Introduction to Computational Scientific Discovery

Pat Langley

Institute for the Study of
Learning and Expertise

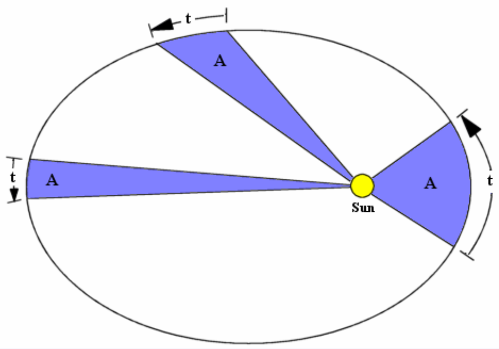
Center for Design Research
Stanford University

European Summer School in Artificial Intelligence
University of Ljubljana, Slovenia
July 24–28, 2023

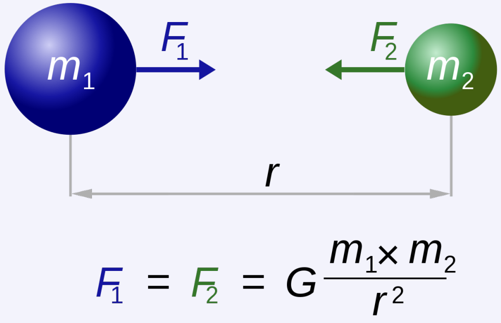
Opening Remarks

Examples of Scientific Discoveries

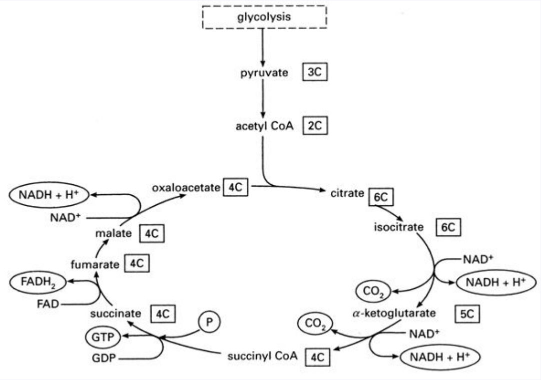
Science is a distinguished by its reliance on formal laws, models, and theories of observed phenomena.



Kepler's laws of planetary motion

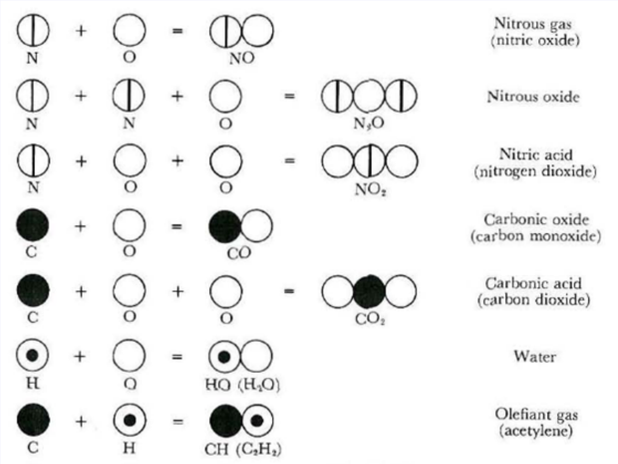


Newton's theory of gravitation



Krebs' citric acid cycle

We often refer to the process of finding such accounts as *scientific discovery*.



Dalton's atomic theory

Mystical Views of Discovery

Many philosophers of science had avoided discovery, believing it immune to logical analysis. E.g., Popper (1934) wrote:

> *The initial stage, the act of conceiving or inventing a theory, seems to me neither to call for logical analysis nor to be susceptible of it … My view may be expressed by saying that every discovery contains an 'irrational element', or 'a creative intuition' …*

Hempel and many others also believed discovery was inherently irrational and beyond understanding.

However, advances made by two fields – *cognitive psychology* and *artificial intelligence* – in the 1950s suggested otherwise.

Scientific Discovery as Problem Solving

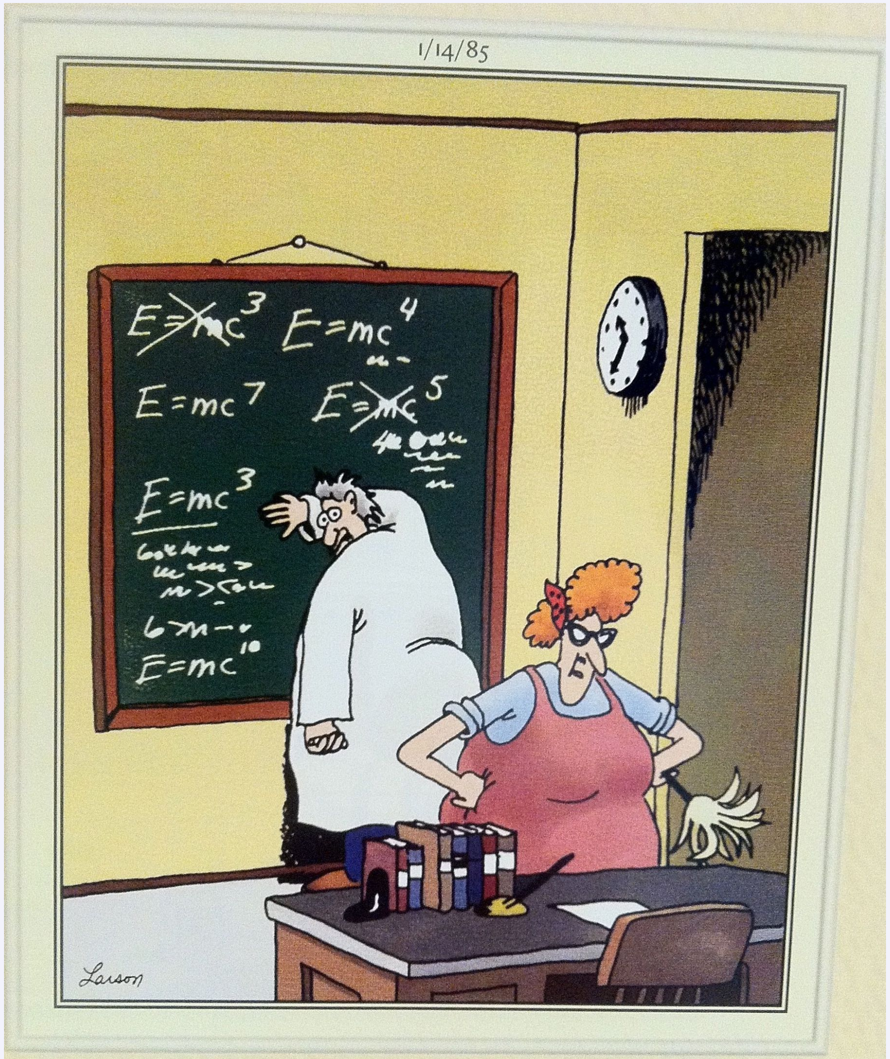
Simon (1966) offered another view – scientific discovery is a variety of *problem solving* that involves:

- *Search* through a space of *problem states*
- Generated by applying mental *operators*
- Guided by *heuristics* to make it tractable

Heuristic search had been implicated in many cases of human cognition, from proving theorems to playing chess.

This framework offered not only a path to understand scientific discovery, but also ways to *automate* this mysterious process.

Einstein's Search Succeeds At Last



"*Now* that desk looks better. Everything's squared away, yessir, squaaaaaared away."

The Task of Scientific Discovery

We can state the discovery task in terms of the inputs provided and the outputs produced:

- Given: *Scientific data or phenomena to be described or explained*
- Given: *Knowledge and heuristics about the scientific domain*
- Given: *A space of candidate laws, hypotheses, or models*
- Find: *Laws or models that describe or explain the observations*

The outputs should not only generalize well; they should be stated in an *established scientific formalism*.

Early Decades of Scientific Discovery

Research on computational scientific discovery has addressed many different forms of laws and models.

1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000
Bacon.1–Bacon.5						Abacus, Coper		Fahrehneit, E*, Tetrad, IDSN				Hume, ARC		DST, GPN LaGrange			SDS		SSF, RF5, LaGramge		
←AM				Glauber		NGlauber					IDSQ, Live						RL, Progol			HR	
←Dendral				Dalton, Stahl			Stahlp, Revolver		Gell-Mann			BR-3, Mendel		Pauli		BR-4					
						IE			Coast, Phineas, AbE, Kekada					Mechem, CDP						Astra, GPM	

Legend Numeric laws | Qualitative laws | Structural models | Process models

Successes of Computational Discovery

AI systems of this type have helped to discover new knowledge in many scientific fields:

- reaction pathways in catalytic chemistry (Valdes-Perez, 1994, 1997)
- qualitative chemical factors in mutagenesis (King et al., 1996)
- quantitative laws of metallic behavior (Sleeman et al., 1997)
- quantitative conjectures in graph theory (Fajtlowicz et al., 1988)
- qualitative conjectures in number theory (Colton et al., 2000)
- temporal laws of ecological behavior (Todorovski et al., 2000)
- models of gene-influenced metabolism in yeast (King et al., 2009)

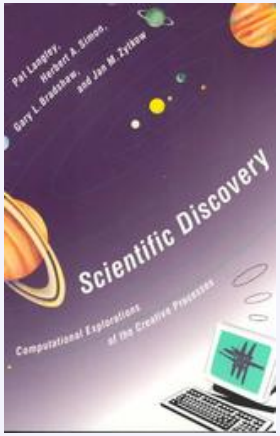
Each of these led to publications in the *refereed literature of the relevant scientific field*.

Meetings on Scientific Discovery

- Stanford University / January 1989
  Symposium on Computational Models of Scientific Discovery and Theory Formation
- Stanford University / March 1995
  AAAI Spring Symposium on Systematic Methods of Scientific Discovery
- Brighton, UK / August 1998
  ECAI-98 Workshop on Machine Discovery
- University of Pavia / December 1998
  Conference on Model-Based Reasoning in Scientific Discovery
- Stanford University / March 2001
  Symposium on Computational Discovery of Communicable Knowledge
- Stanford University / March 2008
  Symposium on Computational Approaches to Creativity in Science
- Arlington, VA / November 2012
  AAAI Fall Symposium on Discovery Informatics
- Carnegie Mellon Silicon Valley / June 2013
  Symposium on Cognitive Systems and Discovery Informatics
- San Mateo, CA / March 2023
  AAAI Spring Symposium on Computational Approaches to Scientific Discovery

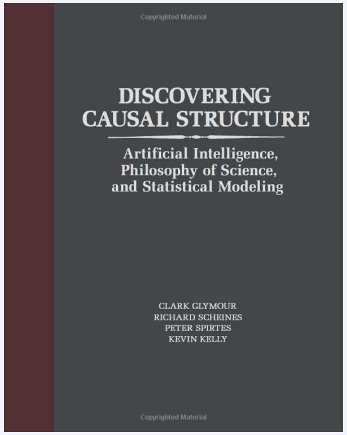
Books on Scientific Discovery

Research on computational discovery has led to multiple books.

