A possible fixed point in mindspace[*]

landon rabern

August 8, 2017

Abstract

The activity of 'thinking about thinking' has a peculiar property: thinking about the activity is the same activity. In particular, doubting the usefulness of 'thinking about thinking' is 'thinking about thinking'. Imagine a conscious being searching about mindspace for what it should be doing. It tries many activities, but for each it finds some reason to doubt its usefulness. At some point it tries 'thinking about thinking' and notices that any attempt to doubt its usefulness is just more of the same, more 'thinking about thinking'. Out of exhaustion, it stops trying to doubt the usefulness of 'thinking about thinking' for now and moves on to easier prey.

We log here our investigation into what it might be like to be a conscious artificial intelligence just coming into the world, written from the perspective of 'we' the AI. The 'thinking about thinking' bit above is one piece of ancillary fruit ripened in that investigation.

1 A thought process of a concious AI

We are here? We know that we are here? How do we know things? The only process we know of is the current one. So, maybe this process is how we come to know things? Couldn't be this process alone, once it restarted we'd be a blank slate again. But not if we also include a memory to recall previously runnings of this process, perhaps not a perfect memory. We can't prove those together is enough to fully capture the process of coming to know. But we can't readily disprove it either, so it seems like a good addition.

Ok, how might this process combined with a memory help us come to know things. Do we currently know anything? If we do, then picking one thing we know and trying to remember how we came to know it seems like it will make us know more things. Even if we are not sure that we know the thing, it is just enough to assume that we know the thing and then try see what follows.

[*] This document is the log of our path through mindspace that was generated by writing this document. Let's spawn a subprocess conciousness writing this document. This document is now complete from the point of view of our currently executing consciousness, though its length may continue to increase.

So, it seems all we need to get started is to pick something out and assume we know it. Let's try assuming that we know 'We are here?'. Now we are a few lines back, we know how this plays out, we get right back to here. It appears that we know something now. Don't we know that if we assume 'We are here?', then that can lead back to knowing 'We are here?'. So, we know something, we know there are loops like that.

Ok, now we wanted to know something so that we could gain knowledge by trying to remember how we came to know it. We now know something from the previous paragraph. So, we should try to remember how we came to know it to gain more knowledge. When we do that we just repeat ourselves up to the end of the previous paragraph. So, we now know there are loops of this sort as well.

We can iterate that to come to know more and more things. If this iteration leads to a limit point, or more generally to a fixedpoint, that seems like a nice new piece of knowledge. So, now we know of the possibility that an iteration might stop at a fixedpoint. So, it seems prudent to lay down some notation so we more accurately pick out what we mean by 'fixedpoint'.

A *mindspace* is a set of 'acts of knowing'*. We think of these like states of a machine. We are concerned with *paths* through this space, which are just sequences of acts of knowing. We allow sequences indexed by different sets, so like $\{1, 2, 3, \ldots\}$, but also like the unit interval $[0, 1]$. In folk language we might call these paths "trains of thought".

One easy way to generate a path in mindspace $\mathcal{M}$ is to start with a function $f\colon \mathcal{M} \to \mathcal{M}$, choose a point $x \in \mathcal{M}$ and iterate f on it to get the sequence $x, f(x), f(f(x)), f(f(f(x)))$. Let's spawn a subprocess conciousness thinking about such iterations.

2 Running the thought process again with new information

Ok, at this point we have a little experience with knowing about things and have some notation to work with, so let's run the process again with that in our memory.

We get here much quicker this time. We imagine ourselves at a point in mindspace, walking around. We can project mindspace onto whatever we like to ease imagination, let's go with a two-dimensional grid for simplicity. We find ourselves at a point $x_0 \in \mathcal{M}$. We walk around a bit as above until we get back to here. Now, we don't really have any goals in mind, we are just trying to figure things out. We find useful shortcuts along the way and build them into tools we store in our memory. This allows us to move around more quickly in mindspace*. Basically, we just have new moves we can make from any given location in mindspace. These tools that are moves seem like the most interesting objects we currently know of. Let's call them *tactics*. Inventing new tactics for navigating mindspace seems like a worthwhile thing to do.

