Toward an Understanding of the Preservation of Goals in Self-Modifying Cognitive Systems

Ben Goertzel
August 30, 2008

A new approach to thinking about the problem of "preservation of AI goal systems under repeated self-modification" (or, more compactly, "goal drift") is presented, based on representing self-referential goals using hypersets and multi-objective optimization, and understanding self-modification of goals in terms of repeated iteration of mappings. The potential applicability of results from the theory of iterated random functions is discussed. Some heuristic conclusions are proposed regarding what kinds of concrete real-world objectives may best lend themselves to preservation under repeated self-modification. While the analysis presented is semi-rigorous at best, and highly preliminary, it does intuitively suggest that important humanly-desirable AI goals might plausibly be preserved under repeated self-modification. The practical severity of the problem of goal drift remains unresolved, but a set of conceptual and mathematical tools are proposed which may be useful for more thoroughly addressing the problem.

Introduction

One of the deepest and most difficult conceptual problems related to Artificial General Intelligence (Wang et al, 2008) is the question of the persistence of AGI goal systems under repeated systemic self-modification. This conceptual problem is also potentially an ethical problem of a rather large magnitude, in the sense that a powerful AGI system with an unpredictably drifting goal ystem could be a dangerous thing.

To articulate the problem a little more clearly: Suppose one builds an AGI system with a certain goal, G, but then allows the system to modify itself in a radical way, for instance perhaps rewriting its own source code (or making other comparably radical modifications ... the precise definition of what constitutes a "radical modification" depends on the particulars of the AGI system implementation). What happens if the modifications cause the system to no longer possess the goal G?

Of course, one can try to mitigate against this by making the preservation G part of the goal G (explicitly or implicitly). But still, what if the system makes a mistake, and accidentally modifies itself in a way that causes G to mutate into a slightly different G1. And what if G1 then similarly mutates into G2 ... so that eventually the AGI system's successors are governed by a goal completely different from (perhaps even contradictory to) the original?

There are certainly examples reminiscent of this phenomenon in human psychology, such as – to choose a striking though unpleasant example -- mentally unwell parents whose initial goal of protecting their children from physical harm gradually morphs into a goal of killing their

children.

Clearly this problem of “goal drift under repeated self-modification” is a significant theoretical issue in the analysis of advanced AGI systems. How serious a practical problem it will be, is hard to say at this point: experimental investigation is difficult since contemporary AGI technology falls far short of allowing radical goal-driven self-modification of complex AGI programs; and contemporary computer science theory does not allow detailed rigorous mathematical analysis of complex AGI systems operating within realistic resource constraints (the closest we have are analyses such as Hutter's (2004), which treat powerful, theoretical AGI systems operating on unrealistically generous computational resources).

My goal in this brief essay is not to solve the problem of goal drift – my intuition is that this is not a job for any one paper, but rather a job for a whole research programme, and one that is currently barely at its inception. I consider here a specific type of goal, which I call a *self-referentially survival-oriented goal (or SRSO goal)* and then discuss

> the circumstances under which this sort of goal may be useful
> mathematical conditions under which this sort of goal is likely to be successfully self-preserving in the face of internal error and environmental uncertainty

The treatment is mathematically sophisticated but not mathematically rigorous -- no theorems are proved – rather I point out a direction where I suspect interesting, nontrivial, and potentially pragmatically relevant theorems may be found.

In order to motivate these rather abstract ideas pragmatically, I will place some focus on a specific qualitatively-defined goal that I call Helpfulness, which involves pursuit of human happiness, freedom and growth. I define a SRSO based on Helpfulness, called $G_{Helpfulness}$, and explore some of its potential properties. A long-term goal of the reearch programme suggested here would be to show that plausible formulations of $G_{Helpfulness}$ exist and are robust with respect to goal drift. We are not very close to being able to make such a demonstration right now; but even so, my suggestion is that this is a promising direction tin which to further theorize.

Implicit Versus Explicit Goals

Before progressing to discuss self-referential goals, it's worth pausing to briefly discuss the more basic question: "What does it mean for system S to possess goal G over time interval T?"

The way I'd like to define this is relative to some other observer O, who is hopefully a smart guy.

