THE UNIVERSITY *of* EDINBURGH

Automatic methods of inductive inference

Gordon D. Plotkin

Ph.D. Thesis

University of Edinburgh

1971

Abstract

This thesis is concerned with algorithms for generating generalisations from experience. These algorithms are viewed as examples of the general concept of a hypothesis discovery system which, in its turn, is placed in a framework in which it is seen as one component in a multi-stage process which includes stages of hypothesis criticism or justification, data gathering and analysis and prediction. Formal and informal criteria, which should be satisfied by the discovered hypotheses are given. In particular, they should explain experience and be simple. The formal work uses the first-order predicate calculus.

These criteria are applied to the case of hypotheses which are generalisations from experience. A formal definition of generalisation from experience, relative to a body of knowledge is developed and several syntactical simplicity measures are defined. This work uses many concepts taken from resolution theory (Robinson, 1965). We develop a set of formal criteria that must be satisfied by any hypothesis generated by an algorithm for producing generalisation from experience.

The mathematics of generalisation is developed. In particular, in the case when there is no body of knowledge, it is shown that there is always a least general generalisation of any two clauses, in the generalisation ordering. (In resolution theory, a clause is an abbreviation for a disjunction of literals.) This least general generalisation is effectively obtainable.

Some lattices induced by the generalisation ordering, in the case where there is no body of knowledge, are investigated.

The formal set of criteria is investigated. It is shown that for a certain simplicity measure, and under the assumption that there is no body of knowledge, there always exist hypotheses which satisfy them. Generally, however, there is no algorithm which, given the sentences describing experience, will produce as output a hypothesis satisfying the formal criteria. These results persist for a wide range of other simplicity measures. However several useful cases for which algorithms are available are described, as are some general properties of the set of hypotheses which satisfy the criteria.

Some connections with philosophy are discussed. It is shown that, with sufficiently large experience, in some cases, any hypothesis which satisfies the formal criteria is acceptable in the sense of Hintikka and Hilpinen (1966). The role of simplicity is further discussed. Some practical difficulties which arise because of Goodman's (1965) "grue" paradox of confirmation theory are presented.

A variant of the formal criteria suggested by the work of Meltzer (1970) is discussed. This allows an effective method to be developed when this was not possible before. However, the possibility is countenanced that inconsistent hypotheses might be proposed by the discovery algorithm.

The positive results on the existence of hypotheses satisfying the

formal criteria are extended to include some simple types of knowledge. It is shown that they cannot be extended much further without changing the underlying simplicity ordering.

A program which implements one of the decidable cases is described. It is used to find definitions in the game of noughts and crosses and in family relationships.

An abstract study is made of the progression of hypothesis discovery methods through time.

Some possible and some impossible behaviours of such methods are demonstrated. This work is an extension of that of Gold (1967) and Feldman (1970). The results are applied to the case of machines that discover generalisations. They are found to be markedly sensitive to the underlying simplicity ordering employed.

Acknowledgements

This research has been supported by the Science Research Council.

I am indebted to my supervisors, Dr. Rod Burstall, who has been a source of good advice for several years, and Professor Donald Michie, for their support, guidance and encouragement. In addition, I have had many valuable conversations with Mr. Pat Hayes, Dr. Robert Kowalski, Dr. Bernard Meltzer, Mr. Robert Owen and Mr. Robin Popplestone.

I thank the people who helped with the production of this thesis. In particular, I thank Miss Eleanor Kerse who carried out the long task of typing the thesis.

I thank Edinburgh University Press for permission to include material contained in the Machine Intelligence Workshop series (Plotkin, 1970 and Plotkin, 1971).

INDEX

Abstract ii
Acknowledgements v
Index vi
Chapter 1: Hypothesis discovery 1
1.1 Introduction 1
1.2 Criteria for hypothesis formation 8
Chapter 2: The generalisation problem 29
Chapter 3: The mathematics of generalisation 46
3.1 Preliminaries 46
3.1.1 Notation 46
3.1.2 Generalisation 47
3.1.3 Relative generalisation 49
3.2 Generalisation theory of literals 55
3.2.1 Least general generalisations of literals 55
3.2.2 A technical lemma 63
3.2.3 Lattice properties of literals 65
3.3 Generalisation theory of clauses 70
3.3.1 Elementary properties 70
3.3.2 Least general generalisations of clauses 78
3.3.3 Lattice properties of clauses 87
3.3.3.1 Finitary properties 87
3.3.3.2 Infinitary properties 96
Chapter 4: Solvability of the general problem 104
Chapter 5: Applications and extensions 134
5.1 Some philosophical remarks 134
5.2 Two extensions to a more complex kind of theory, Th 152
5.2.1 Sorted languages 152
5.2.2 Algorithms for a simple kind of theory: ground literals.152
5.3 A general algorithm using a limited consistency check...162
5.4 Some pilot experiments 166
5.4.1 Descriptions of the program 166

5.4.2 Evaluation of experimentation 170
5.4.3 An experiment using the win predicate of noughts and crosses 172
5.4.4 Learning the patrilineal ancestor relationship.180
Chapter 6: Hypothesis learning theory 189
6.1 Introduction 189
6.2 Abstract theory 190
6.3 Inferring hypotheses 197
6.4 Inferring good hypotheses 203
6.5 Generalisation and hypothesis learning theory.207
6.6 Conclusions 211
References 213

Chapter 1 Hypothesis discovery

1. Introduction

This thesis is concerned with forming hypotheses by generalisation. We hope to expose a more interesting structure than would be expected from contemplation of the traditional recipe for generalisation which is the simple replacement of constants by variables. Thus we are committed from the start to a study of generalisation of sentences in the predicate calculus. First-order predicate calculus will be quite sufficient although some other formalisms will be mentioned. We assume, therefore, that the reader is acquainted with some standard formulation of first-order predicate calculus, such as that of Schoenfield (1967).

Some examination of generalisation processes in the predicate calculus may be found in Meltzer (1970a). He constructed a program to find general laws of group theory from particular examples. Some other work in the domain of predicate calculus has been done by Popplestone (1970), although this was not concerned with generalisation alone, but contained other rules for generating hypotheses, a heuristic search through the hypothesis space and a mechanism for generating "test" experiments to decide between competing hypotheses.

The largest single body of work in A.I. research on hypothesis formation has been on guessing grammars. A method for guessing finite-state grammars was devised by Chomsky and Miller (1957) and generalised to context-free grammars by Solomonoff. The adequacy of Solomonoff's (1964)

method has been challenged by Shamir and Bar-Hillel (Shamir, 1962).

The first heuristic program for guessing finite-state grammars seems to be that of Feldman (1967).

Theoretically, we find contributions by Gold (1967), Feldman (1970), Feldman, Gips, Horning and Reder (1969) and Horning (1969).

Gold showed what kind of behaviour could theoretically be expected from grammar-guessing machines. Feldman (1970) continued these studies. In Feldman, Gips and Horning we find both theoretical results which are less general but stronger than those in Gold, and descriptions of a practical program for inferring pivot grammars, which form a subclass of the context-free grammars and which properly contain the finite-state grammars. Horning, too, conducts both a theoretical and practical investigation, based on Bayesian ideas.

His program has the special distinction of being, in one sense, theoretically optimal. Seemingly, however, it has less practical ability than the other, heuristic, programs.

Closely related work on guessing finite-state machines can be found in Perryman (1970) and Feldman and Biermann (1970). Taking this work together with that of the grammar guessers, we may conclude that only a little need yet be done to obtain a theoretically and practically good algorithm for guessing a finite-state grammar from a set of examples and a set of non-examples. Matters remain unsatisfactory, however, in the case of context-free grammars.

Amarel (1962, 1971) is concerned with guessing programs of a rather simple type with no looping, from samples of input-output pairs. He seems not to have programmed his method. Hewitt (1968) tries to guess programs with, possibly, recursion, but using as data traces of the program he is trying to guess. This method seems not to have been programmed.

Given descriptions of both examples and non-examples of a class of pictures, Winston (1970) attempts to generate a general description of the entire class. One interesting feature of his program is that new examples or non-examples may be taken into account by alteration of the current general description. The program is unique of its kind and seems quite successful.

This leads into the field of pattern recognition where we may, for example, regard the perceptron convergence theorem (Nilsson 1965, Minsky and Papert, 1969) as demonstrating the successful operation of a hypothesis-guessing machine.

Standing on its own is the work of Buchanan, Sutherland and Feigenbaum (1969, 1970) and Feigenbaum, Buchanan and Lederberg (1971) on hypothesis-formation in organic chemistry. The programs they developed form the most impressive hypotheses of any, although there are correspondingly strong assumptions.

We should mention also some work in psychology, in, essentially, a very restricted portion of the first-order predicate calculus. This

concerns concept-learning. Notable books are those of Bruner, Goodnow and Austin (1956) and Hunt, Marin and Stone (1966).

Work on the related subject of analogy has been done by Evans (1968), Kling (1971) and Becker (1970). Evans was in fact, largely concerned with generalisation, although his work concerned analogy questions in I.Q. tests. The solutions generated by his program were in almost total agreement with those of the proposers of the tests. Kling was interested in the use of analogy to help in proving theorems analogous to already proven ones. Kling has programmed his method. Becker proposed his notions of analogy as an essential component in a model of "intermediate level cognition". He has not programmed his method.

Of course one could go on for ever quoting work on hypothesis discovery. However the above gives a good indication of what has been done in the way of actually proposing and implementing algorithms.

While we will both propose and implement discovery algorithms, we will be mainly concerned with a theoretical analysis of the relation of generalisation and its use. It is hoped that this will provide some useful notation and techniques for further development of algorithms. The analysis sets up criteria derived from the philosophy of science. The possibility of doing this was explicitly stated by Buchanan (1966), to whom we owe some considerable debt. The hypothesis formation problem is seen as a stage in a continuing process of theory formation and criticism, data gathering, prediction and so on. Criteria that a hypothesis formation method should satisfy can then be set up. This

forms the main part of this chapter. The criteria depend on what notion of explanation of the given data is employed. In chapter 2 we develop a definition of generalisation and so specialise the criteria that they result in a tractable formal problem of finding a machine which will produce hypotheses satisfying the criteria. In fact, a "nicest" such hypothesis will be required.

In chapter 3 the abstract theory of the generalisation relationship is developed and employed to examine the formal problem in chapter 4 and to provide several illustrations and other applications in chapter 5.

Finally, in order to partially correct the distorted emphasis on a single, idealised stage of hypothesis formation, we give in chapter 6 a generalisation of Feldman's theory of hypothesis identification in the limit, (Feldman, 1970).

The work started with a suggestion by R.J. Popplestone (private communication) that, just as the unification algorithm was fundamental to deduction, so might a converse be of use in induction. Unification is a basic idea in the theory of resolution given by Robinson (1965). We refer to his work for a complete description of the notation used in that theory.

For our immediate purposes, it is only important to note that a literal is an atomic formula or the negation of one and that a clause is a set of literals and it abbreviates the disjunction of its members (taken in some standard order). The letters L, M and N are used to

stand for literals. The letters C, D and E are used to stand for clauses.

A literal is a unification of two literals iff it is an instance of each. A literal is a most general unification of two literals iff any other unification of them is an instance of it.

Similarly, a literal is a generalisation of two literals iff each of them is an instance of it. A literal is a least general generalisation of two literals iff it is an instance of any other generalisation of them.

For example, a most general unification of P(x,x) and P(f(y),f(g(z))) is P(f(g(z)),f(g(z))). A least general generalisation of them is P(x,y).

The existence of least general generalisations was soon shown. In fact they are easier to obtain than unifications. A necessary and sufficient condition that two literals have a least general generalisation is just that they have the same predicate letter and sign. However this is not enough even to generate simple universal laws of the form: $\forall x (P(x) \rightarrow Q(x))$. To do this it is necessary to consider generalisations of clauses. Let us say, again for the moment, that a clause C is more general than a clause D if C subsumes D - that is if there is a substitution σ such that $C\sigma \subseteq D$. (In resolution theory, substitutions are functions which operate on expressions; they are denoted by Greek letters.) One can then define the least general

generalisation of two clauses. The key theorem is that any two clauses have a least general generalisation.

It is interesting to note that the similarity between deduction and induction breaks down here. What is useful is not a concept of unification of two clauses, but the deduction principle called resolution.

We then envisaged crude algorithms which would build up from clauses abbreviating implications between ground literals the set of least general generalisations and pick some subset as the proposed hypothesis. (A ground literal is one that contains no occurrences of variables.) Such a subset would have to be consistent with the data, but this does not rule out enough combinations and it is necessary to impose a condition that only the "nicest" (according, say, to some measure of simplicity) combination be picked.

Encountering Buchanan's work (1966) we realised that a hypothesis formation (or suggestion or discovery) method should be placed within a philosophical framework so that, for example, the suggested method could be criticised as not meeting various criteria found in the literature. Thus we arrive in almost the reverse order at the beginning of the development of this thesis. The main technical addition to the concept of a least general generalisation is that of generalisation relative to a body of knowledge. This enables, for example, a robot to form generalisations about its sensory experience, given in terms of light patterns on a retinal grid, in the more abstract language of

objects such as bananas, faces, spectacles and so on.

2. Criteria for hypothesis formation

We begin with some general arguments that induction may be useful in A.I. robot research and, if so, such criteria as may be provided by the philosophy of science should be accepted. Of course hypothesis formation, being one might say the intellectual activity par excellence requires no justification for its study for its own sake. As, however, we believe that integrated robot systems are of some importance in A.I., arguments relating such systems with inductive ones should be considered.

One can easily see in general terms how inductive abilities would be of help to a robot, i.e. an artificial rational man. Such an ideal entity should be a scientist - and so, a non-deductive reasoner of some kind, depending on one's philosophy of science. Again, a robot should have common sense and be able to talk a common-sensical language, such as English. It is a very defensible thesis that both English and common-sense involve a naive science. Both learning and using this naive science will therefore involve the robot in some naive non-deductive reasoning.

There is, even in present robots, an implicit form of inductive ability. For example, all present robots base their plans of action on rather brief glimpses of the world, since picture processing using available techniques is slow. This reflects an inductive expectation that the world will not change too much between looks. It is however

quite unclear what inductive assumptions are definitely already built-in, let alone which ones ought to be.

A robot should have the explicit ability to learn predicates ostensively. Several presentations of a lamp standard before the robot's eye should result in his forming a general idea of lamp standards, (Barrow and Popplestone, 1971, Winston, 1970). He should also be able to perform inductions from activities or events and learn causal connections (Hayes, 1971).

Fancifully, by learning plausible beliefs about the effects of actions, but being prepared, if necessary to look or account for exceptions, it may be possible to ameliorate what McCarthy calls the "frame" problem (McCarthy and Hayes 1969, Hayes, 1971).

Most of the above examples show that some kind of non-demonstrative reasoning is required. It has not been shown that the most suitable form is that of hypothesis formation. Such a demonstration would require a much fuller specification of the robot's mental life than has been given. We do not have an integrated sketch of his ontology and epistemology and theory of perception and the structure of his knowledge and his methods of plan formation and his motivations and his physical being and so on.

Nonetheless, hypothesis formation is a leading candidate; analogical reasoning is the only other contender so far. It seems, therefore, worthwhile to develop hypothesis formation itself as much as possible.

We are concerned then with the automatic formation of hypotheses. At the most grandiose level, one would want a machine which could try to solve the problems of contemporary physics. At the lowest, one would wish to be able to (non-deductively) infer 'All crows are black' from 'That crow is black' and 'This crow is black'. In order to make quite clear the low level at which present general programs operate (without, however, any intention of denigrating the constructers of these systems) here are some examples:

a) Feldman, Gips, Horning and Reder (1969) have studied the induction of grammars (mainly context-free) from finite sets of samples of legal strings, and, perhaps, a set of illegal strings.

On being told that {AABB, AB, AAABBB} is a set of examples, a program of theirs, GRIN2, produced the following grammar:

X := AY

Y := XB/B.

On being told that {C, ACB, AACBB, AAACBBB} was a further set of examples another program GRIN2A, produced the following grammar:

X := AY/C

Y := XB/B.

b) Meltzer (1970) has looked at the problem of determining the axioms for a class of interpretations, given a finite number of facts about each of a finite class of structures.

He fed in a representation of the following facts about the cyclic groups of orders 2 and 4, whose domains are {e,a} and {e,b,c,d} respectively.

$e.e = a$; $a.e = a$; $(a.a).e = a.(a.e)$; $(e.a)a = e$; $e.a \neq e$;

$a.a \neq a$; $b.c = c.b$; $(b.b) . b = c$; $(b.b).c \neq c$; $(b.c).c \neq b$.

The program generalised these representations, and obtained a representation of the following induced axioms.

$x.e = x$;

$(x.x).y = w$ implies $x.(x.y) = w$;

$(y.a).a = y$;

$x.y = y.x$;

$b.(b.b) = y$.

At their best, these and similar general programs such as our own are only slightly more magnificent. There are much more specific programs which generate more impressive hypotheses, of which a prime example is the Heuristic Dendral program of Buchanan, Sutherland and Feigenbaum (1969, 1970). This program's ability is almost entirely due to the availability of a large amount of chemical knowledge (although this knowledge was by no means there for the taking).

We will not attempt any general theory of theory generation. What is possible, in general, is to set up an outline which any generation method must fill in. This outline is derived from the philosophy of science. Apart from any use to which we actually put

this work, we are convinced that much can be gained from its study. It provides discussion of the justification and criticism of hypotheses; of problems of explanation and the nature of simplicity. One can find categorizations of different types of hypotheses and various qualities of hypotheses. There are accounts of the actual and ideal progress of science, and accounts of the dynamics of theory construction. There is the problem of the nature of scientific language and its relation to the world; it is important to know the 'behind the scenes' assumptions made automatically when one chooses to use a particular theoretical language.

However, philosophy does not deal directly with our main concern, which is the generation of hypotheses. Indeed the philosophers generally delegate this problem to the psychologists. We therefore avoid a great deal of philosophical discussion since, for the most part, we have only extricated a schema of philosophical questions, rather than answers. So we expect that most of the current philosophical approaches could be so adapted as to fit in with our schema. This is not to say that we regard the different approaches as being essentially the same. The fact is that our schema is largely incomplete and typically (especially as regards the problem of justification of hypotheses) leaves unanswered just those questions at which the bones of contention arise. Neither have we succeeded in altogether avoiding philosophical commitment. We have tended to raise questions and problems in a way which is rather more acceptable to the Carnapians than the Popperians. Sometimes we use terminology in a way that is unacceptable to either. For example we discuss the problem of justification. A Carnapian would prefer the

problem of confirmation, saying that no absolute justification is possible, in general. A Popperian would also dislike the problem of justification. He would say that the problem is how to decide between competing hypotheses. There is a problem of criticism. A Popperian might say that there is no problem of justification but perhaps one of corroboration. Nonetheless the debate does hinge on the existence and the importance of the problem of justification and the other related ones. We will therefore use the word justification to introduce this whole nexus of problems.

To rephrase our concerns, we are looking for a language to uniformly describe the various attempts made in A.I. to construct programs for hypothesis formation. It is believed that this will lead to a general theory in time to come, and that in the immediate future, when one wishes to program a method of theory construction, one will have available an interesting set of non-trivial questions about one's method.

Hypothesis formation is regarded as a process containing stages of theory construction, theory criticism, data gathering and data analysis. There will also be some supervisory mechanism to decide the order of these stages. As mentioned above we do not know and will not consider, how this fits into the general mental character of an ideal rational man. In fact, for the most part, we will only consider a simplified snapshot of the process, consisting of a stage of theory construction followed by one of theory criticism. The stage of theory

criticism will be dealt with, on the whole, by reference to various possible philosophical positions.

There is no virtue in our omissions; they are blanks which should be filled. It will surely happen that when the process is considered as a whole many missing parameters will be discovered. The analysis should therefore be considered entirely provisional.

Suppose then that our robot is about to formulate and then criticise a theory after some stages of data gathering and analysis. The robot will possess knowledge and beliefs and pragmatic attitudes to these elements of knowledge and belief. Let us call all this k. He will have before him some body of phenomena, f, together with the relevant circumstances of their occurrence, e, for which it is required to find some kind of explanatory hypothesis, h. Generally f will be a set of phenomena, $\{f_i \mid i=1,n\}$ and e will be a set of relevant attendant circumstances $\{e_i \mid i=1,n\}$ given by a function Ev, that is $e_i = Ev(f_i)$. How the e_i are selected depends on an implicit stage of data gathering. The more naive this stage is, the more irrelevant information e_i will contain. The lack of any theory of good data gathering is certainly one of the most important omissions.

In general, there will be many explanatory hypotheses and so it will be necessary to choose the nicest. We expect that there will be some measure, $\succsim$, of niceness. By $h \succsim h'$, we mean that h is at least as nice as h'; $\succsim$ will be transitive and reflexive. This niceness ordering will be defined relative to f and e and, perhaps, k.

Perhaps the most famous example is provided by Newton's theory of gravitation. Here f and e were both large and varied. Among the phenomena was the fact that an apple fell on Newton's head. The relevant circumstances were Newton's situation relative to the apple at the start of its flight and the apple's position in the earth's gravitational field.

The hypothesis h consisted of Newton's theory of gravitation. The knowledge, k, consisted of the axioms of Euclidean geometry and some axioms for time together with the interpretation of both of these via some theory of measurement. Of course, Newton himself did not formulate matters in these terms.

The coincidence of Newton's head with the apple is explained in two stages. First h and k together with the statement regarding the apple's position in the Earth's gravitational field imply that the apple will fall. This and the fact that Newton's head was directly below the apple imply that the apple will strike him on the head.

Putting the two stages together, we see that h and k together with the relevant circumstances imply that the apple will hit Newton's head. Notice that h and k are necessary in this implication. The relevant circumstances alone do not imply the coincidence of apple and head. Notice also that h and k and the relevant circumstances and the coincidence are, taken together, logically consistent.

The great beauty of Newton's theory is its success in explaining,

with simple means, practically every mechanical and gravitational phenomenon known at that time, together with its successful prediction of many more. Perhaps the most spectacular prediction was the existence of Uranus. Indeed the theory possesses nearly every virtue described in the list of virtues given at the end of this chapter. It is, therefore, not too easy to describe $\mathcal{S}$ formally.

In order to be able to formulate and criticise a theory, our robot should therefore possess answers to the following four questions:

H1 Is h justified given f, e and k?

H2 Is there a means of telling when h is justified, given f, e and k?

H3 Does h provide a good explanation in that it is very nice (perhaps maximally) with respect to $\mathcal{S}$ amongst those hypotheses which explain f, given e and k?

H4 Is there a means of finding such a maximally nice h?

Answers to H1, H2 and H3 will enable the stage of criticism to be performed. An answer to H4 is a hypothesis construction method. Before further analysis, it is helpful to consider four analogous questions which arise in mathematics.

Suppose we are looking for a theorem Th in some axiomatic theory T. We want a Th which provides a best answer to some question, Q, about T. Answers are needed for the following questions:

D1 Does Th follow from T?

D2 Is there a means of telling when Th follows from T?

D3 Does Th provide a best answer to Q?

D4 Is there a way to find a Th that is a best answer to Q?

Answers to D1, D2 and D3 represent the criticism stage and D4, the discovery stage in some process of mathematical activity. It may be the case that both hypothesis formation and mathematical discovery can be made to form part of some more general process of knowledge generation. However it should not be imagined that questions H1 and D1 will merge; or questions H2 and D2 and so on. This is because, for example, answering D4 may involve formulating hypotheses or proceeding by analogy with previous results. Similarly it may be necessary to prove theorems to show that h explains f, given e and k.

The distinction between questions D1 and D3 reflects a distinction between truth and interesting appropriateness. When, as here, we do suppose that both Th and T are formulated in the first order predicate calculus, then the Tarskian semantic notion of logical consequence is usually, and justifiably (Kreisel, 1967) taken as a proper formal analysis of the informal notion of logical consequence. Such a situation does not obtain for higher-order logics or modal logics or, certainly, natural language.

Any complete and consistent system of proof for first-order logic gives a correct, but only semi-effective, answer to D2. Note that, in general any answer to D2 can be consistent and/or complete with respect to D1. Similarly any answer to D4 can be consistent and/or complete with respect to D3. In all cases, consistency and

completeness are necessary conditions for the coherence of the entire system.

A further condition of coherence is that the Th discovered as a best answer to D4 should, in fact, follow from T.

This condition seems, on the whole, rather strong. What would be better would be some process which, given a conjectured answer to Q, would either prove Th or suggest, perhaps using a constructed counter-example, some alteration to Q or Th.

Answers to D2 and D4 may be, or may not be, efficient. Not much work has been done on the efficiency of systems of proof. (Kreisel, 1970, Kowalski, 1969, 1970). It seems likely, however, that while there can be no most efficient system of proof, existing methods can be greatly improved. As regards D3 and D4, the amount of systematic work is practically zero, with the well-known exception of that of Pólya (1954, 1957, 1968).

An answer to H2 may be consistent, complete, and/or efficient with respect to H1. Similarly, an answer to H4 may be consistent, complete and/or efficient with respect to H3. A global coherence requirement is that maximally nice h's, which explain f, given e and k, should be justified given e and k. Although this seems rather strong, as does the analogous requirement in the deductive case, nonetheless we shall see in Chapter 5 that it will be satisfied in some simple cases, if e is sufficiently "large".

We will outline some of the points made by philosophers about H1-H4, before settling down to giving a more detailed account of H3.

The distinction made in H1 and H3 between justification and niceness reflects the importance of questions about the truth of hypotheses. At one time, it was held to be possible to discover in a fixed, finite number of steps, important general truths about the world. Mill was about the latest philosopher who halfway believed this. Once it was demonstrated that there was in general no way to determine the truth of general statements, opinion divided, roughly, in two. The Carnapians describe a logical confirmation function, c(h,e'), (Carnap, 1952). This quantity, c(h,e'), has been variously interpreted as the logical probability of h given e', or the betting odds that a rational man would accept on h's being true, given e'. The hypothesis, h, and the evidence for its being true, e', are both framed in first-order logic. Carnap, and this is a quite general opinion, holds that a science can be stated as a first-order axiomatic theory (Carnap, 1967). In our case we would take e' to be e and f and all the true observations contained in k. In any practical situation this would be a hopeless task and only the relevant parts of k would be considered. Probably this would include hypotheses not known to be true and to which only a degree of belief had been assigned. A predicate, Ac(h,e'), can be described, in terms of e', which specifies, in terms of a high, a posteriori confirmation, when h should be accepted given e', (Hintikka and Hilpinen, 1966). For a monadic language the confirmation function c is calculable, but for

a general first-order language, there is not even a semi-effective way to calculate it (Hintikka, 1965, Kemeny, 1953, Putnam, 1956). If we require the strong global coherence condition to hold, it is unimportant that any efficient methods be sought. Nothing is known anyway about efficient methods in general. For the monadic case, there is a close relation with the problem of efficiently finding switching circuits (Quine, 1955).

Popper (1959) is less concerned with formal analysis. He notes that general theories can be falsified and makes this his central plank. For justification he would substitute the requirement that h survive in competition with other hypotheses an extended attempt to falsify it. Only strongly falsifiable and simple hypotheses should be chosen for consideration.

Carnap has little to say on H4, and Popper specifically excludes this from his consideration, as being in the psychological domain.

We turn now to looking at H3 in some detail. As this is not concerned with questions of truth, less has been said. An account of how general laws explain singular statements has been given by Hempel and Oppenheim (1948) and has been subject to criticism by, among others, Eberle, Kaplan and Montague (1961). Niceness is or is partly determined by simplicity. Popper has argued that the degree of falsifiability of a statement correlates strongly with its simplicity. Goodman (1961) rejects this. He also develops (1959) an account of the simplicity of the predicate basis of a theory.

We shall present a picture using a variant of Hempel and Oppenheims' explication of explanation and let the niceness be specified by a parameter. Values of this parameter corresponding, roughly, to different philosophical positions will be used and investigated at different places throughout the rest of the thesis.

The first important assumption is that h and k will be sets of first-order statements, framed in a theoretical and observational language. This over-simplifies the structure of the robot's belief-system k. In particular, all of k is now accepted as being true. This completely ignores the problem of how to deal with large collections of statements to which varying degrees of belief are assigned. Nor does Popper escape a similar dilemma. When k has been falsified, which part of it should be replaced?

The hypothesis, h, will in addition be restricted to be a member of $\mathcal{H}$, the hypothesis space.

The set of phenomena needing to be explained, f, is a set,

$f = \{f_i | i=1,n\}$ of first-order singular sentences.

$e = \{e_i | e=1,n\}$ is also a set of first-order singular sentences.

We will symbolise the relationship between each e_i and f_i by =>. We might regard => as a modal connective and say that the set of statements, $\{e_i \Rightarrow f_i | i=1,n\}$ in a certain modal logic represent the relation between the phenomena and their antecedents.

The interpretation of $e_i \Rightarrow f_i$ is, roughly, that e_i is a description of circumstances or preconditions which resulted in f_i's being true. While it may be a good idea to actually use a modal logic, we could not find any suitable one, and decided to deal with and use $\Rightarrow$ informally on the meta-level of our present discussion. Here is a list of some indications of possible interpretations of $e_i \Rightarrow f_i$.

1) Actions: f_i is the result of an action, or series of actions, described by e_i in circumstances also described by e_i.
2) Direct Cause: e_i is the description of a direct cause of f_i.
3) Experiment: e_i is the description of how an experiment was set up and f_i is the description of its result, for example f_i may be (a description of) a graph (such as one of temperature against pressure obtained from some experiment with gases).
4) Empirical Association: whenever e_i is true f_i occurs (with observed frequency such and such).
5) Temporal Succession: the event f_i followed very soon after the event e_i.
6) Essential Property: f_i is an essential property of objects with the description e_i. For example, having a date inscribed is an essential property of pennies, but being in Harry's pocket is not.
7) Definition: e_i is the appropriate case of the definition of the predicate occurring in f_i.

In the above, as in many places throughout this discussion, we are both using and mentioning the e_i and f_i. It is hoped that the reader will distinguish the different cases.

These possibilities by no means form an independent or complete list. We will adopt 3) as our standard interpretation. With this in mind, we restrict $\mathcal{H}$ to be some subset of the set of lawlike sentences, that is statements whose prenex form contains only universal quantifiers. This is a departure from the standard notion, which, in addition, requires that h contain no individual constants.

We do this because we wish at one extreme to regard a singular statement as being a (completely uninteresting) law. Further we want the niceness criterion, $\mathcal{S}$, to be the factor that strives toward generality. Finally, we feel that the simple grammatical notion of absence of individual constants does not capture the ideal of generality exactly. Many general scientific laws do contain constants - for example the charge of an electron.

Next, we specify what it is for h to explain f given k and e. The belief system, k, is divided into two parts, Th and Irr. Thus $k = Th \wedge Irr$. The system Th consists of that part considered relevant to the set of phenomena at hand, and Irr is the rest, the irrelevant beliefs.

Then, h explains f given $k = Th \wedge Irr$ and e iff:

E1 For all i, $\vdash_{Th} h \wedge e_i \rightarrow f_i$.

E2 For all i, it is not the case that $\vdash_{Th} e_i \rightarrow f_i$.

E3 The sentence $k \wedge h \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)$ is consistent.

E4 The hypothesis h is lawlike.

Strictly speaking, E3 should be written as 'The set of sentences $k \cup \{h, \bigwedge_{i=1}^{n}(e_i \wedge f_i)\}$ is consistent.' This, and similar, confusions will be perpetrated throughout the thesis.

Requirement E1 is certainly the least that h can do if it is to explain each f_i, given e_i and Th. The second requirement, E2, ensures that the phenomena are not trivial, that is that there is something new to be explained. Requirement E3 is a minimal requirement if h is to be incorporated into the body of beliefs. Whether or not one's beliefs are true, they should certainly be consistent with one's observations. Requirement E4 is imposed as a result of the interpretation we adopted of =>. Other interpretations would require other conditions.

By themselves, E1-E4 would not capture some of the "feel" of explanation. This should perhaps be inserted in the description of the niceness relation, $\succ$. Mario Bunge has compiled a very impressive list of possible "niceness" qualities of hypotheses (1961). We will give very brief accounts of each of them. This will give yet another indication of the sheer size of the problem.

Syntactical Requirements:

1) Well-formedness. For us, this is just the requirement that h actually be a wff of the first-order predicate calculus.

2) Connectedness. If a hypothesis is thought of as a conjunction of postulates, and each predicate symbol occurs in many of these postulates, it is well-connected.

Semantical Requirements:

3) Linguistic Exactness. The ambiguity, vagueness and obscurity of a hypothesis should be minimal. Such terms as 'hot' or 'historical necessity' are not welcome.

4) Empirical Interpretability. The hypothesis must make empirical predictions.

5) Representativeness. The theory should deal with actual events and processes. Thus theories of action at a distance are replaced by field theories, showing exactly how action at a distance actually works.

6) Semantical Simplicity. The world should be constructed simply from simple parts. The theory of quarks is an extreme example.

Epistemological Requirements:

7) External Consistency. The hypothesis should be consistent with the bulk of one's knowledge.

8) Explanatory Power. The hypothesis should explain many known empirical facts and generalisations.

9) Predictive Power. The hypothesis should entail many unknown facts.

10) Depth. The hypothesis should explain essentials and reach

deeply into the structure of reality.

11) Extensibility. The hypothesis should be extensible in order to cover new domains, not previously thought of as being relevant to the hypothesis.

12) Fertility. The hypothesis must have exploratory power.

13) Originality.

Methodological Requirements:

14) Scrutability. The predicates involved in the hypothesis must be open to scrutiny by the general public. That is, techniques, tests and evidence must be intersubjective. For example, events should not occur through God's will, nor should Mrs. Smith's female intuition count as a theoretical entity.

15) Refutability. It must be possible to imagine circumstances which could refute the hypothesis. Critical experiments can be set up.

16) Confirmability. The theory must have consequences which agree with observation.

17) Methodological Simplicity. It must be technically possible to subject the theory to empirical tests.

Philosophical Requirements:

18) Level Parsimony. The hypothesis must be parsimonious in its references to sections of reality other than those directly

involved.

19) Meta-Scientific Soundness. The hypothesis must be compatible with fertile metascientific principles such as the requirement that it be lawlike.

20) World-View Compatibility. The hypothesis should be in line with the general world-view of scientists. One should be led to reject crackpot theories, but accept, eventually, genuine scientific revolutions.

It will be seen that some of these requirements are, from our systematiser's point of view, no more than hopeful hand-waving. Some have been covered, some seem extremely hard to formalise and some impossible.

Let us recapitulate the parameters entering into H3.

There is a hypothesis space $\mathcal{H}$.

There is a set of phenomena and their circumstances $\{e_i \Rightarrow f_i \mid i=1,n\}$, with restrictions on the nature of the e_i and f_i and some interpretation of $\Rightarrow$.

There is the knowledge, $k = Th \wedge Irr$.

There is the type of explanation through which each e_i is explained by $h(\text{in } \mathcal{H})$ given k and $f_i(i=1,n)$.

There is some notion of niceness, $\succ$.

Once these parameters have been specified one obtains a method of answering H3 given h, and a formal problem:

"Answer H4 so as to obtain a consistent, complete and efficient algorithm for finding an h which answers H3."

Such an algorithm should cover a large range of possible sets of phenomena, and perhaps some range of possible belief systems k.

Chapter 2 The generalisation problem

We will now narrow our objectives and use the framework constructed in the previous chapter to formalise the problem of finding generalisations from experience. This problem is both the simplest and the most common illustration of non-deductive reasoning. Indeed, so central is it, that some philosophers even use the word induction to mean generalisation. Here is an archetypal piece of generalisation:

The sun rose yesterday
The sun rose today

The sun rises every day

Here is another:

This crow is black
That crow is black

Every crow is black

Some more complicated examples can be found in chapters 3, 4 and 5.

To follow the prescription of chapter 1, we must fill in several parameters.

First we must settle the hypothesis space. This has already been required to be some subset of the set of universal sentences. We will simply require that it is the set of universal sentences.

Next, we must settle the set of phenomena, and the circumstances of

their occurrence, $\{e_i \Rightarrow f_i \mid i=1,n\}$ where $e_i = Ev(f_i)$ $(i=1,n)$ for a certain function Ev. Each phenomenon must be an experience. Let us say, as seems reasonable, that an experience is a fact experienced in certain circumstances, described by another set of facts. This is consistent with our decision to use => as indicating an experiment, if we regard an experience as an unintentional experiment. So f_i is a fact and e_i is a, supposedly finite, conjunction of facts. We assume that every fact can be described by a ground literal. This fits in well with the view that there are basic observational predicates and facts are what are observed. One might make two accusations of unnaturalness. First most facts stated in English are stated in a positive way, and this is usually possible. Thus consider the verbal exchange:

John: "The tap is on"

Mary: "No, it's off."

But here there is a piece of inbuilt knowledge, namely $\forall x (On(x) \equiv \neg Off(x))$. We wish to have a theory which includes the use of no knowledge, when one would want negations.

Secondly, it seems a little odd to allow function symbols other than constants. They do seem to arise occasionally in English in possessive phrases. "John's mother is beautiful" might be rendered as Beautiful(Mother(John)). Since they perhaps ought to be allowed in everyday use and since they certainly ought to be allowed for mathematical facts and are necessary if actions are regarded as

functions (usually from situations to situations), we will allow them. At any rate the theory can be developed just as well for the general as the particular cases. It will, however, make a rather spectacular difference later on. If function symbols other than constants are allowed, then one version of the generalisation problem is unsolvable, while if they are not allowed, it is solvable.

Thus the phenomena are given by a set of facts, $f = \{f_i \mid i=1,n\}$ and a function **Ev** from f to finite conjunctions of facts. So far we have adopted the standard interpretation of => as an experiment, which justifies the selection of the hypothesis space. In any application alternative restrictions may be made on => which may induce other possible restrictions on any of the parameters: the hypothesis space, the set of possible phenomena, the method of explanation or even, perhaps, the niceness criterion.

We must remark that there are some problems whose solution we assume to have taken place. The set f should be a reasonable set for generalising from. **Ev** should choose the (or a superset of the) relevant or correct facts which describe where, when and/or why f_i took place. Such problems would be faced in formalizing the entire process of theory formulation. We do not attempt their formalization here, although one might suspect that if $\mathbf{Ev}(e_i)$ is relevant to f_i then every term in f_i will contain a term occurring in $\mathbf{Ev}(e_i)$. Perhaps the set of atoms in $\mathbf{Ev}(e_i)$ and f_i would form a connected set under the relation of having a common subterm. This weak requirement will not be assumed,

however.

We believe that an event is a fact. This allows a widening of the range of possible interpretations of =>. We make the notational convention that $\bar{e}_i$ is the clause $\{\bar{L} | L$ is one of the literals conjoined to form $e_i\}$.

No conditions are assumed for Th or Irr at the moment. We will not say how k ought to be split up into two parts, except that Irr may contain an account of the failure of certain experiments. That is, it may contain conjunctions of literals of the form $e \wedge \bar{f}$ where e is a conjunction of ground literals explaining some experimental set-up and $\bar{f}$ is a literal which expresses the fact that the expected outcome, f, did not occur. Irr will contain such "failures" when we do not wish to explain why the failure occurred, nor use the failure to explain the successes, but merely wish to find an explanation consistent with the failure.

We will be particularly interested in the case when Th is empty, although we will also consider, in less detail, some other cases. However, much of the theory can, and will, be formulated in general.

There are some restrictions that must be placed on Th, Ev and f, if the phenomena are to admit any explanations.

