{…}

Being There (1973)

VIZZINI

Now, a clever man would put the poison into his own goblet, because he would know that only a great fool would reach for what he was given. I'm not a great fool, so I can clearly not choose the wine in front of you. But you must have known I was not a great fl; you would have counted on it, so I can clearly not choose the wine in front of me.

MAN IN BLACK

You've made your decision then?

VIZZINI

Not remotely. Because iocane comes from Australia, as everyone knows. And Australia is entirely peopled with criminals. And criminals are used to having people not trust them, as you are not trusted by me. So I can clearly not choose the wine in front of you.

MAN IN BLACK

Truly, you have a dizzying intellect.

VIZZINI
Wait till I get going! Where was I?

MAN IN BLACK
Australia.

— William Goldman: *The Princess Bride*.

Annals of futility, continued:

The *Scientific American* used to have a regular monthly column written by Martin Gardner, a famous guy back in the day, which posed mathematical puzzles for its readers. I picked it up shortly after I got back to Colorado and found a description of what later gained notoriety as Newcomb's Paradox. Gardner had found this in a paper written by Robert Nozick (then a Famous Professor of Philosophy at Harvard, which caught my eye), and passed it on to his own wider public for discussion. I solved it on inspection[1] and forget the matter for a week or two, when I realized that though there was no point in writing Gardner a letter, since as always he would get hundreds, I might try writing Nozick himself. So I typed up a few pages — honestly, less cryptic than usual — stuffed them into an envelope, committed them to the mails, and settled in to await the telegram announcing my accession to the Harvard faculty.

Of course what transpired was nothing of the kind. Gardner, who ironically[2] seemed to have read my mind, announced that he had received so many responses that he turned around and sent

---

[1] A rarity. Usually I stare at a problem without comprehension, forget about it, and the solution pops into my head two years later when I am taking a shower.
[2] See below.

all of them to Nozick, meaning that my clever attempt to receive individual attention would now be buried under an avalanche of bullshit. Thus naturally I got no reply, and when months later Nozick's summary of the proposals he had received appeared in Gardner's column, again as always opinions divided neatly between the two antithetical positions I had carefully explained were both wrong. I doubt he ever read my letter, and if he did, obviously he didn't understand it. — Otherwise, I realized later, there was no guarantee he wouldn't have published it as his own work.

I suppose a sensible person would have written a real paper about this and submitted it to a journal. But then a sensible person would have had access to a university library that got the volume containing the original paper (the essential reference) sooner than ten years later, would have had enough money to promote his manuscript from the slush pile, and would have been able to delude himself this was more than a silly puzzle only worth attending to on the off chance a Famous Professor would take notice of him.

I have, at any rate, the vague impression that there is now a literature on this subject, and that it is completely worthless. — Though really, who gives a shit. — But (modulo a couple of afterthoughts) what I said in the letter was this:

{…}

Suppose a game involving two players, yourself and a mysterious Being, and a pair of boxes, one of which you can see into, one which you cannot. The Being moves first, and puts a thousand dollars into the transparent box and either a million dollars or nothing into the second box. You then have the choice of taking both boxes, or the second box alone.

The twist here is that we suppose the Being can predict what you are going to do, and will punish greed. So if he[3] knows that you are going to take the second box alone, then and only then will it contain the million dollars; if on the other hand he's sure you are going to take both boxes, the second is empty.

Thus there is a payoff matrix which looks something like this:

Being predicts you will take both boxes	$0	$1000
Being predicts you will take the second box only	$1,000,000	$1,001,000
	You take the second box only	You take both boxes

So what should you do?

On the face of it, the arguments are these:

The Being is infallible, and knows what you will do. If you choose to take both boxes, this will have been foreseen; the second box will be empty, and your payoff will be a thousand dollars. On the other hand if you aren't greedy and choose the

[3] I take it for granted that an asshole who thinks he knows everything and is trying to hose you out of a million bucks would have to be male.

second box alone, it will contain the million. Obviously the sensible thing is to take just the second box.

On the other hand the Being moved first, and the money is in the box, or it is not. Whatever he did you will make more by taking both boxes than by taking one. To suppose otherwise is to believe that you can change something that has already happened by occult influence, which is absurd. In effect you are saying that the contents of the box are not determined until you make your decision — that they do not yet lie, as it were, in your back light cone. But that too is ridiculous, because you can imagine that some friend of yours has already looked[4] for you. You can picture him staring at the million bucks and sending you urgent telepathic signals: "Take both.....Take both.... ."[5]

But neither addresses the real question, which is: who is the second player?

You are supposed to picture, i.e., someone like the mysterious stranger in *Last Year at Marienbad* who baffles everyone by always winning at the game of Nim[6] — you imagine an enigmatic smile, a mocking glance which says, I'm looking through you — this is some entity[7] who has read your source code, knows your

---

[4] Actually though this argument is superficially plausible it's also bullshit; the situation is like Schrödinger's Cat, not the Wheel of Fortune (a not-quite-paradoxical conundrum so universally known and discussed that it is explained, e.g., by Kevin Spacey in the movie *21* [Robert Luketic, 2008]). — The cat may know whether it's alive or dead, but I don't acquire the information until I open the box and look; in effect the determination still lies in my future. Same here.

