Against Three Formalist Computer Program Verification Methodologies

Talk Assumes Extended Abstract and Discusses Vienna Circle Empiricism and the Importance of Critisizing Program Verification in the context of the Philosophy of Computation.

Steve Meyer - Pragmatic C Software Corp (smeyer@tdl.com)

Presented June 17, 2008 at ECAP 2008 Montpellier France

1. Introduction

In this talk, I want to accomplish a number of related objectives: 1) Explain why science, methodology and philosophy are really the same activity. 2) Introduce the metaphysics free terminology of the Vienna Circle empirical scientific method (especially the Neurath principle), 3) Explain why the (natural) philosophy of computation has wide importance for science in general, 4) Discuss the computer program verification debate, and finally 5) discuss what I view as the main problem holding back progress in understanding computation. Namely, the metaphysical and seemingly unconscious assumption that computation is finite (computational finitism). This dogma is not just negatively effecting computer science but other sciences as well because experiments in nearly all sciences now involves computer codes.

Consider the following problem suggested by the physicist Andrew Strominger in a lecture disputing a common phenomenological philosophical definition of ’emergence’. His question is: "What can be said about the emergence of physical properties such as light and sound from the surface of the ocean." This is a very difficult problem with many possible approaches and seemingly no formal primitives (axioms?). Strominger claims that physicalist theories and calculations are required no matter whether one is studying the philosophy, methodology or science of the problem. He opposes the phenomenological definition of emergent as something so complicated nothing empirical can be said. Strominger was probably using emergence from the ocean as an analogue of the physical problem - "what happens at the minimum Planck spacial dimension?" Also See the discussion of a similar problem described by Werner Heisenberg. in my extended abstract.

Strominger observed that completely different results (encyclopedia's in Neurath's language) will result from observation made in an airplane flying over the ocean versus those made by a diver swimming at the surface of the ocean. One important activity of science is to attempt to unify the two encyclopedias. I am using the Vienna Circle term encyclopedia instead of the Lakatosian term research programme in this talk. Many people would use the less precise term 'theory' here. The term encyclopedia is better than research programme when discussing computational issues because it entails all of the idea of data bases, facts, experimental results, conjectures and computations.

Notice in Stromminger's problem there is no one correct level of observational detail. Following Vienna Circle discovered empiricism, if the two encyclopedias disagree, both scientists and philosophers should try to devise experiments to test the encyclopedias. Study of the experimental results then might result in unification of two encyclopedias. Unification (called unity of science by Neurath) is very positive since it allows use of common methods. But good faith experimentation may lead to altering facts, splitting encyclopedias (hopefully temporarily) or changing computational thinking. When anomalies are discovered, there are four possible responses. Modify the airplane encyclopedia, modify the surface encyclopedia, replace either or both encyclopedias, or add new encyclopedias. There can be no concept of 'truth' here. Empiricism becomes good faith application of 'the cunning of human reason'. Also notice that science studies particular problems. Unifying different encyclopedias is a laudable goal where possible, but historically problem splitting (new encyclopedia creation) has been more important.

The advantage of Vienna Circle empiricism using the Neurath principle is that it avoids preconceptions and metaphysical (dogmatic) assumptions about the form of good encyclopedias. Chaos theorists learn nothing because they remains in their airplane and perceive just complexity. The logician only sees the simple molecules under the surface because molecules can be logically explained from atoms (axioms).

2. Vienna Circle Metaphysics Free Terminology

In my extended abstract, I used the analogous but more confrontational research programme language of Lakatos, Feyerabend and Kuhn (LFK) from the second half of the 20th century. In this talk I am using the Vienna Circle (VC) language from the first half of the 20th century. Both terminologies attempt to codify the methods developed in the latter part of the 19th Century by the founders of modern physics especially Ernst Mach and Max Planck. Without question modern physics has been the most successful scientific endeavor of all time. The crucial development is that science (called natural philosophy by Planck) studies nature without any metaphysical assumptions and see problems as complex, multifaceted-faceted, multiply interconnected, and without boundaries (Ballungen using Neurath’s language). On initial study any scientific undertaking seems unorganized and containing mysterious phenomena. See *Rediscovering the Forgotten Vienna Circle* edited by Thomas Uebel and *Otto Neurath Philosophical Papers 1913-1946* edited by Robert Cohen and Marie Neurath for a more detailed discussion of the Vienna Circle empirical method and its arguments against metaphysics. (probably better translated as Dogma). Planck, Einstein and Mach strongly influenced both the first 1910 and second 1920s Vienna Circle.

