1

A Scientific Basis for Computational Science

Raúl E. Valdés-Pérez

May 1993

CMU-CS-93-162

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Computational science is a productive intellectual activity. It produces highly useful computer programs that require much creativity and ingenuity to develop. Moreover, computation is a powerful theoretical tool for natural scientists.

However, can computational science have a scientific foundation, quite apart from its roles as a juxtaposition of disciplines and as another theoretical tool for scientists? That is, can computational science develop concepts that enable a broad systematic understanding of inference and discovery in science?

This paper makes a case for an affirmative answer that relies on the concept of "generic scientific task." We will argue that theoretical understanding is to be attained by identifying and automating such tasks. To develop the idea, we configure samples of previous work in computational science (broadly construed), lay a road map to guide further research, and suggest experimental tools to generate research problems and to re-deploy proven techniques.

This work was supported partially by a grant from the W.M. Keck Foundation for advanced training in computational biology at the University of Pittsburgh, Carnegie Mellon University, and the Pittsburgh Supercomputing Center, and partially by the National Science Foundation through grant number DIR-8920118 to the Center for Light Microscope Imaging and Biotechnology.

1. Introduction

There can be little doubt that computational science is a productive intellectual activity. Its various instances - computational biology, computational chemistry, computational physics, and so on - produce highly useful computer programs that require much creativity and ingenuity to develop. Moreover, the idea of computation provides a powerful tool for natural scientists to use in their theoretical development.

However, can computational science have a scientific foundation, quite apart from its important role as a useful juxtaposition of disciplines and from its role as another theoretical tool for scientists? That is, can computational science develop concepts that enable a broad systematic understanding of reasoning and discovery in science?

This paper makes a case for an affirmative answer by proposing a scientific basis for computational science. The basis relies on the concept of *generic scientific task*, and suggests that theoretical understanding is to be pursued by identifying and automating such tasks. To develop the idea, we will place within a coherent framework various samples of previous work in computational science (broadly construed), lay a road map to guide further research in the area, and suggest experimental tools that can serve to generate research problems, or to re-deploy proven techniques.

First, we will proceed by recalling the role of scientific concepts and demonstrating that the usual interpretations and categories of computational science are inadequate to achieve a scientific basis.

2. Scientific Concepts

According to Hempel's *Fundamentals of Concept Formation in Empirical Science* [13], a concept is a complex of characteristics that determines membership in a class that one wishes to distinguish. Hempel points out the two roles of *scientific* concepts: description and theoretical systematization. For example, the concept of "color" can describe the appearance of peas, and of other things, but also enabled Mendel to formulate his genetic laws based on the inheritance patterns of pea color. We conclude that color is a valuable scientific concept, at least for classical genetics.

We can turn to computer science itself to further illustrate the nature and role of scientific concepts, although that field is much less concerned with description due to the generally abstract character of computer science. Some examples of very fruitful concepts are the notion of a Turing machine, of a stored program, of heuristics, and of resource consumption that grows polynomially in the size of input.

However, not all concepts are scientific concepts, since not all concepts contribute much to theoretical systematization. For example, in computer science there is no known reason to regard as fruitful the concept of resource consumption that grows as n^4 in the input size. Consider another example from nature: is the concept "bigger than a limousine" a fruitful scientific concept? This concept includes volcanos, planets, and blue whales, and certainly serves to describe nature. It even permits a degree of theoretical systematization, since the observation that no living creatures bigger than a limousine can fly might lead to an explanation: the muscular strength for flight grows in proportion to cross-sectional area, but mass grows with volume [31]. Hence, in larger creatures the extra muscle that powers flight is quickly overwhelmed by the added weight. After a few such theoretical successes however, one can expect to run out of things to say, since there appears to be little systematic understanding to be reached by comparing volcanos with blue whales.

3. Natural Science

Before proceeding to examine computational science, it is worthwhile to consider what is the basis for cohesiveness in the natural sciences. The cohesiveness of volcanology, for example, is ensured by the common concern of volcanologists with volcanos. Similarly, the cohesion of biologists is a consequence of their common object of inquiry: living organisms. The natural sciences coalesce around certain phenomena, whether life, volcanos, the planets in geophysics, and so on. A cell biologist who builds models of cellular processes has much in common with his next-door colleague who invents new ways to observe experimentally those same processes, even though the reasoning and methods used by the two biologists are quite different. Even computer science, which is not generally regarded as a natural science, coheres around the abstract concept and concrete reality of computers.

Of course, there arise important problems that call on the expertise of multiple disciplines. However, this truth does not undermine our point, which is that the cohesion of a specific natural science rests on a shared concern with similar phenomena.

