# 8

# A Further Note on Inductive Generalization

G. D. Plotkin
Department of Machine Intelligence
University of Edinburgh

**Abstract**

In this paper, we develop the algorithm, given in Plotkin (1970), for finding the least generalization of two clauses, into a theory of inductive generalization.

The types of hypothesis which can be formed are very simple. They all have the form: $(x)Px \supset Qx$.

We have been guided by ideas from the philosophy of science, following Buchanan (1966). There is no search for infallible methods of generating true hypotheses. Instead we define (in terms of first-order predicate calculus) the notions of *data* and *evidence* for the data. Next, some formal criteria are set up for a sentence to be a *descriptive hypothesis* which is a *good explanation* of the data, given the evidence. We can then look for the best such hypothesis.

Although this problem is insoluble in general, some soluble subcases can be distinguished. We programmed one of these and tried some examples.

## INTRODUCTION

Suggesting inductive hypotheses is one of the fundamental problems of artificial intelligence and has been attempted in such domains as finite-state machines, context-free grammars, number sequences, propositional logic, and the predicate calculus. We are concerned with the last of these.

The work presented here originated with a suggestion of R. J. Popplestone that since unification is useful in automatic deduction by the resolution method, its dual might prove helpful for induction. The dual of the most general unifier of two literals is called their least general generalization; it is a literal of which they are both instances and is itself an instance of any other such literal, for example, $P(x, f(y))$ is the least general generalization of $P(a, f(c))$ and $P(b, f(b))$. If the latter two literals form part of some data to be explained, then $P(x, f(y))$ is the least general inductive hypothesis that explains them.

This resulted in a mathematical investigation of generalization of literals and later of clauses, for which subsumption played the role of generalization (Plotkin 1970). Given some phenomena we wish to find an inductive hypothesis from which the phenomena can be deduced as a logical consequence. We restricted attention to a weak form of deduction, namely subsumption. This mimics the usual notion of generalization, so that the hypotheses are mere inductive generalizations rather than fully-fledged scientific theories from which the phenomena may be deduced by complex chains of reasoning. This led us to consider the philosophical background of inductive generalization, where we were strongly influenced by Buchanan (1966).

The paper divides into three parts: philosophy, mathematics, and computation.

The philosophy section contains a general discussion of logics of scientific discovery, which we later specialize to a logic for suggesting hypotheses which are inductive generalizations of the phenomena.

The mathematics section discusses properties of least generalizations for literals and clauses, and goes on to consider the possibility of algorithms for generating inductive hypotheses to explain the phenomena. There is no such algorithm in general, because of a difficulty in checking consistency, but fortunately there are interesting solvable special cases.

Finally, we describe a program for performing inductive generalization, and illustrate its performance in characterizing the description of wins in 'noughts-and-crosses' and that of family relationships.

## PHILOSOPHY

Consider:

This crow is black.
That crow is black.
All crows are black.

This is the simplest possible inductive argument. It is an empirical inductive generalization. If we hope to have a robot capable of forming concepts or drawing inductive conclusions in some formal language, we must at the very least be able to tell it how to form such generalizations.

We aim to give a formal model of a general inductive generalization problem and then look for algorithms for its solution. As a guideline for this, we shall use the philosophy of science. Now, unless our robot is destined to be an automatic scientist, it is not obvious that the problems of hypothesis formation in science are the same as the ones he will have in his environment. There is, however, a parallel between the concerns of the scientist, with his sophisticated view of the world, and those of the robot, with its rather primitive view.

Presumably, the robot will be able to see, handle, and touch things, and so will have an observational vocabulary to refer to these sensory experiences.

Thus it is blessed with an ur-ontology. (I am indebted to Bruce Anderson and Pat Hayes for the prefix ur.) Eventually, if it is free of artificial restrictions, it will come across some phenomenon it does not understand. Or, to make sense of its myriad flow of impressions, it will want to classify its experience, if only for efficiency's sake. It will then need to start its ur-physics; its methodology must surely be found in an ur-philosophy of its ur-science.

More prosaically: it will be necessary to form generalizations about seen objects (Barrow and Popplestone 1971) and to learn, perhaps by experimentation, how objects, or objects and actions, are causally connected (Hayes 1971). It is not difficult to think of other tasks that current robots are asked to perform, which could do with some inductive capability.

Our second assumption is that all concepts are to be formed in the first-order predicate calculus. This is done, primarily, for technical convenience. It is accepted that the theories of science are amenable to the axiomatic method. Carnap (1967) is an impressive exponent of this view. There is, however, a possibility that some amount of modality should be introduced to deal with causality and tense. We do not expect knowledge logics to enter the picture since the hypotheses refer to the world rather than the robot's relation to the world.

### Logics for discovering hypotheses

This paper is an attempt to put into practice some of the views of Buchanan (1966). Where we have not introduced distortions of our own, justification for our philosophical assertions can be found in his work.

Now, we want to find a logic of discovery for performing a particular job; namely, through its use one must be able to find empirical generalizations of facts. We start from an analysis of logics of discovery of hypotheses, and then gradually proceed to the kind we have in mind.

First, we make some disclaimers. We do not mean by 'logic of discovery' some infallible way of suggesting scientific truths, by means of which it will always be possible, in a fixed, finite number of steps, to find a true general law explaining some given phenomena. This was the unrealizable dream of Bacon, Descartes and, to some extent, Mill. We know, however, that there is no truth-preserving way of passing from the particular to the general. So the word 'logic' refers here to a rational corpus of methods for finding and evaluating certain propositions. In general, we use it to refer to considerations which are independent of interpretation, that is, considerations holding in all possible worlds.

Neither are we concerned with the psychological connotations of 'discovery'. Our robot will not be required to dream or have inspirations. The robot will be a rational man. Finally, we ignore the element of time. The notion of co-discovery or precedence in making a discovery will have no meaning.

Four questions arise about logics for discovery of hypotheses $H$, given

knowledge $k$, which are required to do some job of explanation $J$.

H1 Is $H$ justified given $k$?

H2 Are there methods for justifying $H$, given $k$?

H3 What are conditions for an $H$ to do $J$ in a reasonable, interesting way, given $k$?

H4 Are there methods for suggesting a (most) reasonable, interesting $H$ to do $J$, given $k$?

There is an analogous problem of discovering theorems $T$ from a system of axioms $Ax$, which is rather more familiar and better investigated.

T1 Does $T$ follow from $Ax$?

T2 Can we show that $T$ follows from $Ax$?

T3 Under what conditions is $T$ an interesting, possible theorem in the system $Ax$?

T4 Is there a way to generate (most) interesting possible theorems?

All the questions H1–T4 are informal ones to which one may seek to give convincing formal answers.

It is agreed (Kreisel 1967) that T1 is settled for first-order logics by Tarski's definition of truth. T1 has not been settled for higher-order logics. As regards T2, the completeness theorem gives a semi-effective way of testing consequence. Work has hardly started (Kowalski 1970, Kreisel 1968) on efficient ways. The state of progress in T3 and T4 is minimal. Lee (1967) makes some suggestions and Polyà (1957) gives many informal ideas. On the whole, the only reasonably successful method for discovering theorems is to have an inspiration.

Evidently an answer to T2 gives one to T1. Similarly, one to T4 gives one to T3. T1 and T3 are independent. Not all theorems are interesting. Nor is it necessary that all suggested possibilities will be theorems, although there may be practical conditions which ensure that all interesting possible theorems are actually theorems. We expect the degree of interest will be specified by a quasi-ordering '$T1 \prec T2$' will be read as '$T1$ is more interesting than $T2$'. In general there will be many most interesting ones. The definition of $\prec$ will perhaps be relative to some question about $Ax$.

The questions H1 to H4 have even less definite answers. We give a brief account of the opinions of the two main schools of thought: the Popperians and the Carnapians (Carnap 1950, Popper 1959).