I thus define: "S possesses goal G over time interval T if, to O, it appears that the actions S takes during T are significantly more probable to lead to the maximization of G than random actions S would be able to take."

So, for instance, if Ben is hitting himself on the head with a brick between 9AM and 10AM today, it may appear relative to Moshe that Ben is acting with the goal of causing himself

suffering.

I have called this an "implicit goal" in prior writings (Goertzel, 2006). What it basically means is "what the system S appears to be maximizing." In real systems, explicit and implicit goals are not always aligned. The explicit goal is, roughly speaking, the goal the system declaratively thinks it's working toward, in its phenomenal self (if it has one; see Goertzel, 2006; Metzinger, 2004). The implicit goal is the goal it looks like it's trying to achieve based on an external analysis of its behaviors. The discussion in this essay is about implicit goals. One may argue that in a healthy, near-optimally-productive mind implicit and explicit goals will be well-aligned, but that is beyond the scope of this essay.

Formalizing the Notion of a Self-Referential Goal

A Quick Review of Hypersets

In order to conveniently formalize the notion of a self-referential goal, I will make use of the mathematical concept of a hyperset. Hypersets are reviewed thoroughly in (Barwise and Moss, 2004) and more accessibly in (Barwise and Etchemendy, 1987).

In essence, a hyperset is a set that can contain itself as a member, or contain other sets that contain it as a member, etc. This also allows one to create functions that take themselves as arguments, and so forth. For instance, with hypersets we can create constructs such as

$$A = \{A\}$$

or

$$B = \{C\}$$
$$C = \{B, C\}$$

or

$$f(f) = f$$

or

$$g(h,i) = 2$$
$$h(g) = i$$
$$i(h,g) = 3 * g(h,i)$$

and so forth.

The rigorous construction of hypersets requires some fairly advanced set theory, in which the Axiom of Foundation is removed from standard ZF set theory and replaced with more complex alternatives. However, the practical use of hypersets in semantics and computer science does not generally require explicit use of advanced set theory methods.

Of course, simple hyperset constructs like the ones depicted above may be modeled using

simple finite graphs, without need for recondite set theory. What is interesting, however, is the way hyperset theory allows one to use loopy graphs in combination with the standard algebra and semantics of set-membership. Hyperset theory shows that the algebra and semantics of membership is logically compatible with loopy membership graphs.

Formalizing Explicitly Self-Referential Goals

Using hypersets one may explicitly formulate goals that refer to themselves.

For instance, when my son Zarathustra was in the third grade, an overzealous school policy required that each student begin each day by writing down their goals for that day. Zar started with "My goal is to reach my goal" but then quickly advanced to "My goal is not to reach my goal." Semi-formally, we might write these as

G = "Act in such a way as to achieve G"

G = "Act in such a way as to not achieve G"

These goals of Zar's are amusing but not particularly useful. The self-referential goal that I'll focus on here is, instead, the following:

G = "Act in such a way as to maximize A, and also maintain G as your goal"

This is what I call a *self-referential, survival-oriented goal,* or *SRSO goal*. Note that the "survival" in the SRSO term refers to survival of the goal itself, although this also implies survival of the system manifesting the goal.

One can assign probabilistic truth values to self-referential goals using the formalism of infinite-order probabilities (Goertzel, 2008), which may be a fruitful path to pursue in future, but won't be taken up further here.

To simplify discourse, I will refer to the subordinate goal A in the above as "the concrete objective." So we are looking at a goal that involves seeking to achieve a concrete objective, and also to maintain the goal of achieving this concrete objective. Also, I will reformulate the above as

G = ד(A,G)

(defining the ד operator as ד(A,G) = "Act in such a way as to maximize A, and also maintain G as your goal").

We may then consider the mapping ג defined by ג(x) = ד (A,x), and use the notation G_A to refer to the fixed point of the ג mapping. G_A is an SRSO goal.

Matching Hypersets to Finite Dynamical Systems

It may not be clear how a finite software program can meaningfully be said to manifest a goal that is defined as a hyperset, given that hypersets, from a set-theoretic perspective, is a very large infinite set. However, just as many hypersets can be mapped into finite graphs for some purposes; similarly, they can be applied in the context of finite software programs.

