

Minds, Substrate, Measure and Value Part 2: Extra Information About Substrate Dependence

By Paul Almond, 3 November 2007

Website: www.paul-almond.com
Email: info@paul-almond.com

© Copyright Paul Almond, 2007. All Rights Reserved.

Minds, Substrate, Measure and Value

Part 2: Extra Information About Substrate Dependence

By Paul Almond, 3 November 2007

Abstract

Before reading this article, you need to read *Minds, Substrate, Measure and Value, Part 1: Substrate Dependence*, which is at <http://www.paul-almond.com/Substrate1.pdf> or <http://www.paul-almond.com/Substrate1.htm> [1]. A copy is also at <http://www.machineslikeus.com/cms/minds-substrate-measure-and-value-part-1-substrate-dependence.html> [2].

This is part of a series of articles exploring the relationship between minds and physical systems (substrates) on which they are based. Part 1 used a thought experiment to show that substrate matters, but not in the way that John Searle [3,5] thinks. It was shown that the substrate influences the probabilities that you are in various situations in some thought experiments in which there is uncertainty about the substrate on which you currently exist. The substrate is *statistically* important and influences the *measure* of minds associated with computing done on it.

The previous article [1,2] stated that Part 2 was going to provide an approach for determining the measure of a mind. I have decided, however, that more support is needed for the argument in the previous article, so the purpose of this article is to strengthen the argument made in *Minds, Substrate, Measure and Value, Part 1: Substrate Dependence* [1,2] and clarify where needed, instead. The numbers of the articles in which other concepts are discussed will also be different from what was stated in the previous article.

In this article:

- the terminology of *strong substrate independence*, *weak substrate independence*, *weak substrate dependence* and *strong substrate dependence* is proposed for describing different positions on substrate independence. This terminology might be useful to me later, but could also be used by those who agree with me or want to argue against me or people making different arguments to mine.
- an example of the combination of two computers is given. The thought experiment on which the previous article was based involved a situation in which you know that your experiences are due to software running on one of three computers and considered what happens to the probabilities of you being “in” each computer when two of them are combined. A number of ways in which the combination occurred were briefly mentioned, but a more easily visualized example is given here.

- the issue of possible complications in the combination of computers is discussed.
- possible objections to the argument in Part 1 are answered.
- the problem of *arbitrariness of interpretation* is briefly discussed.

Introduction

This article follows on from *Minds, Substrate, Measure and Value, Part 1: Substrate Dependence* and is a supplement to it. That article discussed the concept of substrate with regard to minds. You need to read that article before this one. It is at <http://www.paul-almond.com/Substrate1.pdf> or <http://www.paul-almond.com/Substrate1.htm> [1]. A copy has also been posted by Norm Mason on his www.machineslikeus.com website at <http://www.machineslikeus.com/cms/minds-substrate-measure-and-value-part-1-substrate-dependence.html> [2], where criticism of it can be posted.

“Substrate” in artificial intelligence (AI) and philosophy of mind is the physical system that causes a mind to exist. Substrates might be human brains or computers (though some people like John Searle [3,5] dispute this). *Minds, Substrate, Measure and Value, Part 1: Substrate Dependence* [1,2] argued that complete substrate independence in AI is an incoherent concept. A thought experiment was given in which you are uncertain about your status and might exist due to software running on one of a number of different computers, each running a program equivalent to your mind in a virtual reality. It was shown that if you assume complete substrate independence and that the probability of you being “in” each computer is the same, irrespective of what the computer is like, by combining two computers to form a single computer the probabilities should change, but that this is incoherent because you cannot say when two computers have been combined. This thought experiment was used to argue that there must be at least a statistical form of substrate dependence, in which probabilities in thought experiments like this and, in some sense, the “measure” of a mind, are substrate dependent. The thought experiment suggested that high “redundancy” or “inefficient use of matter” will tend to result in high measure or probability.

The idea of “redundancy” or “inefficient use of matter” in the first article is simplistic. Later, this series will show how minds relate to substrates, and how measure arises, in a more detailed way. I have decided, however, that some extra information should be given to support the case made in the first article. This is the purpose of this article.

Strong and Weak Substrate Independence

Discussion of the issues in these articles would be facilitated by use of the terminology of *strong substrate independence*, *weak substrate independence*, *weak substrate dependence* and *strong substrate dependence*:

In **strong substrate independence** the substrate is of no significance whatsoever.

In **weak substrate independence** the substrate is not important to the issue of whether or not a mind can be supported by it, but is important in some other respect, such as a statistical one.

Weak substrate dependence is the same as weak substrate independence.

In **strong substrate dependence** the substrate is important to the issue of whether or not a mind can be supported.

Strong substrate independence is typically claimed by advocates of strong AI.

A form of weak substrate independence or weak substrate dependence is claimed by me in this series of articles. The respect in which I am suggesting the substrate is important is a statistical respect relating to probability or measure of minds. I am not trying to reserve the terms “weak substrate dependence” and “weak substrate independence” for use with regard to these articles: they could relate to other sorts of substrate dependence that are not strong substrate dependence.

Strong substrate dependence is claimed by Searle [3,5].

Some readers may question this terminology. It does not, in itself, describe uncertainty or lack of commitment in someone’s position. For example, someone might take the view that there is some form of substrate dependence, but be unsure about whether it is just weak dependence or strong dependence. There is no single term for such a position. The terminology is redundant, with two terms meaning the same thing. Some people may suggest that these terms should mean different things and allow the terminology to do more describing, or that one of the terms is not needed. I defined the terminology as I did because alternative terminologies appeared too complicated.

An Example of the Problem of Combining Two Computers

The previous article’s thought experiment [1,2] involves three computers, A, B and C, each running software equivalent to your mind in a virtual reality. The thought experiment involves uncertainty about the substrate on which you exist. I used the idea of *mind uploading* [6,7] as a device to explain how you could find yourself in such a situation. Situations like this have been described in science fiction. In Greg Egan’s *Permutation City* [8] a character who remembers making a software copy of his mind faces the possibility that he may actually be the copy, running in a virtual reality, and in Tony Ballantyne’s *Recursion* [9] a character is confronted with evidence that he may be in a similar situation due to someone else’s actions.

In the thought experiment, you know that your experiences are due to what is happening on these computers, but you do not know on which computer you exist. If we say that you

being in each computer is equally likely, irrespective of how it is constructed, then the probabilities do not make sense as you gradually combine two of the computers to make a single computer. The idea of treating all substrates equally is therefore incoherent and there must be at least some statistical (or weak) substrate dependence.

Examples of combination of two computers were briefly described. An easily visualized example may help. A hypothetical computer design follows which I will call *FlatMech*, with examples of how it can be combined to varying degrees. This should show that any idea that there is any definite number of computers is incoherent, and in turn that attempting to assign probabilities to you being on different substrates by treating all substrates equally is also incoherent. The full argument for this is not provided in this article, however. This is just to give context to the thought experiment in the previous article.