4. The Various Computational Sciences

Computational biology, chemistry, and physics are by now healthy branches of science. For example, each has its own journal named *Journal of Computational Biology (Chemistry, Physics, etc.)*. Besides serving as a label for highly significant and useful juxtapositions of expertise, will these individual computational sciences acquire a coherence that will lead to the systematic understanding of which Hempel writes? Putting it another way, does a *Computational X* inherit the coherence of the parent science *X*? Formulating the question in yet a third way, does the conventional view of the computational sciences depicted in Table 1 represent a theoretically fruitful view?

We think that the answer is negative: the concept of computational biology, for example, does not by itself further a systematic understanding of computation in biological science. The juxtaposition of 'computational' and 'biology' can usefully describe activities that deserve the label, but it does not inherit the coherence of biology. Among cell biologists, the model builder and the developer of experimental techniques have a mutual concern with cells. However, computing for model-building and computing for experimental-technique

Computational Biology
Computational Chemistry
Computational Physics
Computational Metallurgy
Computational Psychology
Computational Astronomy
Computational Geology
Computational Paleontology
Computational Petrology
Computational Zoology
Computational Oceanography
Computational Seismology
Computational Mineralogy
Computational Volcanology
Computational Physiology
Computational Mcteorology ⋮

Table 1: A Conventional View of Computational Sciences

development potentially share nothing except the instrument of computing. And Computational Science does not pursue a systematic understanding of computing as a phenomenon free of context, but as a tool that relates to scientific reasoning and discovery.

Our conclusion is that computational science, viewed as a disparate collection of instances (computational biology, physics, geology, etc.), is not a fruitful scientific notion. It is a valuable description of challenging intellectual activity, but is likely to lead to systematic understanding only in the individual parent science through its role as another theoretical tool. We will argue in what follows for an alternative view of computational science that does lead to theoretical understanding of computation in science.

5. A Scientific Basis

Our proposal is that computational science should be organized around the concept of generic tasks that cut across the various sciences. Rather than slicing up the computational sciences horizontally into computational volcanology, physics, etc. as in Table 1, we propose many vertical slices as in Table 2. The next few sections will develop our meaning by examining in some detail various generic tasks from science. Each generic task involves a single, specific type of scientific reasoning that occurs in more than one science. However, we make no claim of crisp boundaries between these tasks; even the possibility that one task wholly contains another is not ruled out. As with any taxonomy, knowledge of the best descriptive categories, and of relations among them, emerge over time.

	Task 1	Task 2	Task 3	Task 4 ...
Computational Biology				
Computational Chemistry				
Computational Physics				
Computational Metallurgy				
Computational Psychology				
Computational Astronomy				
Computational Geology				
Computational Paleontology				
Computational Petrology				
Computational Zoology				
Computational Oceanography				
Computational Seismology				
Computational Mineralogy				
Computational Volcanology				
Computational Physiology				
Computational Meteorology				
⋮				

Table 2: A Scientific Basis for Computational Science

5.1. Computer algebra

A very familiar, generic task in science is solving for a differential equation. An equation of the same form is solved identically, whether it arises in population biology, mathematical psychology, or elsewhere. Standard symbolic computation systems can solve the differential equation $y' = f(x)$ over restricted classes of elementary functions. This is, of course, also known as indefinite integration: $y = \int f(x)dx$.

Solving indefinite integrals is not an isolated case: there also exist algorithms for solving higher-order differential equations, finding roots of polynomials symbolically, and factoring multivariate polynomials over various coefficient domains. Kaltofen [16] and Davenport et al. [8] provides overviews of the field of computer algebra.

5.2. Optimization

A second example of a generic task in science is optimization. In the case of combinatorial optimization, there already exists the algorithmic theory and implemented software to solve linear programs [27], nonlinear programs, mixed-integer linear programs, and even mixed-integer nonlinear programs. The scope of combinatorial optimization is not limited to engineering-oriented or experiment-planning aspects of science; theoretical model-building itself can at least sometimes be formulated as problems of combinatorial optimization by minimizing the size of the model (e.g., [35, 36]), which in some sense corresponds to seeking the simplest theory. A wholly separate category is continuous optimization; similarly, such tasks are generic in the sense that they cut across sciences. A typical application of continuous optimization arises when theories or models contain free parameters to be optimized

$\{S_{1,l}\}$	→	$\{S_{1,r}\}$
$\{S_{2,l}\}$	→	$\{S_{2,r}\}$
$\{S_{3,l}\}$	→	$\{S_{3,r}\}$
	$\vdots$	
$\{S_{n,l}\}$	→	$\{S_{n,r}\}$

Table 3: Pathways

on the basis of experimental data [4].