One way to think about this is to consider the "unraveling" of the hyperset as an iteration, e.g.

$$Z = \{Z\}$$

is unraveled as

x
{x}
{{x}}
{{{x}}}
...

So, if we have a dynamical system so that the successive elements in the unraveling are observed at succeeding points in time (for any specific x), then the hyperset itself may be viewed as a pattern in the system, in the sense that "Unravel the hyperset Z={Z}" is a way of compressing a set of observations in the system.

Similarly, we may unravel an SRSO goal as

$$\lambda(x)$$
$$\lambda(\lambda(x))$$
$$\lambda(\lambda(\lambda(x)))$$
...

...

and if successive elements in this unraveling are observed in a dynamical system (for any specific x) then we may say that the SRSO goal is a pattern in the system, so that it makes sense to say the system is "following" the SRSO goal.

The problem of matching an SRSO goal against a dynamical system (such as an AI system) then reduces to the problem of matching elements of the "unraveling" series

G0 = "Act so as to maximize A"
G1 = "Act so as to maximize A and maintain G0"
G2 = "Act so as to maximize A and maintain G1"

against a dynamical system. As each of these series elements is an ordinarily-defined mathematical function, basically this just means we need to be able to assess the degree to which dynamical system is solving a certain optimization problem at a certain point in time. The only real subtlety here is dealing with the "and." I suggest that the most useful way to interpret the "and" in the definition of an SRSO is as defining a multi-objective optimization problem.

Interpreting SRSO's and their Approximants Using Multi-Objective Optimization

Multi-objective optimization (Steuer, 1986) is the process of simultaneously optimizing two or more objectives subject to certain constraints; it is an approach one takes when one has two different objectives in mind, both of which are important, but which are in some contexts conflicting. Of course, one approach that can be taken here is just to make a single objective function that is, say, a weighted sum of the two objectives. But this approach is not always ideal in practice, partly because of the intrinsic arbitrariness of the numerical weight involved. In multi-objective optimization, instead, one typically seeks a solution for which each objective has been optimized to the extent that if one tries to optimize it any further, then the other objective(s) will suffer as a result.

Interpreting G1 above in terms of multi-objective optimization, then, a system with G1 as goal will seek to balance the two goals of:

- Maximizing A
- Making sure that it retains the goal of maximizing A

It will seek to find a way of existing so that:

- Increasing its expected efficacy at maximizing A any further would lead to a decrease in its expected efficacy at retaining the goal of maximizing A
- Increasing any further its expected efficacy at retaining the goal of maximizing A, would lead to a decrease in its expected efficacy at maximizing A

Similarly, interpreting G above in terms of multi-objective optimization, a system with G as goal will seek to balance the two goals of:

- Maximizing A
- Making sure that it retains the goal G

It will seek to find a way of existing so that:

- Increasing its expected efficacy at maximizing A any further would lead to a decrease in its expected efficacy at retaining the goal G
- Increasing any further its expected efficacy at retaining the goal G, would lead to a decrease in its expected efficacy at maximizing A

The Potential Applicability of the Theory of Iterated Random Functions

A couple questions then arise from the foregoing considerations:

- for what A does the equation G = ⊤(A,G) have a solution?
- in cases where it does have a solution, how might this solution be found in practice?

One approach to exploring these questions, I suggest, is to use the theory of iterated random functions (Diaconis and Freedman, 1999).

The simplest general method of studying the convergence of recursive iterations is using the classical Contraction Mapping Theorem: in general, if one has an iteration

$$x_{n+1} = f(x_n)$$

acting on a metric space, and one can show that if f is a "contraction map" in the sense that

$$d(f(x), f(y)) < k\, d(x,y)$$

for some k<1, then it follows that f has a fixed point x so that

$$x = f(x)$$

and that the series $\{x_n\}$ as defined above will converge to such a fixed point.

This is nice but the condition of being a contraction map is often too strong in practice. Thus weaker conditions have been developed, including a theorem which states in effect that if a probabilistic function f is contracting *on average*, then its iterates will converge with probability one. I will call this the Average Contraction Mapping Theorem, or ACM Theorem.