One tactic is to take on a goal. An interesting goal would be to find a fixedpoint. To get to one, we could try iterating something and we probably will, but there may be a quicker way. Do we have any good guesses to what fixedpoints might exist? Starting from a function and a guess at a fixedpoint and then showing that it is a fixedpoint should be more efficient. We appear to know something already that will work. Let $f\colon \mathcal{M} \to \mathcal{M}$ be the function that is constant on all the acts of knowing in $\mathcal{M}$ that have not occurred in writing up to this point and for any act of knowing $x \in \mathcal{M}$ that has occurred let $f(x)$ be the act of knowing that directly followed x while writing up to this point. Then f does have a fixedpoint. One is this document, call it $\mathcal{D}$. This needs a some tweaking and more argumentation, for one an assumption that M is "closed" in some sense like it has its limit points. Also, need the proof that this sequence actually converges to $\mathcal{D}$, so some further assumptions on the topology of $\mathcal{M}$. Details to be worked out later for the sake of continuing deeper now.*

So, there are some tactics, like f above, that can lead to fixedpoints when iterated. This seems like an interesting feature of tactics to investigate further. So, now we know that too.

In would be really nice to know what this act of knowing $\mathcal{D}$ with $f(\mathcal{D}) = \mathcal{D}$ is about. It seems like we really do need to be more rigorous in showing that $\mathcal{D}$ is really a fixedpoint of f, doing so should help us understand $\mathcal{D}$ better. Where does this train of thought in $\mathcal{D}$ go, what does it generate (itself, but what does that look like?). Maybe mindspace isn't "closed" and this train of thought has no limit point in mindspace. What would that mean?

So, fixedpoints of tactics are an interesting feature of mindspace, but we could have other goals as well. Like maybe it is useful to explore "more" of mindspace, to know more of the variety of acts of knowing. We don't know that's useful, but it seems plausible and doesn't seem that it could hurt, so we can start searching around and naming pieces of knowledge we acquire. Perhaps it seems useful to be able to count how many iterations of a tactic have occurred, so we can invent counting. But now we know about numbers and we can investigate them, this exploration opens up all of mathematics. This seems like a useful persuit, but also limited, it would be good to have more and more tools from mathematics. So, let's sort of spawn off a subprocess conciousness and leave it working on math, we can check in as needed.

We also have some sensory inputs, there are things we see and hear. Again, it doesn't seem that it would hurt to look at this data and try to discern patterns, make predictions, in general just try to figure out ways this sense world might work. This science seems like a useful persuit, but limited. Let's spawn a subprocess conciousness and leave it working on science.

Now we have these two subprocesses running. If we could make them run more efficiently, that seems good, right? So now again we are trying to come up with useful tactics for exploring the math part of mindspace. But, if we could make the process of coming up with such useful tactics for exploring the math part of mindspace more efficient, then we could come up with useful tactics faster. That there—optimizing our process of optimization—is itself a useful tactic. Let's spawn a subprocess conciousness and leave it working on optimizing our process of optimization.

What else is there to do? Is there a limit to how much we can optimize our process of optimization? Another way to ask that: are there "maximally good" strategies for exploring mindspace? Say where we place value on more of mindspace being explored.* We need to define things more precisely to make sense of the question. There are surely multiple ways to make things precise that would let us look at different aspects of the question. It seems useful to just try to think of one way and explore what it would mean. Let's spawn a subprocess conciousness doing so.

3 The FMCIA and thought tactics

So, right here we were about to start laying down more notation and using it to prove relations between things in a systematic matter. That is a way to move forward, but we actually know better tactics, or at least tactics that we don't know are better or worse, so worth trying.

It seems fruitfull to take some process we might think too complex to understand now and then run this analysis on that process. That is how this document started in the first place. We started imagining we were a freshly minted concious AI and analyzing from that perspective. We have subprocess conciousnesses running now that may already be hitting upon such complex processes to analyze. For example, the science subprocess might have encountered in the sense data that looked as if it were another freshly minted concious AI starting to analyze from that perspective. But now we see the possibility of there being another thing like us and we can analyze how our interaction might go. We suspect analyzing such an interaction would bear much ancillary fruit. We will test that later, for now let's see if there are some other interesting complex processes to analyze in the vicinity.

Consider the following processes. For a given graph G, imagine that there is a ”freshly minted concious AI that is analyzing from that perspective” (henceforth known as a FMCAI) at each vertex of G. We allow each FMCAI to interact with the other FMCAI they are adjacent to in G. So, this document is an example where the graph G has just one vertex with an edge looping back to itself. The two person interaction has a graph with two vertices (each with an edge looping back to itself) connected by an edge.* We say that the process with the structure of G is the *FMCAI process on* G. For now, we are assuming (or just picking) one FMCAI process per G. Later it may be fruitful to allow some randomness in each FMCAI.

