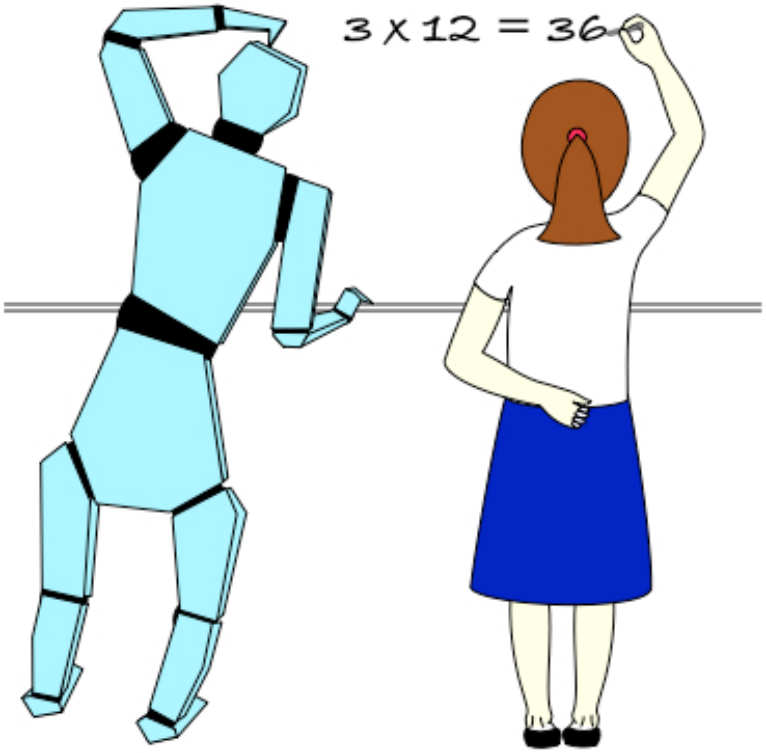
Reduction Considered Harmful

The goal of any science and engineering education is to give the student the ability to “perform Reduction”. Some of you may not be familiar with this term, but you have all done it. It is the most commonly used process in science and engineering and we tacitly assume we will use it at every opportunity. Therefore there has been little need to discuss Reduction as a topic outside of epistemology and philosophy of science.

In what follows, I will be making the claim that for the ***limited purpose*** of creating an Artificial General Intelligence (AGI) we must avoid this common kind of Reduction. This article (second in a series) will discuss what Reduction ***is*** and why it is ***useless*** in the domains where AGI is expected to operate. The third article will discuss why it is also ***unnecessary***. The fourth article will discuss available ***alternatives***. As a bonus, we will come to Understand what it means to Understand something.

I use some common words like “understanding” as technical terms with specific and unchanging definitions and when I do, I spell them with a capital first letter. I define most of these terms here; for others, see Wikipedia.

When you were in grammar school the teacher taught you about numbers and simple arithmetic. This was your first introduction to a Formal System. Then you learned how to do long addition. Take this plus this, write the sum there, and if there's a carry, you write it there... this was your first Algorithm.



Then you faced your first story problem. If Holly has three boxes of candy and there are twelve candies in a box, how many candies does Holly have? After thinking about it, perhaps making a drawing, you realized that you needed to “do 3 times 12 equals 36”. This was your first ***Reduction***.

Nobody ever taught you how to solve any and all story problems you would encounter in your life. Nobody can, because Reduction is not an Algorithm; each Reduction differs in the details. Sure, many problems fall into classes of similar problems. We could imagine a formula for this class of story problems; we could call it "A-containers-of-B-items-how-many-items". It would specify that we get the total number items by multiplying A and B.

A-containers-of-B-items-how-many-items is a “Model" and it is equally valid for candies, eggs, and apples; it doesn't specify what we are putting in the containers or what kind of containers we have, just how many containers we have and how many items there are in each. The Model is "context free". You yourself must provide the analysis of the context. If you have cartons of eggs, you must Understand that the cartons are containers and the eggs are items, not the other way around. You must simplify – "Reduce" – the real life situation ***in your mind*** so that you can select the appropriate Model, plug in the correct values in the right places, then run the Model (by performing multiplication or whatever operation the Model uses), and finally you must interpret and apply the output of the Model to your real life situation. The answer is not "36", it is "Holly has 36 candies".

