Philosophica 61 (1998, 1) pp. 51-76

IS GÖDELIAN MODEL-BASED DEDUCTIVE REASONING COMPUTATIONAL?

Selmer Bringsjord

0. Introduction

Model-based reasoning (MBR), including specifically the deductive variety on which I focus herein (MBR_D), is promising in large part because of the intellectual bravery it reflects. The deductive reasoning carried out by today's automated theorem provers is one-dimensional: such reasoning can be reduced to standard linear and linguistic chains of inference, and such chains, as well as the manipulation thereof, can be wholly captured by computation. In short, machine deduction, as the century turns, is mere symbol manipulation.

But MBR_D moves beyond symbols: MBR_D embraces the likes of diagrams and mental images, despite the dizzying possibility that such things, and the manipulation of such things, may be beyond computation. In this paper I share some thoughts about the sort of deductive model-based reasoning that is part and parcel of our establishing Gödel's first incompleteness theorem. I take the Gödelian case to be paradigmatic; the idea is to generalize from it to MBR_D. More exactly, I am specifically concerned with these two questions, where the second is a generalization of the first:

Q1 Could a computing machine ever prove Gödel's first incompleteness theorem (= hereafter just Gödel I) in the model-based manner human logicians and mathematicians prove it?

Q1* Could a computing machine ever prove theorems in the model-based manner human logicians and mathematicians prove them?

While I can't prove that a negative answer to Q1 and Q1* is correct, that is indeed my suspicion, and I will try to justify my attitude in what follows. If I'm wrong, if computers *can* prove things in the model-based manner of Gödel and other mathematicians and logicians, then this paper should serve as a productive challenge to researchers determined to reduce MBR to computational structures.[1]

My plan is as follows. In section 1 I introduce Gödel I. In section 2 I explain Gödel I in model-based fashion, and conclude with some model-based deduction that is part of a full-blown model-based proof of Gödel I. In section 3 I generalize a bit on the strength of section 2: I set out a partial, circumspect account of MBR_D. Next, in section 4, I show that machine-generated proofs of Gödel I (carried out by the system known as OTTER) fail to capture the kind of model-based deductive reasoning seen in the previous two sections.

In section 5 I adapt John Searle's Chinese Room Argument to show that machine-based proofs of Gödel I, and theorems generally, because such proofs consist in the mere manipulation symbols without ever achieving genuine understanding, cannot be model-based deduction. Finally, in section 6, I rebut the objection that the diagrammatic reasoning concretized by Barwise and Etchemendy in their Hyperproof system (Barwise and Etchemendy, 1994) indicates that MBR_D-based proofs of Gödel I *can* be captured computationally.

This paper is intended to be accessible to students and researchers in and interested in MBR, many of whom, of course, aren't logicians or

---

[1] Gödel I has of course given rise to a rather well-known question with which I'm *not* concerned herein, viz.,

Q2 Do the mathematical facts revealed by Gödel I imply that people have an ability that can never be matched by machines?

Rather a lot of ink has been devoted to Q2 of late (some of it flowing from my own pen). Roger Penrose (Penrose, 1989, 1994), for example, has famously argued that Q2 should be answered in the affirmative. Though I've been elsewhere concerned with Q2 (the bulk of my own writing on Q2 can be found in the chapter "Gödel" (Bringsjord, 1992)), this question is not my concern in the present paper. I'm not concerned here with whether Gödel I itself implies that minds aren't machines; I'm concerned herein, if you will, with whether the human model-based reasoning that goes into proving Gödel I is beyond computation.

mathematicians. I presuppose only a rudimentary understanding of first-order logic (and even in connection with FOL there is a short review). As a result, this paper amounts to a readable introduction to Gödel I. One final thing before we embark: I don't *prove* Gödel I in model-based fashion below. Such a proof would make this paper inaccessible to most. Instead, I work a serviceable compromise: I *explain* Gödel I in model-based fashion, and offer *some* of the model-based proof.[2] The explanation shares with a full-blown model-based proof of this theorem sufficient model-based elements to enable a fruitful investigation of Q1 and Q1*.

1. What Does Gödel's Incompleteness Theorem Say?

If you are to use first-order logic to represent some declarative information, you must settle on your domain, and on some set of key symbols, that is, your relation symbols (which denote relations or properties), function symbols (for denoting functions), and constants (which are like names; they pick out individual objects directly). For example, if your task is to represent romantic facts about the domain of people, including, specifically, Alice and Bertrand, and the fathers of both of them, you might decide to use

- the relation symbol L for 'loves', so that Lxy indicates that x loves y;
- the constants a and b to refer to Alice and Bertrand, respectively;
- the function symbol f to denote the father-of function.[3]

Given this symbol set, $\{L, f, a, b\}$, you are able to pick out individual things in the domain in question by way of what are called *terms* (e.g., a is a term, as are: $f(a)$, $f(f(a))$, b, $f(b)$), and you can create *formulas* to say such things as the following:

---

[2] Note as well that I don't focus on Gödel's original proof of his first incompleteness theorem. Such a focus would entirely preclude the accessibility of this essay.

[3] To ease exposition, I ignore the fact that father-of isn't really a function.