Carnap holds that the only concern of inductive logic is H1 and of deductive logic is T1. We will only use the term 'inductive logic' to refer to an answer to H1. To answer H1 for some particular $H$ and $k$, Carnap would calculate $c(H, k)$, the confirmation of $H$ given $k$. The confirmation function, $c$, has been taken to be the logical probability that $H$ is true given $k$, or as being the odds a rational man would lay that $H$ was true, given $k$. Carnap only defined $c$ for the monadic predicate calculus. Hintikka (1965) extended this to the predicate calculus in general. Unless we specify otherwise, our remarks on Carnapians refer only to the monadic case.

Thus, for Carnap, $H$ is justified to a degree $c(H, k)$ given $k$. If one had to choose between alternative hypotheses one would prefer the more highly-confirmed one. One can also define (Hintikka and Hilpinen 1966) a notion of acceptance; that is, a predicate $Ac(H, k)$ in terms of $c$ and certain properties of $k$. They do, however, use a different $c$ function from Carnap's.

As for H2, the definition of $c$ gives an effective means of calculation in the monadic case. In the general case there is no effective way of calculating $c$.

It was regarded as a paradox when sentences, consistent with $k$, which were meant to do the job of describing all the existing individuals, all received zero confirmation from any $k$ which consisted of a description of some individuals. We interpret this as showing that the criterion of reasonableness conflicted with that of theoretical justification. Hintikka's system avoids this dilemma. Aside from this, the Carnapians have little to say on H3 or H4, as is consistent with their general view on the role of inductive logic.

For Popperians, there can be no absolute justification of a hypothesis. There are instead a series of tests and competitions with other hypotheses. Their analysis is most concerned with universal laws, the form of scientific ones, for example: a law must be refutable and make many testable predictions. It is by making contradictory predictions that one of a competing pair may be eliminated.

Thus, much of their answer to H1 contains an answer to H2. For H3, it is suggested that simple hypotheses with a low *a priori* logical probability be sought.

Again, an answer to H2 gives one to H1, and similarly for H4 and H3. Not all justified hypotheses will be interesting, nor even, perhaps, reasonable. For example, we might wish it to be reasonable, but not a justification, that $H$ be beautiful. Again, although it is not necessary, it may be that all the reasonable, interesting hypotheses are justified.

Again, we expect the degree of interest to be specified by a quasi-ordering $\prec$. This will be defined relative to $k$ and $J$.

The following definitions seem reasonable in the light of the preceding discussion.

An *inductive logic* is an answer to H1.

A *practical inductive logic* is an answer to H1 and one to H2. One may predicate completeness, consistency, or efficiency of a practical inductive logic.

Answers to H3 and H4 constitute a *logic of suggestion* which may also be consistent, complete, and efficient.

Answers to H1–H4 constitute a *logic of discovery* provided $H$'s being most reasonable and interesting implies its being justified. Note that any set of answers to H1–H4 can be, trivially, made into a logic of discovery by changing the requirements for reasonableness to include also those for being justified, and changing the method by using the answer to H2 to check that any $H$ found using the answer to H2 is justified.

We shall give a logic of suggestion for finding inductive generalizations. This will be both complete and consistent in various subcases. For the monadic subcase, we shall give a practical complete and consistent inductive logic, relative to which we obtain a logic of discovery.

**Facts, evidence, theories and the nature of explanation**

Now we examine $k$ and $J$ more closely. The robot's knowledge, $k$, will be $Th \wedge \bigwedge_{i=1}^{n} (e_i \wedge f_i)$ ($\wedge$ denotes logical conjunction), where

K1 The $f_i$ are statements of facts or events.

K2 Each $e_i$ is a statement of background information or evidence relevant to $f_i (i=1, n)$.

K3 $Th$ is a statement of facts and theories relevant to all the $e_i$.

The job, $J$, that $H$ must do is to be a general law which is a potential explanation of each $f_i$ given the corresponding $e_i$ and $Th$. The definition of potential explanation is, following Buchanan, a modification of Hempel's explication of explanation for the purposes of a logic of suggestion.

$H$ is a *potential explanation* of the $f_i$ given $e_i$ and $Th$ iff

E1 $\vdash e_i \wedge Th \wedge H \rightarrow f_i \,.\, (i=1, n)$.

E2 One of the statements in $e_i$ or $Th$ or $H$ must be a general law.

E3 At least one of $e_i$ or $Th$ or $H$ must be empirical (i.e., non-logical and non-theoretical) in nature.

E4 It must not be the case that
$$\vdash e_i \wedge Th \rightarrow f_i \,.\, (i=1, n).$$

E5 $H \wedge Th \wedge \bigwedge_{i=1}^{n} (e_i \wedge f_i)$ is consistent.

Requirement E4 ensures that $H$ enters non-vacuously into the explanation and E5 replaces Hempel's requirement of truth with one of consistency, and this is why we write 'potential'.

Before trying to characterize the form of $Th$, and the $e_i$ and $f_i$, and the type of general law that interests us, let us see how varying $f_i$, $e_i$ and $Th$ can lead us through the whole spectrum of scientific hypotheses.

$Th$ might consist of some axioms of applied science including axioms relating observations to the theory of Euclidean space, $f$ the fact that an apple fell, $e$ the fact that it was flung into the air and $H$ Newton's theory of gravitation.

When a whole theory has broken down, the $e_i$ consist of all previous observational experience in the field plus some anomalies, $Th$ is practically empty, and $H$ will contain many new terms and relations. On the sidelines will be the discredited theory.

$H$ might be an extension, by analogy, of $Th$ designed to carry $Th$'s previous successes into the new field described by the $e_i$ and $f_i$.

$Th$ could be the body of normal science, the $e_i$ and $f_i$ an account of some new phenomena found in some experiments. $H$ would then be an explanation

using the terms of normal science. As an example, descriptions of the phenomenon of superconductivity, framed in the quantum theory, fit this mould.

#### A particular form for the facts and evidence

Let us return to the crows. As an aside, we remark that we would have liked to treat

'This crow is black'

as meaning

$$Crow(crow1) \wedge (Crow(crow1) => Black(crow1))$$

where $=>$ is a modal connective which could denote

causes, or entails as an essential property, or is a good reason for.

However, we do not know of any appropriate modal logic, and so did not do this. Instead, we shall incorporate crows in the above scheme by taking $e_1 = Black(crow1)$; $e_2 = Black(crow2)$; $f_1 = Crow(crow1)$; $f_2 = Crow(crow2)$ and $Th$ empty.

More generally we shall let $F = \{f_i | i = 1, n\}$ be a set of ground literals [for an account of our logical language, *see* Robinson (1965) and Plotkin (1970)], and

$$e_i = Ev(f_i)$$

be a conjunction of ground literals. We use ground literals to represent observations, thus formalizing K1. We do not formalize K2 since what constitutes relevance seems to depend on the subject domain. The only syntactic guide we can give is that $f_i$ and $e_i$ should have terms in common; perhaps the relation between literals of having terms in common should make the set of literals in $f_i$ and $e_i$ connected.

As for K3, we let $Th$ be a conjunction of ground literals. We did this since we could not obtain a good theory for any more complicated $Th$.

There is a group of conditions on the $e_i$ and $f_i$ for partial satisfaction of the E conditions.

First, all vocabulary in the $e_i$ and $f_i$ is to be observational: so E3 is satisfied. Second, no $f_i$ is to appear in the corresponding $e_i \wedge Th$. Thus E4 is satisfied. Finally we require that

$$Th \wedge \bigwedge_{i=1}^{n} (e_i \wedge f_i)$$

is consistent so that it is possible for $H$ to satisfy E5.

#### Generalization as a particular form of explanation

To finish specifying the job that $H$ must do, we say what it is for $H$ to be a general law, which is, given $Th$ and $e_i$, a generalization of $f_i$. $(i = 1, n)$.