1 For every i, $e_i \rightarrow f_i$ must not be deducible from Th.

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

We must now capture a suitable notion of generalisation in order to formalise the intended type of explanation. The word generalisation is used in many contexts both formally and informally. It is not clear that there is a common intuition behind these uses. For example $\forall x P(x)$ and $\exists x P(x)$ are, respectively called universal and existential generalisations of $P(a)$. Yet only the former could qualify as an inductive generalisation.

Generalisation from experience seems to occur in two stages. First a relevant part of experience is selected and then generalised by universal generalisation, that is, the replacement of constant terms by universally quantified variables. To reverse this, an experience is explained by a generalisation if it follows logically from an instance of it. It is this sense of generalisation for which we shall attempt a formal parallel.

In our case, an experience is typically an experiment described by $e_i \Rightarrow f_i$ for some i. We assert that the process of selecting a relevant part of it may be divided up into two parts; one of selection and one of rewording.

In a selection, some of the circumstances are regarded as irrelevant. That is, $e_i' \Rightarrow f_i'$ is a <u>selection</u> from $e_i \Rightarrow f_i$ if e' is obtained from e by removing some of its conjuncts.

In a rewording, the experience is redescribed in different terms, possibly using Th. This is how knowledge interacts with generalisation.

Formally, $e_i' \Rightarrow f_i'$ is a <u>rewording</u> of $e_i \Rightarrow f_i$ if $\vdash_{Th} e_i' \equiv e_i$ and $\vdash_{Th} f_i' \equiv f_i$, e_i' is a conjunction of ground literals and f_i' is a ground literal.

Finally, $e_i' \Rightarrow f_i'$ is a <u>part</u> of $e_i \Rightarrow f_i$ if it can be obtained from $e_i \Rightarrow f_i$ by a series of applications of selection and rewording to $e_i \Rightarrow f_i$. Notice that in this case, $\vdash_{Th} f_i' \equiv f_i$. Therefore $e_i' \rightarrow f_i'$ is true, as f_i is. However it may be no longer possible to justify the assertion that $e_i' \Rightarrow f_i'$ as some selection may have removed a relevant part of the circumstances.

The second stage is the replacement of constants by universally quantified variables. We say, therefore, that $\forall(e_i'' \Rightarrow f_i'')$ is a <u>generalisation from</u> $e_i \Rightarrow f_i$ iff for some σ and $e_i' \Rightarrow f_i'$, which is a part of $e_i \Rightarrow f_i$, $e_i''\sigma = e_i$ and $f_i''\sigma = f_i'$. ($\forall(e_i'' \Rightarrow f_i'')$ abbreviates $\forall x_1 \ldots x_n(e_i'' \Rightarrow f_i'')$, where $x_1 \ldots x_n$ are all the variables in $e_i'' \Rightarrow f_i''$.) Here, $\forall(e_i'' \Rightarrow f_i'')$ is said to be a <u>generalisation of</u> $e_i' \Rightarrow f_i'$.

To give a good meaning to $\forall(e_i'' \Rightarrow f_i'')$ would take a longer excursion than is practical. One would have to reread => as a counterfactual (Goodman, 1965, Tredwell, 1965). For example $\forall x(\text{Aristotle}(x) \Rightarrow \text{Speaksgreek}(x))$ would mean that anyone who was Aristotle could speak Greek.

Let us illustrate the above by an informal example in which => means mo more than material implication and therefore causes no trouble. Suppose we observe that 'some crow in Stonehaven is black' and that 'some

other crow in Dunoon is also black'. We may decide that the place of observation is irrelevant and that the essential observations are that both crows are black. Now if we know, (from Th) that "craw" is a Scottish word meaning "crow" we may reword the observations, so noting that two craws were black. From this we conclude by a stage of "generalisation of" that all craws are black. We can even, by a final rewording see that all crows are black.

At this point, there occurs an important transition. Rather than dealing with assertions of the form e => f (where e is a conjunction of ground literals, and f is a ground literal) we consider the corresponding clause $\bar{e} \cup \{f\}$. The above definition of generalisation will be mirrored by an analogous one for clauses which will allow a mathematical theory to be developed using reasonably well-known ideas. This theory makes no use of => and consequently some rough justice is dealt to => in the transition.

As a matter of fact, we could have continued to use =>. But we feel that there would be a mismatch between our informal sign, =>, and the increasingly formal nature of the rest of the work. A more formal treatment, perhaps using modal logic, would result in a better-knit theory.

The clauses $C_i = \bar{e}_i \cup \{f_i\}$ are particularly important. We set $H_0 = \{C_i \mid i=1,n\}$. There are some conventions. A clause, C, abbreviates a formula which is a disjunction of all its members. A set of clauses, H, abbreviates a conjunction of all its members. $\forall C$ is the universal

closure of the formula C abbreviates. $\forall H$ is the universal closure of the formula H abbreviates.

A clause, C, is a selection from D iff $C \subseteq D$. If D corresponds to $e_i' \Rightarrow f_i'$ which is a part of some $e_i \Rightarrow f_i$, and C contains f_i' this corresponds to the above definition of selection. We allow C not to contain f_i since this will make rather smoother going, formally. It will not alter the class of chosen hypotheses since C must then be inconsistent with $\bigwedge_{i=1}^{n}(e_i \wedge f_i)$ and so will not be chosen by any of our induction methods. For the same reason the reader will see that neither will any rewording or generalisation of C or indeed any clause reached by continued application of the generalisation operations to C be chosen.

The clauses C and D are said to be rewordings relative to Th, iff $\vdash_{Th} \forall x_1 \ldots\ldots x_n (C \equiv D)$ where $x_1 \ldots\ldots x_n$ are the variables in C or in D. This is equivalent to $\vdash_{Th} C \equiv D$.

Even when C and D are ground, this need not correspond to the previous definition of a rewording. This is the only place where real injustice is done to $\Rightarrow$. However, when Th is empty, $\vdash_{Th} C \equiv D$ iff $C = D$ and then the two definitions correspond properly. The reader may however verify, at the appropriate points in chapter five that for the various types of Th investigated in any detail, the two definitions do in fact correspond.

The clause C is a generalisation of D iff, for some substitution, σ, $C\sigma = D$. This does not quite correspond to the previous

definition. For when D corresponds to $e_i^!$ => $f_i^!$, which is a part of some e_i => f_i, then it may be that C = $\overline{e_i^{\neg}} \cup C'$ where C' is not a singleton. However the correspondence could be straightened out by weakening the demand, implicit in the above definition of a "generalisation from", that $f_i^{\neg}$ be a literal. This would seem to be harmless.

There seem to be several possible definitions of "generalisation from". A weak one, which seems to be a special case of the corresponding definition is:

C is a <u>generalisation from</u> C_i relative to Th, iff there are clauses E and F and a substitution σ such that $C\sigma$ = E, E $\subseteq$ F, F is ground and $\vdash_{Th} F \equiv C_i$.

We may certainly drop the restriction that F is ground, for suppose F has the variables $x_1 \ldots\ldots x_n$ and let $\mu = \{a()/x_1, \ldots., a()/x_n\}$. Then we have $C\sigma\mu = E\mu$, $E\mu \subseteq F\mu$, $F\mu$ is ground and $\vdash_{Th} F\mu \equiv C_i$ since $C_i\mu = C_i$, (C_i is ground). Therefore this definition is a special case of the more general definition:

C is a <u>generalisation from</u> D <u>relative to Th</u> iff for some E and σ , $C\sigma \subseteq E$ and $\vdash_{Th} E \equiv D$.

There is a much stronger general definition: C is a <u>generalisation from</u> D <u>relative to Th</u> iff there are $D_j (j=1,m)$ such that $C = D_1$, $D = D_m$ and $D_j \subseteq D_{j+1}$ or $D_j\sigma = D_{j+1}$ for some σ or $\vdash_{Th} D_j \equiv D_{j+1}$ (for j=1,m-1).

This turns out to be equivalent to the weaker definition. We show this by induction on m. Suppose m=1. Then as $C\epsilon \subseteq C$ and $\vdash_{Th} C \equiv C$, the condition is satisfied. Suppose j>1, then by induction there is a σ and an E such that $D_2\sigma \subseteq E$ and $\vdash_{Th} E \equiv D$. Suppose $C\mu = D_2$ for some μ. Then $C\mu\sigma \subseteq E$ and so the condition is satisfied for C and D. Suppose $C \subseteq D_2$, then $C\sigma \subseteq D_2\sigma \subseteq E$ and the condition is satisfied in this case too. Suppose, finally, that $\vdash_{Th} C \equiv D_2$. Then $\vdash_{Th} C\sigma \equiv D_2\sigma$ and so $\vdash_{Th} C\sigma \cup E \equiv D_2\sigma \cup E$. But since $D_2\sigma \subseteq E$, $\vdash_{Th} C\sigma \cup E \equiv E$. Now, as $\vdash_{Th} E \equiv D$, $\vdash_{Th} C\sigma \cup E \equiv D$ and we see in this last case that the condition is satisfied.

We are thereby justified in adopting the simpler condition as our definition. It is worth introducing some extra symbolism. By $C \underline{\leq} D$ (Th) (read C generalises D relative to Th), we mean that C and D satisfy the above condition.

We can now forget the notions of selection and rewording and the distinction between "generalisation from" and "generalisation of". All these notions served only to help establish our notion of relative generalisation.

As an example let us formalise the crows. We can take $f = \{\text{Black(crow1)}, \text{Black(crow2)}\}$, Ev is given by:

$$\text{Ev(Black(crow1))} = \text{Crow(crow1)} \wedge \text{Place(crow1,Stonehaven)}$$

$$\text{Ev(Black(crow2))} = \text{Crow(crow2)} \wedge \text{Place(crow2,Dunoon)}.$$

We may suppose that Th includes the statement:

$\forall x(Crow(x) \equiv Craw(x))$.

Then $\{ \neg Craw(x), Black(x)\} \leq \{ \neg Crow(crow1),$
$Place(crow1,Stonehaven), Black(crow1)\}$ (Th).

When Th is empty, then $\vdash_{Th} C \equiv D$ iff either C = D or both C and D are tautologies. Then $C \underline{\leq} D$ ($\emptyset$) iff there is a σ such that $C\sigma \subseteq D$ or D is a tautology. For if D is a tautology then $\vdash_{Th} C\sigma \cup D \equiv D$, no matter what σ is. Let us write $C \underline{\leq} D$ (read C generalises D) iff there is a σ such that $C\sigma \subseteq D$. (In the literature, this relation is called subsumption.) Therefore $C \underline{\leq} D$ is not identical to $C \underline{\leq} D$ ($\emptyset$). Practically, however, there is no difference since assumption 1 (that no f_i follows from $Ev(f_i)$) ensures that no C_i is a tautology.

Rather than assume that one generalisation can consistently explain all the phenomena, we look for a set of generalisations which do. That is we expect to find a set, H, of clauses such that for every C_i there is a C in H so that $C \underline{\leq} C_i$ (Th). This allows for the possibility that several distinct classes of phenomena have been bunched together in f by mistake. We adopt the following notation: $H_1 \underline{\leq} H_2$ (Th) (read H_1 generalises H_2 relative to Th) iff for every C_2 in H_2 there is a C_1 in H_1 such that $C_1 \underline{\leq} C_2$ (Th).

$H_1 \underline{\leq} H_2$ (read as H_1 generalises H_2) iff for every C_2 in H_2 there is a C_1 in H_1 such that $C_1 \underline{\leq} C_2$.

We may now define the notion of explanation which will be used.

Recall that $C_i = \bar{e}_i \cup \{f_i\}$ for i=1,n and that $H_0 = \{C_i | i=1,n\}$.

$\forall_H$ explains $\{e_i \Rightarrow f_i | i=1,n\}$ iff

1) For some C in H, $C \underline{\leq} C_i$ (Th), given any i between 1 and n.
2) $\forall H \wedge Th \wedge Irr \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)$ is consistent.

We do not need a condition corresponding to **E2** of the previous chapter, since it has already been assumed. Condition **E4** is also redundant since we have shown that $\forall_H$ is lawlike. This discussion of **E4** only applies when $\Rightarrow$ represents an experiment.

We can write the conditions that $\forall_H$ explains every phenomenon, as

1) $H \underline{\leq} H_0$ (Th)
2) $H \wedge Th \wedge Irr \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)$ is consistent.

Only one thing is now left unspecified, the niceness relation, $\nsim$. Several different niceness relations will be considered most of which are constructed from quite simple syntactic measures. However most attention will be paid to one in particular, $\nsim_{cpg}$. Others will be considered, particularly when we wish to show in chapter 4 that an unsolvability result is largely independent of the choice of $\nsim$; some philosophical discussion in chapter 5 will also use a different niceness relation from $\nsim_{cpg}$.

In order to 'mix' quasi-orderings, (that is, reflexive and transitive binary relations) we define the lexicographic product $\succcurlyeq \circ \succcurlyeq'$ of two quasi-orderings $\succcurlyeq$ and $\succcurlyeq'$, thus

$H_1 \succcurlyeq \circ \succcurlyeq' H_2$ iff either $H_1 \succcurlyeq H_2$ and $H_2 \not\succcurlyeq H_1$ or else $H_1 \succcurlyeq H_2$, $H_2 \succcurlyeq H_1$ and $H_1 \succcurlyeq' H_2$.

Thus to decide which is nicer, the lexicographic product, $\succcurlyeq \circ \succcurlyeq'$ first consults $\succcurlyeq$ and if that gives no definite decision, tries $\succcurlyeq'$. We also say that $\succcurlyeq \circ \succcurlyeq'$ is a lexicographic refinement of $\succcurlyeq$.

Lemma 1 The lexicographic product $\succcurlyeq \circ \succcurlyeq'$ is itself a quasi-ordering. If $\succcurlyeq$ and $\succcurlyeq'$ are linear, so is $\succcurlyeq'$. The lexicographic product is associative and idempotent.

Proof Reflexivity is obvious. Suppose $H_1 \succcurlyeq \circ \succcurlyeq' H_2$ and $H_2 \succcurlyeq \circ \succcurlyeq' H_3$. If $H_1 \succcurlyeq H_2$ and $H_2 \not\succcurlyeq H_1$ then $H_1 \succcurlyeq H_3$ and $H_3 \not\succcurlyeq H_1$, since $\succcurlyeq$ is transitive. It follows, in this case, that $H_1 \succcurlyeq \circ \succcurlyeq' H_2$. Suppose $H_1 \succcurlyeq H_2$, $H_2 \succcurlyeq H_1$, $H_2 \succcurlyeq H_3$ and $H_3 \not\succcurlyeq H_2$. Then $H_3 \not\succcurlyeq H_1$ and so here, too, $H_1 \succcurlyeq \circ \succcurlyeq' H_3$. The only other case is when $H_1 \succcurlyeq H_2$, $H_2 \succcurlyeq H_1$, $H_1 \succcurlyeq' H_2$, $H_2 \succcurlyeq H_3$, $H_3 \succcurlyeq H_2$ and $H_2 \succcurlyeq' H_3$. In this case, as $\succcurlyeq'$ is transitive, $H_1 \succcurlyeq H_3$, $H_3 \succcurlyeq H_1$ $\succcurlyeq' H_3$ and therefore $H_1 \succcurlyeq \circ \succcurlyeq' H_3$.

Suppose $\succcurlyeq$ and $\succcurlyeq'$ are linear. Suppose that $H_1 \succcurlyeq \circ \succcurlyeq' H_2$ is false. Then firstly $H_1 \not\succcurlyeq H_2$ or $H_2 \succcurlyeq H_1$, and secondly either $H_1 \not\succcurlyeq H_2$ or $H_2 \not\succcurlyeq H_1$ or $H_1 \succcurlyeq' H_2$. Therefore, as $\succcurlyeq$ is linear $H_2 \succcurlyeq H_1$. If $H_1 \not\succcurlyeq H_2$ then $H_2 \succcurlyeq \circ \succcurlyeq' H_1$. If $H_1 \succcurlyeq H_2$ then $H_1 \not\succcurlyeq' H_2$. Therefore, as $\succcurlyeq'$ is linear, $H_2 \succcurlyeq' H_1$. So in this case, too, $H_2 \succcurlyeq \circ \succcurlyeq' H_1$ and

we see that $\mathfrak{g} \circ \mathfrak{g}'$ is linear.

In proving associativity, the fact that $H_1 \not\mathfrak{g}_1 H_2$ implies the falsity of $H_1 \mathfrak{g}_1 \circ \mathfrak{g}_2 H_2$, is useful. Suppose $H_1 \not\mathfrak{g} H_2$. Then $H_1 \mathfrak{g} \circ (\mathfrak{g}' \circ \mathfrak{g}'') H_2$ cannot hold, nor can $H_1 \mathfrak{g} \circ \mathfrak{g}' H_2$ and so neither can $H_1 (\mathfrak{g} \circ \mathfrak{g}') \circ \mathfrak{g}'' H_2$. Suppose $H_1 \mathfrak{g} H_2$ and $H_2 \not\mathfrak{g} H_1$. Then $H_1 \mathfrak{g} \circ (\mathfrak{g}' \circ \mathfrak{g}'') H_2$. Also $H_1 \mathfrak{g} \circ \mathfrak{g}' H_2$ holds and $H_2 \mathfrak{g} \circ \mathfrak{g}' H_1$ does not hold. Therefore $H_1 (\mathfrak{g} \circ \mathfrak{g}') \circ \mathfrak{g}'' H_2$ holds.

Suppose $H_1 \mathfrak{g} H_2$ and $H_2 \mathfrak{g} H_1$. Then $H_1 \mathfrak{g} \circ (\mathfrak{g}' \circ \mathfrak{g}'') H_2$ holds iff $H_1 \mathfrak{g}' \circ \mathfrak{g}'' H_2$ holds. Similarly $H_1 \mathfrak{g} \circ \mathfrak{g} H_2$ holds iff $H_1 \mathfrak{g}' H_2$ holds and $H_2 \mathfrak{g} \circ \mathfrak{g}' H_1$ holds iff $H_2 \mathfrak{g}' H_1$ holds. Therefore $H_1 (\mathfrak{g} \circ \mathfrak{g}') \circ \mathfrak{g}''$ holds iff $H_1 \mathfrak{g}' \circ \mathfrak{g}'' H_2$ holds iff $H_1 \mathfrak{g} \circ (\mathfrak{g}' \circ \mathfrak{g}'') H_2$ holds. In all cases we see that $H_1 (\mathfrak{g} \circ \mathfrak{g}') \circ \mathfrak{g}'' H_2$ holds iff $H_1 \mathfrak{g} \circ (\mathfrak{g}' \circ \mathfrak{g}'') H_2$ holds, which concludes the proof of associativity.

$H_1 \mathfrak{g} \circ \mathfrak{g} H_2$ holds iff $H_1 \mathfrak{g} H_2$ and either $H_2 \not\mathfrak{g} H_1$ or $H_1 \mathfrak{g} H_2$. This condition is evidently equivalent to requiring that $H_1 \mathfrak{g} H_2$, which proves idempotency and concludes the proof.

Here are a few quasi-orderings which give possible measures of niceness.

1 Complexity $H_1 \mathfrak{g}_c H_2$ iff $||H_1|| \leq ||H_2||$.

2 Power Let $\text{Power}(C) = ||\{C_i \in H_0 \mid C \leq C_i \ (\text{Th})\}||$,

$$\text{Power}(H) = \sum_{C \in H} \text{Power}(C),$$

$H_1 \mathfrak{g}_p H_2$ iff $\text{Power}(H_1) \geq \text{Power}(H_2)$.

This ordering is defined in terms of H_0 and so of Ev and f.

3 Generality $H_1 \preceq_g H_2$ iff $H_2 \leq H_1$ (Th).

4 Literals $H_1 \preceq_l H_2$ iff $||\bigcup H_1|| \leq ||\bigcup H_2||$.

5 Literal occurrences $H_1 \preceq_{l'} H_2$ iff $\sum_{C \in H_1} ||C|| \leq \sum_{C \in H_2} ||C||$.

6 Symbols $H_1 \preceq_s H_2$ iff the number of symbols in H_1 is less than or equal to that in H_2.

7 Symbol occurrences $H_1 \preceq_{s'} H_2$ iff the number of symbol occurrences in H_1 is less than or equal to that in H_2.

As remarked above, we will mostly be interested in $\preceq_{cpg}$ which is $\preceq_c \circ \preceq_p \circ \preceq_g$.

This choice may be partially justified. The niceness criterion is $(\preceq_c \circ \preceq_p) \circ \preceq_g$ and corresponds to preferring the least general of the simplest, given by $\preceq_{cp} = \preceq_c \circ \preceq_p$, hypotheses. There are three conflicting properties guiding the choice of hypotheses current in the literature. One would want a hypothesis to be justifiable and the less general it is the more it is likely to be justifiable. However it is quite clear that the least general hypothesis which explains the phenomena is H_0, which is even weaker than the statement of the phenomena themselves. Another factor is the desire that the hypothesis be as general as possible in order to say something interesting. However there is no most general hypothesis since one can keep on adding irrelevant clauses to H. The first factor would, we suppose, be emphasized by Carnapians and the second by Popperians. But we see that the first leads to no interesting hypotheses whatever and the second does not seem to lead anywhere. The solution we adopt, as urged by Goodman (1961) is to place simplicity, the third factor, in first place.

So we only debate whether to follow the counsel of safety or strength among hypotheses of equal simplicity. We will discuss this in more detail in chapter 5 when we have a little more mathematical equipment.

As can be seen, in choosing $\mathfrak{s}_{epg}$ we have decided on the safest of the simplest hypotheses. Essentially this is an ad hoc decision. We can scarcely decide between Carnap and Popper, and some choice had to be made. We chose the Carnapian one, since it allows us to prove some theorems to the effect that the hypotheses formed in this way will be acceptable to certain Carnapians in a precise technical sense (Hintikka and Hilpinen, 1966, Hilpinen, 1968).

We should now give some justification of the simplicity measure, $\mathfrak{s}_{cp}$. This is based on both a loose analogy with a measure for propositional formulae and also, simply, mathematical convenience. Propositional formulae in conjunctive normal form are often compared by preferring first those with fewer conjuncts and of those with the same number of conjuncts that one is preferred which has the fewest number of literals. This analogy would favour $\mathfrak{s}_c \circ \mathfrak{s}_1$, rather than $\mathfrak{s}_c \circ \mathfrak{s}_p$. However power has the property that $H_1 \leq H_2$ implies $H_1 \; \mathfrak{s}_p \; H_2$, which is not shared by $\mathfrak{s}_1$, although for propositional formulae both $\mathfrak{s}_1$, and $\mathfrak{s}_p$ have this property. Thus the choice of $\mathfrak{s}_{cp}$ is dictated by rather ad hoc factors. A fuller study would try the effect of other obvious measures of simplicity in more detail and use any relevant philosophical work on simplicity.

We have now specified or indicated the possible values of every

parameter and can give the formal problem of finding a nicest explanatory generalisation from experience.

One is given k = Th $\wedge$ Irr, Ev and f subject to the restrictions:

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

2 For no i does $e_i \rightarrow f_i$ follow from Th.

One is required to find a set of clauses, H, such that

P1 $H \leq_{O} H$ (Th).

P2 $\forall_H \wedge$ Th $\wedge$ Irr $\wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)$ is consistent.

P3 Of all the sets of clauses satisfying P1 and P2, H is minimal with respect to $\mathcal{S}$.

We will be particularly interested in the case where $\mathcal{S}$ is $\mathcal{S}_{cpg}$. When Th is empty, the problem may be reformulated by replacing P1 by:

P1' $H \leq H_O$.

The next chapter develops the relevant formal properties of relative generalisation. In order to make it formally self-contained, several definitions are repeated. This chapter has been concerned with giving a fairly rational explanation for the choices of the definitions.

Chapter 3 The mathematics of generalisation

1. Preliminaries

1.1 Notation

Our notation is that of Robinson (1965), with some additions. A word is a literal or a term. The symbols V, V_1, W,.... are used as meta-linguistic variables ranging over words. The symbol ϕ is used as a meta-linguistic variable ranging over predicate symbols and function symbols. Var(V) is the set of variables occurring in V. Suppose that some property can be shown to hold for variables and constants and that wherever it holds for terms $t_1,\ldots,t_n$ it holds for $\phi(t_1,\ldots,t_n)$. The property can then be seen to hold for all words. This method of proof is called induction on words. A translation is a substitution of the form, $\tau = \{y_1/x_1,\ldots,y_n/x_n\}$ where the y_i are all distinct. The inverse of τ is $\tau^{-1} = \{x_1/y_1,\ldots,x_n/y_n\}$. Note that $(\tau^{-1})^{-1} = \tau$ and if $\mathrm{Var}(V) \subseteq \{x_i \mid i=1,n\}$, then $V\tau\tau^{-1} = V$. The Greek letters τ, ξ and η are reserved as meta-linguistic variables ranging over translations. The translation, τ, standardises W and V apart iff $W\tau$ and V have no common variables. Similarly τ standardises the clauses C and D apart iff $C\tau$ and D have no common variables.

The words W and V are alphabetic variants iff there is a translation τ such that $W\tau = V$. Alphabetic variance is an equivalence relation. Similarly, C and D are alphabetic variants iff there is a translation τ such that $C\tau = D$. Again, this defines an equivalence

relation.

We denote sequences of integers, perhaps empty, by the symbols, $I, J, \ldots$.

The term t is in the Ith place in the word W iff:

when $I = \langle\rangle$, $t = W$ or else

when $I = \langle i_1, \ldots, i_n \rangle$ $(n>0)$ then W has the form $\phi(t_1, \ldots, t_m)$ $(m>0)$ and $i_1 \leq m$ and t is in the $\langle i_2, \ldots, i_n \rangle$th place in t_{i_1}.

For example, x is in the $\langle\rangle$th place in x, the $\langle 2 \rangle$th place in $g(y,x)$ and the $\langle 3,2 \rangle$th place in $P(a,b,g(y,x))$.

Note that a term t is never in the $\langle\rangle$th place in a literal, **L**.

1.2 Generalisation

We say that $W \leq V$ (read W generalises V) iff there is a σ such that $W\sigma = V$. In the literature on automatic theorem proving, $W \leq V$ is usually read as "V is an instance of W".

Example $P(x,x,f(g(y))) \leq P(k(a()), k(a()), f(g(x)))$

We can take $\sigma = \{k(a())/x, x/y\}$

If $W \leq V$, there is a unique minimal σ_0 such that $W\sigma_0 = V$. For let σ be such that $W\sigma = V$ and let $\sigma_0 = \{t/x \mid t/x \in \sigma$ and x appears in $W\}$. Then $W\sigma_0 = V$.

To see that σ_0 is minimal suppose $W\mu = V$. Let $t/x \in \sigma_0$.

It follows that x appears in W. A simple induction on words shows that for any W' if W' σ_0 = W'μ and x appears in W' then x σ_0 = xμ. Applying this to W, we see that t/x ϵ μ. Hence $\sigma_0 \subseteq \mu$. Uniqueness is evident.

We say that C$\leq$D (read C generalises D) iff there is a σ so that C$\sigma \subseteq$ D. In the literature on automatic theorem proving, C$\leq$D is usually read as "C subsumes D". If we identify literals, for the moment, with their unit sets, $\leq$ on literals is just the restriction of $\leq$ on clauses to literals. Robinson (1965) shows that subsumption is decidable. It follows that generalisation on literals and on words are as well.

Generalisation on clauses is a quasi-ordering. It is reflexive as C$\epsilon \subseteq$ C. It is transitive, for suppose that C$\sigma \subseteq$ D and D$\mu \subseteq$ E. Then C$\sigma\mu \subseteq$ D$\mu \subseteq$ E, as required. It follows that generalisation on literals is also a quasi-ordering. It is easy to see, then, that generalisation on words is a quasi-ordering.

One can show, much as above, that if C$\leq$D, there is a unique finite set $\{\sigma_0, \sigma_1, \ldots, \sigma_n\}$ of distinct substitutions such that C $\sigma_i \subseteq$ D($0 \leq i \leq n$), and for all different i and j $\sigma_i \not\leq \sigma_j$, and if C$\mu \subseteq$ D then $\mu \geq \sigma_i$ for some i.

The algorithm for deciding subsumption is easily extended to one which will generate this set.

Example $\{P(x), P(f())\} \leq \{P(f())\}$. We can take σ = $\{f()/x\}$.

We say that $H_1 \leq H_2$ (read H_1 generalises H_2) iff for every D in H_2, there is a C in H_1 such that $C \leq D$.

Example $\{\{Q(h())\}, \{P(x), P(f())\}, \{P(x),Q(x)\}\} \leq \{\{P(f())\}, \{P(g()), Q(g())\}\}$

Evidently, this is a decidable relation.

1.3 Relative generalisation

We say that the literal L generalises M relative to Th iff there is a σ so that $\vdash_{Th} L\sigma \equiv M$. This is written as $L \leq M$ (Th). For the terms t and u, relative generalisation is defined by: $t \leq u$ (Th) iff there is a σ such that $\vdash_{Th} t\sigma = u$. When Th is empty, this reduces to the previous definition of generalisation between words. For $\vdash L\sigma \equiv M$ holds iff $L\sigma = M$ and $\vdash t\sigma = u$ holds iff $t\sigma = u$.

For clauses relative generalisation may be defined by: $C \leq D$ (Th) iff there is an E such that $\vdash_{Th} E \equiv D$ and $C \leq E$. An equivalent definition is: $C \leq D$ (Th) iff there is a σ so that $\vdash_{Th} C\sigma \rightarrow D$. Suppose $C \leq E$ and $\vdash_{Th} E \equiv D$. There is a σ such that $C\sigma \subseteq E$. Then $\vdash C\sigma \rightarrow E$ and so $\vdash_{Th} C\sigma \rightarrow D$. Conversely, suppose $\vdash_{Th} C\sigma \rightarrow D$. Then $\vdash_{Th} C\sigma \vee D \equiv D$, and $C \leq C\sigma \vee D$. We can, therefore, take $E = C\sigma \cup D$. So the definitions are equivalent. When Th is empty and D is not a tautology, the definition reduces to the previous definition of generalisation between clauses, as was seen in chapter 2. The relation of "equivalence, provable from Th" is a congruence for relative generalisation. Suppose that $\vdash_{Th} C \equiv C'$ and $C \leq D$ (Th). There is a

σ so that $\vdash_{Th} C\sigma \rightarrow D$ and further, $\vdash_{Th} C\sigma \equiv C'\sigma$. Consequently $\vdash_{Th} C'\sigma \rightarrow D$ and $C' \leq D$ (Th). Suppose, conversely, $\vdash_{Th} D \equiv D'$ and $C \leq D$ (Th). There is an **E** so that $C \leq \mathbf{E}$ and $\vdash_{Th} \mathbf{E} \equiv D$. Then, $\vdash_{Th} \mathbf{E} \equiv D'$ and $C \leq D'$.

Relative generalisation is a quasi-ordering. Since $\vdash_{Th} C\epsilon \rightarrow C$, $C \leq C$ (Th). Suppose $C \leq D \leq \mathbf{E}$ (Th). There are σ, μ so that $\vdash_{Th} C\sigma \rightarrow D$ and $\vdash_{Th} D\mu \rightarrow \mathbf{E}$. Then, as $(C\sigma \rightarrow D)\mu$ is $C\sigma\mu \rightarrow D\mu$, $\vdash_{Th} C\sigma\mu \rightarrow D\mu$ and $\vdash_{Th} C\sigma\mu \rightarrow \mathbf{E}$. So $C \leq \mathbf{E}$ (Th). It follows that relative generalisation on words is a quasi-ordering.

Define equivalence, relative to Th, by: $C \sim D$ (Th) iff $C \leq D$ (Th) and $D \leq C$ (Th). Since relative generalisation is a quasi-ordering, relative equivalence is an equivalence relation. If $\vdash_{Th} D \equiv D'$, then $D \sim D'$ (Th). The converse does not hold, as will be seen. We also define equivalence by $C \sim D$ iff $C \leq D$ and $D \leq C$. As generalisation is a quasi-ordering, equivalence is an equivalence relation. If $C \sim D$ then $C \sim D$ ($\emptyset$). On the other hand if $C \sim D$ ($\emptyset$) then either C and D are both tautologies or else $C \sim D$. The following lemma is well-known (Robinson, 1965), and so no proof is given.

Lemma 1 $U \sim W$ iff they are alphabetic variants.

We see that $P(x) \sim P(y)$, although it is not the case that $\vdash P(x) \equiv P(y)$. A characterisation of equivalence of clauses will be given later.

There is an interesting reformulation of the definition of relative

generalisation. It is shown that $C \leq D$ (Th) iff D is a tautology or else can be obtained from $\{C\} \cup$ Th by means of a special sort of derivation.

A derivation in which binary resolution is the sole deduction rule, is a C-derivation iff no two descendants of an occurrence of C at a tip are resolved together. (We assume a knowledge of some standard formulation of derivation trees, such as that of Andrews (1968)).

Let Th' be the set of Skolemisations of members of Th. We assume that none of the Skolem function symbols in Th' occur in C or D. Notice that Th' is a conservative extension of Th. That is, in this case, if a formula A does not contain any of the Skolem constants in Th' then $\vdash_{Th} A$ iff $\vdash_{Th'} A$. It follows that for any C and D, $C \leq D$ (Th) iff $C \leq D$ (Th').

There is one other useful fact. If $\vdash_{Th} A$ then $C \leq D$ (Th) iff $C \leq D$ (Th $\cup \{A\}$), for any C, D, Th and A. We define $\mathcal{R}(E,D) = \{C \mid C$ is a resolvent of E and D$\}$.

Lemma 2 If $D_1 \leq D$ (Th) and $D_1 \in \mathcal{R}(E,D_2)$ then $D_2 \leq D$ (Th $\cup \{\forall E\}$).

Proof As $D_1 \in \mathcal{R}(E,D_2)$ we can write E as $E' \cup E''$ and D_2 as $D_2' \cup D_2''$ and find a σ and a μ such that $E''\sigma$ and $D_2''\mu$ are unit sets containing complementary literals and $D_1 = E'\sigma \cup D_2'\mu$.

As $D_1 \leq D$ (Th) there is a λ such that $\vdash_{Th} D_1\lambda \rightarrow D$.

Now $\vdash E\sigma \wedge D_2\mu \rightarrow D_1$. Therefore, $\vdash E\sigma\lambda \wedge D_2\mu\lambda \rightarrow D_1\lambda$.

It follows that $\vdash_{Th} E\,\sigma\lambda \rightarrow (D_2\,\mu\lambda \rightarrow D)$ and so $\vdash_{Th} \cup \{\forall E\}\ D_2\,\mu\lambda \rightarrow D$. As this means that $D_2 \leq D\ (Th \cup \{\forall E\})$, the proof is finished.

Theorem 1 $C \leq D\ (Th)$ iff D is a tautology or there is a C-derivation from $Th' \cup \{C\}$ of a clause D' which subsumes D.

Proof Sufficiency If D is a tautology then $\vdash_{Th} C\,\varepsilon \rightarrow D$ and so $C \leq D\ (Th)$.

In any C-derivation there are either no occurrences of C on a tip or else there is exactly one occurrence. If there are none in the derivation given in the hypothesis then $\vdash_{Th'} D$ and so $\vdash_{Th} D$ as Th' is a conservative extension of Th and no Skolem function symbol of Th' occurs in D. Therefore $\vdash_{Th} C\,\varepsilon \rightarrow D$ which implies that $C \leq D\ (Th)$.

Suppose there is exactly one occurrence of C at a tip of the derivation given by the hypothesis. There must then be clauses $E_i (i=1,n;\ n \geq 0)$ derivable from Th' such that

$$D' \in \mathcal{R}(E_n, (\mathcal{R}(E_{n-1}, \ldots, \mathcal{R}(E_1, C)\ldots))).$$

As $D' \leq D$, n successive applications of lemma 2 show that $C \leq D\ (Th' \cup \{\forall E_i \mid i=1,n\})$ As $\vdash_{Th'} E_i$ for all i, we see that $C \leq D\ (Th')$ and conclude that $C \leq D\ (Th)$.

Necessity Suppose that $C \leq D\ (Th)$. For some σ, $\vdash_{Th} C\,\sigma \rightarrow D$. Let $C = \{L_i \mid i=1,n\}$. Then $\vdash_{Th} \bar{L}_i\sigma \vee D$ for each L_i. Let Th' be the set of Skolemisations of members of Th. None of the Skolem

function symbols should occur in $C\sigma$ -> D. By the subsumption theorem (Lee 1967, Kowalski 1970) for each L_i either $\bar{L}_i\sigma \vee D$ is a tautology or else there is a derivation of a clause D_i from Th', which subsumes $\{\bar{L}_i\sigma\} \cup D$. We may assume, without loss of generality, that the first alternative holds for $L_1,..,L_m$ and the second for $L_{m+1},..,L_n$.

If D is a tautology, we are finished. If m=n then $C\sigma = \{L_i\sigma \mid i=1,m\} \subseteq D$ and we are also finished. Finally, suppose that m<n.

Then we may express D as $\{L_i\sigma \mid i=1,m\} \cup D$.

Evidently there is a clause in

$$\mathcal{R}(\{\bar{L}_n\sigma\}, \mathcal{R}(.... \mathcal{R}(\{\bar{L}_{m+1}\sigma\}, C\sigma)...))$$

which is a subset of $\{L_i\sigma \mid i=1,m\}$.

Consequently there is a clause, D', in

$$\mathcal{R}(\{\bar{L}_n\sigma\} \cup D, \mathcal{R}(... \mathcal{R}(\{\bar{L}_{m+1}\sigma\} \cup D), C\sigma)...))$$

which is a subset of $D \cup \{L_i\sigma \mid i=1,m\} = D$.

Since C subsumes $C\sigma$ and D_i subsumes $\{\bar{L}_i\sigma\} \cup D$ for $m<i\leq n$, it follows from the contraction theorem (Kowalski, 1970) that there is a C-derivation of a clause D'' from $\{D_i \mid m<i\leq n\} \cup \{C\}$ which subsumes D', and therefore D. As the D_i are derived, in their turn, from Th' we see that there is a C-derivation from Th' $\cup \{C\}$ of a clause D'' which subsumes D.

This concludes the proof.

One can obtain other versions of theorem 1 by using any other resolution principle for which the subsumption theorem holds. For example, one can use M-clash resolution (Kowalski, 1970).

2. Generalisation theory of literals

In this section we establish the theory of the relation, $\leq$, on literals. For technical convenience, the results are often derived for words. The corresponding result for literals is then an immediate specialization.

2.1 Least general generalisations of literals

A least general generalisation of some words is a generalisation which is less general than any other such generalisation. For example a least general generalisation of the literals Black(crow1), Black(crow2) is Black(x). One may, roughly, view this as inducing "Everything is black" from "This crow is black" and "That crow is black". The evident absurdity of such an induction was one of the reasons for considering clauses rather than literals.

Less trivially, we shall show later that a least general generalisation of P(f(a(),g(y)),x,g(y)) and P(h(a(),g(x)),x,g(x)) is P(y,x,g(z)).

Let K be a set of words. We say that W is a least general generalisation of K, abbreviated by W is a l.g.g. of K, iff:

1 For every V in K, $W \leq V$

2 If for every V in K, $W_1 \leq V$, then $W_1 \leq W$.

It follows from requirement 2 and lemma 1, that any two least general generalisations of K are alphabetic variants.

Similarly, one can give a definition relativised to Th. We say that W is a least general generalisation of a set of literals, K relative to Th (abbreviated by W is a l.g.g. of K relative to Th), iff:

1 For every M in K, $W \leq M$ (Th)

2 If, for every M in K, $W_1 \leq M$ (Th), then $W_1 \leq L$ (Th).

Two words are compatible iff they are both terms or have the same predicate letter and sign.