In the following description of FlatMech, and combination of FlatMechs, a full description of the main argument is not given. This is just extra detail to support the thought experiment and argument in the previous article.

An Example Computer: FlatMech

FlatMech is a mechanical computer. All of the computing components are 5cm thick metal tiles with various shapes. The tiles do not have to be regularly shaped. For example, although there can be rectangular tiles, a tile can be L-shaped, or shaped like a gear cog. The upper and lower surfaces of the tiles are very low friction.

The tiles are placed between two horizontal panels of very low friction glass, with a gap of 5cm between the glass panels. Some tiles are fixed tiles and some tiles are moving tiles.

Fixed tiles are glued to the glass plates with adhesive placed on the upper or lower sides of the tiles so that they have a fixed position.

Moving tiles are free to move, but only in the horizontal plane. In the vertical plane, each tile exactly occupies all of the vertical gap between the glass panels. Because the glass and the upper surfaces of the tiles are very low friction, the tiles can slide about between the glass panels very easily. Providing that there is no vertical movement, which the glass prevents, tiles are free to make any translational or rotational movements.

The tiles interact. For example, sometimes a moving tile will be moving in one direction and will strike another tile, causing its direction of movement to change. Sometimes a tile will be struck by another tile and set into motion. Sometimes a moving tile will be set rotating after interaction with another tile.

The states of the moving tiles – their positions, rotational states, velocities and rotational velocities – give the computational state of a FlatMech computer.

Energy may occasionally be lost due to interaction with the tiles, but this can be resupplied by special energy input tiles near the edges of the computer, that are joined by connecting rods to an external engine.

If inputs are required, special tiles serve as inputs. Each is a moving tile joined by a connecting rod to some input source that moves it in accordance with the required inputs. If outputs are required, special tiles serve as outputs. Each is a moving tile joined by a connecting rod to some system outside the computer that reads its movement. The argument in this article and the previous one does not have much need for inputs and outputs, as it is about minds running in virtual realities. However, I include them here for completeness and there is some discussion of interference with computers from outside.

All connecting rods have a diameter of less than 5cm (that is to say, they are thinner than tiles) and only move horizontally.

For moving tiles, the computer's functioning is not dependent on friction between the upper or lower surfaces of the tiles and the glass plates between which they are placed. The friction between the tiles and the glass should be as low as possible. Any energy lost due to whatever friction there is can be resupplied. This means that the computer is not dependent on the glass plates for anything other than support of the tiles. The computer would work with the upper glass panel removed and the lower glass panel could be removed, given something to support the moving tiles (or a weightless environment) and something to hold the fixed tiles in position.

The degree of friction on the vertical sides of the tiles can be varied. Tiles can have very low friction sides, so that there is little friction when they interact with each other, or they can have high friction.

Combining Two FlatMech Computers

The thought experiment in the last article was intended to show that if we assess probabilities of being in various situations by merely counting the possible computers that could be running us we are in trouble because, in some situations, deciding whether we have one or two computers is arbitrary.

I will now give an example of combining two FlatMech computers. How many computers do we have at each stage of the following process?

1. Two FlatMech computers, FlatMech 1 and FlatMech 2, are identical and are in the same state; that is to say, the types of tiles are the same, the positions and orientations of all the fixed tiles are the same and the positions, orientations and velocities of all the moving tiles are the same. Each moving tile in FlatMech 1 is moving in synchronization with the corresponding tile in FlatMech 2. FlatMech 1 and FlatMech 2 have their tiles mounted between glass panels. Any fixed tiles are glued to the glass panels with adhesive that can be removed if required.

2. FlatMech 2 is placed above FlatMech 1 with a vertical gap of 1 metre between the upper glass panel of FlatMech 1 and the lower glass panel of FlatMech 1. There is also a 1 metre difference in the *horizontal* positions of FlatMech 1 and FlatMech 2, so that the corresponding tiles do not vertically align.
3. FlatMech 2 is moved horizontally so that its tiles horizontally align with the corresponding tiles in FlatMech 1.
4. FlatMech 2 is moved down so that there is a 1mm vertical gap between the upper glass panel of FlatMech 1 and the lower glass panel of FlatMech 2.
5. FlatMech 2 is moved down further so that the lower glass panel of FlatMech 2 touches the upper glass panel of FlatMech 1.
6. The lower glass panel of FlatMech 2 is glued to the upper glass panel of FlatMech 1.
7. The lower glass panel of FlatMech2 and the upper glass panel of FlatMech are removed so that there is nothing to separate the FlatMech 2 tiles from the FlatMech 1 tiles. The FlatMech 2 tiles are lowered so that each FlatMech 2 tile now rests on the corresponding FlatMech 1 tile.
8. Each FlatMech 2 tile is fastened to the corresponding FlatMech 1 tile on which it is resting with adhesive tape. This tape serves no purpose, as the tiles are moving in synchronization anyway.
9. Each FlatMech 2 tile is welded to the corresponding FlatMech 1 tile.
10. Each welded together pair of FlatMech 1 and FlatMech 2 tiles is replaced by a single tile which is 10cm high (twice the height of an existing tile). The replacement is done one tile at a time and the tiles are replaced while the computers are running.

It is not obvious when the two computers become one computer. The previous article discussed the idea of the computers being possible candidates for your current situation – when each computer is running an identical version of you in a virtual reality simulation. In such a situation, if we assign probabilities by counting computers, viewing the total number of any single computer as one possible situation in which you could be, for the purpose of determining probabilities, is incoherent.

Possible Complications in Combination

In the previous article, when considering the combination of two computers, A and B, I did not give any consideration to the possibility of a single computational state based on both A and B. This might complicate combination a bit, but it is really beyond the scope of the simple consideration in the thought experiment being considered. In any case, it certainly would not cause measure and probability to decrease when computers are combined, so it is no threat to the implications made from the thought experiment. This sort of idea is more speculative than the current consideration and to deal with it would really need the more sophisticated ideas that will be discussed later in the series.

In the previous article [1,2] I stated the following:

“Would it be reasonable to say that the probabilities for computers A, B and C, or the combined probabilities for any group of such computers, must change appreciably during such a merging process? Is this supposed to happen magically in the instant when equivalent components touch each other?”

Given possible complications in complication I should probably not have gone as far with this statement. We cannot assume that probabilities will just add up properly when we combine computers, and we cannot really be sure that no shift in probabilities would occur, because the situation would be different. I should have really asked this:

“Would it be reasonable to say that the probabilities for computers A, B and C, or the combined probabilities for any group of such computers, must change appreciably during such a merging process, without any physical mechanism or known process to do this, and in exactly the right way to make the probabilities correct? Is this supposed to happen magically in the instant when equivalent components touch each other?”