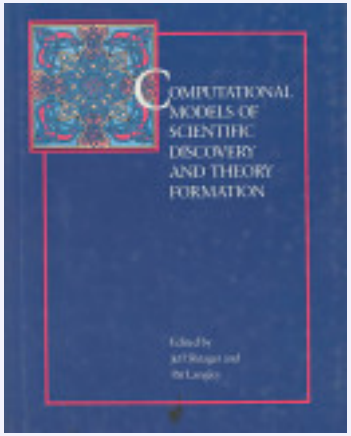


1987

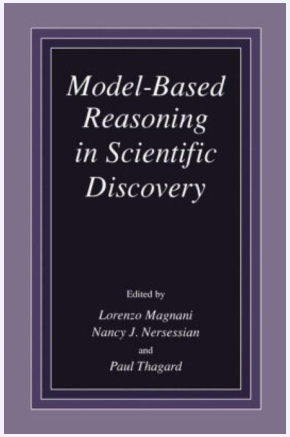
1987



1990



1999



2007

These demonstrate the field's diversity of problems and methods.

The Data Mining Movement

During the 1990s, a new research paradigm – known as *data mining* – emerged that:

- Emphasized the availability of large amounts of data
- Used computational methods to find regularities in the data
- Adopted heuristic search through a space of hypotheses
- Initially focused on commercial applications and data sets

Most work used notations invented by computer scientists, unlike scientific discovery, which used *scientific formalisms*.

Data mining has been applied to scientific data, but the results seldom bear a resemblance to scientific *knowledge*.

Six Varieties of Scientific Discovery

We can divide scientific discovery into six broad classes of computational activities:

- Formation of taxonomic hierarchies
- Discovery of qualitative laws
- Discovery of numeric laws / equations
- Formation of structural models
- Creation of causal models
- Construction of process models

I will cover the topics in this order, which mirrors their typical appearance in the history of science.

In closing, I will discuss their integration and relation to other aspects of science, such as experimentation.

Forming Taxonomic Hierarchies

Scientific Taxonomies

Taxonomies provide the most basic form of scientific knowledge in that they:

- Define categories or types of entities
- Associate specific entities with those types
- Organize these types into an IS-A hierarchy

Taxonomies provide the basis for other varieties of scientific information processing.

Thus, taxonomies *logically precede* other types of knowledge, although later results can modulate them.

Examples of Taxonomies

Taxonomic hierarchies play prominent roles in every scientific discipline. Examples include:

- Astronomy (*planets*, *moons*, *stars*, *asteroids*, *comets*)
- Biology (*animals*, *mammals*, *primates*, *apes*, *homo sapiens*)
- Chemistry (*elements*, *metals*, *nobles*, *compounds*, *organics*)
- Diseases (*bacterial*, *viral*, *parasitic*, *autoimmune*, *cancer*)
- Particle physics (*baryons*, *leptons*, *protons*, *electrons*, *muons*)

Such taxonomies evolve over time as scientists observe and categorize new entities.

Uses of Taxonomies

Scientists use taxonomic hierarchies for a number of purposes. These include:

- Classifying new entities or events into existing categories
- Predicting the features or behavior of new entities
- Describing higher-level knowledge in which types participate)

Thus, taxonomies provide fundamental support for the overall scientific process.

This is reflected by current interest in tools for developing and using *ontologies* like OWL-DL and Protégé.

The Task of Taxonomy Formation

We can specify the problem of taxonomy formation in terms of inputs and outputs:

- *Given:* A set of observed entities with associated descriptors
- *Given:* A space of possible taxonomic hierarchies
- *Find:* A set of categories and entities associated with them
- *Find:* Descriptions for each of these categories
- *Find:* A taxonomy that organizes categories in a hierarchy

This is an unsupervised discovery task that is closely related to the problem of *clustering*.

Taxonomy Formation as Heuristic Search

We can view taxonomy formation as search through a space of taxonomic hierarchies. This requires:

- A direction in which to construct the taxonomy
  - *E.g., from the bottom up or the top down*
- Criteria for assigning entities to categories
  - *E.g., a similarity of distance metric*
- A strategy for characterizing categories
  - *E.g., general to specific, specific to general, statistical*

Most methods carry out batch processing, but incremental approaches are also possible.

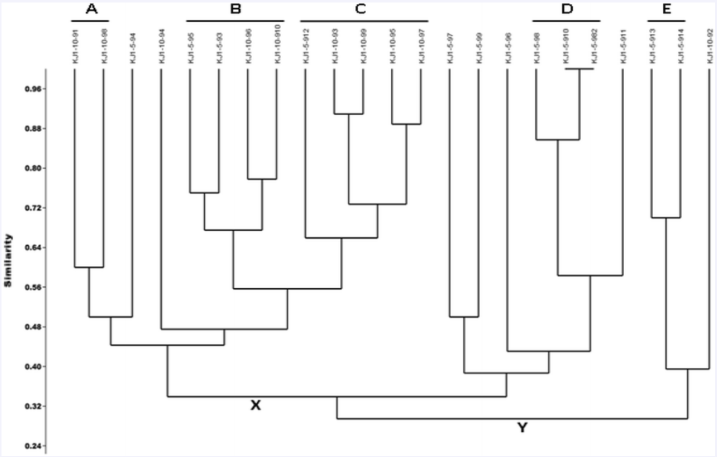
Numerical Taxonomy

An early use of computers in biology was *numerical taxonomy* (Sokal & Sneath, 1963), which offered ways to:

- Represent / store organisms' phenotypes in digital form
  - Often encoded as Booleans features (present or absent)
- Define and compute the *similarity* between two species or taxa
- Use these scores to guide search in the space of *dendrograms*

Techniques typically carried out greedy search, with the results depending on similarity measure.

Modern methods instead focus on *computational phylogenetics*.



Approaches to Taxonomy Formation

Computational researchers have developed three paradigms for taxonomy formation:

- *Agglomerative* methods
  - Find two nearest cases or clusters, merge them, and recurse
- *Divisive* methods
  - Separate cases into classes, then recurse to divide them further
- *Iterative optimization*
  - To find a single partition, assign cases to one of *K* random groups and reassign them iteratively until convergence

Iterative optimization can be used as a subroutine for divisive taxonomy construction.

Case Study: AutoClass

Cheeseman et al. (1988) reported AutoClass, a probabilistic system for taxonomy formation that:

- Represented categories in terms of means and variances
- Initially assigned entities to *K* classes at random
- Used expectation maximization to update class descriptions
- Increased the numbers of categories until no benefit seen

They applied AutoClass to infrared data on 5425 stars at 94 wavelengths to 77 distinct classes.

These included a new class of blackbody stars with significant infrared excess, possibly due to surrounding dust.

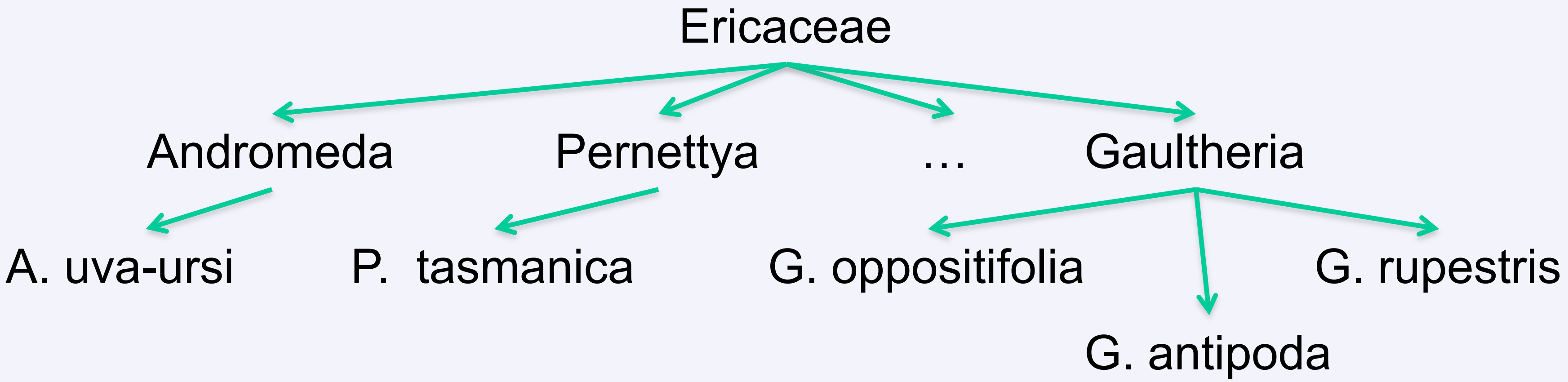
Case Study: RETAX

ReTAX (Alberdi & Sleeman, 1997) revised its taxonomies in response to new observations.

Each case was described a set of features (e.g., leaf size, type of fruit), and a target category.

If a new case did not match the category's defining features, the system attempted to revise its taxonomy.

E.g., ReTAX proposed merging genera *Pernettya* and *Gaultheria*.



Inducing Quantitative Laws

Qualitative Laws

A second form of knowledge, *qualitative laws*, use defined taxonomic concepts to specify:

- *Relations* that hold among entities or their attributes
- The *conditions* under which those relations occur

Such regularities may involve numeric attributes but they do *not* include equations or parameters.

These sometimes have *causal* interpretations but they may be simple associations.

Qualitative laws appear early in the history of a discipline but only after taxonomies have been formed.

Examples of Qualitative Laws

Like taxonomies, qualitative laws occur throughout the sciences. Examples include:

- Astronomy (*Sun rises and sets*, *stars revolve*, *planets wander*)
- Chemistry (*acids react with alkalis*, *iron rusts*, *salt dissolves*)
- Thermodynamics (*heated water boils*, *temperatures equalize*)
- Ecology (*fish live in water*, *dolphins feed on fish*)

Qualitative laws may describe either static relations or ones that involve change over time.

Uses of Qualitative Laws

Scientific researchers can use qualitative laws in a variety of ways. These include:

- Describing the behavior of known classes of entities
- Predicting the behavior of newly discovered entities
- Providing context for stating quantitative relations

Qualitative laws move beyond taxonomic knowledge to specify relations among known categories.

The Task of Qualitative Discovery

We can specify the problem of qualitative discovery in terms of inputs and outputs:

- *Given:* A set of observed entities, their features, and relations
- *Given:* A space of possible rules or generalized relations
- *Find:* A set of qualitative laws that describe the observations
- *Find:* Conditions under which these laws appear to hold

Because many qualitative laws can be stated as rules, the task is closely related to *rule induction*.

Thus, there are some cases in which methods for ‘data mining’ can aid the discovery process.

Qualitative Discovery as Heuristic Search

We can approach this task as heuristic search through a space of qualitative relations. This requires:

- An initial hypothesis or relation from which to start
  - *E.g., an empty set of conditions*
- Operators for generating or modifying candidate hypotheses
  - *E.g., adding or removing conditions*
- Heuristics for evaluating the quality of candidate hypotheses
  - *E.g., ability to summarize the data, law simplicity*
- A termination criterion for when to halt search
  - *E.g., when no further improvement occurs*

Again, this maps nicely onto rule-induction methods, although observations may not be labeled.