As it is a scientific law, $H$ is to be a universal sentence. We allow $H$ to be ground, as a degenerate case.

We say that $H$ is a generalization of $f_i$ given $Th$ and $e_i$ if, roughly, in the proof of $f_i$ from the $e_i$, $Th$, and $H$, the only inferences which do not involve $e_i$ or $Th$ are those in which we instantiate some sentence.

This is to be made more precise in the context of a binary resolution system. $H$ is regarded as a set of clauses, $Th$ as a set of unit clauses and $e_i \rightarrow f_i$ as a clause. It then follows from the subsumption theorem (Kowalski 1970) that E1 holds iff one can derive a clause $D_i$ by resolution from $H \cup Th$ which subsumes $e_i \rightarrow f_i$. We say that $H$ is a *generalization* of $e_i \rightarrow f_i$ given $Th$ iff all the resolutions in the derivation of $D_i$ involve a clause from $Th$.

We need some definitions. If $C$ and $D$ are clauses then $C \leqslant D$ means $C$ subsumes $D$.

Let $L$ be a literal. $\bar{L}$ is that literal with the same atom as $L$, but opposite sign.

Let $Th$ be a conjunction of literals. We set

$$\overline{Th} = \{\bar{L} | L \text{ appears in } Th\}.$$

If $L$ and $M$ are literals, by

$L \leqslant M(Th)$ we mean

$\{L\} \leqslant \{M\} \cup \overline{Th}$.

(Read this as $L$ is more general than $M$, relative to $Th$. The next two definitions should be given similar readings.)

If $C$ and $D$ are clauses, by

$C \leqslant D(Th)$ we mean

$C \leqslant D \cup \overline{Th}$.

Let $H_1$ and $H_2$ be sets of clauses.

By $H_1 \leqslant H_2(Th)$ we mean for every $D$ in $H_2$ there is a $C$ in $H_1$ such that $C \leqslant D(Th)$.

These are all quasi-orderings when $Th$ is ground.

Let $H_0 = \{e_i \rightarrow f_i \text{ (considered as a clause)} \mid 1 \leqslant i \leqslant n\}$

$= \{C_i | 1 \leqslant i \leqslant n\}$, say.

*Lemma* 1

$H$ is a generalization of $e_i \rightarrow f_i$ given $Th (i = 1, n)$ iff $H \leqslant H_0(Th)$.

We do not give a proof of this which depends on the fact that E4 holds. The same result goes through for M-clash resolution. Whenever we do not give proofs in this paper, they will appear in Plotkin's forthcoming PhD thesis.

Note that the definition of generalizations ensures that no new function or predicate symbols can appear in any generalization, $H$.

To sum up, a job $J$ is specified by a set

$F = \{f_i | i = 1, n\}$ of ground literals

and a function $Ev$, such that

$$e_i = Ev(f_i)$$

is a conjunction of ground literals, and a conjunction $Th$ of ground literals.

$F$, $Ev$ and $Th$ are subject to

J1 $f_i$ does not appear in $Th (i = 1, n)$.

J2 $Th \wedge \bigwedge_{i=1}^{n} (e_i \wedge f_i)$ is consistent.

$H$ does $J$ iff

J3 $H \leqslant H_0(Th)$.

J4 $H \wedge Th \wedge \bigwedge_{i=1}^{n} (e_i \wedge f_i)$ is consistent.

**A particular choice of the 'more interesting than' relation**

We now define the quasi-ordering $\prec$ which judges whether or not one hypothesis is more interesting and more reasonable than another.

*Definition.* Let *complexity* $(H)$ = the number of clauses in $H$.

Let *power* $(C)$ = the cardinality of $\{f_i \text{ in } F | C \leqslant C_i(Th)\}$.

Let *power* $(H) = \sum_{C \in H}$ *power* $(C)$.

Then $H_1 \prec H_2$ iff

*complexity* $(H_1) <$ *complexity* $(H_2)$

or the complexities are equal and *power* $(H_1) >$ *power* $(H_2)$

or the powers and complexities are equal and $H_2 \leqslant H_1(Th)$.

The motivation for the first two clauses of the definition of $\prec$ is as follows: Let us define $\prec_{cp}$ by:

$H_1 \prec_{cp} H_2$ iff

*complexity* $(H_1) <$ *complexity* $(H_2)$

or the complexities are equal and power $(H_1) \geqslant$ power $(H_2)$.

Then we say that $\prec'$ is a lexicographic extension of $\prec_{cp}$ if it can be defined by the form

$H_1 \prec' H_2$ iff

$H_1 \prec_{cp} H_2$

or the complexities and powers are equal and $H_1 \prec'' H_2$

($\prec''$ being an arbitrary quasi-ordering).

Then any ordering $\prec'$ which extends $\prec_{cp}$ lexicographically can be shown to be a natural generalization of a measure of simplicity used by Quine (1955) for disjunctive normal forms of propositional functions. It is possible to reduce the problem of finding simplest equivalent disjunctive normal forms of a given proposition to an instance of problem $J$ (*see* below) with $\prec$ replaced by $\prec'$.

The third clause in the definition of $\prec$ makes a solution less general, in opposition to the others. This is both a conservative policy and a confession of ignorance. If we do not know which to choose, then choosing the least general gives a survey of all the other possibilities.

Our choice of $\prec$ is not central to this paper, unlike the other restrictions that we have made. We are perfectly willing to consider other ones, and indeed prove the later unsolvability theorem for a wide range of such choices.

This completes our answer to H3.

**The problem of finding most interesting explanations**

To answer H4, we must find a method of finding a set of clauses, $H$, such that:

(1) $H \leqslant H_0(Th)$.

(2) $H \wedge Th \wedge \bigwedge_{i=1}^{n} (e_i \wedge f_i)$ is consistent.

(3) $H$ is minimal, with respect to $\prec$, amongst those sets of clauses satisfying (1) and (2).

We shall call the problem of finding such an $H$, problem $J$.

### Compatibility between our logic of suggestion and various inductive logics

We have described the structure of $k$, the robot's knowledge, and $J$, the job of explanation, to be performed in the most interesting possible way by a hypothesis $H$. This is formalized by the conditions of problem $J$. In the next section we give a logic of suggestion for various subcases of problem $J$.

There are, in the literature, a number of inductive logics giving criteria for judging hypotheses. To obtain a logic of discovery, relative to some inductive logic, we would need to show first that the inductive logic can be made practical. Secondly, it must be the case that the hypotheses generated by our logic of suggestion are always justifiable by the criteria of that inductive logic. We shall do all this for the monadic case.

Here is the practical inductive logic. The monadic case arises when $Th$ and the $f_i$ and $e_i$ contain only monadic predicate symbols and no function symbols other than constant ones. Hintikka and Hilpinen define a computable binary predicate of monadic sentences, $Ac$.

'$Ac(h, e)$' is to be read as '$h$ is acceptable on the basis of $e$'.

Their system has a parameter $\varepsilon$ which we assume to be fixed, for the purposes of this discussion.

$Ac$ satisfies:

AC1 If $Ac(h_1, e)$ and ... and $Ac(h_k, e)$ and if $\vdash (h \wedge \ldots \wedge h_k) \rightarrow h_0$, then $Ac(h_0, e)$.

AC2 The set $\{h_i | Ac(h_i, e)\}$ is logically consistent.

*Theorem* 1

For $n \geqslant n_0$ ($n$ is the number of $f_i$s) and any solution, $H$, to a monadic problem $J$,

$$Ac(H, Th \wedge \bigwedge_{i=1}^{n} (f_i \wedge e_i))$$

is true. The constant, $n_0$, is computable from the number of predicate symbols.

Acceptance conditions have been studied for languages other than the monadic one (Kemeny 1953, Putnam 1963) but we have not studied their interactions with our system.