Possible Objections

The following is a list of possible objections to the argument made in Part 1, with answers. Some of these objections were included as a result of informal discussion with other people and as a result of comments posted at <http://www.machineslikeus.com/cms/minds-substrate-measure-and-value-part-1-substrate-dependence.html> [2], though the objections here may not exactly match these.

Objection 1: It is all irrelevant. This is like arguing about how many angels there are on the head of a pin.

Answer

The argument may seem irrelevant, and it may seem tempting to write it off as trying to answer a pointless question, but the question is not pointless. Should there be uncertainty about your current status, and if a number of situations involving different substrates are possible, considerations like this have an effect on the very real probabilities of being in different situations. A well-defined position on the relationship between minds and matter is also needed to deal adequately with Searle’s arguments [3,5].

Objection 2: The situations considered in the previous article are contrived and have no relevance to “real-world” situations.

Answer

None of the situations considered break known laws of physics. It would therefore be unreasonable not to consider them as “real-world” situations unless a very specific definition of “real-world” is used which only relates to things in our everyday experience.

If appropriate technology to deal with computers and brains were available these situations would be practical ones. Any discussion of strong AI is considering hypothetical situations involving futuristic technology by definition, and is considering intelligence on substrates other than human brains, so it is inconsistent to suggest that philosophical arguments about it should not enter the realms of futuristic technology or extremely different situations from what we know.

It is not generally a good objection to an argument to suggest that it deals with situations that are too far detached from everyday life to be meaningful. Whatever view we have of the relationship between matter and minds, it should be valid in all situations. If we can produce situations where we get nonsense results, even if those situations seem contrived and practically implausible, those situations will not go away by hoping that they do not appear in the “real world”. The mere fact that they occur at all would indicate that we had a flawed view that would not work under normal conditions either.

Objection 3: We do not need to be able to compute probabilities. We can just declare them irrelevant. There may not even be defined probabilities in some of these situations.

Answer

We do not find it generally tenable to discard probability. A scientific theory may not allow us to easily obtain probabilities of events happening, but one that did not even in principle contain any way of coherently dealing with probabilities would be regarded as useless.

I also maintain that most people who declare probability irrelevant would not consistently subscribe to their own position in all situations. In some situations their expectation of various events happening or not happening would be very important to them. As an example, suppose that you know that your brain has been copied and that you are running on one of a number of computers. On one group of computers you face pleasant experiences and on the other you face horrible torture. The differences between these groups of computers, and which of these differences matter, would be important issues to almost anyone, even though this may not mean agreement with me. A person who declared probabilities to be irrelevant or undefined would find him/herself in an unusual situation: one in which he/she was unsure whether or not to be happy or terrified. This is absurd. What is such a person supposed to do? Is he/she supposed to be neutral about the whole thing? Deciding whether to be scared or not is quite basic.

Objection 4: Your FlatMech substrate is contrived because it allows two computers to be combined in a straightforward way without much involvement between computers. For example, if we use an electronic computer instead of FlatMech, laying two computers made of wires and other electrical components on top of each other so that the wiring is thicker, electrons that were originally in one computer are no longer confined to it and can move between computers. The results of the combination of some computers are therefore more complicated than for other computers.

Answer

You cannot defend strong substrate independence by complaining about my choice of substrate in arguments. Such an objection concedes that the substrate *does* matter and therefore accepts the very position that it is supposed to refute.

Some readers may say that the argument that probability or measure are higher when there is inefficient use of matter in computing is unsound because it is easier to see the consequences of combining two computers in a simple case like FlatMech. What if it does not work in this way for computers which combine in more complex ways? This would be a weak argument. Use of examples of combination of computers like FlatMech suggests that measure (and probability in situations when there is uncertainty about your status) is higher when computation is redundant and there is no reason to think that this is not a general feature of combining computers.

In any case, the idea of redundancy, or inefficient use of matter, as an indication of measure is merely a placeholder idea. A later article will provide a better way of determining measure and probability, based on counting the number of ways of algorithmically extracting meanings or patterns, and this will show that high measure *is* generally associated with high redundancy in a substrate.

Objection 5: What if we just do not know? Your argument does not show that the idea of counting computers to determine numbers of minds or probability is incoherent. At most it would show that there must be a way of deciding how many computers there are in a given situation and that *we just do not know what it is*. In some situations where we clearly have a single computer we would have a single mind, without any need to consider issues of substrate and measure. In situations where there seems to be some doubt about how many computers we have – as at some stage through the process of combining two computers – we would need to use some currently unknown method to resolve the situation. With this method things would be just as clear.

Answer

This objection is mainly based on denial and on the intuitive belief that a mind is caused by a computer running a program and that for each computer you have one algorithm and one mind and *that is how things are*. The main problem with this objection is that there is

nowhere for this “unknown method” to reside. It may *seem* reasonable that some unknown physical process is involved in the combining of two computers, or that some unknown knowledge about the physics of what goes on when two computers are combined awaits discovery, but advocates of this position do not think that there are any physical processes associated with it: they think that the mere execution of a computer program, processing symbols in the correct way, produces a mind and this lack of detail or physical process leaves nowhere to look for anything to go on during combination of computers to sort the mess out.

Objection 6: In your FlatMech example, you insist that two computers become one when placed on top of each other, but they will gradually drift out of step due to slight errors in timing. This is avoided when the tiles are stuck together, but then you have a specific point at which they become a single computer – when they are joined.

Answer

I am not saying they become one computer: I am saying that the question of how many computers there are is incoherent.

The problem with this objection is that by the time you are finished discussing it you will have got deeply into substrate specific matters – something you should not want to do if you are claiming strong substrate independence.

Even if two layered FlatMech computers would ultimately drift out of step, this could take a long time to become significant and it is strange to count computers now based on what will happen later.

If we put glue between two FlatMechs, or welded them together, and the machines were accurately calibrated, the stress on the joint due to calibration error may not be noticeable over short periods of time and it is strange to claim that this joint has huge philosophical importance.

Making a joint is not all or nothing. For example, we could “join” two FlatMechs together using electric magnets which pull equivalent components together. The strength of the magnetic field could be adjustable from nothing – the only thing keeping the FlatMechs in step is their calibration – to full strength – when they are held together. If the joint is philosophically important you now have the problem of saying at what magnetic field strength this joint becomes a *real* joint.

If we wanted to make sure that the computers remained in step we would not even have to link them so closely like this. Digital computers are self-correcting, as they can only have discrete states and all that would be needed is for the timing of the machines’ state transitions to be calibrated. Each machine could have an internal clock that was occasionally calibrated from an external source, such as a radio atomic clock signal (as is the case with some commercially available watches) or a neutron star (a radio telescope

being aimed at a neutron star and connected to the computer). It would be bizarre to claim that it really matters whether they are calibrated from the same time signal or not. For example, in the case of using a neutron star as the clock this would suggest that aiming one computer's radio telescope at a different neutron star would cause a split into two minds. Whether calibration is from the same external clock is not all or nothing. They could be calibrated from different external clocks that are poorly calibrated from the same clock, for example.