[5] Nozick made various attempts to sharpen these arguments, none convincing and at least one based on an elementary blunder involving Bayes' theorem, but of course I didn't see them for another decade. — In any case everything he said is irrelevant or simply annoying. I'm just telling you a story here about my own folly.

[6] A solvable game with a known winning strategy; as was explained, of course, by Gardner, in another column.

[7] Wolfe [*The Right Stuff*]: "the anonymous and uncanny Chief Designer, D-503, Builder of the Integral….. — He computes the future! the mighty Integral!"

Gödel sentence, has looked up the serial number on the back of your eyeballs, possesses some sort of X-ray vision that allows him to see into the black box housing the freedom of the will.

But this is a trick and a con, the diversion that misdirects your eye from the shell that hides the little pea. If this were what he was doing, there wouldn't be a problem. If you were choosing on a whim, or an irrational hunch, or flipping a coin, or throwing the I Ching to decide what to do, there wouldn't be a puzzle. Maybe he could guess your choice in advance, but this would be no more problematic than one of those computers that fits in your shoe and predicts where the roulette wheel is going to stop. That wouldn't be a paradox.

No. There's only a paradox when you try to make *the rational choice*. You are trying to decide what you *should* do.

So what is the Being's problem then? What does he have to be able to predict?

Exactly the same thing: what is the *rational* choice?

So both you and the "mysterious Being" are trying to solve the same problem. The black box is transparent.

And the paradox is essentially the same as with the Cretan liar, i.e., self-reference: what the Being is going to have done (invent tenses as necessary) depends on what you are going to do. So what you are going to do depends on what you are going to do.

You know what the Being is going to do: the rules have explained it. The Being knows what you are going to do: you are going to try to maximize your payoff. Nothing is hidden.

{…}

Why was that so obvious? I had the following example[8] in the back of my mind:

Suppose that the physical world is classical and deterministic and you have a computer (for obvious reasons this could be called a Laplacian machine) that can predict the evolution of any system from its initial conditions by solving the differential equations — or whatever — in a fixed amount of time which can be estimated beforehand.

There would be many questions about how precisely the initial state would have to be measured, whether or not you might have to employ a machine that could compute with real numbers and not floating-point approximations to them (Smale later worked out such a theory), etc., but ignore those for the moment. — Suffice it that it makes perfect predictions *about* the world, *within* the world.

Then suppose you ask the machine to tell you whether a light bulb is going to be on at the end of the computation. And then plug the output of the machine into the power switch for the bulb, so that if the machine outputs "on", it turns it off, and if it says "off", it turns it on.

---

[8] I think this is due to John Kemeny. See (perhaps) *A Philosopher Looks At Science*, though I can't find a copy of the book with which to verify the reference.

What this demonstrates, obviously, is that even if complete predictions were possible, they would be self-defeating if allowed to feed back into the system; for essentially the same reasons that language must be segregated from metalanguage. The machine whose output negates its own prediction presents the same problem as the attempt to assign valuations to the statements on a card which read "The statement on the other side of this card is true" and "The statement on the other side of this card is false."

{…}

It should be obvious, incidentally, that prediction is essentially computation.

This follows, really, from Church's thesis: an algorithm must be employed to make a prediction; any algorithm can be realized as a computation by a Turing machine.

Successive approximations can be realized by providing the answers to a series of binary questions. — There is nothing deep here, in practice it is straightforward.[9]

---

[9] Here elided is a lengthy digression in the original manuscript on the question of successive approximation, i.e. whether improvements in precision must generally be efficacious. The natural way to formulate that was in the familiar style, for every epsilon to which you wish a numerical prediction to be accurate there must exist a delta within which the initial conditions should be specified, etc., and that raised the embarrassing possibility that in the general case dynamical systems could amplify small errors in precision and erase the possibility of prediction entirely. — This was already obvious for systems with even a modest number of degrees of freedom, see the ergodic theorems of statistical mechanics, but it seemed a novel idea that it might hold as well for relatively simple systems. — Later, of course, this became known as the butterfly effect, and it would have annoyed me not to have worked it out in greater detail had it not become apparent that Poincaré had beaten everyone to publication before the turn of the century. — I did, however, include this analysis in a lengthy précis of the difficulties in the concept of prediction for a friend who was a graduate student in philosophy; he didn't understand it, but incorporated it in his paper nonetheless, and his instructor parroted all of it in a public lecture a few weeks later, without attribution

But the ease with which the unrestricted extension of the idea leads to paradox makes it seem very strange that we can build computers within the physical world. Something about that doesn't smell right. How is it possible?

Which raises the complementary question, how complex must a mechanical system be to allow the construction of a universal Turing machine? What is the simplest system that can realize one? Because the evolution of such a system would be recursively indeterminate. It would be impossible to predict.

So this would mean that, even within an apparently deterministic physics, there would be elementary dynamical questions that would be effectively undecidable. There would be mechanical systems instantiating the halting problem.

And then: what is the relation to the question of "exact solvability"? "Integrability"? Can something as simple as the classical problem of three bodies be undecidable in this sense?