The most important principle of Vienna Circle philosophy is that both philosophy and science must be empirical. The empirical methods of physical science are applied (called physicalism by the Vienna Circle - methodology of science by LFK). In opposition to Popper who wanted to eliminate social sciences (particularly Freudian Psychoanalysis and Marxism) from scientific study, the Vienna Circle advocated unified science in the sense that the same method could be applied to any scientific or philosophical problem (i.e. human thought created computation here). In Vienna Circle language, theories were tested by shaking (as opposed to Popper's only allowing negative falsification). In the LFK language, shaking is called research programme testing where testing usually included the suggestion of crucial experiments. The Vienna Circle follows Mach in allowing statements involving observers such as "In the 13th century, monks observed angels".

The Vienna Circle philosophy is more useful for studying computation because in LFK (except for Feyerabend's early work), research programme degeneration is the only mechanism for explaining scientific change. In my opinion, the Vienna Circle use of the word encyclopedia for a scientific or philosophical theory is better for studying computer science than the LFK term 'research programme'. In the Neurath language, encyclopedia includes both experimental results and experimental (behavioral) methods.

3. The Philosophy of Computer Science

An important aspect of science in the 2nd half of the 20th century has been replacement of well understood encyclopedias (research programmes) with formalized metaphysical analogues. Well understood usually means containing few (recognized) anomalies. This post facto formalization makes it difficult to study new problems such as the nature of computation because it not clear what to study, what psychological factors to include, if biological factors are part of the problem, etc. This is exactly what Neurath calls Ballungen. Individual computer systems are Ballungen in the same sense.

Another non obvious problem is the relations between computation and other sciences. Many sciences are dependent on ’software codes’ for empirical predictions. Do changes to or replacements of computational encyclopedias require changes to the encyclopedias for the science that uses the computation? The Vienna Circle language unifies science to allow studying exactly this type of Ballungen. This situation requires Neurath’s description of unified science as needing to rebuild ships at sea instead of being able to design and build ships (scientific theories) from scratch (formalize).

4. Epistemology of Computer Program Verification

The very idea of verifying computer programs arises from redefinition of the meaning of verification. Lakatos in his 1973 lectures on scientific method writes "*Verificationism* originally came from Ludwig Wittgenstein and Moritz Schlick in the Vienna Circle in the early 1920s. According to Wittgenstein and Schlick, only those statement whose truth value is decidable by experiments are valid." (see *For and Against Method* by Feyerabend and Lakatos, p. 52). Lakatos observes that verification is impossible because it is impossible to perform an infinite number of experiments. Current formalist computer science went wrong exactly in the Vienna Circle sense by eliminating experiments - assuming computer program specifications did not require testing (shaking).

All currently published theories of program verification involve two mistakes. First, they attempt to reduce all computer programs to metaphysical principles such as mathematics or the inductive-deductive model. Second, they contain implicit assumption that the number of states of computers and of programs should be viewed as finite (unconscious finitism), because computer calculations are so fast. See the final section of this talk for a discussion of the serious damage of this fallacy to computational understanding and scientific progress. This rest of this section considers flaws in metaphysical theories involving concrete programs (systems). The final section of this talk consider much more serious problems caused by suppression of Vienna Circle thinking (application of the Neurath principle) to abstract computational Thought.

4.1 The Three Published Program Verification Methods

See my extended abstract for a detailed description of the three published methods. It is a simple exercise to disprove the three methods if one adopts two very weak assumptions. that have been assumed by scientists for at least 100 years. The assumptions are:

1. Rationalism requires empiricism. The assumption can also be expressed as metaphysics (dogma) defined as a priori axioms (beliefs) is worse than experimental evidence. This assumption allows applying the Neurath Principle.
2. Science (as computer program development here) is a human activity. A goal of science is to increase human welfare. This assumption is needed to guide scientific reaction to disconfirmatory empirical evidence. Without such an assumption, scientific progress could become suppression of criticism to eliminate empirical testing once a particular theory becomes dominant.

Amazingly, as my intended thesis work showed from the 1970s, program verification using mathematics was empirically incorrect from the beginning. Dijkstra's original book *A Discpline of Programming* contained mistakes in the first example used to illustrate the method. Dijkstra's difficulties are exactly what one should expect from metaphysics (dogma) based theories as opposed to experimental theories. In other words, Dijkstra's preconceptions prevented him from using problem specific knowledge.

Mostly because of the superior software from Bell Labs, the one person project Linux and open source software, program verification is now commonly ignored outside of academic EECS departments. There were two psychological reasons for popularity of metaphysics based program verification methods. First, it allowed academics with no problem specific knowledge to eliminate competition when scientists developed problem specific algorithms. Second, it promoted funding in the artificial intelligence area that had been dealt a crippling blow from the Lighthill Report written by a famous British physicist in 1972. As a recent photo of software development factory floor shows, the metaphysics based program verification methods are dehumanizing (violate assumption 2) because they replace programming as problem specific encyclopedia development that provides satisfying intellectual challenges by assembly line work no different from the cement factory scenes in Brecht plays.