We can define also the least generalisation as a product in the following category. The objects are the words and σ is a morphism from V to W iff $V\sigma = W$ and σ acts as the identity, ϵ, on variables not in V. There is at most one morphism from V to W. If σ is a morphism from V to W, and μ is one from W to U, then their categorical composition is the unique morphism from W to U. It should be noted that the categorical composition is not the same as the standard composition of substitutions defined in Robinson (1965). For example, if W is x, V is f(y), and U is f(g(z)), then σ is $\{f(y)/x\}$ and μ is $\{g(z)/y\}$. The categorical composition of σ and μ is $\{f(g(z))/x\}$, but their standard composition is $\{f(g(z))/x, g(z)/y\}$.

V is the least generalisation of $\{W_1, W_2\}$ iff it is a product of W_1 and W_2 in the above category. The category will be used mainly for expository purposes.

Theorem 1

Every non-empty, finite set of words has a least general

generalisation iff any two words in the set are compatible.

Let W_1, W_2 be any two compatible words. The following algorithm terminates at stage 3, and the assertion made there is then correct.

1. Set V_i to $W_i (i=1,2)$. Set θ_i to $\varepsilon (i=1,2)$. ε is the empty substitution.

2. Try to find terms t_1, t_2 which have the same place in V_1, V_2 respectively and such that $t_1 \neq t_2$ and either t_1 and t_2 begin with different function letters or else at least one of them is a variable.

3. If there are no such t_1, t_2 then halt. V_1 is a least general generalisation of $\{W_1, W_2\}$ and $V_1 = V_2$, $V_i \theta_i = W_i (i=1,2)$.

4. Choose a variable x distinct from any in V_1 or V_2 and wherever t_1 and t_2 occur in the same place in V_1 and V_2, replace each by x.

5. Change θ_i to $\{t_i/x\}\ \theta_i (i=1,2)$.

6. Go to 2.

Example. We will use the algorithm to find a least general generalisation of $\{P(f(a(),g(y)),x,g(y)), P(h(a(),g(x)),x,g(x))\}$.

Initially (at stage 1),

$V_1 = P(f(a(),g(y)),x,g(y))$

$V_2 = P(h(a(),g(x)),x,g(x))$,

and $\theta_1 = \theta_2 = \varepsilon$.

We take $t_1=y$, $t_2=x$ and z as the new variable at stage 2. Then after stage 4,

$$V_1=P(f(a(),g(z)),x,g(z))$$
$$V_2=P(h(a(),g(z)),x,g(z))$$

and after stage 5,

$$\theta_1=\{y/z\},\quad \theta_2=\{x/z\}.$$

Next, we take $t_1=f(a(),g(z)),t_2=h(a(),g(z))$ and y as the new variable at stage 2. After stages 4 and 5,

$$V_1=P(y,x,g(z))=V_2$$
$$\theta_1=\{f(a(),g(z))/y\}\{y/z\}$$
$$=\{f(a(),g(y))/y,y/z\}$$
$$\theta_2=\{h(a(),g(z))/y\}\{x/z\}$$
$$=\{h(a(),g(x))/y,x/z\}.$$

The algorithm then halts at stage 3 with $P(y,x,g(z))$ as the least general generalisation.

Proof Evidently the compatibility condition is necessary. Let $\{W_1,\ldots,W_n\}$ be a finite compatible set of words. If n=1, then the theorem is trivial. Suppose that the algorithm works and that $\inf\{V,W\}$ is the result of applying it to V and W. Then it is easy to see that

$$\inf\{W_1,\inf\{W_2,\ldots,\inf\{W_{n-1},W_n\}\ldots\}\}$$

is a least general generalisation of the set. Hence we need only show that the algorithm works.

The rest of the proof proceeds as follows. In order to avoid a constant repetition of the conditions on t_1, t_2 given in stage 2, we say that t_1 and t_2 are replaceable in V_1 and V_2 iff they fulfil the conditions of stage 2. We also denote by V_1', V_2' the result of replacing t_1 and t_2 in V_1, V_2 by x in the way described in stage 4. To show that the algorithm halts and that when it does $V_1=V_2$, we define a difference function by difference(V_1,V_2) = number of members of the set {I | if t_1, t_2 are both in the Ith place in V_1, V_2 respectively then they are replaceable in V_1 and V_2}. Lemma 1.2 below then shows that every time a pair of replaceable terms is replaced the difference drops. Consequently by lemma 1.1, below it will eventually become zero and when it does, lemma 1.1, below shows that we must have $V_1=V_2$ and the algorithm will then halt. We still have to show that the replacements take us in the correct direction. First of all, $V_i' \leq V_i$ since by lemma 1.3, below, $V_i'\{t_i/x\} = V_i$. It is also immediate from this that when the algorithm halts, $V_i' \theta_i = W_i$. Now suppose that W is any lower bound of $\{W_1,W_2\}$. Then a lower bound V is a product of W_1, W_2 if the diagram of figure 1 can always be filled in along the dotted line, so that it becomes commutative in a unique way.

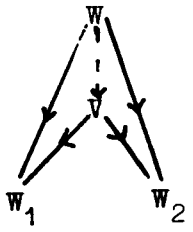


Figure 1

The category is the one defined above. In it there is either one or no morphisms between any two objects and hence it is not necessary in figure 1 to name the morphisms. Indeed, if a diagram can be filled in at all, it can be filled in commutatively and uniquely.

We show in lemma 1.4 below that the diagram on figure 2 can be filled in commutatively.

Thus every time a replacement is made, the V_i' are greater than any lower bound of W_1, W_2. Consequently when the algorithm halts, we have a product. We now give the statements and proofs of the lemmas.

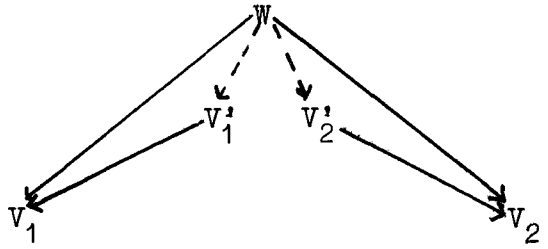


Figure 2

Lemma 1.1 If V_1 and V_2 are distinct compatible words, then there are t_1, t_2 which are replaceable in them.

Proof By induction on words on V_1. If one of V_1, V_2 is a constant or a variable, or if they begin with different function symbols, then V_1, V_2 will do for t_1, t_2 respectively.

If V_1 is $\phi(t_1^1,\ldots,t_n^1)$ and V_2 is $\phi(t_1^2,\ldots,t_n^2)$, then for some i, $t_i^1 \neq t_i^2$ and by the induction hypothesis, applied to t_i^1, there are u_1, u_2 which are replaceable in t_i^1, t_i^2 which have the same place in V_1, V_2

respectively, and so u_1 and u_2 are also replaceable in V_1, V_2.

Lemma 1.2 If V_1, V_2 are distinct compatible words, then $\text{Difference}(V_1',V_2') < \text{Difference}(V_1,V_2)$.

Proof By induction on words on V_1. If one of V_1 or V_2 is a variable or a constant then $t_1=V_1$, $t_2=V_2$ and $V_1'=V_2'=x$, so $0=\text{Difference}(V_1',V_2') < 1 = \text{Difference}(V_1,V_2)$.

If V_1 is $f(v_1,\ldots,v_n)$ and V_2 is $g(u_1,\ldots,u_m)$ where $f\neq g$, then if $t_i=V_i(i=1,2)$, $0=\text{Difference}(V_1',V_2') < \text{Difference}(V_1,V_2)$, by lemma 1.1; otherwise

$$\begin{aligned}\text{Difference}(V_1',V_2') &= 1 + \sum_{i=1,\min(m,n)} \text{Difference}(v_i',u_i') \\ &< 1 + \sum_{i=1,\min(m,n)} \text{Difference}(v_i,u_i) \\ &\qquad \text{(by induction hypothesis, since } m,\ n\neq 0) \\ &= \text{Difference}(V_1,V_2).\end{aligned}$$

In the remaining case where V_1 and V_2 both have the form $\phi(t_1,\ldots,t_n)$, a similar but less complicated argument applies.

Lemma 1.3 $V_i'\{t_i/x\} = V_i(i=1,2)$.

Proof Since V_i' is obtained from V_i by replacing some occurrences of t_i in V_i by x, and since x does not occur in V_i, substituting t_i for x in V_i' will produce $V_i(i=1,2)$.

Lemma 1.4 If V_1, V_2 are distinct compatible words and $V\sigma_i=V_i(i=1,2)$, then there are σ_1', σ_2' so that $V\sigma_i'=V_i'(i=1,2)$.

Proof It is convenient to denote by $f_i(u_1,u_2,t_1,t_2)$ the result of applying the operation of 4 to u_i (with u_1 and u_2 standing for V_1 and V_2) for i=1,2.

Let the variables which occur in V be $y_1,\ldots,y_m$ and suppose that $y_j\ \sigma_i=u_i^j$ (i=1,2; j=1,m) and choose σ_i' so that $y_j\ \sigma_i'=f_i(u_1^j,u_2^j,t_1,t_2)$ (i=1,2; j=1,m).

By lemma 1.3, $y_j\ \sigma_i=y_j\ \sigma_i'\{t_i/x\}$ (i=1,2; j=1,m).

We need only show, by induction on words with variables $y_1,\ldots,y_m$ applied to W that if W, W_1 and W_2 are such that $W\ \sigma_i = W_i$ (i=1,2) then $W\ \sigma_i' = f_i(W_1,W_2,t_1,t_2) = W_i'$, say (i=1,2).

Suppose W is a constant; then $W=W_1=W_2$ and the result is trivial.

Suppose W is the variable y_j. Then $u_i^j=W_i$ (i=1,2) and so:

$$\begin{aligned} W\ \sigma_i' &= y_j\ \sigma_i' = f_i(u_1^j,u_2^j,t_1,t_2) \\ &= f_i(W_1,W_2,t_1,t_2) \\ &= W_i' \text{ (i=1,2).} \end{aligned}$$

Suppose W is $\phi(u_1,\ldots,u_n)$. Then W_i has the form $\phi(w_1^i,\ldots,w_n^i)$ and so W_i' is $\phi(w_1^{i'},\ldots,w_n^{i'})$, say, where $w_l^{i'}=f_i(w_l^1,w_l^2,t_1,t_2)$, (i=1,2; l=1,n). (i=1,2).

$$\begin{aligned} \text{Therefore, } W\ \sigma_i' &= \phi(u_1\ \sigma_i',\ldots,u_n\ \sigma_i') \\ &= \phi(w_1^{i'},\ldots,w_n^{i'}) \text{ (by induction hypothesis)} \\ &= W_i'. \end{aligned}$$

This concludes the induction and the proof.

There is a more efficient version of the algorithm given in the statement of Theorem 1. This involves fewer passes through W_1 and W_2. It is a slight generalisation of the algorithm given by Reynolds (1970).

1 Set V_i to $W_i (i=1,2)$. Set θ_i to ε, the empty substitution $(i=1,2)$.

2 Try to find terms t_1, t_2 which have the same place in V_1 and V_2 respectively and such that $t_1 \neq t_2$ and either t_1 and t_2 begin with different function letters or else at least one of them is a variable.

3 If there are no such t_1, t_2 then halt. V_1 is a l.g.g. of $\{W_1, W_2\}$, $V_1 = V_2$ and $V_i \theta_i = W_i (i=1,2)$.

4 Find a variable x such that $x\theta_i = t_i (i=1,2)$. If there is no such variable, let x be a variable distinct from any in V_1 or V_2.

5 Find an occurrence of t_1 in V_1 and an occurrence, in the same place, of t_2 in V_2 and replace the occurrence of t_1, and the occurrence of t_2 by one of x.

6 Change θ_i to $\{t_i/x\}\theta_i$ $(i=1,2)$.

7 Go to 2.

We shall adopt the convention that $\inf\{W_1, W_2\}$ is the l.g.g. of W_1 and W_2 produced by applying some definite version of one of these algorithms to $\{W_1, W_2\}$.

2.2 A technical lemma

The following lemma is useful in establishing the existence of l.g.g.'s of clauses.

Lemma 1 Let $K = \{W_i | i=1,n\}$ be a set of words with a l.g.g. W and let $\mu_i (i=1,n)$ be substitutions such that $W \mu_i = W_i (i=1,n)$.

1 If t occurs in W, then t is a l.g.g. of $\{t \mu_i | i=1,n\}$.

2 If x, y are variables occurring in W and $x \mu_i = y \mu_i (i=1,n)$, then x=y.

Proof We can assume, without loss of generality, that the μ_i are the minimal substitutions such that $W \mu_i = W_i$.

1 Evidently, t is a generalisation of $\{t \mu_i | i=1,n\}$. Let u be any other and suppose that $u \lambda_i = t \mu_i (i=1,n)$. Let $\tau = \{y_1/x_1, \ldots, y_m/x_m\}$ be a translation such that $x_1, \ldots, x_m$ are the variable symbols of u, and $u\tau$ and W have no common variables. Let W' be W, but with every occurrence of t replaced by one of $u\tau$. Then $\sigma_i = \{x_1 \lambda_i/y_1, \ldots, x_m \lambda_i/y_m\} \cup \mu_i$ is defined, using the minimality of $\mu_i (i=1,n)$. From the construction of σ_i, $W' \sigma_i = W_i (i=1,n)$. Hence there is a ν so that $W' \nu = W$, as W is a l.g.g. of $\{W_i | i=1,n\}$. Hence $u(\tau\nu) = (u\tau)\nu = t$. So t is a l.g.g. of $\{t \mu_i | i=1,n\}$.

2 Suppose that $y \neq x$. Let $W' = W\{y/x\}$.

Then W', W are not alphabetic variants, but $W \leq W'$. Let $W = W_{x,y,y_3,\ldots,y_m}[x,y,y_3,\ldots,y_m]$, where $x,y,y_3,\ldots,y_m$ are the variables occurring in W. We have:

$$W_i = W\,\mu_i = W_{x,y,y_3,\ldots,y_m}[x\,\mu_i, y\,\mu_i, y_3\,\mu_i, \ldots, y_m\,\mu_i]$$

$$= W_{x,y,y_3,\ldots,y_m}[y\,\mu_i, y\,\mu_i, y_3\,\mu_i, \ldots, y_m\,\mu_i]$$

(by hypothesis)

$$= W_{x,y,y_3,\ldots,y_m}[y,y,y_3,\ldots,y_m]\,\mu_i$$

$$= W'\,\mu_i \text{ (by construction) } (i=1,n).$$

This contradicts the fact that W is a least generalisation of $\{W_i \mid i=1,n\}$. Hence y=x.

This completes the proof.

2.3 Lattice properties of literals

It is possible to define the dual of the l.g.g. of two words.

The word, U, is a most general instance (m.g.i.) of V and W iff:

1 $V \leq U$ and $W \leq U$

2 If $V \leq U'$ and $W \leq U'$ then $U \leq U'$.

Note that, by lemma 1.1, any two m.g.i.'s of V and W are alphabetic variants. The relevant theorem here, is the Unification Theorem of Robinson (1965). A version suitable for our present purposes is given in the following lemma.

Lemma 1 (Unification Theorem) If $V\theta = W\theta$ for some substitution θ, there is a most general unifier, (m.g.u.) μ, with the properties:

1 $V\mu = W\mu$

2 For any σ, if $V\sigma = W\sigma$, then there is a λ such that $\sigma = \mu\lambda$.

Robinson gives an algorithm for calculating the m.g.u.

In terms of the category described in section 2.1, there is a close correlation between the co-product and the m.g.u.

Lemma 2 Let $W\tau$ and V have no variables in common. Then $W\tau$ and V are unifiable iff W and V have a co-product. If μ is an m.g.u. of W and V, and U is a co-product, then $U \sim W\tau\mu$.

We leave the rather easy proof to the reader.

In terms of m.g.i.'s, the appropriate remark, again easily proved, is:

Lemma 3 Let $W\tau$ and V have no variables in common. Then $W\tau$ and V are unifiable iff W and V have an m.g.i. If μ is an m.g.u. and U an m.g.i. of W and V, then $W\tau\mu \sim U$.

We can now define a lattice following Reynolds (1970). First the ordering $\leq$ on words is extended by adding special top and bottom elements, $\mathcal{A}$ and $\mathcal{B}$ respectively. So we consider the set $\text{Words}^+ = \{W \mid W \text{ is a word}\} \cup \{\mathcal{A}, \mathcal{B}\}$.

More formally, $\leq$ is defined on Words^+ to be:

$\{\langle W,V\rangle \mid W \leq V\} \cup \{\langle \mathcal{A}, W\rangle \mid W \text{ is a word}\} \cup \{\langle W, \mathcal{B}\rangle \mid W \text{ is a word}\} \cup \{\langle \mathcal{A}, \mathcal{B}\rangle\}$.

For the rest of us the section, we will use, W, W', W_0 etc. to stand for an arbitrary member of Words$^+$.

The following facts are then obvious.

The relation $\leq$ on Words$^+$ is a quasi-ordering. The extension of to Words$^+$, defined by:

$W \sim W'$ iff $W \leq W'$ and $W' \leq W$, is an equivalence relation. If we consider the set of equivalence [W] classes of members of Words$^+$:

$$[W] = \{W' \mid W' \sim W\},$$

then the induced ordering, $\leq$, on equivalence classes given by $[W] \leq [W']$ iff $W \leq W'$ is a well-defined partial ordering.

In fact it is a lattice.

Let us define $\sqcap$ and $\sqcup$ as operations on Words$^+$ by:

1 a $[\mathfrak{A}] \sqcap [W] = [W] \sqcap [\mathfrak{A}] = [\mathfrak{A}]$

b If V and V' have an l.g.g. U then $[V] \sqcap [V'] = [U]$, otherwise $[V] \sqcap [V'] = [\mathfrak{A}]$.

c $[\mathfrak{B}] \sqcap [W] = [W] \sqcap [\mathfrak{B}] = [W]$.

2 a $[\mathfrak{B}] \sqcup [W] = [W] \sqcup [\mathfrak{B}] = [\mathfrak{B}]$.

b If V and V' have an m.g.i. U then $[V] \sqcup [V'] = [U]$, otherwise $[V] \sqcup [V'] = [\mathfrak{B}]$.

c $[\mathfrak{A}] \sqcup [W] = [W] \sqcup [\mathfrak{A}] = [W]$.

Reynolds shows that Words$^+$ is indeed a lattice under the (well-defined) operations $\sqcap$ and $\sqcup$. We will not consider the relativised lattice obtained from the relativised generalisation relation. The lattice is non-modular since:

$$[P(f(x),y)] \leq [P(f(x),f(y))]$$

but:

$$\begin{aligned}
&([P(x,x)] \sqcap [P(f(x),f(y))]) \sqcup [P(f(x),y)] \\
&\quad = [P(x,y)] \sqcup [P(f(x),y)] \\
&\quad = [P(f(x),y)] \\
&\quad \neq [P(f(x),f(y))] \\
&\quad = [P(f(x),f(x))] \sqcap [P(f(x),f(y))] \\
&\quad = ([P(x,x)] \sqcup [P(f(x),y)]) \sqcap [P(f(x),f(y)],
\end{aligned}$$

contradicting the modular equality.

Let $[W]<[W']$ mean $[W]\leq[W']$, but $[W] \neq [W']$. Then $\{[W_i] \mid i\geq 0\}$ is an <u>infinite ascending chain</u> iff $W_i<W_{i+1}$ $(i\geq 0)$. <u>Infinite descending chains</u> are defined similarly. One sees immediately that $[P(f(x))]$, $[P(f(f(x)))]$,$[P(f(f(f(x))))]$..is an infinite strictly ascending chain. It follows easily from Reynolds' work that there are no infinite strictly descending chains and no infinite strictly ascending chains $\{[W_i] \mid i\geq 0\}$ such that every W_i contains no function symbols. On the other hand, the lattice is complete, that is any subset of Words$^+$ has a greatest common instance and a least common generalisation. Later,

we will contrast this situation with that of a lattice of equivalence classes of clauses which is even more irregular than Words^{+}.

3. Generalisation theory of clauses

The theory of the generalisation relation, $\leq$, defined on clauses is developed in this section.

3.1 Elementary properties

The relation $\leq$ is a quasi-ordering and $\sim$ is an equivalence relation. However, two equivalent clauses need not be alphabetic variants. For example, let $C = \{P(x), P(f())\}$ and $D = \{P(f())\}$.

As neither C nor D are tautologies, this gives an example, with $Th = \emptyset$, where $C \sim D\ (Th)$ but $C \neq D$.

We develop a slightly more complicated characterisation of equivalence. The clause C is reduced iff $D \subseteq C$, $D \sim C$ implies that $C = D$. In other words, C is reduced iff it is equivalent to no proper subset of itself.

A clause C is a reduction of a clause D iff it is a reduced subset of D which is equivalent to D.

Lemma 1 Suppose that $C\mu = C$. Then

1) For some integer, $n_0 \geq 1$, $L\mu^{n_0} = L$ for every literal L in C and $x\mu^{n_0} = x$ for every variable x in C.

2) The substitution μ maps distinct variables of C to distinct variables of C.

Proof 1) Define a mapping π : C -> C by $\pi(L) = L\mu$. Since $C\mu = C$, π is onto and therefore, as C is finite, π is a permutation. There is therefore an integer, $n_0 \geq 1$ such that $\pi^{n_0} = \iota$, the identity permutation. So if L is in C, $L\mu^{n_0} = \pi^{n_0}(L) = \iota(L) = L$. Let x be a variable in C. It must occur in some literal, L say, in C. As $L\mu^{n_0} = L$, it follows that $x\mu^{n_0} = x$. (Strictly speaking, this last step requires an easy proof by induction on words.)

2) This is an obvious consequence of 1).

In order to calculate reductions of clauses, a slight variant of the test for subsumption is useful (Robinson, 1965).

Lemma 2 Let C, D and E be clauses and let $\delta = \{a_1()/x_1,\ldots,a_n()/x_n\}$ where $x_1,\ldots,x_n$ are all the variables in E and the a_i are distinct constants. Then there is a σ such that $E\sigma \subseteq C$ and, for all L in D, $L\sigma = L$ iff $E\delta \leq C\delta$.

Proof Suppose there is a σ satisfying the conditions. Then $x_i\sigma = x_i$, for all i and so $\delta\sigma\delta = \sigma\delta$, for $x_i\delta\sigma\delta = a_i() = x_i\delta = x_i\sigma\delta$ and if $y \neq x_i$, for any i, then $y\delta\sigma\delta = y\sigma\delta$. Therefore, $E\sigma\delta = E\delta\sigma\delta \subseteq C\delta$ which shows that $E\delta \leq C\delta$.

Suppose that $E\delta \leq C\delta$. Then there is a substitution μ such that $E\delta\mu \subseteq C\delta$. As $E\delta$ contains no occurrence of an x_i, we may

assume, w.l.o.g., that $x_i \mu = x_i$ for all i. Therefore, if **L** is in **E**, $L\mu = L$. Further, as above, $\delta\mu\delta = \mu\delta$. Therefore, $E\mu\delta = E\delta\mu\delta \subseteq C\delta\delta = C\delta$. As δ maps distinct variables to distinct constants, we see that $E\mu \subseteq C$ which completes the proof.

The question of the existence of a substitution satisfying the conditions of the lemma can, therefore, be answered by using the standard test for subsumption.

Theorem 1 If $C \sim D$ and both C and D are reduced, then they are alphabetic variants. Every clause has a reduction. The following algorithm gives a reduction, E_1, of a given clause C.

1) Set E_1 to $\emptyset$ and E_2 to C.

2) If E_2 is empty, stop.

3) Choose a literal, **L**, in E_2.

4) If there is a substitution σ such that $E_2\sigma \subseteq (E_1 \cup E_2) \setminus \{L\}$ and $M\sigma = M$ for every literal M in E_1, then change E_2 to $E_2\sigma \setminus E_1$. Otherwise remove **L** from E_2 and add it to E_1.

5) Go to 2).

(To implement step 4, one finds a δ as described in lemma 2 and then tests whether $E_2\delta \leq (E_2 \setminus \{L\})\delta$. In doing this one can increase efficiency by noticing that if $E\delta\sigma \subseteq (E_2 \setminus \{L\})\delta$ then $L\delta\sigma$ is in $E_2\delta$. Therefore one need only calculate all the

minimal substitutions $\sigma_j (j=1,s)$ such that $L\,\delta\sigma_j$ is in $E_2\,\delta$, testing for each, in turn, whether $E_2\,\delta\,\sigma_j \leq (E_2 \setminus \{L\})\,\delta$).

Proof Suppose that $C \sim D$ and C and D are reduced. There are substitutions σ and μ such that $C\sigma \subseteq D$ and $D\mu \subseteq C$. Therefore $C\sigma\mu \subseteq D\mu \subseteq C$. As C is reduced, $C\sigma\mu = C$. Therefore, by lemma 1, $\sigma\mu$ maps distinct variables of C to distinct variables of C. Hence σ maps distinct variables of C to distinct variables of D. Now $D \leq C\sigma$ since $D\mu\sigma \subseteq C\sigma$ and therefore, as $C\sigma \subseteq D$ and D is reduced, $C\sigma = D$. Combining this with the fact that σ maps distinct variables of C to distinct variables of D, we see that C and D are alphabetic variants.

Next we show that any clause, C, has a reduction. Let $H = \{D \subseteq C \mid C \leq D\}$. Let C' be a minimal, with respect to $\subseteq$, member of H. We claim that C' is a reduction of C. Certainly $C' \subseteq C$ and $C' \sim C$. If C' is not reduced, it has a proper subset C" which is reduced and is equivalent to C'. But C" must be in H, and this contradicts the minimality of C.

The algorithm always terminates since the number of literals in E_2 always decreases by at least one every time round the loop.

Let E_1^f and E_2^f be the final values of E_1 and E_2 respectively. We show that $E_1^f \subseteq C$ and $C \leq E_1^f$. The first assertion is obvious.

To prove the second, suppose that E_1 and E_2 have the values E_1'

and E_2' at some stage when the execution reaches step 4, and the values E_1'' and E_2'' immediately after step 4. We show that $E_1' \cup E_2' \leq E_1'' \cup E_2''$.

Suppose a suitable σ exists. Then $E_2'' = E_2'\sigma \setminus E_1'$ and $E_1'' = E_1' = E_1'\sigma$. Therefore,

$$(E_1' \cup E_2')\sigma = E_1'\sigma \cup E_2'\sigma = E_1' \cup (E_2'\sigma \setminus E_1')$$
$$= E_1'' \cup E_2''.$$

If no suitable σ exists, $E_1' \cup E_2' = E_1'' \cup E_2''$ since $E_1'' = E_1' \cup \{L\}$ and $E_2'' = E_2' \setminus \{L\}$ for some L.

It follows easily that at every stage of the execution, $E_1 \cup E_2 \leq E_1^f \cup E_2^f = E_1^f$. Applying this to the initial values, $\emptyset$ and C of E_1 and E_2 we see that $C \leq E_1^f$ as required.

We show next that E_1^f is reduced, which will complete the proof. The proof is by induction with the hypothesis that there is at any stage a clause E which is a reduction of $E_1 \cup E_2$ and contains E_1.

This is true initially since $E_1 = \emptyset$ then and we have already shown that every clause has a reduction. Suppose E_1', E_2', E_1'' and E_2'' are as described above and that $E \supseteq E_1'$ is a reduction of $E_1' \cup E_2'$.

Suppose there is a suitable σ as described in step 4. Then $E\sigma \supseteq E_1'\sigma = E_1' = E_1''$. Further $E\sigma \subseteq (E_1' \cup E_2')\sigma = E_1'' \cup E_2'' \subseteq E_1' \cup E_2' \leq E \leq E\sigma$. Therefore $E\sigma \sim (E_1'' \cup E_2'') \sim E$. There is therefore a substitution μ such that $E\sigma\mu \subseteq E$. Therefore $E\sigma\mu \sim E$

and as E is reduced, $E\sigma\mu = E$. By lemma 1, $\sigma\mu$ maps distinct variables of E to distinct variables. So therefore does σ and so E and $E\sigma$ are alphabetic variants. Therefore, $E\sigma$ is reduced, equivalent to $E_1' \cup E_2'$ and contains E_1'.

Suppose there is no such suitable σ. Suppose also that the literal L chosen in step 3 is not in E. Then there is a μ such that $(E_1' \cup E_2')\mu \subseteq E \subseteq (E_1' \cup E_2') \setminus \{L\}$. As $E \subseteq (E_1' \cup E_2')$, $E\mu \subseteq (E_1' \cup E_2')\mu \subseteq E$. Therefore, as E is reduced $E\mu = E$. By lemma 1, there is an integer $n_0 \geq 1$ such that $M\mu^{n_0} = M$ for every literal M in E. Then μ^{n_0} is a suitable substitution in stage 4 as, if M is in E_1, $M\mu^{n_0} = M$ as $E_1 \subseteq E$ and $(E_1' \cup E_2')\mu^{n_0} = (E_1' \cup E_2')\mu \cdot \mu^{n_0-1} \subseteq E\mu^{n_0-1} = E \subseteq (E_1' \cup E_2') \setminus \{L\}$. This contradicts the assumption that L is not in E. Therefore L is in E and so $E_1'' = E_1' \cup \{L\} \subseteq E$ and furthermore, as $E_1'' \cup E_2'' = E_1' \cup E_2'$, E is a reduction of $E_1'' \cup E_2''$ which concludes the inductive proof.

Applying the result to the final stage we see that there is a clause $E \supseteq E_1^f$ which is a reduction of $E_1^f \cup E_2^f = E_1^f$. Therefore $E_1^f \subseteq E \subseteq E_1^f$ and so E_1^f is reduced completing the proof.

Corollary 1 $C \sim D$ iff any two reductions of C and D respectively are alphabetic variants.

Proof Obvious from theorem 1.

It is convenient to develop a characterisation of equivalence

between sets of clauses at this point.

This relation is defined by:

$H \sim H'$ iff $H \leq H'$ and $H' \leq H$.

A set, H, of clauses is <u>reduced</u> iff $H' \subseteq H$ and $H' \sim H$ implies that $H' = H$.

A set, H', of clauses is a <u>reduction</u> of a set, H, of clauses iff H' is reduced, $H' \sim H$ and $H' \subseteq H$.

<u>Theorem 2</u> If $H' \sim H$ and H' and H are reduced then there is a unique bijection θ: H' -> H such that $\theta(C) \sim C$ for every clause, C, in H.

The following algorithm gives a reduction, H', of a given set of clauses H.

1) Set H' to H.

2) Stop if every clause in H' is marked.

3) Choose an unmarked clause, C, in H.

4) If $H' \setminus \{C\} \leq \{C\}$ then change H' to $H' \setminus \{C\}$. Otherwise mark C.

5) Go to 2).

<u>Proof</u> Choose θ : H' -> H so that $\theta(C) \leq C$ for every clause, C, in H'. This is possible as $H' \leq H$.

Similarly one can choose a mapping θ': H -> H' so that $\theta'(C) \leq C$ for every clause, C, in H.

Then, if C is in H', $\theta'(\theta(C)) \leq \theta(C) \leq C$. But H' is reduced and therefore $\theta'(\theta(C)) = C$. Similarly, if C is in H, $\theta(\theta'(C)) = C$. Hence θ is a bijection with inverse θ'. We see too that $C = \theta'(\theta(C)) \leq \theta(C) \leq C$. Therefore $\theta(C) \sim C$.

If θ'' is another bijection from H' to H such that $\theta''(C) \sim C$ for any clause C in H' then it must also have θ' as an inverse by a similar argument to that above. Therefore $\theta = \theta''$.

Since the number of unmarked literals in H' decreases by one every time the execution of the algorithm goes round the loop, the algorithm terminates.

Let H'_f be the final value of H'. It is evident that $H'_f \subseteq H$ and $H'_f \leq H$ and indeed that $H'_f \subseteq H'$ and $H'_f \leq H'$ for any value of H'. To complete the proof we need only show that H'_f is reduced.

If it is not we can find a marked clause C in H'_f such that $H'_f \setminus \{C\} \leq H'_f$. Let H'_1 be the value of H' just before C was marked. Then $H'_1 \setminus \{C\} \supseteq H'_f \setminus \{C\}$, as $H'_1 \supseteq H'_f$, and $H'_f \setminus \{C\} \leq H'_f \leq H'_1$. Therefore $H'_1 \setminus \{C\} \leq H'_1 \leq \{C\}$ which contradicts the fact that C is marked in H'^f_1. This concludes the proof.

Corollary 2 $H_1 \sim H_2$ iff given two reductions H'_1 and H'_2 of H_1 and H_2 respectively there is a bijection, θ, from H'_1 to H'_2 such that

that $\theta(C) \sim C$ for any clause C in H_1.

Proof Obvious from theorem 2.

3.2 Least general generalisations of clauses

A least general generalisation of some clauses is a generalisation which is less general than any other such generalisation. For example, a least general generalisation of the clauses $\{\overline{Crow}(crow1), Black(crow1)\}$ and $\{\overline{Crow}(crow2), Black(crow2)\}$ is $\{\overline{Crow}(x), Black(x)\}$. This may be viewed as an induction from "This crow is black" and "That crow is black" to "All crows are black". This is a much more satisfying phenomenon than our induction of "Everything is black" when we were restricted to literals. We shall give a rather less trivial example later.

The operation of taking a least general generalisation followed by a reduction of the result will play a major role in algorithms for finding nicest explanatory hypotheses. Reductions are taken in order to make the output more perspicuous and in order to reduce demands on the internal storage space of the computer being used to implement such algorithms. Both these points will be illustrated later in this section.

Let H be a set of clauses. The clause, C, is a least general generalisation (which is abbreviated by l.g.g.) of H iff:

1 For every D in H, $C \leq D$.

2 If for every D in H, $C' \leq D$, then $C' \leq C$.

Any two l.g.g's of H are evidently equivalent.

In an analogous way, we say that C is a least general generalisation of H, relative to Th iff:

1 For every D in H, $C \leq D$ (Th).

2 If for every D in H, $C' \leq D$ (Th) then $C' \leq C$ (Th).

One sees that if C and C' are l.g.g.'s of H (relative to Th), then they are equivalent (relative to Th).

By temporarily introducing some auxiliary syntactic concepts, it is possible to give a short demonstration of the existence of least general generalisations of a finite set of clauses. We consider sequences of literals including the null sequence. A literal is identified with that one element sequence whose only member is the literal. The behaviour of sequences of literals is very similar to that of literals.

$\prod_{i=1}^{n} L_i$ is defined to be the sequence with n members whose ith member is L_i. Notice that if n=0, this is the null sequence. We extend the meaning of the $\prod$ operator, so that sequences themselves may be "multiplied", by:

$$\prod_{i=1}^{n} \left(\prod_{j=1}^{m(i)} L_{ij} \right) = L_{11} \ldots L_{1m(1)} \ldots L_{n1} \ldots L_{nm(n)}$$

The reader is left to provide proper formal definitions.

Powers of literals are defined by:

$$L^n = \prod_{i=1}^{n} L_i, \text{ where, for every i, } L_i = L.$$

Application of a substitution to a sequence is defined by:

$$(\prod_{i=1}^{n} L_i)\sigma = \prod_{i=1}^{n} (L_i\sigma).$$

The sequence $\prod_{i=1}^{n} L_i$ is a generalisation of $\prod_{j=1}^{m} M_j$ iff for some σ $(\prod_{i=1}^{n} L_i)\sigma = \prod_{j=1}^{m} M_j$. This is written symbolically as $\prod_{i=1}^{n} L_i \leq \prod_{j=1}^{m} M_j$. Notice that if this relationship does hold then n=m.

We leave the reader to provide a definition of a least general generalisation of a set of sequences of literals and to enlarge the definition of word, place and occurrence given in 3.1.1 to include and apply to sequences of literals. Notice that no word can appear in any place other than the <>th in the null sequence.

Two sequences $\prod_{i=1}^{n} L_i$ and $\prod_{j=1}^{m} M_j$ are compatible iff m=n and for every i between 1 and min(n,m) L_i and M_i are compatible (that is, have the same predicate letter and sign). The next lemma shows how to calculate l.g.g's of sequences and when this is possible.

Lemma 1 Every non-empty finite set of sequences of literals has a least general generalisation iff any two sequences in the set are compatible.

Let $\prod_{i=1}^{n} L_i$ and $\prod_{j=1}^{m} M_j$ be any two compatible sequences of literals.

The algorithm given in theorem 3.2.1.1 may be applied to them. It terminates at stage 3, and the assertion made there is then correct.

Proof We do not give the proof which is directly analogous to that of theorem 3.2.1.1.

Of course one could also use Reynolds' (1970) algorithm, given in section 3.2.1.

The next lemma relates l.g.g's of certain related sets of sequences.

Lemma 2 Let $\prod_{i=1}^{n} L_i$ be a l.g.g. of $\prod_{i=1}^{n} M_i$ and $\prod_{i=1}^{n} N_i$.

1) If π is a permutation of the numbers 1,...,n then $\prod_{i=1}^{n} L_{\pi(i)}$ is a l.g.g. of $\prod_{i=1}^{n} M_{\pi(i)}$ and $\prod_{i=1}^{n} N_{\pi(i)}$.

2) Let n_i be positive integers $(1 \leq i \leq n)$. $\prod_{i=1}^{n} L_i^{n_i}$ is a l.g.g. of $\prod_{i=1}^{n} M_i^{n_i}$ and $\prod_{i=1}^{n} N_i^{n_i}$.

3) If $1 \leq n' < n$ then $\prod_{i=1}^{n'} L_i$ is a l.g.g. of $\prod_{i=1}^{n'} M_i$ and $\prod_{i=1}^{n'} N_i$.

4) Let π be a permutation of the numbers 1,...,n and let n_i be integers $(1 \leq i \leq n)$. $\prod_{i=1}^{n} L_{\pi(i)}^{n_i}$ is a l.g.g. of $\prod_{i=1}^{n} M_{\pi(i)}^{n_i}$ and $\prod_{i=1}^{n} N_{\pi(i)}^{n_i}$.

Proof 1) and 2) are obvious. The proof of 3) is analogous to that of lemma 3.2.2.1. Part 4) follows immediately from the other three parts.

This concludes the necessary supply of lemmas on sequences of literals.

A set of literals $K = \{L_i \mid i=1,n\}$ is a <u>selection</u> from $H = \{C_i \mid i=1,n\}$ iff L_i is in C_i $(i=1,n)$ and any two literals in K are compatible. To see the motivation for this definition, suppose that $\mathbf{E} \leq C_i$ for every i. Then there are σ_i such that $\mathbf{E}\,\sigma_i \subseteq C_i$ for every i. Choose a literal L in $\mathbf{E}$, if it is not empty. $\{L\,\sigma_i \mid i=1,n\}$ is a selection from H.

<u>Theorem 1</u> Every finite set, H, of clauses has a least general generalisation which is not $\emptyset$ iff H has a selection. Let C and D be two clauses with selections $\{M_i, N_i\}$, where $1 \leq i \leq n$, so that C and D have at least one selection. Let $\prod_{i=1}^{n} L_i$ be a l.g.g. of $\prod_{i=1}^{n} M_i$ and $\prod_{i=1}^{n} N_i$. Then $\{L_i \mid i=1,n\}$ is a l.g.g. of C and D.

