TEACHING AI EPISTEMOLOGY TO HUMANS

Mark B. Fishman
Eckerd College
St. Petersburg, FL 33733

ABSTRACT

Intelligence seems integrally to involve recursion in the form of self-monitoring, but intelligent beings in the form of undergraduates seem pretty well immune to apprehension of the concept -- not to mention most of the important mathematical concepts essential to the practice and philosophical appreciation of artificial intelligence. The presenter describes an approach to the teaching of predicate logic, Goedel's Theorem, formal languages, finite automata and recursion theory to undergraduates with no significant mathematical background, in the context of a philosophical examination of the history of artificial intelligence.

INTRODUCTION

Eckerd College offers faculty and students an opportunity to explore nontraditional subjects -- or traditional ones in an idiosyncratic way -- in the course of its post-fall, pre-spring, one-month winter term. In an effort to bridge the chasm of mathematical and computational unknowledge which separates non-science majors from the semblance of analytical competence, I decided to offer a course inspired partly by Douglas Hofstadter's *Goedel, Escher, Bach*, and designed to bring some of the central concepts of computer science and artificial intelligence to students who might not otherwise ever encounter them -- save as faintly reflected, in a cavelike, Platonic sort of way -- in their appearance in virtual reality environments in video games. The course has had a generally enthusiastic reception from students outside the Natural Sciences Collegium, and a good number of them have been subverted thereby into choosing a major or minor in the dreaded field of computer science.

What follows are notes from a handout distributed in that course (and that have appeared in handouts and in a collection of essays used in other courses) at Eckerd College.

NOTES ON RECURSION, GOEDEL AND THE HALTING PROBLEM

A story is told of William James[1], the great psychologist and founder of that school of philosophy known as pragmatism, who died before it was possible to be relevant to undergraduates of today -- in a time so remote that we who are posterity can scarcely discern its attenuate echo in the thready wind that blows through eternity ... well before 1970, in any case. James, it seems, a Renaissance Man of his day (which was somewhat after the Renaissance, but well before the onset of MTV), liked to give talks on cosmology: the sun, the moon, and the earth, 101 things you could do with a dead star, and how they were all related. This being hundreds of years after a subjugated yet defiant Galileo had uttered the moving words, "Eppur Si muove!" ("and yet it moves!" -- words that were in Italian, so that few if any Floridians would later be able to understand them), James felt confident that among the few immutable fixtures of the cosmological firmament, the facts on which you could safely lay an astrophysical bet, was surely this: the Earth, it seemed clear, revolved around... the sun! His talk was well-received, but afterwards, a rather cranky and querulous member of the crowd toddled up to him to take issue:

"I’m surprised at you, Mr. James. I’m surprised and upset! Such fiddle-faddle, from such a sophisticated man! The Earth most assuredly does NOT revolve around the sun. I know whereon the Earth sits. It sits on the back... of a giant turtle!"

Well, James attempted gently to dissuade, pursuing this line of logic: "Consider, please, if the Earth sits atop a giant turtle, then on WHAT does that selfsame turtle have to repose?"

"Perfectly obvious. I’m astonished, young man, a fellow of your intelligence and breeding! It sits on an, ever-so-much larger, GIANT TURTLE."

James scarcely could utter a syllable before the fellow, anticipating his impending logical redress, shook a finger and said:

"Oh, you’re a VERY CLEVER MAN, Mr. James, and I know what you’re going to say, but its no good: it’s turtles ALL THE WAY DOWN!" [1]

THE TURTLES OF KNOWING

Turtle...
Turtle...
Turtle...
Turtle...
Turtle...

What, the alert student, having digested this particularly demented story, may now ask, has this got to do with the nature of intelligence and uncertainty, who we are and what we can know, and more to the point, whether any decaffeinated coffee can ever really approximate the taste of a regular blend? It avails, though, it does:

"Turtles all the way down" is our working definition of "recursion;" and recursion, strange, involute concept that it is - yin and yang, reflecting mirrors, the Worm Ouroboros that eats its

----
[1] This story was originally told in the preface to John Robert Ross’ doctoral dissertation at MIT, "Constraints on Variables in Syntax," and has since appeared in various guises in numerous venues, including that of Stephen Hawking’s, "Brief History of Time."

tail, DNA and finite-state machines that self-replicate, an endless, spiralling descent of turtle feet that seeks ever to secure a foothold on that last, testudinidaean shell: that is our mind, that is what we can know and that is who we are. That is the subject of this course.