{…}

You also see that, in general, time travel paradoxes are essentially the same as paradoxes of self-reference and paradoxes of prediction. The ability to see the future is equivalent to the ability to send messages into the past. You don't need to imagine anything as grisly as physically traveling back in time to shoot your grandfather; information transfer is sufficient to generate paradox. The Being may be able to predict what you will do, or

---

either to him or (of course) to the ghostwriter, me. — I briefly considered beating the shit out of the guy, but then realized, as usual: why bother. What's the use.

the Being may have a tachyonic telephone[10] with which he can call himself in the past the moment after you have made the choice, it makes no difference.[11]

So that's the story at first glance: self-reference should be forbidden; the game and the Being are, therefore, impossible.

{…}

Indeed it is a mistake to assume the proposition "There is money in the first box" *has* a truth-value; that its contents have determinate value; that it *has* contents.

{…}

At second glance it's a trifle more interesting.

The Being predicts you'll take one box [1] or both [2].

If [1], then the second box contains a million, thus taking both boxes yields a million plus a thousand, thus that is the optimal choice, thus the correct prediction is [2].

---

[10] Tachyons are hypothetical particles which travel faster than light, invented by bored theoreticians to entertain themselves by bullshitting their way out of paradoxes. Absent *ad hoc* baroque complication, anything that travels faster than light can, in the special theory of relativity, be turned by Lorentz transformation into something travelling backward in time; thus permitting the communication with the past of useful information like where the markets will close and which way to swerve to avoid an oncoming bus. A tachyonic telephone is, accordingly, a useful shorthand for precognition on demand.

[11] Once again: (nonrelativistically) the past is what is known; the future is what isn't. If the Being *knows* what you will do, your future lies in his past. (Relativistically the past light cone is what you know about, the future light cone is what will know about you, and the rest is elsewhere, causally disconnected from here and now.)

If on the other hand he predicts [2], then there's nothing in the second box, but you still gain more by taking both. Thus he should still predict [2] — though: what happened to the million dollars? — and the system has, as it were, an attractor.

Well. — We might believe that for a moment. But consider this: you and the Being have essentially the same problem. This means that the Being, too, is trying to make the choice that optimizes your payoff. Therefore when the Being analyzes the payoff matrix, by the same argument that leads you to select the second column, he must select the second row; thus to maximize your payoff, he's compelled to make the wrong prediction, and say that you will take the first box only. So the inconsistency, or metainconsistency, seems intrinsic after all.

So from this point of view the problem is that the payoff matrix should be symmetric with respect to transposition; the fact that it is not is, then, the root of the confusion. — This isn't consistent with the story we have been telling about taking one box versus taking both, but we can always make up another story. At any rate the matrix should look like this:

Being's choice 2	y	z
Being's choice 1	x	y
	Your choice 1	Your choice 2

where if x < y < z or x > y > z there's a self-consistent strategy, whereas if x, z < y or y < x, z there is not.

I don't know that I take this argument seriously, but it's no dumber than what we started with.

{…}

Superficially it might seem that the paradox might be tamed by fuzzier logic, but introducing probabilities makes no difference: if e.g. to evaluate your optimal choice you assign p[1], p[2] as the probabilities the Being will make those choices, you then observe that

$$p_1 = prob(\eta_{11}p_1 + \eta_{12}p_2 > \eta_{21}p_1 + \eta_{22}p_2)$$

which means $p_1 = 0$ or $p_1 = 1$ (since the inequality is true or false)

but if $p_1 = 1$ then $p_2 = 0$ and

$$p_1 = prob(1000000 > 1001000) = 0$$

so $p_1 = 1$ implies $p_1 = 0$.

Similarly $p_1 = 0$ implies $p_1 = 0$, etc., so the argument is identical, and we are driven to the fixed point $p_1 = 0$, $p_2 = 1$.

{…}

It isn't difficult to translate the logic of the situation into a computer program. Any language that permits recursive definition will do; in Lisp, e.g., taking the expected payoff as a function of choice, and taking another function with no

arguments to represent the prediction, the relevant definitions are:

```
(defun payoff (choice)
   (cond
      ((eq choice 'one)  (if (eq (prediction) 'one) 1000000 0))
      ((eq choice 'both) (if (eq (prediction) 'one) 1001000 1000))
   (T nil)))

(defun prediction ()
   (cond
      ((> (payoff 'one)  (payoff 'both)) 'one)
      ((> (payoff 'both) (payoff 'one))  'both)
   (T nil)))
```

(You may read the first as "if the choice is one box, if the prediction was one box then the payoff is one million, else it is zero; if the choice was both boxes, if the prediction was one box the payoff is one million plus one thousand, else it is one thousand; these are all the possibilities," and the second as "if the payoff for choosing one box is greater than the payoff for choosing both boxes, predict 'one'," etc.)

Naturally though these compile into working code (their mutual dependence does not in itself entail that they are ill-defined), if you try evaluating either function the result is a stack overflow, i.e. the computational equivalent of smoke pouring out from under the hood[12] or a loud feedback squawk.