## MATHEMATICS: SUBSUMPTION, LEAST GENERALIZATION AND A LOGIC OF SUGGESTION

### Least generalization for literals

We give some theory for the 'less general relative to $Th$' relation. The theory

follows essentially the plan of Plotkin (1970) to which we refer for results and notation.

We write $L \sim M(Th)$ when $L \leqslant M(Th)$ and $M \leqslant L(Th)$. This is an equivalence relation. We define similarly $C \sim D(Th)$ and $H_1 \sim H_2(Th)$.

*Lemma* 2

$L \sim M(Th)$ iff $\{L\}$ and $\{M\}$ are alphabetic variants or else both subsume $\overline{Th}$.

*Proof.* Sufficiency.

If $\{L\} \leqslant \overline{Th}$ then

$\{M\} \leqslant \{L\} \cup \overline{Th}$ (by assumption)

$\leqslant \overline{Th} \cup \overline{Th} = \overline{Th}$.

Similarly $\{M\} \leqslant \overline{Th} \rightarrow \{L\} \leqslant \overline{Th}$.

Otherwise $\{L\} \leqslant \{M\} \cup \overline{Th}$

implies $L \leqslant M$ and similarly $M \leqslant L$; and so $L$ and $M$ are alphabetic variants.

Necessity.

Obvious.

*Definition.* $L$ is a least general generalization of $M$ and $N$ relative to $\overline{Th}$ iff

(1) $L \leqslant M(Th)$ and $L \leqslant N(Th)$

(2) If $L' \leqslant M(Th)$ and $L' \leqslant N(Th)$ then $L' \leqslant L(Th)$.

*Lemma* 3

Let $L = inf\{M, N\}$ be the least general generalization of $M$ and $N$ as given in Plotkin (1970). Then $L$ is a least general generalization of $M$ and $N$ relative to $Th$.

*Proof.* (1) is easy.

(2) Let $L' \leqslant M, N(Th)$.

If $\{L'\} \leqslant \overline{Th}$ then $L' \leqslant L(Th)$.

Otherwise $L' \leqslant M, N$ and so $L' \leqslant L$.

**Reduction and least generalization for clauses**

*Definition.* $C$ is *reduced* relative to $Th$ iff $D \subseteq C$, $D \sim C(Th)$ implies $C = D$.

*Lemma* 4

If $C \sim D(Th)$ and $C$ and $D$ are reduced relative to $Th$ then they are alphabetic variants. The following algorithm gives a reduced subset $E$, relative to $Th$, of $C$ such that $E \sim C(Th)$.

(1) Set $E$ to $C$.

(2) Find an $L$ in $C$ and a substitution $\sigma$ so that $E\sigma \subseteq (E \setminus \{L\}) \cup \overline{Th}$, else stop.

(3) Change $E$ to $E\sigma$ and go to 2.

*Proof.* There is a $\mu$ so that

$C\mu \subseteq D \cup \overline{Th}$.

So $C = C_1 \cup C_2$ where $C_1\mu \subseteq D$, $C_2\mu \subseteq \overline{Th}$.

Hence $C \sim C_1(Th)$ and since $C$ is reduced relative to $Th$, $C_2 = \phi$.

Thus $C\mu \subseteq D$.

Similarly, $Dv \subseteq C$ for some $v$. So $C \sim D$. But being reduced relative to $Th$ implies being reduced. Hence by theorem 2 in Plotkin (1970), $C$ and $D$ are alphabetic variants.

The proof of the second part is an exact analogue of the proof of the corresponding part of theorem 2 in Plotkin (1970).

*Definition.* $C$ is a least general generalization of $D$ and $E$ relative to $Th$ iff

(1) $C \leqslant D(Th)$ and $C \leqslant E(Th)$

(2) $C' \leqslant D(Th)$ and $C' \leqslant E(Th)$ implies $C' \leqslant C(Th)$.

*Lemma* 5

Let $C = inf\{D \cup \overline{Th}, E \cup \overline{Th}\}$ be the result of applying the algorithm of theorem 3 of Plotkin (1970) to $D \cup \overline{Th}$ and $E \cup \overline{Th}$. Then, $C$ is the least general generalization of $D$ and $E$ relative to $Th$.

*Proof.*

(1) $C \leqslant D \cup \overline{Th} \Rightarrow C \leqslant D(Th)$.
$C \leqslant E \cup \overline{Th} \Rightarrow C \leqslant E(Th)$.

(2) $C' \leqslant D(Th)$ and $C' \leqslant E(Th)$
$\Rightarrow C' \leqslant D \cup \overline{Th}$, $C' \leqslant E \cup \overline{Th}$
$\Rightarrow C' \leqslant inf\{D \cup \overline{Th}, E \cup \overline{Th}\} = C$.

It follows that any finite non-empty set of clauses $H$ has a least general generalization relative to $Th$ which is unique to within equivalence relative to $Th$. By *inf* $H$ we mean one such.

**Reduction for sets of clauses**

A set, $H$, of clauses is *reduced* relative to $Th$ iff $H' \subseteq H$, $H' \sim H(Th)$ implies $H' = H$.

*Lemma* 6

If $H' \sim H(Th)$ and $H'$ and $H$ are reduced relative to $Th$ then there is a bijection $\theta : H' \to H$ such that $\theta(C) \sim C(Th) (C \in H')$.

The following algorithm gives a reduced, relative to $Th$, subset $H'$ of $H$ such that $H' \sim H(Th)$.

(1) Set $H'$ equal to $H$.

(2) Find a clause $C$ in $H'$ such that $H' \setminus \{C\} \leqslant \{C\}(Th)$. If this is impossible, stop.

(3) Change $H'$ to $H' \setminus \{C\}$ and go to (1).

*Proof.* Choose $\theta : H' \to H$ so that $\theta(C) \leqslant C(Th) (C \in H')$.
This is possible since $H \leqslant H'(Th)$.
Similarly, choose a $\theta' : H \to H'$ so that $\theta'(C) \leqslant C(Th)$ $(C \in H)$.
Then for $C$ in $H'$, $\theta'(\theta(C)) \leqslant \theta(C) \leqslant C(Th)$. But $H'$ is reduced, so $\theta'(\theta(C)) = C$. $(C \in H')$.

Similarly, $\theta(\theta'(C)) = C (C \in H)$.
Hence $\theta$ is a bijection with inverse $\theta'$ and $C \sim \theta(C)(Th)$.

Evidently the algorithm terminates with some $H' \subseteq H$. If $H'$ is not reduced there is a proper subset $H''$ of $H'$ so that $H' \sim H''(Th)$. Let $C$ be in $H' \setminus H''$.

Then $H'\backslash\{C\} \leqslant \{C\}(Th)$ and the algorithm could not have stopped. This contradiction completes the proof.

**Reformulation of conditions for a best explanation**

Let us repeat the statement of the formal problem $J$. We must find a set of clauses, $H$ such that:

(1) $H \leqslant H_0(Th)$.

(2) $H \wedge Th \wedge \bigwedge_{i=1}^{n} (e_i \wedge f_i)$ is consistent.

(3) $H$ is minimal, with respect to $\prec$, amongst those sets of clauses satisfying (1) and (2).

It will turn out that every solution $H$ will be found, to within equivalence relative to $Th$, in a lattice $\mathscr{L}(H_0)$ of least general generalizations relative to $Th$.

The following definition applies to an arbitrary set of clauses $H$.

$\mathscr{L}(H)$ is a lattice of clauses with carrier $\{inf\, H' | H' \subseteq H, H' \neq \phi\}$, and lattice operations given by

$\inf H_1 \sqcap \inf H_2 = \inf H_1 \cap H_2$ and

$\inf H_1 \sqcup \inf H_2 = \inf H_1 \cup H_2$.

The lattice may also be viewed as being generated by $H_0$ using the above $\sqcup$. Let us illustrate with $H_0 = \{C_1, C_2, C_3\}$.