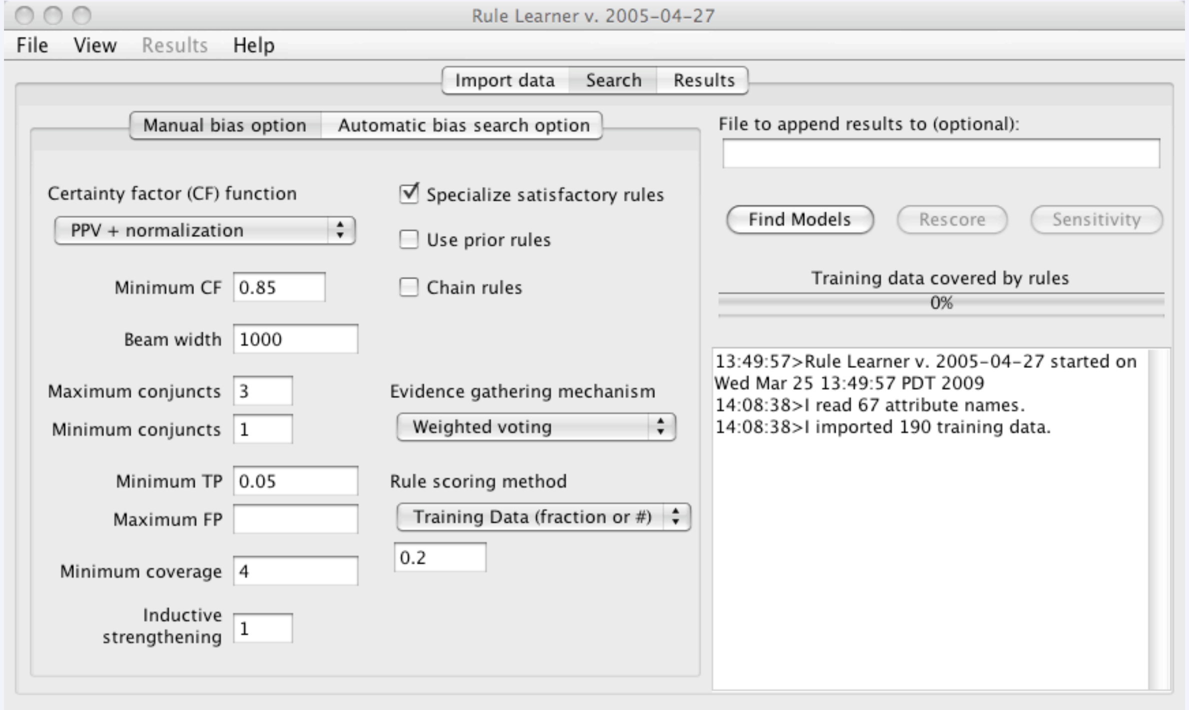
Case Study: The RL System

The RL system (Lee et al., 1998) induced qualitative laws that it expressed as logical rules.

Each rule stated that, if certain conditions held for an entity or situation, then it was a positive instance.

As input, RL took labeled training cases and details about:

- A hierarchy over attributes' values
- Constraints among rules' attributes
- Minimum accuracy
- Maximum attributes



Certainty factor (CF) function: PPV + normalization

Minimum CF: 0.85

Beam width: 1000

Maximum conjuncts: 3

Minimum conjuncts: 1

Minimum TP: 0.05

Maximum FP:

Minimum coverage: 4

Inductive strengthening: 1

☑ Specialize satisfactory rules

□ Use prior rules

□ Chain rules

Evidence gathering mechanism: Weighted voting

Rule scoring method: Training Data (fraction or #)

0.2

File to append results to (optional):

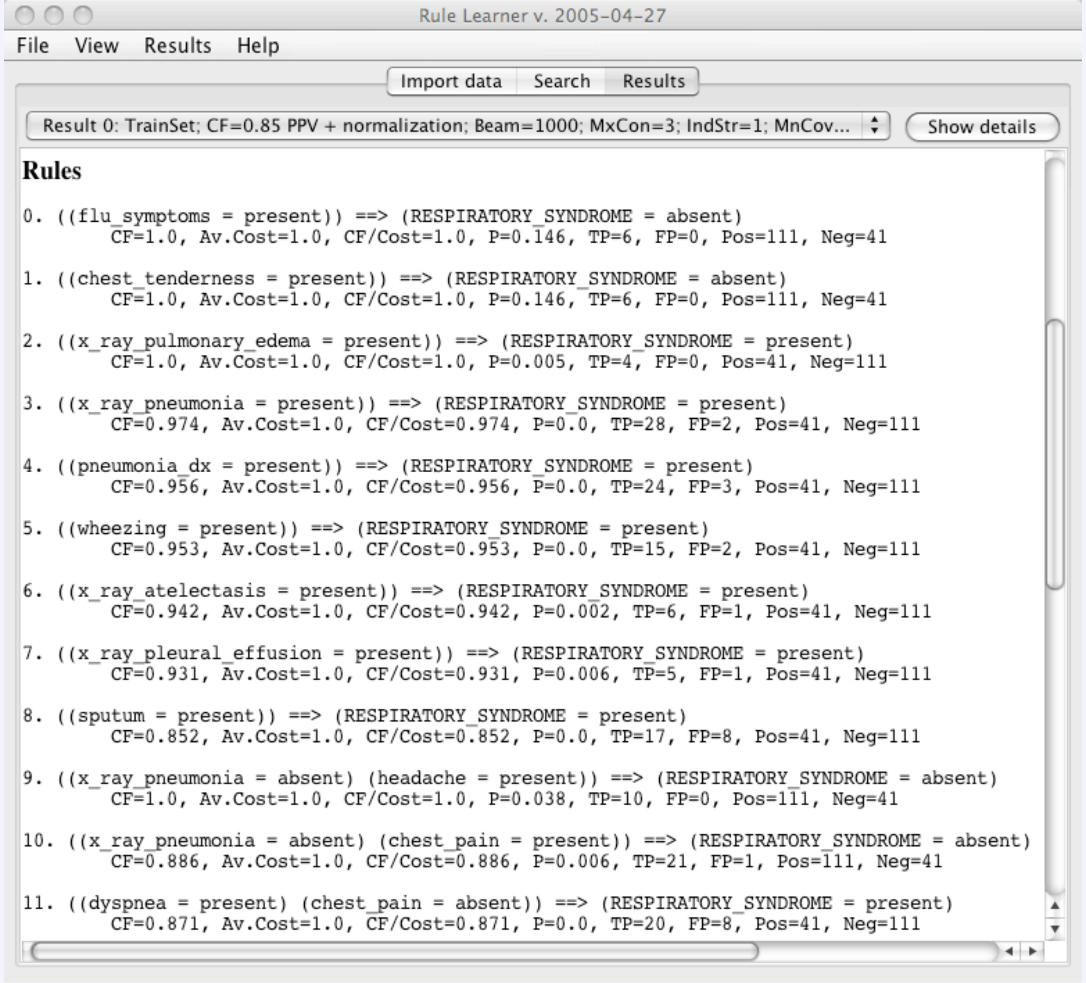
```
13:49:57>Rule Learner v. 2005-04-27 started on
Wed Mar 25 13:49:57 PDT 2009
14:08:38>I read 67 attribute names.
14:08:38>I imported 190 training data.
```

Case Study: The RL System

One application of RL involved finding links from symptoms to *lower respiratory syndrome*.

Each induced rule included a measure of support that it received from the data.

Other RL demonstrations included identification of *carcinogens* and prediction of *crystal formation*.



Result 0: TrainSet; CF=0.85 PPV + normalization; Beam=1000; MxCon=3; IndStr=1; MnCov...

Rules

```
0. ((flu_symptoms = present)) ==> (RESPIRATORY_SYNDROME = absent)
        CF=1.0, Av.Cost=1.0, CF/Cost=1.0, P=0.146, TP=6, FP=0, Pos=111, Neg=41

1. ((chest_tenderness = present)) ==> (RESPIRATORY_SYNDROME = absent)
        CF=1.0, Av.Cost=1.0, CF/Cost=1.0, P=0.146, TP=6, FP=0, Pos=111, Neg=41

2. ((x_ray_pulmonary_edema = present)) ==> (RESPIRATORY_SYNDROME = present)
        CF=1.0, Av.Cost=1.0, CF/Cost=1.0, P=0.005, TP=4, FP=0, Pos=41, Neg=111

3. ((x_ray_pneumonia = present)) ==> (RESPIRATORY_SYNDROME = present)
        CF=0.974, Av.Cost=1.0, CF/Cost=0.974, P=0.0, TP=28, FP=2, Pos=41, Neg=111

4. ((pneumonia_dx = present)) ==> (RESPIRATORY_SYNDROME = present)
        CF=0.956, Av.Cost=1.0, CF/Cost=0.956, P=0.0, TP=24, FP=3, Pos=41, Neg=111

5. ((wheezing = present)) ==> (RESPIRATORY_SYNDROME = present)
        CF=0.953, Av.Cost=1.0, CF/Cost=0.953, P=0.0, TP=15, FP=2, Pos=41, Neg=111

6. ((x_ray_atelectasis = present)) ==> (RESPIRATORY_SYNDROME = present)
        CF=0.942, Av.Cost=1.0, CF/Cost=0.942, P=0.002, TP=6, FP=1, Pos=41, Neg=111

7. ((x_ray_pleural_effusion = present)) ==> (RESPIRATORY_SYNDROME = present)
        CF=0.931, Av.Cost=1.0, CF/Cost=0.931, P=0.006, TP=5, FP=1, Pos=41, Neg=111

8. ((sputum = present)) ==> (RESPIRATORY_SYNDROME = present)
        CF=0.852, Av.Cost=1.0, CF/Cost=0.852, P=0.0, TP=17, FP=8, Pos=41, Neg=111

9. ((x_ray_pneumonia = absent) (headache = present)) ==> (RESPIRATORY_SYNDROME = absent)
        CF=1.0, Av.Cost=1.0, CF/Cost=1.0, P=0.038, TP=10, FP=0, Pos=111, Neg=41

10. ((x_ray_pneumonia = absent) (chest_pain = present)) ==> (RESPIRATORY_SYNDROME = absent)
        CF=0.886, Av.Cost=1.0, CF/Cost=0.886, P=0.006, TP=21, FP=1, Pos=111, Neg=41

11. ((dyspnea = present) (chest_pain = absent)) ==> (RESPIRATORY_SYNDROME = present)
        CF=0.871, Av.Cost=1.0, CF/Cost=0.871, P=0.0, TP=20, FP=8, Pos=41, Neg=111
```

Case Study: PROGOL

King et al. (1996) reported the discovery of qualitative factors that determine *mutagenicity*.

Given 230 nitro compounds, some mutagenic and others not, their ILP system PROGOL:

- Used heuristic search to find a rule that covered some cases
- Repeated this process to find others to cover the remainder

E.g., *a compound is mutagenic if it has an aliphatic carbon atom attached by a single carbon bond in a six-member aromatic ring*.

These relational descriptions offered insights into the deeper causes of mutation, unlike statistical approaches.

Case Study: The Glauber System

Langley et al.'s (1987) Glauber induced qualitative laws of chemistry from observations like:

- *(tastes HCl sour), (tastes NaOH bitter), (tastes NaCl salty)*
- *(reacts {HCl NaOH} {NaCl}), (reacts {HNO3 NaOH} {NaNO3})*

From these, the system formed quantified generalizations like

- *∀Acid (tastes Acid sour)*
- *∀Alkali (tastes Alkali bitter)*
- *∀Salt (tastes Salt salty)*
- *∀Acid ∀Alkali ∃Salt (reacts {Acid Alkali} {Salt})*

Glauber interleaves defining new categories with substituting their names into observed relations to generate laws.