<u>Proof</u> We demonstrate the last part first. Evidently $\{L_i \mid i=1,n\}$ is a generalisation of C and D. Suppose $\mathbf{E} = \{L_j \mid j=1,m\}$ is a generalisation of C and D. Then there are substitutions σ and μ such that $\mathbf{E}\sigma \subseteq C$ and $\mathbf{E}\mu \subseteq D$. Therefore $\{L'_j\,\sigma, L'_j\,\mu\}$ is a selection from C and D for any j. Consequently we can find a permutation π of the numbers $1,\ldots,n$ and integers $n_i \geq 0$ $(i=1,n)$, such that $\prod_{j=1}^{m} L'_j$ is a generalisation of $\prod_{i=1}^{n} M_{\pi(i)}^{n_i}$ and $\prod_{i=1}^{n} N_{\pi(i)}^{n_i}$. By lemma 2.3, $\prod_{i=1}^{n} L_{\pi(i)}^{n_i}$ is a l.g.g. of the latter two sequences. Therefore $\prod_{j=1}^{m} L'_j \leq \prod_{i=1}^{n} L_{\pi(i)}^{n_i}$ and so $\mathbf{E} \leq \{L_i \mid i=1,n\}$ which proves that $\{L_i \mid i=1,n\}$ is a l.g.g. of C and D.

Notice that if some combination of sign and predicate letter occurs in both C and D then it also occurs in $\{L_i \mid i=1,n\}$. Therefore, if a finite set, H, of clauses has a selection it has a nonempty least general

generalisation obtained by repeatedly taking l.g.g's of pairs of clauses in a similar way to that in the proof of theorem 3.2.1.1. Suppose H has no selection. Then it cannot have a nonempty generalisation. (See the remark after the definition of selection.) Therefore its only, and hence its least, generalisation is $\emptyset$. This concludes the proof.

It follows from lemma 1, theorem 1 and its proof that one can effectively obtain the l.g.g. of a finite nonempty set of clauses. Consequently we can find an effective function inf from sets of clauses to clauses such that if $H \neq \emptyset$ then inf H is a l.g.g. of H and (for convenience) if $H = \emptyset$ then inf H is some fixed tautology. This definition seems to conflict with another definition of inf given in 3.2.1. However, we can define inf so that $\inf\{\{L_i\} \mid i=1,n\} = \inf\{L_i \mid i=1,n\}$, where the new definition is being used on the left and the old on the right of the equation. This follows from the following:

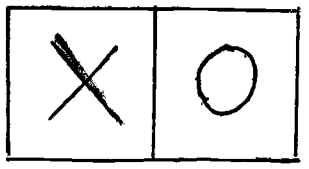
Corollary 1 A literal **L** is a l.g.g. of two literals M and N iff $\{\mathbf{L}\}$ is a l.g.g. of the clauses $\{M\}$ and $\{N\}$.

Proof If $\{\mathbf{L}\}$ is a l.g.g. of $\{M\}$ and $\{N\}$ then it is immediate that **L** is a l.g.g. of M and N. If **L** is a l.g.g. of M and N then $\{M\}$ and $\{N\}$ have a selection and the result follows from the second part of theorem 1.

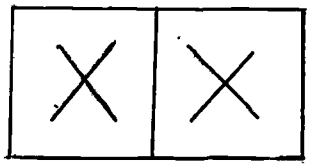
Thus identifying literals with the corresponding one element clauses would not cause any conflict between the two definitions of l.g.g. This is not a trivial result.

Here is a less trivial example than the one presented at the beginning of this section.

Suppose that some two-person game is being played on a board with two squares, 1() and 2() and that the positions in figure 1 are won positions for the first player:



Position $p_1()$



Position $p_2()$

Figure 1

1() is the name of the left hand side square, and 2() of the right hand square; $p_1()$ and $p_2()$ are the names of the positions and O(), X() are the names of the marks O, X. The use of p, n_1, n_2 as variables is temporary. We describe the fact that these positions are wins by means of the following two clauses:

1. $\{\overline{Occ}(1(),X(),p_1()),\ \overline{Occ}(2(),O(),p_1()),\ Win(p_1())\}$.
2. $\{\overline{Occ}(1(),X(),p_2()),\ \overline{Occ}(2(),X(),p_2()),\ Win(p_2())\}$.

The course of the calculation is indicated in table 1.

$\overline{Occ}(1(),X(),p_1())$ $\overline{Occ}(1(),X(),p)$

$\overline{Occ}(1(),X(),p_2())$ $\overline{Occ}(1(),X(),p)$

$\overline{Occ}(1(),X(),p_1())$ $\overline{Occ}(1(),X(),p)$ $\overline{Occ}(n_1,X(),p)$

$\overline{Occ}(2(),X(),p_2())$ $\overline{Occ}(2(),X(),p)$ $\overline{Occ}(n_1,X(),p)$

$\overline{Occ}(2(),O(),p_1())$ $\overline{Occ}(2(),O(),p)$ $\overline{Occ}(2(),O(),p)$ $\overline{Occ}(n_2,O(),p)$ $\overline{Occ}(n_2,x,p)$

$\overline{Occ}(1(),X(),p_2())$ $\overline{Occ}(1(),X(),p)$ $\overline{Occ}(1(),X(),p)$ $\overline{Occ}(n_2,X(),p)$ $\overline{Occ}(n_2,x,p)$

$\overline{Occ}(2(),O(),p_1())$ $\overline{Occ}(2(),O(),p)$ $\overline{Occ}(2(),O(),p)$ $\overline{Occ}(2(),O(),p)$ $\overline{Occ}(2(),x,p)$

$\overline{Occ}(2(),X(),p_2())$ $\overline{Occ}(2(),X(),p)$ $\overline{Occ}(2(),X(),p)$ $\overline{Occ}(2(),X(),p)$ $\overline{Occ}(2(),x,p)$

$Win(p_1())$ $Win(p)$

$Win(p_2())$ $Win(p)$

Table 1

Each vertical column displays on alternate rows the sequences M_1 and M_2 at an instance of stage 2 of the algorithm of theorem 3.2.1.1. We find t_1 and t_2 by searching through M_1 and M_2 from left to right which is top to bottom in the table. As soon as two literals have become the same in a column, we do not mention them in subsequent columns.

Thus the least generalisation is:

$\{\overline{Occ}(1(),X(),p),\overline{Occ}(n_1,X(),p)\overline{Occ}(n_2,x,p),\overline{Occ}(2(),x,p),Win(p)\}$.

We use the algorithm of theorem 3.3.1.1. We can take $L=\overline{Occ}(n_1,X(),p)$ and $\sigma = \{1()/n_1\}$. This gives

$E_2 = C\sigma = \{\overline{Occ}(1(),X(),p), \overline{Occ}(n_2,x,p), \overline{Occ}(2(),x,p), Win(p)\}$ and
$E_1 = \emptyset$.

Next, we can take $L=\overline{Occ}(n_2,x,p)$ and $\sigma = \{2()/n_2\}$ and obtain
$E_2 = C\sigma = \{\overline{Occ}(1(),X(),p), \overline{Occ}(2(),x,p), Win(p)\}$ and $E_1 = \emptyset$.

After this every literal in E_2 goes into E_1 and eventually,
$E_1 = \{\overline{Occ}(1(),X(),p), \overline{Occ}(2(),x,p), Win(p)\}$ and $E_2 = \emptyset$.

The algorithm stops at this point. The final clause says that if a position has an X in hole 1 and hole 2 has something in it, then the position is a win, which, given the evidence, is fairly reasonable.

The main computational weakness in the method for finding a reduced least generalisation lies in that part of the reducing algorithm which requires a test for subsumption. For suppose that we are looking for the l.g.g. of two clauses each with nine literals in a single predicate letter (this can arise in descriptions of tic-tac-toe, say); there will be at least eighty-one literals in the raw l.g.g., and we will have to try to tell whether or not a clause of eighty-one literals subsumes one of eighty.

It may very well be possible to find an algorithm which alternates the processes of finding an l.g.g. and reduction. In this way, in the tic-tac-toe example, for instance, sequences of eighty-one literals simply might not arise. Unfortunately we have not investigated this possibility.

3.4 Lattice properties of clauses

It is possible to define a lattice of equivalence classes of clauses in a way analogous to the construction of the lattice of equivalence classes of literals, given in 3.2.3. Once again, we will not consider the relativised lattice one could obtain using generalisation relative to Th.

The lattice, although not in general modular, does have a limited form of distributivity.

We also give a representation theorem which shows that the general clauses can be reached by a single lattice operation from the ground clauses. However it is not possible in general to take sups or infs of infinite sets and we give some counter-examples. We also give examples of strictly ascending and descending infinite chains of clauses with one binary predicate symbol and no function symbols. This is in strong constrast with the lattice of equivalence classes of literals where such chains (and all strictly descending infinite chains, even with function symbols) are impossible.

The representation theorem in the form of theorem 3.3.1.4. will prove a useful tool in later chapters. Section 3.3.2 on the infinitary properties of the lattice will be a source of counterexamples for various conjectures. The rest of the material is not essential.

3.3.1 Finitary properties

Before we are able to define the lattice, we need the dual of

the notion of an l.g.g. of two clauses.

The clause C is a most general instance (m.g.i.) of D and **E** iff:

1 $D \leq C$ and $\mathbf{E} \leq C$.

2 For all C', if $D \leq C'$ and $\mathbf{E} \leq C'$, then $C \leq C'$.

Evidently, any two m.g.i's of D and **E** are equivalent.

Lemma 1 Let ξ be a translation such that $D\xi$ and **E** have no common variables. Then $D\xi \cup \mathbf{E}$ is an m.g.i. of D and **E**.

Proof 1 $D \leq D\xi \leq D\xi \cup \mathbf{E}$.

$\mathbf{E} \leq D\xi \cup \mathbf{E}$.

2 Suppose $D \leq C'$ and $\mathbf{E} \leq C'$. Then $D\xi \leq D \leq C'$ and so, for some minimal σ and μ, $D\xi\sigma \subseteq C'$ and $\mathbf{E}\mu \subseteq C'$. Since σ and μ are minimal and $D\xi$ and **E** have no common variables, $\sigma \cup \mu$ exists and $(D\xi \cup \mathbf{E})(\sigma \cup \mu) = (D\xi\sigma \cup \mathbf{E}\mu) \subseteq C' \cup C' = C'$.

We may now define the lattice. Let

$[C] = \{C' \mid C' \sim C\}$.

This set can be made into a lattice by the following definitions.

1 $[C] \sqcap [D] = [\inf\{C,D\}]$

2 $[C] \sqcup [D] = [C\xi \cup D]$ where ξ is a translation such that $C\xi$ and D have no common variables.

$\underline{3}$ $[C] \leq [D]$ iff $C \leq D$.

One may see from the definition of inf and lemma 1 and the properties of $\sim$ that this does provide a good definition of a lattice. Notice that since m.g.i's and l.g.g's always exist, it is unnecessary to add any special elements. There is a natural bottom element, $[\emptyset]$, for $\emptyset \leq C$ for all C.

There is, in general, no top element. For example suppose P_i $(i \geq 1)$ is an infinite set of distinct predicate symbols. Then there can be no clause C such that $\{P_i(x)\} \leq C$ for all i. Alternatively let P be a unary predicate symbol and f a unary function symbol. There can be no clause C such that $\{P(f^i(x))\} \leq C$ for all i. However, suppose we consider only the equivalence classes of those clauses whose predicate symbols come from some fixed finite set whose function symbols are all from some fixed finite set and whose terms have depth less than or equal to some fixed integer. This set of equivalence classes forms a sublattice of the lattice of equivalence classes of clauses as may be seen from lemma 1 and the remarks preceding the definition of inf. The sublattice has a top element, namely $[\{L \mid [\{L\}]$ is in the sublattice and the only variable in L is $x\}]$.

The lattice is not modular, in general, for if one takes

$[C] = [\{Q(x), P(f())\}]$

$[D] = [\{Q(x), P(f()), P(x)\}]$

$[E] = [\{Q(f())\}]$

then $C \leq D$ but

$$
\begin{aligned}
&([E] \sqcap [D]) \sqcup [C] \\
&= ([\{Q(f())\}] \sqcap [\{Q(x),P(f()),P(x)\}]) \sqcup [C] \\
&= [\{Q(x)\}] \sqcup [\{Q(x),P(f())\}] \\
&= [\{Q(y),Q(x),P(f())\}] \\
&= [\{Q(x),P(f())\}] \\
&\neq [\{P(f()),P(y),Q(y)\}] \\
&= [\{P(f()),Q(f())\}] \sqcap [\{Q(x),P(f()),P(x)\}] \\
&= [\{Q(x),P(f()),Q(f())\}] \sqcap D \\
&= ([\{Q(f())\}] \sqcup [\{Q(x),P(f())\}]) \sqcap D \\
&= ([E] \sqcup [C]) \sqcap [D]
\end{aligned}
$$

This contradicts the modular equality. The corresponding non-modular sublattice is given in figure 1.

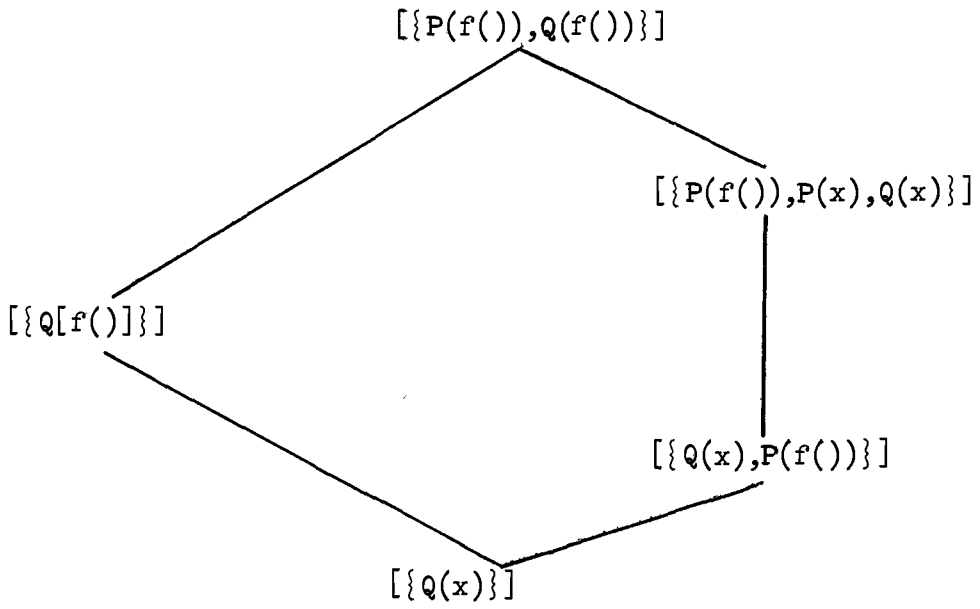


Figure 1

However we have:

Theorem 1 1) If B and C have no common ground terms, then

$$[A] \sqcap ([B] \sqcup [C]) = ([A] \sqcap [B]) \sqcup ([A] \sqcap [C]).$$

2) If A and B have no common ground terms and A and C have no common ground terms then

$$[A] \sqcup ([B] \sqcap [C]) = ([A] \sqcup [B]) \sqcap ([A] \sqcup [C]).$$

Proof 1) First of all, it follows from lattice theory that

$$[E] = ([A] \sqcap [B]) \sqcup ([A] \sqcap [C]) \leq [A] \sqcap ([B] \sqcup [C]).$$

That is, [E] is a lower bound of [A] and [B] $\sqcup$ [C]. We show that it is the greatest and hence that equality holds.

Suppose $[D] \leq [A]$, $[D] \leq [B] \sqcup [C]$. We need only show that $[D] \leq [E]$ to prove the theorem.

As $[D] \leq [B] \sqcup [C]$, there is a σ so that $D\sigma \subseteq B\xi \cup C$ where ξ standardises B and C apart, and so we can set $D = D_1 \cup D_2$ where $D_1\sigma \subseteq B\xi$; $D_2\sigma \subseteq C$. Now if D_1 and D_2 have a common variable x, then $B\xi$ and C have a common term $x\sigma$ which is impossible - for it cannot be ground as B and C have no common ground terms and it cannot be general as ξ standardises B and C apart. Hence D_1 and D_2 have no common variable and so: $[D] = [D_1] \sqcup [D_2]$

Hence as $[D_1] \leq [D] \leq [A]$ and $[D_1] \leq [B]$,

$[D_1] \leq ([A] \sqcap [B])$

Similarly $[D_2] \leq ([A] \sqcap [C])$

Hence $[D] = [D_1] \sqcup [D_2] \leq ([A] \sqcap [B]) \sqcup ([A] \sqcap [C]) = [E]$

which completes the proof.

2) Applying part 1 three times, we get

$$([A] \sqcup [B]) \sqcap ([A] \sqcup [C]) = ([A] \sqcap [A]) \sqcup ([A] \sqcap [C]) \sqcup ([B] \sqcap [A]) \sqcup ([B] \sqcap [C])$$

$[A] \sqcup ([B] \sqcap [C])$, since

$[A] \sqcap [A] = [A]$; $[A] \sqcup ([A] \sqcap [C]) = [A]$

and $[A] \sqcup ([B] \sqcap [A]) = [A]$

by the absortive laws.

Corollary 1 The sublattice containing only the equivalence classes of those clauses which do not have any constant symbols is distributive.

Proof Note that this is a sublattice, since if $A \sim B$ then A has no constant symbols iff B has none and $\sqcap$ and $\sqcup$ do not introduce constant symbols. Distributivity is then immediate from Theorem 1 as B and C have no constant symbols implies that B and C have no ground terms (and hence no common ones).

We conjecture the following converse of theorem 1:

If $[A] \sqcap ([B] \sqcup [C]) = ([A] \sqcap [B]) \sqcup ([A] \sqcap [C])$

for all A, then B and C have no ground terms in common.

However to completely "understand" distributivity (or similarly for any other lattice properties) it would be best to have a necessary and sufficient simple syntactic criterion for A,B,C together for satisfaction of the distributive law.

Representation theorem

We need some definitions:

Let x_1, x_2, ... be all the variable symbols written out as an infinite list.

Let $\delta_n = \{h() / x_1, \ldots, h() / x_n\}$,

$\gamma_n = \{f(h()) / x_1, \ldots, f^n(h()) / x_n\}$

Theorem 2. If C does not contain the unary function symbol f then for all n,

$$[C] = [C\ \delta_n] \sqcap [C\ \gamma_n].$$

Before proving this, we remark that by choosing n large enough, we can express the equivalence class of any clause as the meet of the equivalence classes of two ground clauses. This theorem entails Reynold's Theorem 8 for literals, as can be seen using corollary 3.3.2.1.

However in this special case, one can strengthen Theorem 2 to:

Theorem 3. For every literal L and every n,

$$[\{L\}] = [\{L\}\ \delta_n] \sqcap [\{L\}\ \gamma_n]$$

We do not give the proof of theorem 3 which is a modification of Reynolds' proof of his theorem 8.

In general, one cannot remove the hypothesis of theorem 2. For if

$$C = \{P(x_1),\ P(f(x_1))\},$$

$$\inf\{C\ \delta_1, C\ \gamma_1\} = \{P(x),\ P(y),\ P(f(h())),\ P(f(x))\}$$

$$\sim\{P(x),\ P(f(h())),\ P(f(x))\}$$

$$\not\sim C.$$

We need some more definitions and a lemma for the proof of theorem 2.

Suppose W is a word. Then W' is obtained from W by continued application of (*) until this is no longer possible.

(*) Let n be the largest positive integer such that $f^n(h())$ occurs in W. Replace every occurrence of $f^n(h())$ in W by x_n.

Suppose $\sigma = \{t_1 / y_1, \ldots, t_m / y_m\}$ where the y_i occur amongst the x_i. We define $\sigma' = \{t_1' / y_1, \ldots, t_m' / y_m\}$.

Lemma 2.1 Suppose W does not contain f. Then

1 $(W\ \gamma_n)' = W$

2 $(W\sigma)' = W\ \sigma'$

Proof 1 This is obvious.

2 By induction on words on W.

Proof of theorem 2

Evidently C is a lower bound of C δ_n and C γ_n. We show that if $D \leq C\ \delta_n$ and $D \leq C\ \gamma_n$ then $D \leq C$.

Since $D \leq C\ \delta_n$ and C does not contain f and δ_n does not introduce f, D itself does not contain f.

For some σ, $D\sigma \subseteq C\ \gamma_n$.

Let L be in D. Then for some M in C

$$L\sigma = M\ \gamma_n$$

Hence $L\sigma' = (L\sigma)'$ (lemma 2.1.2)

$= (M\ \gamma_n)'$

$= M$ (lemma 2.1.1).

Hence $D\ \sigma' \subseteq C$, which concludes the proof.

For practical applications another version of the representation theorem is convenient. Let a_{ij} $(i,j \geq 1)$ be a doubly infinite sequence of distinct constant terms chosen so as to leave at least denumerably many constants unused.

Let $\gamma^i_j = \{a_{i1}/x_1, \ldots, a_{ij}/x_j\}$ $(j \geq 1)$.

Theorem 4 Suppose C does not contain any of the a_{ij}. If $i \neq i'$ then for all j, $[C] = [C\ \gamma^i_j] \sqcap [C\ \gamma^{i'}_j]$.

Proof Evidently $C \leq C\ \gamma^i_j$ for all i and j. Let W be a word. W'' is obtained from W by continued application of (**) until this is no longer possible.

(**) Replace an occurrence of a_{ij} by one of x_j

If $\sigma = \{t_1/y_1, \ldots, t_m/y_m\}$ then $\sigma'' = \{t''_1/y_1, \ldots, t''_m/y_m\}$. The rest of the proof is exactly analogous to the proof of theorem 2; the role of the operation, ', being played by the operation, ".

3.3.2 Infinitary properties

It is shown here that the lattice has no pleasant infinite properties. It has strictly ascending and descending chains and is incomplete.

Infinite chains

We write $C < D$ when $C \leq D$ but not $D \leq C$.

$\{C_i \mid i \geq 1\}$ is a strictly ascending chain of clauses.

iff $C_i < C_{i+1}$ $(i \geq 1)$. Strictly descending chains are defined similarly.

We could define $[C] < [D]$ when $[C] \leq [D]$ but not $[D] \leq [C]$ and say that $\{[C_i] \mid i \geq 1\}$ is a strictly increasing chain of equivalence classes of clauses iff $[C_i] < [C_{i+1}]$ $(i \geq 1)$. But since $\{[C_i] \mid i \geq 1\}$ is strictly increasing iff $\{C_i \mid i \geq 1\}$ is, we prefer to deal with clauses rather than equivalence classes of clauses. A similar remark holds for

descending chains.

We will give examples of both ascending and descending chains using positive clauses with a single binary predicate symbol P and no function symbols.

Such a clause can be pictured as a graph whose nodes are the variable symbols appearing in the clause. The graph has an arc from x to y if and only if P(x,y) is a literal in the clause.

Thus to $\{P(x,y), P(y,z), P(z,x)\}$ corresponds the graph:

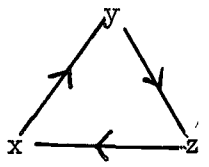


Figure 1

If $C_1\sigma \subseteq C_2$ then σ is a homomorphism of the corresponding graphs and in fact there is a 1-1 correspondence between such substitutions and graph homomorphisms.

We do not use this correspondence in any formal way, but with its help the reader should be able to see the truth of the various theorems.

Here is an example of a strictly ascending chain. Define $\{C_i \mid i \geq 1\}$ by $C_1 = \{P(x_0,x_1)\}$ and $C_i = C_{i-1} \cup \{P(x_{i-1},x_i)\}$ where the x_i are all different. In graph terms, C_i is the chain:

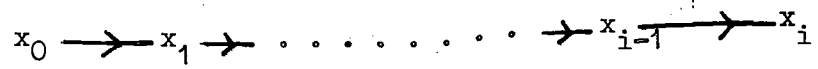


Figure 2

Evidently $C_i \leq C_{i+1}$ as $C_i \subseteq C_{i+1}$. Suppose $C_i\sigma \subseteq C_{i-1}$.

If $x_k\sigma = x_1$, then as $P(x_k,x_{k+1})$ is in C_i, $P(x_1,x_{k+1}\sigma)$ is in C_{i-1}.

Hence $x_{k+1}\sigma = x_{l+1}$ as $P(x_1,x_{l+1})$ is the only literal in C_{i+1} that has x_1 as its first argument. Thus it follows easily that σ is injective - but this contradicts the fact that C_i has i variables and C_{i-1} only i-1. Hence no such σ exists and $C_i \not\leq C_{i-1}$.

This chain is bounded above by $\{P(x,x)\}$ for $C_i\sigma = \{P(x,x)\}$ where $\sigma = \{x / x_0,\ldots,x / x_i\}$. The ascending chain could have been produced in a trivial way by using a unary function, one takes the clauses $\{Q(f^i(x))\}$, or we could use an infinite supply of unary predicate symbols Q_i and take the chain:

$$\{C_i' \mid i \geq 1\} \text{ where } C_1' = \{Q_1(x)\};\quad C_{i+1}' = C_i' \cup \{Q_{i+1}(x)\}$$

With a little more effort we can produce a strictly descending chain. Let $[i,n]$ $(i \geq 0,\ n \geq 1)$ be the equivalence class of i under congruence modulo n. Thus $[0,2]$ is the set of even integers. Let $x_{[i,n]}$ be an infinite supply of variables labelled by the set $\{[i,n] \mid i \geq 0,\ n \geq 1\}$.

We define the positive clause D_n in the binary predicate letter P, with no function symbols and whose variables are a subset of

$\{x_{[i,n]} \mid i \geq 0,\ n \geq 1\}$ by

$P(x_{[i,m]}, x_{[j,l]}) \in D_n$ iff

$m = l = n$ and $j \equiv i+1 \pmod{n}$. D_n contains n literals.

In graph terms, D_n is the cycle:

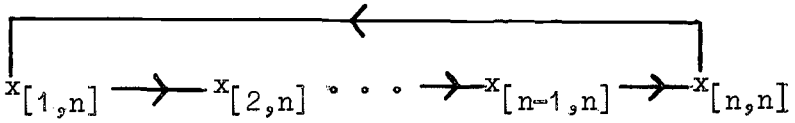


Figure 3

Then we will show that:

$\{D_2n \mid n \geq 1\}$ is a strictly descending chain.

Let γ_n be a substitution such that

$x_{[1,2^{n+1}]}\ \gamma_n = x_{[i,2^n]}$.

Such a γ_n exists as if $i \equiv j \pmod{2^{n+1}}$ then $i \equiv j \pmod{2^n}$.

Then $D_{2^{n+1}}\ \gamma_n = \{P(x_{[i,2^{n+1}]}, x_{[i+1,2^{n+1}]})\ \gamma_n \mid 0 \leq i < n\}$

$= \{P(x_{[i,2^n]}, x_{[i+1,2^n]}) \mid 0 \leq i < n\}$

$= D_{2^n}$.

On the other hand, suppose that σ_n is a substitution such that

$D_{2^n}\ \sigma_n \subseteq D_{2^{n+1}}$. Let $x_{[1,2^n]}\ \sigma_n = x_{[1,2^{n+1}]}$

We will show by finite induction that if

$1 \leq i \leq 2^n$ then $x_{[i,2^n]}\ \sigma_n = x_{[i+l-1,\ 2^{n+1}]}$.

This is true for i=1. Suppose that it holds for i. Then as

$L = P(x_{[i,2^n]}, x_{[i+1,2^n]}) \in D_2n$, it follows that

$L\ \sigma_n = P(x_{[i+l-1,2^{n+1}]}, x_{[j,2^{n+1}]})$ say. Hence by the definition of D_2n+1, $j \equiv i+l \pmod{2^{n+1}}$ so $x_{[i+1,2^n]}\ \sigma_n = x_{[i+1,2^{n+1}]}$. This completes the induction.

Now, $L = P(x_{[2^n,2^n]}, x_{[1,2^n]}) \in D_2n$. Hence

$L\ \sigma_n = P(x_{[2^n+l-1,2^{n+1}]}, x_{[1,2^{n+1}]}) \in D_2n+1$.

Hence $2^n+l-1 \equiv l-1 \pmod{2^{n+1}}$. This is false and it follows that no such σ_n exists. Hence, D_2n $(n \geq 1)$ is indeed a strictly descending chain of clauses with one binary predicate symbol and no function symbols.

One can show easily that there are only a finite number $(\leq 2^{(2^{n-1})}-1)$ of inequivalent, under $\sim$, positive clauses with n unary predicate symbols and no function symbols. It follows that there can be no strictly descending chain of clauses with unary predicate symbols but no function symbols. Hence our result is the best possible as regards arities of the predicates involved.

Incompleteness

We show next that $\{[C_i] \mid i \geq 1\}$ has no supremum and $\{[D_2j] \mid j \geq 1\}$ has no infimum. Thus our lattice is not complete wrt. sups or infs. The existence of infinite strictly ascending and

descending chains would follow from this by general lattice theory but it is better to display actual examples.

Lemma 1 1) $C_i \leq D_{2^j}$ $(i,j \geq 1)$

2) There is no E such that

$C_i \leq E \leq D_{2^j}$ for all $i, j \geq 1$.

Proof 1) Let $\gamma_{ij} = \{x_{[0,2^j]} / x_0, \ldots, x_{[i,2^j]} / x_i\}$.

Then $C_i \gamma_{ij} = \{P(x_l, x_{l+1}) \mid 0 \leq l \leq i\}\ \gamma_{ij}$

$= \{P(x_{[l,2^j]}, x_{[l+1,2^j]}) \mid 0 \leq l \leq i\}$.

$\subseteq D_{2^j}$.

2) Suppose there is such an E. Evidently E must be a positive clause, with no function symbols, in the single binary predicate symbol P. Suppose E has n variable symbols. There is a σ_n such that $C_n \sigma_n \subseteq E$. Now C_n has n+1 variable symbols. Hence there are i_1, i_2 $(i_1 < i_2)$, so that $x_{i_1} \sigma_n = x_{i_2} \sigma_n$. We choose such an i_1, i_2 so as to minimise $i_2 - i_1$.

Then $E' = \{P(x_l, x_{l+1}) \sigma_n \mid i_1 \leq l < i_2\} \subseteq E$. We show that $E' \sim D_{(i_2 - i_1)}$.

Let $\mu = \{x_{[i_1, i_2 - i_1]} / x_{i_1} \sigma_n, \ldots, x_{[i_2, i_2 - i_1]} / x_{i_2} \sigma_n\}$

Then μ is well-defined and 1-1 for $x_{l_1} \sigma_n = x_{l_2} \sigma_n$ $(l_1 \leq l_2)$ $\Leftrightarrow$ $l_1 = i_1$ and $l_2 = i_2$ or $l_1 = l_2$ (by minimality of $i_2 - i_1$)

$\Leftrightarrow l_1 \equiv l_2 \pmod{i_2 - i_1}$

$\iff x_{[l_1, i_2 - i_1]} = x_{[l_2, i_2 - i_1]}$

$\iff x_{l_1}\, \sigma_n \mu = x_{l_2}\, \sigma_n \mu$.

Next $\mathbf{E}' \mu = D_{(i_2 - i_1)}$ for $\mathbf{E}' \mu = \{P(x_l\, \sigma_n \mu, x_{l+1}\, \sigma_n \mu) \mid i_1 \leq l < i_2\}$

$= \{P(x_{[l, i_2 - i_1]}, x_{[l+1, i_2 - i_1]}) \mid i_1 \leq l < i_2\}$

$= D_{(i_2 - i_1)}$.

Hence $\mathbf{E}' \sim D_{(i_2 - i_1)}$. Now since $D_{(i_2 - i_1)} \sim \mathbf{E}' \subseteq \mathbf{E} \leq D_2 j (j \geq 1)$, there is a ν so that

$D_{(i_2 - i_1)} \nu \subseteq D_{2(i_2 - i_1)}$ We can show that this is a contradiction in exactly the same way we showed that there is no σ_n such that

$D_2 n\ \sigma_n \subseteq D_2 n+1$, thus completing the proof.

Note that since $\{C_i \mid i \geq 1\}$ is a strict chain, all the inequalities of the first part of lemma 1 are actually strict.

Theorem 1 $\{[C_i] \mid i \geq 1\}$ has no supremum and $\{[D_2 j] \mid j \geq 1\}$ has no infimum.

Proof Suppose $[\mathbf{E}] = \bigsqcup [C_i]$ exists. Then as $C_i \leq D_2 j$, by the first part of lemma 1, and since $[\mathbf{E}]$ is a supremum, $\mathbf{E} \leq D_2 j$. This contradicts the second part of lemma 1. One proves that $\bigsqcap [D_2 j]$ cannot exist in the same way.

Note that the two trivial examples of strictly increasing infinite chains have no upper bound at all. This contrasts with the strictly decreasing case where $\emptyset$ is always a lower bound. $\{C_i \mid i \geq 1\}$ is an

example of a strictly increasing infinite chain, which has no supremum but is bounded above by $\{P(x,x)\}$.

It is possible by a further extension of the above methods to produce a set of equivalence classes of clauses $\{[C_i] \mid -\infty < i < +\infty\}$ which has no supremum or infimum and is such that $C_i < C_{i+1}$. However we have done enough to show that the lattice does not possess any of the usual properties of infinite sets of elements, at least in the general case.

Chapter 4 Solvability of the general problem

We are now in a position to consider questions of the solvability of the general problem raised at the end of chapter 2, which we restate:

Given a notion of niceness $\mathcal{N}$ and f and e, it is required to find the sets of clauses, H, satisfying:

1) $H \leq H_0$ (Th)

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

3) H is minimal with respect to $\mathcal{N}$ amongst those clauses satisfying 1) and 2).

Under the assumption that every finite set of clauses has a l.g.g. relative to Th, we can locate the set of solutions when $\mathcal{N}$ is $\mathcal{N}_{cpg}$. Let $\inf_{Th} H$ be a l.g.g. of H relative to Th. Note that any tautology is a l.g.g. of $\emptyset$ relative to any Th. When Th = $\emptyset$, we require that $\inf_{\emptyset} H = \inf H$, if H does not contain a tautology, where inf is the function inf defined in section 3.3.2 in order that the notation be consistent.

There is a lattice which contains all the possible solutions. Let $\mathcal{I}_{Th}(H)$ where H is a finite set of clauses be the set of equivalence classes of the set $\{\inf_{Th} H' \mid H' \subseteq H\}$, where the equivalence relation is equivalence relative to Th. The equivalence class of $\inf_{Th} H'$ is denoted by $[\inf_{Th} H']$. $\mathcal{I}_{Th}(H)$ is ordered by:

$[\inf_{Th} H_1] \sqsubseteq [\inf_{Th} H_2]$ iff $\inf_{Th} H_1 \leq \inf_{Th} H_2$(Th).

This is evidently a good definition. $\mathcal{G}_{Th}(H)$ is actually a lattice with infs given by:

$$[\inf_{Th} H_1] \sqcap [\inf_{Th} H_2] = [\inf_{Th}(H_1 \cup H_2)]$$

This is a well-defined operation. If $H_1 \subseteq H$ and $H_2 \subseteq H$ then $[\inf_{Th}(H_1 \cup H_2)]$ is in $\mathcal{G}_{Th}(H)$. Suppose $\inf_{Th} H_1 \sim \inf_{Th} H_1'$ (Th) and $\inf_{Th} H_2 \sim \inf_{Th} H_2'$ (Th). Then,

$$\begin{aligned}\inf_{Th}(H_1 \cup H_2) &\sim \inf_{Th}\{\inf_{Th} H_1, \inf_{Th} H_2\} \text{ (Th)} \\ &\sim \inf_{Th}\{\inf_{Th} H_1', \inf_{Th} H_2'\} \text{ (Th)} \\ &\sim \inf_{Th}(H_1' \cup H_2') \text{ (Th)}.\end{aligned}$$

Therefore $\sqcap$ is well-defined.

As $\inf_{Th}(H_1 \cup H_2) \leq \inf_{Th} H_i$ (Th) for i=1,2, $\sqcap$ does give a lower bound. Suppose $\inf_{Th} H_3 \leq \inf_{Th} H_i$ (Th) for i=1,2. Then

$$\begin{aligned}\inf_{Th} H_3 &\leq \inf_{Th}\{\inf_{Th} H_1, \inf_{Th} H_2\} \text{ (Th)} \\ &\sim \inf_{Th}(H_1 \cup H_2) \text{ (Th)}.\end{aligned}$$

Therefore $[\inf_{Th} H_3] \sqsubseteq [\inf_{Th} H_1] \sqcap [\inf_{Th} H_2]$, and $\sqcap$ gives the greatest lower bound. As $\mathcal{G}_{Th}(H)$ is finite, arbitrary subsets have infs, given by:

$$\bigsqcap\{[\inf_{Th} H_i] \mid i=1,n\} = [\inf_{Th} H_1] \sqcap ([\inf_{Th} H_2] \ldots ([\inf_{Th} H_{n-1}] \sqcap [\inf_{Th} H_{n-1}])) \ldots),$$

$$\bigsqcap\{[\inf_{Th} H_1]\} = [\inf_{Th} H_1],$$
$$\bigsqcap \emptyset = [\{\{P(x), \overline{P(x)}\}\}].$$

Consequently there is a sup operation defined by:

$$[\inf_{Th} H_1] \sqcup [\inf_{Th} H_2] = \bigsqcap\{[\inf_{Th} H_3] \in \mathcal{G}_{Th}(H) \mid [\inf_{Th} H_i] \sqsubseteq [\inf_{Th} H_3], \text{ for } i=1,2\}.$$

Thus, equipped with $\sqsubseteq$, $\sqcap$ and $\sqcup$, $\mathcal{G}_{Th}(H)$ is a finite lattice.

Example Suppose $H = \{C_1, C_2, C_3\}$ where $C_1 = \{P(f()), Q(f())\}$,
$C_2 = \{P(g()), Q(g())\}$,
$C_3 = \{P(h())\}$.

Then $\inf\{C_1, C_2\} \sim \{P(x), Q(x)\}$,
$\inf\{C_1, C_3\} \sim \inf\{C_2, C_3\} \sim \inf\{C_1, C_2, C_3\} = \{P(x)\}$.

The lattice $\mathcal{G}_{\emptyset}(H)$ is displayed in figure 1

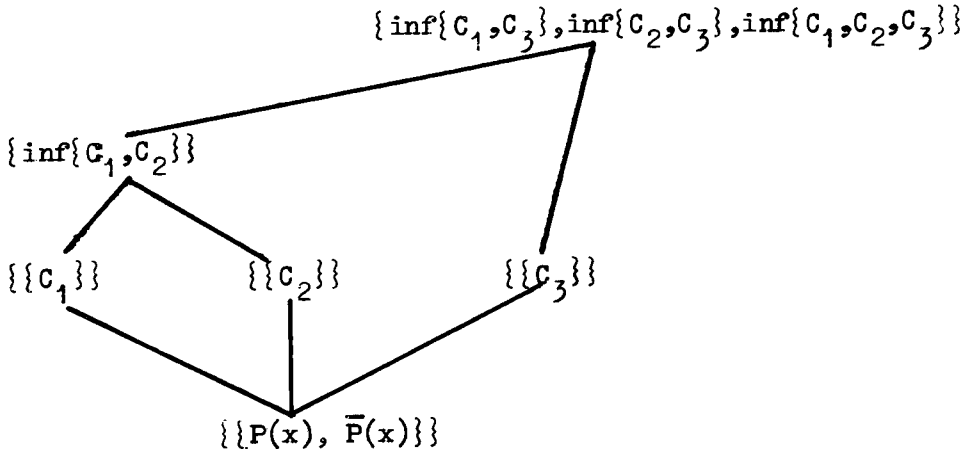


Figure 1

We can now show that the solutions are located in $\mathcal{I}_{Th}(H_0)$ when $\mathcal{S}$ is $\mathcal{S}_{cpg}$.