The above two task examples are perhaps easily seen to be generic, since they are closely tied to mathematics and to algorithmic theory, both of which are abstract disciplines that emphasize generic representations and solutions. If our concept of generic scientific task were limited to such examples, then it would not be very systematic, since most scientific reasoning does not involve such crisp formulations.

5.3. Inferring laws from data

Both artificial intelligence and statistics have addressed the problem of finding algebraic laws hidden in data, a task of evident generality and of both historical and current relevance. Statistics, for example, has developed methods of regression that lead to algebraic relations among measured variables [23]. The original BACON program from cognitive science emphasized the creation of new terms from data using a heuristic search that noticed correlations between two terms; psychological plausibility was also a motivating factor in the design of the program. Much work in AI subsequent to BACON has attempted improvements.

The first three cases above of generic tasks derive their generality from the following fact: their starting points consist of context-free equations, other mathematical expressions, or numeric data. One may begin to suspect that only such mathematical or numerical problems can be generic. To dispel this suspicion, we will next consider in some detail a task drawn from our own work that involves substantial reasoning specific to a particular science, but which nevertheless is of a promising degree of generality.

5.4. Pathway elucidation

There are many problems across science whose solutions take the form of a pathway (or, a set of steps), as illustrated in Table 3. $S_{i,l}$ and $S_{i,r}$ refer to a set of items on the left and right (respectively) of the ith step. Typically, a pathway describes some physical process that consists of multiple steps, and a typical means to elucidate the multi-step process is by collecting experimental evidence on its various characteristics, as permitted by available instrumentation.

For example, an experimental biochemist's study of a multi-step reaction may lead to the conclusion that the underlying reaction pathway looks like this:

$$\begin{aligned} \mathit{ornithine} + \mathit{ammonia} + \mathit{carbon\ dioxide} &\rightarrow \mathit{water} + \mathit{citrulline} \\ \mathit{ammonia} + \mathit{citrulline} &\rightarrow \mathit{arginine} + \mathit{water} \\ \mathit{arginine} + \mathit{water} &\rightarrow \mathit{urea} + \mathit{ornithine} \end{aligned}$$

Hans Krebs proposed this very pathway in 1932 to describe the formation of urea *in vivo.*

Not all chemical pathways are this simple. Many common cases involve a dozen or so steps, and some important reactions are suspected to consist of more than a hundred steps. We have made a significant advance in the automation of this chemical problem, to the point where a computer program named MECHEM can propose highly plausible pathways of moderate size on the basis of experimental evidence [37, 34, 33].

Chemistry is not the only source of pathway-elucidation problems. Determining multi-step processes on the basis of experimental data is a problem that arises in nuclear physics [14], petrology [15], and most especially biology. A biological problem of great current importance is protein folding; the experimentalist's role is to elucidate the pathways that are supposed to describe how a protein chain folds into its natural state from a denatured state [21, 28]. Without further, massive citing of biological literature, we point out that numerous problems in biology involve the attempt to reconstruct some sort of pathway on the basis of experimental evidence. One of our projects addresses the automation of one such problem: endocytic processing in cells [25, 38].

To further support the idea that there is a rich set of generic tasks that cuts across the sciences, we can adduce the surprising link of pathway elucidation with the science of psychology, specifically cognitive psychology. Anderon et al. [2] discuss an interactive tutoring system that has been designed to test Anderson's ACT* (now PUP) theory of cognition, which is based on the formalism of production systems. A crucial part of the tutor involves *model tracing*, which is the task of inferring the production system in the student's head from his interaction with the tutor. Now, a production system is formally an instance of the scheme in Table 3, and the tutor/student interaction is the experimental data analogous to what is learned from studying a chemical reaction in the laboratory. There is not yet any hard evidence that these similarities will be exploited at the algorithmic level, but elsewhere we have explored in some detail these and other links among "pathway elucidation" problems in science [38].

The cohesive basis for pathway elucidation is that, typically, intermediate products of the "reaction" are observed experimentally, all of which must be linked via a pathway. Usually there is a rich hierarchy of constraint on pathways, ranging from what are plausible participants, plausible "reactants" and "products" of a step, plausible steps, and plausible combinations of steps. Sometimes pathway-like models are called other names: paths, assemblies, nucleations, channels, mechanisms, etc.; the main issue is whether they consist of discrete steps.

The reader may have noticed the formal analogy between the pathways of Table 3 and a proof in logic, in which '→' is logical implication. Indeed, there is no restricted meaning to the symbol '→' as used above, which allows for its adaptation here both to the '→' of cognitive psychology and the '→' of chemistry; a logical proof certainly fits the pathway schema also. However, this noteworthy similarity of pathway to proof neither strengthens nor weakens the assertion that pathway elucidation is a cohesive class of scientific task.