Models of this kind, like model airplanes, are ***simplifications*** of reality. You simplify by ignoring a lot of details; model airplanes don’t have built-in instrumentation. You are in essence cutting out a tiny piece of your reality as a simpler and purer subsystem, trying to make it independent of the context you removed it from. You could say a million things about a box of candy – its size, color, materials, shape, etc. For this Model, you ignore everything except “how many items does it contain”. ***Reduction discards context as a nuisance***.

Let us use as another example one of Newton's laws, "F = ma" . This is an equation and therefore also a Model, not much different from A-containers-of-B-items-how-many-items : It states that whenever we have an accelerating mass we can compute the force causing this acceleration by multiplying the mass and the acceleration. If you want to use this, you need to Understand forces, masses, accelerations, and how to measure them. You are taught thousands of such Models throughout your education. Formulas. Equations. Rules. You have invented some of your own.

You may notice that some of your own Models work all the time whereas some only work most of the time. We share our most reliable and most useful Models with each other and we call this sharing Science. The general idea of creating simple Models describing fragments of reality is called Reductionism and while under-appreciated, it is one of the greatest inventions our species has ever made. It has solved innumerable problems since the seventeenth century when people like Descartes, Galileo, Newton, and Bacon refined and formalized the known practices of incremental Model creation into what we now call the Scientific Method. Of course, Reduction itself goes back much further.

Several kinds of Reductive simplifications are possible. Above we discussed “extraction from the environment/context”. If the extracted subproblem is still too large to analyze then we can repeat the process, dividing the extracted system into subsystems. If the scientific discipline we are operating in doesn’t provide answers then we might want to analyze the system in the “next lower discipline” – for instance, biological problems might be analyzed in terms of biochemistry. If we have a number of similar Models, such as similar equations describing several phenomena, we might want to consider whether a more general Model might cover them all. When we have multiple Models describing the same phenomenon we may prefer to use the simplest of them that utilizes the information we have. When considering various structures, such as chemical

substances, we might want to find a description of matter that describes them all, such as quarks. The quest for a Theory of Everything is a truly Reductionist endeavor.

All of these flavors of Reduction have been identified by philosophers and epistemologists and they love giving them names like “Ontological Reduction” and “Methodological Reduction”. I find these names confusing and like to discuss them all as “Reduction”. Context and experience will dictate which kinds are appropriate. This larger scope is what makes “**Reduction**” different from “**Abstraction**”. In simple cases we could have used either word, but “Abstraction” is typically used only where we actually have found a correct abstraction, so its use implies the process is perfect. This article is highlighting the problematic cases and I therefore prefer “Reduction” since it won’t let us forget that the process is fallible.

To give some examples, when a Reductionist (my shorthand description of someone attempting to solve a problem using a predominately Reductionist stance) attempts to understand a frog, they take the frog into the laboratory (isolation from environment), dissect it (subdivision) and study separate subsystems such as digestive system or blood circulation separately (again, subdivision). In order to understand what the frog’s blood does they need to drop disciplines from biology to biochemistry and study how hemoglobin transports oxygen. In contrast, in a more Holistic (context utilizing) discipline such as Ecology, we would study how the frog interacts with other frogs and its environment.

When studying a predominately Reductionist discipline, such as physics, you often see textbook phrases like "All else being constant..." or "In a closed system..." which indicate that some degree of Reduction has already been made, so that what follows in the problem statement is all that matters. By giving you a large part of the Reduction, by means of example, the textbook is coaching you to form Intuitions about how to perform Reduction. These intuitions will allow you to do this yourself later, in the real world, in vastly more complex situations. The key intuitions are those that tell you which parts of the real-world context you can safely ignore.

Reduction is an Intuition based skill. We can tell, because we get better with practice as we get more experience, which is a hallmark of any Intuition based skill. More about Intuition later; for now, let’s just say it is a subconscious problem solving method that utilizes our past experiences without doing Reduction. This sounds confusing only because we’re starting to glimpse the core of the problem. At this level, Reduction is the goal, not the means. We cannot **implement** Reduction by using Reduction. Please stay with me; we’ll circle the issue until we see it clearly.