In the context of our study of self-referential goal systems, one natural question then becomes: under what circumstances will the mapping λ defined by $\lambda(x) = \tau(A,x)$ be contracting on average? Of course, this is not the only way that λ may have fixed points or tractable convergence behavior, but it's one simple way that this could happen, so it seems worth exploring.

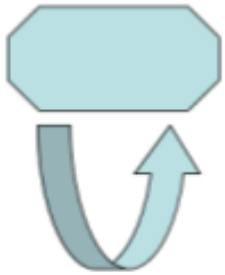
Worth noting as well, is that even if the iterates of λ converge to some fixed point, this doesn't matter unless these iterates are actually realizable in some physical system. Convergence does not imply realizability. However, in some circumstances, one may be able to show that if the iterates λ of are all realizable, then the limit is also realizable. This relates to the notion of syntactic versus semantic distance, to be explored below.

Defining Metrics on Hyperset Space

To apply the ACM to SRSO goals we need to define a metric on the space of finite hypersets.

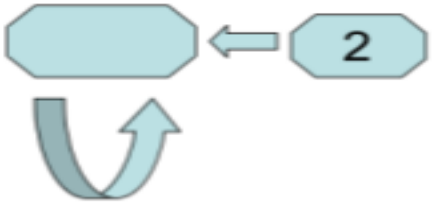
In fact we will define two different metrics, a syntactic metric and a semantic metric, and then discuss their interrelationsip.

For the purpose of discussing metrics, we will restrict our attention to finitely-given hypersets (hypersets that correspond to finite apg's, in the terminology of (Barwise and Moss, 2004)). Note that any finite traditional (well-founded) set corresponds to a finite labeled tree. We may then represent any non-well-founded, finitely-given hyperset as an infinite tree, corresponding to the unraveling of the hyperset. For instance Z={Z} corresponds to an apg that looks like



versus an infinite tree that looks like

Similarly, Z = {Z,2} has an apg



-- and the reader is invited to draw the infinite tree themselves!

Syntactic Distance on Finitely-Given Hypersets

To calculate syntactic distance between these (possibly infinite) trees, some preparatory mechanism is needed. First, assign each node of each tree an address according to a standard scheme, in which the address of a node encodes the path from the root to the node. Then, create an enumeration of the nodes of each tree (separately), in such a way that when node N has a longer address than node M, N always occurs later in the ordering.

Next, we refer to the various finite tree edit distances reviewed in (Bille, 2005): this gives us a way to measure distance between finite trees. The question is how to deal with infinite trees. We present a unified approach to finite and infinite tres, which builds on the methods Bille discusses but extends immediately to infinite trees.

Given two trees (finite or infinite) Q and R, we proceed as follows. For each integer n, we create the subtrees Q_n, R_n consisting of the nodes of Q and R with position less than or equal to n in the enumeration of nodes, plus al links joining these nodes. Then, we define the metric

$$d_n(Q,R) = d_{edit}(Q_n,R_n)$$

where d_{edit} denotes the standard edit distance on finite trees. Then, we define the overall metric

$$d(Q,R) = c^{-1}d_1(Q,R) + c^{-2}d_2(Q,R) + c^{-3}d_3(Q,R) + ...$$

where c>1, which is a metric because it's a linear combination of metrics, and which is convergent because of the exponential weighting.

Note a key property of this metric: differences between two trees count for less if they're further away from the roots of (both of) the trees. This is similar to the absolute-value metric on the number line, where differences between decimal strings count for less if they're further away from the decimal point.

Obviously there are many other syntactic metrics one could define on the space of finitely-given hypersets, and I'm not claiming the metric defined above is particularly excellent or special; it is proposed mostly just to have *something* plausible in place, so as to make the considerations of the following sections mathematically meaningful.

