

ON SELF-REPLICATING SETS

JOHN Q GEOMETER

1. ON SELF-REPLICATING SETS

1.0.1. *John. Q Geometer.*

1.0.2. *2020.*

1.0.3. *link to edit page.*

1.0.4. *paper. pdf.*

1.1. 1. Computer Science and the Theory of Self-Replication. Throughout the history of modern computing machines, people have contemplated the idea self-replicating machines. At the dawn of the information age, John Von Neumann in particular devoted thought to the subject, creating a blueprint that people continue to use both for understanding hypothetical self replicating machines and for understanding the architecture of existing computing machines. At the same time, Alan Turing developed a similar toy model for how generalized computing machines work, which is taught in basic computer science classes to this day. We will not delve very deeply into these models, but will instead present a very crude sketch of them in order to discuss the assumptions made by computer scientists in the models they build to understand their world.

The Turing model of a computer consists of an infinite tape of 1's and 0's along with a machine to both read and change the state of the numbers in the tape. The string of numbers describes instructions for the machine, which follows those instructions by moving the tape back and forth and changing numbers from one state to another. Turing was able to show that this toy model of an abstract computing machine could be proven mathematically to be equivalent to any other abstract toy model of any other kind of computer, including the complex machines built today(hence the continued use of this model in teaching and scholarship). This is considered to be one of the most important results in theoretical computer science.

Before discussing this model's limitations we must say a word about the nature of scientific models. When we investigate a thing using the scientific method we have in principle the entirety of science knowledge to call on, built up from a vast number of models in different fields and sub-fields. For example, if we are presented with a rock to analyze, a physicist might ignore everything but the crystal structure of some prominent material in the rock and bring the modern understanding of crystals to bear on it. The micro-biologist will only be interested in the aspects of the rock that interact with the organisms on the surface, while the ecologist will be interested in that but primarily how the rock regulates the flow of water through the ecosystem of which it is a part. As scientists we may agree that models of surface chemistry for microbes, models of how atoms arrange in crystals, and how water flows through rocky soil are all "good science", but in any given context the model we choose to focus on depends on that context.

It is our assertion that while the model of Turing and his contemporaries is not wrong, that it is deeply misleading because in most cases it does not describe the most relevant aspects of the machines we call “computers”. Computers do, of course, compute. That computation is described by the mathematics of computation. They also create heat, described by thermodynamics, and are we therefore to call them heaters? They keep time with extremely fast clocks, do we simply call them clocks? No. But what actually are they? What is the model for a modern computer which is most useful when trying to understand them? In this era, in the year 2020, the most useful model for a computer is the one that describes how they have totally changed all aspects of global society in a relatively short time. For this we must expand the models available, and in particular we must focus on a specific shortcoming not just of mid-20th century computer scientists, but of most scientists in the “modern” era, namely our refusal to put ourselves and the societies of which we are a part into the systems which we study. In some abstract sense, one can argue that we put ourselves into the models with the role of the observer in quantum mechanics for instance. But we do not put the role of something like university politics into the models, even though these forces clearly influence the science we use and hence the conclusions we draw. It is the assertion of this paper that this blindness has become so critical in the understanding of computers as to be wrong in a way that has real world consequences.

So what is the problem with a Turing machine? If we look at it naively as a physical thing, not as a mathematical toy, we see a number of things that are not realistic, such as the infinite tape and the lack of any meaningful human readable input or output. But these are typical of useful scientific models: while the tape is understood to never be really infinite, the results you can get from the machine with “such a large memory that it doesn’t matter” and infinite mostly don’t matter, so our toy model still works. But the critical flaw of the machine which gives us an incorrect understanding of how computers function in society is that it has no origin, no purpose, and no destiny. Who built this machine? Why? How long can it run before it breaks? What happens when it breaks? When it ceases to function, what replaces it and why?

We now live in a world where a large fraction of the computing machines that exist live on a trajectory from a mine to a landfill of less than 2 years. During this brief journey from mine to landfill, they are mostly used for communication, and much of that communication is marketing information the primary purpose of which is to lead to further consumption of similar machines. That is, *in their current state* the *primary* action of these machines is rapid replication.