---

[12] One of the earliest electromechanical logic machines was built by two students of Quine, William Burkhart and Theodore Kalin, in 1947, and solved problems in the propositional calculus by evaluating truth tables. "It is interesting to note," says Gardner [*Logic Machines and Diagrams*, New York: McGraw-Hill, 1958, p. 130] "that when certain types of paradoxes are fed to the Kalin-Burkhart machine it goes into an oscillating phase, switching rapidly back and forth from true to false. In a letter to Burkhart in 1947 Kalin described one such example and concluded, 'This may be a version of Russell's paradox. Anyway, it makes one hell of a racket.'"

But there's an ambiguity here as well, related to the distinction in programming semantics between call-by-name and call-by-value. — Which is actually much more complicated, there are a bewildering variety of possible strategies for evaluation[13] — but: in evaluating an expression one may have the option of performing a syntactic transformation upon it first; for instance, some algebraic manipulation that may simplify it.

(The traditional [Leibnizian] interpretation of the derivative, as a quotient of infinitesimals, involves a kind of call-by-name strategy: you compute the ratio *before* allowing the values to go to zero. — Unsurprisingly, this procedure becomes difficult to analyze in cases involving nested series of limiting processes — the order does of course affect the result — and in this sense the subtleties first encountered in the formulation of the calculus prove symptomatic of deeper difficulties in the theory of computation.)

In general if a program terminates on all inputs the results are the same, but if it does not the order of execution can make a difference.[14]

Lisp functions usually employ call-by-value,[15] so that every subexpression is evaluated and the result is passed to the routine that calls it, but conditional expressions are an exception, and

---

[13] See Harold Abelson and Gerald Jay Sussman, *Structure and Interpretation of Computer Programs*. [Cambridge: The MIT Press, 1996.] — The discussion that follows is drastically oversimplified; there may be no subject more complex than the semantics of programing languages.

[14] This is already true in the lambda calculus, see for instance section 5.8 of Joseph Stoy, *Denotational Semantics*. [Cambridge: MIT Press, 1977.]

[15] Evaluation can be turned off with the (metalinguistic) quote function (and turned back on within the scope of a quote with a backquote). These devices are useful, e.g., when writing programs that can rewrite their own text while they are running. — I suppose I should illustrate this by exhibiting a Lisp program which on execution erases its own text and thus can only produce an output if and only if it does not, but let's leave that as an exercise for the reader.

whether a function terminates or does not can depend on the order in which the tests are performed.

This is because an expression like (reverting to a more Pascal-like syntax)

    if ([Boolean] test) then A else B

is evaluated by first evaluating the test, and then evaluating A or B according to whether it returns true or false. Thus if one knew for some other reason that the test always returns true, the expression can be replaced with A; while otherwise B might be some function that fails to terminate.

E.g. one might define

    f(x) = if (x = 1) then 6 else f(2)

which depending on how x is defined elsewhere in the program will return 6 or a stack overflow.

What this means in the case at hand is that the functions given above might be defined in some other way to avoid the ungrounded recursion. One might, e.g., try something like

```
(defparameter *payoff-matrix* '((1000000 1001000) (0 1000)))

(defun payoff (i j) (nth i (nth j *payoff-matrix*)))

(defun best-choice ()
   (cond
        ((and
           (> (payoff 0 0) (payoff 1 0))
           (> (payoff 0 1) (payoff 1 1)))
           'one)
        ((and
           (> (payoff 1 0) (payoff 0 0))
           (> (payoff 1 1) (payoff 0 1)))
           'both)
        (T nil)))

(defun players-move () (best-choice))

(defun beings-move () (best-choice))
```

which produces the (unconvincing) result

```
7 > (players-move)
BOTH
7 > (beings-move)
BOTH
```

{…}

Let's amend the rules slightly to construct a more consistent paradox: say that a clause in the contract specifies that if you take the second box only and the Being predicts you will take both, you can sue for breach of infallibility. Then Mister Mastermind will have to settle out of court for, say, $100,000, and the revised matrix reads:

Being's choice 2	$100,000	$1000
Being's choice 1	$1,000,000	$1,001,000
	Your choice 1	Your choice 2

This eliminates the bogus attractor, and reduces the paradox to the pure Cretan form: if you *should* take one box, you *should* take two boxes; if you *should* take two boxes, you *should* take one box. — Moreover if you start at (1,1) and alternate moves with the Being, you proceed through every node in the matrix: (1,1) entails (2,1) entails (2,2) entails (2,1) entails (1,1). — Surely that's more like it.