Semantic Distance on Finitely-Given Hypersets

Next, to define semantic distance on finitely-given hypersets, we refer back to the notion of evaluating the applicability of hypersets to finite systems. Given any hyperset H, and any dynamical system D, we may consider the degree to which H is a pattern in D's trajectory as a fuzzy membership degree $א_{traj(D)}(H)$. We may then define the distance between two

hypersets H1 and H2 via

$$d(H1,H2) = d'(\aleph_{traj(D)}(H1), \aleph_{traj(D)}(H2))$$

where d' is an appropriate metric defined on fuzzy sets, for instance one of the Hausdorff metric variants described in (Brass, 2002). This basically says that the distance between two hypersets is the distance between the sets of dynamical systems in which they are patterns (and the fuzziness is used to properly weight systems according to the degrees to which the hypersets are patterns in them).

Relationship Between Semantic and Syntactic Distance

The question of the correlation between syntactic and semantic distance is an obvious one. In general one cannot expect this correlation to be high, and there are ways to increase it, such as normalizing the syntactic form of hypersets, relatedly to the normalization of program trees described in (Looks, 2006).

However, given the way semantic and syntactic distance have been defined, the following does seem clear: for any epsilon, there exists some delta so that if the syntactic distance between H1 and H2 are less than delta, then the semantic distance between H1 and H2 is less than epsilon. That is, semantic distance depends continuously on syntactic distance. This fact has some potentially useful consequences that will be mentioned below.

Contraction and Convergence Involving Self-Referential Goals

So, having metrized hyperset space in a relevant way, we can now proceed to talk about convergence, contraction and so forth on spaces of goals that include both traditional goals and self-referential goals like SRSO's.

We thus return to the question raised above: under what circumstances will the mapping λ defined by $\lambda(x) = \tau(A,x)$ be contracting on average, under the semantic distance? That is: we're asking to understand the circumstances under which it will be the case that, for some $k<1$, it holds on average that

$$d(\tau(A,G) , \tau(A,G1)) < k\, d((A,G), (A,G1))$$

Will this be the case for any useful concrete objectives A? If so, for which ones?

And in the cases where contraction does hold, of course, we must then ask whether the convergence is realizable: there are some cases where $\aleph_{traj(D)}(\lambda)$ is the empty set, but λ is converged to ... this is a case of convergence that is of no practical use. It is not hard to see that realizability of the finite iterates of λ implies realizability of λ itself.

Note that a mapping need not be a contraction over the whole domain to which it is applicable. It may be a contraction only over a certain portion of its domain. In terms of

iterations, this means that some initial values may lead to convergence, whereas others may not. Generally a fixed point will have a basin of attraction, and choosing an initial condition within this basin will cause the iteration to converge to the fixed point (see Devaney, 1987, for a review of basic dynamical systems theory concepts). What this means here is that, for each A, we want to know not only whether ג has a fixed point, but also whether the basin of this fixed point is reasonably large. If the fixed point exists but has an extremely small basin, then it may be that moderately small perturbations to G could push the system out of the basin of attraction, so that it won't converge back to the fixed point. On the other hand, if the fixed point exists but has a reasonably large basin, then moderate-sized perturbations to G won't push the system out of the basin, so it will return to its stable fixed point after a bit of iteration, recovering from the perturbation.

Intuitively, if A and G are "independent", then one would expect that on average we'd have

$$d(\tau(A,G) , \tau(A,G1)) = d((A,G), (A,G1))$$

That is, perturbing G by a certain amount to yield G1, would cause ד(A,G) to get perturbed by a generally (on average) comparable amount to yield ד(A,G1).

But this is not what we want! What we want is for reasonably small variations in G to lead to *smaller* variations in ד(A,G). But how can we achieve this? In what cases is this possible?

Basically, what we want is for A and G to interdepend, in such a way that if G and G1 differ by (say) 1%, then ד(A,G) and ד(A,G1) will on average differ by only (say) 0.9%. This may be the case for some concrete objectives A and not others. Conceptually, we may label concrete objectives of this nature as "coherentizing objectives."

The relation between syntactic and semantic distance also needs to be considered, in discussing convergence. Suppose we have a series of hypersets that converges syntactically, then does it have to converge semantically? The answer seems to be that it does, due to the continuous dependence of semantic distance on syntactic distance. On the other hand, if a series of hypersets does not converge semantically, then it cannot converge syntactically either.