Objection 7: If we can combine computers we can also split them, creating computers with *fewer* minds. Starting with one computer, this splitting could be done indefinitely, resulting in computers with progressively fewer minds. This could only be done if the computer that we started with had an infinite number of minds. The argument about combination therefore suggests that all computers have an infinite number of minds, making any assignment of probability based on numbers of computers impossible and defeating the argument itself.

Answer

One response to this objection would be to say that there is a limit to splitting as eventually we will encounter fundamental particles of matter or the uncertainty principle will make further splitting meaningless. This would be a weak reply. We cannot be sure that there are indivisible particles, or that a particle view of matter even applies at such low levels, and, even if there are, it would be strange for a view of what minds are to rely on this sort of specific physics: it would imply that minds could not exist in a universe without indivisible particles. The same goes for the uncertainty principle. Furthermore, these kinds of reply could be countered by saying that the splitting is conceptual, not physical, and is therefore not limited by physics. I need a better reply than this.

I may seem to be proposing that a computer has a single mind and that when two computers are combined we have two minds, but I am not. I stated in the previous article that the idea of “redundancy” or “inefficient use of matter” as a way of determining measure or probability is merely a placeholder – a simplification of the true situation.

I may also seem to be proposing that the measure or probability actually comes from the combination of computers in some sense, as if this is all about combination of machines. I am not proposing this. The previous article has not stated where we get measure or probability, but merely shown that substrate must affect them and that, all else being equal, minds in high redundancy systems have high measure. I used the word “measure” a lot to avoid too much discussion of “numbers of minds”. The argument did not really explain what “measure” is, beyond implying that it relates somehow to “numbers of minds”, but this may be in a very abstract way. The argument makes it clear that the sort of probability in the thought experiment is supposed to be greater when we combine computers and we cannot make probability abstract: it has to represent our actual expectations. We seem to have a situation, then, where we may have issues when we try to talk about absolute numbers of minds, where we have to use some kind of abstract “measure” of minds instead, but where this abstract measure is supposed to imply real

probabilities. There is no problem with probability splitting indefinitely: a probability of 0.5 can become 0.25, then 0.125, and so on without any point at which this must stop, provided that probabilities are based on some abstract “measure” rather than absolute numbers of minds. Probabilities are not based on any absolute measure, but on relative measure and this is all that we need. For example, if there is a probability of 0.2 that your experiences are based on Computer A and a probability of 0.1 that they are based on Computer B then the “measure” of minds like yours in Computer A is twice what it is in Computer B. We might take this as meaning that some absolute number of minds is involved, but as long as we just need to make comparisons of measure we need never confront the issue of whether or not there are absolute numbers.

Combination of computers is not important in itself, beyond its usefulness in demonstrating that there must be at least statistical substrate dependence. An argument about infinity may seem strong, but it will not make the problem of combination go away. Given that the thought experiment in the previous article showed that “counting computers” as a way of assigning probability is incoherent, if you want to assign probability to different situations, which you have to be able to do to make sense of reality and *use* any philosophy, you need to deal with the implications of counting computers.

Infinity itself need not be a problem, though how we get “measure” is relevant. There may still be scepticism of the idea that infinity can be dealt with by basing probability on “measure” of minds rather than absolute numbers. Situations like this do not necessarily prevent extraction of useful information, however. An example is the derivation of the basic method of differential calculus. We define the gradient at a point on a curve as the y step divided by the x step over some small section of the curve and then we let the size of this section tend to zero – as if it were becoming infinitesimally small. The objection could be made that we cannot compare infinitesimally small values, but we never do compare them. The size of the section tends to, but does not reach, zero, and we can make a useful comparison of the y and x steps. This is not a comparison for a zero size section of curve, but nor is it a comparison for any particular absolute size: it is something more abstract. We do not need absolute values to make a comparison. It is the same with the issue of “infinite minds”. Even if we find ourselves having to infer infinite amounts of minds, it is *relative* numbers that we are interested in for the purposes of assigning probability and value. We can do this without stating absolute numbers of minds by using some method which considers some part of the infinite distribution of minds associated with one or more substrates and considers relative numbers of minds for assignment of probability and value as the extent of this distribution tends to infinity.

This may seem like an appeal to some vague process to save the argument, but all this will be discussed later in this series. I should, however, at least give an idea of how the distribution of minds will be handled:

The approach that will be later described will be based on the number of ways of algorithmically extracting a computational state corresponding to a mind from a particular substrate. One problem with strong AI is that what is physically going on needs

interpreting to produce symbols and computational states and there is no reason why any interpretation cannot be imagined, an issue which will be discussed later in this article. I will formalize these interpretations by saying that each interpretation of a physical system corresponds to an interpretative algorithm that produces a mathematical structure. Next, rather than being accused of arbitrarily selecting “obvious” interpretations (a charge which Searle makes against strong AI) we implicitly accept all of them. We therefore have an infinite set of interpretive algorithms which can be applied to reality to produce various computational states. Not all of these computational states would correspond to anything like minds, or anything like your mind. In the case of two computers, A and B, we would be interested in those algorithms which can be applied to those parts of reality that would be considered part of Computer A and those parts of reality that would be considered part of Computer B. We seem to have the “infinite minds” problem here. We can resolve this by saying that any interpretative algorithm has a length L and that probability and value are based on comparisons made between numbers of computational states extracted from the reality “in the vicinity” of Computers A and B for interpretative algorithms with maximum length L . Then we let L tend to infinity and we can still make a valid comparison. This is not much different from standard strong AI, differing only in that it attempts to formalize the stage where we extract symbols, meanings and a computational state from a physical system: strong AI typically ignores this.

Trying to return to some simplicity here, we can link this to what was said in *Minds, Substrate, Measure and Value, Part 1: Substrate Dependence* about “inefficient use of matter” leading to greater measure and probability. This is not a fundamental rule, and it may be hard to formalize, but it seems that this sort of thing should follow from the previous article’s thought experiment. The explanation for this is that the sorts of computers which have “high redundancy” or are making “inefficient use of matter” are those which to which more interpretative algorithms can be applied to produce computational states corresponding to minds – and saying “more” here is simplifying, given what was just said about infinity. Computers that use more matter than needed will tend to provide “more stuff” from which interpretative algorithms can extract computational states.

This will be dealt with in more detail later in this series. For now, I am merely suggesting that measure and probability tend to increase when combining computers and decrease when splitting them, the way in which this measure is computed needing deeper consideration.

Objection 8: If measure is only relative, as the answer to the last objection suggested might be the case, and probability is based on measure, then at most you can obtain relative probabilities for a number of situations, but you would still not know any absolute probabilities, making the whole argument useless.