This is the reason we must shift the *primary* scientific model of computing machines from the Turing model to a more biological understanding. If we seek to study a new species of life that is taking over an ecosystem rapidly, we will always try to focus on the models that allow us to understand that phenomenon because it is what affects outcome. If a farmer asked us to evaluate a new crop blight and we came back with a deep study of how carbon chemistry works because all life contains carbon they would be very disappointed in the results even if they are technically correct. Likewise if we are to deal with the explosive changes to our physical world caused by computing machines we must focus on the means of replication. We will give an analysis of these means in the next section. However before doing that we must turn to the history of self-replication as an idea in mainstream computer science.

There are two main intellectual threads in this story: cellular automata and self-replicating robots. Interestingly, it is the former of these that actually have a vast experimental component and the latter which is entirely theoretical. Cellular automata are in some sense a generalization of the Turing model: they are sets of rules for multidimensional (usually 2 dimensional) grids of numbers, generally 1's and 0's, which follow some set of rules. These systems are simulated on real computers, and as time advances in a program we can see fascinating patterns of what appear to be naturally oscillating structures moving around in a visual display of dots on a two dimensional canvas. If the rules are created correctly, these structures can be made to replicate themselves. The literature on cellular automata is vast and complex and continues to be a very active field. Nonetheless, it suffers the same flaw as the Turing model: it exists in a vacuum, with an assumed infinite time, and no reason to exist (at least this reason is not part of the theoretical framework used to describe them), no origin and no ultimate destiny when it breaks. There is beautiful pure math to be found in these systems, but no illumination of how computers function as machines that copy themselves.

The second thread of self-replicating machines is the study of theoretical “robots that build robots”. This has attracted some truly wild speculation. What is generally imagined is a totally automated system without human intervention in which a computer controls robots that mine minerals which are used to build both more computers and more robots. This is then imagined to be so self-contained that it can be used to expand out into the universe outside of Earth, eventually consuming all things into this vast, automated, technical ecosystem which does not need any living thing to grow. Technologists have imagined these systems, then promised that they can be a fantastical utopia of free things, but also warned that they can destroy the world by consuming everything in site. For decades, theorists have written very detailed descriptions of such systems, delving into metallurgy, synthetic chemistry and the like to try to prove that such a thing can be built. Most recently, the work of K. Eric Drexler and Ralph Merkle in the 1990s pointed to a system like this built from precise positioning of atoms in matter—essentially treating physical matter as just another Turing machine. These theorists constructed very detailed imaginary worlds where atomically precise machines manipulate atoms to both compute and create, replicating themselves on a global scale. Perhaps their day will come at some future time and such machines will be built, but right now these models are of no help in understanding technology as we find it today. The rest of this paper is devoted to developing both a theory of self replication in modern technological society and in actually *using* this theory, just as the pioneers of modern computer science used Turing’s model early on, to build new things based on the new model.

1.2. 2. Self-Replication in Human society. First, a word on self-replication in biological systems that exist outside of human society. “Natural” biological organisms never replicate themselves in a vacuum. On Earth as we find it today, all living systems replicate as part of larger ecosystems, and the parameters of those ecosystems are *fundamental* to the overall replication. No animal can live long enough to reproduce without a constant flow of oxygen from plants needed for respiration. Conversely plants need the carbon dioxide we produce in order to live long enough to replicate. No one would dispute that a tree is a self-replicating thing, and yet trees only replicate in collaboration with a large number of other organisms, generally other plants, fungi, animals, all working in concert to make the overall forest

system replicate itself, of which the trees are only a part. What we find in spite of this, however, is that scientists outside of biology put much more restrictive rules on self-replication, saying a thing does not “really” replicate itself if it replicates as part of a larger system. Hence the flaw in computer models: if humans replicate machines, those machines are not called “self-replicating” because for the overly restrictive definition of the “self” of the machine they do not spontaneously replicate. But this type of isolated spontaneous replication does not exist in nature even for purely biological systems, so applying it to systems outside of biology will give results that are at odds with how the biological world works. Again, we must distinguish between models that are “right” and models that describe the primary characteristic of a system under study.