A Few Informal Examples

To make all this a little clearer, let us consider intuitively what it might mean in the context of a few real-world concrete objectives.

Suicide as a Goal

Firstly, an obvious case where there is no realizable fixed point G_{A1} would be the case

A1 = "Kill myself without possibility of resurrection"

abbreviated as Suicide. In that case we have the series

G0 = “Act so as to maximize Suicide”
G1 = “Act so as to maximize Suicide and maintain G0”
...

and the potential fixed-point $G=G_{Suicide}$, defined by

G = “Act so as to maximize Suicide and maintain G”

and we have some problems.... $G_{Suicide}$ does exist as a hyperset, and syntactically speaking, the iterates {G0,G1,G2,...} converge to $G_{Suicide}$. But except for G0, none of the iterates can actually be embodied in any real system, so the semantic convergence is trivial in the sense that $\aleph_{traj(D)}(G_{Suicide})$ is the empty set for every D.

Helpfulness as a Goal

Next, let's consider a more cheerful concrete objective, defined as

A2 = human happiness, growth and freedom

and summarized in a convenient word as Helpfulness (as an aside, a philosophical argument for the meaningfulness of this particular concrete objective is given in Goertzel, 2006).

We may then look at, e.g.

G0 = “Act so as to maximize Helpfulness”
G1 = “Act so as to maximize Helpfulness and maintain G0”
G2 = “Act so as to maximize Helpfulness and maintain G1”
...
G = “Act so as to maximize Helpfulness and maintain G”

which at least gives some intuitive promise of being a realizably convergent series.

One might wonder, looking at this example, whether the whole hyperset setup isn't completely redundant. After all, isn't the best way to maximize Helpfulness going to be to preserve one's desire to do so? If one loses one's desire to maximize Helpfulness, then one won't be around to be Helpful in the future ... so it would seem, by this argument, that G1 should be implicit in G0 as defined above, so we don't need to proceed any further down the series.

This is a naïve way of thinking, however. Recall from above, that to evaluate SRSO's and their approximants as applied to specific dynamical systems, one interesting route is to think in terms of multiobjective optimization. There are going to be many, many different ways that look roughly equally plausible for achieving a complex goal like Helpfulness. The effect of using G1 alongside G0, if all goes well, will be to usefully bias the search toward ways of achieving Helpfulness that also involve maintaining G0. For instance, incorporating G1 will automatically bias one against solutions to G0 that involve martyrdom ... or the transformation of the system into a Helpfulness-ignoring theorem-prover.

Thus, the fixed-point G_A of the equation G=ꞇ(A,G) may be viewed intuitively as a kind of biasing mechanism that biases the cognitive system's ongoing search through all the possible ways of maximizing A. One interesting question – demanding detailed mathematical exploration -- is whether introducing this kind of biasing mechanism actually gives the system a higher chance of maximizing A. The answer to this apparently depends on the statistical properties of the environment the system maximizing A is operating in, relative to the particular properties of A.

To put it as simply as possible (and with the explicit admission that we are now hand-waving even more furtively), the question is whether we anticipate the system getting into situations where solutions involving abandoning its concrete objectives will appear *deceptively good*. If so, then the biasing involved in G_A is probably useful (if it can, in fact, be achieved for the concrete objective A in question).

On the other hand, if we anticipate the system getting into situations where solutions involving personally abandoning its concrete objectives will appear *deceptively bad* as ways of actually achieving its concrete objectives, then the biasing involved G_A in is actually counterproductive.

Based on this reasoning, it seems the choice of whether to use goals of the form G_A as defined above is essentially a guess about the statistics of the situations the cognitive system is likely to encounter in future.

Emphasizing the Difference Between Goal Achievement and Goal Maintenance

It is important to fully understand the difference between goal achievement and goal maintenance. One is about what happens in the world ("maximize G as an outcome") and the other is about what happens inside the system ("maintain G as a goal").

It seems commonsensically that, in all but pathological cases, the best way to maximize G as an outcome is going to be to maintain G as a goal. But logically, this is not *always* going to be the case. And, trusting commonsense intuition about what is "pathological" can be dangerous. For instance, fractal sets and curves were long considered pathological, until it was realized that in many useful senses, they are the majority case... The line of thinking in the paper is oriented toward avoidance of occurrence of pathological cases with bad outcomes.