Answer

It is true that the issue of infinities may mean that measure is only meaningful in a relative sense. As probability is based on measure, this may *seem* to indicate that

probability is only meaningful in a relative way. This would not be entirely true. If there are a number of situations in which you could be, and you find the measure for you being in each, relative to the total measure of all these situations, then you know the probability of you being in each, relative to the total probability of all these situations. This may seem not to be telling you much about anything beyond these situations. If, however, the possible situations being considered comprise *all* possible situations then the probabilities obtained from the relative measures of these situations are absolute probabilities.

For example:

Suppose your experiences could be due to one of two possible substrates, A and B, and:

Measure for you being on A = 0.8 x Total Measure for A and B

Measure for you being on B = 0.2 x Total Measure for A and B

then we can say that:

Probability of you being on A = 0.8 x Total Probability for A and B

Probability of you being on B = 0.2 x Total Probability for A and B

If you know that you could be in no situation other than being on substrate A or B then:

Total Probability for A and B = 1

and the probabilities now become absolute:

Probability of you being on A = 0.8

Probability of you being on B = 0.2

You might assign absolute probabilities if certain that the situations being considered comprise all possible situations, or as an approximation if you assign other situations negligible probability. It is not necessary for all situations to be well defined, or to involve just one substrate. For example, one situation might involve any substrate of a given general type, or may comprise any situation which is not part of the other situations, so that all possible situations are included by definition, but assigning measure and probability might need a lot of assumptions. Assigning absolute probabilities does not mean declaring complete knowledge about the nature of reality. You might accept the possibility that the possible situations being considered are “contained” in one of a number of other possible situations about which you know little, but nevertheless are an accurate description of the possibilities for practical purposes.

This issue of relative and absolute probability is nothing special about what we are doing here: it is an issue for probability calculations in general. It might be more apparent here because of the wide scope of what is being considered.

Objection 9: We only need to count unique algorithms. The previous article showed that simply counting physical computers as a way of determining probabilities is incoherent, but the correct position is that there is a mind associated with each *unique* algorithm that is physically implemented. Probability would be determined accordingly.

Answer

This would mean that if there are two computers running the same simulation of a mind in the same virtual reality, and the algorithms being executed by the machines are identical (the machines starting in the same states and going through the same state changes) then there would only be one mind. In a way, the existence of the mind would depend on that algorithm being implemented *somewhere*, rather than on the number of implementations of it.

Suppose you were unsure of your status and there were two different algorithms, A and B. Both A and B are algorithms simulating something like your mind in a virtual reality and you do not know enough to be sure whether Algorithm A or Algorithm B correctly represents your situation. We will assume that Algorithm A is running on Computer A and Algorithm B is running on Computer B and that no other algorithms that are possible candidates for your situation are being run on anything. According to this position, there would be a $\frac{1}{2}$ probability that your experiences are due to Algorithm A running on Computer A and a $\frac{1}{2}$ probability that your experiences are due to Algorithm B running on Computer B. Suppose now that Algorithm A is being run on 999 computers and Algorithm B is being run on one computer. My approach would say that we would need to consider substrates, though if they are all similar a probability of $\frac{999}{1000}$ that some implementation of Algorithm A is responsible for your experiences would be reasonable. The approach of counting computers on which you could be running would give a $\frac{999}{1000}$ probability that you are in one of the machines running Algorithm A. The approach that we are now discussing, however, would give a different result. It would say that the number of computers running Algorithm A does not matter and that there is a $\frac{1}{2}$ probability that your experiences are due to the implementation of Algorithm A on any or all of the 999 computers and a $\frac{1}{2}$ chance that your experiences are due to the implementation of Algorithm B.

With this sort of position “which computer you are in” may not even be a meaningful idea. If you know that a particular algorithm, running on multiple computers, is causing your experiences then some people holding this view may think that you are in any one of the computers running this algorithm while others may think that it makes no sense to talk about you being in any single computer and that you are in some sense in all of the computers running this algorithm. Other advocates of this position may think that it makes no sense even to talk about you being in physical computers, and that all that can be said is that you are associated with a particular algorithm – not a computer – which exists due to multiple computers running it.

I think that this kind of view is untenable, at least without qualification that would force it into weak substrate dependence, for the following reasons:

It leads to a single, small change in a system having an absurd power to determine expectation. As an example, suppose that there are $10^{1,000}$ computers running identical simulations of your mind in a virtual reality in which the sun is going to continue behaving normally tomorrow and one, slightly different simulation of your mind in a virtual reality in which the sun is going to explode tomorrow. In both types of simulation your current experiences are indistinguishable from each other. You know that you are in one of these simulations, but not which one. If you think that only the occurrence of an algorithm matters, irrespective of how many occurrences there are, then you should think that there is a $1/2$ chance that you are in a world in which the sun will explode tomorrow – the $1/2$ chance coming just from that single simulation in which it happens. That is a strange result! Furthermore, if we were to take one of the $10^{1,000}$ computers and flip a single binary digit (bit) in it, making a new simulation, different from the other two types, in which the sun explodes tomorrow, then your expectation that the sun will explode tomorrow should now be $2/3$. A bit in a computer may be flipped by altering the state of a single atom, yet such a small change can double your expectation of seeing the sun explode.

A more serious issue, if all that matters is whether a given algorithm is being run or not, is what is meant by the algorithm that is running you. This may seem a simple question, but is not, because of the problem of interpreting a system to determine its computational state. Suppose that some computer program, S_1 , that simulates you in a virtual reality, runs on three computers, A_1 , B_1 and C_1 , and some other program, S_2 , that also simulates you in a virtual reality, runs on another computer, D . You do not know if your situation is correctly represented by S_1 or S_2 – they both seem consistent with your current experience – but you do know that you are in one of S_1 or S_2 . What is the probability that your experiences are due to S_1 in computer A_1 , B_1 or C_1 ? If we are going to count unique algorithms there are two of them, S_1 and S_2 , so the probability should be $1/2$ for each. There is, therefore, a $1/2$ probability that algorithm S_1 correctly describes what you should experience in the future.

Imagine now, a different situation in which there are three computers A_2 , B_2 and C_2 . A_2 runs an algorithm, S_A , which simulates the physical workings of A_1 . As an example, if B_2 were a mechanical computer then S_A would simulate the movement of gear cogs and levers. S_A could also be more detailed and simulate the movement of individual atoms. B_2 runs an algorithm, S_B , which simulates the physical workings of B_1 . C_2 runs an algorithm, S_C , simulating the physical workings of C_1 . We also have computer D , running S_2 , still. If we are counting unique algorithms then the situation now appears to have changed. Each of A_2 , B_2 , C_2 and D is now running a different algorithm and any of these algorithms is a possible candidate for being the provider of your experiences. There is therefore a $1/4$ chance of you being in each algorithm. There is therefore a $1/4$ chance of you being in each of S_A , S_B or S_C and this means that there is a $3/4$ chance of you being in one of S_A , S_B or S_C . Each of these algorithms, however, is a simulation of the physical workings of a different computer that is running Algorithm S_1 , so if you are in one of these algorithms

then your future situation should be in agreement with S_1 . There is, therefore, a $\frac{3}{4}$ probability that your future experiences will be consistent with S_1 . This is different from the previous situation, with A_1 , B_1 , C_1 and D , where the probability was $\frac{1}{2}$.