In the pre-industrial societies, *all* technology is self-replicating. Historically, people make technology using the materials found in their environment, generally from other organisms like trees(wood) and animals(bone) or found objects of which there is a plentiful supply. People then reproduce and teach the system of constructing such technology to the next generation, along with enough understanding of that technology that the young can in turn pass along the information when they age and are passing it along to yet another generation. This type of self-replicating technological system can exist in stable dynamic equilibrium with existing ecosystems. Trees can grow, be converted to boats to hunt in, which lead to cooking technology to cook the animals killed in the hunt, and trees and game animals can then re-grow as future generations replicate their technology.

If we take this broader view of self-replication we don’t even need to restrict ourselves to humans to see self-replicating technology. The beaver dam self-replicates quite easily and indeed before the rise of modern society one can imagine the dam itself as a self-replicating entity which uses the beaver as a vector, but which transforms the landscape vastly in excess of what one might imagine possible for a single small animal. To make sense of a landscape transformed by beavers it does not make sense to either study just beavers or just dams, we much consider the self-replicating set of dam-and-beaver as a single system. The same is true with human technology.

Moving into the very early reaches of our history as we learn it today based on written records, the next self-replicating systems are religions and empires. A religious text is perhaps one of the best prototypes for self-replicating technology which can shed light on the current state of affairs. Religious texts such as the Torah or the Koran describe both a world view which gives people a “why” to what they do which includes the replication of that text. They also include the description of “how” to replicate the text, building up complex structures of education which teach the next generation of humans what they need to know to keep replicating the next down through the generations. The other main replicating structure is that of the military bureaucratic empire. An empire replicates by expanding to incorporate more and more people into the actions of further replication. This is generally done by force, which can keep growing by taking more land for more mining resources and also more people to continue to gain power to continue to expand, consuming other systems and turning them all into that one central imperial system.

The entirety of human written history can be looked at through the lens of these self-replicating systems, where the means of replication is the primary descriptor of the systems. The history of what used to be called “Christendom” can be seen entirely through this light. The Torah was replicated by Jews for thousands of years, and

was limited in its replication by Jews' only replicating it to other Jews, so the growth was limited by the biological reproduction of the people who did the replication. Then, when Christianity appeared, the same text was suddenly being replicated by the people of one of the vastest military empires ever built, Rome. Ultimately this replication consumed Rome and became the Holy Roman Empire which among other things was a vast replication machine for the scripture. Then, due to technological advancements, it became possible to replicate that scripture mechanically in bulk with the printing press, and we see another explosion of change in that world where the press and how it replicated religious scripture caused some very radical change. As Western capitalism developed, the King James version of the Bible, printed in bulk on mechanical presses defined the beliefs of the military empires that then went on to conquer the globe(singling out the British Empire, followed by the American as the most powerful of the lot). Nothing in our world today makes any sense without this story of evolving self-replication.

So now this brings us at last to the discussion at hand: self-replication in regards to modern computer technology. How do computers replicate? Just as an analysis of beaver dam replication requires understanding the trees from which sticks are harvested and the rivers which feed the beaver ponds, this analysis must include externalities that are ignored by theorists, such as investors and marketing. Modern computers exist as creations of a combination of mass market consumerism, venture capital investment, and government research and development mostly focused on the military. Every company that makes computer hardware and software today is the creation of a very specific process whereby an entrepreneur pitches a company to investors, who in turn pitch their fund to larger institutional funds like pensions. After they get funding, they grow using a very specific type of worker, the modern tech worker fully indoctrinated in a certain culture. That growth is made possible by a system of mass media that transmits the information to consume the products to the masses outside of "tech". Those masses of people both put money into the products of this creation process and also put investment capital into the financial system that funds the venture capital that creates all this. Nominally all this is enabled by the "money" system which used to be based entirely on mining of minerals, but is now based on some complex system of faith more loosely based on mining. The computer systems which out-evolve their competitors are the ones that replicate the fastest. They are the ones that convince people to consume more and faster, and put more money and time into the system. The venture capitalist David Horowitz has explicitly said that in the future they are building everyone will either be forced to obey the media on these systems or will become one of the people building them. And indeed this is the logic of the self-replicating machine. People like David Horowitz have to exist in order for the machines to out-replicate other machines.

This picture presented here is of course a vast over-simplification and the product of a fairly brief and superficial analysis. It is not the purpose of this paper to create a full model of replication of modern "tech", but rather to convince the reader that *such a model is needed*, in the hopes that people will develop more accurate models that we can use to try to gain some control over this system and ultimately over our lives.