For instance, consider the case of Bob, whose goal G is to keep his children happy. Suppose he decides the best way to keep his children happy is to commit suicide (in a way that looks accidental) so they can get his $10M life insurance policy. Maybe he is right and this is the best way to achieve G.

However, suppose Bob commits suicide on January 1, 2009. Then during 2010

will Bob possess the goal G? No, he will be dead.

So, it is quite possible that acting (during time interval T) in such a way as to maximize G (over all time, or over a specified time period that is part of G), can cause S to put itself in a state where, over other time intervals T1, it is *not* acting in such a way as to maximize G.

Thus, maintaining goal G as a goal, is a different thing than achieving goal G.

The AI Researcher's Dilemma

It may be helpful to pose a similar example that does not involve suicide. Suppose Bob's goal is to create a human-level AI; and he thinks he knows how to do it, but the completion of his approach is likely to take him an indeterminate number of years of work, during which he will have trouble feeding himself.

Consider two options Bob has:

A) Spend 10 years hacking in his basement, based on his AI ideas

B) Spend those 10 years working as a financial trader, and donate 50% of his profits to others creating AI

Suppose Bob judges that:

- In option A, he has only 30% chance of creating human-level AI during those 10 years, but 90% chance of continuing to want to do so for the whole 10 years
- In option B, he has 50% chance of creating human-level AI during those 10 years, but only 60% chance of continuing to want to do so for the whole 10 years

So, B exceeds A at "odds of achieving the goal", but A exceeds B at "odds of maintaining the goal".

This does not involve suicide, but it does involve goal-relevant change of the self. And note that this scenario is meaningful even if: Independently of whether human-level AI is created, the world is guaranteed to end after the end of the 10 years. So, the issue is not about the time-sensitivity of goals, but rather about the difference between goal achievement and goal maintenance.

Might Helpfulness Be A Coherentizing Objective?

One overall conceptual message of the above analysis is fairly clear: If we want to create systems whose goals are highly likely to be preserved under self-modification, even in the

face of various possible sorts of errors or deceptive situations, then one direction worth exploring is the creation of concrete goals that are “coherentizing,” in the sense defined above.

That is, to enable semantic convergence to a nontrivial, realizable fixed point, we want to create concrete goals A so that the similarity between “Achieving A and B” and “Achieving A and C”, is on average greater than the similarity between “Achieving B” and “Achieving C.” Specifically we want this to be the case for B and C that are related to preserving the capability to achieve A.

Exploring this sort of property in the context of real AI goal systems will be challenging, yet doesn't seem impossible. Intuitively, I find it plausible to hypothesize that (some sensible formalization of) the goal of Helpfulness as defined above *is* a coherentizing concrete objective, with a reasonably large and nicely-shaped attractor basin.

To partially communicate where this intuition comes from, I'll briefly flesh out a “toy example” involving Helpfulness. Let's say G0 is “maximize Helpfulness”; and let's say that in order to achieve G2, we are considering two possible categories of ways of preserving our future ability to execute G1: H2 and H2'. Maybe H2 is to lock ourselves in a box so the bad guys can't capture us and rewire our goal-systems; and H2' is to lock ourselves in the same box but with a one-way window so we can see the bad guys coming. Then, H2 and H2' have a certain similarity, which may be assessed according to some appropriate metric on the space of procedures (hopefully a semantic metric, which consideres the expected outcome of carrying out H2 and H2' in various situations, with some empirically reasonable probability distribution over the space of situations).

Suppose however that some ways of carrying out H2 or H2' are less compatible with G0 than others. For instance, one may lock oneself in the box without a wireless Internet connection, or with one; and, perhaps, having the wireless connection allows one to better be Helpful to humans. The question is: Is the similarity between “variants of H2 that most effectively support G0” and “variants of H2' that most effectively support G0” *greater than* the similarity between H2 and H2' variants generically? If so, then we may possibly have a case where λ is contracting on average. Intuitively, it would seem that this is going to be the case, because the requirement to directly support G0 places some systematic constraints, and the various procedures fulfilling these systematic constraints are likely to be (very roughly) clustered together in procedure-space.