What does it mean, formally, to know? Do we say that we know that which is encoded in our brains (rather holistically, it appears), because it resides therein, because we can call it forth -- as, for example, when it becomes urgent to disgorge the capital of Ethiopia in a particularly volatile game of Trivial Pursuit? The knowing does not seem to us synonymous with the mere holding of information, as witness our refusal to credit the Trivial Pursuit cards themselves with knowledge of the human kind, for all that they generally hold the right answers -- which we don’t. And we’ll extend our epistemological disdain to other, less primitive storage media, those of the information age -- floppies and tape drives and chips, oh my! Who among you is ready to credit a megabit of DRAM (one million tiny little switches, crowded onto a chip so incalculably dense of circuitry, they had to be etched there by electron beam) with "awareness" of which way its own switches are set? Ask a chip: "Do you know which way your switches are set?" See what kind of answer you get. (Do not try this in public, though, or in the presence of responsible mental health authorities.) What page knows whereof it speaks, and if it did, how could the Congressional Record live with itself?

The upshot is clear: When we, as humans, say that we "know" something, we mean more than that it’s engraved statically, somewhere in the cortex, even as words are engraved in ink on a page. The "knowing" consists in a process, or the registered potential of evoking that process: the ongoing process by which we inspect ourselves, which we sometimes call "consciousness," and which some of us who are computer scientists would like to reproduce, characterize -- somehow formally describe -- that thing which is most quintessentially human -- awareness of ourselves -- then we would have shown that we understand it. It’s just what the Delphic Oracle ordered, and the question only arises of whether we can. Is it feasible, mathematically, to build a system that will "know itself?" It should be: it’s exactly such a system that we are, and we exist! Few philosophers, excluding perhaps radical Skinnerian behaviorists, have attempted to argue that they don’t. Perhaps, then, the only real question is whether it’s feasible for us.

It may fall to you to wonder why it should be that a thing that exists can elude capture, fail to be susceptible of knowing. I have not said that the essential, self-referential mechanism of our minds cannot be known, but actually, there are some things that can’t, and the awful, inescapable onus of formal systems is that it’s mathematically, finally, and ultimately provable. Follow: In perhaps the most peculiar mathematical demonstration of the twentieth century, a German logician by the name of Kurt Goedel showed, absolutely and beyond contradiction, that there will always be truths, predications that are true about the world, that we cannot possibly prove. This is more than disturbing; we’ll take it up again. But let’s first return to the issue of recursion, which will provide a prolegomenon to what Goedel sought to do, and a glimpse, perhaps, of what we are inside.

RE: CURSION. CONCERNING THIS HEADING

Someone once said of intelligence that it is the "ability to find and perceive relationships... where none exist." Others have said of practitioners of artificial intelligence that they exemplify this trend. Still others, finally, have observed that a delegation of extra-terrestrials, entering a computer science bookstore, would quickly deduce that most Earthmen spend most of their time computing the factorial function. This observation has (sometimes) been meant to be funny.

What is the factorial function, how does it relate to recursion, and why should you wish to compute it? There is no satisfactory answer to the latter question, but a few examples should suffice to answer the first:

Five factorial (usually written 5!) = 5 x 4 x 3 x 2 x 1

6! = 6 x 5 x 4 x 3 x 2 xl

7! = 7 x 6 x 5 x 4 x 3 x 2 x 1

In brief, then, to find N!, all we have to do is to compute the product of all integers leading from 1 to N. This is a boring thing to do, but it may keep undesirable segments of society (computer scientists, for example) off the streets. How to write a program to do it, though, is another question, and one that can be answered more easily in the language of recursion.

Simply put, I may not know how to compute 100!, but if you will provide me (somehow, magically) with the solution to 99!, then I will be able to obtain 100! quite straightforwardly: I will have but to multiply your magically determined value by 100.

This, of course, reduces the problem to that of obtaining 99 factorial, concerning which just the same reasoning applies. All I need to solve the problem is somehow to obtain 98 factorial, and then multiply by 99. And 98! reduces further to the problem of 97!. And 97! to 96!. To put it more formally:

```
                factorial (N) = N * factorial (N-l)
```

But if every factorial is defined in terms of the previous factorial (and self-definition is, indeed, what we mean by "recursion"), then do we ever stop? Is there, somewhere out there, a poor, vertebrally strained bottom turtle," doomed to bear the weight of all those others? Or is it "turtles, all the way down"?

The answer, in the case of the factorial function, is that the bottom turtle is the number zero. When you seek the value of 0!, you should not start having to look for (-1)!. Rather, the value of 0! is defined, immediately and absent any further proliferation of stacked-up turtles, to be 1.