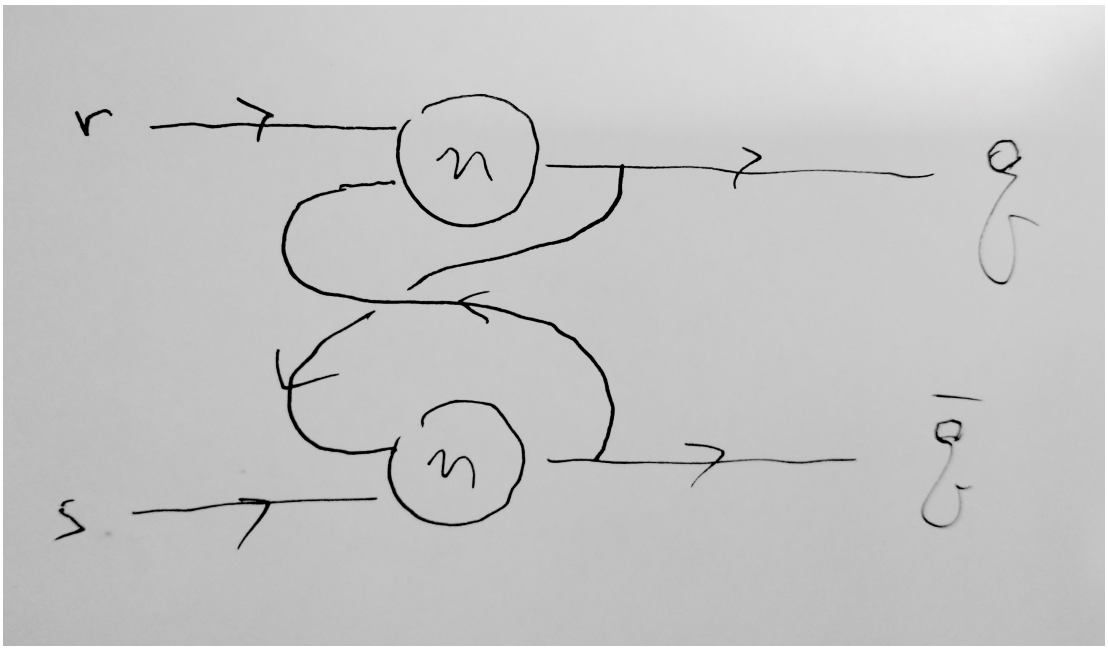
Taking p = "*should* take one box" and q = "*will* take one box" — thus not p = "*should* take both boxes", not q = "*will* take both boxes" — the sequential logic of the situation can be diagrammed as follows:

p	q	p'	q'
F	F	T	F
F	T	F	F
T	F	T	T
T	T	F	T

{…}

Apparently, then, for simple problems anyway, a program of elementary complexity suffices, the logic of the situation can be modeled by a Boolean circuit, and the self-referential part of it, the part that seems to require the tachyonic telephone, is expressed by feeding back the outputs into the inputs.

In other words another simple kind of temporal paradox might be expressed by a circuit such as the following:



where $\eta(r, s)$ is the NAND function):

r	s	NAND(r,s)
F	F	T
F	T	T
T	F	T
T	T	F

which in Lisp is:

```
(defun nand (p q) (not (and p q)))
```

This circuit is the famous flip-flop.[16] Far from being paradoxical, it is an extremely useful electronic component,[17] because its stable states can be used to store information.

If this is coded recursively as

```
(defun top (r s) (nand r (bottom r s)))

(defun bottom (r s) (nand s (top r s)))
```

the result, unsurprisingly, is a stack overflow, but if you observe that a false input to a NAND gate always produces a true output, then the (syntactically)[18] equivalent definitions

[16] One of them, anyway. There are many variations on the theme.
[17] It was used as a storage device as early as the codebreaking Colossus of 1943.
[18] I.e., insinuating a call-by-name strategy.

```
(defun flip-flop-top (r s)
   (if (not r) t (nand r (flip-flop-bottom r s))))

(defun flip-flop-bottom (r s) (flip-flop-top s r))

(defun flip-flop (r s)
   (list (flip-flop-top r s)
         (flip-flop-bottom r s)))
```

terminate on inputs (F,F), (T,F), and (F,T).

The behavior of the circuit is summarized by the truth table:

r	s	q	q'
F	F	T	T
F	T	T	F
T	F	F	T
T	T	q	q'

which can be interpreted as follows: the values (q, q') are assumed given (grounding the recursion) and are to be maintained as complements; thus the input (F, F) is forbidden. The inputs (F, T) and (T, F) flip the values of (q, q') — thus the name. The input (T, T) produces a well-defined result if (q, q') are given, and leaves them unchanged.