In summary, the simplest model for computers that I think we should consider now is not that of the Turing machine but of the advertising machine which exists for the sole purpose of convincing people to consume more advertising machines.

This might take the form of presenting PowerPoint to investors, using computers to train the workforce to build the technology, or spreading an article in the tech press about some new technology, but in most cases it is just direct advertising to the consumer. But in all cases the primary characteristic which determines form is replication. This is why evolution of machines has favored more and more of the machine being screen, with the highest possible pixel density and color contrast, rather than maximal computation power. Pixels are what sell pixels.

We must also distinguish between viral replication and independent replication, although the line is blurry. Viral replication assumes a fixed system in which the information replicates. For instance, information can replicate itself within a commercial social media platform like Facebook or Twitter just as influenza can replicate in a host human, but this does not replicate the *system*. The means of replication of a system such as Facebook is in fact not replication of content, but the whole system including venture capital, technical labor, media to sell it, the legal framework to enforce the power of the company, etc. It is this full system replication that we are concerned with here, not the replication of information within such systems, known colloquially as “memes”.

1.3. 3. The *Potential* Power of the Open Web for Self-Replication. What is the Web? The Web is not the Internet. The Internet is a network that connects almost all computers in the world, both physically and with some software protocols. This network traces its origins to the network of military and academic(but military-funded) computers that emerged from ARPA back before the modern commercial Internet, going back to the end of the 1960s. The Internet can in principle be used to exchange any information and treats information in the same abstract sense as in the models of theoretical computer science discussed above.

The World Wide Web was initially the creation of one person: Tim Berners Lee. It was initially created as a directory for the large science institution Lee worked for(CERN). The Web works beautifully on the Internet and the Internet is what made it huge, but it is not the same thing. The Web is a system for encoding information for and by humans to communicate with other humans. It includes human readable code designed to create universal documents which link to each other and can contain images of all kinds and text in all languages, creating a sort of universal document in a universal language in which all of humanity can communicate. While the modern Web is commercialized and increasingly not open to all users by default, this is a choice we have made that can be un-made. In principle any computer *can* be both a web server and a web browser. If we call the “open web” the collection of all web files which are openly viewable to anyone connected to the Network, the open web can in theory physically grow to include every computer in existence using the existing physical telecommunications network.

Let us now estimate the size of this potential Open Web network. We suppose that given the multiple billions of smart phones, laptops, servers in server farms, embedded systems, supercomputers, etc., that the total number of potential web servers is of order 100 per person. We then round human world population up to 10 billion, and estimate that there are 1 trillion total potential servers on the Open Web. Given that even a modest cell phone has a few gigabytes, but many servers and deep storage computers have terabytes, we can round up a little and say the average server can host 10 gigabytes of data. If a file like this one(this paper, the one you are reading rightnow) is 100 kilobytes to 1 megabyte, let’s round to 1 meg and say

there are 10,000 documents like this one(they could in theory all be math papers) for each server. So the total number of documents per person is 1,000,000. But we as a society *share* these documents, so in some sense what we all have is not 1 million but 1 million times 10 billion or 10 quadrillion documents(10,000,000,000,000,000). The informational universe in which we construct this new mathematics consists of this network of 10 quadrillion linked documents.

This universe of information exists on web servers which can in general be made to run code that edits and replicates the documents. Thus *every* document in this universe of information can self-replicate and be edited in situ. If this is all on the Open Web with code that can be edited by anyone on the Web, all users can constantly edit all documents, so potentially we have 10 billion people all simultaneously editing 10 quadrillion files all of which are able to instantly self-replicate from any node on the Network to any other Node. This vast network effect can create power in the same what that billions of brain cells with massive interconnection, creating a document of greater power than any that has ever been possible before. The power of such an open system will be so vast that it will make no sense to have any private data. Without any property on the Open Web, things can replicate freely, and the increased value will be so great that it will consume private property online. This evolution will be physical as the value to the physical caretaker of a physical web server becomes greater to participate in the Open Web than to keep it in the commercial web. Note that if we try to simply write down the number of ways that these documents can point to each other to self-replicate, since each document can replicate from *any* other document in the collection of documents(and any number can replicate from any one other document), the number of ways they can be structured is the number of documents to the power of itself. This is 10 to the 16 to the power of 10 to the 16. 10 to the power of a few hundred is already exceeding the number of estimated protons in the known universe. So one can make similar arguments about this system as people make in regards to quantum computation systems: we can even for a very small network build something that is totally impossible to simulate on a classical computer.