5.5. Finding differential-equation models

Another generic task in science concerns modelling a physical phenomenon with differential equations (or difference equations). The starting point for this task is knowledge of the object phenomenon and/or experimental data, plus a blank sheet of paper. The result is a set of differential equations, written on the paper, that model the dynamics. In many cases, no publishable results ensue, since deriving the model is straightforward; for example, the basic law of chemical kinetics indicates how to write a set of nonlinear differential equations from a given reaction pathway of elementary steps [18]. In other, less well-understood cases, the model does lead to a publishable result, e.g., [17].

It may be objected that the task of differential-equations modelling does not lend itself to generic computer implementation, even if the specific scientific knowledge is separated from the reasoning. It may turn out that the only commonality obtainable is at a conceptual level, rather than an algorithmic level or a program level. Should such cases still qualify as generic tasks in computational science?

It turns out that there already exists work on inferring ordinary-differential-equation models from time series data [7, 9]. In any case, however, the key issue is systematic understanding: in the ideal case, systematicity will reach the level of a single computer program that can execute the task. In other more difficult cases, the only reachable systematicity is at the level of guidelines or heuristics. The latter cases should be included in the general picture being painted here. The goals are systematic understanding and computer automation, but small steps toward these goals are completely valid.

5.6. Discrete model building

A significant number of tasks in natural science consist of proposing discrete, structural models of phenomena. The earliest significant work that involved computer scientists was the DENDRAL project [22], which started with the discovery by J. Lederberg at Stanford of an algorithm to generate the possible isomeric structures of a given molecular formula [20]. That algorithm led to a sustained effort to automate the elucidation of molecular structure on the basis of mass-spectrometric and other data. Part of the motivation for DENDRAL was to investigate (or demonstrate) the crucial role of abundant, specific knowledge in task performance, contrary to the prevailing *zeitgeist* that emphasized general reasoning methods and downplayed knowledge of the domain.

The subsequent work at Stanford could have taken several fruitful directions; the chosen direction led to the MYCIN system and the expert-systems movement. An alternative direction could have been to examine various tasks of structure discovery or model building across several sciences to see whether a generic category could be identified. We believe that such a direction is still relevant and important, and it is perhaps clearer now where it would lead and how one might proceed.

We have taken some initial steps to identify a generic task of building discrete models in science [39], by analyzing six computer programs that modelled or automated different aspects of scientific reasoning in biology, chemistry, and physics. The common framework extracted from these systems involves search in spaces of matrices (in which simpler models correspond to smaller matrices) constrained by domain laws such as conservation and analogous constraints. Unfortunately, DENDRAL was not included in this framework, but a direction for further research has been laid.

5.7. Detection of structure in high-dimensional imagery

Another broad category of scientific task is the discovery of interesting structure in images. This task is made especially current - and laborious - by modern instruments that are capable of collecting large amounts of data in several dimensions, e.g., 3-dimensional NMR data of an animal's brain. Scanning the periodicals databases (exactly how is discussed below in section 8), one comes across many examples of structure detection that involve simple structures apparent to the untrained eye.[1]

We will mention just three examples. One author [12] reports in the journal *Icarus* the detection of a hexagonal feature in map-projected images of Saturn's north pole. Another paper in *Geology* reports a large circular structure beneath Lake Huron found in aeromagnetic images [11]. In *Nature*, other authors reported unexpected heterogeneity in a stearic-acid monolayer observed with an electron microscope: many densely scattered small holes were observed [32]. Many more instances of structure discovery can be found by scanning scientific journals.

We emphasize that the case of structure discovery addressed here is not necessarily one of contradiction with prevailing theory. Many important papers report structure where no theory applied, for example, there was no theory that predicted an absence of hexagonal features on Saturn's north pole.

Viewing structure discovery as a generic task, one can proceed to consider how to provide a computer aid to the scientist that will autonomously select images having good chances of containing interesting structure. In our own work, we are currently studying the feasibility of such an aid.

[1] Witkin and Tannenbaum [41] mention experiments done by Rosenfeld in 1964 that showed that secretaries were about as competent as meteorologists at outlining clouds in weather photographs.

5.8. Uncovering causal relations

Inferring causal relations among variables from data is a generic task in medical science and in social sciences such as psychology and economics as well. Spirtes, Glymour, and Scheines [30] have succeeded in developing efficient algorithms to automatically generate causal models from sample data, and have implemented the algorithms in the TETRAD II system. Spirtes et al. describe applications of the methodology to problems from medicine and social science.