Discovering Quantitative Laws

Quantitative Laws

A third type of knowledge, *numeric* or *quantitative laws*, moves beyond qualitative relations to specify:

- Functional forms that relate the attributes of entities
- Parameters associated with these functional forms
- The conditions under which these numeric laws hold

As with qualitative laws, these may be either causal relations or simple associations

Quantitative laws invariably appear after qualitative relations, which provide context for them.

Examples of Quantitative Laws

Quantitative laws are just as pervasive as taxonomies and qualitative relations. Examples include:

- Astronomy (*planetary periods*, *Kepler's laws*)
- Chemistry (*laws of combining weights*, *volumes*)
- Physics (*Coulomb's law*, *momentum*, *Snell's law*, *Ohm's law*)
- Thermodynamics (*ideal gas law*, *Black's law of specific heat*)

Such laws are the poster children of science, often presented in textbooks and popular treatments.

Researchers use quantitative laws in much the same ways as qualitative ones, but with more precision.

The Task of Equation Discovery

We can specify the problem of equation discovery in terms of inputs and outputs:

- *Given:* A set of observed entities with numeric descriptors
- *Given:* A space of possible functional forms with parameters
- *Find:* One or more numeric laws that describe the observations
- *Find:* Conditions under which these laws appear to hold

This task is similar to regression in statistics, but considers a much wider range of functional forms.

Note: Although sometimes called ‘symbolic regression’, this term is an oxymoron, as *all* regression is symbolic.

Equation Discovery as Heuristic Search

We can view equation discovery as heuristic search through a space of numeric laws. This requires:

- An initial equation structure from which to start
  - *E.g., an empty formula assuming an attribute is constant*
- Operators for generating or modifying candidate equations
  - *E.g., adding or removing terms, revising parameters*
- Heuristics for evaluating the quality of a candidate law
  - *E.g., terms are nearly constant, fit the observations*
- A termination criterion for when to halt search
  - *E.g., when the observations are fit sufficiently well*

Exhaustive search is possible in special cases, but many settings rely on heuristics to make search tractable.

Case Study: The Bacon System

Langley (1979, 1981) reported Bacon, an early AI system for quantitative discovery that:

- Carried out search in a problem space of theoretical terms;
- Using operators that combined old terms into new ones;
- Guided by heuristics that noted regularities in data; and
- Applied these recursively to formulate higher-level relations.

Bacon rediscovered a variety of numeric laws from the history of physics and chemistry.

The system was named after Sir Francis Bacon because it used a *data-driven* approach to discovery.

Bacon on Kepler’s Third Law

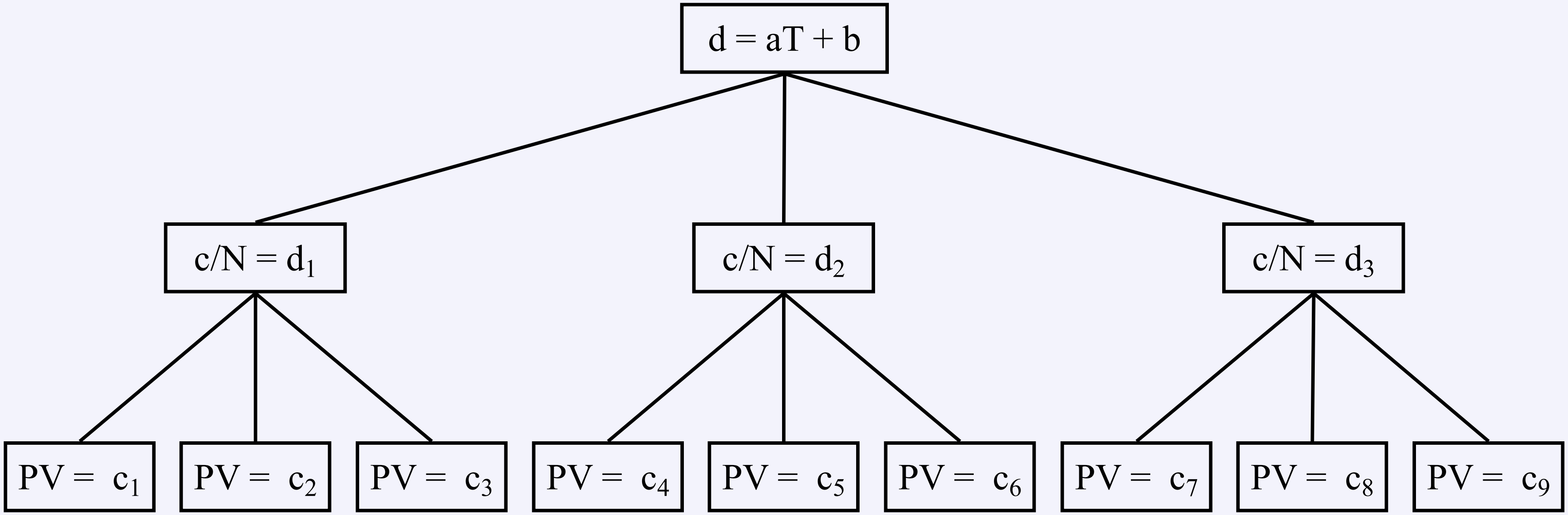
The Bacon system carried out heuristic search, through a space of numeric terms, looking for constants and linear relations.

moon	d	p	d/p	d^2/p	d^3/p^2
A	5.67	1.77	3.20	18.15	58.15
B	8.67	3.57	2.43	21.04	51.06
C	14.00	7.16	1.96	27.40	53.61
D	24.67	16.69	1.48	36.46	53.89

This table shows its progression from the distance and period of Jupiter’s moons to a term with nearly constant value.

Bacon on the Ideal Gas Law

Bacon rediscovered the ideal gas law, $PV = aNT + bN$, in three stages, each at a different level of description.



$d = aT + b$

$c/N = d_1$ $c/N = d_2$ $c/N = d_3$

$PV = c_1$ $PV = c_2$ $PV = c_3$ $PV = c_4$ $PV = c_5$ $PV = c_6$ $PV = c_7$ $PV = c_8$ $PV = c_9$

Parameters for laws at one level became dependent variables in laws at the next level, enabling discovery of complex relations.

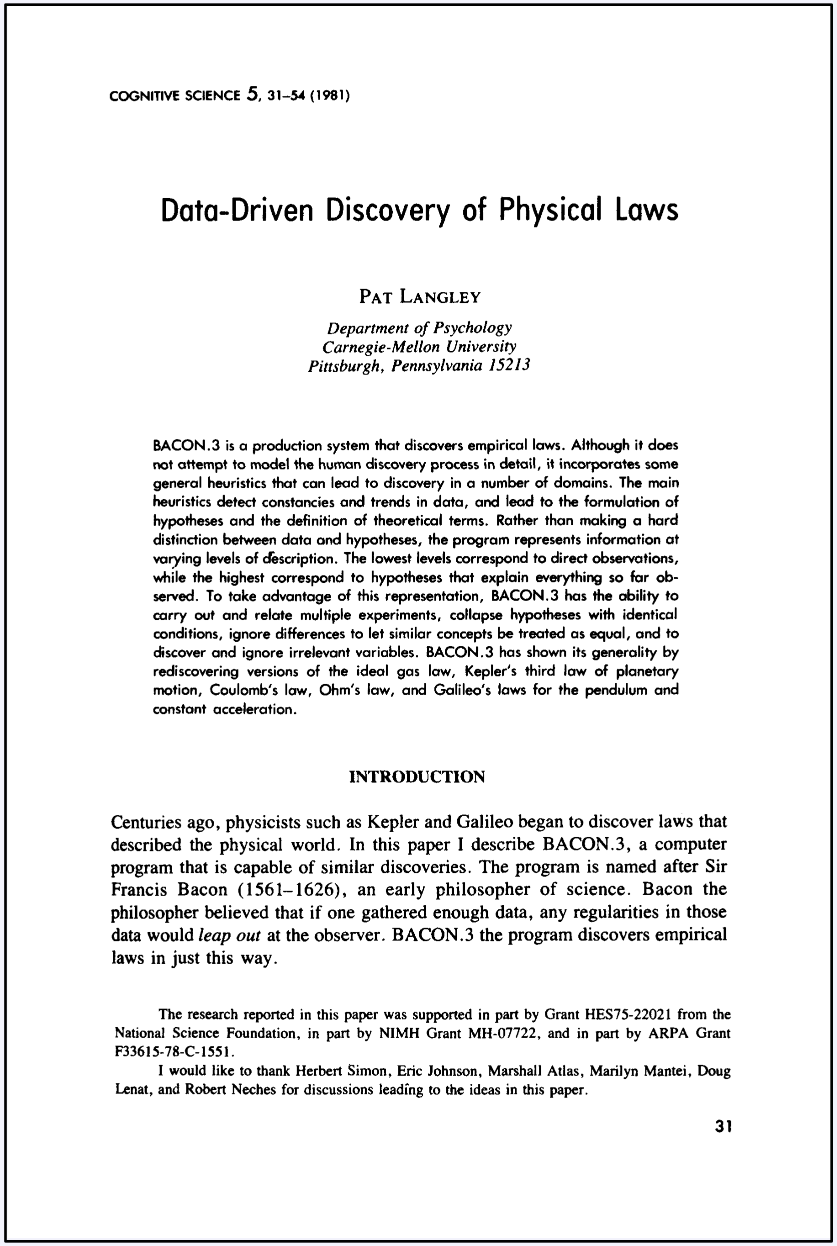
Numeric Laws Discovered by Bacon

Basic algebraic relations:

• Ideal gas law	$PV = aNT + bN$
• Kepler's third law	$D^3 = [(A - k) / t]^2 = j$
• Coulomb's law	$FD^2 / Q_1Q_2 = c$
• Ohm's law	$TD^2 / (LI - rI) = r$

Relations with *intrinsic properties*:

• Snell's law of refraction	$\sin I / \sin R = n_1 / n_2$
• Archimedes' law	$C = V + i$
• Momentum conservation	$m_1V_1 = m_2V_2$
• Black's specific heat law	$c_1m_1T_1 + c_2m_2T_2 = (c_1m_1 + c_2m_2) T_f$



COGNITIVE SCIENCE 5, 31–54 (1981)