Models are created and verified by scientists. But if you are a scientist doing research then nobody tells you what Model to create next, or what hypothesis you should invent to be later tested and verified by experimental work. Graduate school teaches you the verification process including things like confidence intervals, chi-squared tests, and peer review but just like when learning to solve story problems in grammar school, nobody can tell you how to invent hypotheses to explore. At best, they can coach you to learn it on your own, from experience. This is another hallmark of Intuition based skills: they cannot be taught as high level rules, they have to be experienced bottom up. This is, incidentally, also the difference between Teaching and Coaching.

Engineers ***use*** the Models that scientists ***create***. Scientific Models are very reliable when used correctly in the situations they were designed for. As an engineer – a Model user – you define the borders of your Reduced subsystem when you are cutting it free from its environment, from its context. If you perform this Reduction incorrectly, then the Model you use may well give you a wrong answer. You may have left behind some context that would have affected the result. I call this a Reduction Error. There are many kinds of these: Sampling errors when measuring waveforms, selecting a Model that ignores friction when it actually matters, or ignoring emergent effects in complex systems. And in Models of complex systems, such as in rule-based Expert Systems in the style of AI of decades past, most failures are caused by incompleteness of the Model. When the system does not have rules for the current situation it will fail, and often fail spectacularly and catastrophically. This tendency to fail in surprising ways at the edge of the system's competence is called "Brittleness". ***Brittleness is the main symptom of any attempt to Reduce the irreducible***.

And once again, just like in grammar school, nobody can tell you where to make the separating cut, what to discard as irrelevant, or exactly what measurement or value goes where in the Model (the formula or computer program). They can't even teach you a sure-fire way to decide which Model to use, out of thousands that you know. Perhaps you need to combine several Models or perform some algebraic transformation to some Model to get one that fits the current situation. Perhaps you should split your problem into several simpler parts and use a different Model for each. This kind of analysis is done through conscious Logical Reasoning – you manipulate Models to re-factor them into new Models that are more useful to your real-world situation. But even knowing which direction to re-factor in requires Understanding, experience, and Intuition.

To summarize so far: Reduction, Model creation, and Model use all ***require Intuitive Understanding*** of the problem at hand and of the problem domain. This Understanding cannot be achieved (implemented) by using Models of anything in the problem domain.

Computers run programs, which are all Models. What computers cannot currently do is to perform Reductions, because current ***computers don't have the necessary Understanding*** of their target problem domains. This is the sharp line separating computer science from AGI. The difference between these disciplines is, in a nutshell, simply ***who*** is doing the Reduction. If a human is doing the Reduction, then the human is just programming (creating a new Model) or running a program (using a Model). If a computer could perform a Reduction without human help, then it would have demonstrated that it actually, truly Understands this problem domain, and thus should be labeled an Artificial Intelligence.

It is very tempting to use these observations to create a formal definition of Intelligence:

Intelligence is the ability to perform Reduction. Artificial General Intelligence is a computer program with the ability to automatically perform Reduction.

Given the discussion above about what Reduction is I have to say this could be a pretty clear-cut definition of Intelligence, at least if we compare it to commonly used informal definitions. We will remember this and put it away into our epistemological toolkit. It makes a worthy ***goal*** for AGI research.

We note that many historical AGI projects did not have this as a **goal**; they used Reduction as the ***means***. This was (and is) a major mistake. If we want to create an AGI capable of doing Reduction, then ***we*** should ***not*** be doing this Reduction for it while we are writing the code for the AGI. Consider projects like CYC, the most ambitious Reductionist AI project in the world. Their ultimate charter is to create Ontologies describing everything in the world. The CYClists (the Ontological Engineers at CYCorp) have entered millions of human-made Models expressed as statements in a language roughly equivalent to First Order Predicate Calculus into the CYC system. But these high-level concepts are unconnected – not "grounded" – to any web of low level experience that could provide the Understanding that the system would require in order to do Reduction on its own. Without Understanding to support it, CYC will forever be unable to add to its own knowledge base. It will always be dependent on humans doing all necessary Reduction before it can even start Reasoning about a new problem. Any sufficiently Reductionist AI is indistinguishable from programming.