English	*Formulas in FOL*
Alice loves Bertrand.	Lab
Alice loves Bertrand's father.	$Laf(b)$
Alice loves everyone.	$\forall x\, Lax$
Bertrand loves those who love Alice's father.	$\forall x(Lxf(a) \rightarrow Lbx)$
People who love themselves are not loved by Alice	$\forall x(Lxx \rightarrow \neg Lax)$
Someone is loved by both Alice and Bertrand.	$\exists x(Lax \wedge Lbx)$

In the case of Gödel I, the domain is the natural numbers $\mathbf{N} = \{0, 1, 2, \ldots\}$, and the symbol set in question is one configured for simple arithmetic, viz., $= \{+, 0, 1, <\}$. With this symbol set we can make assertions about arithmetic, such as that "Every number multiplied by one returns itself" ($\forall x\ x \cdot 1 = x$), and "There is no greatest number" ($\neg \exists x \forall y\, y < x$). What Gödel astonishingly showed is that given a set Φ of formulas about arithmetic that meets three particular conditions, there is a formula about arithmetic, ϕ_g, which is such that neither it nor its negation can be proved from Φ. The three particular conditions are that Φ must be (i) consistent (i.e., for now, no contradiction can be derived from Φ), (ii) decidable (which, for now, amounts to: an ordinary computer program exists which decides, for a formula ψ, whether or not $\psi \in \Phi$), and (iii) representable (or just "Rep" for short). These three conditions are explained in model-based fashion in the next section. For now, you can rely on your intuitive understanding of the parenthetical explanations of (i) and (ii), and (iii) may be provisionally understood to mean that Φ can be used to perfectly model the operation of any ordinary computer program. So we have:

> *Theorem 1 (Gödel I)* Suppose that Φ is consistent and decidable, and that Rep Φ as well. Then there is an S_{ar}-sentence ϕ_g such that neither $\Phi \vdash \phi_g$ nor $\Phi \vdash \neg\phi_g$.

2. A Model-Based Explanation of Gödel's Incompleteness Theorem

I'm willing to bet that for the newly initiated, Gödel I as just presented is rather difficult to assimilate. A model-based explanation can change

that - but in order to give such an explanation of Gödel I, we need to first assimilate model-based versions of the "building block" concepts of

- Turing machines
- Gödel numbering
- consistency and inconsistency
- decidability
- representability

Once these are digested, Gödel I can be rather easily explained and grasped by all sufficiently motivated readers.

2.1 Simian Machines: Model-Based Turing Machines

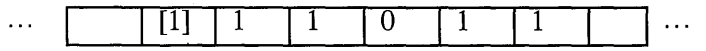
We begin by using some mental imagery to characterize the first core concept in Gödel I: Turing machine computation. Imagine a monkey who controls a boxcar on railroad track that stretches infinitely in both directions across an infinite, barren plain. There are no other devices on this track. The boxcar works by way of cogs and levers; it's very primitive. The *state* of the boxcar is always exactly one from a finite number of states q_1, q_2,..., q_n. The railroad track is divided into squares; and each square is filled with a blackboard. Upon the blackboard can be written any one *symbol* a_i from some finite *alphabet* $\Sigma = \{a_1, a_2,\ldots,a_n\}$. (For the moment, assume that the alphabet is {0, 1}). The monkey has been trained to follow a kind of simple instruction that never varies in form. This type of instruction requires that he be able to move the boxcar one square to the right or left (by recognizing and acting on *L* for "move left" and *R* for "move right"), write or erase symbols on the blackboard below the boxcar, and place that boxcar into a state q_i. The instructions always come in the form of quadruples. An example would be

$$q_3 1 L q_2$$

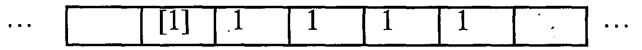
which says to the monkey: "If your boxcar is in state q_3 positioned over a blackboard on which is written symbol 1, then move left and throw the boxcar into state q_2".

If you have a clear image of the simian machine just described, you have nearly everything you need to grasp the mathematical essence of

Turing machine computation. (Our monkey is really no different than the "computists" Turing (Turing, 1936) used to anchor his seminal mathematization of digital computation.) All that you're missing is a consistent way to position input for the monkey and to receive its output. Accordingly, imagine that input to the monkey is always given in the form of a contiguous string of 0's and 1's upon the track, with his boxcar positioned over the leftmost square of this input. For example, here is what some particular input might look like:



If we wanted the simian machine to implement ordinary addition, then the desired output, to be left after the monkey is finished working (with his boxcar atop the leftmost symbol), would be:



I leave it to readers to devise a sequence of quadruples that would implement addition. Suffice it to say that everything a modern, high-speed digital computer can do, and everything a Turing machine (or Register machine or cellular automaton, etc.) can do, a simian machine can do.

2.2 A Model-Based Gödel Numbering Scheme

Now for a model-based explanation of the concept of Gödel numbering. The basic idea here is to regiment a scheme for associating a natural number with a first-order formula, and *vice versa* - where this association must be completely mechanical. Sometimes this ingenious concept is presented in impenetrable ways, but with help from our simian machines, and simple diagrammatic aids (viz., tables), the concept becomes transparent[4].

To begin, recall that a particular use of first-order logic is based on the selection of constants, and function and relation symbols. The set S_{ar}

[4] The tables used here are pulled from Boolos and Jeffrey (1989).

$= = \{+, \times, 0, 1, <\}$, central to Gödel I, contains two (binary) function symbols, two constants, and one (binary, or 2-place) relation symbol, respectively. But what about the general case? If we leave aside constants, and we write f_n^m for a the mth n-ary function symbol, and R_n^m for the mth n-ary relation symbol, then all first-order formulas are built by concatenating symbols from this (infinite) table.

(	)	$\wedge$	$\exists$	x_0	f_0^0	f_0^1	...	R_0^0	R_0^1	R_0^2	...
		$\vee$	$\forall$	x_1	f_1^0	f_1^1	...	R_1^0	R_1^1	R_1^2	...
		$\neg$		x_2	f_2^0	f_2^1	...	R_2^0	R_2^1	R_2^2	...
		$\leftrightarrow$		.	.	.		.	.	.	
		$\rightarrow$		.	.	.		.	.	.	
				.	.	.		.	.	.	

Next, consider this corresponding (infinite) table, which has a structure isomorphic to the one just given.

1	2	3	4	5	6	68	...	7	78	7888	...
	29	39	49	59	69	689	...	79	789	7889	...
		399		599	699	6899	...	799	7899	78899	...
		3999		.	.	.		.	.	.	
		39999		.	.	.		.	.	.	
				.	.	.		.	.	.	