Data-Driven Discovery of Physical Laws

PAT LANGLEY

Department of Psychology
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

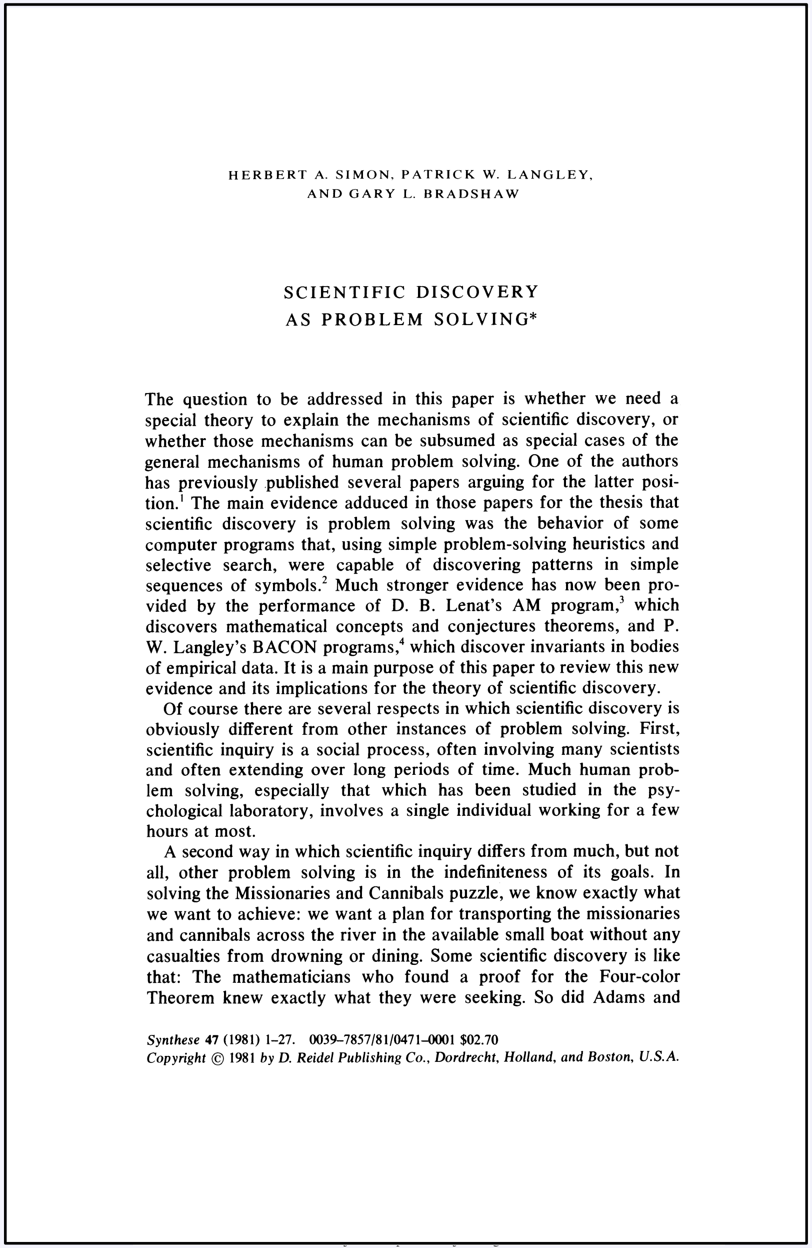
BACON.3 is a production system that discovers empirical laws. Although it does not attempt to model the human discovery process in detail, it incorporates some general heuristics that can lead to discovery in a number of domains. The main heuristics detect constancies and trends in data, and lead to the formulation of hypotheses and the definition of theoretical terms. Rather than making a hard distinction between data and hypotheses, the program represents information at varying levels of description. The lowest levels correspond to direct observations, while the highest correspond to hypotheses that explain everything so far observed. To take advantage of this representation, BACON.3 has the ability to carry out and relate multiple experiments, collapse hypotheses with identical conditions, ignore differences to let similar concepts be treated as equal, and to discover and ignore irrelevant variables. BACON.3 has shown its generality by rediscovering versions of the ideal gas law, Kepler's third law of planetary motion, Coulomb's law, Ohm's law, and Galileo's laws for the pendulum and constant acceleration.

INTRODUCTION

Centuries ago, physicists such as Kepler and Galileo began to discover laws that described the physical world. In this paper I describe BACON.3, a computer program that is capable of similar discoveries. The program is named after Sir Francis Bacon (1561–1626), an early philosopher of science. Bacon the philosopher believed that if one gathered enough data, any regularities in those data would *leap out* at the observer. BACON.3 the program discovers empirical laws in just this way.

The research reported in this paper was supported in part by Grant HES75-22021 from the National Science Foundation, in part by NIMH Grant MH-07722, and in part by ARPA Grant F33615-78-C-1551.

I would like to thank Herbert Simon, Eric Johnson, Marshall Atlas, Marilyn Mantei, Doug Lenat, and Robert Neches for discussions leading to the ideas in this paper.



HERBERT A. SIMON, PATRICK W. LANGLEY,
AND GARY L. BRADSHAW

SCIENTIFIC DISCOVERY AS PROBLEM SOLVING*

The question to be addressed in this paper is whether we need a special theory to explain the mechanisms of scientific discovery, or whether those mechanisms can be subsumed as special cases of the general mechanisms of human problem solving. One of the authors has previously published several papers arguing for the latter position.[1] The main evidence adduced in those papers for the thesis that scientific discovery is problem solving was the behavior of some computer programs that, using simple problem-solving heuristics and selective search, were capable of discovering patterns in simple sequences of symbols.[2] Much stronger evidence has now been provided by the performance of D. B. Lenat's AM program,[3] which discovers mathematical concepts and conjectures theorems, and P. W. Langley's BACON programs,[4] which discover invariants in bodies of empirical data. It is a main purpose of this paper to review this new evidence and its implications for the theory of scientific discovery.

Of course there are several respects in which scientific discovery is obviously different from other instances of problem solving. First, scientific inquiry is a social process, often involving many scientists and often extending over long periods of time. Much human problem solving, especially that which has been studied in the psychological laboratory, involves a single individual working for a few hours at most.

A second way in which scientific inquiry differs from much, but not all, other problem solving is in the indefiniteness of its goals. In solving the Missionaries and Cannibals puzzle, we know exactly what we want to achieve: we want a plan for transporting the missionaries and cannibals across the river in the available small boat without any casualties from drowning or dining. Some scientific discovery is like that: The mathematicians who found a proof for the Four-color Theorem knew exactly what they were seeking. So did Adams and

Synthese 47 (1981) 1–27. 0039–7857/81/0471–0001 $02.70

Successors to Bacon

Bacon inspired many additional systems for equation discovery:

- ABACUS (Falkenhainer, 1985) and ARC (Moulet, 1992)
- Fahrenheit (Zytkow, Zhu, & Hussam, 1990)
- COPER (Kokar, 1986) and E* (Schaffer, 1990)
- IDS (Nordhausen & Langley, 1990)
- Hume (Gordon & Sleeman, 1992)
- DST (Murata et al., 1994) and RF5 (Saito & Nakano, 1997)
- *LaGrange (Dzeroski & Todorovski, 1994) and PRET (Stolle, 1998)*
- *SSF (Washio et al., 1997) and GP (Koza et al., 2001)*

These relied on different methods but also searched for explicit mathematical laws that matched data.

Case Study: The RF5 System

Saito and Nakano's (1997) RF5 system used neural network technology to discover numeric laws by:

- Transforming a functional form into a three-layer network
  - With *product units* for hidden layer, *additive units* for top layer
- Using second-order gradient descent to search for parameters
- Halting search on finding weights that minimize an MDL score
- Transforming the induced network into a *polynomial expression*

This approach demonstrates that neural networks can produce interpretable results if used properly.

An extension, RF6, can also find conditions on numeric laws.

Case Study: The LaGramge System

Todorovski and Dzeroski’s (1997) LaGramge was a Bacon-like system that discovered *dynamic laws* from:

- Measurements for a *multivariate time series*
- A set of *dependent variables* to predict
- A space of possible equations specified as a *grammar*

The system produced an algebraic or differential equation for each dependent variable.

The developers applied LaGramge to ecological, hydrological, and other dynamic data sets.

E → E + F | E – F | F

F → F * T | F / T | T

T → *constant* | *variable* | (E)

Other Work on Dynamic Law Discovery

In recent years, a new line of research of inducing differential equation models has emerged:

- Brunton, Proctor, and Kutz (PNAS, 2016)
- Chen, Rubanova, Bettencourt, and Duvenau (NeurIPS, 2019)
- Cranmer et al. (NeurIPS, 2020)
- Iten, Metger, Wilming, Rio, and Renner (Phy Rev Letters, 2020).
- Raissi and Karniadakis (J Comp Physics, 2018)
- Schmidt and Lipson (Science, 2009)
- Wang, Maddix, Wang, Faloutsos, and Yu (NeurIPS Wkshp, 2020)
- Wu and Tegmark (Physical Review E, 2019)
- Zhang and Lin (Proc Royal Society, 2018)

This work emphasizes statistics more than older efforts, but also searches a space of models stated in scientific formalisms.

Forming Structural Models

Description vs. Explanation

The early stages of any scientific field focus on *descriptions* that summarize observed regularities.

Mature sciences instead emphasize the creation of *models* that *explain* phenomena in terms of interacting elements.

- Explanatory models move beyond description to provide deeper accounts linked to theoretical constructs.
- The activity of generating such accounts is often *abductive* rather than *inductive*.

We will examine three types of explanatory models: *structural*, *causal*, and *process*.

Laws vs. Models

To understand the core difference between descriptions and explanations, we must distinguish between:

- *Laws*, which are isolated (often relational) statements
  - E.g., *hydrogen reacts with oxygen*, $PV = aNT + bN$
- *Models*, which are collections of linked law-like elements
  - E.g., *chains of reactions*, *sets of equations*

Thus, law-like elements are the *building blocks* of models that move beyond simple description.

An important feature is that some model elements may be *inferred* rather than observed.

Structural Models

A *structural model* is a variety of scientific explanation that specifies:

- An *observed entity* and its associated *descriptors*
- A set of *constituents* which compose that entity
- A set of *relations* among those constituents (optional)

A collection of such models typically share some inferred constituents.

The models also share assumptions about how to *derive* the observed features from constituents.

Examples of Structural Models

Structural models arise in many fields of science. Examples include:

- Chemical structures (*H2O*, *NH3*, *NaOH*, *benzene*, *acetone*)
- Gene sequences (*for different organisms*)
- Geological deposits (*proportions of different minerals*)
- Stellar compositions (*proportions of hydrogen, helium, carbon*)

These specify entities' building blocks and, in some instances, how they fit together.

Such models always have qualitative structure but they may also include numeric information.

Uses of Structural Models

Scientists can use structural models in multiple ways, such as invoking them to explain:

- Why observed entities have their measured characteristics
- Why some entities occur in nature but others do not
- How to create instances of these entities from components

Such models take one beyond description to provide a deeper understanding of phenomena.

The Task of Structural Modeling

We can specify the problem of structural modeling in terms of inputs and outputs:

- *Given:* A set of observed entities with associated descriptors
- *Given:* A space of possible structural models
- *Find:* Structural models that explain the observed entities
- *Find:* Unobserved but inferred entities and relations (optional)

This task typically involves *abductive inference* rather than induction from data.

Structural Modeling as Search

We can view this task as heuristic search through a space of structural models. This requires:

- An initial set of models from which to start
  - *E.g., an empty model for each observed entity*
- Operators for generating or revising current models
  - *E.g., adding or removing constituents*
- Heuristics for evaluating the quality of a candidate model
  - *E.g., ability to account for observed descriptors*
- A termination criterion for when to halt search
  - *E.g., when all observed entities have been explained*

Naturally, details will differ depending on the class of structural models being considered.

Case Study: The Dalton System

Langley et al.'s (1987) Dalton inferred the constituent structure of substances from chemical reactions.

E.g., starting from the reaction (hydrogen oxygen → water), it inferred the model:

({{h h} {h h}} {{o o}} → {{h h o} {h h o}})

This account asserted that:

- Hydrogen and oxygen molecules are diatomic; and
- Hydrogen and oxygen molecules combine in a 2:1 ratio to produce two water molecules.

DALTON arrived at its discoveries through a heuristic search guided by knowledge available to 19th Century chemists.

Case Study: Gell-Mann

Zytkow and Fischer’s (1990) GELL-MANN system postulated hidden structures in particle physics.

- As input, it took a collection of known particles and their quantum properties;
- As output, the system produced a ‘bag’ of components for each particle and associated property values.