A more recent example: If asked, the Semantic Web enthusiasts are divided about whether they are working on an AGI or not. Some say it's just a system to streamline web-based commerce. Others work hard at creating snippets of Ontology about the web world (and to some extent, the real world) in OWL and other ontology languages, hoping to create a CYC-like but larger-than-CYC distributed Ontology. This is clearly nothing but Modeling and Reduction. Anyone hoping that the Semantic Web will become a significant ***enabling*** component in a future AGI is going to be disappointed.

Note that all programming is Reduction. Even when writing an AGI, we're using nothing but Reductionist methods. The main issue is ***"are we modeling learning or are we modeling the world"?*** In other words, what domain are we Reducing from? If you want to write an AGI, then it is perfectly reasonable to invent a theory for how Intelligences learn about their world from experience, to create a Model of Intelligence itself if you will, and to implement that Model as a program. You can test it by letting it experience some input in its target domain, such as language, vision, or the real world through sensors. But you must let it experience the world, learn from its experiences, and to figure out its own abstractions all by itself. Eventually the system may become experienced enough to start doing Reduction and to make its own naive Models of its target domain. Do not attempt to do any Reductions on its behalf; it would be counterproductive cheating, like cramming questions from past exams rather than trying to learn the subject.

Further, the "Let's start by giving it enough Models so that it can bootstrap from there" argument used by CYCorp and others is counterproductive. The easiest Reductions are the ones we can do directly on the low level input stream. As an example, if we are writing an AGI program to learn to Understand a language like English, then we must refrain from parsing the text into words before giving it to the learning algorithm. If the system cannot figure out that separators like spaces separate the text into "recurring subsequences" then the system won't be able to figure out higher level concepts either. Testing and debugging will always be easiest at the lowest levels of complexity. Do not waste this opportunity by Modeling "what words are" just because writing a word detecting parser is something you know how to do. Instead, feed it characters one by one and use the constructed system's ability to discover the Model of "words are

recurring sequences of non-separator characters" as a test case for whether your theories of learning are correct; at Syntience, this typically takes about four pages of reading Jane Austen. Same goes for movement detection in vision and for graceful locomotion in robotics.

Learning can't be very difficult since frogs can do it. I estimate that the most advanced theory of general learning could be coded in 10,000 lines of Java or C; the code base at Syntience for our experimental AGIs typically clocks in at that magnitude including debugging and evaluation subsystems. The details may be tricky and few people claim to have implementable theories of general learning but it is not going to be a big coding effort. We might learn how learning works from Neuroscience; Numenta is pursuing this. But it will likely be much faster to derive this information directly from epistemology; that's the path taken at Syntience. Contrast either of these to any attempt to describe the world, a task that by definition will never be done. If a cheap and clever way to create an AGI exists, then it's going to be to create an ***Understanding*** machine that can ***learn*** on its own, rather than one that has to be ***taught*** what the world looks like. Perhaps I should create a little rule of thumb:

If your AGI requires more than 10,000 lines of code then you are doing it wrong.

Now I'd like to examine the limits of Reduction and Scientific Models, and the limits of Science. Actually, it is not so much about the limits of Science as a whole as it is about the limits of physics, mathematics, and computer science. Physics is the most Reductionist "main" discipline, and the other two are very, very Reductionist "support" disciplines for all of science. The best known examination of the limitations of physics is Erwin Schrödinger's "What is Life" that observed that physics could not explain biology, in other words, that life and living phenomena could not be ***Reduced*** to the simple principles that physics had identified without losing the essence of life itself. I often joke and say that as the Reductionist is cutting apart a living frog to "see what makes it tick", the "ticking" disappears, but all the pieces of the frog are still there. Schrödinger's book gets re-read and re-debated by each generation of life scientists. The Life Sciences have found ways to make progress without using Reductionist Models. We will discuss these methods in the fourth article in this series, but for a preview, watch the video of my talk "Science Beyond Reductionism" at http://videos.syntience.com.