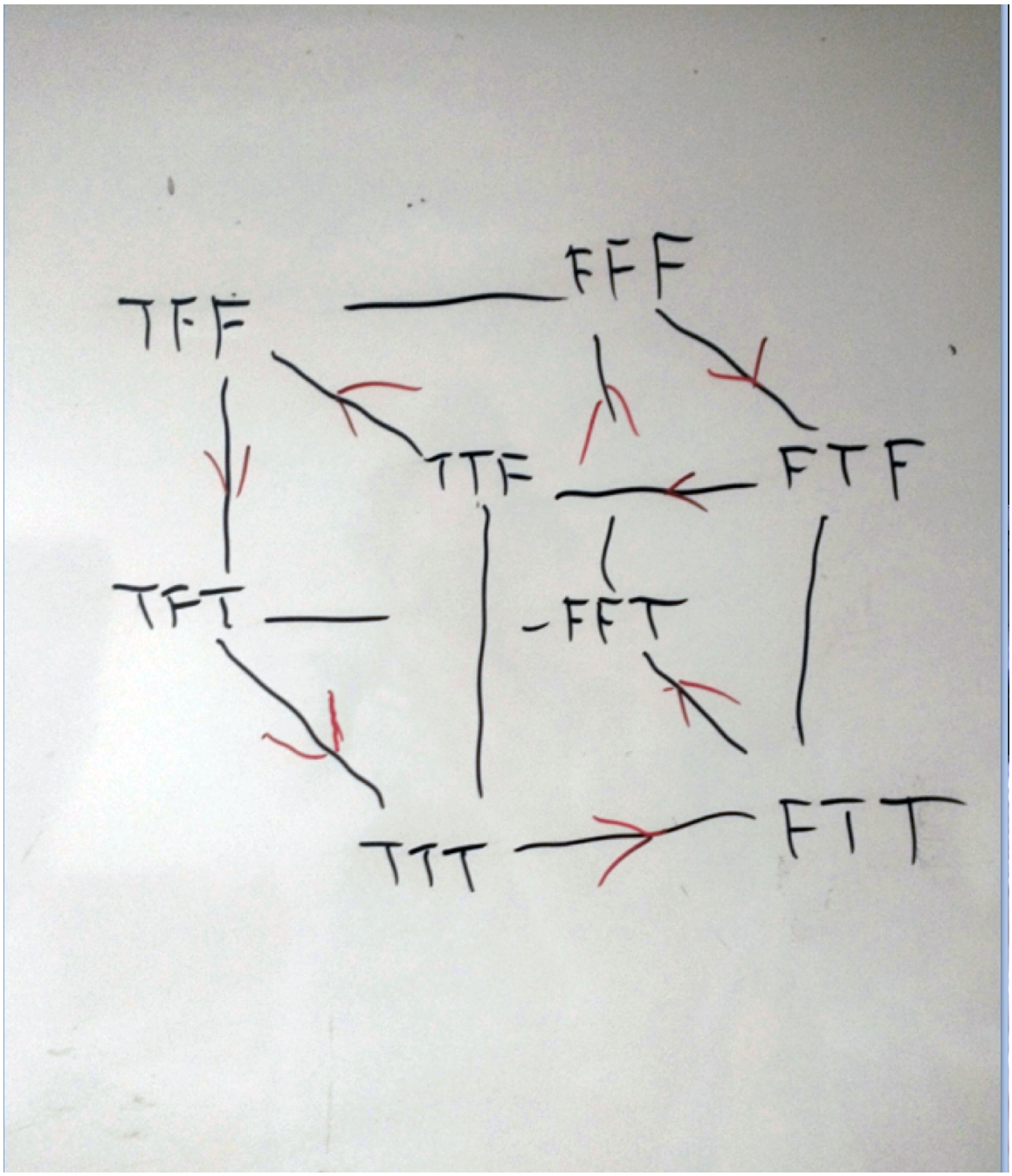
In other words what seems an impenetrable conundrum to the philosopher is a trivial commonplace for the electrical engineer. I suppose this should be embarrassing.

{…}

Why stop with two players? why not a game with three? Again we suppose the Player, the infallible Being, and as a third party introduce — not a Cartesian Demon, exactly — a Prankster, let us say, who may as well be female, who can intervene in the game as follows: she has the power (say by hacking into their computers)[19] to reverse the Being's perception of what move the Player has made/will make, and when she does so she also reverses the Player's judgment as to which payoff is greater than the other (switches "<" to ">" and vice-versa); since it pisses her off when the Being tries to weasel out of forking over the million bucks, she only flips this switch when he predicts the Player will take both boxes and intends to pocket the money himself — but then doesn't bother flipping the switch back until there is another change from Being-predicts-one to Being predicts-two.

Interpreting this sequentially, and supposing the Player, the Being, and the Prankster takes turns in that order, the following state transition diagram results:

[19] Since the point of this entire discussion is that the players can all be replaced by computers, there is no loss of generality.



i.e.