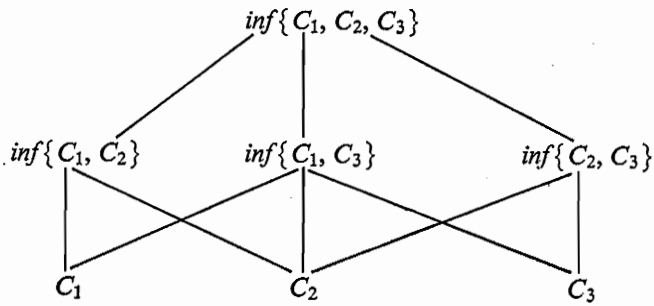



Figure 1

We can now identify the solutions to problem $J$.

*Theorem* 2

$H$ is a solution iff it is equivalent relative to $Th$ to an $H'$ satisfying:

(1) $H' \subseteq \mathscr{L}(H_0)$.

(2) $H' \leqslant H_0(Th)$.

(3) $H' \wedge Th \wedge \bigwedge_{i=1}^{n} (e_i \wedge f_i)$ is consistent.

(4) $H'$ is minimal, w.r.t. $\prec_{cp}$, amongst those sets of clauses satisfying (1), (2) and (3).

Further, the problem has at least one solution.

Thus to determine all the solutions, we need only calculate the above set. $\mathscr{S}(H_0)$ is calculable, from the results of Plotkin (1970), $H \leqslant H'(Th)$ is an effective relation (it is really just subsumption) and $\prec_{cp}$ is also. However we do not know *prima facie* whether '$H \wedge Th \wedge \wedge (e_i \wedge f_i)$ is consistent' is decidable or not. In general consistency is not semi-decidable. However, because of the special way our $H$s arise, we might hope that this is an exception.

**General unsolvability and some solvable subcases, with examples**

Unfortunately we have

*Theorem* 3

There is no algorithm which will, given *Th* and the $e_i$ and $f_i$, produce a solution of problem *J*.

This theorem holds even if we restrict *Th* and $e_i$ and $f_i$ as follows.

(1) *Th* is empty.

(2) Only one predicate symbol, *P* say, occurs in the $f_i$.

(3) *P* does not occur in any of the $e_i$.

We can also change the definition of $\prec$ a little. Indeed the theorem holds for an arbitrary relation $\prec'$ such that

(1) $\prec'$ is a lexicographic extension of $\prec_{cp}$

(2) The new problem *J*, obtained by changing $\prec$ to $\prec'$ always has a solution.

To show that it is worthwhile going on to find solvable subcases we give a couple of examples. First, the last appearance of the crows.

Table 1 shows the $f_i$ and $e_i$.

Table 1

| $f$ | $e$ |
|---|---|
| $f_1 = Black(crow1)$ | $e_1 = Crow(crow1)$ |
| $f_2 = Black(crow2)$ | $e_2 = Crow(crow2)$ |

Also *Th* was empty. We have $H_0 = \{C_1, C_2\}$ where

$$C_1 = \overline{Crow}(crow1) \vee Black(crow1)$$

$$C_2 = \overline{Crow}(crow2) \vee Black(crow2)$$

Now, $C_3 = inf\{C_1, C_2\} = \overline{Crow}(x) \vee Black(x)$

Evidently $C_3 \wedge e_1 \wedge e_2 \wedge f_1 \wedge f_2$ is consistent and so $\{C_3\}$ is the only solution. We have induced 'All crows are black'.

Next, we give a less trivial example from Hunt, Marin, and Stone (1966). We must learn that all bears or large animals are dangerous. Our observational data consists of a description of various animals, both dangerous and non-dangerous, in terms of the attributes *Size*, *Animality*, and *Colour*. Again *Th* is empty.

Now, $H_0=\{C_i|1\leqslant i\leqslant 7\}$. The members of $\mathscr{I}(H_0)$ which are consistent with $\bigwedge_{i=1}^{7}(e_i\wedge f_i)$ are, apart from $C_1$ to $C_7$,

$$C_8=inf\{C_1, C_2\}$$
$$=\overline{Size}(x, s)\vee\overline{Colour}(x, black)\vee\overline{Animal}(x, bear)\vee Dangerous(x)$$

and

$$C_9=inf\{C_3, C_6, C_7\}$$
$$=\overline{Size}(x, large)\vee\overline{Colour}(x, c)\vee\overline{Animal}(x, a)\vee Dangerous(x).$$

The solution is $H=\{C_8, C_9, C_4, C_5\}$ and includes the following version of the generalization to be learnt:

'Anything that has a size and is a black bear is dangerous' and 'Anything that is a large coloured animal is dangerous'.

Notice that if we assume that all animals have a colour and all bears have a size, then this generalization is equivalent to the original one.

Table 2

| $f$ | $e$ |
|---|---|
| $f_1=Dangerous(animal1)$ | $e_1=Size(animal1, small)\ Colour(animal1, black)$ $Animal(animal1, bear)$ |
| $f_2=Dangerous(animal2)$ | $e_2=Size(animal2, medium)\ Colour(animal2, black)$ $Animal(animal2, bear)$ |
| $f_3=Dangerous(animal3)$ | $e_3=Size(animal3, large)\ Colour(animal3, brown)$ $Animal(animal3, dog)$ |
| $f_4=\overline{Dangerous}(animal4)$ | $e_4=Size(animal4, small)\ Colour(animal4, black)$ $Animal(animal4, cat)$ |
| $f_5=\overline{Dangerous}(animal5)$ | $e_5=Size(animal5, medium)\ Colour(animal5, black)$ $Animal(animal5, horse)$ |
| $f_6=Dangerous(animal6)$ | $e_6=Size(animal6, large)\ Colour(animal6, black)$ $Animal(animal6, horse)$ |
| $f_7=Dangerous(animal7)$ | $e_7=Size(animal7, large)\ Colour(animal7, brown)$ $Animal(animal7, horse)$. |

Let us consider some special solvable subcases.

s1 If there are no function symbols other than constant ones in $Th\wedge\bigwedge(e_i\wedge f_i)$, then the same will be true of the members of $\mathscr{I}(H_0)$ and consistency of

$H\wedge Th\wedge\bigwedge(e_i\wedge f_i)$ where $H\subseteq\mathscr{I}(H_0)$, will always be decidable.

Notice that this gives the last link for a logic of discovery in the monadic case.

s2 Suppose the underlying predicate calculus has two sorts, situations and objects, and that there are situation constant symbols and variable

symbols, but no other function symbols which produce situations. The whole of our theory goes through as before, with only small alterations.

Then if,

(1) there are situation constant symbols $s_1() \ldots s_n()$ such that each predicate symbol $Q_j$ which appears in an $f_i$ or $e_i$ $(i=1, n)$ has a situation place which is filled with $s_i()$ when $Q_j$ appears in $e_i$ or $f_i$. $(i=1, n)$, and

(2) no predicate symbols in $Th$ have any situation places, it follows that

$H \wedge Th \wedge \bigwedge (e_i \wedge f_i)$ is consistent iff for all $i$, and $C \in H$,

$C \not\leqslant (\bar{e}_i \cup \{\bar{f}_i\})(Th)$ . $1 \leqslant i \leqslant n$, $H \subseteq \mathscr{I}(H_0)$.

This is decidable.

#### An algorithm for the solvable subcases

By the remarks made after theorem 2, we have now a consistent and complete logic of suggestion for the subcases s1 and s2. Furthermore the method of finding solutions is effective. We can now give an explicit algorithm. Consider the following 'derivation' rules.

(1) $H => (H \setminus \{C\}) \cup \{inf\{C, C_i\}\}$ (provided $C \in H$ and $C \not\leqslant C_i(Th)$ and $H \wedge inf\{C, C_i\} \wedge Th \wedge \bigwedge_{i=1}^{n} (e_i \wedge f_i)$ is consistent).