5.9. Analysis of differential equations

Section 5.5 discussed the task of developing from scratch a model of a phenomenon or of data in terms of differential equations. After developing a model, a typical scientific task is the analysis of the differential equations to infer their qualitative behavior [29]. Projects to automate several such analyses by a group at MIT was overviewed by Abelson et al. [1].

5.10. Generating research problems

It is interesting to consider the possible generic nature of one of the more unstructured tasks from science: formulating research problems. There is healthy skepticism that anything like a systematic understanding of problem formulation is possible. In the face of this skepticism, there is not yet enough hard evidence that it is fruitful to view problem formulation as a generic task. However, there has been work in the area, which we will now interpret.

S. Fajtlowicz has developed a computer program named GRAFFITI [10] that generates mathematical conjectures in graph theory. Some conjectures made by the program have been picked up by research mathematicians, and the ensuing proofs have been published as novel theorems.[2] Fajtlowicz's program mainly addresses conjectures in graph theory, although he has also addressed other branches of mathematics. Although it is very unclear whether GRAFFITI's methods could lead to useful problem generation in other disciplines, this example illustrates one extreme along the performance/generality map, and is important evidence that systematic approaches to problem generation are practical and possible, at least in specific domains having a rather formal character.

From a very different source, J.E. Oliver has written at length (and excellently) on scientific discovery (especially in geophysics), including a considerable amount on the formulation of research problems [26]. Oliver's interest is in providing informal guidelines rather than systematic - much less computational - approaches. However, it is significant that Oliver believes that even the apices of scientific activity are susceptible to improvement through instruction through books such as his. The clear implication is that at least some systematic

[2] For example, a theorem that the independence number of a connected graph is $\geq$ the average distance between vertices [6] was first conjectured by GRAFFITI.

elements are present, even if his guidelines are not now usable by any existing computer program.

Finally, the book by Langley, Simon, Bradshaw, and Zytkow [19] has addressed problem generation from a middle point: less concerned with performance than Fajtlowicz, but more concerned with computer implementation than Oliver. The latter parts of their book give several specific heuristics for formulating research problems that could be incorporated into a computer program.

One may hypothesize that formulating research problems is a generic task in science that is, like the other tasks of this paper, susceptible to a good degree of systematic understanding, even automation in some cases. Unlike the previous cases, we are not aware of any frontal assaults on the task that aspire to a significant degree of generality.

6. New View of Computational Science

The previous sections have explicated the concept of generic tasks in considerable detail. The view advanced here, which is intended to found a computational science on a scientific basis, contrasts with the orthodox view of the computational sciences depicted previously in Table 1. Table 4 exemplifies the new view, which is an elaborated (and transposed) version of Table 2.

The new view calls for developing a descriptive taxonomy of scientific activity. The taxonomic concepts, which serve to place members in groups, derive their worth by how well they identify computationally similar tasks.

7. Other views

The most common, concise interpretation of computational science is that it represents science done on a computer, presumably in contra-distinction to science done by observing, experimenting, or theorizing. There currently exists no *Journal of Computational Science* for the same reason that there is no current *Journal of Observational Science* - the category is too broad. On the other hand, the field of *Scientific Computing* does have its own *Journal*, and the work reported there is typically of a somewhat generic nature (i.e., across sciences), although largely concerned with numeric computation. Roughly, the current difference between computational science and scientific computing is that domain scientists largely occupy the former, while a greater fraction of computer scientists are represented in the latter.

For our purposes, we do not see a useful distinction between the two terms, since we are concerned with science done both *on* and *by* computer, in the service both of the client sciences (which can make use of the automation products) and of a nascent science whose object of inquiry is the computational logic of science. For us, computational science is distinguished from human science, not experimental, observational, nor theoretical science.

	Computational Biology	Computational Chemistry	Computational Physics	Computational Metallurgy	......
computer algebra					
optimization					
inferring laws from data					
pathway elucidation					
finding differential equation models					
discrete model building					
detection of structure in imagery					
uncovering causal relations					
analysis of differential equations					
generating research problems ⋮					

Table 4: A Scientific Basis for Computational Science

Of course, collaboration between human and computational science is encouraged.

7.1. Computational science as a third mode of science

In an excellent article describing grand challenges in the application of supercomputers to science [40], K.G. Wilson interprets computational science as a third mode of scientific activity that is distinct from the classical modes of theory and experiment. Moreover, computational science actually competes with experiment, in the sense that modelling a physical phenomenon and simulating the model can produce data that are not practically obtainable via experimentation.