Now, recall our monkey and simian machines; fix again the relevant mental images. It's a trivial matter to have our monkey obtain the Gödel number of a given formula (or to have him work in the opposite direction), using the the two tables. An example should make this clear. Consider the formula

$$\psi = \forall x_0 \exists x_1 R_2^2 x_1 x_0$$

This formula has a (unique) Gödel number of

49545978899595.

In order to produce this result, visualize our monkey simply matching up the symbols ψ with their corresponding numbers in the second table,

writing these numbers down, and continuing, left to right. Of course, the monkey could also work in the opposite direction: from number to corresponding formula.

Finally, some simple notation that will prove crucial below: Where ϕ is a formula, let n^{ϕ} denote the Gödel number of ϕ.

2.3 Consistency in Model-Based Terms

Normally, to say that a set Φ of first-order formulas is consistent is to say that from this set no contradiction can be derived using a fixed set of inference rules that produces a sequence of ordinary symbols. There is an easier, model-based way: in order to show that a set Φ is consistent it suffices to build a visual model in which all of the members of this set are true. Consider the following example (from Bertrand Russell): "In a certain village in England, there was a barber who claimed to shave all and only those men who did not shave themselves". We can represent the relevant facts here in first-order logic by these two assertions:

- $\forall x(Mx \vee Wx)$("Everyone in the village is either a man or a woman").
- $\forall x(Mx \rightarrow (Sbx \leftrightarrow \neg Sxx))$ ("All men are such that the barber shaves them if and only if they don't shave themselves").

And we can use a diagrammatic representation to specify the visual model in which these two assertions are both true. For this representation, let circles denote males, of which we suppose there are four in the village, and let squares denote females, of which there are also four, one of them being the barber. A *Y* indicates that one person does shave another, an *N* indicates that one person doesn't shave another, and a blank indicates that the information is unknown. Then here is a table that shows the scenario described above is consistent.

	$\bigcirc_1$	$\bigcirc_2$	$\bigcirc_3$	$\bigcirc_4$	$\square_1$	$\square_2$	$\square_3$	$\square_b$
$\bigcirc_1$	N							
$\bigcirc_2$		Y						
$\bigcirc_3$			Y					
$\bigcirc_4$				N				
$\square_1$								
$\square_2$								
$\square_3$								
$\square_b$	Y	N	N	Y				

To say, as Gödel I does, that a set Φ of first-order formulas about arithmetic is consistent, is to say that we can write down a (very large!) series of tables like this in which all of the assertions in Φ are true.

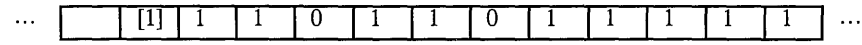
2.4 Decidability in Model-Based Terms

Providing a model-based characterization of decidability is trivial: Imagine that the monkey in simian machines has two new special symbols **Y** and **N** at his disposal. His job is to decide whether or not a string of symbols given him as input is or is not in a certain set. With respect to Gödel I, the monkey will work as follows. He will be given a string of symbols that compose a first-order formula ϕ, as well as the set Φ referred to in Gödel I. If instructions (in the quadruple format presented above, of course) can be given to him in order to enable him to produce **Y** as output if and only if $\phi \in \Phi$, and N as output if and only if $\phi \notin \Phi$, then Φ is decidable. It's as simple as that.

2.5 A Model-Based Account of Representability

Certain relations and functions on the natural numbers are within the power of our simian machines. A function from $\mathbf{N}^n$ to **N** is within the power of these machines if such a machine can perfectly model the input-output behavior of the function. For example, consider multiplication, $\times$; this function can indeed be perfectly captured by our monkey. (Diligent readers are encouraged to devise a collection of quadruples that will instruct the monkey accordingly). What about a *relation* on **N**? (An *n*-ary relation on **N** is a subset of the *n*-fold Cartesian product $\mathbf{N} \times \mathbf{N} \times \ldots \times$

N). What does it mean to say that such a thing is within the power of a simian machine? The idea is very simple, and harkens back to the two special symbols **Y** and **N** and the related concepts of decidability we have already characterized. Consider the ternary relation of "between" in the natural numbers. Imagine that our monkey is given as input the three numbers 3, 2, and 5, each of them represented as a string of 1's separated by a 0; this, then, is how the track of blackboards would look initially:



The question here is: Is the first of the numbers in this input string between the second and third? Given the input here, the answer will be given by the following tape upon completion:

But in Gödel I's hypothesis, representability is ascribed to the set Φ. What does this mean? What does it mean to say Rep Φ? It means that for every n-ary relation R and function f on N that can be handled by a simian machine, there is an arithmetic formula ϕ that "mirrors" this relation and function, and that formula can be proved from Φ. For example, and a bit more precisely: if $(n_1, n_2, \ldots, n_k) \in R$, and R is a relation that a simian machine can decide, then you can substitute constants that denote these numbers into ϕ, and ϕ can be proved from Φ.

2.6 Toward a Model-Based Proof of Gödel I

Given the foregoing, we can re-express Gödel I in model-based terms that make the theorem much easier to understand. (Readers are encouraged to return to the presentation of the theorem above, and to read it again, but this time with the model-based explanations firmly in mind ... Back? *Now* the theorem should make some real sense to you, even if you're new to it). But what about *proving* Gödel I? Well, as I said earlier, proving Gödel I herein would make this paper too long, and too technical. But rest assured that a model-based proof *can* be given; it's how I and others

across the globe teach Gödel I (and, for that matter, Gödel's second incompleteness theorem as well), and it's how Gödel himself pondered and eventually established his famous result. (Readers interested in the "model-based" mind of Gödel himself are encouraged to read (Wang, 1995) for a marvelous introduction, delivered by Gödel's friend Hao Wang). Let me give a brief sample that trades on the model-based elements introduced above.