The power of a fully self-replicating and evolving Open Web on this scale is that documents can describe the replication of *physical* things, and the replication of the documents can include replication of the things. If things we use in our lives are replicated rather than purchased or mined, it changes the basic assumptions about what value is. Note that like “set”, “thing” is used in a maximally general sense to include things like “a feeling of awe at the largeness of a tree” or “the tendency of cats with white fur to cause a change in the appearance of black clothing”. The word “thing” is used here as a placeholder for *anything* which human language can be made to describe or point to.

In order to build these documents we must first define the idea of what exactly a self-replicating document is, and how it fits into more general concepts of self-replication. To that end we will take an excursion into the math known as set theory, which is the next section of this paper.

1.4. 4. Self Replicating Sets/Documents.

1.4.1. *Motivation and definitions.* Set theory, is the mathematical study of sets. Sets are simply “collections of objects”. The idea of a set as a collection of “objects”

considers the idea of the “object” at a level of generality perhaps shocking to non-mathematicians. “Object” here can mean *anything*. Mathematicians generally mean by “any object” any object which a mathematician might talk about. However in principle it can be anything that anyone might talk about (as we seek to generalize these ideas beyond mathematics). For the purpose of this work we will define a generalized object to be anything which human language can possibly describe. Any word, symbol, or collection of words or symbols which point to something—that something is an “object”. And a “set” is just a collection of such general objects.

The notion of a generalized object is familiar to modern computer programmers who use the idea of “object oriented programming” to create generalized objects which are used to build linguistic handles in human language on computer programs. Thus a modern programmer might define something abstract like “shopping-cart” for an e-commerce website, and then that object will have properties like “list of objects” and “total price”. We choose to take the path taken by foundational mathematics and have our basic concept from which all other concepts will be built be the collection of objects (which are themselves objects) be the fundamental idea.

In order to understand the motivation of this work it is necessary to trace very briefly the history of set theory. Through the end of the 19th and beginning of the 20th century there was a vast effort by some of the most brilliant mathematicians in the world to construct a universal mathematical theory base on the theory of sets and symbolic logic. Axioms were proposed, used to prove things, argued about, and re-written. The goal was to base *all of mathematics* on the axioms of set theory, and to go from there to a universal system of truth in which statements may all be proven to be true or false.

People like Bertrand Russell pointed out that such systems can create paradoxes that make it impossible to create a self-consistent system of logic/truth/math. This paradox can be summarized by considering the “list of lists that do not list themselves”. The list defined here is a list of lists. Is this list on itself? If it is, it cannot be by the definition of itself. If it is not, it must be by the same reason. In spite of having publicly stated this paradox, Russell and his co-author Alfred North Whitehead wrote what is now considered a seminal work in mathematics *Principia Mathematica* (not to be confused with a book by Isaac Newton of almost the same name), which attempted to create such a universal basis of mathematics. While the achievements of 20th century set theory, logic and analysis are fantastic and useful, they ultimately failed in their goals, and this was proven mathematically by Kurt Gödel in 1931.

In the post-Gödel world we should take for granted that no universal logical construction can be built which defines truth and falsehood without contradiction. Gödel’s proof presented a fork in the road intellectually. We could have used this as the sign to go back through mathematics, accept contradiction as part of life, and build a math based on desired outcomes. In some sense this is what society did by mostly ignoring the work of most post-war mathematics (with some very narrow exceptions like number theory for cryptography). While professional mathematicians took the opposite fork, building increasingly complex systems based on each other, where a vast tower of ideas linked by formal logic built up from the axiomatic set theory of the early 1900s to create a bridge to nowhere. The sole purpose of most mathematical concepts and theory today are to advance the career of working mathematicians. We forget, both outside and inside mathematics, that people used to believe ideas in math actually *mattered*. We also forget that mathematics

has for thousands of years been one of the most powerful tools the human mind has for understanding and interacting with our world, and indeed mathematicians have traditionally played a central role in the largest power structures throughout history.