Criticism of program verification is still important because the theory seems to have come back to life in this year's Turing Prize. The award was presented for verification by proof of computer hardware models. Since implementation by hardware versus software is an experimental question, it appears that the CS establishment is attempting to resurrect mathematical proof based CS in a new form. Model checking uses the metaphysical formalism of temporal logic that contains the same problems as formal program verification: finitism fallacy and immediate empirical disconfirmation. See the responses to the ACM announcement on the www.slashdot.com web site. The responders, who naturally prefer intellectually challenging work, describe a number of hardware circuit type specific new encyclopedias and discuss projects that failed because of formal model checking.

4.2 Computational Metaphysics Regrettably has Followed Vienna Circle History

A particularly unfortunate historical part of the success of metaphysics based computer science is the history of bad faith in preventing publication of criticism that continues a similar suppression of Vienna Circle logical empiricism (Neurath principle particularly) first in the 1930s and then again in the 1950s. I am following the sociological analysis of Vienna Circle suppression in the *Discovering the Forgotten Vienna Circle* book. and mis-interpretation documented particularly by Friedrich Stadler. I believe it is important to record this history to prevent the academic mis-conduct from happening again.

I was an undergraduate student at Stanford and then a graduate student in the UC Berkeley CS department during that period. Vienna Circle empirical methods were popular at Stanford and Berkeley until program verification by mathematical proof was used to suppress it (Lakatos calls this behavior scientific thought police). Empiricist mathematics that did not make the mistake of viewing computation as finite was taught at Stanford under the influence of George Polya. Even now one can see degeneration in computational thinking by asking engineering professors why they do not like teaching calculus using the Apostol Bolzano Weierstrass based text books but rather prefer modern derivative based textbooks. Popular Stanford German professor Peter Foulkes was translating many of the original Vienna Circle documents at the time.

At the UC Berkeley literature and science school based CS department, anti-formalism and empiricism thrived. Jay Earley had just developed his context free language parsing algorithm that is still the fastest algorithm. It did not use the finite thinking based algorithm design from algorithm efficiency proof method. James Morris published a paper that showed programming language types were not mathematical sets. Richard Karp and Stephen Cook (at Toronto then after being denied tenure in the UC Berkeley math department) had just proposed the P=NP problem. My recollection is that the proposal was intended as a criticism of the Tarski satisfiability based semantic definition of truth.

By 1980 P=NP had been turned into metaphysical truth. Publication of any other proposal for evaluating computer program efficiency was suppressed. Jay Earley and James Morris had been denied tenure at UC Berkeley and were forced out of the academic computer science system. The UC Berkeley EE department had annexed the Literature and Science based CS department (an eerie echo of Nazi annexation of Austria after that country had just suppressed Vienna Circle logical empiricism). CS graduate students from the original department were denied Phds. In my case, I attended Paul Feyerabend's philosophy of science seminars. Feyerabend helped me look for flaws in the obviously unscientific (in my view then) methodology of computer science based on mathematical proofs. Feyerabend's teaching and continuing development of the Neurath

Principle (it was not explicitly named because of the political suppression of Vienna Circle history) led to my discovery of the empirical failure of Dijkstra's program correct proof method. The idea to try to shake (I used the word disprove at the time) formalist computer science came from presentations and discussions in Feyerabend's seminar. Also, from Feyerabend's seminar I was able to recognize the empirical failure as a strong sign of Lakatos style research program degeneration. At that time, UC Berkeley philosophy professor and phenomenolist Hubert Dreyfus was attempting to have Feyerabend fired, and also a very unfair review of "Against Method" appeared in the BJPS. Although, Feyerabend was given a chance to reply. The reason empirical (using the Neurath principle) criticism of formalist CS has taken so long before my opportunity now to present it here is that my CACM submitted paper was not published. (I still believe it was not even reviewed), and UC Berkeley EECS professor Susan Graham made a point of sending letters whenever I applied for an academic job. At Stanford, Polya retired and George Forsythe died and Stanford CS was taken over by professors who believed in computer science as finitary mathematical proofs. Probably the most important event in elimination of empirical criticism of program verification by mathematical proof was the publication of Knuth's paper on "goto" statements. After that paper was published, it was decreed that no more papers on programming methods would be published.