(2) $H => (H \setminus \{C\}) \cup \{\hat{C}\}$ (provided $C \in H$, $C$ is not reduced and $\hat{C}$ is a reduced form of $C$).

(3) $H => H \setminus \{C\}$ (provided $C \in H$ and $C$ is dependent on $H \setminus \{C\}$).

*Definition.* For any clause $C$, let $Explainset(C) = \{i \mid C \leqslant C_i\}$. Then $C$ is *dependent* on $H$ iff $Explainset(C) \subseteq \bigcup_{D \in H} Explainset(D)$.

These rules are effective provided that the consistency question is decidable.

It is possible to apply the derivation rules only a finite number of times to $H_0$. Thus one obtains a set $\{H \mid H_0 => H$, it is not possible to apply any rule to $H\}$. This is the set of *irredundant* sets of clauses. The solutions are, to within equivalence relative to $Th$, the sets of clauses in this set which are minimal with respect to $\prec_{cp}$. Finally all irredundant $H$s are reduced, and all their member clauses are reduced.

We look at some subcases where it is possible to put extra restrictions on the rules.

There is a common situation where $H$ can be split into two parts. Let us say that $Th, F, E$ form a *positive* problem $J$ if any predicate symbol that appears in $F$ appears, if at all, only positively in $Th$ or any $e_i$.

Let $H_0^+ = \{C_i \mid f_i$ is positive, $1 \leqslant i \leqslant n\}$

$H_0^- = \{C_i \mid f_i$ is negative, $1 \leqslant i \leqslant n\}$

Then any solution $H$ to problem $J$ can be written in the form $H = H^+ \cup H^-$ where $H^+ \subseteq \mathscr{I}(H_0^+)$, $H^- \subseteq \mathscr{I}(H_0^-)$ (or at least it can to within equivalence relative to $Th$).

One can define *negative* problems and get the same result.

For both positive and negative problems, one may require that rule 1 only be applied when $inf\{C, C_i\}$ is a member of $\mathscr{I}(H_0^+) \cup \mathscr{I}(H_0^-)$.

If $J$ is positive (similarly for negative) and falls under subcase S2, then in the test for consistency, we can insist that if $C$ is in $\mathscr{F}(H_0^+)$ then we only test that $C \not\leq (\bar{e}_i \cup \{\bar{f}_i\})$ for negative $f_i$. Again, if $C$ is in $\mathscr{F}(H_0^-)$ we need only try positive $f_i$.

#### A general algorithm using a limited consistency check

Finally, we mention an alternative to looking for consistent subcases, suggested by Meltzer (1970). We replace E5, by the requirement that a determined effort has been made (but failed) to prove that $H \wedge Th \wedge \wedge (e_i \wedge f_i)$ is inconsistent. Of course, this means that, in general, we shall not obtain a logic of discovery. We could still add on at the end a (perhaps computationally very expensive) test for consistency. This combination (which, for the appropriate subcases would be a logic of discovery) might be quite practical.

For our problem we decided to formalize the 'determined' test as $H \wedge Th \wedge \wedge (e_i \wedge f_i) \wedge Ax$ is $l$-consistent, where $Ax$ is an arbitrary set of sentences that the robot believes about its observational vocabulary and $l$-consistent means that a binary resolution theorem prover has not found a contradiction at level $l$. The theory goes through much as before, and theorem 2 holds when the evident changes have been made.

#### An example from group theory

We hand-simulated the method, with $l$ set equal to ten, for the problem that Meltzer tried. The facts are represented using a binary predicate symbol $E$, for equality, and a binary function symbol $f$ for multiplication. Thus $ab=cd$ is represented by two facts, $f_1 = E(f(a, b), f(c, d))$ and $f_2 = E(f(c, d), f(a, b))$ (sim. for $ab \neq cd$). The corresponding $e_i$ are empty. $Th$ was empty, but one took $Ax$ to be the axioms for equality. It can be shown that $E(t_1, t_2)$ is in the solution $H$ iff $E(t_2, t_1)$ is (sim. for $\bar{E}(t_1, t_2)$) so we present the input and output in the ordinary notation.

The facts given were:

$ee=e$
$ae=a$
$(aa)e=a(ae)$
$(ea)a=e$
$ea \neq e$
$aa \neq a$
$bc=cb$
$(bb)b=c$
$(bb)c \neq c$
$(bc)c \neq b$.

The solution was:

1 $xe=x$
2 $xa \neq x$

3 $(aa)e=a(ae)$
4 $(ea)a=e$
5 $xy=yx$
6 $(bb)c\neq c$
7 $(bc)c\neq b$
8 $(bb)b=c$

Thus we found the right-identity and commutative laws. When we added

$(ab)b=a(bb)$

the solution was as above except that 3 is replaced by $(xy)z=x(yz)$, the associative law. This is the same kind of result as Meltzer obtained.

## COMPUTATION

### The program

We have written a program for a logic of suggestion in the POP-2 programming language. The program assumes that the problem is positive, that it falls under the soluble case, s2, and that *Th* is empty. It calculates the $H^+$ part of an irredundant $H$. If run with signs reversed, it will calculate the $H^-$ part. A heuristic approximation to an irredundant $H$ is found, rather than the solution proper, so that the present, rather crude program should have reasonable running times. At the moment, these times vary from three to fifteen minutes on the ICL 4130 machine. We estimate that it would take up to four times as long to produce a list of all the irredundant $H$s.

The flow chart of the program is figure 2. Various heuristics are used in making the choices mentioned in the flow chart.

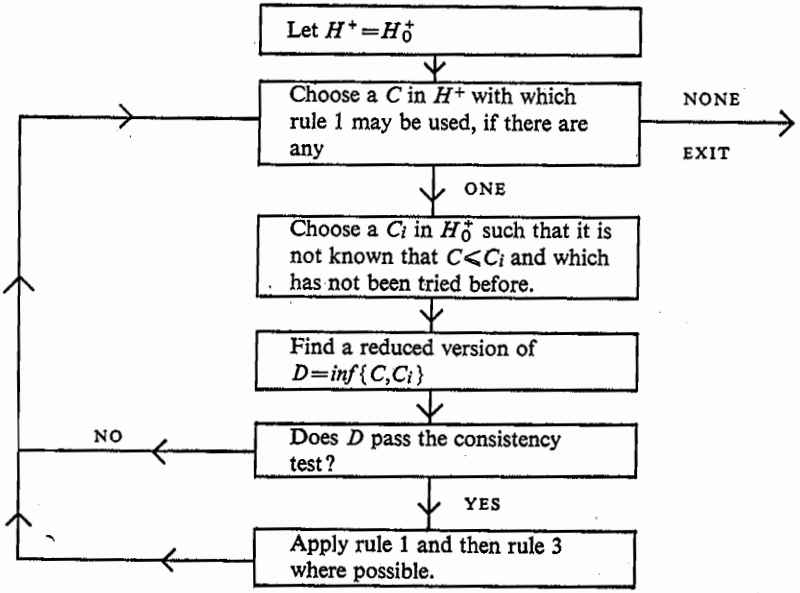



Figure 2

Let $Fail(C)=$
$\{C_i|C_i$ is in $H_0$ and in the course of building up $C$, an attempt to use $C_i$ failed$\}$.

Let $Success(C)=$
$\{C_i|C_i$ is in $H_0$ and in the course of building up $C$, an attempt to use $C_i$ succeeded$\}$.

The clause, $C$, can only be selected if $Fail(C)\cup Success(C)\neq H_0$.

The program chooses that $C$ with largest $Success(C)$ and of two such clauses prefers the one with the smaller failure set.

Clause $C_i$ is chosen to be one for which there is a minimal number of $C$s in $H$ such that $C_i$ is in $Success(C)$. This reflects our belief that the chance of failing, and so wasting a lot of computational effort, grows with the number of previous successes in explaining $C_i$.