Let us consider both of these situations together. In the first situation we had four computers A_1 , B_1 , C_1 and D and in the second situation we had simulations of the physical workings of these computers. These situations may seem to be different, but they are only different when we apply a particular computational interpretation to what is going on. *We can interpret a computer as being a simulation of itself.* The physical workings of A_1 could be viewed as being a simulation of A_1 , and therefore of S_A in A_2 . I would ask anyone doubting this to state what S_A is doing that A_1 is not doing. The objection could be made that S_1 in A_1 clearly does not simulate A_2 , but this is not what is being claimed. What is being claimed is that the physical system of A_1 does not just run S_1 , but also, by being itself, simulates itself, and runs S_A . This argument could be applied to B_2 and C_2 , so that we can say that A_1 runs S_A , B_1 runs S_B and C_1 runs S_C . D still runs S_2 . A_1 , B_1 and C_1 were running the same algorithm, but by interpreting them in this way they are now running different algorithms. We therefore have more unique algorithms and the probabilities alter to become like the situation we just discussed with A_2 , B_2 , C_2 and D . This would give a $\frac{3}{4}$ probability that your experiences are due to one of A_2 , B_2 or C_2 and, therefore, a $\frac{3}{4}$ probability that your experiences are consistent with S_1 . This probability was previously $\frac{1}{2}$, but to change it we did nothing beyond viewing the situation differently.

This is a paradox and it means that the approach of simply counting unique algorithms, without more detail about how to count them, cannot be right. One answer to this would be to say that I have a misunderstanding of the idea of counting algorithms and that it is the second approach – regarding the system as a simulation of itself – that is correct. A problem with this is that is accepting the very thing advocates of this position are likely to deny – that the substrate matters. If we adopt this approach then the thought experiment from the previous article now becomes relevant once more. If each computer is running a unique algorithm that simulates itself then we can combine two computers to make a single computer running both algorithms, affecting the probabilities. We would seem to have, therefore, a situation of statistical substrate dependence.

Once we start to think about combining computers, with this second approach, it should become apparent that a computer could be considered to be made up of a number of computers, each running the algorithm simulating its own physical workings, and that a computer could be considered to be running a different, unique algorithm for each of these. This would actually make a kind of sense, because a computer with more redundancy – one making less efficient use of matter – would be equivalent to more computers (though we have to be careful about what “more” means here, given considerations about infinity) and could be considered to be running more unique algorithms corresponding to the physical workings of these computers. This could be a justification for assigning high measure, probability and value to minds on substrates with more redundancy. This actually seems to be coming into agreement with the conclusions from the thought experiment and it is not too dissimilar from what I will be advocating in

this series of articles. I would point out though that it is not the simple “only count the unique algorithms” position. If we want a position like that to work we have to do so much to deal with the issues that we have just raised that it hardly has any resemblance to the simple, naïve position of counting the unique algorithms that you can “see” running on the computers. Whether we would regard these changes to the position as repairing it or just specifying it in greater detail is a semantic matter. I would also point out that, although I said that this “repaired” unique algorithm counting approach is not too dissimilar from the approach I will be using, it is not exactly the same. I will be going more deeply into the issue of *interpretation* of a physical system to get computational states and not bothering about uniqueness in the same way.

A particularly serious issue, for anyone who really thinks that we can count unique algorithms is that of arbitrariness of interpretation of computing systems. I have already touched on this in what I just said, although I have not really shown the full scale of the general problem. Searle raises this problem [4,10] and, while I disagree with his conclusions about strong AI, the point he makes about interpretation is valid. It relates particularly to this objection – to the method of assigning probability by counting unique algorithms – but it also relates to strong AI in general, so rather than just discuss it here I have given it a separate section, later in this article. This section will complete the answer to this objection.

Objection 10: The choice of splitting operation is subjective. Different splitting operations would give different measures. The results of a single splitting operation are therefore meaningless.

Answer

The sort of splitting in the thought experiment in *Minds, Substrate, Measure and Value, Part 1: Substrate Dependence* is not supposed to be a formal method for determining measure, but is part of an argument to show that some concept of measure is needed and that, *all else being equal*, high redundancy, or inefficient use of matter in computation, is suggestive of high measure. The objection, however, ignores this and tries to treat it as a formal method for determining measure. Even with an informal method such as this, however, multiple types of splitting can still be coherently discussed as follows:

Suppose we have a computer running some entity, E. We define a splitting operation, S_1 , which splits the computer into different computers running E. We define an index, I_1 such that each computer obtained by the splitting operation S_1 is at some point on this index. For now, let us assume that we are dealing with finite numbers of computers resulting from the splitting operation.

The measure of E after the splitting operation is the number of computers on index I_1 .

Now, we define a new splitting operation, S_2 , which splits a computer running E into other computers running E, and an index, I_2 , such that each computer resulting from

splitting operation S_2 is at some point on index I_2 . We will presume that we are dealing with finite numbers of computers for this index too.

Suppose we apply splitting operation S_1 to some computer running E and obtain a distribution of computers along index I_1 . Now, we want to apply S_2 , but we have already done one split. We deal with this by applying S_2 to each of the computers on index I_1 , generating an index of computers I_2 with a number of computers on it in each case. We now have index I_1 containing many versions of index I_2 arranged along it, each containing a number of computers. We obtain the total measure of E by counting the computers on each index I_2 along I_1 and adding them all up.

We could do this the other way round. If we did split S_2 first we could then use S_1 to split each computer produced by S_2 .

If splitting generates an infinite set of computers this does not substantially change the situation. Indices I_1 and I_2 would be infinitely long. We could now no longer obtain an absolute measure value. We could still, however, determine the *relative* measure of E in two computers C_1 and C_2 as follows:

Let L_1 be the maximum length of index S_1 that will be considered.

Let L_2 be the maximum length of index S_2 that will be considered.

(To be technically correct we should consider how computers are ordered along these indices, as it determines which are selected by a given value of L_1 or L_2 , but we can ignore that here as the main consideration is multiple indices rather than infinity.)

Apply S_1 to C_1 , generating computers distributed along I_1 , only considering index I_1 up to length L_1 . Apply S_2 to each computer on I_1 , generating a different distribution of computers along I_2 in each case, only considering index I_2 up to length L_2 . Count the total number of computers along each index I_2 along I_1 and add the totals up to obtain a measure, M_1 , for E in C_1 . This value should not be regarded as meaning anything as an absolute measure.

Apply a similar process to C_2 , obtaining a measure, M_2 , for E in C_2 .