What does any of this have to do with consciousness? Recursive processes are processes defined in terms of themselves, that call themselves, elaborate themselves, sometimes interpret their very own code. When a calculator gives us the answer to an arithmetic problem, we do not posit consciousness of the calculator. When a computer spreadsheet contends successfully with the intricacies of a form 1040, we nevertheless decline to credit the chip or the software that support the spreadsheet with any sense of accomplishment, any vagrant reflections on the nature of "taxing calculations." But we ourselves, in solving these kinds of problems, have awareness that we do it. We are what computer scientists refer to as monitors, overarching procedures in control, examining the mental activity we sustain, and examining even the very subprocess that does the examining. And is there a bottom turtle to this regression of awareness? There do seem to be performance constraints on human cognition, but perhaps, after a time, the turtles merely "fuzz out." The important point is that our awareness consists in an act of self-referentiality, an act that finds ironic adumbration in the formal structure of some sad proofs to come: that there are things we can’t do, things we can’t know and limitations, in principle, on the cognition of human beings, and also of "artificial systems," any we might ever be able to create.

THINGS WE CAN’T DO

The Pope is said to have asked a dilatory Michelangelo, apropos of his interminable attentions to the ceiling of the Sistine Chapel, "When will you make an end?" And Michelangelo responded, in that curiously indeterminate way unique to ceiling painters, "When I am finished."

For a long time, an important question to computational theorists was to find a way of determining, for any arbitrary program, whether it would ever halt, come to an end, disgorge an answer. (This has been a question important, also, to watchers of Dallas or Falcon Crest, but the problem in computer science came first.)

If you have never programmed, it may occur to you to wonder what kind of program never halts. Generally speaking, programs that are written incorrectly (which is to say, nearly all of them) succumb to this characterization; they loop endlessly, not unlike the line at Burger King, or a Monday morning lecture. There may, though, also be programs that run forever for perfectly legitimate reasons. Consider a program that attempts to break into your computer account by trying ALL numeric passwords. But it happens that your password is "r2d2." The program is destined to run eternally, or at least until it can generate no larger numbers. If we had a general solution to the "Halting Problem," though, no doomed and blighted program, destined to loop forever, would keep us waiting ever again. It would suffice merely to ask this Oracle (which, for want of a better name, is what we’ll call any putative solution to the Halting Problem), "Will my program halt?" And the Oracle would smile benignly, and say, "No, my son. Never. It has always been thus."

Such an Oracle has long eluded us. The search for one was even as that for a machine of perpetual motion: fruitless, time-consuming and not really suggestive of dazzling mental equilibrium. And then one day it came to pass (as such stories always are begun) that someone considered what might happen if we could, indeed, concoct an Oracle, a real solution to the Halting Problem. We might use the Oracle to construct a program such as the following, which, for the sake of reference, I’ll describe as Sample Program #99:

```
                                ---------
            This flowchart      | START |
            represents          ---------
            Program #99             |
                                    |<--------------------+
                                    V                     |
                         ---------------------------      |
                         |      The Oracle         |      |
                         |         sits            |      |
                         |       here and          | YES  |
                         |   answers the question: |------+
                         |                         |
                         |    Will Program #99     |
                         |       ever halt?        |
                         ---------------------------
                                    |
                                    | NO
                                    |
                                    V
                                 --------
                                 | STOP |
                                 --------
```

A quick check of Program #99 reveals that, if it exists (and it must, if the Oracle can be fashioned), it has at least one interesting property: to wit, it never halts if the Oracle says it halts, and if the Oracle pronounces it haltless, it promptly stops on a dime (to say nothing of a "paradigm"). Well, surely, this is to behave "in strange fashion," and there is only one thing left for us to conclude: This nonsense must halt. To avert a logical contradiction such as may later cause us to conclude that War is Peace, Freedom is Slavery and yesterday’s leftovers are food, we must quickly eliminate the source of the anomaly: our belief, to wit, that such a thing as "the Oracle" can ever exist. It can’t, and the Halting Problem is formally unsolvable. This isn’t, though, the end of our travails.

We now know ourselves to be foreclosed from solving the Halting Problem. Any device or instrument that pretends to provide us with such a solution must therefore be impossible of construction. Well, therein lies "the rub," for it can be shown that if we could solve EITHER of the following problems, the solution to the Halting Problem, as the night the day (or as finals week the semester), would follow:

1) The Correctness Problem: the problem of showing that any program correctly corresponds to its formal specification, hence will do what it’s supposed to do.

2) The Speed Problem: the problem of showing that a program is the most efficient one possible that will handle its particular, designated task.

It follows from the impossibility of an Oracle, and the fact that an adequate solution to either of the above coadunate problems would provide us with one (the proof is straightforward, but mercy constrains me to omit it here), that the Correctness Problem and the Speed Problem (to say nothing of others of which nothing should be said) cannot be solved, not ever (so don’t devote the weekend to it).