Chapter 51 of the second book of Cervantes' immortal (Cervantes, 1999) is Gödel's key move in a nutshell. Sancho Panza is governor of an island and must preside as judge over some rather tough cases, one of which is presented to him as follows:

> My lord [Sancho Panza], a broad river separates the two parts of a single domain [...]. Now, there's a bridge over this river, and at one end there stands a gallows and a court building, in which four judges usually preside, applying the law formulated by the lord of this river, this bridge, and this entire realm, which runs as follows: "Anyone passing over this bridge, from one section of this domain to the other, must first declare under oath where he is coming from and where he is going, and if he swears truly, he shall be allowed to pass, but if he lies, he shall be hanged from the gallows standing nearby, without any appeal or reprieve allowed". [...] Well, it happened, one day, that a man came and swore the required oath, saying among other things that he had come to be hanged on that gallows, and for no other purpose. The judges considered his oath, saying: "If we simply let this man cross the bridge, his oath will be a lie, and then, according to the law, he ought to die, but if we hang him, the oath he swore about being hanged on this gallows will be true, and then the same law decrees that he be allowed to cross over in peace". Please consider, my lord governor, your grace, what the judges should do with this fellow, for even now they remain anxious and unsure how to proceed, and, having been made aware of your grace's keen mind and sublime understanding, they have sent me to implore your grace to tell them how you view this singularly complicated and puzzling case (Cervantes, 1999, p. 629).

Now I know that you, reader, are blessed with wisdom well beyond what Sancho Panza and even his noble friend Don Quijote wielded, but I doubt that you can fare any better than Sancho in judging the following case, which I unblushingly confess has driven my brain dizzy, with no verdict

forming therein, let alone forthcoming. Of course, this is the famous Liar Paradox in narrative form. Gödel found a way to turn it to his advantage.

A streamlined version of the paradox runs as follows. Is the following sentence true or false?

This sentence is false.

If this sentence is true, then since it says it's false, it *is* false. If, on the other hand, this sentence is false, then since that's what it says, it's true. So it's true if and only if it's false, an outright contradiction.

All this is perhaps interesting, but how does the paradox help establish anything relevant to Gödel I? And how is it to work in model-based fashion? Well, here's a way to make the connection[5]: Return to our monkey, and recall that he leaves output for us on the railroad tape when he is finished. Let us imagine that the monkey can thus be said to **print** things for us. To simplify things, let's give the monkey the alphabet

$$\neg \; P \; M \; 1 \; 0$$

with which to work (rather than S_{ar} from above), and let's set up a streamlined system for the monkey to obtain Gödel numbers, and to move from such numbers to formulas. Specifically, let's have him identify natural numbers with their correlates in binary notation through the following table.

¬	P	N	1	0
10	100	1000	10000	100000

Let's say the *norm* of an expression is that expression followed by the Gödel number of that expression obtained by the monkey.

So sticking with our example, the norm of $P\ N\ N\ P$ is $P\ N\ N\ P$ 10010001000100. Next, let's stipulate that a formula is an expression having one of the four forms $P\ X$, $P\ N\ X$, $\neg\ P\ X$, and $\neg P\ N\ X$, where

[5] The following scheme is adapted from (Smullyan, 1992).

X is any number in binary. We also say the following.

- $P\ X$ is true if and only if X is the Gödel number of an expression that the monkey can print.
- $P\ N\ X$ is true if and only if X is the Gödel number of an expression whose norm is printable by the monkey.
- $\neg\ P\ X$ is true if and only if $P\ X$ is not true.
- $\neg P\ N\ X$ is true if and only if $P\ N\ X$ is not true.

If we assume that our monkey never prints a false sentence, can we find a true sentence that he cannot print? If so, we will have found something perfectly analogous to the formula ϕ_g in Gödel I. There is such a sentence: $\neg PN101001000$. Here's the "Lar-like" proof: We know that $\neg PN101001000$ is true if and only if $PN101001000$ is not true, i.e., if and only if 101001000 is the Gödel number of an expression whose norm is not printable by the monkey. But 101001000 is the Gödel number of $\neg PN$, and the norm of $\neg PN$ is $\neg$PN101001000 itself! So $\neg$PN101001000 is true if and only if it's not printable by the monkey. This implies that either $\neg$PN101001000 is true and not printable, or is printable and not true. Since by hypothesis the monkey never prints out untrue formulas, the formula is true and not printable.

What about more precise uses of the Liar? I end this section by giving a more careful version of the core reasoning behind Gödel I.

First, recall some simple propositional logic: If you can prove a biconditional expression p if and only if q $(= p \leftrightarrow q)$, and you can prove p, you can chain from left to right across the biconditional to obtain q. And if you have proved a biconditional p if and only if q, as well as that p is false, i.e., $\neg\ p$, you can safely infer to $\neg q$.

Now, assume that we have previously proved that for every property F that can be expressed of a constant in first-order logic, there is a sentence ϕ that is true exactly when the Gödel number of ϕ has F, and that this fact can be proved from the set Φ referenced in the hypothesis of Gödel I. (Remember that the Gödel number of ϕ is written n^{ϕ}). That is,

(1) $\Phi \vdash \phi \leftrightarrow Fn^{\phi}$

Now let F be the property "is provable from Φ", denoted by F^{p}. So we

can move to

(2) $\Phi \vdash \phi \leftrightarrow \neg F^p n^\phi$

Now, suppose that ϕ is provable from Φ, i.e., $\Phi \vdash \phi$. Then from this together with (2) it follows that $\Phi \vdash \neg F^p n^\phi$, which is to say that ϕ is *not* provable - contradiction. Suppose on the other hand that $\neg\phi$ is provable from Φ; then from this and (2) it follows that $\Phi \vdash F^p n^\phi$, which is to say that ϕ *is* provable from Φ - contradiction again! So it follows that ϕ is not provable from Φ, and neither is $\neg\phi$ provable.

2.7 A Key Remaining Question

However, a key question remains: What, exactly, is distinctive about a model-based explanation and proof of Gödel I? If we can answer this question, then we can move to consider, as promised, machine proofs of Gödel I, with an eye to ascertaining whether or not such proofs are, or can be, model-based in nature. So, what *is* special about a model-based approach to Gödel I? I now turn to this question.