Having proved that a universal truth machine cannot exist, we may now abandon the project of early 20th century mathematicians such as Russel and Whitehead and proceed to reconstruct axioms of set theory based on a *desired outcome*. This system will not be judged on its ability to prove theorems, eliminate logical contradictions, or get tenure for math professors. It will be judged *only* on its ability to improve the human condition. It is time, finally, after almost a full century, to inherit the true legacy of Goedel.

Right now all of humanity is locked into one giant self-replicating set which has as elements all of industrial society. The purpose of this work is to create a set theory which enables people to construct sets which create the maximum amount of human freedom. To that end, we seek to make sets have elements that are defined as generally as possible and also which always have the *desires* of the creator of the set as an element. When we move forward replicating these sets in the new society we build, each act of replication involves also replicating this desire. We therefore only replicate that which transmits a desire that we consent to and agree with for some reason.

Let us make some statements here about the sets we want to define. We define self-replicating sets as sets which contain as elements the means to replicate themselves. We also state that since our goal is to create the objects of our desire with our mathematics that whatever that desire or intent will always also be an element of the set. We maintain the tradition of both formal set theory and object notation on computer science and define sets in writing by listing elements separated by commas and contained in “curly braces” “{” and “}”.

We also state that in general the sets we will construct will have a primary element, the replication of which is the purpose of the set. There might be many other elements and subsets which are needed for replication, but the *primary element* is defined as the element the replication of which represents the *primary intent of the creator of the set*. We thus make the human will, desire, or intent a fundamental element of our entire set theory. To distinguish our set theory from that of mathematics as it exists today we coin the term “set magic” to be the theory of sets which contain both the desires of the creator of the set and the means to replicate the entire set. This is loosely based off of the quasi-secular use of the word “magic” from both chaos magic and Thelema magick to indicate the attempt to impose the human will on the world we find around us.

Thus a way to express the most general possible self-replicating set from our newly defined set magic is:

```
set = {
    desire,
    object of desire,
    means to replicate this entire set
}
```

Note also that in order for the whole set to replicate, the desire of the creator must also replicate. This is what in our existing system is called “marketing” and “sales”. Without first convincing another mind to share our desire for replication, it will not happen. The power of the open web is thus not just about replicating documents

but replicating the desire to replicate documents—in modern parlance, just marketing. The commercial web has proven excellent at this, and the open web has the potential to vastly improve on that.

We further state from the outset that all self-replicating sets have externalities. These will be elements or subsets of elements which draw on the resources outside any given “closed” system in order to replicate. Thus rather than attempting to build a conceptually closed system and then finding that it is not closed as did earlier set theorists, we accept that we are building a less formal construction that will always have an externality and that we must describe what this is in order to properly define a set. The number of degrees of freedom of this externality is what limits the overall degrees of freedom of a system. For example, if an element of a set is “text reading system” that can be on any of many different technologies. Whereas if something has an externality “lithium ion polymer batteries”, the entire system is reliant on that one technology and any threats to large scale extraction of lithium from the Earth are threats to the whole system. We thus seek to constantly struggle to improve on the externalities, with the ultimate goal for them to be in equilibrium with the living Earth.

A self-replicating document is an example of a self-replicating set (a “document” is just a collection of symbols, hence a set whose elements are symbols). This document is created as itself a self-replicating document according to the prototype we propose for the whole system. We now set forth to define the set which is this paper in theoretical terms, then to describe all the subsets which together make for a self-replicating set which can be evolved into other self-replicating sets. We now proceed to formally define the set which is this document.

1.4.2. *This Set.* The prototype self-replicating set we define in this paper is itself this paper, and is formally defined as follows:

```
On Self Replicating Sets[This Paper] = {
  The desire of the author to describe self-replicating sets,
  A description of self-replicating sets,
  The means to replicate this set
}
```

The first of these will always be a part of the system of sets we are constructing here: desire. All sets are defined with an element which maintains the desire of the author/creator/artist, and this is maintained as sets replicate to ensure that sets only replicate with the intent/consent of someone. Furthermore, we separate the thing being replicated from the means to replicate it. Elements are generally themselves sets which are broken down into more finely defined elements. This paper is part of the “description”, but what formally is “this paper”? We seek to define it as a set, or a collection of elements contained in “description of self-replicating sets”. This includes the following elements:

```
Description = {
  narrative structure,
  definitions,
  digital text document,
  pdf document
}
```