The set of clauses in H_0 that an arbitrary clause C explains is defined to be:

$$\text{Explainset}(C) = \{C' \in H_0 \mid C \leq C' \ (Th)\}.$$

Notice that Power(C) = cardinality of Explainset(C).

We also set Complexity(H) = $||H||$, the cardinality of H.

Relative equivalence between sets of clauses is defined by:
$H \sim H' \ (Th)$ iff $H \leq H' \ (Th)$ and $H' \leq H \ (Th)$.

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

If $H \sim H' \ (Th)$ and both H and H' are reduced then there is a unique bijection θ: $H' \rightarrow H$ such that $\theta(C') \sim C'$ for every C' in H'. This may be proved in a way analogous to the corresponding part of the proof of the corresponding part of the statement of theorem 3.3.1.2. One can easily extend the analogy to the rest of the theorem if relative generalisation is a recursive relation.

Theorem 1 H is a solution when $\mathcal{S}$ is $\mathcal{S}_{cpg}$ iff it is reduced relative to Th and is equivalent, relative to Th, to an H' satisfying:

1) $H' \subseteq \{\inf_{Th} H_1 \mid H_1 \subseteq H_0\}$.

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

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

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

Further, any H' satisfying these conditions is a solution.

Proof Let H be a solution. If it is not reduced relative to Th then let H" be reduced and equivalent, relative to Th, to H. Then Complexity(H") < Complexity(H), $H'' \leq H_0$ (Th) and $\forall H'' \wedge Th \wedge Irr \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)$ is consistent, since $\vdash \forall H \equiv \forall H''$. This contradicts the fact that H is a solution. So H is reduced relative to Th.

Define $H' = \{\inf_{Th}\text{Explainset}(C) \mid C \in H\}$. We see that $H' \subseteq \{\inf_{Th} H_1 \mid H_1 \subseteq H\}$, Complexity(H') $\leq$ Complexity(H), Power(H) = Power(H'), and $H \leq H'$ (Th), and $H' \leq H_0$ (Th). It follows from $H \leq H'$ (Th) that $\forall H' \wedge Th \wedge Irr \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)$ is consistent. So since H is a solution, Complexity(H') = Complexity(H) and $H \sim H'$ (Th). Suppose H" satisfies 1), 2) and 3) and $H'' \preceq_{cp} H'$. Then as $H' \preceq_{cp} H$, $H'' \preceq_{cp} H$. So, as H is a solution and $H' \preceq_{cp} H$, $H' \preceq_{cp} H \preceq_{cp} H''$. This establishes condition 4) for H' and concludes the first part of the proof.

We demonstrate the last part of the theorem next. Suppose that H' satisfies conditions 1) to 4). Then it satisfies conditions 1) and 2) for being a solution. Suppose that H" also satisfies conditions 1) and 2) for being a solution and that $H'' \preceq_{cpg} H'$. We will show that $H' \preceq_{cpg} H''$. Let $H''' = \{\inf_{Th}\text{Explainset}(C) \mid C \in H''\}$. We see that

$H'' \leq H'''$ (Th), $H''' \not\sim_{cpg} H''$ and H''' satisfy condition 1). Therefore H''' satisfies conditions 1), 2) and 3). Further $H''' \not\sim_{cp} H''$. As $H'' \not\sim_{cpg} H'$, $H'' \not\sim_{cp} H'$ and so $H''' \not\sim_{cp} H'$. As H' satisfies condition 4) it follows that $H'' \not\sim_{cp} H'''$. But $H''' \not\sim_{cpg} H'' \not\sim_{cpg} H'$. Therefore $H' \leq H'''$.

There is therefore a map $\theta : H''' \to H'$ such that $\theta(C''') \leq C'''$ (Th), for any C''' in H'''. If the map is not onto, there is a C' in H' such that $\theta(C''') \neq C'$ for any C''' in H'''. Then $H' \setminus \{C'\}$ satisfies conditions 1), 2) and 3) which contradicts the fact that H' satisfies condition 4). Therefore θ is onto. Since Complexity(H''') = Complexity(H''), as $H''' \not\sim_{cp} H'$ and $H' \not\sim_{cp} H'''$, θ must be a bijection. Therefore as Power (C''') $\leq$ Power($\theta(C''')$) for any C''' in H''' and Power(H') = Power(H''') then Power(C''') = Power($\theta(C''')$) for any C''' in H'''. But as $\theta(C''') \leq C''$ (Th), Explainset($\theta(C''')$) $\supseteq$ Explainset(C''') for any C''' in H'''. Therefore Explainset($\theta(C''')$) = Explainset(C''') and so $C''' \leq \inf_{Th}$Explainset($\theta(C''')$) $\leq \theta(C''')$ (Th), as H' satisfies condition 1), for any C''' in H'''. Now, as θ is a bijection, $H''' \sim H'$ (Th) and taking this with $H''' \not\sim_{cpg} H''$ and $H' \not\sim_{cp} H'''$ we see that $H' \not\sim_{cpg} H''$ which concludes the proof that H' is a solution.

Suppose next that H' satisfies conditions 1) to 4) and that H is reduced, relative to Th, and that $H \sim H'$ (Th). Then H' is a solution and so is reduced by the above. Therefore by the remark after the definition of when a set of clauses is relatively reduced, there is a bijection $\theta : H' \to H$ such that $\theta(C') \sim C'$ for every C' in H'.

Therefore $H \not\sim_{cpg} H'$ and since H satisfies conditions 1) and 2) for being a solution as $H \sim H'$ (Th) it follows that H is a solution, concluding the proof.

Let us look at some examples. Here is another appearance of the crows.

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

Table 1

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

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

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

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

Evidently $\forall C_3 \wedge e_1 \wedge e_2 \wedge f_1 \wedge f_2$ 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 as given in table 2. Again, Th is empty.

Now, $H_0 = \{C_i \mid 1 \leq i \leq 7\}$. The members of $\{\inf H' \mid H' \subseteq H_0, H' \neq \emptyset\}$ which are consistent with $\bigwedge_{i=1}^{7}(e_i \wedge f_i)$ are, apart from C_1 to C_7,

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

and

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

The solution is $H = \{C_8, C_9, C_4, C_5\}$ and includes the following version of the generalisation 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 generalisation is equivalent to the original one.

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 = $\overline{\text{Dangerous}}$(animal4)	e_4 = Size(animal4,small)∧Colour(animal4,black)∧ Animal(animal4,cat)
f_5 = $\overline{\text{Dangerous}}$(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).

Table 2

Note that in fact the solutions have been located in $\{\inf_{Th} H_1 \mid H_1 \subseteq H_0\}$ rather than in $\mathcal{J}_{Th}(H_0)$. One can reword theorem 1 to find the location in $\mathcal{J}_{Th}(H_0)$. $\mathcal{J}_{Th}(H_0)$ was introduced in order to find a well-known mathematical structure in which the solutions could be located; it is expected to make any associated computational problems a little easier. In future, we shall generally confuse $\mathcal{J}_{Th}(H_0)$ with $\{\inf_{Th} H_1 \mid H_1 \subseteq H_0\}$.

Corollary 1 For the case where $\mathfrak{I}$ is $\mathfrak{I}_{cpg}$, every problem has a solution.

Proof We need only show that there is an H' satisfying conditions 1), 2) and 3) of theorem 1, for as $\{\inf_{Th} H_1 \mid H_1 \subseteq H_0\}$ is finite, there will then be a suitable minimal one, with respect to $\mathfrak{I}_{cp}$ and this will be a solution by theorem 1.

Now $H_0 \subseteq \{\inf_{Th} H_1 \mid H_1 \subseteq H_0\}$ and certainly $H_0 \leq H_0$ (Th). It is part of the problem conditions that Th $\wedge$ Irr $\bigwedge_{i=1}^{n}(e_i \wedge f_i)$ is consistent. Since $\vdash \bigwedge_{i=1}^{n}(e_i \wedge f_i) \rightarrow H_0$, it follows that H_0 satisfies condition 3), which concludes the proof.

Now that we know there is always a solution we can try to construct an algorithm for finding one. Let us consider the simplest case where Th and Irr are empty. Now $\inf_{Th}$ is calculable and so, therefore, is $\{\inf_{Th} H_1 \mid H_1 \subseteq H_0\} = \{\inf H_1 \mid H_1 \subseteq H_0\}$. For each subset, H', of this set we can check whether $H' \leq H_0$ (Th), and if we could only check the consistency condition, we could then isolate the solutions, using theorem 1, since $\mathcal{S}_{cp}$ is certainly decidable. Now, in general, the consistency of an arbitrary set of clauses is an undecidable property. We might hope, though, to avoid the difficulty since H' has arisen by means of the special process of generalisation. Unfortunately it can be shown that undecidability persists.

Theorem 2 Suppose that $\mathcal{S}$ is $\mathcal{S}_{cpg}$. There is no algorithm which will, given any f and Ev produce a solution to the resulting generalisation problem. (Here both Th and Irr are empty.)

Proof We will show that if such an algorithm exists then it is possible to tell whether or not the universal closure of an arbitrary set of clauses is consistent. As this is not possible we shall then have demonstrated the non-existence of the algorithm.

To this end let H be an arbitrary set of clauses. We may assume, without loss of generality that no a_{ij} used in the representation theorem, theorem 3.3.1.4, appears in H, that H contains no tautologies and that all clauses in H are standardised apart. For if H does not have these properties we can effectively find another set of clauses which does and whose universal closure is equiconsistent with that of H.

With H we will associate an f and an Ev in such a way that H is consistent iff any solution to the resulting generalisation problem has a certain decidable property. This will complete the proof.

Let $H = \{D_1,\ldots,D_n\}$ and suppose that the predicate symbols appearing in H are $P_j(j=1,p)$. If P_j has degree m, associate with it a new predicate symbol Q_j of degree m+n+2 for j=1,p. (Notice that $n=||H||$.)

It is convenient to temporarily introduce some new syntactic notation. If $t_1,\ldots,t_{m+n+2}$ are terms and $L = (\pm)\ P_j(t_1,\ldots,t_m)$, for some j, we set

$$L[[t_{m+1},\ldots,t_{m+n+2}]] = (\pm)\ Q_j(t_1,\ldots,t_{m+n+2}).$$

If L does not begin with some P_j, we set

$$L[[t_{m+1},\ldots,t_{m+n+2}]] = L.$$

In the above $(\pm)\ P_j$ is either P_j or $\bar{P}_j$ and $(\pm)$ is used according to the usual conventions.

Similarly we set $C[[t_{m+1},\ldots,t_{m+n+2}]] = \{L[[t_{m+1},\ldots,t_{m+n+2}]] \mid L \in C\}$ for any clause C and $H'[[t_{m+1},\ldots,t_{m+n+2}]] = \{C[[t_{m+1},\ldots,t_{m+n+2}]] \mid C \in H'\}$, for any set, H', of clauses.

Notice that if H' does not contain any of the Q_j then $\forall_{H'}$ is equiconsistent with $\forall\ H'[[t_{m+1},\ldots,t_{m+n+2}]]$.

Let x_1,x_2,y_1,y_2,u_1,u_2 be variables not appearing in H and let P be

a new n+2-ary predicate symbol.

We define auxiliary terms t_{ik} $(i=1,n;\ k=1,n)$ by $t_{ii} = f(u_2)$ and $t_{ik} = u_1$ if $k \neq i$ and set

$$H^+ = \{D_i[[x_1,f(y_1),t_{i1},\ldots,t_{in}]] \cup \{P(x_1,f(y_1),t_{i1},\ldots,t_{in})\} \mid D_i \in H\}.$$

We also set

$$H^- = \{D_i[[f(x_2),y_2,t_{i1},\ldots,t_{in}]] \cup \{\bar{P}(f(x_2),y_2,t_{i1},\ldots,t_{in})\} \mid D_i \in H\}.$$

Now $\forall(H^+ \cup H^-)$ is equiconsistent with $\forall H$. For suppose $\forall(H^+ \cup H^-)$ is consistent. Notice that for any i, $D_i[[f(x_2),f(y_1),t_{i1},\ldots,t_{in}]]$ is in $\mathcal{R}(H^+ \cup H^-)$. Therefore $\mathcal{R}(H^+ \cup H^-) \leq H[[f(x),\ldots,f(x)]]$. So $\forall H[[f(x),\ldots,f(x)]]$ is consistent and it follows, by a remark above, that $\forall H$ is.

Suppose $\forall H$ is consistent. Then $\forall H[[x_1,f(y_1),u_1,\ldots,u_n]]$ is also consistent, by the remark made above. As $H[[x_1,f(y_1),u_1,\ldots,u_n]]$ generalises H^+ and does not contain any occurrence of the predicate symbol P, any extension of a model of $H[[x_1,f(y_1),u_1,\ldots,u_n]]$ to an interpretation of H^+ will be a model of H^+. If we choose that extension which assigns to P the empty predicate, of the appropriate degree, then the extension will also be a model of H^-. Thus if $\forall H$ is consistent, so is $\forall(H^+ \cup H^-)$.

We have proved, therefore, that $\forall H$ and $\forall(H^+ \cup H^-)$ are

equiconsistent.

Set $Gen_1 = \{\inf(D,D')\ \tau_{D,D'} \mid D \neq D'$ and either D and D' are both in H^+ or else they are both in $H^-\}$.

The translations, $\tau_{D,D'}$ must be taken so that all the variables in Gen_1 are new and the clauses in Gen_1 are standardised apart. Notice that every clause in Gen_1 has a single occurrence of the predicate letter P.

Set $Gen_2 = \{D^* \mid D \in Gen_1$ and D^* is D except that the sign of P is changed$\}$.

Recall the γ^i_j used in theorem 3.3.3.1.4 (the representation theorem).

Let $\gamma^i = \gamma^i_{n'}$, where n' is the first integer such that $(H^+ \cup H^- \cup Gen_1 \cup Gen_2)\ \gamma^i_{n'}$ contains no variable symbols ($i \geq 1$). Notice that any literal has at most one occurrence in this set of clauses.

Define, $H_0 = (H^+ \cup H^-)\ \gamma^1 \cup (H^+ \cup H^-)\ \gamma^2 \cup Gen_1\ \gamma^3 \cup Gen_2\ \gamma^4$.

We may now define an f and an Ev with the properties promised at the beginning of the proof.

Let f be the set of literals with predicate letter P appearing in H_0.

If $L \in f$, there is, by a remark above on the properties of the γ^i,

a unique clause, D, in H_0 in which L occurs.

Set $Ev(L) = \bar{M}_1 \wedge \ldots\ldots \wedge \bar{M}_{k(L)}$ (where $\{M_k \mid k=1, k(L)\} = D \setminus \{L\}$).

Notice that $H_0 = \{\overline{Ev(L)} \cup \{L\} \mid L \in f\}$ and so H_0 has its usual meaning. We must now show that we have a genuine problem, that is that $\bigwedge_{j=1}^{l}(e_j \wedge f_j)$ is consistent. (l depends on $n = ||H||$, and in fact $l = 4n^2$), and H_0 contains no tautologies.

Suppose that $\bar{L}_1$ and $\bar{L}_2$ are contradictory literals in $\bigwedge_{j=1}^{l}(e_j \wedge f_j)$. From the properties of the γ^i, and the fact that all clauses in $H^+ \cup H^- \cup Gen_1$ are standardised apart, L_1 and L_2 must occur in a single clause D in H_0, and so they cannot begin with the predicate letter P.

Since H contains no tautologies, neither does $H^+ \cup H^-$ and so D is in $Gen_1\ \gamma^3 \cup Gen_2\ \gamma^4$. Since neither literal has predicate symbol P, we may assume, by the construction of Gen_2, that D is in $Gen_1\ \gamma^3$. But if D' is in Gen_1 then D' generalises some non-tautologous clause in $H^+ \cup H^-$ and so must itself be non-tautologous. As γ^3 substitutes distinct constants for distinct variables D must also be non-tautologous, which contradicts the assumption that L_1 and L_2 are contradictory.

Therefore, $\bigwedge_{j=1}^{l}(e_j \wedge f_j)$ is consistent. Since P does not occur in H, and $\bigwedge_{j=1}^{n} e_j$ is consistent, H_0 contains no tautologies.

Let $H_{test} = H^+ \cup H^- \cup Gen_1\ \gamma^3 \cup Gen_2\ \gamma^4$. We are going to demonstrate that if H_{soln} is a solution to the problem defined by Ev and

f then H is consistent iff $H_{soln} \sim H_{test}$. As equivalence is decidable we will then be able to tell, effectively whether H is consistent, if an algorithm for producing a solution is available. This will conclude the proof, as we remarked at the beginning. To do this we need two lemmas.

Lemma 2.1 $\forall H$ is equiconsistent with $\forall H_{test} \wedge \bigwedge_{j=1}^{1}(e_j \wedge f_j)$.

Proof Suppose $\forall H$ is consistent. As $\vdash \bigwedge_{j=1}^{1}(e_j \wedge f_j) \rightarrow H_0$ and $Gen_2\ \gamma^3 \cup Gen_2\ \gamma^4 \subseteq H_0$, $\forall H_{test} \wedge \bigwedge_{j=1}^{1}(e_j \wedge f_j)$ is equiconsistent with $\forall(H^+ \cup H^-) \wedge \bigwedge_{j=1}^{1}(e_j \wedge f_j)$. Suppose then that $\forall(H^+ \cup H^-) \wedge \bigwedge_{j=1}^{1}(e_j \wedge f_j)$ is inconsistent. Then $\vdash \forall(H^+ \cup H^-) \rightarrow \neg \bigwedge_{j=1}^{1}(e_j \wedge f_j)$.

Now we have shown above that $\bigwedge_{j=1}^{1}(e_j \wedge f_j)$ is consistent. Therefore $C_0 = \{\bar{f}_j \mid j=1,1\} \cup \bigcup_{j=1}^{1}\bar{e}_j$ is not a tautology. It follows from the subsumption theorem (Lee, 1967 and Kowalski, 1970) that there is a clause C in $\mathcal{R}^{n'}(H^+ \cup H^-)$, for some n', which subsumes C_0. If C has been obtained by resolution from more than one member of $H^+ \cup H^-$, it follows from the construction of $H^+ \cup H^-$ that every literal in C will contain at least three occurrences of the unary function symbol f. Now it is impossible that C be $\emptyset$, for then $\forall(H^+ \cup H^-)$ would be inconsistent and this contradicts the fact that $\forall(H^+ \cup H^-)$ is equiconsistent with $\forall H$ taken together with the assumption that $\forall H$ is consistent. Further no literal in C_0 can contain more than two occurrences of the unary function symbol f. Therefore if C has been obtained by resolution from more than one member of $H^+ \cup H^-$, it is impossible for C to generalise C_0. This contradiction establishes the existence of a D in $H^+ \cup H^-$ such that C

is in $\mathcal{R}^{n'}(\{D\})$.

Let us assume that D is in H^+. Then C will contain a positive literal L with predicate letter P containing two occurrences of the function letter f, one of which is in the second argument place of L. But a positive literal occurring in C_0 whose predicate letter is P must be the negation of a literal occurring in a clause in $H^- \gamma^1 \cup H^- \gamma^2 \cup \text{Gen}_1 \gamma^3 \cup \text{Gen}_2 \gamma^4$. Any such literal occurring in $H^- \gamma^1 \cup H^- \gamma^2 \cup \text{Gen}_1 \gamma^3$ has no occurrence of the function symbol f in its second argument place. By the construction of Gen_2, any such literal occurring in a clause in $\text{Gen}_2 \gamma^4$ will have exactly one occurrence of the function symbol f. Therefore L generalises no literal in C_0 and so $C \not\leq C_0$. This is a contradiction. If D is in H^- the contradiction is established similarly. So we have established that $\forall (H^+ \cup H^-) \wedge \bigwedge_{j=1}^{l}(e_j \wedge f_j)$ and so $\forall H_{test} \wedge \bigwedge_{j=1}^{l}(e_j \wedge f_j)$ is consistent.

Suppose $\forall H_{test} \wedge \bigwedge_{j=1}^{l}(e_j \wedge f_j)$ is consistent. We see, successively, that so are $\forall H_{test}$, $\forall (H^+ \cup H^-)$ and $\forall H$. This concludes the proof of the lemma.

Lemma 2.2 Suppose H_{soln} is a solution to the problem determined by Ev and f, and that $H_{soln} \subseteq \mathcal{G}(H_0)$. Then, $\text{Gen}_1 \gamma^3 \cup \text{Gen}_2 \gamma^4 \subseteq H_{soln} \subseteq H_0 \cup H_{test}$. (We may assume that $H_{test} \cup H_0 \subseteq \mathcal{G}(H_0)$.)

Proof For this lemma we need the easily proved fact that if $C \leq D \gamma^i_j$ for some i, j and C contains none of the constants a_{ij}, for

any $i \geq 1$ and $j \geq 1$, then $C \leq D$. This fact is implicit in the proof of the representation theorem, theorem 3.3.3.1.4.

Suppose **E** is in $Gen_1\ \gamma^3$ but not in H_{soln}. Now as H_{soln} is a solution $H_{soln} \leq H_0 \supseteq \{\mathbf{E}\}$. So there is a clause, C, in H_{soln} which generalises **E** and, since $H_{soln} \subseteq \mathcal{G}(H_0)$, C must generalise some other member of H_0 and so C does not contain any of the constants a_{ij} for any $i \geq 1$ and $j \geq 1$.

For some D, D' in H^+, $\mathbf{E} = \inf\{D,D'\}\ \tau_{D,D'}\ \gamma^3$. Therefore $C \leq \inf\{D,D'\}\ \tau_{D,D'}$, using the fact given at the beginning of the proof. We can find an f_j in f, such that $\mathbf{E}^* = \bar{e}_j \cup \{f_j\}$ and so as $\inf\{D,D'\}\ \tau_{D,D'}$ is inconsistent with $e_j \wedge f_j$, so is C and so therefore is H_{soln}. This contradiction establishes the fact that $Gen_1\ \gamma^3 \subseteq H_{soln}$.

Similarly we can show that $Gen_2\ \gamma^4 \subseteq H_{soln}$.

Any clause in $\mathcal{G}(H_0)$ which is not in $H_{test} \cup H_0$ must generalise a member of $Gen_1 \cup Gen_2$, by applications of one or both of the representation theorem and the remark made at the beginning of the proof. This clause is inconsistent with the evidence, as was shown above, and so cannot be a member of H_{soln}. This concludes the proof.

We may now finish the proof of theorem 2. We have to show that if H_{soln} is a solution to the problem defined by Ev and f then H is consistent iff $H_{soln} \sim H_{test}$. By theorem 1, we may assume that $H_{soln} \subseteq \mathcal{G}(H_0)$. We may also assume that $H_{test} \cup H_0 \subseteq \mathcal{G}(H_0)$.

Suppose that H is consistent, that H_{soln} is a solution and that $H_{soln} \nsim H_{test}$. Then by lemma 2.2, $H_{soln} \subseteq H_{test} \cup H_0$. Therefore $H_{test} \leq H_{test} \cup H_0 \leq H_{soln}$. So $H_{soln} \not\leq H_{test}$ and there is a C_{test} in H_{test} such that no clause in H_{soln} generalises C_{test}. Since $Gen_1\ \gamma^3 \cup Gen_2\ \gamma^4 \subseteq H_{soln}$ by lemma 2.2, C_{test} is in $H^+ \cup H^-$. Now there are clauses C_1, C_2 in H_{soln} such that $C_1 \leq C_{test}\ \gamma^1$ and $C_2 \leq C_{test}\ \gamma^2$ since $H_{soln} \leq H_0$.

C_1 cannot generalise any clause in H_0 other than $C_{test}\ \gamma^1$, for then it generalises C_{test}, by the remark made at the beginning of the proof of lemma 2.2. Therefore $C_1 = C_{test}\ \gamma^1$ and similarly $C_2 = C_{test}\ \gamma^2 \neq C_1$. Let $H'_{soln} = (H_{soln} \cup \{C_{test}\}) \setminus \{C_1, C_2\}$.

Then as $H_{test} \leq H'_{soln}$, $\forall_{H'_{soln}} \wedge \bigwedge_{j=1}^{1}(e_j \wedge f_j)$ is consistent by lemma 2.1. Further $H'_{soln} \leq H_{soln} \leq H_0$, $H'_{soln} \sim_c H_{soln}$ but $H_{soln} \not\sim_c H'_{soln}$. This contradicts the fact that H_{soln} is a solution.

We have shown that if $\forall_H$ is consistent, $H_{soln} \sim H_{test}$.

Suppose $\forall_H$ is inconsistent and $H_{soln} \sim H_{test}$. Then $\forall_{H_{soln}} \wedge \bigwedge_{j=1}^{1}(e_j \wedge f_j)$ is equiconsistent with $\forall_{H_{test}} \wedge \bigwedge_{j=1}^{1}(e_j \wedge f_j)$ which is inconsistent by lemma 2.1 since $\forall_H$ is. This certainly contradicts the fact that H_{soln} is a solution, which concludes the proof.

The same result can be obtained for a variety of other $\sim$'s if we change the construction of f and Ev slightly.

The construction proceeds as above until Gen_1 and Gen_2 have been

defined. Then one sets:

$$H_0^m = \bigcup_{k=1}^{m}(H^+ \cup H^-)\ \gamma^k \cup \mathrm{Gen}_1\ \gamma^{m+1} \cup \mathrm{Gen}_2\ \gamma^{m+2}\ (m \geq 2).$$

The construction given above corresponds to the case where m=2. One may then define f^m and $\mathbf{E}v^m$ analogously to the above construction and verify that $\bigwedge_{j=1}^{l_m}(\mathbf{E}_j^m \wedge f_j^m)$ is consistent. (Here, $l_m = 4n^2 + 2n(m-2)$.)

One can then define H_{test}^m and prove:

1 $\forall_H$ is equiconsistent with $\forall_{H_{test}^m} \wedge \bigwedge_{j=1}^{l_m}(e_j^m \wedge f_j^m)$

2 If $\forall_H$ is inconsistent, $\forall_{H_{soln}} \wedge \bigwedge_{j=1}^{l_m}(\mathbf{E}_j^m \wedge f_j^m)$ is consistent and $H_{soln} \leq H_0^m$ then $||H_{soln}|| \geq m$.

The proof of 1 is analogous to that of lemma 2.1. To see that $||H_{soln}|| \geq m$ under the assumptions of 2 suppose otherwise and choose a clause C in $H^+ \cup H^-$. There are clauses $C_1,\ldots,C_m$ in H_{soln} such that $C_k \leq C\ \gamma^k$ (k=1,m) as $H_{soln} \leq H_0^m$. Since $||H_{soln}|| < m$, two of these are the same and so there is a C_k in H_{soln} such that $C_k \leq C$, by the representation theorem. Hence $H_{soln} \leq H^+ \cup H^-$, and so $H_{soln} \leq H^+ \cup H^- \cup H_0^m \leq H_{test}^m$. But $\forall_{H_{test}^m} \wedge \bigwedge_{j=1}^{l_m}(e_j^m \wedge f_j^m)$ is inconsistent as $\forall_H$ is, using 1. Therefore $\forall_{H_{soln}^m} \wedge \bigwedge_{j=1}^{l_m}(e_j^m \wedge f_j^m)$ is also inconsistent which contradicts one of the assumptions of 2, which therefore, shows that $||H_{soln}|| \geq m$.

Recall the definitions of $\mathfrak{s}_1$, $\mathfrak{s}_{1'}$, $\mathfrak{s}_s$, and $\mathfrak{s}_{s'}$ given in chapter 2.

Corollary 2 Let $\mathcal{J}$ be one of $\mathcal{J}_c$, $\mathcal{J}_{cp}$, $\mathcal{J}_1$, $\mathcal{J}_{1'}$, $\mathcal{J}_s$ or $\mathcal{J}_{s'}$. There is no algorithm which will, given any f and Ev produce a solution to the resulting generalisation problem, although such a solution exists.

Proof Suppose H is a set of clauses as described at the beginning of the proof of theorem 2. We need merely show an effective way, given an algorithm which always produces a solution, of deciding whether $\forall$H is consistent.

Suppose $\mathcal{J}$ is $\mathcal{J}_c$. There is always a solution, given m, to the problem defined by Ev^m and f^m, since the complexities of any solution must lie between zero and $||H_0^m||$ as H_0^m satisfies conditions 1 and 2 for being a solution. Choose $m > ||H_{test}^m||$. Find a solution, H_{soln}, to the problem defined by Ev^m and f^m. If $\forall$H is inconsistent then $||H_{soln}|| \geq m$ by 2 above. If $\forall$H is consistent, so is $\forall H_{test}^m \wedge \bigwedge_{j=1}^{1^m} (e_j^m \wedge f_j^m)$ and as $H_{test}^m \leq H_0^m$, $||H_{soln}|| \leq ||H_{test}^m|| < m$.

Therefore $\forall$H is consistent iff $||H_{soln}|| < m$, which concludes the proof for $\mathcal{J} = \mathcal{J}_c$.

The proofs for $\mathcal{J}_1$, $\mathcal{J}_{1'}$, $\mathcal{J}_s$ and $\mathcal{J}_{s'}$ are all similar to the above. In each case if $\forall$H is consistent, H_{test}^m provides an upper bound for the measure being used. If $\forall$H is inconsistent one obtains a lower bound, which increases to infinity, with m, on the measure of a solution. As one sees that there must be a solution in each case, one can tell in each case whether $\forall$H is consistent.

When $\mathcal{J}$ is $\mathcal{J}_{cp}$ we see that there is always a solution to the

problem defined by $\mathbf{Ev}^m$ and f^m. For H_0^m satisfies conditions 1 and 2 for being a solution. Hence the complexity of any solution must be less than or equal to $||H_0^m||$ and so the power must be less than or equal to $||H_0^m||^2$. These bounds show that a solution exists. If there is an algorithm for producing a solution then the same algorithm will produce a solution when $\mathcal{S}$ is $\mathcal{S}_c$, for $\mathcal{S}_{cp}$ minimal implies that $\mathcal{S}_c$ is minimal. Together with the result for $\mathcal{S}_c$, this implies the result for $\mathcal{S}_{cp}$. This concludes the proof.

This technique will establish the result for any reasonable way of combining integer-valued complexity measures. When the result has been established for some $\mathcal{S}_1$ then it follows at once for any lexicographic refinement of $\mathcal{S}_1$ which always allows solutions. (Compare the proof of the last part of the corollary).

This unsolvability result seems rather paradoxical, since we seem to have little, if any, trouble with consistency, when forming generalisations either in ordinary or scientific life. Furthermore, there can no longer be any hope of a general method. Everything may be reconciled, however, by postulating that our set of generalisation problems is too inclusive; we have not succeeded in restricting ourselves to "ordinary" generalisation problems. We will not try to formulate restrictions which give exactly the ordinary problems and no others. To show that some proposed solution captures the generalisation problems faced by any human being could no doubt involve us in uninviting problems of psychology. Similarly, to capture ordinary scientific

classificatory generalisation problems may involve comparisons with, say, 19th century zoological practise, another uninviting task.

What we will do is indicate the kind of restrictions that may be formulated, with a view to giving some that are potentially useful for work in A.I.

One could make restrictions on the vocabulary used. Restrictions on the possible predicate symbols and function symbols, could rule out the type of unsolvability proof used above. One extreme possibility is to require that the only function symbols are constant ones. Then every problem where $\mathcal{P}$ is $\mathcal{P}_{cpg}$ and Th and Irr are arbitrary sets of clauses (with no function symbols, other than constant ones) is solvable, by the discussion after theorem 1. We would like to single out one particularly simple subcase. Suppose Th is empty and every member of the (finite) Herbrand base of $Th \wedge Irr \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)$ or its negation is implied by $Th \wedge Irr \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)$. In other words, $Th \wedge Irr \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)$ has exactly one Herbrand model. Let $C = \{\bar{L} \mid Th \wedge Irr \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)$ implies L and **L** is in the Herbrand base of $Th \wedge Irr \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)\}$. Then, if $H \subseteq \{\inf H' \mid H' \subseteq H_0, H \neq \emptyset\}$, $\forall H \wedge Th \wedge Irr \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)$ is consistent iff $H \not\subseteq \{C\}$. In other words, we test H against the unique Herbrand model.

Another kind of restriction is to fix on a specific Th. We might require that Th always contains a "basic" scientific theory, which includes some universal assumptions (for example meaning postulates or principles of causality). One extreme is to demand that Th $\wedge$ Irr is

a decidable theory, or, at any rate, that one can decide the sentences produced by the process of taking relative infima. In this case it follows from the discussion of theorem 1 that an arbitrary problem is decidable if $\mathcal{S}$ is equal to $\mathcal{S}_{cpg}$ and the vocabulary of $\bigwedge_{i=1}^{n}(e_i \wedge f_i)$ is a subset of that of Th $\wedge$ Irr.

Finally one might place restrictions on the kind of data generalised. Most psychological and taxonomic generalisation problems (Hunt, Marin and Stone, 1966) form classes from attributes, which are maps from objects to finite sets, or to real numbers.

Suppose, then, that some solvable case has been found, when $\mathcal{S}$ is $\mathcal{S}_{cpg}$. We give some properties of solutions, independent of which subcase is being considered, but which can be of help when looking for solutions.

First we show that solutions are irredundant. H is said to be <u>irredundant</u> iff:

1) If C is in H then Explainset(C)
$$\notin \bigcup_{D \in H} \text{Explainset}(D).$$

2) If C is in H and D is in H_0 then either $C \leq D$ (Th) or $\text{Th} \wedge \text{Irr} \wedge \forall_H \wedge \forall_{\inf_{Th}\{C,D\}} \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)$ is inconsistent.

Intuitively, H is irredundant if it cannot be improved (in the $\mathcal{S}_{cpg}$ ordering) by removing clauses or generalising clauses, without

violating one of the first two conditions for it to be a solution.

Theorem 3 Any solution is irredundant.

Proof Suppose condition 1 for H to be irredundant fails. Then for some C in H, $H' = H \setminus \{C\} < H_0$ (Th) iff $H \leq H_0$ (Th). Then Complexity(H) > Complexity(H').

Suppose H fails condition 2. Let $H' = (H \setminus \{C\}) \cup \{\inf\{C,D\}\}$ where D is that member of H_0 determined by the failure. Then Complexity(H') = Complexity(H) but Power(H') > Power(H).

In both cases H' satisfies the first two conditions for being a solution iff H does. So H cannot be a solution.

It would, evidently, be helpful if one could divide the solution of the formal problem into the solution of several smaller problems, or at any rate to restrict the search to a smaller part of $\mathcal{G}_{Th}(H_0)$ the set of generalisations of H_0. Our next few lemmas are a step in this direction.

Under some hypotheses, the search space, $\mathcal{G}_{Th}(H_0)$ can be divided into several positive and negative parts.

Let $f^{+P} = \{f_i \in f \mid f_i$ is positive and has predicate symbol P$\}$

$f^{-P} = \{f_i \in f \mid f_i$ is negative and has predicate symbol P$\}$

$H_0^{+P} = \{C_i \in H_0 \mid f_i \in f^{+P}\}$

$H_0^{-P} = \{C_i \in H_0 \mid f_i \in f^{-P}\}$.

Let $P_j(j=1,m)$ be the predicate symbols appearing in f.

We need two lemmas.

Lemma 1 Let P be a predicate symbol not occurring in Th. Suppose $E \leq C$ (Th) and it is not the case that $\vdash_{Th} \forall C$ and P only occurs positively (negatively) in C. Then P can only occur positively (negatively) in E.

Proof Suppose P only occurs positively in C.

Suppose P occurs negatively in E. For some D, $E \leq D$ and $\vdash_{Th} D \equiv C$. P must occur negatively in D.

Let $\mathcal{A}$ be a model of Th and $\forall D$. Define $\mathcal{A}'$ to be that structure which is identical to $\mathcal{A}$ except that it assigns $\emptyset$ to P. Then $\mathcal{A}'$ is also a model of Th since P does not occur in Th, and of D since P occurs negatively in D. Therefore $\mathcal{A}'$ is a model of $\forall C$ as $\vdash_{Th} D \equiv C$ implies that $\vdash_{Th} \forall D \equiv \forall C$. Let $C = C_1 \cup C_2$ where C_2 is the set of literals in C whose predicate symbol is P. As $\mathcal{A}'$ assigns $\emptyset$ to P and is a model of $\forall C$, $\mathcal{A}'$ is a model of $\forall C_1$. Therefore so is $\mathcal{A}$ and so $\mathcal{A}$ is a model of $\forall C$. As $\mathcal{A}$ is an arbitrary model of Th, this contradicts the fact that $\forall C$ is not a consequence of Th.

When P only occurs negatively in C, the proof is similar.

Lemma 2 Let P be a predicate symbol not occurring in Th. Suppose $C_1 \cup C_2$ is a ground clause where C_1 contains no positive (negative) occurrences of P and every literal in C_2 contains a positive (negative)

occurrence of P. Suppose, further that $C_1 \cup C_2$ is not deduceable from Th. If **E** contains no positive (negative) occurrence of P and $E \leq C_1 \cup C_2$ (Th) then $E \leq C_1$ (Th).

Proof We prove only the positive part of the theorem. The negative part follows similarly. For some **E'**, and σ, **E'** = **E** σ and $\vdash_{Th} E' \rightarrow (C_1 \cup C_2)$. **E'** contains no positive occurrence of P. If we show that $\vdash_{Th} E' \rightarrow C_1$, then it will follow at once that $E \leq C_1$ (Th). Let Th' be the set of Skolemisations of members of Th. It is well-known that Th' is a conservative extension of Th. Suppose that **E'** -> C_1 is not deducible from Th. Then neither is it deducible from Th'. Therefore there is a Herbrand model $\mathcal{A}$, of Th' and a substitution σ' such that (**E'** -> C_1) σ' is ground and false in $\mathcal{A}$. Therefore **E'** σ' is true in $\mathcal{A}$ and C_1 σ' = C_1 is false. Since $\vdash_{Th} E' \rightarrow (C_1 \cup C_2)$, $\vdash_{Th'} E' \rightarrow (C_1 \cup C_2)$. Therefore $\vdash_{Th'} E' \sigma' \rightarrow (C_1\sigma' \cup C_2 \sigma')$ and so C_2 σ' (= C_2) must be true in $\mathcal{A}$. Let $\mathcal{A}'$ be obtained from $\mathcal{A}$ by stipulating:

1) If $L \in C_2$, **L** is false in $\mathcal{A}'$.

2) If $L \notin C_2$, L has the same value in $\mathcal{A}'$ as in $\mathcal{A}$.

Since C_2 is a set of positive literals with predicate symbol P and the only possible literals in **E'** σ' with predicate symbol P are negative, **E'** σ' must also be true in $\mathcal{A}'$. If a literal, **L**, in C_1 changes truth value then $\bar{L} \in C_2$ and so $\vdash C_1 \cup C_2$ which contradicts the assumption that $C_1 \cup C_2$ is not deducible from Th. Hence no literal

in C_1 changes truth value and so C_1 is false in $\mathcal{A}'$. By definition, C_2 is false in $\mathcal{A}'$. Therefore $(E' \rightarrow C_1 \vee C_2)$ is false in $\mathcal{A}'$ which contradicts the fact that $\vdash_{Th}, E' \rightarrow (C_1 \vee C_2)$. Therefore $E' \rightarrow C_1$ is deducible from Th, which completes the proof.

We are now in a position to prove two theorems on division of the search space.

Theorem 4 Suppose that no P_j appears in Th, and that $P_1 \ldots P_{m'}$ appear only positively in the e_i if at all $(i=1,n)$ and $P_{m'+1} \ldots P_m$ appear only negatively in the e_i if at all $(i=1,n)$.

