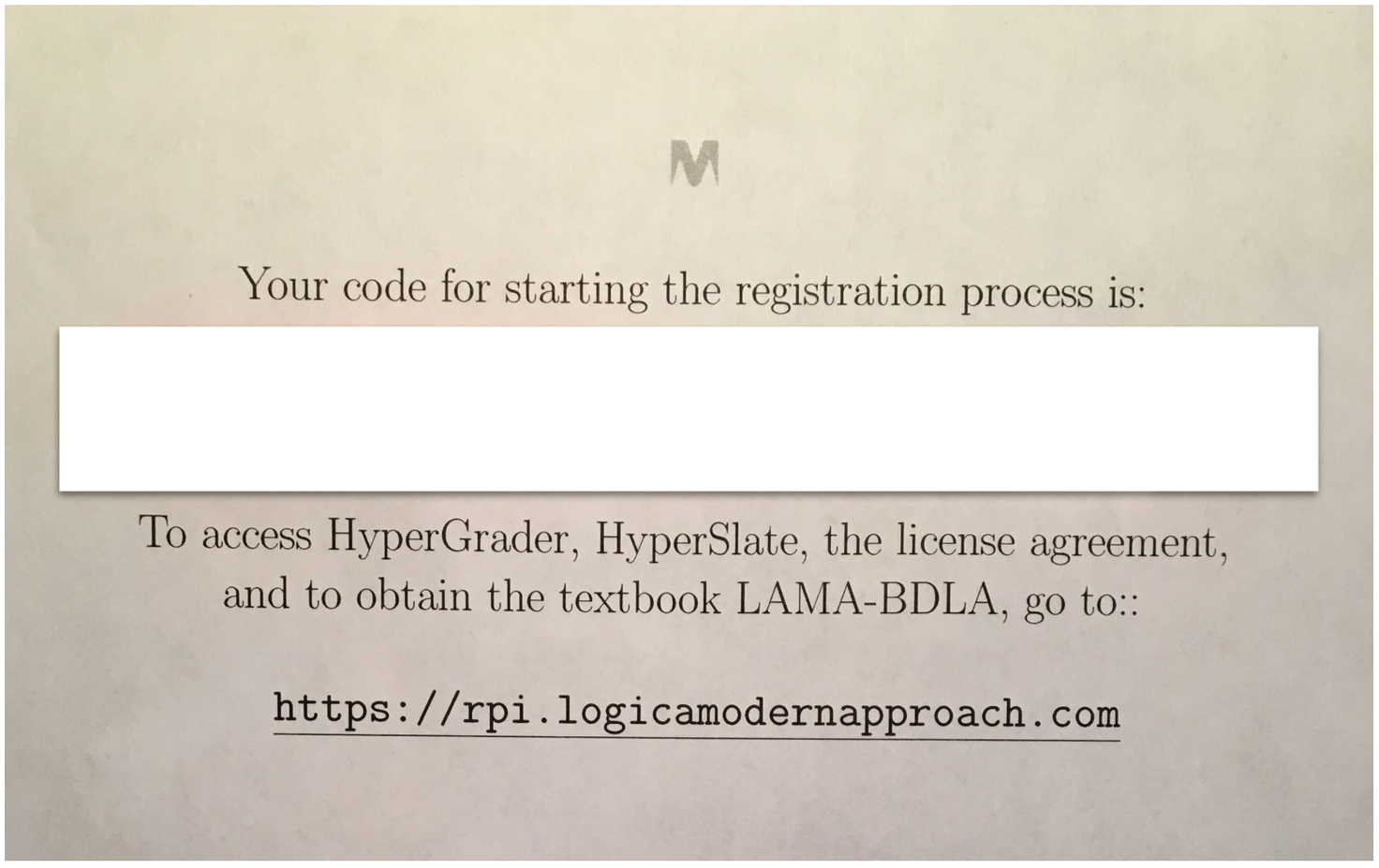


all of the above

Again,

Logistics for Registration etc …

The Starting Code to Purchase in Bookstore


M

Your code for starting the registration process is:

To access HyperGrader, HyperSlate, the license agreement, and to obtain the textbook LAMA-BDLA, go to::

https://rpi.logicamodernapproach.com

Once seal broken on envelope, no return. Remember from first class, any reservations, opt out of ILBAI!

The email address you enter is case-sensitive!

Your OS and browser must be fully up-to-date; Chrome is the best choice, browser-wise (though I use Safari :)).

Watch that the link emailed to you doesn't end up being classified as spam.

Now for live tutorial etc on this …

Kunstig intelligens bør være logisk!