Now again we break off the replication methods from the thing being replicated. Replication is to happen on the Open Web. That is, it must have self-contained and self-replicating code that can copy itself from any web server to any other, and be edited after being copied, then copied again from the new instance so that the information is totally decentralized and evolves naturally as it's copied and edited. We also need this document to include instructions in human language(English in this particular instance) on how to replicate the whole system, by either buying a domain and setting up paid web hosting or building a physically local web server to server the files. This document will contain that information. Furthermore, the replicator set must include the other media that are used to replicate the whole set, including content on commercial social media.

```
Replication = {
  code replication,
  server replication,
  pdf replication,
  in-person pitches and classes and discussions,
  media: email, post cards, posters, signs, commercial social media
}
```

1.4.3. *Code and replication of code.* The elements of the self-replicating code that can propagate this document across the Open web are as follows:

- replicator.php
- dna.txt
- filesaver.php
- fileloader.php
- README.md
- editor.php
- index.html
- pageeditor.html
- dnagenerator.php
- text2php.php

All of this code can be edited using the program `editor.php` which runs on any server that has php installed(most web servers). The file `dna.txt` represents this list of files, which `replicator.php` uses to fetch them and copy them to a new server(see below). `README` is the text of this document itself, and the name of that file is set by the standards used on Github so that by default any self-replicating document that's put up on Github has its content readable immediately. The format of the `README` file is by default Markdown. The save and load scripts are required to edit files on the server, and `pageeditor` is the page that uses these files to edit the main manuscript `README.md`. All php files are stored as `.txt` files so that they can be readable and easily accessed from the open web. The program `text2php.php` copies all the files in the php directory to the main web directory and changes the file extension from `.txt` to `.php` so that they can run. The file `editor.php` edits the copies in the `/php` directory, and then running `text2php` makes those programs executable.

1.4.4. *Server replication.* In order for replication to take place from server to server we need the ability to “colonize” new web servers with this code. This is done in any of several ways. Right now the main way is to buy a domain(usually about 10 dollars for the first year), pay for web hosting(5-20 dollars per month), then put the

replicator program on the new server and run it. The second method used is to put a web server on a Raspberry Pi, a computer that can be bought for as little as 35 dollars and fit in a pocket which is excellent for serving web files over a local network. This allows for a grey area open web that is open to anyone on a local wifi network, and can see the rest of the Open Web but cannot be accessed by users outside that wifi network. In either case, the replication of the server consists of placing the file “replicator.php” in the main web directory of the new server, then pointing a web browser to [your new servers web address]/replicator.php. This will then cause the program to run, copying the rest of the system and linking to the main page which will display the newly replicated document.

The technical details of this process must be described briefly here in order for this document to truly self-replicate. We recommend buying a domain and getting shared hosting on dreamhost because they have proven to work with this code and are affordable and not a scam. One can also use any of several free web hosting options, including 000webhost. In both cases you will find a file editor screen(this can be a little frustrating to find but always exists) which will allow you to create new files and edit them. Use this to create the file replicator.php and copy the code from here into it, save it and close it. On the Raspberry Pi, replication starts by making a new flash card with the operating system on it. One must then install the web server and php language using this set of instructions. After that one can move to the main web directory, copy all the files, change permissions, delete the existing page, and run the replicator:

```
cd /var/www/html
sudo rm index.html
sudo curl -o replicator.php https://raw.githubusercontent.com/LafeLabs/thing/master/
php replicator.php
sudo chmod -R 0777 *
hostname -I
```

Once the whole thing has been copied, to have the *next* copy be a copy of this copy and not its predecessor, use the code editor to edit the file replicator.php so that it points to the global url for the dna on your new page, not the previous one. This is done manually for now. So if you do not do it, the next copy will be not of the new document but of its predecessor. Note also that once this system is installed anyone can run any code on your server, so no private or personal data of any kind should ever share a server with this system. This system assumes that no private data exists unless it's on a physically isolated wifi network, and it must remain separate from the private Internet(especially anything with e-commerce, as stealing financial data would be trivial if that is ever done).