TTT —> FTT —> FFT —> FFF —> FTF —> TTF —> TFF —> TFT —> TTT

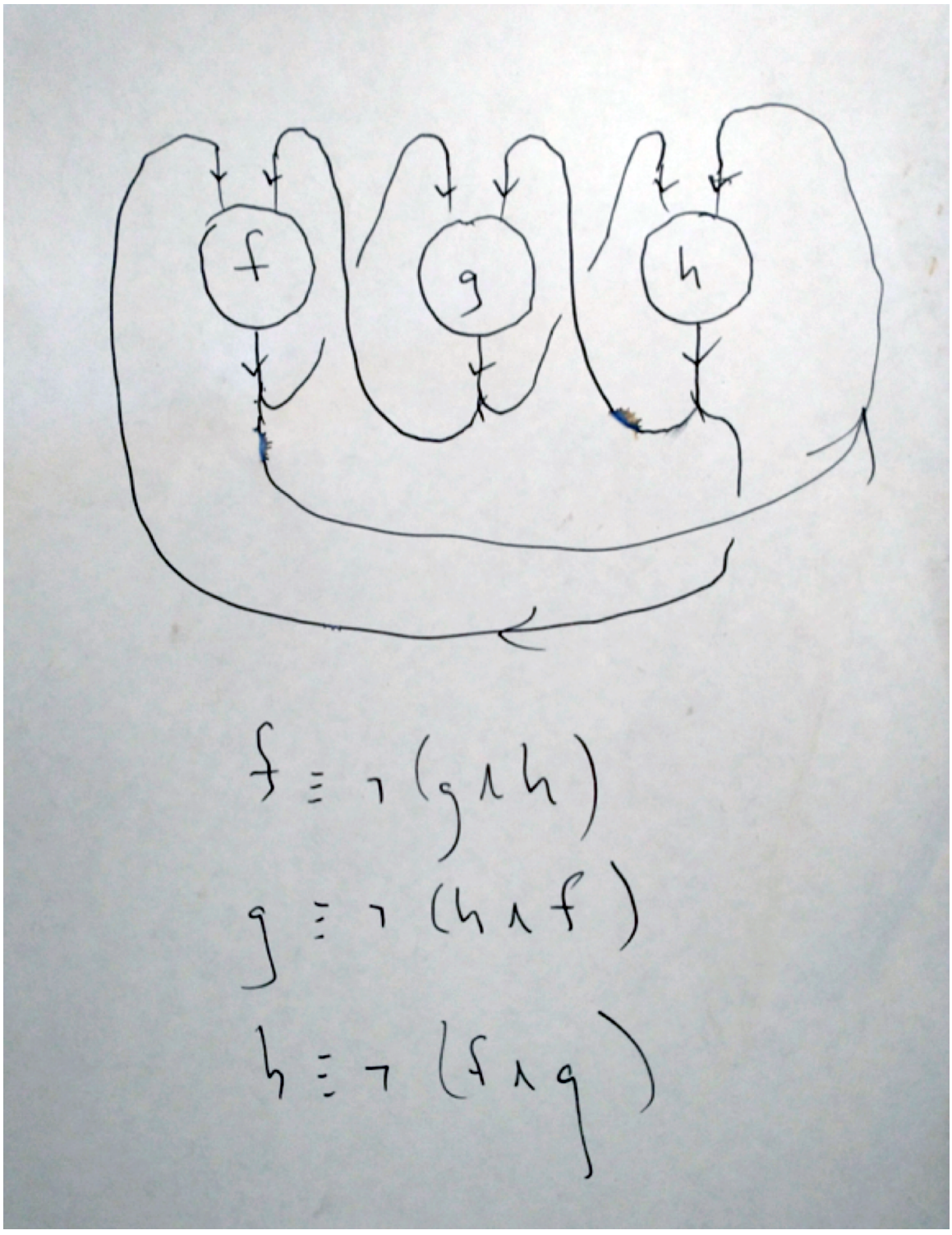
with the truth table:

x	y	z	x´	y´	z´
F	F	F	F	T	F
F	F	T	F	F	F
F	T	F	T	T	F
F	T	T	F	F	T
T	F	F	T	F	T
T	F	T	T	T	T
T	T	F	T	F	F
T	T	T	F	T	T

where x = “Player does/does not take one box”, y = “Being does/does not predict Player takes one box”, and z = “Prankster does/does not confuse the perceptions of the other two”. So here there are no fixed points and only one cycle.

{...}

A better illustration of the general case (make up your own story) is provided by the Boolean circuit:



where f, g, h are all NAND gates.