This examination of the limits of Reductionist science has been done many times by many people; the result is often a list of meta-types of problem domains where Models cannot be made or reliably used. These lists end up rather similar; see for instance the Wikipedia page about Complex Systems.

I like to divide this list into four parts:

1. Chaotic Systems.
Chaotic systems are generally unpredictable, are sensitive to initial conditions, and are easily perturbed at any time. The whole point of a Model is the ability to predict the future behavior of the modeled system, and Chaotic Systems are unpredictable in the long term; this is the definition of a Chaotic System. Chaotic behavior can result from many things but most commonly we find some combination of components with multiple

interactions with numerous other components, components with hidden state (memory), and components with non-linear responses to inputs, which may lead to race conditions and indeterminism. Examples of such components could be neurons in the brain, animals in ecologies, cells or even organs in multicellular bodies, humans in societies, or corporations and other agents in market economies. We immediately recognize that all these problem domains are really difficult to model.

2. Irreducible Systems

These are systems that behave differently if you attempt to split them up or isolate parts of them from their environment. All Models are always simplifications, but for Irreducible Systems all possible simplifications discard something vital which means the predictions made by the Model will be incorrect. Take a frog out of its habitat into a laboratory and it will behave very differently. Blood in circulation in a body behaves very differently from blood in a test tube. The price of a company's stock a month from now depends not only on the company and its actions but on the economy at large. Consider the laws of Thermodynamics; they only apply in closed systems. Any attempt to use them in the open world will fail because the energy interactions with the environment cannot be fully tracked.

John McCarthy and Patrick Hayes, two famous pioneers of Reductionist AI, observed in 1969 that Irreducibility was a fact of life and named it "The Frame Problem". Except for some narrowly defined special cases, no real progress has been made on this problem. This is not surprising, since the problem is the Reductionist stance itself.

3. Emergent Effects

Emergent effects are system-level effects that cannot be observed at the component level. As an example, a single water molecule doesn't have a temperature since temperature is defined only for groups of molecules interacting with each other. And depending on the temperature, these water molecules will form water vapor, liquid water, or solid ice. These three have very different properties. Could these different behaviors be predicted from the properties of individual molecules, such as van der Waals forces? It is difficult, and the reason we would even go looking for the connection is because we would have observed these emergent effects at the system level.

Consider a car. Its quality, lifespan, drive-ability, and beauty are not in any single component of the car. These are all in every component, in their design, the materials used, the design as a whole, the effort, precision, and conscientiousness that went into the manufacture of all parts and their assembly, the experience of the designers creating it, etc. All of these things matter. Same thing for human lifespan and beauty. These are all emergent phenomena that cannot be taken apart or simplified since everything matters. To manipulate them, for instance to improve the quality of the next generation of car, requires a Holistic (context utilizing) stance, Understanding, and experience.

There is also downward causation. Consider a single word, like "like", alone in the middle of a blank page. What does it mean? "like" has about a dozen major meanings and hundreds of shades of meaning. In language, words rarely stand for unique concepts. Words get their meaning from context – from surrounding words, from the topic of the page, the language used, and the shared experience of the writer and the

reader. As you are reading a page, the words you are reading build up a high level context in your mind that influences how you interpret each individual word (low-level observation) that follows. This high level context exerts downward causation on the lower level word disambiguation process. As another (rather silly) example of downward causation, if some set of neurons in your brain decide that you should take a break and walk to Starbucks, then the whole brain will have to come along. The emergent effects, that cannot be observed in the individual components, nevertheless affect the behavior of the components. This makes some Reductionists uncomfortable since in Reductionist systems, all causation is of the upward kind and is known in detail and this downward causation sounds like some kind of ghost in the machine.

Getting back to AGI... intelligence is an emergent phenomenon. It must emerge from the interactions of non-intelligent components. In some sense, this statement is trivially true, since if we had intelligent components we'd be done before we started. But the design of low-level components like simulated neurons to generate emergent effects like intelligence is still largely unexplored territory. Researchers at companies like Syntience and Numenta are exploring this nascent field and are exploring how to work with what I have named "***Connectome Algorithms***". The ability to understand and ***manipulate*** emergence should be a required skill for any AGI researcher.