3. Characterizing Model-Based Deductive Reasoning

I take it that the central locution to be characterized is "agent *S* proves *P* in model-based fashion", where *S* refers to any intelligent agent, human or otherwise, and *P* is expressed in the language (call it "L") of mathematics that marries various formal languages (e.g., first-order logic) with natural languages (e.g., English and Italian).

Obviously, neither "intelligent agent" nor L admit of complete formalization. Artificial Intelligence (AI) and Cognitive Science (CogSci) take the notion of an intelligent agent to amount to nothing more than an entity which, upon taking in input from the environment, generates output. (A book-length description of intelligent agents in the context of AI is provided by (Russell and Norvig, 1995); see Figure 1 for the schema they employ to describe such entities.)

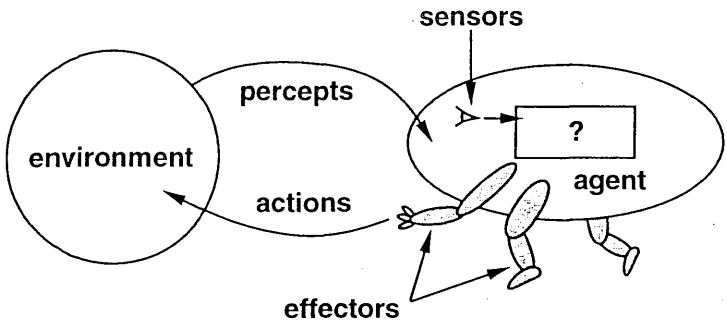


FIGURE 1: Structure of an Intelligent Agent

As to L, there is literally no agreement on what it is, formally speaking. Some believe that L can be a standard first-order language; others believe that that is provably false (in light of such facts as that even the Peano axioms for arithmetic are inexpressible in first-order logic).

For our purposes it's sufficient to note that significant theorems almost invariably combine natural and formal languages. Gödel I, which we have seen spelled out above, is itself a perfect example.

Now, I don't pretend to have an outright *definition* of the central locution, in the sense of a set of jointly necessary and sufficient conditions for it's exemplification. But to defend negative answers to Q1 and Q1* I don't *need* a definition. It would be sufficient if I had just some necessary conditions, as long as those conditions are ones machines can't seem to satisfy. I believe I have such conditions:

[C1] If agent S proves P in model-based fashion, then
(1) S understands underlying objects, concepts, propositions, and relations, which we can list as o_1, o_2, ..., o_n;
(2) S represents o_1, ..., o_n in *both* linguistic and visual modes (the visual mode yielding $\tilde{o}_1,...\tilde{o}_n$); and
(3) S manipulates $\tilde{o}_1,...\tilde{o}_n$ in order to carry out inferences.

C1 should seem quite plausible in light of our the model-based explication of Gödel I provided in the previous section. Consider, for example, our discussion of simian machines. I had in mind an underlying concept: computation, or effective procedure. This concept is independent of, and prior to, any particular representation. I then appealed to a visual repre-

sentation scheme to express this concept, one that made use of mental imagery and diagrams. Finally I manipulated this representation (e.g., to give a Gödel numbering system).

Let us now proceed to see if this kind of activity occurs when machines prove such things as Gödel I.

4. A Computerized Proof of Gödel I

A mechanized proof of Gödel I has recently been engineered by Art Quaife (1992). This proof was was carried out by OTTER, a purely syntactic resolution-based theorem prover particularly well-suited to reasoning in first-order extensional logic. The trick that allows OTTER to prove a "deep" meta-mathematical theorem like Gödel I is Quaife's encoding of this theorem in the modal system K4. All this is unpacked in the following sequence: I describe OTTER, then K4, and then the OTTER-based proof of Gödel I.

4.1 OTTER Encapsulated

Consider the following simple "natural deduction" style proof in the propositional calculus of the fact that from a conditional $p \rightarrow q$ one can derive $\neg q \rightarrow \neg p$ (this move is known as contraposition or transposition).

$p \rightarrow q$ (given)
$\neg q$ (assumption)
$\neg\ p$ (*modus tollens*, lines 1 and 2)
$\neg\ q \rightarrow \neg\ p$ (lines 2-3, conditional proof)

This is the sort of simple proof that students of logic learn at the outset of their education. Notice the use of three rules in this little proof: assumption, *modus tollens*, and conditional proof. Normally, many such rules are added to the arsenal of the human who learns to do proofs in first-order logic. By contrast, OTTER really only has, at bottom, one rule: resolution. Here is an actual OTTER input file for the problem of finding a proof of transposition:

```
% This propositional logic problem, by the way, was
% the "most difficult" (!) theorem proved by the
% original Logic Theorist of 1957.
set(auto).
formula_list(usable).
((p -> q) <-> (-q -> -p)).
end_of_list.
```

The lines that begin with the character % are comments (and they may well reveal how far theorem proving has come in just over four decades!). The line `set(auto)` simply tells OTTER to attack the problem "autonomously" as it sees fit, without using any particular strategies. There then follows a list flanked top and bottom by `formula_list(usable.` and `end_of_list`; in this case the list only has one element, viz., `-((p->q) <-> (-q->-p)).`[6]

This is the negation of the theorem to be proved. The theorem is negated because if OTTER can derive a contradiction from this negation conjoined with consistent information given it, it will follow by indirect proof that transposition is valid. OTTER does indeed find such a proof instantaneously; here is the actual output.

```
---------------- PROOF ----------------

1 [] -p|q.
2 [] -q.
3 [] p.
4 [hyper,3,1] q.
5 [binary,4.1,2.1] $F.

------------ end of proof -------------
```

Lines 1, 2, and 3 represent the result of reformulating the formula

```
-((p -> q) <-> (-q -> -p))
```

in clausal form. The formula `-((p -> q) <-> (-q -> -p))` is therefore equivalent to the conjunction of `-p|q`, `-q`, and `p`. (A truth table will reveal that this conjuction is true and false under exactly the same truth-