For example, when given descriptions of seven elementary particles, it produced the *baryon octet* model.

The system also mapped baryons to arrangements of quarks and conjectured values for their properties.

Case Study: Gell-Mann

Inputs:

particle	charge	isospin	strange.
p	1	1/2	0
n	0	-1/2	0
Σ^+	1	1	-1
Σ^0	0	0	-1
Σ^-	-1	-1	-1
Ξ^0	0	1/2	-2
Ξ^-	-1	-1/2	-2

Given descriptions of elementary particle, GELL-MANN infers the standard octet quark model.

Outputs:

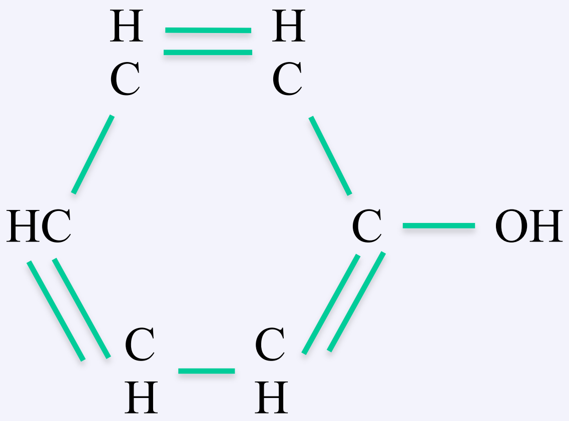
quark	charge	isospin	strange.
u	2/3	1/2	0
d	-1/3	-1/2	0
s	-1/3	0	-1

part.	ch.	iso.	str.	quarks
	1	0	1	uuu
p	1	1/2	0	uud
n	0	-1/2	0	uus
Σ^+	1	1	-1	udd
Σ^0	0	0	-1	uds
Σ^-	-1	-1	-1	uss
	-1	-3/2	0	ddd
Ξ^0	0	1/2	-2	dds
Ξ^-	-1	-1/2	-2	dss
	-1	0	-3	sss

Inferring Chemical Structures

DENDRAL (Lindsay et al., 1980) inferred a molecule's chemical bonds given its *component formula* and a *mass spectrogram*.

E.g., from the formula C_6H_5OH and other information, it found organic structures like:



DENDRAL relied on heuristic search to infer structural models, using knowledge from 20th Century chemistry as a guide.

Many of its results appeared in the refereed chemistry literature.

Case Study: Inferring Genomes

Early methods for DNA sequencing reconstructed a complete genome by combining many fragments.

Systems like ARACHNE and Celera Assembler addressed this problem by:

- Finding subsequences repeated across fragments
- Detecting and correcting errors
- Joining overlapping fragments into contiguous regions

The systems included several checks to ensure the resulting structure was well supported by the data.

More recent sequencing methods have made this less critical.

Discovering Causal Models

Causal Models

A *causal model* is an abstract form of scientific explanation that specifies:

- A set of *variables* or *events*, at least some of them observable
- A set of *causal links* that connect these variables or events
- Assumptions about how to *combine* causal influences

That is, a causal model is a collection of law-like elements, either qualitative or quantitative in character.

A causal model may have *deterministic* influences, *stochastic* influences, or a *mixture* of them.

Abstract causal models are rare in science, but they appear in biology, medicine, and the social sciences.

What is a Causal Influence?

We can define causality in abstract but clear terms; we will say that variable X *causally influences* variable Y if:

- A change in X’s value results in a change to Y’s value
- Provided that other variables are held constant

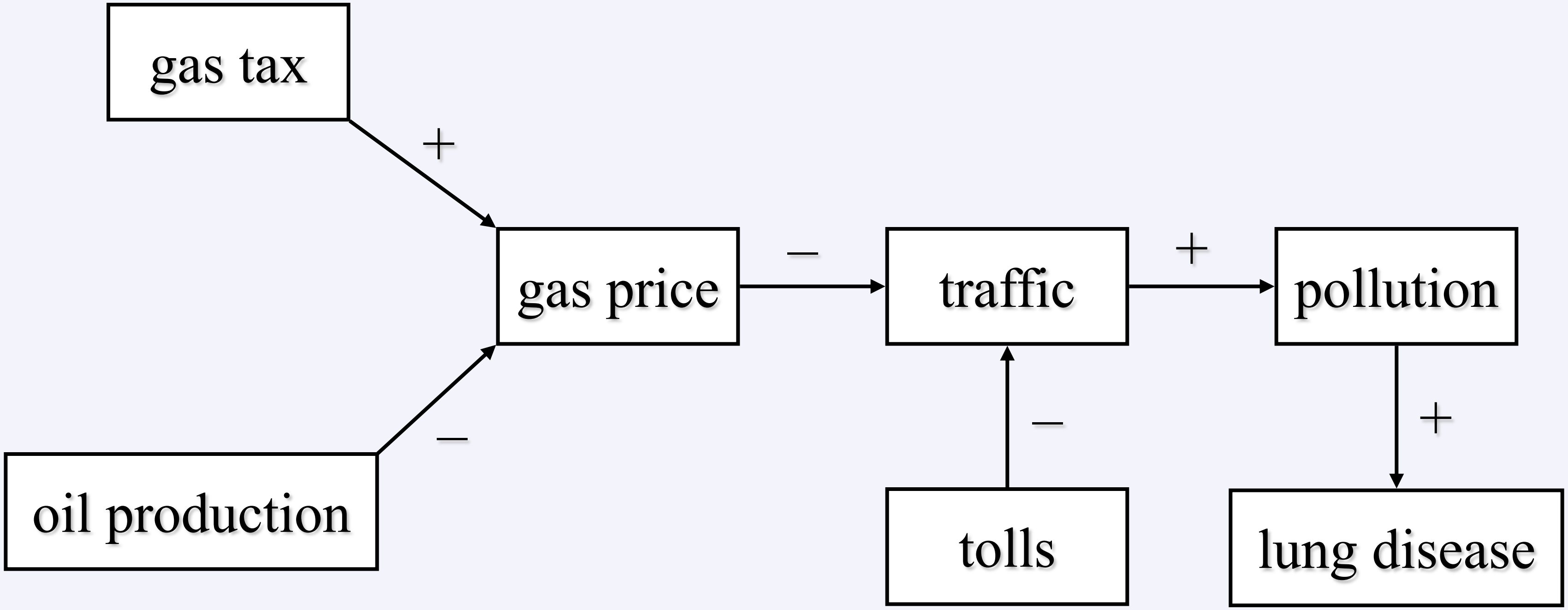
Note that this definition of causality does not state:

- That X is the only causal influence on Y
- The functional form of the causal relation

Such abstract information can be useful even when influences are probabilistic rather than deterministic.

A Qualitative Causal Model

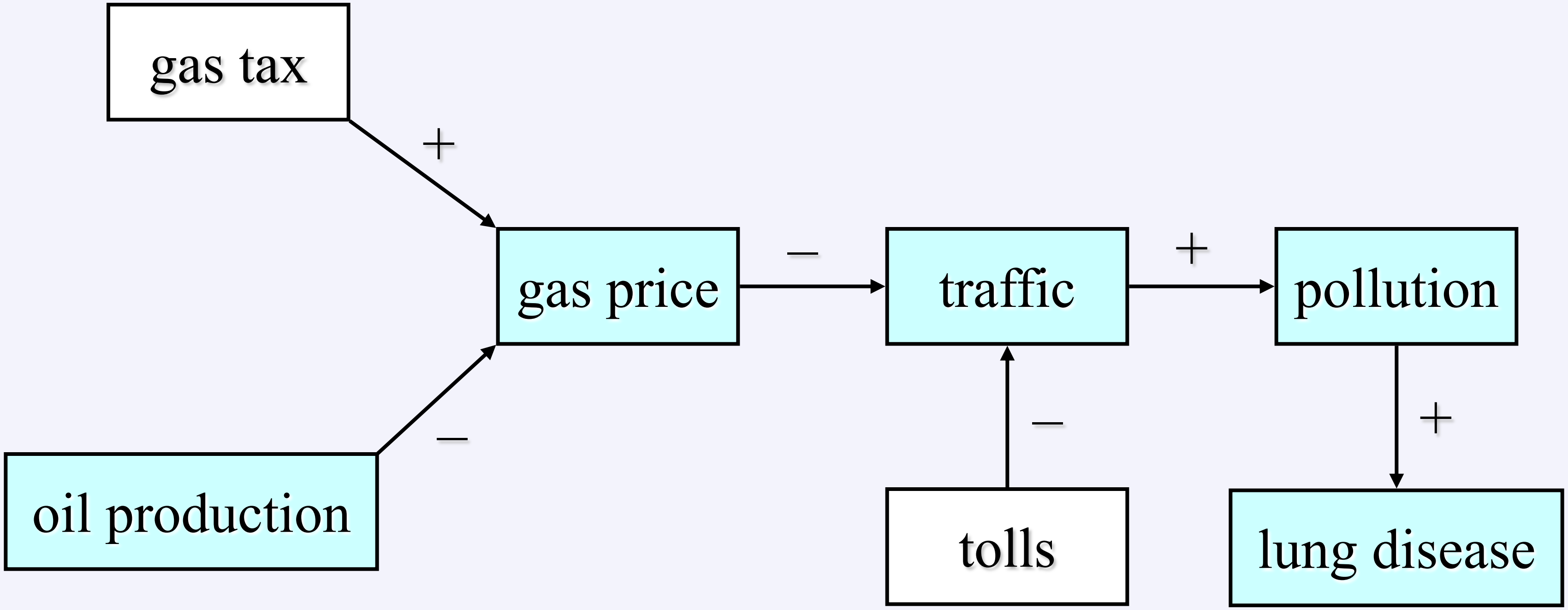
Consider a simple qualitative causal model about lung disease.



This model includes a set of *qualitative* causal influences among *quantitative* variables.

Using a Qualitative Causal Model

We can use causal chaining to make predictions from our model.

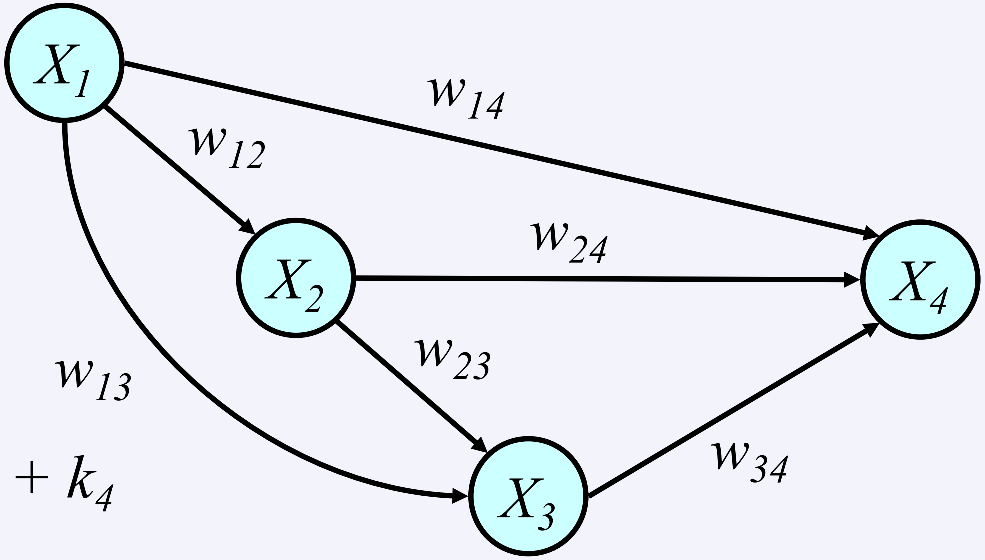


This pathway indicates that an increase in oil production will lead to an increase in lung disease.