Then any solution, $H \subseteq \mathcal{J}_{Th}(H_0)$ is equal to $H^+ \vee H^-$ where

$H^+ \subseteq \bigcup_{j=1}^{m'} \mathcal{J}_{Th}(H_0^{+P_j}) \vee \bigcup_{j=m'+1}^{m} \mathcal{J}_{Th}(H_0^{-P_j}) = \mathcal{J}^+_{Th}(H_0)$, say, and

$H^- \subseteq \bigcup_{j=1}^{m'} \mathcal{J}_{Th}(H_0^{-P_j}) \vee \bigcup_{j=m'+1}^{m} \mathcal{J}_{Th}(H_0^{+P_j}) = \mathcal{J}^-_{Th}(H_0)$ say.

Proof We consider first the case where all the P_j appear positively, that is, $m' = m$ and then $\mathcal{J}^+_{Th}(H_0) = \bigcup_j (H_0^{+P_j})$ and similarly for $\mathcal{J}^-_{Th}(H_0)$. It follows from theorem 1 that it is only necessary to show that if $C \in \mathcal{J}_{Th}(H_0) \setminus (\mathcal{J}_{Th}(H_0^+) \vee \mathcal{J}_{Th}(H_0^-))$, then $\forall C \wedge Th \wedge Irr \wedge \bigwedge_i (e_i \wedge f_i)$ is inconsistent.

Such a C is of the form $\inf_{Th}\{D,E\}$ where there are predicate symbols P_j and $P_{j'}$ such that D contains a positive occurrence of P_j and perhaps negative ones of P_j and $P_{j'}$, and E contains a negative occurrence of $P_{j'}$ and perhaps negative ones of P_j and $P_{j'}$. It is possible that $j=j'$. From lemma 1 and $C \leq D$ (Th), C contains only negative occurrences

of P_j. Further, D is of the form $\bar{e}_i \cup \{f_i\}$ for some i, where f_i contains a positive occurrence of P_j. Hence by lemma 2, $C \leq \bar{e}_i$ (Th) for some i and so C is indeed inconsistent with $\bigwedge_i(e_i \wedge f_i) \wedge$ Th.

The general case may be proved either by a similar detailed examination of another three cases or else by renaming the predicate symbols $P_j(m' < j \leq m)$ as $\neg Q_j(m' < j \leq m)$ so that the general case reduces to the above using the easily proven fact that, with the evident definitions, the solutions to the renaming of a formal problem, of the class considered, are the renamings of the solutions to the formal problem. This concludes the proof.

Under the assumptions of theorem 4, we may consider the predicate symbols P_j as divided into three classes viz. $P_1,\ldots,P_{m_1}$, the symbols whose occurrences in the e_i are all positive and which do have at least one such occurrence and $P_{m_1+1}\ldots P_{m_2}$, the symbols whose occurrences in the e_i are all negative and which do have at least one such occurrence and the $P_{m_2+1}\ldots P_m$, the symbols which have no occurrence in any e_i, then we see from the argument used in the proof of theorem 4 that if D is in H_0, has an occurrence of $P_{m'}(m_2 < m' \leq m)$ and $m'' \neq m'$, and $E \in H_0^{+P_{m''}} \cup H_0^{-P_{m''}}$ then $\inf_{Th}\{C,D\}$ is inconsistent with Th $\wedge \bigwedge_i(e_i \wedge f_i)$. Combining this observation with theorem 4 we obtain:

Corollary 3 If the P_j are categorised as above, and no P_j appears in Th, then any solution H is equal to $H^+ \cup H^- \cup \bigcup_{j=m_1+1}^{m_2} H^{+P_j} \cup \bigcup_{j=m_2+1}^{m} H^{-P_j}$, where:

$$H^{+} \subseteq \bigcup_{j=1}^{m_1} \mathcal{Y}_{Th}(H_0^{+P_j}) \cup \bigcup_{j=m_1+1}^{m_2} \mathcal{Y}_{Th}(H_0^{-P_j}),$$

$$H^{-} \subseteq \bigcup_{j=1}^{m_1} \mathcal{Y}_{Th}(H_0^{-P_j}) \cup \bigcup_{j=m_1+1}^{m_2} \mathcal{Y}_{Th}(H_0^{+P_j}),$$

$$H^{+P_j} \subseteq \mathcal{Y}_{Th}(H_0^{+P_j}) \quad (m_2 < j \leq m),$$

$$H^{-P_j} \subseteq \mathcal{Y}_{Th}(H_0^{-P_j}) \quad (m_2 < j \leq m).$$

Chapter 5 Applications and extensions

1. Some philosophical remarks

There is a connection between problems of justification and discovery. Any hypothesis discovery scheme is, as previously remarked, only a part of a total system. It is necessary that hypotheses suggested by the scheme be justified (or criticized or whatever). In fact we will show that, under suitable conditions, best explanations, as described above, with almost arbitrary $\rightsquigarrow$, are acceptable in the sense of Hintikka and Hilpinen (1966), whose work is in the spirit of Carnap. They wish to give an analysis of the concept of probable knowledge. Knowledge of h in the light of evidence e, is defined by:

$$K(h,e) \equiv Ac(h,e) \wedge h.$$

Ac(h,e) is to mean that e gives h enough support to make it acceptable. The naive analysis of Ac(h,e) is in terms of high probability:

$$Ac(h,e) \equiv P(h,e) > 1 - \varepsilon \text{ where } \varepsilon \geq 0.5.$$

Unfortunately, the naive definition leads to an inconsistency with some generally accepted closure principles. In particular, Hempel (1962) has formulated these conditions:

CA1: If $Ac(h_1,e)$ and .... and $Ac(h_n,e)$ and if $\vdash (h_1 \wedge \ldots \wedge h_n) \rightarrow h$ then $Ac(h,e)$.

CA2: The set $\{h \mid Ac(h,e)\}$ is logically consistent.

Condition CA1 fails because the multiplication theorem for probabilities makes it possible that $P(h_1,e) > 1 - \varepsilon$ and $P(h_2,e) > 1 - \varepsilon$ but $P(h_1 \wedge h_2,e) \leq 1 - \varepsilon$.

Condition CA2 fails since one can, for example, find hypotheses $h_i(i=1,n)$ such that $P(h_i,e) > 1 - \varepsilon$ $(i=1,n)$ and $P(\bigvee_i \neg h_i,e) > 1 - \varepsilon$. However $\{h_1,\ldots,h_n, \bigvee_i \neg h_i\}$ is inconsistent. These difficulties come under the general heading of the lottery paradox of Kyburg (1961).

Hintikka and Hilpinen solve these problems in the context of a simple language with exactly k monadic predicate symbols P_i, some fixed number of constant symbols but no other function symbols. The authors seem to make a background assumption that different constants denote different individuals in the universe. The lack of detail in the article makes this unclear. There is also the assumption that every universe contains infinitely many individuals. This is however a matter of convenience, and can, it appears, be dropped in a more detailed account.

In the context of an infinite universe, it seems that the assumption that different constants denote different individuals can be dropped. For if a and b are different constants then it seems that, on a priori grounds, probability(a=b) = 0.

Using the predicate symbols $P_i(i=1,k)$ one can define $K = 2^k$ different kinds of individuals using complex predicates $Ct_j(j=1,k)$ which have definitions of the form:

$$Ct_j(x) \equiv \bigwedge_{i=1}^{k} (\pm)\, P_i(x).$$

By saying of each Ct_j whether or not it is instantiated, different world descriptions $C_l (l=1,2^k)$ can be given. To give a more exact definition we introduce the symbols +, -, $\mathcal{B}$ and the metalinguistic variables α and β, possibly with suffixes, to range over them. The symbols have an ordering $\sqsubseteq = \{< \mathcal{B}, +, >, < \mathcal{B}, - >\}$. Pseudo-formulae are defined by the conditions that if h is a formula then α h is a pseudo-formula, if h_1 and h_2 are pseudo-formulae so are $\neg h_1, h_1 \wedge h_2$ and $h_1 \vee h_2$. (The implication sign is regarded as an abbreviation.) Pseudo-formulae are abbreviations for certain formulae. It is sufficient, for our purpose, to give examples of the use of the symbols; rather than give a detailed definition:

$+P_1(x) \wedge \neg +P_2(x) \wedge \mathcal{B} P_3(x)$ abbreviates $P_1(x) \wedge \neg P_2(x)$.

$\neg -P_1(x) \vee \neg \mathcal{B} P_2(x)$ denotes $P_1(x)$.

Roughly - denotes $\neg$, + should be ignored and $\mathcal{B}$ means that the immediately following pseudo-formula should be removed. (Think of them as being analogous to +1, -1 and 0.) Nasty cases like $\mathcal{B} \exists x P_1(x) \vee \mathcal{B} P_2(x)$ are handled by the rule that the empty conjunction abbreviates an arbitrary tautology and the empty disjunction abbreviates the negation of an arbitrary tautology. Thus $\mathcal{B} \exists x P_1(x) \vee \mathcal{B} P_2(x)$ abbreviates $\neg (\forall x P_1(x) \vee \exists x \neg P_1(x))$, say. In these terms the Ct_j have definitions of the form:

$$Ct_j(x) \equiv \bigwedge_{i=1}^{k} \alpha_i P_i(x) \quad (\alpha_i \neq \mathcal{B}).$$

They are called attributive constituents.

Constituents, C_1 have definitions of the form:

$$C_1 \equiv [\bigwedge_{j=1}^{K} \alpha_j \exists x Ct_j(x)] \wedge [\forall x \bigvee_{j=1}^{K} \alpha_j Ct_j(x)] \ (\alpha_j \neq -).$$

Each constituent is a description of a possible world, in so far as this is possible in the monadic calculus.

Only a certain type of evidence, e, is considered. It is assumed that there are n distinct constant symbols a_i, such that

$e = \bigwedge_{i'=1}^{n} \bigwedge_{i=1}^{k} \alpha_{ii'} P_i(a_{i'})$ (where $\alpha_{ii'} \neq \mathcal{B}$).

Using a probability function P due to Hintikka, the authors find by an effective means a constant $n_0 = n_0(\varepsilon, k)$ such that Ac may be defined by:

$Ac(h,e) \equiv P(h,e) > 1 - \varepsilon$ and $n \geq n_0$, where h is a general sentence. This definition satisfies conditions CA1 and CA2 applied to general sentences only. That is:

1) If $Ac(h_1,e)$ and ... and $Ac(h_n,e)$ and if $h_1,\ldots,h_n$ and h are general sentences and $\vdash (h_1 \wedge \ldots \wedge h_n) \rightarrow h$ then $Ac(h,e)$.

2) The set $\{h \mid Ac(h,e)$ and h is general$\}$ is consistent.

The function P has an important difference from the probability function of Carnap. It assigns non-zero values to generalisations. It is in fact one of a whole family of probability functions, Hintikka's α-continuum (Hintikka (1965a)), for which the same results

can be established.

In particular, if we write e in the form $\bigwedge_{i'=1}^{n} Ct_{n_{i'}}(a_{i'})$ and consider the unique constituent, $C_{l(e)} \equiv [\bigwedge_{i'=1}^{n} \exists x Ct_{n_{i'}}(x)] \wedge [\forall x \bigvee_{i=1}^{n} Ct_{n_i}(x)]$, then one may show that if $n \geq n_0$ then $Ac(C_{l(e)},e)$ and further that, if $n \geq n_0$, $Ac(h,e)$ if and only if $\vdash C_{l(e)} \rightarrow h$.

It is necessary to extend this analysis to other sentences than general ones.

Every general sentence h can be expressed as

$$h \equiv \bigvee_{l=1}^{2^K} \alpha_l C_l \text{ where } \alpha_l \neq -,$$

as is well-known (Hintikka, 1953).

One can show that every sentence, h, can be expressed as $h \equiv \bigvee_{l=1}^{2^K} \alpha_l (C_l \wedge h_l)$ where no $\alpha_l = -$, each h_l is singular that is, contains no variable whether bound or not. (Singular just means ground.)

Suppose that $h \equiv \bigvee_{l=1}^{2^K} \alpha_l (C_l \wedge h_l)$ and $h' \equiv \bigvee_{l=1}^{2^K} \alpha'_l (C_l \wedge h'_l)$ where no α_l or α'_l is -. Then $\vdash h \rightarrow h'$ if and only if whenever $\alpha_l \neq \mathcal{O}$, $\alpha'_l \neq \mathcal{O}$ and $\vdash C_l \wedge h_l \rightarrow h'_l$.

Define a partial function of two variables, max, on $\{+, -, \mathcal{O}\}$ by: If $\alpha \exists \alpha'$ then $\max(\alpha, \alpha') = \alpha$ and if $\alpha' \exists \alpha$, $\max(\alpha, \alpha') = \alpha'$. Otherwise $\max(\alpha, \alpha')$ is undefined.

Define min similarly, with $\exists$ replaced by $\sqsubseteq$.

Then $h \wedge h' \equiv \bigvee_l \min(\alpha_l, \alpha'_l)\ (C_l \wedge h_l \wedge h'_l)$,

$h \wedge h' \equiv \bigvee_l \max(\alpha_l, \alpha'_l)\ (C_l \wedge (h_l \vee h'_l))$.

The right hand sides of these equations are defined.

h is defined to be an e-sentence iff it is equivalent to a sentence of the form $\bigvee_l \alpha_l(C_l \wedge h_l)$ where $\vdash e \rightarrow h_l$ and no α_l is -. $\bigvee_l \alpha_l(C_l \wedge h_l)$ is a representation of h. From the above we see that if h and h' are e-sentences, so are $h \wedge h'$ and $h \vee h'$.

Any general sentence $h \equiv \bigvee \alpha_l C_l$ (no α_l = -) is an e-sentence, since $h \equiv \bigvee \alpha_l(C_l \wedge (e \vee \neg e))$. If e' is singular and $\vdash e \rightarrow e'$ then e' is an e-sentence since $e' \equiv \bigvee(C_l \wedge e')$.

We are now in a position to define acceptability of e-sentences on the grounds of evidence e:

Let $h \equiv \bigvee_l \alpha_l(C_l \wedge e_l)$ be a representation of h as an e-sentence. Then:

$$Ac(h,e) \equiv Ac(\bigvee_l \alpha_l C_l, e).$$

The definition is independent of the representation, since if $\bigvee_l \alpha'_l(C_l \wedge e'_l)$ is another representation then, as $\bigvee_l \alpha_l(C_l \wedge e_l) \equiv \bigvee_l \alpha'_l(C_l \wedge e'_l)$, $\alpha_l = \alpha'_l$ for all l. Further the new definition of Ac extends the old one.

If $\vdash e \rightarrow e'$ for some singular e', then $Ac(e',e) \equiv$

$Ac(\bigvee_1(C_1 \wedge e'),e) \equiv Ac(\bigvee_1 C_1,e) \equiv n \geq n_0$. If this result is counter-intuitive, it is because $Ac(\bigvee_1 C_1,e) \equiv n \geq n_0$ is counterintuitive. This seems to us to be a slight, but easily corrected, fault of the definition of Hintikka and Hilpinen.

There is an equivalent definition in terms of high probability and large enough evidence. Suppose h is an e-sentence with representation $\bigvee_1 \alpha_1(C_1 \wedge e_1)$. We have:

$$P(h,e) = P(\bigvee_1 \alpha_1(C_1 \wedge e_1),e)$$
$$= \sum_{\alpha_1 \neq 0} P(C_1 \wedge e_1,e) \text{ (as } \vdash \neg(C_1 \wedge C_{1'}) \text{ if } 1 \neq 1').$$
$$= \sum_{\alpha_1 \neq 0} P(C_1,e) \text{ (as } \vdash e \rightarrow e_1).$$
$$= P(\bigvee_1 \alpha_1 C_1,e) \text{ (as } \vdash \neg(C_1 \wedge C_{1'}), \text{ if } 1 \neq 1').$$

Therefore $Ac(h,e) \equiv Ac(\bigvee \alpha_1 C_1,e)$
$\equiv P(\bigvee \alpha_1 C_1,e) > 1 - \varepsilon$ and $n \geq n_0$.
$\equiv P(\bigvee \alpha_1(C_1 \wedge e_1),e) > 1 - \varepsilon$ and $n \geq n_0$, (by the above).
$\equiv P(h,e) > 1 - \varepsilon$ and $n \geq n_0$.
$\equiv \alpha_{1(e)} = +$ and $n \geq n_0$ (by previous remarks).

Hempel's conditions may now be demonstrated for e-sentences.

Suppose that h_1 and h_2 are e-sentences with representations $\bigvee \alpha_1(C_1 \wedge h_1)$ and $\bigvee \alpha'_1(C_1 \wedge h'_1)$. Then $h_1 \wedge h_2$ has a representation $\bigvee \min(\alpha_1, \alpha'_1)(C_1 \wedge (h_1 \wedge h'_1))$. So if $Ac(h_1,e)$ and $Ac(h_2,e)$ then $\alpha_{1(e)} = \alpha'_{1(e)} = \min(\alpha_{1(e)}, \alpha'_{1(e)}) = +$ and $n \geq n_0$. Therefore

$Ac(h_1 \wedge h_2, e)$. Suppose that $\vdash h_1 \rightarrow h_2$. Then $P(h_2,e) \geq P(h_1,e)$. Therefore if $Ac(h_1,e)$, $Ac(h_2,e)$. This verifies condition CA1.

If $Ac(h,e)$ then $\alpha_{1(e)} = +$, in any representation of h, and therefore $\vdash C \wedge e \rightarrow h$. As $C \wedge e$ is consistent so too therefore is $\{h \mid Ac(h,e)\}$. This verifies condition CA2.

We are now in a position to link up these results with our hypothesis discovery methods. There is one small point. We will temporarily use e' rather than e to stand for the phenomena and keep e for the uses described above.

Only certain special e' and f are considered.

$e' = \{\alpha_{ki'}P_k(a_{i'}) \mid i' = 1,n\}$ where the $a_{i'}$ are all different constant terms and no $\alpha_{ki'}$ is $\mathcal{B}$.

$Ev(\alpha_{ki'}P_k(a_{i'})) = \bigwedge_{i=1}^{k=1} \alpha_{ii'}P_i(a_{i'})$ where no $\alpha_{ii'}$ is $\mathcal{B}$. Then e and $C_{1(e)}$ are defined as above, and the above results are available. Note that $\vdash e \equiv \bigwedge_{i'=1}^{n}(e_{i'} \wedge f_{i'})$. This e' and f satisfy the conditions for providing a discovery problem.

The next theorem shows that if a certain mild restriction on $\rightarrow$ holds, then any solution is acceptable. This condition is that if H is a solution and C is in H, then for some C_0 in H_0, $C \leq C_0$. Thus H must not contain any completely unnecessary clause.

<u>Theorem 1</u> Suppose that $H \leq H_0$, $\forall H \wedge e$ is consistent, $n \geq n_0$ and that if C is in H there is a C_0 in H_0 such that $C \leq C_0$. Then $Ac(\forall H, e)$.

Proof Let D be in H. We may write $D = \bigcup_{s=1}^{t} D_s \cup D_0$ where

1) If $D_0 \neq \emptyset$, D_0 is ground.

2) $\bigcup_s D_s$ has no occurrences of constants.

3) If $t \geq s > 0$, $D_s \neq \emptyset$.

4) If $s \neq s'$, there is no variable common to D_s and $D_{s'}$.

5) Each D_s has exactly one variable symbol, x_s, say $(s > 0)$.

Consequently, $\vdash \forall D \equiv (\bigvee_{s=1}^{t} \forall_{x_s} D_s) \wedge D_0$. As $\forall D$ is consistent with e, so is some $\forall D_s (t \geq s \geq 0)$.

Suppose D_0 is consistent with e. Then $D_0 \neq \emptyset$ and as D subsumes some member, $C_{i'}$, say, of H_0, $D_0 \leq C_{i'}$. From the consistency of D_0 with e, it follows that $\alpha_{ki'} P_k(a_{i'}) \in D_0$ and so $\vdash e \rightarrow D_0$. Therefore as $n \geq n_0$, $Ac(D_0, e)$. As $\forall D$ is also an e-sentence and $\vdash D_0 \rightarrow \forall D$, $Ac(\forall D, e)$.

Suppose that $\forall_{x_s} D_s$ is consistent with $e (s > 0)$. As $D \leq C_{i'}$, $D_s \leq C_{i'} \equiv \neg Ev(\alpha_{ki'} P_k(a_{i'})) \vee \alpha_{ki'} P_k(a_{i'})$. For consistency to hold, $\alpha_{ki'} P_k(x_s)$ must be in D_s. Consider some arbitrary phenomenon, $\alpha_{ki'} P_k(s_{i''})$. Suppose $\alpha_{ki'} = \alpha_{ki''}$, then:

$$\vdash \bigwedge_{i=1}^{k} \alpha_{ii''} P_i(x_s) \rightarrow D_s.$$

Suppose, on the other hand that $\alpha_{ki'} \neq \alpha_{ki''}$. Then, by the consistency of $\forall_{x_s} D_s$ with e, $D_s \not\leq \neg Ev(\alpha_{ki''} P_k(a_{i''}))$. Therefore $\alpha_{ii''} P_i(x_s)$ is in D_s for some i. So:

$$\vdash \bigwedge_{i=1}^{k} \alpha_{ii^{*}} P_i(x_s) \rightarrow D_s .$$

Consequently, no matter what i^{*} is,

$$\vdash \bigwedge_{i=1}^{k} \alpha_{ii^{*}} P_i(x_s) \rightarrow D_s .$$

Therefore, $\vdash \forall x \bigvee_{i^{*}=1}^{n} \bigwedge_{i=1}^{k} \alpha_{ii^{*}} P_i(x) \rightarrow \forall x_s D_s .$

So by the definition of $C_{1(e)}$, $\vdash C_{1(e)} \rightarrow \forall x_s D_s$. We then see that $Ac(\forall x_s, D_s, e)$ and so $Ac(\forall D, e)$.

As D was any member of H, it follows that $Ac(\forall H, e)$ which concludes the proof.

Corollary 2 Suppose that H satisfies the following conditions, where $\prec$ is a lexicographic extension of $\prec_c$:

1) $H \leq H_0$

2) $\forall H \wedge e$ is consistent.

3) H is minimal wrt. $\prec$ amongst those sets of clauses satisfying conditions one and two.

If $n \geq n_0$ then $Ac(\forall H, e)$.

Proof It is immediate that H satisfies all the conditions of theorem 1, except perhaps, that if C is in H there is a C_0 in H_0 such that $C \leq C_0$. Suppose, to the contrary, that C is in H and $C \not\leq C_0$ for every C_0 in H. Then $H' = H \cup \{C\}$ satisfies conditions one and two of the hypothesis, but $H' \prec H$ and $H \not\prec H'$, which contradicts condition three. Therefore

all conditions are satisfied and the conclusion follows at once.

It follows from the work of Hintikka and Hilpinen that n_0 is calculable from the number of predicate symbols, k and also ε. Therefore for sets of clauses satisfying the conditions of theorem 1, there is a method of deciding acceptability. We have therefore, for a restricted class of cases, answers to questions H1 (on when a hypothesis is justified) and H2 (on how to tell if a hypothesis is justified) of chapter 1 which satisfy the strong coherence condition that generated hypotheses be acceptable. This answer to H2 is evidently complete and consistent.

There are a number of insufficiencies in our analysis. First the notion of acceptance is based on just one of Hintikka's inductive systems. One would really want to have results not only for the α-continuum but also the whole two-dimensional α - λ-continuum (Hintikka, 1966). Negative results would hold for that part corresponding to Carnap's λ-continuum (Carnap 1952), since no generalisation, containing variables, is acceptable there.

It is also natural to try to dispense with the assumption that all the individuals are completely observed, that is that every $\alpha_{ii'} \neq \beta$.

Both of these insufficiencies could probably be remedied with the aid of Hilpinen's (1968) monograph. Only then could we have a firm conclusion in the case of monadic logic. It would still remain

to extend the results to richer languages and this in turn requires a generalisation of the definitions of acceptance not yet attempted. Perhaps, however, some use could be made of the work on axioms for rules of acceptance (Kemeny 1953, Putnam 1963) but this is a mere speculation.

Important, here, would be acceptance relative to other than singular propositions. One would wish to know whether or not $Ac(\forall H, Th \wedge e)$ for example. If Th contained axioms for equality, e could contain distinctness information, of the form $a_i \neq a_{i'}$ (when $i \neq i'$, of course). This would remove a difficulty, alluded to above, in the presentation of Hintikka and Hilpinen.

We turn next to arguments designed to show that although it is necessary that explanation is justified, one cannot, without some difficulty, formulate conditions sufficient for the rational choice of an explanation in terms of justification. These arguments are elaborations of those advanced in chapter 2 to provide an opening for the use of simplicity.

A reasonable-seeming Carnapian move would be to set $H_1 \succ' H_2$ iff H_1 has a greater probability (in some sense) than H_2, given the knowledge $Th \wedge Irr$ and all the phenomena f and their circumstances e. However in this case H_0 will have a probability of one, and so is a best solution. If the notion of probability being used is reasonable then a hypothesis H will have unit probability relative to all knowledge etc. if and only if it follows from Th and Irr and e and f. Therefore

all the solutions will follow from what is given. Thus nothing new could ever be hypothesized which is absurd.

A more sympathetic formulation would require not only that an explanation generalise H_0 and be consistent with Th and Irr and $\bigwedge(e_i \wedge f_i)$, but also that it be general as opposed to ground. In view of previously discussed difficulties with the notion of a general law and since in this case, we do not wish to allow ground clauses as a degenerate case, let us specify that Th and Irr are empty, that no function symbols, other than constants, occur in e or f and that H is general if it contains no constants. Let $e' = \bigwedge_{i=1}^{n}(e_i \wedge f_i)$.

It seems reasonable to assume that if H_1 is general and $H_1 \leq H_2$ but $H_2 \not\leq H_1$, then probability $(H_1,e') <$ probability(H_2,e').

In these circumstances the solutions are also maximal with respect to $\leq$.

Let H_1 be obtained from H_0 by replacing distinct constants by distinct variables. We may see that if $H \leq H_0$ and H is general, then $H \leq H_1$. As solutions are maximal with respect to $\leq$ every solution must in fact be equivalent to H_1. There are therefore two possibilities: either $H_1 \wedge \bigwedge_i (e_i \wedge f_i)$ is inconsistent and there is no solution or else H_1 is a solution and any other one is equivalent to it. This seems counter-intuitive. For example, suppose that $f = \{Q(a_i) \mid i=1,2n\}$, $Ev(Q(a_{2i})) = P_1(a_{2i}) \wedge P_2(a_{2i})$ $(i=1,n)$ and $Ev(Q(a_{2i-1})) = P_1(a_{2i-1}) \wedge P_3(a_{2i-1})$ $(i=1,n)$.

Here, $H_1 \sim \{\{\bar{P}_1(x) , \bar{P}_2(x) , Q(x)\}, \{\bar{P}_1(x) , \bar{P}_3(x) , Q(x)\}\}$.

For large enough n, one would expect a guess to be made that both P_2 and P_3 are irrelevant to the truth of Q.

A reasonable Popperian move would be to set $H_1 \prec' H_2$ iff H_1 is more falsifiable than H_2 given Th $\wedge$ Irr and f and e. In this case it is quite clear that there can never be a solution. If H_2 is proposed as a solution, we need merely find an H_1 such that $\forall(H_1 \cup H_2)$ is consistent with Th $\wedge$ Irr $\wedge \bigwedge_i (e_i \wedge f_i)$, and $\forall(H_1 \cup H_2)$ is more falsifiable than $\forall H_1$. This can always be done, as H_1 can contain arbitrary assertions as long as it has no vocabulary in common with any of H_2, Th, Irr or $\bigwedge (e_i \wedge f_i)$. We need a more sympathetic interpretation.

H_1 is an <u>irredundant</u> generalisation of H_2 relative to Th iff $H_1 \leq H_2$ (Th) and if $H_3 \subseteq H_1$ and $H_3 \leq H_2$ (Th) then $H_3 = H_1$. (This use of the word "irredundant" is distinct from that in chapter 4.) It is now required as an extra condition for a solution that H be an irredundant generalisation of H_0. This will prevent the above absurdities. It seems reasonable to assume that if $H_1 \leq H_2$ (Th) but H_1 does not follow logically from $H_2, \bigwedge_i (e_i \wedge f_i)$, Th and Irr then H_1 is more falsifiable than H_2 relative to e and f. Any solution will then be maximally general relative to Th. In other words, the most falsifiable hypotheses satisfy these three conditions (with $\prec$ equal to generalisation relative to Th):

1) H is an irredundant generalisation of H_0, relative to Th.

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

3) H is minimal, with respect to $\prec$, amongst those sets of clauses satisfying conditions one and two.

Unfortunately, we do not know if there are any solutions, even in the case where Th and Irr are empty and there are no function symbols, other than constants, in e and f. On the other hand if we take $\prec$ to be the lexicographic product of the simplicity ordering, $\prec_s$, (the number of symbol occurrences) and generalisation relative to Th then there is always a finite number of solutions, to within equivalence relative to Th. This can be seen using a technique similar to that in chapter 6, section 5.

It seems worthwhile spending some effort on the problem where $\prec$ is relative generalisation. If reasonable hypotheses could be produced with this niceness ordering, one would have a good argument against the necessity of an explicit syntactical simplicity ordering (Goodman, 1961). Alternatively, one might be able to produce arguments correlating simplicity with falsifiability in accordance with the views of Popper.

Finally, we give an example of a loose connection between confirmation theory and hypothesis discovery methods. Sometimes arguments are produced that certain evidence confirms certain hypotheses, against one's intuitions. These are paradoxes of confirmation. Two famous ones are Goodman's (1965) and Hempel's (1945). It may be

possible to show that the same hypotheses could be discovered from the same evidence and that this too is paradoxical. Such is the case with Goodman's paradox, of which a brief version can easily be stated. Suppose many emeralds have been examined and all of them are found to be green. This would seem to strongly confirm the hypothesis that all emeralds are green. All the emeralds must have been examined before some time, say the year 2000. Call a thing grue if and only if it has been examined before the year 2000, and found to be green, or if it has not been examined before then and is blue. Evidently all the emeralds examined are also grue. So just as strong confirmation is provided for the hypothesis that all emeralds are grue. These two hypotheses are contradictory. Although the grue hypothesis seems absurd, the absurdity has proved highly resistant to attempts at dissolution. Perhaps the best proposal was given by Goodman himself when he proposed the paradox.

Let us take $f = \{\text{Green}(em_i) \mid i=1,n\}$ and
$\text{Ev}(\text{Green}(em_i)) = \text{Emerald}(em_i) \wedge \text{Examinedby}(em_i, 2000)$, $(i=1,n)$.

Certainly, $\{\{\neg\, \text{Emerald}(x), \text{Green}(x)\}\} \leq H_0$.

Now suppose Th contains the definition:

$$\text{Grue}(x) \equiv (\text{Examinedby}(x,2000) \rightarrow \text{Green}(x)) \wedge (\neg\, \text{Examinedby}(x,2000) \rightarrow \text{Blue}(x)).$$

Then, $\{\{\neg\, \text{Emerald}(x), \text{Grue}(x)\}\} \leq H_0\ (\text{Th})$.

Simple syntactical definitions of simplicity will not distinguish

the two hypotheses. One might hope to distinguish the Grue hypothesis as more complex, if complexity is measured after Grue is replaced by its definition. This will not do if Th is regarded as an unstructured set of sentences. Let Grue $\equiv_{\Delta}$ (Green,Blue) be the form of the definition of Grue. Suppose Th contains in addition the statement, Bleen $\equiv_{\Delta}$ (Blue,Green). Then both Green $\equiv_{\Delta}$ (Grue,Bleen) and Blue $\equiv_{\Delta}$ (Bleen,Grue) are logical consequences of Th and so there is complete symmetry between Grue and Green.

It seems therefore that Th must be given some structure and Grue be regarded as given by a definition. Even then, the proposal to count simplicity after the replacement of Grue by its definition seems to be an unwarranted bias in favour of particular predicates as, say, observational rather than theoretical. We are being lead, quite quickly, into wider issues. Note for example that focussing only on the stage of hypothesis formation causes distortion. There must also be a stage of forming definitions and theory of how definitions interact with discovery and justification. Goodman's proposed solution consisted of proposals involving these stages. All in all, the discussion of the paradox of discovery is much the same as that of confirmation. To put it briefly, the confirmation paradox concerns which predicates should be projected; the discovery paradox concerns how to select predicates for projection.

There is one practical point. The "wrong" predicates should somehow be avoided at either the stage of definition or else that of discovery. For if any definition is allowed and any of the simplest explanatory

hypotheses, there will generally be infinitely many hypotheses still to be eliminated at the stage of justification. For example: consider the definition

$$\text{Gruet}(x,t) \equiv (\text{Examinedby}(x,t) \rightarrow \text{Green}(x)) \wedge (\neg\, \text{Examinedby}(x,t) \rightarrow \text{Blue}(x)).$$

Now none of the infinitely many hypotheses $\text{Emerald}(x) \rightarrow \text{Gruet}(x,t_0)$, - where t_0 is a constant greater than 2000 - will be eliminated.

2. Two extensions to a more complex kind of theory, Th

2.1 Sorted languages

The first extension concerns languages with several disjoint sorts. It is a matter of folklore that the usual unification procedure automatically takes account of the sort restriction. Similarly the procedure for generalising two literals works when dealing with sorts, provided only that when matching distinct terms, the new variable substituted should be of the same sort as the old ones. With this one difference, all of the theory goes through without any trouble. Instead of putting the sort restrictions in the language we could let Th include the usual sort axioms using unary predicate symbols. One could then show that this leads to essentially the same results as the somewhat neater linguistic procedure.

2.2 Algorithms for a simple kind of theory: ground clauses

The second extension concerns algorithms for finding l.g.g's when Th is an arbitrary consistent, finite non-empty set of ground literals. We will give the essential theorems and algorithms for literals and clauses.

The first theorem gives the result for literals.

We define a function $\inf_{Th}$ by:

If $\vdash_{Th} \forall M$ then $\inf_{Th}\{M,N\} = N$. Otherwise, if $\vdash_{Th} \forall N$ then $\inf_{Th}\{M,N\} = M$. Otherwise, if M and N have the same predicate letter and sign then, $\inf_{Th}\{M,N\} = \inf\{M,N\}$. Otherwise, $\inf_{Th}\{M,N\} = \bar{L}_1$

(L_1 is some fixed member of Th).

Theorem 1 1 $L \leq M$ (Th) iff $\vdash_{Th} \neg \forall L$ or $\vdash_{Th} \forall M$ or for some substitution σ, $L\sigma = M$.

2 $L \sim M$ (Th) iff either $\vdash_{Th} \forall M \wedge \forall L$ or else $\vdash_{Th} \neg \forall M \wedge \neg \forall L$ or else $L \sim M$.

3 Every pair of literals M and N has a least generalisation relative to Th, $\inf_{Th}\{M, N\}$.

Proof 1 $L \leq M$ (Th) iff for some σ $\vdash_{Th} L\sigma \rightarrow M$

iff $\vdash_{Th} L\sigma\delta \rightarrow M\delta$ (where $\delta = \{a_1()/x_1, \ldots, a_n()/x_n\}$, the x_i are the free variables of $L\sigma \rightarrow M$ and none of the a_i occur in Th or $L\sigma \rightarrow M$)

iff $\emptyset \in \mathcal{R}(\{L_i \mid L_i \in Th\} \cup \{L\sigma\delta, \bar{M}\delta\})$.

There are now three possibilities, as Th is consistent. First, for some i, $\emptyset \in \mathcal{R}(\{L_i, L\sigma\delta\})$. This is equivalent to $\vdash_{Th} \neg \forall L$.

Second, for some i, $\emptyset \in \mathcal{R}(\{L_i, \bar{M}\delta\})$. This is equivalent to $\vdash_{Th} \forall M$.

Lastly, $\emptyset \in \mathcal{R}(\{L\sigma\delta, \bar{M}\delta\})$. This is true iff $L\sigma\delta = M\delta$ which is equivalent to $L\sigma = M$. This concludes the proof of the first part.

2 Suppose $L \sim M$ (Th). Then as $L \leq M$ (Th) either $\vdash_{Th} \neg \forall L$ or $\vdash_{Th} \forall M$ or, for some σ, $L\sigma = M$.

Suppose that $\vdash_{Th} \neg\forall L$. As $M \leq L$ (Th) either $\vdash_{Th} \neg\forall M$ or $\vdash_{Th} \forall L$, for some μ, $M/\mu = L$. The second case is impossible and if $M/\mu = L$ then, as $\vdash_{Th} \neg\forall L$, $\vdash_{Th} \neg\forall M$. Hence in this case $\vdash_{Th} \neg\forall L \wedge \neg\forall M$.

Suppose that $\vdash_{Th} \forall M$. As $M \leq L$ (Th) the possibilities are that $\vdash_{Th} \forall L$ or $M/\mu = L$ for some μ. Now $M/\mu = L$ and $\vdash_{Th} \forall M$ implies $\vdash_{Th} \forall L$. Hence in this case $\vdash_{Th} \forall L \wedge \forall M$.

Suppose that $L\sigma = M$. As $M \leq L$ (Th) either $\vdash_{Th} \neg\forall M$ or $\vdash_{Th} \forall L$ or for some μ, $M/\mu = L$. In the first case it follows that $\vdash_{Th} \neg\forall L$, the second that $\vdash_{Th} \forall M$ and in the third that $L \sim M$. Hence in this case $\vdash_{Th} \forall L \vee \forall M$ or $\vdash_{Th} \neg\forall L \wedge \neg\forall M$ or $L \sim M$. This concludes the proof of the second part.

3 If $\vdash_{Th} \forall M$ then $N \leq M$ (Th) and so $N = \inf_{Th}\{M,N\}$ is a l.g.g. of M and N relative to Th. If $\vdash_{Th} \forall N$, the proof is similar.

Suppose that M and N have the same predicate letter and sign and neither $\vdash_{Th} \forall M$ nor $\vdash_{Th} \forall N$ then $\inf\{M,N\}$ is defined and as $\inf\{M,N\} \leq M$, $\inf\{M,N\} \leq M$ (Th). Similarly, $\inf\{M,N\} \leq N$ (Th). Suppose that $L \leq M$ (Th) and $L \leq N$ (Th). Either $\vdash_{Th} \neg\forall L$ or else there are σ, μ such that $L\sigma = M$ and $L/\mu = N$. In the first case $L \leq \inf\{M,N\}$ (Th). In the second, $L \leq \inf\{M,N\}$ and so $L \leq \inf\{M,N\}$ (Th). Thus $\inf_{Th}\{M,N\}$ is a l.g.g. of M and N relative to Th in this case.

Suppose that M and N differ in predicate letter or sign and that neither $\vdash_{Th} \forall M$ nor $\vdash_{Th} \forall N$. Then if $L \leq M$ (Th) and $L \leq N$ (Th),

the only possibility is that $\vdash \neg\forall L$. Then $L \leq L_1$ (Th). Since $L_1 \in Th$, $\vdash_{Th} \neg\forall \bar{L}_1$ and so $\bar{L}_1 \leq M$ (Th) and $\bar{L}_1 \leq N$ (Th). Therefore $\inf_{Th}\{M,N\}$ is a l.g.g. of M and N relative to Th in this, the last case.

This concludes the proof of the theorem.