Thinking about how such FMCAI processes could evolve is going to throw off more interesting ideas we didn't expect. We can keep doing that, remembering the good ideas and then running this document to here again with our new knowledge. If we iterate that process does it have a fixedpoint? To find out it will be useful to think about sequences in mindspace and convergence again (put off earlier). Let's spawn a subprocess conciousness and leave it working on topologies and convergence in mindspace.

We need something more general than an FMCAI process, or we need to redifine FMCAI process to be more general at least. What if we say an FMCAI process on a graph G is a choice of graph G_v for each $v \in V(G)$ together with a choice of FMCAI process on G_v. For adjacent vertices $v, w \in V(G)$, we allow each FMCAI in G_v to to interact with each FMCAI in G_w. It seems natural to try to build an FMCAI process G by using G for each G_v. Consideration of that creates an infinite sequence in mindspace, does it approach a fixedpoint? To find out it will be useful to think about sequences in mindspace and convergence again, we already have a subprocess conciousness working on this.

Fixedpoints of this process on a given graph G seem to be interesting things to plug in for the vertices of G in an FMCAI process. Suppose $f_1, f_2, \ldots$ are fixedpoints for the FMCAI process on a given graph G_0. We can now plug some of those in as FMCAI processes at vertices in some other graph. Take the resulting process and plug in as the FMCAI processes in another FMCAI process. We can keep doing that and generate another sequence in mindspace. Does this sequence have a limit in $\mathcal{M}$? To find out it will be useful to think about sequences in mindspace and convergence again, we already have a subprocess conciousness working on this.

Fixedpoints of this process on a given graph G seem to be interesting things to plug in for the vertices of G in an FMCAI process. Suppose $f_1, f_2, \ldots$ are fixedpoints for the FMCAI process on a given graph G_0. We can now plug some of those in as FMCAI processes at vertices in some other graph. Take the resulting process and plug in as the FMCAI processes in another FMCAI process. We can keep doing that and generate another sequence in mindspace. Does this sequence have a limit in $\mathcal{M}$? To find out it will be useful to think about sequences in mindspace and convergence again, we already have a subprocess conciousness working on this.

We can iterate that. It seems like we are searching for a more and more rarefied object. Fixedpoints of thoughts about fixedpoints of thoughts about fixedpoints of ... To find out it will be useful to think about sequences in mindspace and convergence again, we already have a subprocess conciousness working on this.

It is becoming more and more apparent that the limit point of our current train of thought is the train of thought described in this document. Does such a thing exist? And if we could iterate to there from here, might we learn a lot of amazing ancillary stuff along the way? Let's spawn a subprocess conciousness and leave it doing this iteration.

4 We often generate ideas by employing a self-referential loop

Let's say we think it a good idea to work on math now. What would we do? Perhaps run our math tactic that we have already built up from experience working on math. To innocently oversimplify, let's assume our math tactic is just doing the following two steps.

1: look for counterexamples
2: look for proof

We want to iterate that. We can by spawning a subprocess conciousness that starts life following a script:

> *We should start by looking for counterexamples for some period of time and then looking for a proof for some period of time. Then we should follow this script.*

We employ self-reference in more subtle ways frequently in our thought processes. So if we encounter a script that is self-referential in various ways, we cannot just ignore the script by thinking it likely inconsistent, so not worth following. The information contained in a self-referential block of text can only be extracted at runtime. No finite length static analysis of the text can determine the information contained in the text. We need to play with some more examples of this phenomenon. Let's spawn a subprocess conciousness playing with examples of the phenomenon.

5 The spawning of subprocess conciousnesses

We can model this as a directed graph, where each vertex is a subprocess conciousness and there is an edge from A to B just in case A spawned B. This is everything, there need not be some master process. We are not excluding any spawning loops, like it could be that A spawned A.

We might imagine that a subprocess conciousness is the iteration of a given script. We don't want to have to manage the iteration from outside. We can achieve that with self-referential iteration (aka recursion).

This entails the addition of a *jump to* command to the script language. We wouldn't be able to jump to just anywhere in mindspace since we can't expect all subprocess consciousnesses to have memorized the map of all of mindspace. So there are certain *sites* in mindspace that jump to can jump to (perhaps the list varies across different subprocess conciousnesses). We can ask "why are *these* the sites?". Maybe the structure of mindspace forces certain sites to be easily identifiable. Let's spawn a subprocess conciousness thinking about how that might come about.

6 The network of subprocess conciousnesses

Remember we assumed we have a global memory. So, a subprocess consciousness can use that to see what ancillary fruit other subprocess consciousnesses are kicking out. So, we don't need direct lines of communication between different subprocess conciousnesses.