Now, $D$ is found using the algorithm of Plotkin (1970). Only an approximate form of the reduced clause is found. We order $D=\{L_1, \ldots, L_n, \ldots, L_m\}$ as follows. The literals $L_1, \ldots, L_n$ all have different predicate symbol and sign pairs. Further if a predicate symbol and sign pair occurs in $D$, it occurs in $L_1, \ldots, L_n$. The literal $L_i$ has less variables than any other literal with that predicate symbol and sign. $(i=1, n)$. For $n<i\leqslant m$, $L_i$ has less variables (not in $L_1, \ldots, L_{i-1}$) than in any other of the $L_i \ldots L_m$.

We then take as our reduced version, the reduced version, calculated properly, of

$\{L_i|1\leqslant i\leqslant min(l, m)\}$, where $l$ is a program parameter. We set $l$ to be 11.

The consistency test checks whether

$C\leqslant \bar{e}_i\cup\{\bar{f}_i\}$ for some negative $f_i$.

#### An experiment using the win predicate of noughts and crosses

Our first example was taken from the field of noughts and crosses. This problem does fall under the s2 subcase; positions play the part of situations.

The board is considered to be a three by three matrix. There are three predicate symbols, $XX$, $OO$ and $Win$.

| | | |
|---|---|---|
| | X | O |
| O | X | X |
| | X | O |

Figure 3

$XX(i, j, p)$ is true iff position $p$ has an X in square $(i, j)$. The predicate $OO$ is defined similarly. $Win$ has the obvious definition. As an example, suppose that $p()$ is the name of the position in figure 3.

The fact, $f$, to be explained is

$f=Win(p())$.

The relevant evidence, $e$, is taken to be a description of the board,

$$
\begin{aligned}
e &= Ev(f) \\
&= XX(1, 2, p()) \wedge XX(2, 2, p()) \\
&\wedge XX(2, 3, p()) \wedge XX(3, 2, p()) \\
&\wedge OO(1, 3, \mathrm{p}()) \wedge OO(2, 1, p()) \\
&\wedge OO(3, 3, p()).
\end{aligned}
$$

Note that 1, 2, 3 and $p()$ are constant symbols. We give the result for a set of positions taken from Popplestone (1970). The win and non-win positions are displayed in tables 3 and 4 respectively. The program found an $H^+ = \{C_1, C_2\}$ where:

$$
\begin{aligned}
C_1 = &\overline{XX}(i, 3, p) \vee \overline{XX}(m, i, p) \\
&\vee \overline{XX}(m, 1, p) \vee \overline{XX}(i, m, p) \\
&\vee W(p).
\end{aligned}
$$

Table 3

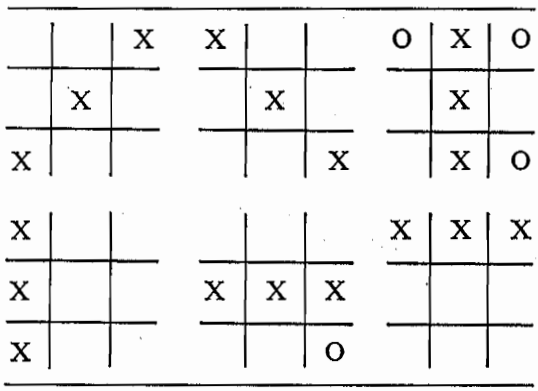
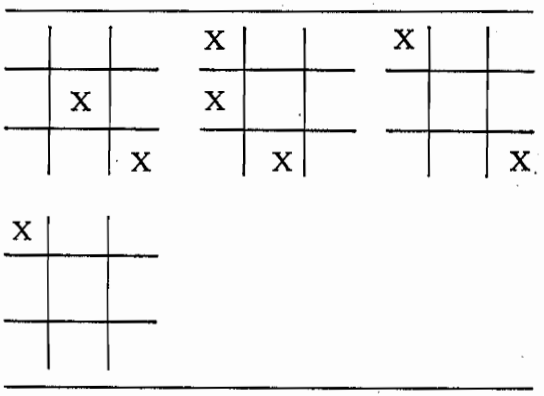


Table 4



and

$$C_2 = \overline{XX}(3, d, p) \vee \overline{XX}(d, d, p)$$
$$\vee \overline{XX}(2, g, p) \vee \overline{XX}(g, g, p)$$
$$\vee \overline{XX}(i, i, p) \vee \overline{XX}(1, i, p)$$
$$\vee W(p).$$

Note that $i$, $p$, $m$, $d$ and $g$ are variable symbols.

The clause $C_1$ has as a consequence that any position containing a column of xs or a forward diagonal row is a win; $C_2$ does the same for the rows and the backward diagonal. These rather elegant explanations were available because there was not enough negative evidence. For example, $C_1$ implies that any position containing an x in (1, 3) and (3, 1) is a win.

We might have used an additional predicate symbol $BB$ to indicate the presence of blanks. Thus to $Ev(f)$, for the position in figure 3 we would add the literals $BB(1, 1, p())$ and $BB(3, 1, p())$. Alternatively, we could dispense with $XX$, $OO$ and $BB$ and use $Occ$ instead where, for example, $Occ(X, i, j, p)$ means that position $p$ has an x in square $(i, j)$. Both these alternatives result in much more data processing and the need for many more examples of wins and loses before reasonable hypotheses are generated. This is because the richer language allows the expression of more hypotheses which must be eliminated before reaching a good one.

If we changed the win criterion so that as well as a line of xs we required also that (1, 1) did *not* have an $O$ then with the method of describing positions in terms of $XX$ and $OO$ that we use, we would *never* obtain a reasonable hypothesis. If the negative information (such as $\overline{OO}(1, 1, p())$) were added, we would be able to get a good hypothesis. This is the problem of determining just what is relevant to a given fact $f$.

Many of these ills might be cured by extending the theory to allow $Th$ to include such sentences as:

$$\forall i, j, p\ XX(i, j, p) \equiv \neg OO(i, j, p).$$

#### Learning the patrilineal ancestor relationship

Our next example is concerned with family relationships. We tried to discover sufficient conditions for the satisfaction of the binary relation, patrilineal ancestor which is recursively defined by

$$anc(x, y) \equiv father(x, y) \vee \exists z(father(x, z) \wedge anc(z, y)).$$

This problem falls under case s1 rather than s2. However, we used the program as an indicator of the results a more thorough one would produce. The family tree is given in figure 4.

We used three binary predicate symbols, *Father*, *Daughter*, and *Anc*, with the evident meanings. $Ev(f)$ was chosen differently according to whether or not $f$ was positive. If $f$ was $Anc(x, y)$ then we chose a reasonably small set of literals which established some sort of link between $x$ and $y$. A link is a set of literals $\{L_i | i = 1, n\}$ where each $L_i$ is of the form $P_i(x_i, y_i)$ or $P_i(y_i, x_i)$ and $x = x_1$, $y_i = x_{i+1} (1 \leqslant i \leqslant n-1)$ and $y_n = y$. If the smallest link for $f$ had $n$