[6] Notice that since OTTER takes input from an ordinary keyboard, the negation symbol ¬ becomes -, → becomes ->, ∨ becomes |, etc.

value assignments to `p` and `q` as `-((p -> q) <-> (-q -> -p))` is.) Each of these conjuncts is composed of a disjunction of literals, where a literal is either a propositional letter (in which case it is said to be *positive*) or the negation of one (in which case it's said to be *negative*). Lines 4 and 5 in the OTTER proof are applications of the rules of inference "binary resolution" and "hyperresolution", respectively. These rules are really quite straightforward. Binary resolution for the propositional case is just

$$\frac{\phi \vee \psi \quad \neg\phi}{\phi}$$

where the formula below the horizontal line is inferred from the formulas above the line. (The greek letters here stand for arbitrary formulas in the propositional calculus). You should be able to see now, after looking at this rule, why the inference in line 5 of the OTTER proof goes through. Hyperresolution is a little more complex. To understand it, notice that each disjunction of literals, e.g., `p|q|r|-s`, can be viewed as a set; this particular disjunction would become $\{p, q, r, \neg s\}$. Now, intuitively, hyperresolution simply says that contradictory literals are cancelled out to leave positive literals. In line 4 of the above OTTER proof, for example, `p` in line 1 and `-p` in line 3 contradict each other and cancel out, leaving `q`. The general schema for hyperresolution is as follows.

$$\frac{\begin{array}{ll}\Phi_1 \cup \{\neg\phi_1, \neg\psi_2, \ldots, \neg\psi_n\} & \text{all } \Phi j \text{ positive} \\ \Phi_1 \cup \{\neg\psi_{i1}, \neg\psi_{i2}, \ldots, \neg\psi_{ik}\} & 0 \leq i_k \leq n \\ \vdots & \\ \Phi_{n+1} \cup \{\neg\psi_{i1}, \neg\psi_{i2}, \ldots, \neg\psi_{im}\} & 0 \leq i_m \leq n\end{array}}{\Phi_1 \cup \Phi_2 \cup \ldots \Phi_{n+1}}$$

4.2 Introduction to Encoding in OTTER

Consider again the formula $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$ which, as we have noted, is a tautology in that part of first-order logic known as the propositional calculus. As we have observed, this formula can be written in the syntax of OTTER (as, recall, `(p -> q) <-> (-q -> -p)`), and can then be negated in OTTER in order to generate a machine proof of a contradiction. But there is another way. We can *encode* the formula in OTTER;

here's how. We can use the predicate **T** in OTTER to stand for "is a theorem in first-order logic", and we can represent formulas in first-order logic as terms in OTTER. In order to do this, we can use the following table, where "$O[\phi]$" denotes the representation of ϕ in OTTER code[7].

Formula in FOL	*Term in* OTTER
$\phi \rightarrow \psi$	`i(O[φ],O[ψ])`
$\neg\phi$	`n(O[φ])`
$\phi \leftrightarrow \psi$	`c(O[φ],O[ψ])`

Here is an example of this table in action; the example is an OTTER input file that encodes four propositions which together axiomatize all of the propositional calculus (i.e., all tautologies in the propositional calculus can be derived from these four propositions).

```
set(auto).
formula_list(usable).
% Modus Ponens:
all x all y ((T(i(x,y)) & T(x)) -> T(y)).
% Three more axioms:
all x all y (T(i(x,i(y,x)))).
all x all y all z
(T(i(i(x,i(y,z)),i(i(x,y),i(x,z))))).
all x all y (T(i(i(n(x),n(y)),i(y,x)))).
% The theorem negated to produce a contradiction,
i.e., $F
-(all x all y T((i(i(x,y),i(n(y),n(x)))))).
end_of_list.
```

This input file, when run by OTTER, does indeed produce $F. (It just took 44 lines on my computer.) But the important thing to note is that the technique of encoding formulas is one Quaife exploited in order to get OTTER to prove Gödel I. However, Quaife's encoding was a bit trickier: he first translated this theorem (and the propositions needed to prove it) into a (modal) logic known as K4, and then encoded this translation in OTTER. We turn now to K4.

---

[7] When pondering the following table, and the OTTER code that makes use of it, keep in mind that `T(c(O[φ],O[ψ]))`,- which essentially says that the biconditional $\phi \rightarrow \psi$ is a theorem of first-order logic - is represented in OTTER as a conjunction of the two conditionals that make up the biconditional, viz., `T(i(O[φ],O[ψ])) & T(i(O[φ],O[ψ]))`.

4.3 The Modal System K4, Briefly

K4 is a (so-called normal) modal logic, and as such would traditionally be taken to systematize the concepts of logical necessity and logical possibility. To the language of ordinary first-order logic, modal logic adds two new operators, namely $\Diamond$ and $\Box$. When we write $\Diamond\phi$, where ϕ is any well-formed formula, we are essentially saying that it is logically possible that ϕ, that is, intuitively, that ϕ is coherent or free from contradiction. (E.g., it is physically impossible, but nonetheless logically possible, that s = Selmer swims across the Atlantic; hence we could assert $\Diamond s$.) When we write $\Box\phi$ we are essentially saying that it is logically necessary that ϕ. (For example, $\Box$ 2+2=4 would be true, since it's not coherent to suppose that the sum of two and two isn't 4). The particular system K4 is composed of the following axioms and rules of inference.

Axioms

- Propositional tautologies.
- K: $\Box(\phi\rightarrow\psi)\rightarrow(\Box\phi\rightarrow\Box\psi)$
- 4: $\Box\phi\rightarrow\Box\Box\phi$

Rules of Inference

- If $\vdash\phi\rightarrow\psi$ (i.e., if it's provable that $\phi\rightarrow\psi$) infer to $\vdash\phi$ then $\vdash\psi$

RN: If $\vdash\phi$ infer to $\vdash\Box\phi$

4.4 The OTTER-Based Proof of Gödel I

In Quaife's OTTER-based proof of Gödel I, the operator $\Box$ is interpreted to mean *provable* (and thus to stand for $\vdash$, with which you're now familiar). The key enabling fact underlying the OTTER-based proof is (essentially) that every theorem in K4 corresponds to a truth in a representable system. Given this fact, the following scheme allows Quaife to encode Gödel I in OTTER.