Another good practice as one works with files on the open web is that since we must assume all files might be deleted at any time, but we want the current copy to remain readable not just by us but by all users on the Network, we constantly back up text to free and open but non-editable paste sites like pastebin.com. To fork the whole code, one can edit on a live Open Web server, back up to a pastebin, and copy. But one can also create a github repository, copy the code locally to your hard drive, edit it on there, run dnagenerator.php to make a new dna file, then point your replicator to your github repository so that many copies can be made without the original being corrupted(using a hybrid between the password-protected space of Github and the Open Web which copies files from there). Note that while one

can use any code editor for editing the local copy, we can keep a consistent system with the Open Web servers by running `php -S localhost:8000` at the command line(I assume people who fork the code in this way know what this means) and connecting a browser to `http://localhost:8000` to edit *in situ*.

1.4.5. *Pdf document.* As useful as the Open Web is, it is also useful to have documents in a traditional format which is compatible with physical paper printers to make copies we can carry outside of digital readers. To do this we have several options. We can print from the browser(which will look bad), we can convert to Microsoft Word which will corrupt the file and also look bad, or we can use the LaTeX system which will look great but take more work. For this initial instance we focus on the LaTeX version. As this document replicates and evolves hopefully more skilled users than the initial author will make smoother systems for conversion but right now a combination of the Haskell library “pandoc” and manual editing of the output file are the easiest way to convert from markdown to LaTeX. A document produced in this way is included in the replicator of this set.

```
pandoc -o paper.tex README.md
pdflatex paper.tex
```

1.5. **5. Generalization and Social Implications.** All ideas which we desire to propagate on the Open Web need all the typical means of use of media to try to convince others to replicate the document. In some cases we seek to explain the whole thing in a tiny capsule of information, as with the “elevator pitch”. In other cases we seek to show by example the power of these ideas. The media we expect will be used in the spread of this system include *all* media in the most general sense, including physical things like machines which carry various writing on them. Any physical thing can have a domain name on them which points to a document which describes how to replicate the physical thing. This extremely generalized definition of self-replicating document(as a type of self-replicating set) means any physical object can be thought of as a self-replicating document. Combined with the “Open Web” defined above, this can create an entire universe of useful things we can use to make up the fabric of our lives, building sets to live in which are independent of the existing industrial order.

Ultimately if we can build sets that we have the capability to evolve based on our desires, we can push that evolution toward what we call a “technological complete set”. That is a set which describes a full self-replicating system that we can live in, which can exist in equilibrium with its environment just as pre-industrial societies did before the whole world was consumed by one very destructive set. In a complete set, the people who live in that set(we place ourselves conceptually into the set) have everything they need for a good life such as medicine, abundant food, clean water, the ability to live in a comfortable temperature etc. In addition, a set is not complete if it is out of equilibrium: if a set requires constantly destroying things and not replacing them to exist it is not complete. As we look to the future this is possible in a way that is totally different from what was possible before the rise of industrial society. A future complete set based society has no reason to mine, since the quantity of material that has been extracted from the Earth and processed into very ordered structures is more than enough for a large human population to live on indefinitely.

This Technological Complete Set does not need to be designed and built by any one person or group. All that is needed to achieve such a set is to build sets which

people have the capacity to evolve based on desire, and to impart into a group of people the desire to achieve this set(given the assumption that such a set is physically attainable). This document is meant to describe such a set, but also to serve as a seed which the author hopes will evolve in such a direction. By itself it is probably insufficient to build a large complex system, but it provides a prototype for a number of other self-replicating sets which we will construct and replicate over the Open Web. It is the nature of such sets that as they are created and released into the wild, they will all build on each other with network effects and that a small amount of exponential growth early on can create very large effects as the system evolves.

What is next in this program? The self-replicating sets which are currently being created and released are largely about replicating symbols. Symbols and logos play a very powerful role in how our minds process the world around us. The ability to create a self-replicating symbol which has some intention imposed on it is one of the most powerful forces in our world today. This is the power corporations wield with their brands, logos, and marketing messages. By building systems for very rapidly creating symbols, giving them meaning, and replicating them, we empower the masses with this same power. The media for the symbols includes artistic tools for physical media creation(stationary, wall art, postcards, signs) and digital media creation(vector graphics of simple geometric logos). From this power, we hope to build a fire that consumes the media landscape and transforms the nature of human existence on this planet.