Structural Equation Models

There are also quantitative types of causal accounts, as in *structural equation models*.

$$X_1 = k_1$$
$$X_2 = w_{12} X_1 + k_2$$
$$X_3 = w_{13} X_1 + w_{23} X_2 + k_3$$
$$X_4 = w_{14} X_1 + w_{24} X_2 + w_{34} X_3 + k_4$$



Sometimes called *linear causal models*, these take the form of directed acyclic graphs, which have no loops.

These are closely akin to Bayesian networks, although they were introduced in the 1920s.

Causal Model Discovery as Search

As before, we can view causal model discovery as search in a space of model structures if we specify:

- An initial model from which to start the search
  - E.g., *an empty model or a fully connected graph*
- Operators for generating or revising current models
  - E.g., *adding or removing a causal link*
- Heuristics for deciding whether to add or remove a link
  - E.g., *ability to explain observed variations*
- A termination criterion for when to halt search

Experimental control is a powerful aid for causal inference, but it is definitely not required.

A primary counterexample is Glymour et al.'s (1987) TETRAD.

Heuristic Search in TETRAD

TETRAD used the ‘PC’ algorithm to search through a space of linear causal models by:

- Initializing the model to be a *complete*, *undirected graph*.
- Removing an edge between variables if they are *conditionally independent* given values of other terms.
- Giving each edge a direction based on the connectivity of the reduced undirected graph.

The tests for conditional independence involve four-way relations among variables’ partial correlations.

This approach lets TETRAD infer causal influences / directions from *nonexperimental* data.

Case Study: Modeling Gene Regulation

An important application of causal model discovery involves the inference of *gene regulatory networks*:

- *Given:* Expression levels for genes in different situations
- *Given:* Background knowledge of gene types / functions
- *Find:* Which genes influence the expression of other genes
- *Find:* Whether these influences facilitate or inhibit activity

There is a large literature in computational biology on inferring gene regulation networks.

Bayes nets are popular, but Bay et al. (2003) used TETRAD and Zupan et al. (2003) used abductive reasoning.

Discovering Process Models

Process Models

A *process model* is another form of scientific explanation that specifies:

- Observed *entities* and *descriptors* at different times
- A set of *processes* involving those and other entities
- A set of *interactions* among those processes

Taken together, the processes and their interactions explain the observations, typically through chaining.

Such process models often refer to the constituents of entities and thus build on structural accounts.

Examples of Process Models

Process models also occur throughout the sciences. Examples include:

- Metabolic pathways (e.g., glycolysis, urea cycle)
- Nuclear reaction networks (e.g., stellar nucleosynthesis)
- Geological process models (e.g., evolution of landforms)
- Ecological process models (e.g., food and nutrient chains)

These specify a system’s constituent processes and interactions among them.

Models specify the qualitative organization but may include numeric annotations.

Uses of Process Models

Scientists can use such process models in multiple ways, such as to clarify:

- Why observed entities have their measured characteristics
- Why some entities occur in nature but others do not
- How to create instances of these entities from components

They let one move beyond description to deeper understanding of dynamic phenomena.

Most process models have a *causal interpretation* but organize content in higher-level terms.

The Task of Process Modeling

We can specify the problem of process modeling in terms of inputs and outputs:

- *Given:* A set of entities described at different points in time
- *Given:* A space of possible process models
- *Find:* A set of interacting processes that explain this behavior
- *Find:* Unobserved but inferred entities in the processes (optional)

Again, this task typically involves abductive explanation rather than induction from data.

Process Modeling as Heuristic Search

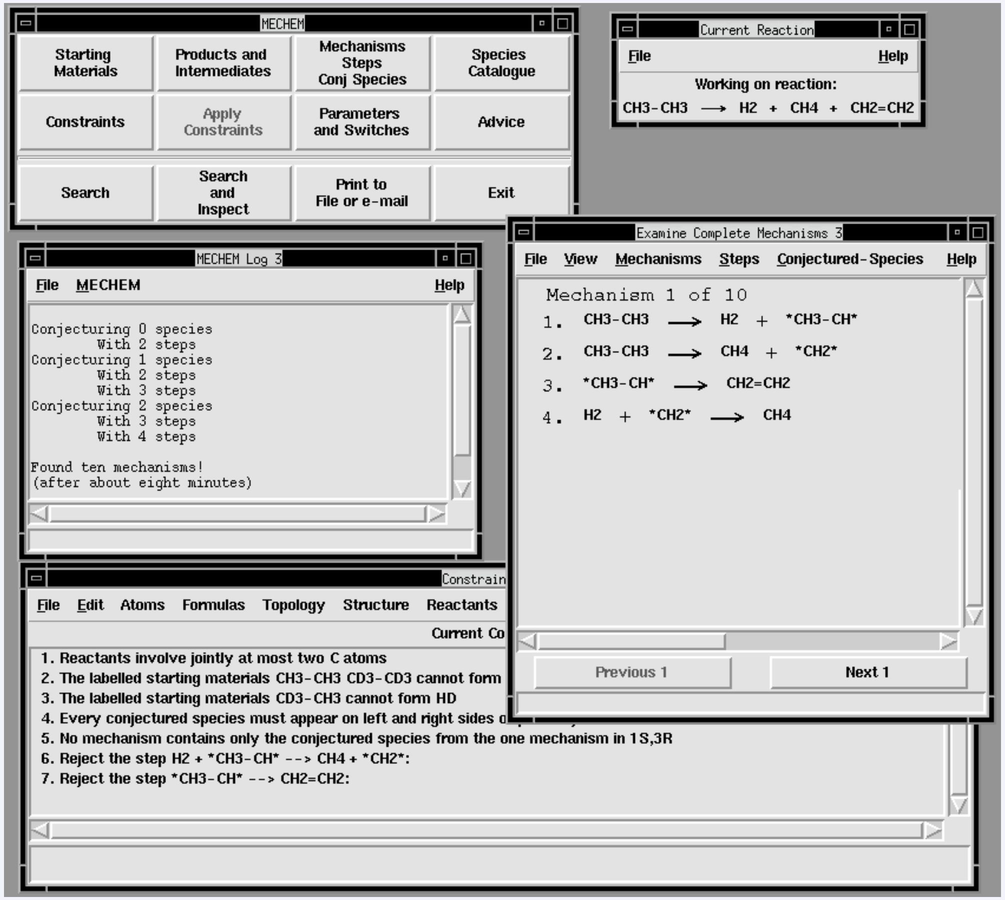
We can view this task as heuristic search through a space of process models. This requires:

- An initial model from which to start
  - *E.g., an empty model with no processes*
- Operators for generating or revising current models
  - *E.g., adding or removing processes*
- Heuristics for evaluating the quality of a candidate model
  - *E.g., ability to account for observed descriptors*
- A termination criterion for when to halt search
  - *E.g., when all observed phenomena are explained*

Both the organization of the search space and heuristics are crucial to making this tractable.

Case Study: MECHEM

MECHEM (Valdes-Perez, 1994) generated chemical pathways to explain observed reactions.



Working on reaction:

CH3-CH3 ⟶ H2 + CH4 + CH2=CH2

```
Conjecturing 0 species
        With 2 steps
Conjecturing 1 species
        With 2 steps
        With 3 steps
Conjecturing 2 species
        With 3 steps
        With 4 steps

Found ten mechanisms!
(after about eight minutes)
```

```
Mechanism 1 of 10
1.  CH3-CH3  ⟶  H2 + *CH3-CH*
2.  CH3-CH3  ⟶  CH4 + *CH2*
3.  *CH3-CH*  ⟶  CH2=CH2
4.  H2 + *CH2*  ⟶  CH4
```

1. Reactants involve jointly at most two C atoms
2. The labelled starting materials CH3-CH3 CD3-CD3 cannot form
3. The labelled starting materials CD3-CH3 cannot form HD
4. Every conjectured species must appear on left and right sides o
5. No mechanism contains only the conjectured species from the one mechanism in 1S,3R
6. Reject the step H2 + *CH3-CH* --> CH4 + *CH2*:
7. Reject the step *CH3-CH* --> CH2=CH2:

The system used constrained exhaustive search to generate candidate explanations.

Users could select constraints they deemed relevant to the current task.

MECHEM found numerous pathways that led to articles in the chemistry literature.

Case Study: MECHEM

MECHEM used constrained exhaustive search through a space of candidate pathways by:

- Favoring pathways with few species and steps
- Ensuring the unique generation of each pathway
- Requiring balanced chemical equations
- Limiting steps to two reactants and two products

These general constraints limited search drastically even before users added task-specific knowledge.

1. $H_2 + MM \rightarrow 2MH$
2. $CO + MM \rightarrow M_2CO$
3. $MH + M_2CO \rightarrow M_2CHOM$

Partial reaction pathway found by MECHEM

Case Study: The ACE System

Anderson et al. (2014) reported ACE, a system for *cosmogenic dating* in geology that:

- Inputs nucleotide densities for rocks from a landform
- Incorporates knowledge about possible geological processes
- Generates process models for how the landform was produced
- Weighs arguments for and against each process explanation

ACE was downloaded ~600 times and was used actively by many geologists to understand their data.

Case Study: Food Webs in Ecology

In other process model work, Bohan et al. (2011) used abductive logic programming to:

- Process data on relative abundances on invertebrates in fields
- Use knowledge about relative size, cooccurence, and predation
- Infer a three-level food web that relates 45 distinct species

Examination of the literature showed that most of these links were consistent with known predatory relations.

However, the system also hypothesized novel predations that ecologists found interesting and important.

Unified and Integrated Discovery

Discovery as Search in a Matrix Space

Valdes-Perez, Simon, and Zytkow (1993) offer a unified analysis of seven disparate discovery systems:

- Stahl, which discovers chemical compounds from reactions
- Dalton, which infers molecular models from reactions
- MECHEM, which postulates chemical reaction pathways
- Gell-Mann, which infers structures of elementary particles
- BR-3 and PAULI, which posit new properties for particles
- Mendel, which infers genotype interactions from phenotypes

They describe each system as searching through a space of two or more *matrices* that can vary in size.