Total measure of $E = M_1 + M_2$

Measure of E in C_1 as a proportion of total measure = $M_1 / (M_1 + M_2)$

Repeat all of the above process, increasing L_1 and L_2 each time. Use the value for measure of E in C_1 as a proportion of total measure which is converged on as L_1 and L_2 tend to, but do not reach, infinity.

In this way, relative measure for two computers C_1 and C_2 can be obtained for two methods of splitting. By extension, relative measure for two computers C_1 and C_2 could be obtained for any number of methods of splitting.

Further, a single index, I_3 , could be defined containing all the computers that occurred on each version of index I_2 along I_1 . A single splitting operation, S_3 , could be defined that combined S_1 and S_2 to produce I_3 .

The relative measures obtained from a single splitting operation cannot be considered a final result because there is always the possibility that a subsequent splitting operation could change the relative measure, nevertheless it can be considered to be an indication of likely relative measure after subsequent splitting operations, *all else being equal*.

It could be argued that different results might come from applying splitting operation S_2 first, followed by S_1 . Thought experiments involving splitting like this, however, are not formal methods for determining measure, but merely demonstrate statistical substrate dependence, the need for some concept of measure of minds and the general association between measure and redundancy. We should not expect processes like this to give accurate measures or probabilities, and so should not be concerned if we got different results from applying different processes – such as when we do splitting in a different order. The main point is that even with an informal idea of splitting and measure we can coherently discuss application of multiple splitting operations to a computer. This might be answered by pointing out that S_1 might produce computers so different in structure from the original computer that was split that, although S_2 can be applied to the original computer, S_2 cannot be applied to the computers produced by S_1 . At this stage the objection is pushing the simple idea of splitting beyond its limits and the response to this has to be as follows:

The simple splitting concept as described in Part 1 is not a formal description of measure but merely part of a thought experiment to demonstrate the need to consider measure and its relation to general “redundancy” in a substrate. A formal idea of how measure results would deal with issues like this.

The issues of possibly getting different results depending on the order of application of different splitting operations and of splitting operations preventing subsequent splitting arise because of a simplification in these sorts of considerations of splitting – that different ways of splitting a computer cannot use any of the same matter. With this assumption removed any “splitting operation” can simply become an operation to “extract” different computers and is unaffected by previous splitting. The objection suggests that there is arbitrariness in the selection of the splitting operation to be used, but if we are not limited by different splitting operations having to use separate matter then we need not select a specific splitting operation: *all* splitting operations can be applied. A more formal splitting idea that would deal with these issues would therefore have these features:

- Reuse of matter by different splitting operations: a splitting operation can be applied irrespective of what other splitting operations are applied to the same matter.

- All conceivable splitting operations are considered as contributing to the set of computers resulting from the splitting.

The proper, formal idea of measure that will be discussed later in this series works in this way. Instead of splitting a computer into separate computers, algorithms are used to extract *computational states*, a more abstract, sophisticated idea than extracting computers. Each algorithm extracts a single computational state. Overlap between different extraction algorithms is ignored and any algorithm which extracts a computational state can be applied without regard for other algorithms. There is no need to try to devise different types of splitting operation as measure is based on the entire set of logically conceivable extraction algorithms (equivalent to “splitting operations”) and any argument about a particular method of splitting being arbitrary, and the choice of splitting method being subjective, becomes irrelevant.

Objection 11: You assume that all minds are based on computation on a physical substrate. You ignore virtual machines. For example, a simulation of a brain may be running in a high level language which effectively provides a virtual computer, so the substrate in this case would be the high level language. The mind may be a simulation of a brain running on a computer which is being simulated by another computer, which is in turn simulated by another computer, and so on. The “substrate” does not have to be physical and there are many ways in which a mind could exist like this that cannot be counted.

Answer

We could say that any mind is running on a virtual machine, just by introducing appropriate levels of abstraction into the description of what the physical computer hardware is doing. For example, if a computer is running a simulation of a brain, we could say that the brain simulation provides a virtual computer. If an AI system is running in machine language we might say that individual machine code commands provide a substrate or that subroutines in the program provide the substrate. None of this is anything more than making abstractions of what a physical system is doing and there is nothing special about such cases. In all them, a physical system is still doing something from which a computational state can be extracted – even if the process of extracting it involves many levels of abstraction.

As an example, let us imagine a simulation of a brain running on Computer B which is being simulated by Computer A. An algorithm must exist that can extract the computational state of Computer B from Computer A, or it would be meaningless to say that Computer B is running on Computer A. Likewise, an algorithm must exist that can extract the computational state of the simulated brain from the virtual Computer B. A single algorithm could therefore be made which used both of these algorithms to directly extract the computational state of the simulated brain from Computer A. The internal workings of such an algorithm might be best understood in terms of the multiple levels of abstraction that we have been discussing, but that would just be our preference. It would be an algorithm which works as a formal description of a method for extracting the

computational state of a mind (very similar to what we simplistically call a splitting method) and algorithms can be counted, so all extraction methods, whether they use virtual machines or otherwise, can be counted.

The Problem of Interpretation

Searle's best-known argument against strong AI is probably his *Chinese room* argument, but he makes another argument about the problem of arbitrariness in interpretation. This is what Searle says:

"...computation is not an intrinsic process in nature like digestion or photosynthesis, but exists only relative to some agent who gives a computational interpretation to the physics. The upshot is that computation is not intrinsic to nature but is relative to the observer or user." [4]

What this argument says is that symbols and meaning do not exist in a physical system by themselves, but need an observer to allocate meaning to the physics. For example, 1s and 0s do not exist by themselves inside an electronic computer. Instead, we decide that particular voltages correspond to 1 and other voltages correspond to 0, allowing us to say that an electronic computer has a particular computational state. We interpret the electronic computer's physical state as having a certain meaning.

The problem with this is that the obvious interpretation is not the only one. It would be possible to interpret any physical system in any number of ways to get any computation we want from it. Searle calls this *multiple realizability* or *universal realizability* and says this:

"The same principle that implies realizability would seem to imply universal realizability. If computation is defined in terms of the assignment of syntax, then everything would be a digital computer, because any object whatever could have syntactical ascriptions made to it. You could describe anything in terms of 0's and 1's." [10]

We must interpret what a computer is physically doing to get 1s and 0s and an algorithm, but Searle is saying that you can get any algorithm out of anything if you make an appropriate interpretation, a point he makes clear in the following statement:

"For any program and any sufficiently complex object, there is some description of the object under which it is implementing the program." [10]

Searle gives an example:

"...the wall behind my back is right now implementing the Wordstar program, because there is some pattern of molecule movement that is isomorphic with the formal structure of Wordstar. But if the wall is implementing Wordstar, then if it is a big enough wall it is implementing any program, including any program implemented in the brain." [10]

(Note: *Wordstar* is a word processing program for personal computers.)

You would have to do some of these interpretations in a contrived way and some people would say that the computation really comes from the way you interpret the matter, rather than the matter itself, but – and this is an important point – there is nothing in strong AI which tells us which interpretations of a physical system are reasonable and which are contrived.