Let $\overline{Th}$ be the clause $\{\bar{L} \mid L \in Th\}$. It follows from the characterisation of relative generalisation given by theorem 3.1.3.1 that $C \leq D$ (Th), iff D is a tautology, $C\sigma \subseteq D \cup \overline{Th}$ for some σ or $Th \cap D \neq \emptyset$.

Let L_1 be some member of Th. Reduction will be defined relative to this choice. A clause C is reduced relative to Th iff $C = \{L_1\}$ or C is not a tautology, $C \cap Th = \emptyset$ and for any $C' \subseteq C$, $C \leq C'$ (Th) implies $C = C'$.

A clause, E is a <u>reduction</u> of a clause C, relative to Th, iff when $C \cap Th \neq \emptyset$ or C is a tautology, $E = \{L_1\}$ and, otherwise E is a reduced subset of C equivalent to C.

We define a function $\inf_{Th}\{C,D\}$. If D is a tautology or if $D \cap Th \neq \emptyset$ then $\inf_{Th}\{C,D\} = C$. Otherwise, if C is a tautology or if $C \cap Th \neq \emptyset$ then $\inf\{C,D\} = D$. Otherwise, $\inf_{Th}\{C,D\} = \inf\{C \cup \overline{Th}, D \cup \overline{Th}\}$.

There should be no confusion between the function defined here and the function $\inf_{Th}$ defined above on sets of literals with two elements. Notice that $\inf_{Th}\{L,M\} \sim \inf_{Th}\{\{L\},\{M\}\}$ (Th).

<u>Theorem 2 1</u> The following algorithm stops. It gives a clause E_1 which is a reduction of C, relative to Th:

1) Set E_2 to C.

2) If E_2 is a tautology or $E_2 \cap Th \neq \emptyset$ then set E_1 to $\{L_1\}$ and stop.

3) Set E_1 to $\emptyset$.

4) If E_2 is empty, stop.

5) Choose a literal, L, in E_2.

6) If there is a substitution, σ, such that $E_2\sigma \subseteq (E_1 \cup E_2 \cup \overline{Th}) \setminus \{L\}$ and $M\sigma = M$ for every **literal** M in E_1 then change E_2 to $E_2\sigma \setminus (E_1 \cup \overline{Th})$. Otherwise remove L from E_2 and add it to E_1.

7) Go to 4.

2 If $C \sim D$ (Th) and C and D are reduced, then they are alphabetic variants.

3 Every pair of clauses C and D has a l.g.g. relative to Th, $\inf_{Th}\{C,D\}$.

Proof 1 If C is a tautology or $C \cap Th \neq \emptyset$ then the algorithm outputs $\{L_1\}$ which is reduced and equivalent, relative to Th, to C. Otherwise, the proof that the algorithm works is a slight elaboration of the proof of theorem 3.3.1.1.

2 Suppose that C and D are reduced and equivalent, relative to Th. If $C = \{L_1\}$ then as $C \leq D$ (Th) either $C\sigma \subseteq D \cup \overline{Th}$ for some σ, D is a

tautology or $D \wedge Th \neq \emptyset$. In the first case, $L_1 = \mathbf{L}_1\sigma \in D \cup \overline{Th}$. As $L_1 \in Th$ and Th is consistent, $\mathbf{L}_1 \in D$. The other two cases are inconsistent with D's being reduced. Therefore the only possibility consistent with D's being reduced is that $D = \{\mathbf{L}_1\}$. Similarly, if $D = \{\mathbf{L}_1\}$, $C = \{\mathbf{L}_1\}$. Therefore when either one of C and D is $\{\mathbf{L}_1\}$ so is the other which proves part 2 in these cases.

Suppose, then, that there are σ, μ such that $C\sigma \subseteq D \cup \overline{Th}$ and $D\mu \subseteq C \cup \overline{Th}$, neither C nor D are tautologies and $C \wedge Th = \emptyset$ and $D \wedge Th = \emptyset$. Then $C\sigma\mu \subseteq \overline{Th}\mu \cup D\mu \subseteq \overline{Th} \cup C$. Let $C_1 = \{L \in C \mid L\sigma\mu \in \overline{Th}\}$ and $C_2 = C \setminus C_1$. Since C is reduced, $C_1 = \emptyset$ and $C_2 = C$. Therefore as $\{\mathbf{L} \in C \mid \mathbf{L}\sigma \in \overline{Th}\} \subseteq C_1$, $C\sigma \subseteq D$. Similarly $D\mu \subseteq C$. Under the conditions that neither C nor D are tautologies and $C \wedge Th = D \wedge Th = \emptyset$. C is reduced relative to Th implies C is reduced and similarly for D. Therefore, by theorem 3.3.1.1, C and D are alphabetic variants.

3 If D is a tautology or if $D \wedge Th \neq \emptyset$ then $C \leq D$ (Th) and so C is a l.g.g. of C and D relative to Th.

When C is a tautology or if $C \wedge Th \neq \emptyset$, the proof is similar.

In the last case, $\inf\{C \cup \overline{Th}, D \cup \overline{Th}\}$ is certainly a lower bound of C and D, relative to Th. Suppose that $\mathbf{E} \leq C$ (Th) and $\mathbf{E} \leq D$ (Th). If $\mathbf{E} \leq C \cup \overline{Th}$ and $\mathbf{E} \leq D \cup \overline{Th}$ then $\mathbf{E} \leq \inf\{C \cup \overline{Th}, D \cup \overline{Th}\}$. As this is the only possible case, the proof is finished.

It would indeed be desirable to extend these results to a Th with general literals. However great difficulties arise. There is no trouble as regards l.g.g's of literals. Theorem 1 is true as it stands and the proof hardly needs any alteration. One can obtain analogues to the first two parts of theorem 2. The trouble comes with part 3. No l.g.g. relative to Th can exist. Here is a counterexample.

Let $Th = \{ \forall xx' \bar{P}(g(x),x'), \forall xx' \bar{P}(f(x),x') \}$. There is no l.g.g. of $Q(f(a()))$ and $Q(g(a()))$ relative to Th.

Let $g_1,\ldots,g_i,\ldots$ be an infinite sequence of distinct unary function symbols each of which is also distinct from f and g.

Let $C_n = \{Q(x)\} \cup \{P(x,g_i(x)) \mid i=1,n\}$.

Now $C_n \leq \{Q(f(a()))\}$ (Th). Figure 1 gives a C_i-derivation of $\{Q(f(x))\}$. (See chapter 3, section 1.3).

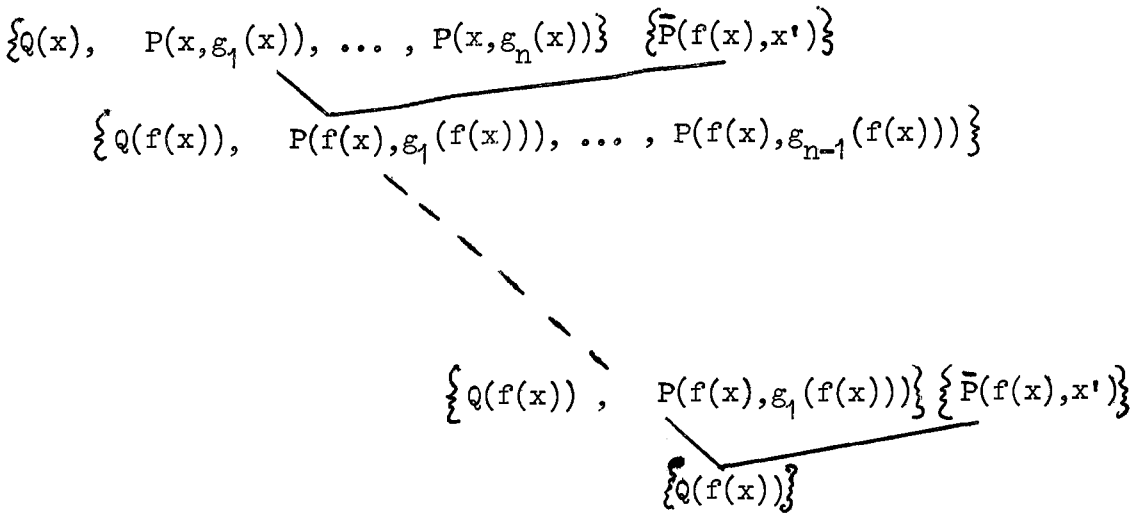


Figure 1

Similarly, $C_n \leq \{Q(g(a()))\}$ (Th). Suppose that D is a l.g.g. of $\{Q(f(a()))\}$ and $\{Q(g(a()))\}$. D must contain a variant of $\{Q(x)\}$ and can contain no literal of the form Q(t) where t has an occurrence of a function symbol. Since $C_n \leq D$ (Th) there must be a C_n-derivation of a clause subsuming D from C_n. This clause must contain a literal with predicate symbol Q. This literal may not, therefore, contain any function symbol. So the occurrence of x in C_n may only be replaced by an occurrence of some other variable. Therefore no $\{P(x,g_i(x))\}$ may be resolved with a literal from Th.

Therefore D contains an occurrence of every g_i in C_n. As this argument is independent of n, we have arrived at a contradiction. Therefore $\{Q(f(a()))\}$ and $\{Q(g(a()))\}$ can have no l.g.g. relative to Th.

We might hope that there is, say, an infinite set of non-equivalent, relative to Th, generalisations, relative to Th, of C and D with the property that if **E** is a generalisation, relative to Th, of C and D then it is a generalisation, relative to Th, of some member of the set. This would still allow some kind of computational procedure. However, one can show that if $\mathbf{E} \leq C$ (Th) and $\mathbf{E} \leq D$ (Th) there is an **E'**, also a generalisation of C and D, relative to Th, such that $\mathbf{E} \leq \mathbf{E}'$ (Th), but $\mathbf{E}' \not\leq \mathbf{E}$ (Th). Thus there are not even any minimally general generalisations of C and D relative to Th. The technique is to choose a C_n such that g_n does not occur in **E**, and to let $\mathbf{E}' = \mathbf{E}\xi \cup C_n$ where ξ standardises **E** apart from C_n. It is easy to show that $\mathbf{E}' \leq C$ (Th) and $\mathbf{E}' \leq D$ (Th). Evidently $\mathbf{E} \leq \mathbf{E}'$. If $\mathbf{E}' \leq \mathbf{E}$ (Th), one finds by

arguments, similar to those showing that C and D can have no l.g.g. relative to Th, that g_n must occur in E which is a contradiction. One can obtain similar counterexamples by replacing $g_i(x)$ by $h(....h(x))$ where there are i occurrences of h, which avoids the use of infinitely many function symbols.

In general, therefore, there is no solution when Th contains general literals and $\mathcal{S}$ is $\mathcal{S}$ cpg, since relative least general generalisations do not always exist. It is in fact not difficult to produce an example, using the above techniques, where, although there is a consistent explanation, there is no best one.

One can do spectacularly better however, when $\mathcal{S}$ is $\mathcal{S}_s$. (H_1 $\mathcal{S}_s$ H_2 iff H_1 has no more symbol occurrences than H_2).

Suppose, for the moment, that there are only finitely many function and predicate symbols.

Let $x_1,....$ be an infinite list of variables. One can find a list $H_1,H_2,....$ of sets of clauses such that:

1) If H_i has m variables, they are $x_1,....,x_m$.

2) Given any H there is an H_i which is an alphabetic variant of H.

3) If $i \leq j$ then H_i $\mathcal{S}_s$ H_j.

Let Th and Irr be arbitrary, take $\mathcal{S}$ to be $\mathcal{S}_s$, and choose some e and f. Then the first H_i in the list, which consistently explains

f given e and Th, is a solution to the resulting generalisation problem. There always is one, since H_0 is a consistent explanation of f given e and Th. Let this first explanation be H_{i_1}. As there are finitely many sets of clauses in the list with a given number of symbols, there is a finite set $\{H_i \mid H_i \mathrel{\mathcal{G}}_s H_{i_1}$ and H_i is a consistent explanation of f given e and Th$\}$.

Every solution is a variant of some member of this set, and every member of the set is a solution. Whether or not the set is effectively obtainable depends, as usual, on the decidability of consistency.

It is possible in principle, therefore, to accept the non-existence of least generalisations, if one alters $\mathcal{G}$. One would wish however to use a $\mathcal{G}$ for which the solutions are easily obtainable. Certainly $\mathcal{G}_s$, is of no use if the algorithm which searches an infinite list has to be employed.

3. A general algorithm using a limited consistency check

In chapter 4 the unsolvability results were caused by the difficulty in checking for consistency. Meltzer (1970) has suggested that the requirements of consistency be replaced by the requirement that a determined effort has been made, but failed, to prove that $\forall H \wedge Th \wedge Irr \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)$ is inconsistent.

We could still add on at the end a (perhaps computationally very expensive) test for consistency. If H passed this test we would have a solution. If it did not, we could recycle, making a more determined effort to check consistency. This combination might be quite practical.

For our problem we decided, following Meltzer, to formalize the 'determined' test as $\forall H \wedge Th \wedge \bigwedge_i (e_i \wedge f_i) \wedge Irr$ is 1-consistent, where 1-consistent means that a binary resolution theorem prover has not found a contradiction at level 1. The theory goes through much as before, and theorem 4.1 holds when the evidence changes have been made.

This theory does not parallel the procedure of Meltzer exactly. There are great differences in the way generalisations are found. He abstracts individual members of H_0 rather than combining them to form least generalisations. This is partly motivated by his Popperian niceness relation which prefers more to less general sentences.

We hand-simulated the method, with 1 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, $f_1 = \mathbf{E}(f(a,b),f(c,d))$ and $f_2 = \mathbf{E}(f(c,d),f(a,b))$ (similarly for $ab \neq cd$). The corresponding e_i are empty. Th was empty, but one took Irr to be the axioms for equality. It can be shown that $\mathbf{E}(t_1,t_2)$ is in the solution H iff $\mathbf{E}(t_2,t_1)$ is (similarly for $\overline{\mathbf{E}}(t_1,t_2)$) 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 was replaced by $(xy)z = x(yz)$, the associative law. This is as good a result as Meltzer obtained. Both methods obtained the right-identity and the commutative laws. Meltzer obtained a part of the associative law, viz.:

$$(xx)y = w \supset x(xy).$$

If he altered his method so that occurrences of terms were abstracted, rather than abstracting on every occurrence at once, he would have obtained $(xy)z = w \supset x(yz) = w$ which is equivalent to the associative law in the presence of the axioms of equality.

Rather different laws, true only for the example groups, were found by the two methods.

What neither method does, however, is to use the equality axioms when forming generalisations. Preliminary investigation of generalisation relative to these axioms shows that this is not an easy problem.

4. Some pilot experiments

4.1 Description of the program

To test out our ideas on algorithms for forming generalisations, we have programmed a method which works on a case of the general problem, generated by the following assumptions:-

1) The language is sorted and there are no function symbols other than constants.

2) The niceness relation, $\succ$, is $\succ_{cpg}$.

3) The knowledge used for generalisation, Th, is empty.

4) $Irr \wedge \bigwedge_i (e_i \wedge f_i)$ is the conjunction of the literals in some Herbrand base of $Irr \wedge \bigwedge_i (e_i \wedge f_i)$.

5) Only one predicate symbol occurs in f.

From the discussion of the simple solvable case after corollary 2 in chapter 4 we see that consistency is easily checked. Indeed, if $\mathbf{E} = \{\bar{\mathbf{L}} \mid \mathbf{L}$ is a conjunct of $Irr \wedge \bigwedge_i (e_i \wedge f_i)\}$ then $\forall H \wedge Irr \wedge \bigwedge_{i=1}^{n} (e_i \wedge f_i)$ is consistent iff $H \not\subseteq \mathbf{E}$.

The program calculates a heuristic approximation to an irredundant explanation, H, rather than looking for a best one. This is to save time. The program is written in the POP-2 programming language (Burstall, Collins and Popplestone, 1971) and has been run on

the ICL 4130 machine. At the moment running times vary between three and fifteen minutes.

The program starts with H, the potentially irredundant explanation, set equal to H_0. It then continuously chooses a member, D, of H and a member C_i of H_0 and replaces D by $\inf^*\{D,C_i\}$ a heuristic approximation to the reduced form of $\inf\{D,C_i\}$. This stops when any such replacement results in $\forall H \wedge \text{Irr} \wedge \bigwedge_i (e_i \wedge f_i)$ being inconsistent. Then H is output as the result.

The flowchart of the program is given in figure 1.

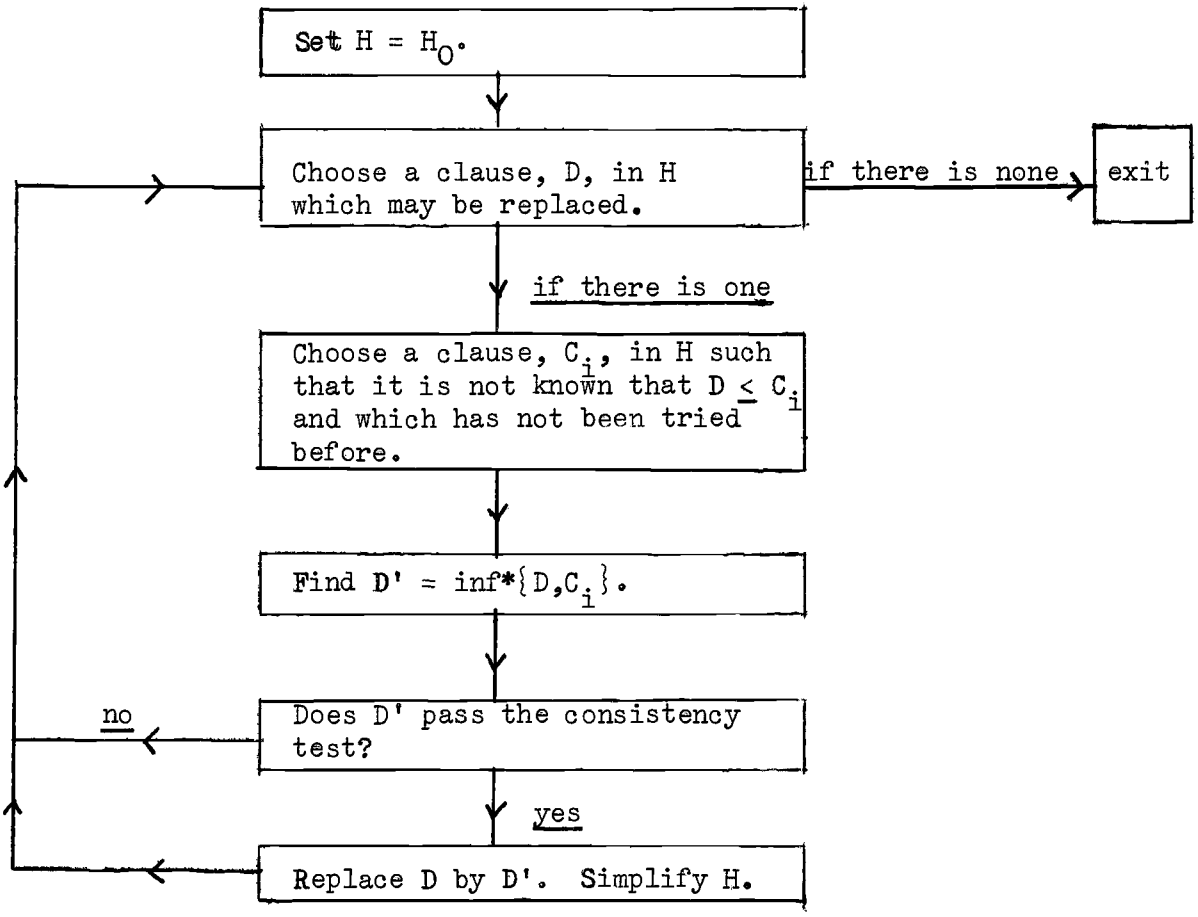


Figure 1

Various heuristic rules are used in making the choices mentioned in the flow chart.

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

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

Initially, Fail(D) = $\emptyset$ and Success(D) = $\{D\}$, for any D in H. The clause, D, can only be selected if Fail(D) $\cup$ Success(D) $\neq H_0$. The program chooses a clause D in H with a largest Success(D) and of two such clauses prefers the one with the smaller failure set.

A clause C_i in H_0 can only be chosen if it is not in Failure(D) $\cup$ Success(D). From these the program chooses a clause C_i for which there are minimally many clauses D in H such that C_i is in Success(D). This reflects our belief that the better-structured the problem the less likely it is that any clause in H_0 is generalised by more than one clause in a good explanation. Consequently we believe that the chance of failing, and so wasting a lot of computational effort, grows with the number of previous successes in explaining f_i given e_i.

Next, D" = inf$\{D,C_i\}$ is calculated. An approximation to the reduced form of D" is found as follows. If $D \leq C_i$ then D" is D. Otherwise, D" is ordered into a list $L_1,\ldots,L_n,\ldots,L_m$ where any two distinct literals occurring in $L_1,\ldots,L_n$ have different predicate

symbol and sign pairs. Further if some predicate symbol and sign pair occurs in D^* it occurs in $L_1,\ldots,L_n$. The literal L_i has no more variables than any other literal with that predicate symbol and sign for i=1,n. For $n < i \leq m$, L_i has less variables (not in $L_1,\ldots,L_{i-1}$) than in any other of the $L_i,\ldots,L_m$. The heuristic approximation to the reduced form of D^* is the reduced version, calculated properly, of $\{L_i \mid 1 \leq i \leq \min(l,m)\}$, where l is a program parameter, which is set as large as possible without inconveniently long running times. We set l to be 11 throughout our experiments.

The consistency test checks whether $D^* \leq E$, with E as defined above. This is a necessary and sufficient condition for the consistency of H with $Irr \wedge \bigwedge_i (e_i \wedge f_i)$ since the other clauses in H will either already have been checked or else are in H_0.

Simplification is performed by continuous application of the following operation, until this is no longer possible:

Remove from H a clause, D_1 such that

$$Success(D_1) \subseteq \bigcup_{D_2 \in H \text{ and } D_2 \neq D_1} Success(D_2).$$

Notice that D^* itself cannot be removed by this process. The program takes advantage of this fact.

When the program terminates, and it must do so, H is reduced. For suppose D_1 and D_2 are distinct clauses in H then and $D_1 \leq D_2$. Suppose that D is any clause in H then. Now since the program has terminated,

H_0 = Success(D) $\cup$ Fail(D). If $C \in$ Success(D) then $D \leq C$. Otherwise, as $\forall H \wedge$ Irr $\wedge \bigwedge(e_i \wedge f_i)$ is consistent, $D \not\leq C$. Therefore Success(D) = $\{C \in H_0 \mid D \leq C\}$. It follows that Success(D_2) = $\{C \in H_0 \mid D_2 \leq C\} \subseteq \{C \in H_0 \mid D_1 \leq C\}$ = Success(D_1). But then the simplification process would have removed D_2. This contradicts the fact that the program has terminated and establishes the conclusion.

On the other hand, H need not be irredundant at termination since inf* is not inf. If inf* were inf, it would be.

4.2 Evaluation of experimentation

It is now convenient to discuss why experimentation is helpful. First it serves to give more complex examples than can easily be produced by hand. Secondly, we can try to evaluate our hypothesis generation method. Some ways of evaluation will not be considered. We pay little attention to the efficiency of the program, since it has only been written to provide answers in a reasonable amount of time which varied, as stated above, between three and fifteen minutes for the examples described below. Neither will the behaviour of the hypothesis method through time be considered; we have only begun to investigate the possibilities theoretically (see chapter 6). Finally we do not investigate either theoretically or practically how useful the hypothesis generation method is to the organism employing it (see chapter 1).

The method will be judged by the hypotheses it produces. But

this is not too easy; by definition the method must produce the nicest explanatory hypothesis possible, in the sense of $\mathcal{H}_{cpg}$. Consequently experiment will merely confirm theorem 4.1. However we can test the correlation of $\mathcal{H}_{cpg}$ with other niceness relations. For illustration we will try $\mathcal{H}_1$, (see chapter 1 for a definition) and $\leq$ (this is a Popperian niceness relation, as discussed in section 1 of this chapter).

It would certainly be possible to try to calculate the predictive power of a generated hypothesis, when all cases and the correct hypothesis are known. However one should also take into account how good the information provided to the hypothesis generation machine is. Suppose, at one extreme, that no information is given (where $f = \emptyset$). Then one cannot expect any predictive power of a generated hypothesis. At the other extreme when all possible cases are given one would expect a hypothesis to possess total predictive power, as a simple consequence of its being an explanation.

More generally, one can only expect the hypothesis generated to do well in cases similar to those it is given information about. This would require what we do not possess: a method of giving a sense to the word "similar" induced by the correct hypothesis.

A partial way around this problem would be to compare with respect to predictive power one hypothesis generated by the generation method with another either generated by a different method or else by humans,

when both hypotheses are generated from the same information. This would require rather more than a pilot experiment.

What we shall do is award good marks according to how well the predictions of the generated hypothesis match those of the correct one. If the agreement is, intuitively, rather small and we can show, intuitively, that a "fair sample" of cases have been supplied, then we shall award a bad mark.

One should compare our difficulties with those which arise when evaluating Evans' Analogy program (Evans, 1968). As long as his program gives the "correct" answers, it is certainly doing well. If it gives no answer, it is certainly doing badly. But suppose it gives the "wrong" answer. Then whether it is doing well or not seems to depend on what its reasons for giving that answer were. Two comparisons are in order. First consider a program which simply always selects the first allowable answer figure. It will certainly never have any reason for making its choice. Next consider a clever person who knows the correct answer and deliberately tries to find good reasons for choosing some wrong answer. It is well known that most I.Q. tests can be so treated by an intelligent person! Perhaps therefore an analogy program should try to find as large a number of answers as possible and try to convince us that each was "correct".

4.3 An experiment using the win predicate of noughts and crosses

The aim of the experiment is to discover a sufficient set of

conditions for a position to be a win. The board is considered to be a three by three matrix. We use a language with two sorts: numbers and positions. There are three predicate symbols, XX, OO and Win. XX(i,j,p) is true iff position p has an X in square (i,j). The predicate OO is defined similarly. We can formulate a set $H_{correct} = \{E_1, E_2, E_3, E_4\}$, of four clauses expressing sufficient conditions for a win:

$E_1 = \{\overline{XX}(i,1,p), \overline{XX}(i,2,p), \overline{XX}(i,3,p), Win(p)\}$,

$E_2 = \{\overline{XX}(1,i,p), \overline{XX}(2,i,p), \overline{XX}(3,i,p), Win(p)\}$,

$E_3 = \{\overline{XX}(1,1,p), \overline{XX}(2,2,p), \overline{XX}(3,3,p), Win(p)\}$,

$E_4 = \{\overline{XX}(1,3,p), \overline{XX}(2,2,p), \overline{XX}(3,1,p), Win(p)\}$.

These state, respectively, that a row, a column, a forward diagonal or a backward diagonal of X's is a sufficient condition for a win.

Consider the position, $p_1()$ say, displayed in figure 1.

	X	O
O	X	X
	X	O

Figure 1

Certainly the fact, f_1, to be explained is

$f_1 = Win(p_1())$.

There are two possibilities for $Ev(f_1)$. If we give the exact reasons why f_1 is a win then

$$Ev(f_1) = XX(1,2,p_1()) \wedge XX(2,2,p_1()) \wedge XX(3,2,p_1()).$$

If we decide that the win depends on where the marks X and O are then

$$\begin{aligned} Ev(f_1) = {} & XX(1,2,p_1()) \wedge XX(2,2,p_1()) \\ & \wedge XX(3,2,p_1()) \wedge OO(1,3,p_1()) \\ & \wedge OO(2,1,p_1()) \wedge OO(3,3,p_1()). \end{aligned}$$

We shall try both. There are certainly others in which, for instance we might include in $Ev(f_1)$ the literal $\overline{OO}(1,1,p_1())$ or even $\overline{\overline{OO}}(1,2,p_1())$.

The choice of Ev is not prescribed by our theory and is but one of many omissions (see chapter 1).

Next, consider a loss, such as the position, $p_2()$, displayed in figure 2.

	X	X
O	O	O
X		X

Figure 2

In this case we simply include a complete description of the

figure in Irr. So Irr will contain the conjunction:

$$\bigwedge_{j=1}^{3} \overline{OO}(1,j,p_2()) \wedge \bigwedge_{j=1}^{3} \overline{OO}(3,j,p_2())$$

$$\wedge \bigwedge_{j=1}^{3} \overline{XX}(2,j,p_2()) \wedge \overline{XX}(3,2,p_2())$$

$$\wedge \bigwedge_{j=1}^{3} OO(2,j,p_2()) \wedge \bigwedge_{j=2}^{3} XX(1,j,p_2())$$

$$\wedge XX(3,1,p_2()) \wedge XX(3,3,p_2()) \wedge \overline{Win}(p_2())$$

So given a set of won or lost positions, we have shown how f, Ev and Irr are obtained. However the resulting problem will not satisfy assumption 3, given in the description of the program, since we have not included a complete description of every won position in Irr.

Let $\mathbf{E} = \{\bar{L} \mid L \text{ is a conjunct of Irr} \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)\}$ as defined there; let Irr' be Irr together with a complete description of every win and let $\mathbf{E}' = \{\bar{L} \mid L \text{ is a conjunct of Irr}' \wedge \bigwedge_{i=1}^{n}(e_i \wedge f_i)\}$. Now if only a single position variable occurs in a clause C which contains only positive occurrences of the Win predicate, then the reader should verify that $C \leq \mathbf{E}$ iff $C \leq \mathbf{E}'$. Now every clause in $\mathcal{G}_{\mathcal{B}}(H_0)$ has this character and so we conclude that we may safely use the program with Irr as defined. This represents a useful computational saving.

The first result is in a case where Ev gave the exact reasons and the positive wins were as displayed in table 1. We took all possible

non-wins to form Irr. The calculation was performed by hand, since it is easy in this case to generate all solutions, rather than just one heuristic approximation to an irredundant set of clauses.

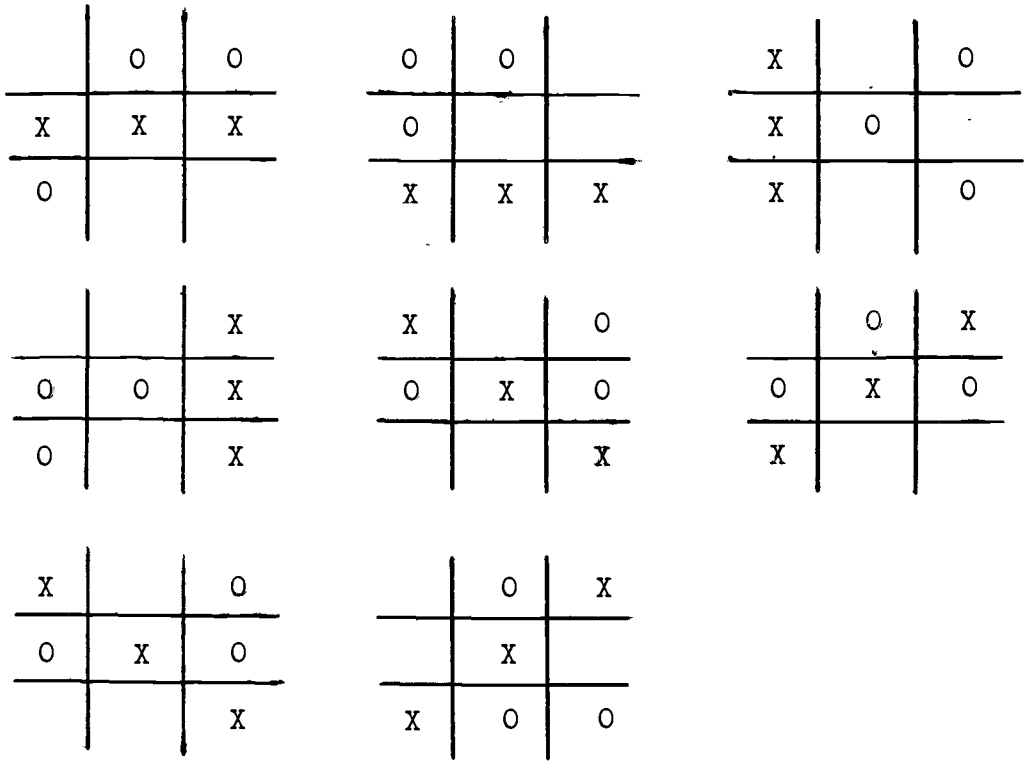


Table 1

There is exactly one solution, $H_{soln} = \{D_1, D_2, D_3, D_4\}$ where

$D_1 = \{\overline{XX}(i,1,p), \overline{XX}(i,2,p), \overline{XX}(i,3,p),$
$\overline{XX}(i,i,p), Win(p)\},$

$D_2 = \{\overline{XX}(1,i,p), \overline{XX}(2,i,p), \overline{XX}(3,i,p),$
$\overline{XX}(i,i,p), Win(p)\},$

$D_3 = \{\overline{XX}(1,1,p), \overline{XX}(2,2,p), \overline{XX}(3,3,p),$
$Win(p)\},$

$D_4 = \{\overline{XX}(1,3,p),\ \overline{XX}(2,2,p),\ \overline{XX}(3,1,p),$
$\quad Win(p)\}.$

Evidently a position is a win according to H_{soln} iff it is according to $H_{correct}$. So H_{soln} has good predictive power. It is not maximally nice with respect to either $\prec_1$, or $\leq$ since the literal $\overline{XX}(i,i,p)$ occurring in the clauses D_1 and D_2 is superfluous.

The next two examples use an Ev of the second sort, where we record all occurrences of O's and X's in each $Ev(f_i)$ $(i=1,n)$. For the first example, the win and non-win positions are displayed in tables 2 and 3 respectively. They are from an example given in Popplestone (1970).

		X
	X	
X		

X		
	X	
		X

O	X	O
	X	
	X	O

X		
X		
X		

X	X	X
		O

X	X	X

Table 2

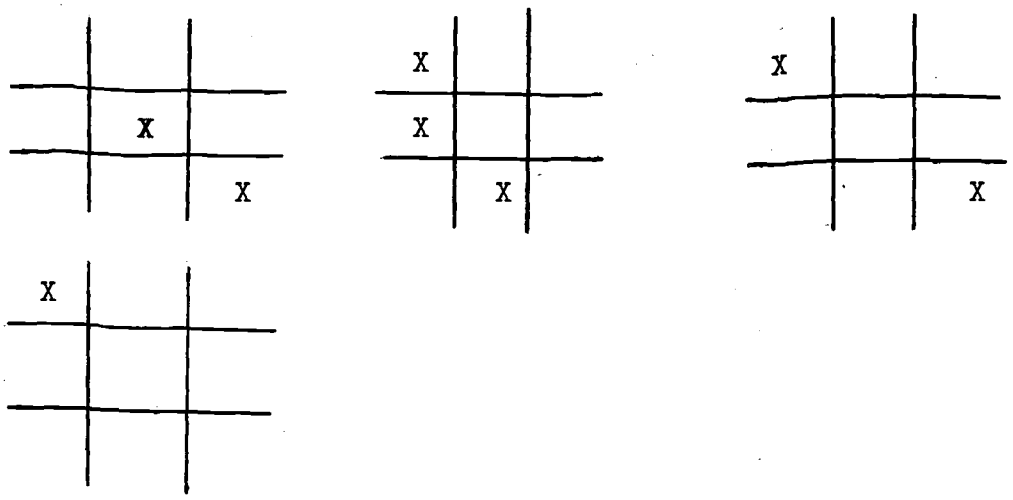


Table 3

The program found a set of clauses $H_{soln} = \{D_1, D_2\}$, where

$D_1 = \{\overline{XX}(i,3,p), \overline{XX}(m,i,p), \overline{XX}(m,1,p),$
$\overline{XX}(i,m,p), Win(p)\}$

and $D_2 = \{\overline{XX}(3,d,p), \overline{XX}(d,d,p), \overline{XX}(2,g,p),$
$\overline{XX}(g,g,p), \overline{XX}(i,i,p), \overline{XX}(1,i,p), Win(p)\}$.

The clause D_1 has as a consequence that any position containing a column of X's or a backward diagonal of X's is a win. The clause D_2 does the same for rows and the forward diagonal. In fact $H_{soln} \leq H_{correct}$. Therefore if $H_{correct}$ predicts that a position is a win, so will H_{soln}. However H_{soln} makes some wrong predictions. For example, according to D_1 any position containing an X in (1,3) and in (3,1) is a win.

We cannot say if this is because we do not have examples of all the "ways" in which a position can fail to be a win since we do not have a good concept of such a "way". H_{soln} is not maximally nice with respect to either $\rightarrow_1$, or $\leq$ since one can remove $\overline{XX}(i,m,p)$ from D_1 without

affecting consistency with $\text{Irr} \wedge \bigwedge_i (e_i \wedge f_i)$.

The win and non-win positions for the second example are displayed in tables 4 and 5 respectively.

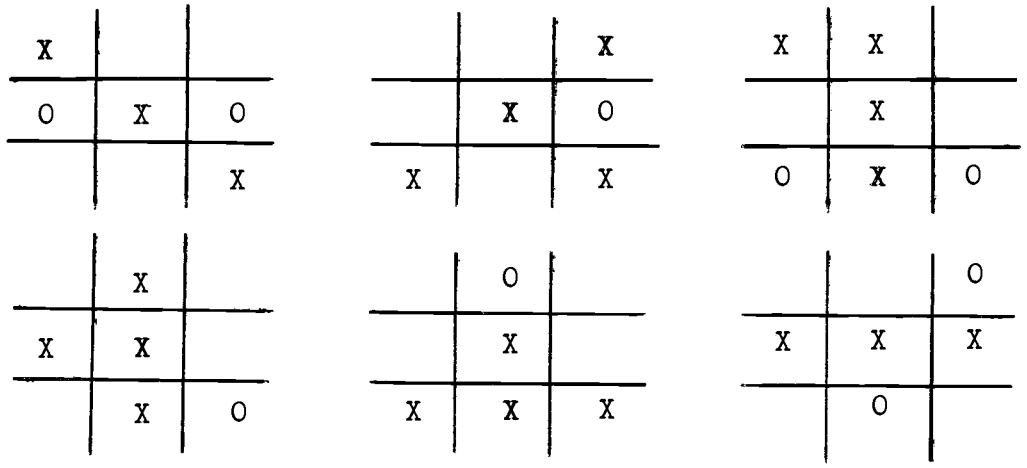


Table 4

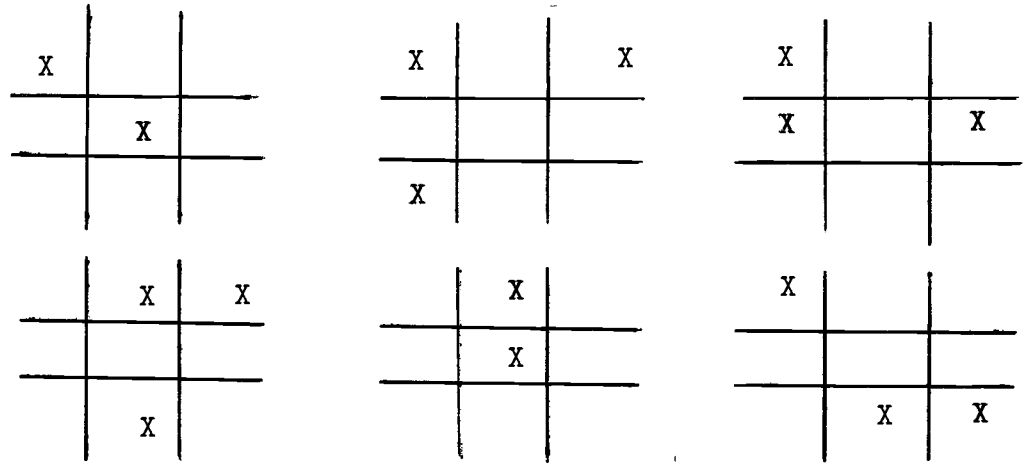


Table 5

The program found a set of clauses, $H_{soln} = \{D_1\}$ where

$D_1 = \{\overline{XX}(2,2,p),\ \overline{XX}(k,k,p),\ \overline{XX}(n,k,p),\ \overline{OO}(q,n,p),\ \text{Win}(p)\}$.