The IDS System

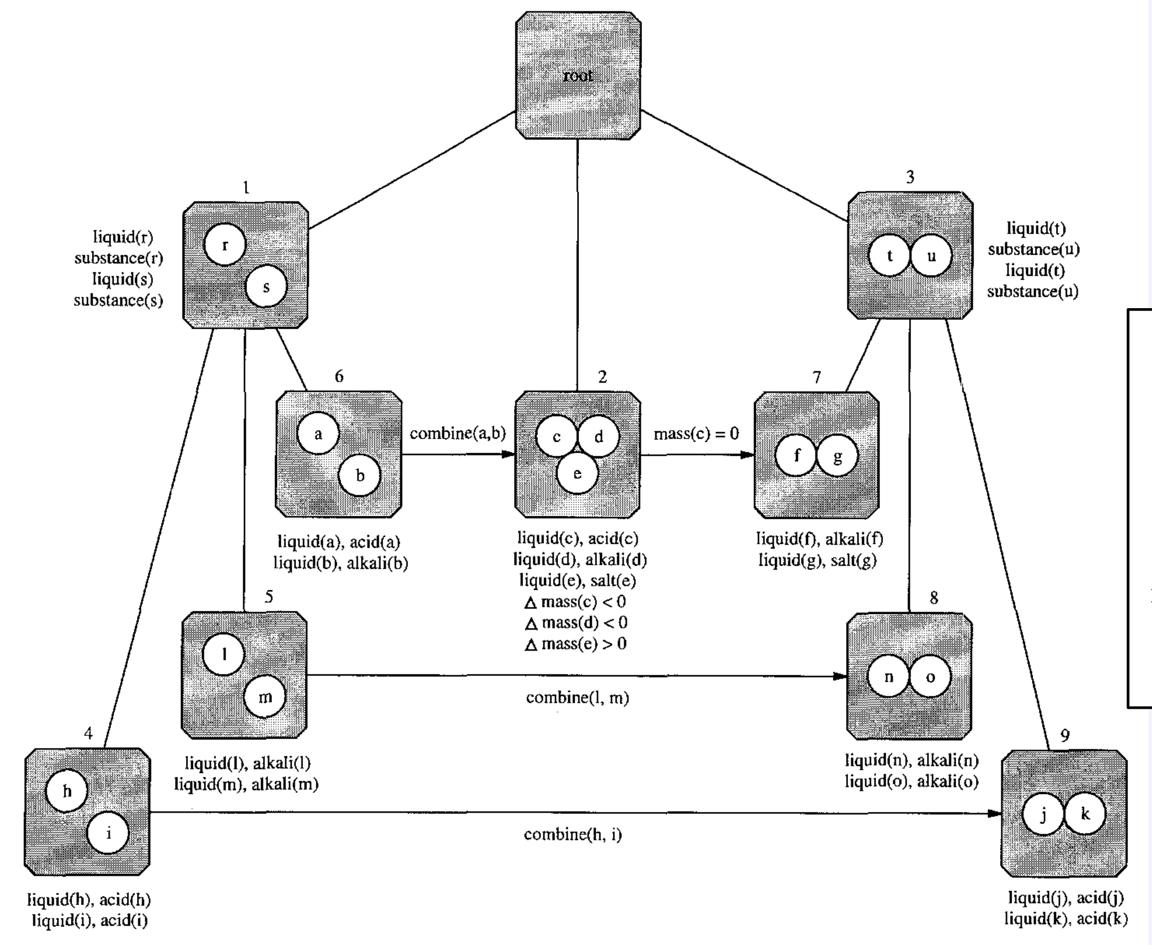
Nordhausen and Langley (1993) reported IDS, an integrated discovery system that:

- Created a taxonomy from observed qualitative states
  - E.g., *HCl and NaOH decrease when in contact, NaCl increases*
- Induced qualitative laws about temporal relations among states
  - E.g., *HCl and NaOH continue to change until one is consumed*
- Found numeric relations both within and between these states
  - E.g., *final NaCL amount function of initial HCl, NaOH amounts*

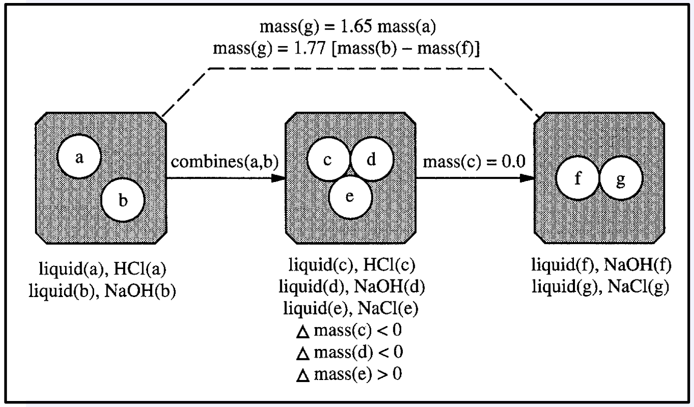
Each layer of description provided ***context*** for later discoveries.

IDS rediscovered laws about chemical reactions, Black's heat law, and conservation of momentum.

IDS Results for Chemistry



Quantitative laws



Taxonomy and qualitative laws

A Robot Scientist for Electrochemistry

Zytkow et al. (1990) reported an integrated system for discovery in electrochemistry that:

- Designed and ran experiments with a portable laboratory
- Used the Fahrenheit system to induce numeric laws
  - Including characterizations of maxima and minima
- Tested these hypothesized laws with further experiments

The system found novel results for both low ion concentrations and repeatability of peaks.

This was the first example of a robotic scientist that supported ***closed-loop scientific discovery***.

A Robot Scientist for Cell Biology

King et al. (2009) have constructed an integrated system for biological discovery that:

- Designs *auxotrophic growth studies* with yeast gene knockouts
- Runs these experiments using a *robotic manipulator*
- *Measures growth rates* for each experimental condition
- *Revises its causal model* for how genes influence metabolism

This ***closes the loop*** between experiment design, data collection, and model construction in biology.

The system found improved models of metabolic regulation in yeast.

Closing Remarks

Scientific Discovery as Heuristic Search

Scientific discovery does not involve any mystical or irrational elements; we can study and even partially automate it.

Our explanation of this fascinating set of mechanisms relies on:

- Heuristic search through a space of laws or models
- Using operators for generating structures and parameters
- Guiding search by data and by knowledge about the domain

Systems discover laws and models stated in the formalisms and concepts familiar to scientists.

This paradigm has already started to aid the scientific enterprise, and its importance will only grow with time.

Observations on Scientific Discovery

Research on scientific discovery offers some important lessons:

- Science adopts *explicit formalisms* for theories and models that are communicable to others.
- Scientific research is not entirely data driven; it often uses *existing knowledge* to aid the discovery process.
- Data is not the sole driver of discovery; science is a *closed loop* of model revision and data collection.
- Science is concerned with more than prediction; mature fields insist that observations be *explained* in deeper terms.
- Scientific insights do not require large amounts of data; in many fields, one must work with *sparse samples*.

We need less work on large data sets and more work on scaling to *complex models* and to *large spaces of models*.

Further Observations

Computational scientific discovery is not a new field, with decades of work dating back to the 1970s.

- There was originally great resistance to the idea that computers might discovery scientific laws and models.
- Computers are not number crunchers; they are *general symbol processors* that can encode *any* scientific content.
- Researchers have addressed a diverse set of discovery tasks, not all of them involving *induction from data*.
- Philosophers of science, cognitive psychologists, and artificial intelligence researchers all played roles in the movement.

Research in this tradition continues, but the number of active groups has been modest until recently.

Myths about Computing in Science

We should also debunk three myths about computing in science:

- Computing is changing the basic nature of scientific research.
  - No. Science has *always been a computational endeavor*, and digital computers does not alter its basic operation.
- Traditional science stood on two legs – theory and observation – and computing offers two more – simulation and data analysis.
  - No. *Every facet of science is computational*, and we can develop digital aids to make them each more efficient and effective.
- Computer-aided science is best pursued with domain-specific tools.
  - No. There are *general principles* of science that apply to many fields, and we can encode them in programming abstractions.

We need less rhetoric on how ‘computers will change everything’ and more work on how to aid the current scientific process.

Publications on Computational Scientific Discovery

Dzeroski, S., Langley, P., & Todorovski, L. (2007). Computational discovery of scientific knowledge. In S. Dzeroski & L. Todorovski (Eds.), *Computational discovery of scientific knowledge*. Berlin: Springer.

Langley, P. (1981). Data-driven discovery of physical laws. *Cognitive Science*, *5*, 31–54.

Langley, P. (2000). The computational support of scientific discovery. *International Journal of Human-Computer Studies*, *53*, 393–410.

Langley, P., Simon, H. A., Bradshaw, G. L., & Zytkow, J. M. (1987). *Scientific discovery: Computational explorations of the creative processes*. Cambridge, MA: MIT Press.

Langley, P., & Zytkow, J. M. (1989). Data-driven approaches to empirical discovery. *Artificial Intelligence*, *40*, 283–312.

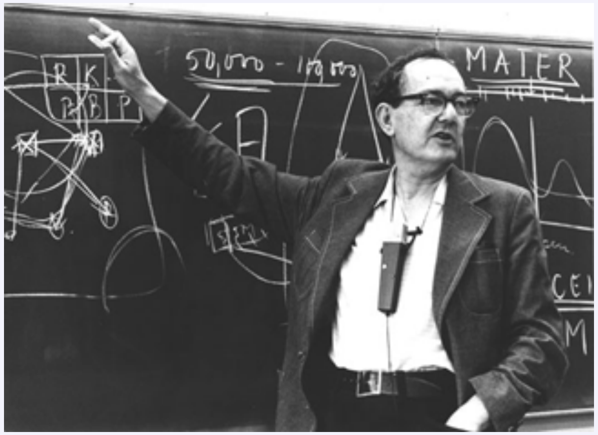
Lindsay, R., Buchanan, B. G., Feigenbaum, E. A., & Lederberg, J. (1980). *Applications of artificial intelligence for organic chemistry: The Dendral Project*. New York: McGraw.

Shrager, J., & Langley, P. (Eds.) (1990). *Computational Models of Scientific Discovery and Theory Formation*. San Francisco: Morgan Kaufmann.

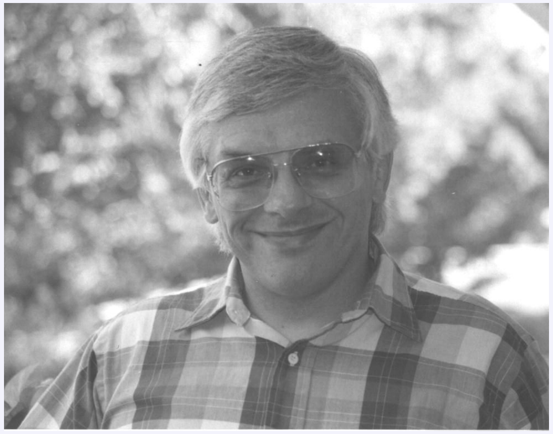
Simon, H. A. (1966). Scientific discovery and the psychology of problem solving. In R. G. Colodny (Ed.), *Mind and cosmos*. University of Pittsburgh Press: Pittsburgh, PA.

Valdés-Pérez, R. E. (1996). Computer science research on scientific discovery. *Knowledge Engineering Review*, *11*, 57–66

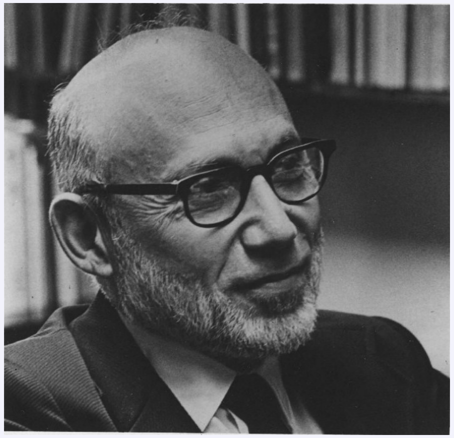
Founders of the Movement



Herbert A. Simon
(1916 – 2001)



Jan M. Zytkow
(1945 – 2001)



Joshua Lederberg
(1925 – 2008)

Edward A. Feigenbaum
(1936 – present)