Wilson asserts that computational science is appropriately a new discipline, since long experience and professional training are required for success, at least at the supercomputer level. Wilson adds that computational science is not computer science, because "computational science is concerned with a specific set of computer applications and requires training in the discipline of the application being considered, whereas computer science addresses generic intellectual challenges of the computer itself." Wilson addresses further the important issue of what is quality research in computational science. Contributions of quality include new algorithms that are numerically accurate or that reduce the computations by large factors. Other contrib''tions involve model development and testing, mathematical insights into numerical accuracy, and the engineering of complex scientific software.

We do not see sharp conflict between our proposal and Wilson's views. On the contrary, our proposal is a generalization of them, but supplemented with an outline of how systematic understanding can be attained within the much broader scope that we advocate for computational science. We diverge, however, at the contention that computational science is not computer science, which is said to address "generic intellectual challenges of the computer itself." There is little interesting to say, *empirically*, about a computer by itself; perhaps one could determine how its durability and reliability over time are affected by temperature and the like, but further empirical knowledge depends on specifying something to compute, and that something is generic only to a degree.

For example, the classical subfield of operating systems arose from systematic study of how best to satisfy the dynamic resource demands of programmers and applications programs. However, even such a basic topic as multi-user operating systems is not completely generic, since some small, ultramodern computers permit only a single user. Things are generic to one degree or other: many scientific tasks have degrees of genericity, and these should be sought out by computer science, as well as by any other interested disciplines.

7.2. Generic tasks in expert systems

Chandrasekaran has previously developed the concept of generic tasks in knowledge-based expert systems [5], and J. McDermott has expressed similar views [24]. Chandrasekaran views generic tasks as elementary building blocks of more complex, but still rather generic, tasks such as heuristic classification. His examples of elementary generic tasks include hierarchical classification, hypothesis matching or assessment, abductive assembly, and others. Our concept of generic scientific task is quite related to Chandrasekaran's, although our concern is wholly with science and not at all with general intelligence.

There are a couple of reasons to think that the concept of generic scientific task may prove more systematic and fruitful than its predecessor. Firstly, empirical science as a whole shares many values and procedures as the results of instruction and indoctrination in scientific methods. Secondly, science strives to formulate its knowledge formally and precisely when possible, which is a boon to those who would automate it. For these reasons, and contrary to prevailing opinion, much of science may turn out easier to automate than the tasks of non-science. The modern methods developed for knowledge-acquisition, which depend on having a strong model of the intended problem-solving, may prove adaptable to scientific problem-solving in the future.

7.3. Science and artificial intelligence

Danny Bobrow quoted the late Allen Newell as making the following prediction of a convergence between the sciences and artificial intelligence [3]:

> We should, by the way, be prepared for some radical, and perhaps surprising, transformations of the disciplinary structure of science (technology included) as information processing pervades it. In particular, as we become more aware of the detailed information processes that go on in doing science, the sciences will find themselves increasingly taking a metaposition, in which doing science (observing, experimenting, theorizing, testing, archiving, ...) will involve understanding these information processes, and building systems that do the object-level science. Then the boundaries between the enterprise of science as a whole (the acquisition and organization of the knowledge of the wo:ld) and AI (the understanding of how knowledge is acquired and organized) will become increasingly fuzzy.

We regard the proposal of this paper as entirely consonant with Newell's perception, although we have added the concept of generic tasks that cut across the sciences as in Table 4. Newell's vision of building computational systems tu do "object-level" science corresponds to our emphasis on science automation; our view of computational science as task-centered and discipline-generic calls at least for adjustments to the disciplinary structure of science, if not for Newell's *transformations.*

7.4. Is systematic science likely?

To those who doubt that scientific reasoning and discovery are themselves susceptible to scientific analysis, we offer the following analogies. The number of atoms in the universe is very large, but chemistry teaches that the number of their types - or elements - is only 103. Similarly, the number of rocks in the world is very large, but petrologists point out that their distinct types are much fewer. Finally, the number of published scientific papers is large; for example, the number of records added weekly to the Science Citation Index is about 14,000 (728,000 per year). Is it really plausible that there are 728,000 types of publishable results in science, and that these are renewed yearly? Or, can one impose order on the seemingly endless variety, just as science so successfully achieves elsewhere? We think the answer is quite plain: scientific reasoning and discovery are ripe for systematic understanding, and their automation through computation is one bright direction.

8. A Research Plan

A fruitful research plan follows quite simply from the proposed view of computational science. Given the current stage of understanding, any of the fol!owing tractable steps will advance the subject:

- Identify a new task of some generality.
- Automate a new instance of an already-identified generic task.
- Compare individual automations of a generic task, and identify their common core.
- Automate a novel task from a single science, postponing considerations of generality.
- Identify common cores among multiple generic tasks.