- Interpret "`b(x)`" to mean "`x` is provable in a set Φ which is such that *Rep*Φ".
- Formulas of K4 become terms in OTTER, just as above, formulas in the propositional calculus became terms in OTTER.
- Specifically, the predicate `ThmK4(x)` means that the formula `x` is a theorem of K4 (just as `T(x)` stood above for "is a theorem in proposi-

tional logic").

This means that (most of[8]) Gödel I can be encoded in OTTER as:

```
[ThmK4(c(x,n(b(x)))) & ThmK4(x)] -> ThmK4(F)
```

What this says is that if it's a theorem of K4 that some formula is a theorem if and only that theorem isn't provable, and that formula is provable, then a contradiction can be derived. If you think about it, you will grasp that this is another way of putting (most of) Gödel I. Now here, at long last, is the actual proof of the encoded Gödel I. Note that lines 1-5 are things already known to to be true (if you spend some time reflecting on them with pencil and paper, I'm confident you'll find them to be intuitively correct), and 6-8 constitute the negation of the OTTER-encoded version of Gödel I, with `p` representing a particular formula[9]. So the entire proof has only four inferences!

```
---------------- PROOF ----------------
1 [] ThmK4(i(c(x,y),i(x,y))).
2 [] ThmK4(i(c(x,y),i(c(y,z),c(x,z)))).
3 [] ThmK4(c(n(x),i(x,F))).
4 [] -ThmK4(x)|-ThmK4(y)|-
ThmK4(i(x,i(y,z)))|ThmK4(z).
5 [] -ThmK4(c(x,i(b(x),y)))|-
ThmK4(i(b(y),y))|ThmK4(y).
6 [] ThmK4(c(p,n(b(p)))).
7 [] ThmK4(n(b(F))).
8 [] -ThmK4(F).
9 [hyper,7,4,3,1] ThmK4(i(b(F),F).
10 [hyper,6,5,3,2] ThmK4(c(p,i(b(p),F))).
11 [hyper,20,5,9] ThmK4(F).
12 [binary,11,8] $F.
------------ end of proof -------------
```

[8] The part of the theorem that says that if **n(x)** is a theorem of K4 then a contradiction ensues is the missing part.

[9] Also, for cognoscenti: I've compressed the proof by among other things omitting declarative representation of the interconnection between the functor **c** for the biconditional and the functor **i** for the conditional.

5. A Searlean Argument For Negative Answers to Q1 and Q1*

Presumably, the OTTER proof just given strikes you as being as far from model-based reasoning as the east is from the west - but such intuitions need to be refined in order to defend my negative answers to Q1 and Q1*: the more refined argument is really quite simple, and we can get right to it. To begin, we simply instantiate C1 from section 3 so as to refer to a computer running OTTER as directed by the Quaifean scheme covered in the previous section:

[C1´] If a computer *C* running OTTER proves Gödel I in model-based fashion, then
(1) *C* understands underlying objects, concepts, propositions, and relations, which we can list as o_1, o_2, ..., o_n;
(2) *C* represents o_1, ..., o_n in *both* linguistic and visual modes (the visual mode yielding $\tilde{o}_1,...\tilde{o}_n$); and
(3) *C* manipulates representations $\tilde{o}_1,...\tilde{o}_n$ in order to carry out inferences.

Next, note that none of clauses 1, 2, or 3 in C1´ are true. From the negation of these clauses it follows by *modus tollens* that a computer *C* running OTTER fails to prove Gödel I in model-based fashion. In the case of such a computer, as we have seen, there are no visual representations of anything; that is why clauses 2 and 3 are indeed false. And we can overthrow clause 1 via the following Searlean argument:

The Searlean Argument
(4) If *C* understands underlying objects, concepts, propositions, and relations o_1, o_2, ..., o_n, when proving Gödel I in the OTTER-based fashion specified by Quaife, then if you simulate *C*'s proof, you will understand o_1, o_2, ..., o_n as well.
(5) It's not the case that if you simulate *C*'s proof of Gödel I you will understand o_1, o_2, ..., o_n.
(6) *C* doesn't understand underlying objects, concepts, propositions, and relations o_1, o_2, ..., o_n when proving Gödel I in the OTTER-based fashion specified by Quaife.

I have attempted to ascertain whether or not premise (5) is true in the

following (confessedly unsystematic) manner. *Before* teaching students Gödel I in the model-based manner summarized earlier in this paper, I have given them this theorem encoded in OTTER, along with the OTTER-based proof of this theorem you've just seen. No student ever has any idea what is going on. (Put yourself in my students' shoes: reflect upon whether you would have any understanding of Gödel I if your only exposure to it were the above proof that OTTER gives of the encoded version of theorem.) What this shows is that when a computer proves Gödel I by using OTTER as above it is just moving symbols around without any understanding whatsoever - and yet the one thing model-based deductive reasoning clearly involves is genuine understanding of underlying constructs (which can then be represented in non-standard ways).

6. The Objection From Hyperproof

Against the argument offered in defense of a negative answer to Q1 and Q1* I expect to encounter: "At best, Bringsjord, you have shown that this *particular* machine proof of Gödel I falls well short of model-based reasoning (as partially characterized by C1). OTTER, after all, is just one theorem prover out of many extant ones; and surely you will agree that many others will arrive in the future. You might retort that I am simply expressing faith that MBR_D can be reduced to computation; you might, specifically, demand to see a computational system that can do MBR_D. Well, as a matter of fact, Bringsjord, I have such a system on hand: Hyperproof (Barwise and Etchemendy, 1994). Consider, for example, the model-based solution to the barber problem you discussed above. You used a simple table to provide a solution. Hyperproof can be used to solve this problem visually as well - by way of the visual situation shown in Figure 2. In this situation, the men are on the left (dodecahedrons) the women are on the right (tetrahedrons), and the barber is `b`, the frontmost tetrahedron. If an arc runs from an object *o* to an object *o´*, and the arc has a heart on it, it means that *o* shaves *o´*. If an arc runs from an object *o* to an object *o´*, and the arc has a *broken* heart on it, it means that *o* does *not* shave *o´*. Hyperproof includes, as well, standard, symbolic, first-order deduction.