5. Formalist Computer Science Suffers from the Fallacy of Finitism

The result of suppression of empiricist computer science is not only lack of progress in computational thinking but also interference with other sciences for which computational codes are needed for experimentation. I view the fallacy of finitism as the main current consequence of of lack of empirical studies in computer science. There are a number of alternative ways to see the current problem. One example is many arguments, especially in the P=NP area, assume unbounded but finite is the same as countably infinite. In fact, throughout the historical development of the concept of infinity, infinity has been an abstract conceptual quanity. The main use of this type of finiteness is to avoid the inconsistency results from Goedelian integers.

Some logicians have reacted to the finiteness fallacy by studying only the infinite elements in the recursively enumerable sets. One can consider a Turing machine chugging along, but because all sets are infinite no conclusions can be drawn because waiting an unbounded number of ticks is impossible. Whatever can be discovered (proved) must come from the algebra of infinity.

It has taken some time for the fallacy of finiteness to filter into concrete problems, but it is beginning to occur. Two examples are:

1. One area involves hardware model checking. Here the problem comes from approximating the number of states inside a computer circuit as finite, The problem is that the computer used to perform the model checking is at least one generation in hardware complexity (number of transistor that implies number of states and switch speed) behind the circuit whose model is formally being checked.

2. Another area where the finiteness approximation seems to be failing in concrete situations is financial modeling. Financial modeling is so wide spread and so

> much effort and competitiveness take place that the number of modeled states can not be approximated as finite. The consequences of this are failures in probability models and actual distributions that do not match calculated ones. A possible explanation is that probability axioms are problematic when applied to counting of infinite and empirical quantities. Formalist metaphysics is so ingrained that the failures are almost always attributed to mistakes (lack of risk control) by the modelers. One way to describe this problem is that the number of encyclopedias becomes large relative to the overall human population.

It is even possible that applying the Neurath principle to computational thinking may require a change to the mathematical nature of infinity. The problem background is that uncertain and problematic world of discontinuities and lack of bounded physical quantities in current physical theories especially string theories (encyclopedias) and theories of gravitation. The various problems fit the Neurath description of Ballungen.

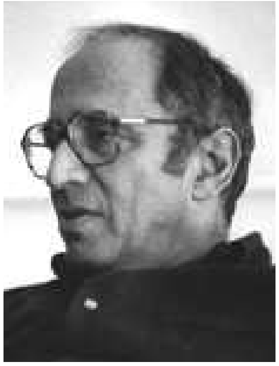
Modern physics began at the end of the 19th century with Planck's calculation of black body radiation. The calculation depended on the existence of a countably infinite number of states. One way to solve the P=NP problem is to alter the concept of infinities to add exactly one infinity possibly called aleph-one-half between aleph-zero (countably infinite) and aleph-one (number of reals). The number of non deterministic Turing machines then becomes aleph-one-half instead of aleph-zero. This change is not only consistent with the Neurath principle but only possible with it because of the ability to use anomalies to change basic properties of encyclopedias (or add new encyclopedias) as well as change facts, data or protocol sentences. It is not clear how the number of non deterministic Turing machines can be put in one-to-one correspondence with a different set, but following Mach it may be necessary to adopt the new definition of infinity for further scientific progress.

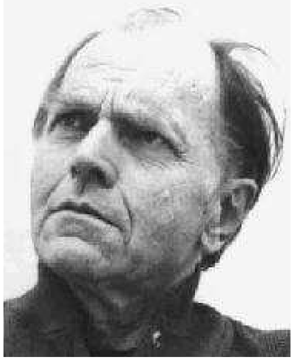
See Smolin's book *The Trouble with Physics* for the argument that string theory is failing empirical tests. In particular, Smolin believes that the current CERN effort to find the crucial Higgs Boson will fail. The predicted date of discovery during at the beginning of the 21st century has already passed. It is possible to read Smolin's criticism of physical theories over the last 30 years as an explication of problems with concept of infinity. It is not clear if adding an infinity smaller that the infinite number of reals would solve the problem of physical quantities with discontinuities and unboundedness, but it seems like it is an avenue that should be explored.

The Center for Austrian Studies presents

The Vienna Circle's Successors in Minnesota and America

The Lakatos~Feyerabend~Kuhn Program





Steven Meyer

Pragmatic C Software Corp.

Thursday, January 27, 3:30 p.m.

Ford Room, 710 Social Sciences Building (WB)

College of Liberal Arts

Vienna Circle (Neurath)	**Lakatos-Feyerabend-Kuhn**
Logical Empiricism	Defense of Rationality
Encyclopedia	Research Programme
Metaphysics	Irrationalism
Neurath Principle	Degenerating Research Programme
Protocol Sentence	<none>
Physicalism	MRSP
Encyclopedia Creation	Problem Splitting
Neurath Principle	Content Increasing Discovery
Shaking	Anomaly (Crucial Experiment?)
<none?>	Ad Hoc
<none?>	Protective Belt
<none?>	Hard Core