4. Unreliable Information

This one may be easiest to understand. Reductionist Models, Logic, and in general any scientific approach require good and solid input data. But in many problem domains, such data is not available, and worse, can never be made available to a degree sufficient to allow us to reliably use Reductionist Models. In our everyday life, information is incomplete, ambiguous, incorrect, and sometimes patently misleading. How well is a Reductionist Model going to predict the future if we lie to it? In SF novels you sometimes find the term "GIGO" which stands for Garbage In, Garbage Out. It is amazing how seldom you hear programmers and other Reductionists use that term. It is rarely discussed since in many situations, nothing can be done about it.

A cute detail: The brain is internally unreliable. Neural signals are propagated by neurotransmitter diffusion across a synaptic gap. This means there is an indeterminate delay before the receiving end gets enough molecules to notice the signal; given the high parallelism of the brain, we get race conditions everywhere. No sane Reductionist would design a system like this. But apparently the brain has enough Holistic (context utilizing) redundancy and other checks and balances (as opposed to Reductionist solutions like checksums and retransmission) to create an ***emergent robustness***. And this robustness actually extends outside of the brain. If your neurons can sort out their collective mistakes then they are likely to be able to use the ***same*** mechanisms to guard against contradictions and lies in the input data such as it is received by the senses. Emergent Robustness is the cure for Reductionist Brittleness. You'll know you are on the right track when your AGI system makes human-like mistakes.

So we have four flavors of impossible-to-Model systems: ***Chaotic Systems, Irreducible Systems, Emergent Effects,*** and ***Unreliable Information***. Each one of these will prevent Reductionist Models from being made (or in the last case, from

providing useful results). But in many of the hardest problem domains we find ***all four of these at once***. Following the lead of Dr. Kirstie Bellman at Aerospace Corporation I call systems that we cannot Model "Bizarre Systems" and their problem domains "Bizarre Domains". It is good to have a memorable label for these; we need to be on the lookout for telltale signs of these kinds of systems so that we don't waste our time attempting to create Models of them.

We find examples of Bizarre Domains in many places outside of the Reductionist haven of physics, math, and computer science. Life is Bizarre, and this affects the Life Sciences. Genomics, Physiology, Ecology, Psychology, and Biology in general are full of situations where Models won't work. Other Bizarre Domains: We gave the Nobel Prize in Economics to Friedrich Hayek for telling us that the Economy cannot be modeled. It doesn't stop people from trying to make computer based Models of the stock market, and small gains may be possible for a while, but like all Reductionist contraptions these trading programs will fail. And when they do, they fail catastrophically, which we already said is the hallmark of Brittle Reductionist Models.

Learning a human language takes a lot of time and effort because we need to gather a lot of experience before we Understand it. But human languages are Bizarre. This is why the use of word frequency based data mining algorithms like TF-IDF, grammars, taxonomies, and ontologies (which are all Models, at best still somewhat useful for limited purposes) will never lead to true Understanding of language; this requires a Holistic (context utilizing) approach.

For AGI the most important Bizarre Domain (besides language) is our everyday mundane reality. It is deeply complex, ever changing, and contains many agents with goals at odds with our own. This is where AGI must operate, performing the simple everyday tasks that people with normal intelligences do so effortlessly and that no hypothetical Reductionist AGI could ever analyze. How come humans can do this? We learn Reduction in school, and use it in science and engineering, but in our everyday lives we operate Holistically and Intuitively. ***We don't Reduce, we just Understand***. This will be the focus of the next article. The point I wanted to make here is:

In the very domains where AGI has to operate, Reduction is impossible. The confusions about what Understanding is, who should be doing the Reduction, and when Reduction is even required are the main reasons we don't already have working Artificial General Intelligence.

– Monica Anderson

First article in this series
Reductionism in Wikipedia
Science Beyond Reductionism video
Erwin Schrödinger: What Is Life?
The Frame Problem
Bizarre Systems on artificial-intuition.com
Bizarre Systems video
CYC
Scientific Modeling in Wikipedia

Illustration by author