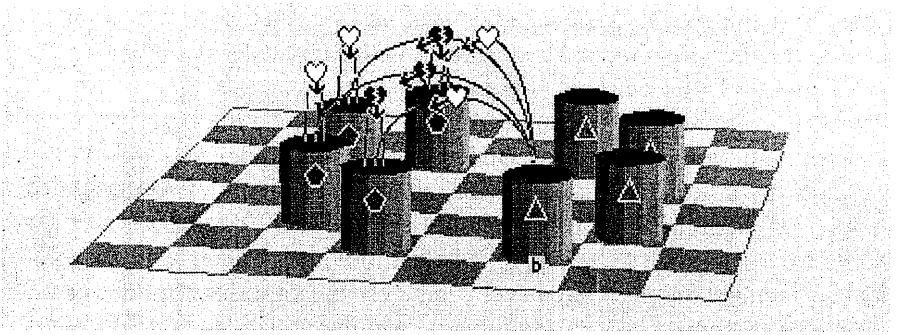


FIGURE 2: Hyperproof situation that solves the barber problem.

Despite the fact that Hyerproof should be learned and experimented with by anyone serious about model-based deductive reasoning, this objection fails - for two main reasons. The first is that while Hyperproof allows for the construction of visual situations, and deduction thereto and therefrom, these situations are very primitive. It is not possible, for example, to cast simian machines in Hyperproof. (Technically, it's not even possible to cast the two-table representation I gave above for Gödel numbering in Hyperproof. But I think it's fair to say that one can at least see how this representation could be cast in something *like* Hyperproof). Hyperproof is primarily intended to be a system for teaching first-order logic, and on that count it succeeds magnificently. (For evidence of this, see my (Bringsjiord, et. al., 1998). But this means that the graphical side of Hyperproof must be understandable in terms of first-order logic. And yet temporally extended mental images (or TEMI's as they are called in (Bringsjord and Bringsjord, 1996)), such as the image of a simian machine working through time, are not known to be reducible to first-order logic.

The second problem with the objection is more straightforward, and more serious. It is simply that Hyperproof is essentially OTTER with a facility to represent, in the form of a grid, first-order formulas. Hyperproof is not a system that can do anything like what C1 says is necessary for model-based dedutive reasoning. Hyperproof *itself* does not represent underlying information; rather, people *use* Hyperproof to represent information *they* understand.

Will some future computational system genuinely understand information which it casts in model-based representations? I doubt it. Certainly none of today's systems, as we move to the new millenium, provide any reason whatsoever to think that such a computational system will arrive. Today's systems, as we've seen in connection with Gödel I, provide instead reason to be pessimistic about the prospects for machine-based MBR_D. That is why Q1 and Q1* should be answered today in the negative. If future researchers prove these answers wrong, and I'm still alive, I'll be happy that this paper served to goad them on.

Rensselaer Polytechnic Institute

REFERENCES

Barwise, J. and Etchemendy, J. (1994), *Hyperproof.* Stanford, CA:CSLI.

Boolos, G.S.and Jeffrey, R.C. (1989), *Computability and Logic.* Cambridge, UK: Cambridge University Press.

Bringsjord, S. (1998), Chess is Too Easy, *Technology Review* 101.2: pp. 23-28.

Bringsjord, S. (forthcoming), 'The Zombie Attack on the Computational Conception of Mind', *Philosophy and Phenomenological Research.*

Bringsjord, S., Bringsjord, E. and Noel, R. (1998), 'In Defense of Logical Minds', in *Proceedings of the 20th Annual Conference of the Cognitive Science Society.* Hillsdale, NJ:Lawrence Erlbaum Associates, pp. 173-178.

Bringsjord, S. and Bringsjord, E. (1996), 'The Case Against AI From Imagistic Expertise', *Journal of Experimental and Theoretical Artificial Intelligence* 8:383-397.

Bringsjord, S. (1992), Chapter VIII: Gödel, in his *What Robots Can and Can't B*, Dordrecht, The Netherlands: Kluwer.

Bringsjord, S. and Ferrucci, D. (1999), *Artificial Intelligence and Literary Creativity: Inside the Mind of Brutus$_1$, A Storytelling Machine.* Hillsdale, NJ: Lawrence Erlbaum.

Cervantes, M. (1999), *Don Quijote.* New York, NY: Norton.

Ebbinghaus, H.D., Flum, J. and Thomas, W. (1994), *Mathematical Logic.* NY: Springer-Verlag.

Penrose, R. (1969), *The Emperor's New Mind.* Oxford, UK: Oxford University Press.

Penrose, R. (1994), *Shadows of the Mind.* Oxford, UK: Oxford University Press.

Quaife, A. (1992), *Automated Development of Fundamental Mathematical The-*

ories. Dordrecht: Kluwer.

Russell, S. and Norvig, P. (1995), *Artificial Intelligence: A Modern Appraoch*. Englewood Cliffs, NJ: Prentice-Hall.

Turing, A. (1936), 'On Computable Numbers, with an Application to the Entscheidungsproblem', *Proceedings of the London Mathematical Society*, series 2, 45: 161-228.

Smullyan, R.M. (1992), *Gödel's Incompleteness Theorems*. Oxford, UK: Oxford University Press.

Smullyan, R.M. (1982), *Alice in Puzzleland*. NY: Morrow.

Wang, H. (1995), 'On "Computabilism" and Physicalism: Some Subproblems', in J. Cornwell (ed.), *Nature's Imagination: The Frontiers of Scientific Vision*. Oxford, UK: Oxford University Press.