One could generate more research steps by considering schematically (generically) what is being proposed: a taxonomy of science founded on computation. Taxonomies require development of concepts, demonstration of their worth, hierarchical organization, etc. Perhaps even the scientific task (carried out here in this section by the author) of proposing a research plan on the basis of a taxonomic view could be studied systematically.

What tools can help us thus to develop computational science? For some time we have been browsing databases of scientific and engineering abstracts (e.g., INSPEC, PERIODICALS, MEDLINE, etc.), first motivated by the desire to find problems of pathway elucidation analogous to the chemical problem on which we had worked for some years. Slowly, we came to realize the great value of these databases as a tool for understanding the content of science in a complementary way to how historians of science present it. Finally, in our preliminary study of computer-aided discovery of hidden structure in imagery, we have quite consciously searched these databases to gain an understanding of the breadth and importance of the task, as well as the typical discoveries that are made along those lines.

A convenient means to generate new research problems in this new-style computational science is to browse these databases using keywords that characterize scientific results or activities that one is familiar with. For example, one might search for all records containing the two adjacencies 'differential equations' and 'experimental data,' and then look up a good number of the articles to see what activities are involved, the logic and heuristics underlying the activities, and what are the publishable contributions. Then, one can set out to accomplish the task (or a subtask) with computation, or at least to design a computer aid as an important - and useful - first step.

9. Conclusion

We have proposed a new view of computational science that is founded on the concept of generic tasks. Each generic task is general because it is present in various sciences, but at the same time is specific because it involves one mode of reasoning, or a small number of them.

This view of computational science houses disparate pieces of work from computer algebra, computational statistics, operations research, theoretical computer science, and artificial intelligence under a single roof whose common purpose is the automation of science. Computational science is fundamentally an empirical science, in the sense that the large problems it poses are drawn from scientific practice, and to a large degree its solutions should be measured in terms of practice. For example, the problem of finding automatically the solution to a differential equation can be motivated by the empirical fact that such problems arise in science.

A research plan follows from the key features of computational science: its empirical motivation, its concern with automation via computation, and the need for a systematic taxonomy. Like other empirical science, observational tools are needed, and we have pointed one out: browsing electronic databases of abstracts. The experimental tools remain the same as in computer science as a whole: implementation and testing.

References

[1] Abelson, H., Eisenberg, M., Halfant, M., Katzenelson, J., Sacks, E., Sussman, G., Wisdom, J., and Yip, K. Intelligence in scientific computing. *Communications of the ACM 32*, 5 (May 1989), 546–562.

[2] Anderson, J., Boyle, C., Corbett, A., and Lewis, M. Cognitive modeling and intelligent tutoring. *Artificial Intelligence 42*, 1 (February 1990), 7–49.

[3] Bobrow, D. G., and Hayes, P. J. Artificial intelligence - where are we? *Artificial Intelligence 25*, 3 (1985), 375–415.

[4] Box, G., and Hunter, W. The experimental study of physical mechanisms. *Technometrics 7*, 1 (1965), 23–42. (Reprinted in: The Collected Works of George E.P. Box).

[5] Chandrasekaran, B. Generic tasks in knowledge-based reasoning: high-level building blocks for expert system design. *IEEE Expert 1* (1986), 23–30.

[6] Chung, F. The average distance and the independence number. *Journal of Graph Theory 12*, 2 (1988), 229–235.

[7] Cremers, J., and Hubler, A. Construction of differential equations from experimental data. *Zeitschrift fur Naturforschung A 42*, 8 (August 1987), 797–802.

[8] Davenport, J. H., Siret, Y., and Tournier, E. *Computer Algebra — Systems and Algorithms for Algebraic Computation.* Academic Press, London, 1988.

[9] Eisenhammer, T., Hubler, A., Packard, N., and Kelso, J. Modeling experimental time series with ordinary differential equations. *Biological Cybernetics 65*, 2 (1991), 107–112.

[10] Fajtlowicz, S. On conjectures of Graffiti. *Discrete Mathematics 72* (1988), 113–118.

[11] Forsyth, D., Pilkington, M., Grieve, R., and Abbinett, D. Major circular structure beneath southern Lake Huron defined from potential field data. *Geology 18*, 8 (1990), 773–777.

[12] Godfrey, D. A hexagonal feature around Saturn's north pole. *Icarus 76*, 2 (November 1988), 335–356.

[13] Hempel, C. G. *Fundamentals of Concept Formation in Empirical Science.* University of Chicago Press, 1972.