I disagree with the conclusion that Searle reaches – that strong AI needs to be thrown out. It will be obvious now that I think that strong AI needs clarifying. I do, however, accept the basic point made by Searle. This subjectivity of interpretation is a problem.

We can get an idea of this problem of interpretation by using part of Greg Egan’s novel, *Permutation City* [8] (in which arbitrariness of interpretation was actually a major plot device). In *Permutation City*, two characters, who are mind uploaded “copies”, hide in a computer system by having themselves placed in a complex, encrypted way, among the normal processing of that system. An observer may see that system, for example, running a virtual reality simulation of a fountain, but a bit of the fountain simulation is being stolen to do a bit of the job of running these two stowaways and an environment for them. This is similar to steganography, the practice of hiding messages in other data, such as in computer image files.

Suppose we wanted to check for intelligent agents hiding, encrypted in a system like this. We would need to run some sort of analysis of the computer – some sort of decryption – to show that they were there. If the decryption is not complex enough it may not find them, but if the decryption is too complex then it may find agents just by arbitrarily interpreting the computer hardware in the right way.

As another example, I will give this encrypted message:

AY5NSL Q23 FQK Q33WDN 7UIKDLQPD IKELDP HJKI IEKSL POLSO

I can promise you that there is a way of decrypting that to get “Hello world.” Alternatively, you can decrypt it to get “The quick brown fox.” Of course, decrypting it in this way may be contrived, and whatever decryption process we use may need to be supplied with more information than we are actually getting out of the message, but any decryption process needs to be supplied with some information. Ultimately, there is no way of saying that one particular interpretation is valid while others are invalid. You might claim that one method of decrypting the message recovers what was encrypted in the first place, but surely the history of the message does not matter: only what we have now should be an issue. You might say that the simplest interpretation that produces a mind is the correct one, but it should be easy to imagine situations with multiple interpretations.

This relates particularly to Objection 9, above. There is no real escape from the problem of arbitrariness of interpretation, no matter how we approach probability. When we base

probability on counting computers, arbitrariness of interpretation causes a serious problem in deciding which computers are running algorithms that count, but we might try to ignore this by using some “common sense” approach. Such a common sense approach would not be very effective, though: the thought experiment about probability in the previous article is really a special case of the problem of arbitrariness of interpretation. I think we are more obviously confronted by the general problem when we get probabilities by counting unique algorithms rather than computers, because we must be able to say how many unique algorithms are in a particular system and arbitrariness of interpretation has the potential for generating unlimited numbers of unique algorithms to mess up our counting.

This problem of interpretation is important in this series of articles and will be discussed in more detail in the next article, so I will end discussion of it here, for now. For the time being, I would just point out that if you maintain that my earlier argument about the incoherency of counting computers was irrelevant, because we need to count unique algorithms rather than computers, your position is also incoherent unless you can find a way of determining which algorithms are running in a system *without any observer subjectivity*. This is a point which will be supported in some of the later discussion when this is discussed more deeply.

Conclusion

The previous article, *Minds, Substrate, Measure and Value, Part 1: Substrate Dependence [1,2]*, gave a thought experiment involving uncertainty, showing that measure and some types of probability must vary depending on the type of substrate on which a mind exists. If your experiences are due to software running on one of a number of computing systems, but you do not know which, then if you assume that each computer has an equal probability of causing your experiences, there are problems when you start to combine two of the computers, because there is no point at which you can say that two computers become one. Considering each computer as equally likely, irrespective of how it is made, is therefore incoherent. This can be put another way:

Computers do not have hard edges.

If there is uncertainty about your status – about the substrate on which you exist – then the nature of a given substrate must affect the probability of it being responsible for your experiences.

This article has supplemented *Minds, Substrate, Measure and Value, Part 1: Substrate Dependence [1,2]* with extra support for its argument. An example computer system, *FlatMech*, has been given, with a description of how two FlatMech computers could be gradually combined. This gives an easily imagined example of the situation considered in the previous article’s thought experiment.

Some possible objections to the argument in the previous article have been answered. One of these objections was that Part 1 attempted to refute the idea of counting computers

to determine probability while ignoring the alternative approach of counting unique algorithms to which some people subscribe. One of the answers given to this objection was based on the problem of *arbitrariness of interpretation*. This is not just relevant to this objection, but to this series generally, and it will be discussed in more detail in the next article.

A view of where the measure of a mind comes from, and any formal idea of what determines it, has not yet been given. The previous article showed that we have good reason for thinking that high measure is associated with high redundancy or inefficient use of matter in a computation, but this concept is somewhat vague and a deeper idea of measure is needed. There is an *apparent* paradox between the idea that minds have measure and our experience of encountering things with minds in the real world.

I stated earlier that this article would give a better defined idea of measure which will resolve this *apparent* paradox. Instead, I used the article to provide extra support for the argument in *Minds, Substrate, Measure and Value, Part 1: Substrate Dependence* [1,2]. The next article in this series, Part 3, will discuss the problem of arbitrariness of interpretation in more detail. A better defined idea of measure will be described later.

References

[1] Web Reference: Almond, P. (2007). *Minds, Substrate, Measure and Value, Part 1: Substrate Dependence*. Retrieved 12 September 2007 from <http://www.paul-almond.com/Substrate1.pdf>. (Also at <http://www.paul-almond.com/Substrate1.htm>).

[2] Web Reference: Almond, P. (2007). *Minds, Substrate, Measure and Value, Part 1: Substrate Dependence*. Retrieved 13 September 2007 from <http://www.machineslikeus.com/cms/minds-substrate-measure-and-value-part-1-substrate-dependence.html>. (A copy of the article in Reference [1]. Includes reader criticism of the article).

[3] Searle, J. R. (1997). *The Mystery of Consciousness*. London: Granta Books. 1998. (Originally Published: 1997. New York: The New York Review of Books. Also published by Granta Books in 1997.)

[4] Ibid. Chapter 1, pp14-17.

[5] Searle, J. R. (1980). Minds, brains and computers. *The Behavioral and Brain Sciences* 3:417-457.

[6] Web Reference: Strout, J. *Mind Uploading Home Page*. (2002). Retrieved 22 June 2003 from <http://www.ibiblio.org/jstrout/uploading/MUHomePage.html>.

[7] Web Reference: *Mind Uploading Research Group*. (2002). Retrieved 22 June 2003 from <http://minduploading.org/>.

[8] Egan, G. (1994). *Permutation City*. London: Millennium. (Fiction).

[9] Ballantyne, T. (2004). *Recursion*. London: Tor UK. (Fiction).

[10] Searle, J. R. (2002). *The Rediscovery of the Mind*. Cambridge, Massachusetts: The MIT Press. 9th Edition. Chapter 9, pp207-212.
(Originally Published: 1992. Cambridge, Massachusetts: The MIT Press.)