The mapping of inputs to outputs this defines is

x	y	z	x´	y´	z´
F	F	F	T	T	T
F	F	T	T	T	T
F	T	F	T	T	T
F	T	T	F	T	T
T	F	F	T	T	T
T	F	T	T	F	T
T	T	F	T	T	F
T	T	T	F	F	F

which has the stable states

FTT —> FTT
TFT —> TFT
TTF —> TTF

while

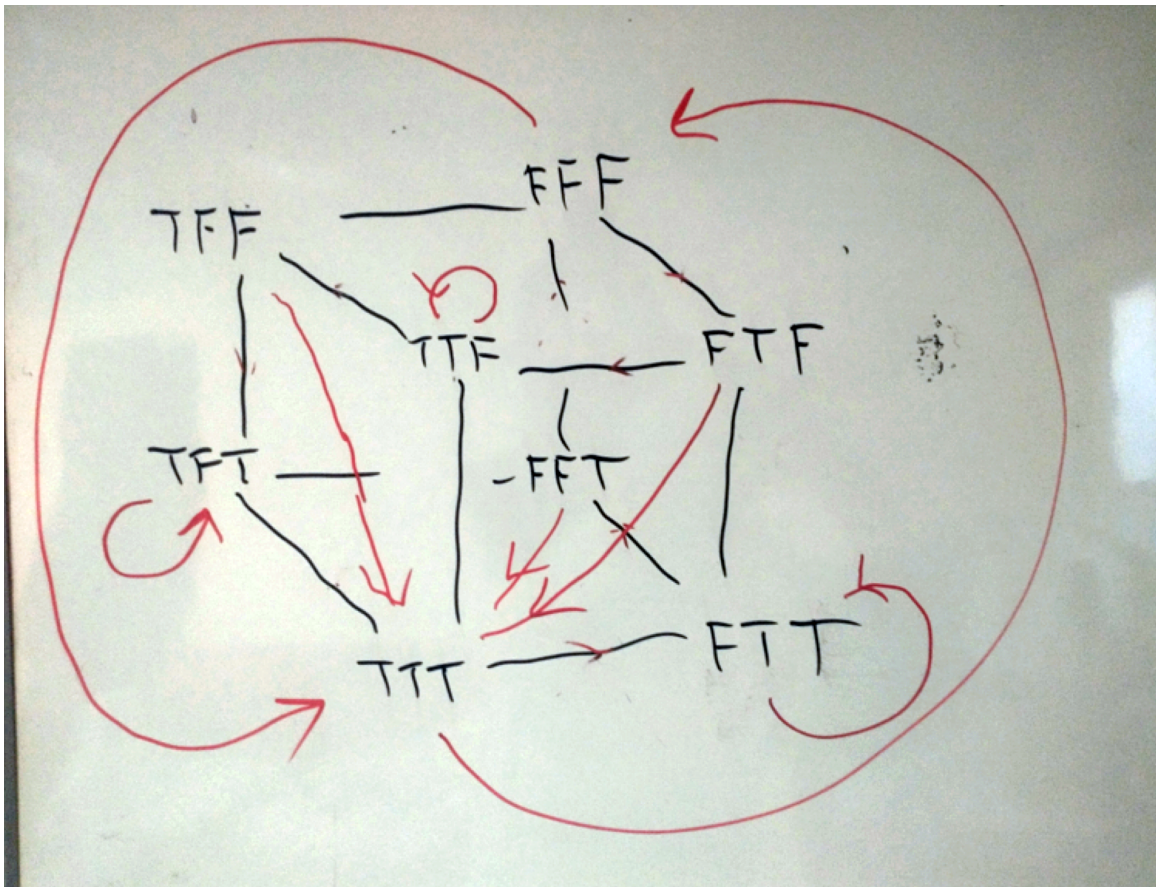
{FFT, FTF, TFF} —> TTT

and

TTT —> FFF —> TTT —> …

is a cycle.

This may be summarized by the state-transition diagram:



There are, in other words, three fixed points — representing, if you will, the self-consistent world histories in which nobody shoots his grandfather — and three states driven to an attractor, a fundamental cycle that flips back and forth between all on and all off.

(Note incidentally that oscillating behavior may be exactly what you're trying to produce with the feedback loop; in fact the original use of the flip-flop was as a circuit to produce square waves, thus the alternative designator "multivibrator".)

The generalizations to any number of players and arbitrary directed graphs with Boolean functions at the nodes[20] is straightforward. The arrangement of fixed points and cycles is largely arbitrary, but it is obvious that, for any finite network, any evolution must terminate in either a fixed point or a cycle.[21] So in this simplified model of the history of the world, at least, some form of eternal recurrence is guaranteed.

{…}

A related application of the idea of a Boolean circuit whose outputs feed back into its inputs is that of the genetic regulatory network: groups of related genes are governed by sets of rules of the form "A is expressed if B is expressed and C is not expressed or …"; Stuart Kaufmann conjectures that the stable states of such networks (self-consistent sets of choices) correspond to cell types, has shown empirically that the number of such fixed points in random networks is proportional to the square root of the number of nodes, and presents evidence that the number of cell types (which would correspond to given sets of genes being turned on and off) is correlated to the size of the genome in roughly this fashion for a variety of species.[22]

{…}

If we think of these networks as dynamical systems — we might extend the Boolean model by making the functions probabilistic, for instance — questions about the stability of equilibria become

---

[20] Waving my hands here. A bit of care in the definitions is required.
[21] Similar theorems hold for continuous dynamical systems, given the appropriate topological preconditions (some form of compactness). — The situation for infinite discrete Boolean networks is more complicated, containing as it does the case of the two-state cellular automata studied by Wolfram among others, and can entail difficulties like the halting problem for Turing machines.
[22] Kauffman, Stuart. *The Origins of Order*. Oxford: Oxford University Press, 1993.

significant: how do they behave under perturbations? What is the expected lifetime of a (quasi)stable state? — etc., etc.

But in any case the problem of the temporal loop has been reduced to the problem of feedback; meaning the idea isn't as absurd as it first sounds, though (as always) complications appear on a closer analysis.

{…}

Irwin translates all this into postmodernism:

> As Johnson sees it, taking a position on the numerical structure of the tale means, for Lacan and Derrida, taking a numerical position, choosing a number, but that means playing the game of even and odd, the game of trying to be one up on a specular, antithetical double. And playing that game means endlessly repeating the structure of "The Purloined Letter" in which being one up inevitably leads to being one down. For if the structure created by the repeated scenes in the tale involves doubling the thought processes of one's opponent in order to use his own methods against him—as Dupin does with the Minister, as Derrida does with Lacan, and as Johnson does with Derrida—then the very method by which one outwits one's opponent, by which one comes out one up on him, is the same method that will be employed against oneself by the next player in the game, the next interpreter in the series, in order to leave the preceding interpreter one down.[23]

Admittedly cute. But I think Vizzini said it better.

---

[23] John T. Irwin, "Mysteries We Read, Mysteries of Rereading: Poe, Borges, and the Analytic Detective Story." *MLN* Vol. 101, No. 5, Comparative Literature (Dec. 1986), pp 1168-1215.