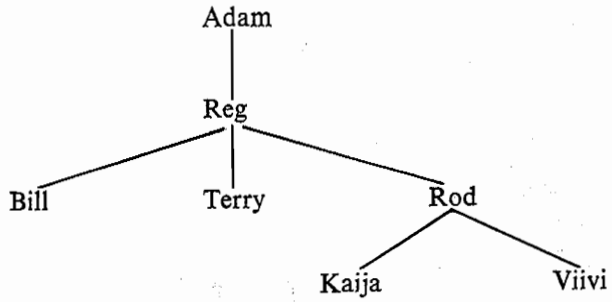



Figure 4

Table 5

| $f$ | $e$ |
|---|---|
| $f_1 = Anc(Rod, Kaija)$ | $e_1 = Daughter(Kaija, Rod)$<br>$Father(Rod, Kaija)$ |
| $f_2 = Anc(Reg, Terry)$ | $e_2 = Father(Reg, Terry)$ |
| $f_3 = Anc(Reg, Kaija)$ | $e_3 = Father(Reg, Rod)$<br>$Daughter(Kaija, Rod)$<br>$Father(Rod, Kaija)$<br>$Anc(Rod, Kaija)$<br>$Anc(Reg, Rod)$ |
| $f_4 = Anc(Adam, Kaija)$ | $e_4 = Anc(Adam, Reg)$<br>$Anc(Reg, Kaija)$<br>$Anc(Rod, Kaija)$<br>$Father(Reg, Rod)$<br>$Father(Rod, Kaija)$ |
| $f_5 = \overline{Anc}(Terry, Bill)$ | $e_5 = Anc(Reg, Bill)\ Anc(Reg, Terry)$<br>$Anc(Adam, Terry)\ Anc(Adam, Bill)$<br>$Anc(Adam, Reg)\ Anc(Adam, Rod)$<br>$Anc(Rod, Kaija)\ Anc(Adam, Kaija)$<br>$Father(Reg, Terry)\ Father(Reg, Bill)$<br>$Father(Reg, Rod)\ Father(Rod, Kaija)$<br>$Daughter(Kaija, Rod)\ Anc(Adam, Viivi)$<br>$Anc(Reg, Rod)\ Anc(Reg, Kaija)$<br>$Anc(Reg, Viivi)\ Anc(Rod, Viivi)$<br>$Father(Rod, Viivi)\ Daughter(Viivi, Rod)$ |
| $f_6 = \overline{Anc}(Terry, Rod)$ | $e_6 = e_5$ |

literals we took $Ev(f)$ to be the union of a few of the links of length $n$. If $f$ was negative, we simply let $Ev(f)$ be the whole family tree. The $e_i$ and $f_i$ are displayed in table 5.

The program found an $H = \{C_1, C_2\}$ where

$$C_1 = \overline{Father}(x, y) \vee Anc(x, y)$$

and

$$\begin{aligned} C_2 = {} & \overline{Anc}(p, q) \vee \overline{Father}(p, q) \\ & \vee \overline{Anc}(Reg, q) \vee \overline{Father}(Reg, Rod) \\ & \vee \overline{Father}(Rod, Kaija) \vee \overline{Anc}(Rod, Kaija) \\ & \vee \overline{Anc}(k\ \ n) \vee \overline{Anc}(n, Kaija) \\ & \vee Anc(k, Kaija) \end{aligned}$$

Note that $x, y, p, q, k$ and $n$ are variable symbols.

$C_1$ explains $f_1$ and $f_2$. $C_2$ explains $f_3$ and $f_4$. Now, one can show that

$$C_2 \sim C_3(\bigwedge_{i=1}^{6} (e_i \wedge f_i))$$

where

$$\begin{aligned} C_3 = {} & \overline{Anc}(k, n) \vee \overline{Anc}(n, Kaija) \\ & \vee Anc(k, Kaija). \end{aligned}$$

We plan to add this facility of reduction, relative to all known facts, to the program.

The clauses $C_1$ and $C_3$ constitute exactly the sufficient half of a good definition of $Anc(x, Kaija)$. We could not have expected more, since the only examples we gave where the recursive part of the definition had to be invoked were of patrilineal ancestors of *Kaija*.

This experiment shows up rather clearly that, although the choice of *Ev* seems *reasonable*, it is desirable to have a more thoroughgoing *theory* of the choice of *Ev*, and the interaction of that choice with the generalization theory.

## CONCLUSIONS

We claim three things. First we have a reasonable philosophically-oriented theory of generalization. This is meant to be a study of the weakest possible kind of hypothesis formation. It may very well not lead to anything more general. Nonetheless an ability to form generalizations will not be useless for a robot. Secondly, there is an interesting mathematical theory which seems open to development. Lastly, pilot experiments have shown promise of a practical algorithm.

**Acknowledgements**

I am indebted to my supervisor, Dr Rod Burstall, for his support and guidance. Further thanks are owed to Mr Pat Hayes and Dr Bernard Meltzer for discussions and criticism. The work was supported by an SRC research studentship.

## REFERENCES

Barrow, H.G. & Popplestone, R.J. (1971) Relational descriptions in picture-processing. *Machine Intelligence 6*, paper no. 23 (eds Meltzer, B. & Michie, D.). Edinburgh: Edinburgh University Press.

Buchanan, B.G. (1966) Logics of scientific discovery. *Stanford Artificial Intelligence Memo. No. 47.* Department of Computer Science, Stanford University.

Carnap, R. (1950) *The logical foundations of probability.* Chicago: University of Chicago Press. (Second edition, 1963.)

Carnap, R. (1967) *The logical structure of the world.* London: Routledge and Kegan Paul.

Hayes, P.J. (1971) A logic of actions. *Machine Intelligence 6*, paper no. 27 (eds Meltzer, B. & Michie, D.). Edinburgh: Edinburgh University Press.

Hintikka, J. (1965) Towards a theory of inductive generalization. *Proc. 1964 Intern. Congress for Logic, Methodology, and Philosophy of Science*, pp. 274–88 (ed. Bar-Hillel, Y.). Amsterdam: North Holland.

Hintikka, J. & Hilpinen, R. (1966) Knowledge, acceptance and inductive logic. *Aspects of Inductive Logic*, pp. 1–20 (eds Hintikka, J. & Suppes, P.). Amsterdam: North Holland.

Hunt, E.B., Marin, J. & Stone, P.J. (1966) *Experiments in induction.* New York & London: Academic Press.

Kemeny, J.G. (1953) The use of simplicity in induction. *Phil. Rev.*, **62**, 391–408.

Kowalski, R. (1970) Studies in the completeness and efficiency of theorem-proving by resolution. *Ph.D. Thesis.* Metamathematics Unit, University of Edinburgh.

Kreisel, G. (1967) Informal rigour and completeness proofs. *Problems in the philosophy of mathematics*, pp. 138–57 (ed. Lakatos, I.). Amsterdam: North Holland. Reprinted with a postscript in *The Philosophy of Mathematics*, pp. 78–94 (ed. Hintikka, J.). Oxford: Oxford University Press, 1969.

Kreisel, G. (1970) Hilbert's Programme and the search for automatic proof procedures. *Lecture notes in mathematics*, pp. 128–46 (eds Dold, A. & Eckmann, B.). Heidelberg & Zurich: Springer-Verlag.

Lee, C. (1967) A completeness theorem and a computer program for finding theorems derivable from given axioms. *Ph.D. Thesis.* University of California, Berkeley.

Meltzer, B. (1970) Power amplification for theorem-provers. *Machine Intelligence 5*, pp. 165–79 (eds Meltzer, B. & Michie, D.). Edinburgh: Edinburgh University Press.

Plotkin, G.D. (1970) A note on inductive generalization. *Machine Intelligence 5*, pp. 153–63 (eds Meltzer, B. & Michie, D.). Edinburgh: Edinburgh University Press.

Pólya, G. (1957) *How to Solve It.* New York: Doubleday.

Popper, K.R. (1959) *The Logic of Scientific Discovery.* London: Hutchinson.

Popplestone, R.J. (1970) An experiment in automatic induction. *Machine Intelligence 5*, pp. 203–15 (eds Meltzer, B. & Michie, D.). Edinburgh: Edinburgh University Press.

Putnam, H. (1963) Degree of confirmation and inductive logic. *The Philosophy of Rudolf Carnap*, pp. 761–83 (ed. Schilpp, P.A.). La Salle, Illinois: Open Court Publishing Co.

Quine, W. (1955) A way to simplify truth functions. *Amer. Math. Month.*, **62**, 627–31.

Robinson, J.A. (1965) A machine-oriented logic based on the resolution principle. *J. Ass. comput. Mach.*, **12**, 23–41.

END