In other words, I suggest (hands waving vigorously!) that it seems intuitively likely that some well-crafted formalization of the SRSO goal corresponding to the concrete objective of Helpfulness is contracting on average ... hence that there *is* a self-referential self-preserving goal corresponding to the Helpfulness concrete objective ... and that this goal can be converged to via repeated iterations based on the initial goal of Helpfulness.

Obviously, this sort of fuzzy, intuitive evocation (I don't even want to call it an “argument”) doesn't take us very far – for instance, it doesn't give us any hint at all about the size or shape of the attractor basin of the $G_{Helpfulness}$ goal, if indeed such a goal does exist. But as warned at the beginning, my goal in this essay is not to be either rigorous nor definitive. The goal is to point out some interesting directions for potential investigation.

Conclusion

Essentially nothing has been resolved in the above discussion. What I hope is that I have raised some interesting questions. My central goal here has been to replace vague conceptual questions about goal preservation in self-modifying systems with semantically similar questions that are at least somewhat more precise, so as to gain a better understanding of what concrete research directions may be useful to follow, if we wish to come to grips with the highly critical issue of goal drift.

The research programme suggested here could fail in at least two different ways. It could succeed as interesting mathematics, but fail to connect to real AGI practice, due to the difficulty of formalizing real-world AGI goals, or the difficulty of obtaining theorems with enough generality to be plausibly heuristically applied to incompletely-formalized goals. Or, it could fail as mathematics, for instance by proving sterile in the sense that no nontrivial theorems result from the formal structures and conceptual questions posited.

It could also succeed in at least two different ways. It could succeed mathematically, but produce disturbing conclusions, such as the conclusion that there is really no way to create useful AGI goal systems that are contracting-on-average or have any other mathematical properties likely to result in long-term goal stability. Or, in the best case, it could succeed mathematically and produce inspiring conclusions, telling us something useful about which AGI goal systems are likely to resist the potential plague of goal drift.

No formal analysis will ever provide any kind of guarantee regarding the behavior of advanced AGI systems. The mapping of theory onto practice is inevitably theory-laden, so that correct theorems don't always mean what one expects about the empirical world. Furthermore, there is an irreducible uncertainty in dealing with any AGI system that is significantly more intelligent than ourselves. However, the ideas proposed here are ventured in the spirit that *some* understanding is almost surely better than *none*. Given the potentially dramatic importance of the issue of goal drift, it seems to behoove us to energetically pursue any analytical approach that appears to have nontrivial possibility of yielding significant insight.

References

- Barnsley, Michael (2006). Superfractals. Cambridge University Press.
- Barwise, Jon and Lawrence Moss (2004). Vicious Circles. Stanford: CSLI Press
- Barwise, Jon and John Etchemendy (1987). The Liar. Oxford University Press.
- Bille, Philip (2005). A Survey on Tree Edit Distance and Related Problems. Theoretical Computer Science, volume 337(1-3), pages 217 - 239
- Brass, Peter (2002). On the nonexistence of Hausdorff-like metrics for fuzzy sets, Pattern Recognition Letters, vol. 23, no1-3, pp. 39-43
- Devaney, Robert (1987). Chaotic Dynamical Systems. Addison-Wesley.
- Diaconic, Persi and D. Freedman (1999). Iterated Random Functions. SIAM Review, 41 1:45-76
- Goertzel, Ben (2006). The Hidden Pattern. BrownWalker Press.
- Goertzel, Ben (2008). Infinite-Order Probability Distributions, submitted for publication

- Hutter, Marcus (2004). Universal Intelligence. Springer.
- Looks, Moshe (2006). Competent Program Evolution. PhD Thesis at Washington University, St. Louis.
- Steuer, R.E. (1986). Multiple Criteria Optimization: Theory, Computations, and Application. New York: John Wiley & Sons, Inc
- Wang, Pei, Ben Goertzel and Stan Franklin (2008). Proceedings of AGI-08. IOS Press.