[14] Hodgson, P. Multistep processes in nuclear reactions. *Contemporary Physics 29*, 5 (Sept-Oct 1988), 457–476.

[15] James B. Thompson, J., Laird, J., and Thompson, A. B. Reactions in amphibolite, greenschist and blueschist. *Journal of Petrology 23* (1982), 1–27.

[16] KALTOFEN, E. Computer algebra algorithms. *Annual Review of Computer Science 2* (1987), 91–118.

[17] KIMMEL, M., AND ARINO, O. A system of differential equations modeling the G_1 phase of the cell cycle. *Computers & Mathematics with Applications 18*, 10-11 (1989), 907–917.

[18] LAIDLER, K. J. *Chemical Kinetics.* Harper & Row, 1987.

[19] LANGLEY, P., SIMON, H., BRADSHAW, G., AND ZYTKOW, J. *Scientific Discovery: Computational Explorations of the Creative Processes.* MIT Press, Cambridge, Mass., 1987.

[20] LEDERBERG, J. Topological mapping of organic molecules. *Proceedings of the National Academy of Sciences 53*, 1 (1965), 134–139.

[21] LEVINTHAL, C. Are there pathways for protein folding? *J. Chim. Phys. 65* (1968), 44–45.

[22] LINDSAY, R., BUCHANAN, B., FEIGENBAUM, E., AND LEDERBERG, J. *Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project.* McGraw Hill, New York, 1980.

[23] MANDEL, J. *The Statistical Analysis of Experimental Data.* Dover, New York, 1984.

[24] MCDERMOTT, J. Preliminary steps toward a taxonomy of problem-solving methods. In *Automating Knowledge Acquisition for Expert Systems*, S. Marcus, Ed. Kluwer Academic Publishers, Boston, MA, 1988.

[25] MURPHY, R. F. Processing of endocytosed material. In *Advances in Cell Biology*, vol. 2. JAI Press, 1988, pp. 159–180.

[26] OLIVER, J. E. *The incomplete guide to the art of discovery.* Columbia University Press, New York, 1991.

[27] PAPADIMITRIOU, C., AND STEIGLITZ, K. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

[28] RICHARDS, F. M. The protein folding problem. *Scientific American 264*, 1 (January 1991), 54–63.

[29] SACKS, E. Automatic analysis of one-parameter planar ordinary differential equations by intelligent numeric simulation. *Artificial Intelligence 48*, 1 (1991).

[30] SPIRTES, P., GLYMOUR, C., AND SCHEINES, R. *Causation, Prediction, and Search.* Springer Verlag, New York, 1993. Lecture Notes in Statistics.

[31] TRIBUTSCH, H. *How Life Learned to Live: Adaption in Nature.* MIT Press, Cambridge, MA, 1982.

[32] UYEDA, N., TAKENAKA, T., AOYAMA, K., MATSUMOTO, M., AND FUJIYOSHI, Y. Holes in a stearic acid monolayer observed by dark-field electron microscopy. *Nature 327* (1987), 319–321.

[33] VALDES-PEREZ, R. E. Computer-aided elucidation of reaction pathways: application to catalyzed hydrogenolysis of ethane. *Industrial & Engineering Chemistry Research.* Submitted for publication.

[34] VALDES-PEREZ, R. E. Conjecturing hidden entities via simplicity and conservation laws: machine discovery in chemistry. *Artificial Intelligence.* In press.

[35] VALDES-PEREZ, R. E. Direct induction of phenomenological selection rules. Submitted to *Journal of Computational Physics.*

[36] VALDES-PEREZ, R. E. Discovery of conserved properties in particle physics: a comparison of two models. *Machine Learning.* Accepted for publication.

[37] VALDES-PEREZ, R. E. Algorithm to generate reaction pathways for computer-assisted elucidation. *Journal of Computational Chemistry 13*, 9 (November 1992), 1079–1088.

[38] VALDES-PEREZ, R. E., SIMON, H. A., AND MURPHY, R. F. Discovery of pathways in science. In *Proceedings of the ICML-92 Workshop on Machine Discovery* (1992), pp. 51–57.

[39] VALDES-PEREZ, R. E., ZYTKOW, J. M., AND SIMON, H. A. Scientific model-building as search in matrix spaces. In *Proceedings of Eleventh National Conference on Artificial Intelligence* (1993). To appear.

[40] WILSON, K. G. Grand challenges to computational science. *Future Generation Computer Systems 5*, 2-3 (September 1989), 171–189.

[41] WITKIN, A. P., AND TENENBAUM, J. M. On the role of structure in vision. In *Human and Machine Vision.* Academic Press, New York, 1983, pp. 481–543.

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890