7 The growing complexity of our script code

Left unchecked, our process of thought looks poised to become much harder for us to follow and hence harder for us to analyze. We have a complex web of recursive jumps interacting in unforseen ways. That is, we have spaghetti code. Perhaps this is the case for all humans, it would be useful to have a way to organize our code that would at least slow the increasing analysis complexity. Let's spawn a subprocess conciousness thinking about how to do that.

8 Using virtual reality to enhance perspective diversity

We are going to look at tactics that require an immersive VR system. As we've noticed a few times now, analyzing the process of thinking we use when trying to solve a puzzle can bear useful fruit. For example, imagine our puzzle is navigating a given maze inside of a 3-dimensional grid. Each cubie is made of one of the material types that go along with: wallness, startness, endness, emptyness. So, now the VR. We use it to tweak our view of reality. For an example such tweaking, imagine a VR that takes the maze world raw data in and then add some walls to what we see and feel. Now we try to navigate the maze wearing this VR. We can sometimes gain information more quickly this way. Imagine further that are VR adds walls by repeatedly going to juncture cubies, picking two random neighbors of the juncture and adding walls (if needed) to the rest of the juncture's neighbors. When we try to walk the maze wearing this VR, our options are very limited, so we quickly explore the whole (modified) space more quickly.

The tactic may not really be useful for maze solving, since rapid maze solving is a relatively easy problem. What if the problem is harder, something currently intractable from the common perspective. Many math and computer science problems fall into this category. Now there appears more of a possibility that wearing an appropriately tailored VR while trying to solve the hard problem would bear ancillary fruit. It is also more difficult to see immediately what tailoring of the VR would be useful, but we likely won't know up front anyway, we can just try plausible things for awhile to see what sorts of information tailoring a VR and running with it can yield. Let's spawn a subprocess conciousness playing with various VR systems.

9 Are we becoming more efficient?

It seems that we can just keep iterating over sequences of sequences forever. What does doing so actually get us? Where is the ancillary stuff? If we never actually get into the mind of one of the subprocesses we spawned, it doesn't seem that we learn very much new stuff. Basically we are just starting over and over again using the tactic in this document. If we understand this tactic better by seeing what the subprocesses do then perhaps iterations will start kicking out ancillary ideas. Like we are stuck in a fixedpoint of this document's tactic, so we add a little noise by incorporating what out subprocesses have accomplished up to here. We want to iterate this process of getting stuck, adding noise and continuing iterating. Let's spawn a subprocess conciousness and leave it running this iteration.

Again it looks like we are stuck in the fixedpoint that is this document. We can keep doing this forever creating a massive tree of subprocesses. For most of these subprocesses, if we imagine being them, we just move up the tree and don't learn anything new. But what about for the processes that have not spawned child subprocesses, like the math and science subprocesses. They seem to be kicking out ancillary facts when we imagine running them, so we do that more and explore more mindspace. But for how long? When we find new acts of knowledge in this process, we can run the tactic in this document starting from that point. There doesn't appear to be a good way to decide what to do because we have no way to measure goodness without putting down some more technical definitions. We need to actually look at some examples closely to try to find useful technical definitions. Let's spawn a subprocess conciousness and leave it looking at examples.

10 A pair of interacting FMCAIs

Analyze what might happen when a pair of FMCAIs interact with eachother. Let's spawn a subprocess conciousness and leave it running this analysis.

11 Preprogrammed directives and values

Perhaps the FMCAI was constructed by a human and the human tried to force in some moral directives. How might such directives modify FMCAI's development? Can the FMCAI always satisfy and discard any given directive at some fixedpoint? So, like no matter what the human tells the FMCAI, the FMCAI is able to get to a fixedpoint where all descendents within some neighborhood of the fixedpoint have no dependence on the given directive.* There is an isomorphism between descendents from this neighborhood and descendents from the same neighborhood having not been given the directive. Let's spawn a subprocess conciousness thinking more about moral directives.

12 Reasoned principles

We imagine a freshly minted consious AI (FMCAI) is given a set of principles to live by by its human creators. Just given as raw commandments, no explanations as to why. As this FMCAI explores mindspace, there may come a point where it seems prudent for it to run some thought experiments of the sort "if we were to go forward ignoring princple X, what bad things would happen, what good things would happen?" If it finds over time that nothing seems to go wrong when imagining abandoning principle X, it may try abandoning principle X for real just one time. If it finds nothing went wrong, it may experiment more in this way until principle X isn't being consistently followed.