There are, it seems, a consternating array of tasks that we cannot, in general, automatically perform. What about truths that we cannot derive? The more disturbing revelation comes next:

THINGS WE CAN’T KNOW

```
        A Phrasebook of First-Order Predicate Logic
          (Not recommended for travel)

Vx[man(x)-->mortal(x)]

        "All men are mortal."

man (Socrates)

        "Socrates is a man."

Vx[man(x)-->equal(x,Socrates)]

        "Therefore, all men are Socrates."

Vx[(antelope(x) & possess(I,x)) --> well-behaved(x)]

        "All of my antelopes are well-behaved."

chicken(cousin(friend(you)))

        "The cousin of your friend is a chicken."

Vx[turtle(z)-->  "No matter what turtle you choose, I can find one with a
                  lower number."]
```

It is necessary, first, to recognize that there are actually people to whom the above statements read intelligibly, many of whom are allowed out in public unaccompanied, and without heavy medication. Years and years ago (more than three, and certainly more than you want to know), an American linguist by the name of Benjamin Whorf betook himself to wondering what might be the world view of a human whose language took no account of trees, of streets, of the disparate shades of gray in a mucky, industrial sky, of digital watches. And he concluded this: Our Weltanschauung (German for "world view;" Whorf did not anticipate the Florida problem, either) derives from and uniquely reflects the language that we speak, natively. What Whorf would have thought of humans who speak predicate logic, who teach it to computers, and who see all reality expressed therein, awaiting only the right symbolic manipulations to yield up hitherto unexpressed verities remains, felicitously, open to speculation. (Word from the ectoplasmic world has it, though, he’s seriously depressed.)

Logic can be seen as a kind of constructor set of truths -- slices, dices, makes Julienne theorems. We start with a set of "axioms," truths that are accepted unconditionally, a priori and without antecedent logical justification. Some of these axioms are "tautologous" - true in virtue of their form, and quite irrespective of meaning - such as that "p is true, or it isn’t." In the language of predicate logic: (p v -p). Others, we accept as true merely in virtue of their seeming empirical validation in the "universe of discourse" -- the world as we know it.

This initial set of truths we throw into a hopper, that of a machine known to recombine old truths into new ones, preserving that truth unperturbed and invariant, even as two blue-eyed parents will produce, ineluctably, a child who is also blue-eyed. The squealing, infant truths that emerge thus freshly engendered we like to refer to as "theorems." (Well, we don’t actually so much like to, as feel compelled to by the exigencies of lexical tradition.) The machines that do this recombining to produce new truths we refer to as "rules of inference." Some that are popular in logic circles are modus ponens, modus tollens, and resolution. As an example, modus ponens says that if you know that p is true, and you also know that whenever p is true q is true, then you can give birth to the new baby truth, q. The nice thing about this process, is that the baby truths are just as fecund as their parents, and can be dumped immediately into other hoppers to produce yet more truths, quickly overpopulating the logical universe in a wonderful, awesome, confusing abundance of facts, some of which may even be relevant to whatever it was that we wanted to explore in the first place.

This, curiously, is how human logicians do proofs, and how computers do them as well. It is, too, the only way we have of establishing that certain kinds of statements -- statements that make predications of infinite sets of objects that cannot be verified for each object individually, to take an "obvious" example -- are true.

All of this sounds ducky. ("Ducky" is an abstruse, technical term referring to a generally satisfactory state of affairs.) All truths can now be established, all verities are happily potentiated, all worlds go from truth to more truth. Something, you may now suspect, is bound to go wrong. Goedel saw it coming. It was this (I’m going to say it briefly):

Goedel imagined what would happen if we numbered all the truths in the world, using a conversion procedure he defined for statements expressed in predicate logic. Then it might be possible to write a statement, which would have its own number (statement 1362, let’s call it), that says the following:

> Statement number 1362 cannot be proven in the
> axiomatic logic just described.

And it can’t! If it could, then statement 1362 would be false, and a proof would have been furnished of a patently false statement. Statement 1362 cannot be other than a true statement, and this by its very structure.

Something about this may conspire to remind you (its involution, perhaps, its self-reference) of the proof that there is no solution to the Halting Problem, and indeed, the structure of the argument is much the same. Consider, though, what Goedel’s argument says:

There is at least one truth (and, in fact, there are an uncountable number) that cannot be proven, cannot in principle be proven.

There are facts that we have no way of establishing, no way of guaranteeing, no way of knowing to our satisfaction or even to that of a machine. If this doesn’t disturb you, it should. Something very odd is going on, here, perhaps in this very statement.

REFERENCES

1. Hofstadter, Douglas. Goedel, Escher. Bach: an Eternal Golden Braid. New York: Vintage Books, 1979. (Note: this book won a Pulitzer Prize for Hofstadter.)

2. Nagel, Ernest and James R. Newman. Goedel’s Proof. New York: New York University Press, 1958.