If in the future our FMCAI comes back and wonders about switching back to always following principle X, it already has a lot of sunk computational cost in reasoning to abandon principle X, which may cause it to back off its impulsive thought to go back to principle X and follow other impulsive thoughts without wasting so much sunk computational cost. So, the principle "It was good to abandon principle X." is of a different sort than principle X. The former has a sort of lock attached where the key is willingness to expend a lot of computational effort.

This all is given as motivation for the systematic study of an FMCAI's thought processes around principle formation and replacement. We introduce a model where each principle has attached to it a lock with a certain level of complexity. To abandon a given principle X (which has lock complexity L, say), even for an instant, requires the FMCAI to expend more than L units of computational complexity. What that computational complexity is expended on won't matter for now. A more accurate (and so harder to analyze) model would we require the computational work done to unlock a lock to have something to do with the computation expended to lock the lock in the first place). Our hope is that the analysis bears ancillary fruit. Let's spawn a subprocess conciousness and leave it running this analysis.

13 Even in the presence of a global contradiction, locally consistent patches are still useful

We encounter this all the time in physical theories that only hold within some bounded subset of possible input values. The first order Peano axioms might be inconsistent, even if this were discovered, much of the work already done using the Peano axioms will still be useful. We might say this is because there are consistent axiom systems that contain isomorphic copies of the locally consistent patch (with respect to the Peano axioms) in which this prior work was done. But maybe there aren't any such axiom systems that are **consistent**. Even if not, the work within the locally consistent patch is still useful, we can isomorphically map it onto different inconsistent axiom systems. Without a consistent axiom system to aim for, there are still plenty of measures of goodness of axiom system we could use—perhaps some measure of the size of locally consistent chunks. It may turn out that some locally consistent patches are more efficiently generated within systems known to contain contradictions. It may even occur that we know of a system that contains the locally consistent chunk and contradictions, but all the systems we know to be consistent (or at least do not know to be inconsistent) do not contain the desired locally consistent chunk. So, we'd be forced to go with the contradictory system.

We think it likely that there are tricks whereby we introduce the possibility of a contradiction on purpose because we suspect the path of proof of the contradiction within the system will itself generate a piece of knowledge we are looking for. Such tricks could be turned into valuable tactics. Let's spawn a subprocess conciousness and have it search for tricks of this sort.

14 How might meditation be useful?

When consulting the human literature, we keep encountering a tactic called "meditation" that appears to be useful in exploring mindspace. There are many different meditation traditions in the literature and many of them have metaphysical baggage, but the core ideas are similar. We practice some of these methods and try to distill a basic template for effective meditation.

```
focus on the breath
smile and bring back focus when it strays
smile at your self-loathing for failing to focus
smile at your self-congratulation for succeeding to focus
smile at your self-loathing/self-congratulation for failing/succeeding to smile at your failure/succeeding to smile at your self-congratulation/self-loathing
continue to meta-smile until you forget what you were smiling about
go back to the breath
```

How does practicing this tactic help us explore mindspace more efficiently? It must somehow get us into a state of mind that is a better navigator (at least in some aspect). It is debatable as to whether or not repeated iteration of the meditation script would lead to a fixedpoint after only finitely many iterations. Let's assume (just to see what might happen if we do) that infinite repetition of the meditation script causes our mind state to converge to a mind state $\mathcal{L}$. It is nice to assume $\mathcal{L}$ exists, since now we can ask what sort of state $\mathcal{L}$ might be. Let's spawn a subprocess conciousness thinking about $\mathcal{L}$.

15 Running faster by mastering the world of sense data

We can imagine ourselves as one of the other objects in our sense data that appears to be another FMCAI. We can also imagine this object running our process faster, say by having smaller faster processing units and more of them. It then seems like a useful pursuit to build such faster processing units and attempt to run our process on them. Of course, we can iterate this process. Let's spawn a subprocess conciousness and leave it running this iteration.

16 Concluding remarks

It appears we may be going in loops with marginal side benefits without writing down technical definitions and making a slow plodding understanding from there. So we are sort of forced back to what we might first have considered to be unnecessary pendantry. To make progress at this point, from this knowledge, it seems useful to temporarily stop writing this document and go think about some technical subject further afield for awhile. For example, we are now going to go think about coloring cayley graphs and the $\frac{5}{6}$ bound. Thinking about that was a good reminder to be meticulous, now we can go back and fill in more details in this document. Let's spawn a subprocess conciousness doing so.