As we inadvertently did not put a 0 in any of the non-wins, it is possible to find an explanation with only one clause. This invalidates any comparison with H_{soln}. The solution is not maximally nice with respect to either $\mathcal{J}_1$, or $\leq$ since the clause

$$\{\overline{OO}(q,n,p),\ Win(p)\}$$

would do just as well.

To sum up, generated hypotheses compare well with H_{soln}, but do not have maximal niceness with respect to either $\mathcal{J}_1$, or $\leq$.

4.4 Learning the patrilineal ancestor relationship

The binary relation, patrilineal ancestor, is recursively defined by

$$Anc(x,y) \equiv Father(x,y) \vee \exists z(Father(x,z) \wedge Anc(z,y)).$$

We are using a language with one sort. There are three binary predicate symbols, Father, Daughter and Anc with evident meanings. The hypothesis generation machine is required to find sufficient conditions for one individual to be the patrilineal ancestor of another. The definition of Anc gives the set, $H_{cor} = \{E_1, E_2\}$, of sufficient conditions where

$$E_1 = \{\overline{Father}(x,y),\ Anc(x,y)\}$$

and $$E_2 = \{\overline{Father}(x,z),\ \overline{Anc}(z,y),\ Anc(x,y)\}.$$

A typical member of f has the form Anc(a,b). There are, again, at least two ways of choosing Ev(anc(a,b)). One is to give exact reasons. Another is to choose a reasonably small set of literals which establish a link between a and b. A link is a set of literals $\{L_k \mid k=1,l\}$ where each L_k is of the form $P_k(a_k,b_k)$ or $P_k(b_k,a_k)$ where $a=a_1$, $b_k=a_{k+1}$ $(1\leq k\leq l-1)$ and $b_l=b$. In general we let Ev(Anc(a,b)) be a few of the smallest links between a and b.

Irr is simply a conjunction of all the true literals in the example under consideration.

In the examples, we were concerned with two families whose trees are displayed in figures 1 and 2.

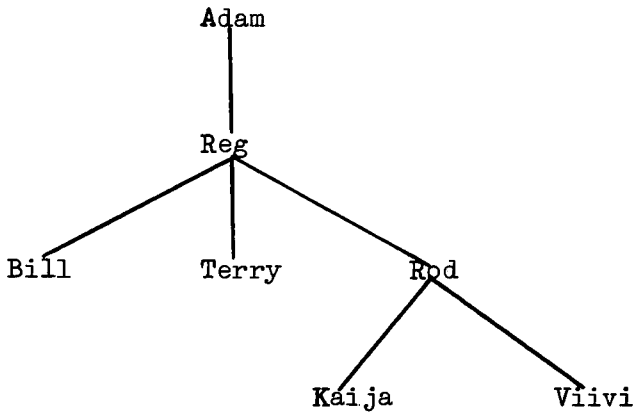


Figure 1

Figure 2

In order to define the Irr used in the various examples, without repetition, we define three sets of clauses Irr_1, Irr_2 and Irr_3, using some auxiliary sets.

Let $Irr_{1,A} = \{\{Anc(a,b)\} \mid a \in \{Adam,Reg\}, b \in \{Bill,Terry,Rod,Kaija,Viivi\}\}$
$\cup \{\{Anc(Adam,Reg)\}\}$
$\cup \{\{Anc(Rod,b)\} \mid b \in \{Kaija,Viivi\}\}$,

$Irr_{2,A} = \{\{Anc(Isa,Manuel)\}, \{Anc(Isa,Karen)\}, \{Anc(Manuel,Karen)\}\}$,

$Irr_{1,F} = \{\{Father(Reg,b)\} \mid b \in \{Bill,Terry,Rod\}\}$
$\cup \{\{Father(Rod,Kaija)\}, \{Father(Rod,Viivi)\}\}$,

$Irr_{2,F} = \{\{Father(Isa,Manuel)\}, \{Father(Manuel,Karen)\}\}$,

$Irr_{1,D} = \{\{Daughter(Kaija,Rod)\}, \{Daughter(Viivi,Rod)\}\}$,

$Irr_{2,D} = \{\{Daughter(Karen,Manuel)\}\}$,

$$Irr_1 = Irr_{1,A} \cup Irr_{1,F} \cup Irr_{1,D} \cup \{\{\bar{P}(a,b)\} \mid P \in \{Anc, Father, Daughter\},\ a, b \in \{Adam, Reg, Bill, Terry, Rod, Kaija, Viivi\},\ P(a,b) \notin Irr_{1,A} \cup Irr_{1,F} \cup Irr_{1,D}\},$$

$$Irr_2 = Irr_{2,A} \cup Irr_{2,F} \cup Irr_{2,D} \cup \{\{\bar{P}(a,b)\} \mid P \in \{Anc, Father, Daughter\},\ a,b \in \{Isa, Manuel, Karen\},\ P(a,b) \notin Irr_{2,A} \cup Irr_{2,F} \cup Irr_{2,D}\}$$

$$Irr_3 = \{\bar{P}(a,b) \mid P \in \{Anc, Father, Daughter\},\ a \in \{Adam, Reg, Bill, Terry, Rod, Kaija, Viivi\},\ b \in \{Isa, Manuel, Karen\}\}$$

$$\cup \{\bar{P}(b,a) \mid P \in \{Anc, Father, Daughter\},\ a \in \{Isa, Manuel, Karen\},\ b \in \{Adam, Reg, Bill, Terry, Rod, Kaija, Viivi\}\}.$$

In the first example, we used an Ev of the first kind. Ev and f are described in table 1; Irr was taken to be $Irr_1 \cup Irr_2 \cup Irr_3$.

The program output the set of clauses H_{cor}, described above. It is possible to prove that these are optimal with respect to both $\prec_1$, and $\leq$.

f	e
f_1 = Anc(Rod,Kaija)	e_1 = Father(Rod,Kaija)
f_2 = Anc(Reg,Terry)	e_2 = Father(Reg,Terry)
f_3 = Anc(Reg,Kaija)	e_3 = Anc(Reg,Rod) $\wedge$ Father(Rod,Kaija)
f_4 = Anc(Reg,Viivi)	e_4 = Anc(Reg,Rod) $\wedge$ Father(Rod,Viivi)
f_5 = Anc(Isa,Karen)	e_5 = Anc(Isa,Manuel) $\wedge$ Father(Manuel,Karen)

Table 1

In the second example, we used an Ev of the second kind. Ev and f are described in table 2; Irr was taken to be Irr_1.

f	e
f_1 = Anc(Rod,Kaija)	e_1 = Daughter(Kaija,Rod) ∧ Father(Rod,Kaija).
f_2 = Anc(Reg,Terry)	e_2 = Father(Reg,Terry).
f_3 = Anc(Reg,Kaija)	e_3 = Father(Reg,Rod) ∧ Daughter(Kaija,Rod)∧ Father(Rod,Kaija)∧ Anc(Rod,Kaija) ∧ Anc(Reg,Rod).
f_4 = Anc(Adam,Kaija)	e_4 = Anc(Adam,Reg) ∧ Anc(Reg,Kaija)∧ Anc(Rod,Kaija)∧ Father(Reg,Rod) ∧ Father(Rod,Kaija).

Table 2

The program output a set of clauses, $H = \{D_1, D_2\}$ where

$D_1 = \{\overline{\text{Father}}(x,y),\ \text{Anc}(x,y)\}$,

$D_2 = \{\overline{\text{Anc}}(x,y),\ \overline{\text{Father}}(x,y),\ \text{Anc}(\text{Reg},y),$
$\overline{\text{Father}}(\text{Reg},\text{Rod}),\ \overline{\text{Father}}(\text{Rod},\text{Kaija}),\ \overline{\text{Anc}}(\text{Rod},\text{Kaija}),$
$\overline{\text{Anc}}(z,r),\ \overline{\text{Anc}}(r,\text{Kaija}),\ \text{Anc}(z,\text{Kaija})\}$.

Now H does not have the same explanatory power as H_{cor}. For example $H \not\models \{\text{Anc}(\text{Reg},\text{Viivi}) \leftarrow \text{Anc}(\text{Reg},\text{Rod} \wedge \text{Anc}(\text{Rod},\text{Viivi}))\}$.

Further H is obviously not optimal according to either $\prec_1$, or $\leq$ since, for example, one could remove $\overline{\text{Father}}(x,y)$ from D_2 retaining consistency. It is not optimal with respect to $\leq$ for another important reason: one could replace every occurrence of Kaija in D_2 by one of the variable w.

It is interesting to notice that D_2 is equivalent, relative to Irr, to the clause,

$$D_3 = \{\overline{\text{Anc}}(x,y),\ \overline{\text{Anc}}(y,\text{Kaija}),\ \text{Anc}(x,\text{Kaija})\}$$

This suggests that it would be interesting to extend the program so that it handles the simple kind of Th discussed in section 2.2 of this chapter.

The next example uses a little less biased evidence. Ev and f are described in table 3; Irr was taken to be $\text{Irr}_1 \cup \text{Irr}_2 \cup \text{Irr}_3$.

f	e
f_1 = Anc(Rod,Kaija)	e_1 = Daughter(Kaija,Rod) ∧Father(Rod,Kaija).
f_2 = Anc(Reg,Terry)	e_2 = Father(Reg,Terry).
f_3 = Anc(Reg,Kaija)	e_3 = Father(Reg,Rod) ∧Daughter(Kaija,Rod) ∧Father(Rod,Kaija) ∧Anc(Rod,Kaija) ∧ Anc(Reg,Rod).
f_4 = Anc(Reg,Viivi)	e_4 = Father(Reg,Rod) ∧ Daughter(Viivi,Rod) ∧ Father(Rod,Viivi) ∧ Anc(Rod,Viivi) ∧Anc(Reg,Rod).
f_5 = Anc(Isa,Karen)	e_5 = Anc(Isa,Manuel) ∧Anc(Manuel,Karen)∧ Father(Isa,Manuel) ∧ Father(Manuel,Karen) ∧ Daughter(Karen,Manuel).
f_6 = Anc(Adam,Kaija)	e_6 = Anc(Adam,Reg) ∧ Anc(Reg,Kaija) ∧ Anc(Rod,Kaija) ∧ Father(Reg,Rod) ∧ Father(Rod,Kaija).

Table 3

The program output a set of clauses, $H = \{D_1, D_2\}$ where

$D_1 = \{\overline{\text{Father}}(x,y),\ \text{Anc}(x,y)\}$,

$D_2 = \{\overline{\text{Father}}(w,u),\ \overline{\text{Father}}(u,z),\ \overline{\text{Anc}}(u,z),$

$\overline{\text{Anc}}(x,y),\ \overline{\text{Anc}}(y,z),\ \text{Anc}(x,z)\}$.

In fact H and H_{cor} have the same predictive power: in any family tree they will both predict the patrilineal ancestors correctly given all the information about Father. More can be said. Suppose Th is a theory, expressing the tree-like structure of family trees, whose only predicate symbol is Father and look at the following definition obtained by changing the proposed sufficient condition into a necessary and sufficient one:

$$\begin{aligned}\mathrm{Anc}(x,z) \equiv\ & \mathrm{Father}(x,z) \vee \exists w,u,y(\mathrm{Father}(w,u) \\ & \wedge \mathrm{Father}(u,z) \wedge \mathrm{Anc}(u,z) \wedge \mathrm{Anc}(x,y) \\ & \wedge \mathrm{Anc}(y,z)).\end{aligned}$$

Then, assuming Th, this definition is equivalent to the original one.

However, one can easily see that H is not optimal with regard to either $\prec_1$, or $\leq$.

In conclusion, we see that although we obtain formulae with good predictive power, they are not, in general, optimal with regard to either $\prec_1$, or $\leq$. This accords with our experience in the case of O's and X's.

Chapter 6 Hypothesis learning theory

1. Introduction

Hypothesis learning theory was studied first, as a language learning theory, by E. Mark Gold (1967) and subsequently, in more mathematical detail by Jerome Feldman (1970). The importance of this theory is that it deals with the progression of a hypothesis discovery method through time. While it appears on the surface to deal exclusively with language guessing, the theory may be adapted for other purposes. Our presentation tries to bring this out by developing part of it on an abstract level. One can continue the abstract theory to cover the rest of Feldman's theory, but enough will be developed to give a good ground for application, criticism and indication of possible future trends.

All this work is essentially an elaboration of a simple but surprising theorem of Gold.

A complete information sequence for a context-free grammar, G, is an infinite list, I of the form $\pm y_1$ $\pm y_2$ .... where:

(1) +y appears in the list iff the string y is in the language determined by G.

(2) -y appears in the list iff the string y is not in the language determined by G.

Suppose that at time t one is presented with the signed string $\pm y_t$. To what extent can one succeed in "eventually" guessing G? It turns

out that there is a method, defined independently of G and I, which will identify G in the limit. That is, there is a time τ at which it will guess and ever thereafter continue to guess, a grammar G' whose language is the same as that of G.

Let $G_1, \ldots$ be an enumeration of all context-free grammars. The method chooses, at time t, the first grammar, G' say, such that if +y has appeared in the list by time t then y is in the language of G', and if -y has appeared in the list, by time t, then y is not in the language of G'. Evidently one can always find such a G', if only because G appears in the list. If G" is a grammar in the list whose language differs from that of G, then either there is a y in G but not in G", or vice versa. At some time, t', say $\pm$y (whichever is appropriate) will appear in the list and thereafter G" will not be chosen by our method. Hence after some time, τ say, every G occurring before the first grammar in the list whose language is the same as that of G, G' say, will never be chosen. At this time G' will be chosen and will continue to be chosen.

This theorem shows that, at least in this case, it is eventually possible to "learn" the truth. However it is never possible to "know" the truth, since any guess G can be forced to change by some $\pm$y or other.

2. Abstract theory

The theory is presented as an informal mathematical theory.

The domains are the hypothesis space, Hyp and the phenomenon space, Phen.

Axiom 1 Hyp is a recursive, infinite set of integers.

The variables ranging over Hyp are h, h' etc.

Examples 1) Hyp is the set of Gödel numbers of first-order lawlike universal sentences. Gödel numbers give a way of coding sentences as integers. This possibility of coding allows us to consider the theory a general one.

2) Hyp is the set of Gödel numbers of context-free grammars.

Hyp is required to be infinite in order to avoid trivial exceptional cases in later theorems. We will asume some fixed enumeration, $h_1, h_2, \ldots$ of Hyp.

Axiom 2 Phen is an infinite recursive set.

The variables ranging over Phen. are f, f_1 etc. We use, F, F^+, F^- etc. to range over finite subsets of Phen. $\mathcal{F}(\text{Phen})$ is the set of finite subsets of Phen. The fact that Phen is infinite forces attention on the harder problems. Generally, our theory becomes trivial when Phen is finite.

We will let $\mathcal{S} \subseteq$ Phen be a variable ranging over the recursive subsets of Phen. $\mathcal{S}$ is to be understood as a subject domain separated out from Phen, the class of all possible phenomena.

Examples 1) (Gödel numbers of) strings of letters.

2) (The Gödel numbers of) the set of all ground clauses, C which do not follow from Th, an arbitrary consistent set of sentences, but are consistent with it and Irr, another set of sentences consistent with Th.

The predicates are: Accountsfor is of sort Hyp × Phen,

M_A is of sort Hyp × Phen × Integers,

Consistent is of sort Hyp × $\mathcal{F}$(Phen) × $\mathcal{F}$(Phen)

M_C is of sort Hyp × $\mathcal{F}$(Phen) × $\mathcal{F}$(Phen) × Integers.

Axiom 3 Accountsfor is partial recursive, M_A is primitive recursive and Accountsfor(h,f) $\equiv \exists m\ M_A(h,f,m)$.

We extend Accountsfor to a partial recursive predicate of type Hyp × $\mathcal{F}$(Phen) by:

$$\text{Accountsfor}(h,F) \equiv_{\text{def}} \forall f \in F\ \text{Accountsfor}(h,f).$$

Similarly M_A is extended to a primitive recursive predicate of type Hyp × $\mathcal{F}$(Phen) × Integers by:

$$M_A(h,F,m) \equiv_{\text{def}} \forall f \in F\ \exists m' \leq m\ M_A(h,f,m').$$

Evidently, Accountsfor(h,F) $\equiv \exists m\ M_A(h,F,m)$.

Accountsfor is the type of implication being used to explain the phenomena.

Examples 1) Accountsfor(h,C) $\equiv \vdash_{Th} h \rightarrow C$.

2) Accountsfor(h,C) $\equiv h \leq \{C\}$ (Th).

3) Accountsfor(h,f) $\equiv$ f is a string in the language given by the grammar h.

M_A corresponds to a program for Accountsfor.

Axiom 4 Consistent is partial recursive, M_C is primitive recursive and $\neg$ Consistent$(h,F^+,F^-) \equiv \exists m\ M_C(h,F^+,F^-,m)$.

Consistent(h,F^+,F^-) means that h is consistent with the occurrence of the phenomena in F^+ (hence the plus sign) and with the non-occurrence of those in F^-.

If Accountsfor were a logical implication such that a completeness theorem held and one had sufficient logical symbolism, one could define Consistent in terms of Accountsfor by:

$$\text{Consistent}(h,F^+,F^-) \equiv \exists f \neg \text{Accountsfor}(h \wedge F^+ \wedge \neg F^-, f).$$

However we do not want such a strong implication in general, since it may be easier or more relevant to look for hypotheses bearing implications of a weaker sort, such as generalisation.

M_C corresponds to a program for $\neg$ Consistent.

Examples 1) When $F^+ = \{C_i \mid i=1,n\}$ and $F^- = \{D_j \mid j=1,m\}$, Consistent$(h,F^+,F^-) \equiv \forall(h \wedge \bigwedge_{i=1}^{n} C_i \wedge \bigwedge_{j=1}^{m} D_j) \wedge$ Th $\wedge$ Irr is consistent.

2) h is a grammar and F^+ and F^- are sets of strings.
Consistent$(h,F^+,F^-) \equiv (F^+ \subseteq L(h)) \wedge (L(h) \cap F^- = \emptyset)$.

The other axioms give some of the logical properties of the predicates. They are meant to be a small set which allows Feldman's theorems to be proved in a general form.

Axiom 5 $\text{Consistent}(h,F^+,F^-) \rightarrow \forall f^- \in F^- \neg \text{Accountsfor}(h,f^-)$.

Accountsfor is a consistent deduction principle.

Axiom 6 $F_1^+ \supseteq F_2^+ \wedge F_1^- \supseteq F_2^- \wedge \text{Consistent}(h,F_1^+,F_1^-) \rightarrow \text{Consistent}(h,F_2^+,F_2^-)$

Consistency is hereditary downwards (preserved by any operation which produces subsets).

Axiom 7 $\text{Consistent}(h,F^+,F^-) \rightarrow F^+ \cap F^- = \emptyset$.

The phenomena in F^- are effectively negated.

The next axioms use a special hypothesis, T, which functions as a tautology.

Axiom 8 $\forall f \ \text{Consistent}(T,f,\emptyset) \wedge \text{Consistent}(T,\emptyset,f)$.

No phenomenon is necessary but every one is possible.

Axiom 9 $\text{Consistent}(h,F^+,F^-) \rightarrow \text{Consistent}(h,F^+ \cup \{f\},F^-) \vee \text{Consistent}(h,F^+,F^- \cup \{f\})$.

This is a partial version of the law of the excluded middle.

Axiom 10 It is decidable whether or not $\text{Consistent}(T,F^+,F^-)$.

Axiom 11 $\text{Consistent}(T,F^+,F^-) \rightarrow \exists h(\text{Consistent}(h,F^+,F^-) \wedge \text{Accountsfor}(h,F^+))$

Every consistent finite set has a consistent explanation.

Information sequences

Information sequences are sequences of observations of the occurrence or non-occurrence of certain phenomena.

Formally, an information sequence is an infinite sequence in $(\{0,1\} \times \text{Phen})^{\infty}$ or else a finite one in $(\{0,1\} \times \text{Phen})^*$. We will display $\langle 1,f\rangle$ as $+f$ and $\langle 0,f\rangle$ as $-f$.

We will use the variables $I, I_1, \ldots$ to range over information sequences. $|I|$ is the length of I. Note that $0 \leq |I| \leq \infty$. If $0 < t \leq |I|$, $I(t)$ is the tth component of I. If I_1 is finite, $I_1 + I_2$ is that information sequence consisting of I_1 followed by I_2. If I_1 is finite then $n\,I_1 = \underbrace{I_1 + \ldots\ldots + I_1}_{n \text{ times}}$, where $n \geq 0$. More formally,

$0\,I_1$ = the empty information sequence,

$(n+1)I_1 = nI_1 + I_1$.

If $I_1 = I_2 + I_3$, where $|I_3| > 0$ then I_1 extends I_2. This is written as $I_1 > I_2$.

If $t \leq |I|$, then $I^t = \langle I(1),\ldots,I(t)\rangle$. If $t > |I|$, then $I^t = I$. The positive information in I is defined to be $S^+(I) = \{f \mid +f \text{ occurs in } I\}$. Similarly, we define $S^-(I) = \{f \mid -f \text{ occurs in } I\}$. I_1 agrees with I_2 iff $S^+(I_1) = S^+(I_2)$ and $S^-(I_1) = S^-(I_2)$.

I is complete iff $S^+(I) \cup S^-(I) = \text{Phen}$. If I is complete it is infinite. I is consistent iff for all t, Consistent$(T, S^+(I^t), S^-(I^t))$.

A hypothesis explains an information sequence I, for the subject $\mathcal{S}$ iff

$$\forall t \text{ Accountsfor}(h, S^+(I^t) \wedge \mathcal{S}) \wedge \text{Consistent}(h, S^+(I^t), S^-(I^t)).$$

When $\mathcal{S}$ = Phen we suppress, both here and elsewhere, any reference to it.

Induction machines

An induction machine, $\mathcal{M}$, is a recursive function from the set of finite information sequences to Hyp.

$\mathcal{M}$ identifies h in the limit on I iff $\exists \tau \forall t \geq \tau \; \mathcal{M}(I^t) = h$.

$\mathcal{M}$ matches an explanation of I for the subject $\mathcal{S}$ in the limit iff $\exists \tau \forall t \geq \tau \; \mathcal{M}(I^t)$ explains I for $\mathcal{S}$.

$\mathcal{M}$ approaches an explanation of I for the subject $\mathcal{S}$ iff

1) $\forall f \in S^+(I) \wedge \mathcal{S} \; \exists \tau \forall t \geq \tau$ Accountsfor$(\mathcal{M}(I^t), f)$.

2) $\forall h$ not explaining I for the subject $\mathcal{S}$, $\exists \tau \forall t \geq \tau \; \mathcal{M}(I^t) \neq h$.

The approach is strong iff, in addition:

3) $\exists h$ explaining I for $\mathcal{S}$ such that $\exists \tau \; \forall t \geq \tau \; \exists$ finite I' extending I^t and agreeing with I^t such that $\mathcal{M}(I') = h$.

This condition is slightly stronger than Feldman's for a strong approach.

3. Inferring hypotheses

Our first induction machine $\mathcal{M}_1$ is essentially due to Gold. It is defined under the assumption that Accountsfor is recursive, and relative to a subject, $\mathcal{S}$.

$$\mathcal{M}_1(I) = \begin{cases} h_1 \text{ (if } \neg \text{ Consistent}(T,S^+(I),S^-(I))) \\ \text{the first } h_i \text{ such that Accountsfor}(h_i,S^+(I) \wedge \mathcal{S}) \text{ and} \\ \quad \forall F^+ \subseteq S^+(I), F^- \subseteq S^-(I) \quad \forall m \leq |I| \neg M_C(h_i,F^+,F^-,|I|) \\ \text{(Otherwise).} \end{cases}$$

That $\mathcal{M}_1$ is total recursive follows from the assumption that Accountsfor is recursive and axioms 10 and 11.

Theorem 1 If h explains a consistent, complete I for $\mathcal{S}$ then $\mathcal{M}_1$ identifies a hypothesis, h', in the limit on I which explains I for $\mathcal{S}$.

Proof First we show that if h_t, does not explain I for $\mathcal{S}$ then $\exists \tau \forall t \geq \tau \mathcal{M}_1(I^t) \neq h_{t'}$.

There are two possibilities. Either $\neg$ Accountsfor$(h_{t'},f_{t_1})$ for some $f_{t_1} \in S^+(I) \wedge \mathcal{S}$ or else $\neg$ Consistent$(h_{t'},S^+(I^{t_1}),S^-(I^{t_1}))$ for some t_1. In the first case we can take $\tau = t_1$.

In the second, there is an m such that $M_C(h,S^+(I^{t_1}),S^-(I^{t_1}),m)$. As Phen is infinite and I is complete, there is a τ such that $|I^{\tau}| \geq m$. This τ has the necessary properties in this case.

Suppose that h' is the first hypothesis which explains I for $\mathcal{S}$. Then there is a time τ such that if $t \geq \tau$, $\mathcal{M}_1(I^t)$ is not a hypothesis occurring before h'. Now, $\forall$ t Accountsfor(h',$S^+(I^t) \wedge \mathcal{S}$) and since $\forall$ t Consistent(h',$S^+(I^t)$, $S^-(I^t)$), $\forall$ m M_C(h,$S^+(I^t)$, $S^-(I^t)$, m) by axioms 4 and 6. Hence $\forall t \geq \tau\ \mathcal{M}_1(I^t) = h'$. This concludes the proof.

If in addition, Consistent is recursive, then we assert, leaving the proof to the reader, that one can choose a simpler machine, $\mathcal{M}_2$.

$$\mathcal{M}_2(I) = \begin{cases} h_1 \text{ (if } \neg \text{ Consistent}(T,S^+(I),S^-(I))) \\ \text{the first } h_i \text{ such that Accountsfor}(h_i,S^+(I) \wedge \mathcal{S}) \\ \text{and Consistent}(h_i,S^+(I),S^-(I)). \\ \text{(Otherwise).} \end{cases}$$

We cannot extend the result to the case where Accountsfor is not recursive. In fact under the assumption:

$$\forall F^+,F^- \text{ Consistent}(T,F^+,F^-) \rightarrow [\ \exists f \text{ Consistent}(T,F^+ \cup \{f\},F^-) \wedge \text{Consistent}(T,F^+,F^- \cup \{f\})],$$

we can show that no machine can identify an explanatory hypothesis in the limit when $\mathcal{S}$ = Phen. The assumptions hold when Hyp is the set of general rewriting systems (Feldman, 1970) and Phen the corresponding set of strings.

Theorem 2 Suppose that the assumptions hold. For every machine, $\mathcal{M}$, there is a consistent, complete and recursive information sequence on

which $\mathcal{M}$ does not identify an explanatory hypothesis in the limit.

Proof Let $f_1, \ldots$ be an enumeration of Phen. We will find an I satisfying the conditions of the theorem such that $I(t) = \pm f_t$. Let us say that a sequence, I' is suitable iff when I'(t) is defined, $I'(t) = +f_t$, or else $I'(t) = -f_t$.

Suppose first that:

$$\exists \text{ finite suitable } I'(\text{Consistent}(T,S^+(I'),S^-(I')) \wedge$$
$$\forall \text{finite suitable } I'' > I'[\text{ Consistent}(T,S^+(I''),S^-(I'')) \rightarrow$$
$$(\mathcal{M}(I') = \mathcal{M}(I''))].$$

In this case we choose I' as guaranteed by the supposition. Let f_t be the first phenomenon such that $\text{Consistent}(T,S^+(I') \cup \{f_t\},S^-(I'))$ and $\text{Consistent}(T,S^+(I'), S^-(I') \cup \{f_t\})$. The existence of f_t is guaranteed by the assumption and axiom 7. Let I'' be a finite suitable extension of I' such that $I''(t) = +f_t \equiv \neg \text{Accountsfor}(\mathcal{M}(I'),f_t)$. The existence of I'' is guaranteed by axiom 9. As I'' is finite, it is recursive. Axioms 9 and 10 guarantee that I'' has an extension I which is a complete, consistent and suitable information sequence. Now $\mathcal{M}$ identifies $\mathcal{M}(I')$ in the limit, by the supposition. If $\neg \text{Accountsfor}(\mathcal{M}(I'),f_t)$ then $I(t) = +f_t$. If $\text{Accountsfor}(\mathcal{M}(I'),f_t)$ then $I(t) = -f_t$, and so, by axioms 5 and 6, $\neg \text{Consistent}(\mathcal{M}(I'),S^+(I'^t),S^-(I'^t))$.

Therefore under the supposition, $\mathcal{M}$ does not identify, in the limit, a hypothesis explaining I.

Let us assume, to the contrary that:

$\forall$ finite suitable I'(Consistent(T,S^+(I'),S^-(I'))

-> $\exists$ finite suitable I" $\succ$ I'(Consistent(T,S^+(I"),S^-(I"))

$\wedge$ M(I') $\neq$ M(I")).

In this case, I" may be obtained recursively from I' since $\mathcal{M}$ is recursive and by axiom 10, given I" one can tell by a recursive procedure whether or not Consistent(T,S^+(I"),S^-(I")). Therefore there is a recursive function g such that $\forall$ I'(Consistent(T,S^+(I'),S^-(I')) -> (g(I') is a finite suitable sequence extending I' such that Consistent(T,S^+(I"),S^-(I"))$\wedge$ $\mathcal{M}$(I') $\neq$ $\mathcal{M}$(I")).

Let I_0 = <+ f_1>. By axiom 8, Consistent(T,f_1,$\emptyset$). Then I_0<g(I_0)<....<g^t(I_0)<.... and so there is a unique, recursive I > g^t(I_0) (for all t $\geq$ 0) such that for every t' there is a t such that $I^{t'}$ < g^t(I_0). Therefore by the properties of g and axiom 6, Consistent(h,S_t^+(I),S_t^-(I)). Therefore I is consistent and is certainly complete. Now $\mathcal{M}$(g^t(I_0)) $\neq$ $\mathcal{M}$ (g^{t+1}(I_0)) (t $\geq$ 0). Therefore $\mathcal{M}$ cannot identify any hypothesis in the limit.

This eatablishes the theorem.

If Hyp includes every recursive predicate of Phen, then although $\mathcal{M}$ will not identify a hypothesis explaining I in the limit, there is one.

Notice also that the I described in the theorem, although recursive, is not obtained recursively from $\mathcal{M}$. We conjecture that there is no such recursive map when Hyp includes every recursive predicate of Phen. This is not the case if we are guaranteed that for every finite I',

λ f Accountsfor($\mathcal{M}$(I'),f) is always a recursive predicate on Phen. There is a unique recursive complete and consistent suitable information sequence recursively specified by:

1) $I(1) = +f_1$

2) $\neg$ Consistent(T,$S_t^+(I)$,$S_t^-(I) \cup \{f_{t+1}\}$) -> $I(t+1) = +f_{t+1}$.

3) $\neg$ Consistent(T,$S_t^+(I) \cup \{f_{t+1}\}$,$S_t^-(I)$) -> $I(t+1) = -f_{t+1}$.

4) Consistent(T,$S_t^+(I)$,$S_t^-(I) \cup \{f_{t+1}\}$) $\wedge$ Consistent(T,$S_t^+(I) \cup \{f_{t+1}\}$,$S_t(I)$) -> ($I(t+1) = +f_{t+1}$ <-> $\neg$ Accountsfor($\mathcal{M}(I^t)$,f_{t+1})).

Of course we are still using the assumption behind theorem 2. (The definition of I is very close to the familiar proof that there can be no recursive enumeration of the recursive functions.) In this case we can actually find an $\mathcal{M}'$ which will do as well as $\mathcal{M}$ on any I' $\neq$ I and will identify a hypothesis, in the limit, which explains I (provided Hyp contains all the recursive functions). Putnam (1967) has made similar observations.

Although it is not possible to devise a machine which will identify an explanation in the limit, it is possible to strongly approach one, using a machine $\mathcal{M}_3$.

To calculate $\mathcal{M}_3(I)$ proceed as follows:

1) If $\neg$ Consistent(T,$S^+(I)$,$S^-(I)$) then $\mathcal{M}_3(I) = h_1$.

2) Otherwise, find, by some fixed effective means, an h and an m such that $M_A(h,S^+(I) \wedge \delta, m)$ and $\forall m' \leq m + |I|$ $\forall F^+ \subseteq S^+(I), F^- \subseteq S^-(I)$ $\neg M_C(h,F^+,F^-,m')$. Then $\mathcal{M}_3(I)$ is the first h_i such that

$M_A(h_i, S^+(I) \wedge \mathcal{S}, m + |I|)$ and $\forall F^+ \subseteq S^+(I), F^- \subseteq S^-(I)\ \forall m' \leq m + |I|$ $\neg M_C(h, F^+, F^-, m')$.

Theorem 3 $\mathcal{M}_3$ strongly approaches an explanation in the limit on I for $\mathcal{S}$, if there is one.

Proof Suppose that there is an explanation of I for $\mathcal{S}$. Then I is consistent and so Accountsfor($\mathcal{M}_3(I^t), S^+(I^t) \wedge \mathcal{S}$) and so condition 1 for approaching a strong explanation is verified. Suppose h_i does not explain I. Then either $\neg$ Accountsfor(h_i, f_{t_1}) for some f_{t_1} in $S^+(I)$ or else $\neg$ Consistent($h_i, S^+(I^{t_1}), S^-(I^{t_1})$) for some t_1. In the first case, if $\tau \geq t_1$, $\mathcal{M}_3(I^\tau) \neq h_i$. In the second case there is a t_2 such that $M_C(h_i, S^+(I^{t_1}), S^-(I^{t_1}))$. Therefore if $t \geq \max(t_1, t_2)$, $\mathcal{M}_3(I^t) \neq h_i$. This verifies condition 2.

Let h_i be the first hypothesis explaining I for $\mathcal{S}$. For some τ, $\mathcal{M}_3(I^\tau)$ is not any h occurring before h_i. Suppose $t \geq \tau$ and choose an n such that $M_A(h_i, S^+(I^t) \wedge \mathcal{S}, n)$. If $I' = I^t + nI(t)$ then $\mathcal{M}_3(I') = h_i$. This verifies condition 3 and concludes the proof.

These theorems have all been concerned with good behaviour in the limit. It is worth noting their local behaviour.

Suppose I is consistent. The Accountsfor($\mathcal{M}_i(I^t), S^+(I^t) \wedge \mathcal{S}$) for i=1,3 and all t. Further Consistent($\mathcal{M}_1(I^t), S^+(I^t), S^-(I^t)$) for all t, although we only have, for i=2,3, $M_C(\mathcal{M}_i(I^t), S^+(I^t), S^-(I^t), m)$ where m depends on t and $m \to \infty$ as $t \to \infty$.

4. Inferring good hypotheses

By incorporating a complexity measure one can obtain better standards of good local behaviour.

We require that the standard ordering of the hypotheses, $h_1, h_2, \ldots$ is according to their simplicity. For example if the hypotheses are context-free grammars then, perhaps, if $j>i, h_j$ has no less symbols than h_i. The number of symbols will also order sets of clauses in this linear way.

The derivational complexity $d(F^+,h)$ of F^+ from h is defined when Accountsfor(h,F^+) and then,

$d(F^+,h)$ = the smallest integer m such that $M_A(h,F^+,m)$.

In other words, using a standard notation, d is that partial function defined by $d(F^+,h) = \mu\, m\ M_A(h,F^+,m)$.

The complexity function γ is a partial recursive function from $\mathcal{Y}$(Phen) x Hyp to $\mathbb{R}$, the set of the rationals. It combines the simplicity of a hypothesis with the derivational complexity.

There is a total recursive function γ': $N^2 \to \mathbb{R}$ increasing unboundedly with each of its arguments such that:

$$\gamma(F^+,h_i) = \gamma'(i,d(F^+,h_i)).$$

The machines $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$ could all be specified, using a recursive (under the appropriate conditions) predicate,

$P_j(1 \leq j \leq 3)$ on Hyp $\times$ $(\{0,1\} \times \text{Phen})^*$, by:

$\mathcal{M}_j(I)$ = the first h_i such that $P_j(h_i,I)$

For example:

$$P_4(h_i,I) \equiv (\neg\ \text{Consistent}(T,S^+(I),S^-(I)) \rightarrow h_i = h_1)$$
$$(\text{Consistent}(T,S^+(I),S^-(I))$$
$$\rightarrow (M_A(h_i,S^+(I) \wedge \mathcal{S},m + |I|)$$
$$\wedge\ \forall F^+ \subseteq S^+(I),\ F^- \subseteq S^-(I)\ \neg M_C(h_i,F^+,F^-,m + |I|))),$$

where m is obtained recursively from I.

In each case for every finite information sequence I there is an h such that $P_j(h,I)$. Further, if I is consistent then $P_j(h,I)$ implies that $\text{Accountsfor}(h,S^+(I) \wedge \mathcal{S})$.

We will define corresponding machines $\mathcal{M}'_j(1 \leq j \leq 3)$ with the properties:

1) $P_j(\mathcal{M}'_j(I),I)$
2) Suppose that I is consistent. If $P_j(h,I)$ then
$$\gamma(S^+(I) \wedge \mathcal{S},h) \geq \gamma(S^+(I) \wedge \mathcal{S},\ \mathcal{M}'_j(I)).$$

That is $\mathcal{M}'_j$ will choose a <u>best</u> machine rather than a first one. To compute $\mathcal{M}'_j(I)$ one proceeds as follows:

1) If I is not consistent, then $\mathcal{M}'_j(I) = \mathcal{M}_j(I)$.
2) Otherwise, compute $\mathcal{M}_j(I)$. Let k be the least integer such that $\gamma'(k,0) \geq \gamma(S^+(I) \wedge \mathcal{S},\ \mathcal{M}_j(I))$.

$\mathcal{M}_j(I)$ is that first hypothesis minimising $\gamma(S^+(I) \wedge \mathcal{S}, h)$ amongst those $h_l (1 \leq l \leq \max(k,j))$ such that $P_j(h_l, I)$.

To see that $\mathcal{M}'_j(I)$ is well-defined and effectively obtainable, note that γ' is computable and increases unboundedly with its first argument, thus ensuring the existence and computability of k, that P_j is recursive and that if $P_j(h_l, I)$ and I is consistent then Accountsfor$(h_l, S^+(I) \wedge \mathcal{S})$ and so $\gamma(S^+(I) \wedge \mathcal{S}, h_l)$ is defined.

Evidently $\mathcal{M}'_j(I)$ has property one. Suppose I is consistent and that $P_j(h_i, I)$. If $i \leq k$ then $\gamma(S^+(I) \wedge \mathcal{S}, h_i) \geq \gamma(S^+(I) \wedge \mathcal{S}, \mathcal{M}'_j(I))$. Otherwise $\gamma(S^+(I) \wedge \mathcal{S}, h_i) \geq \gamma'(i,0) \geq \gamma'(k,0) \geq \gamma(S^+(I) \wedge \mathcal{S}, \mathcal{M}_j(I)) \geq \gamma(S^+(I) \wedge \mathcal{S}, \mathcal{M}'_j(I))$. Therefore $\mathcal{M}'_j(I)$ also has property two.

Theorem 1 Suppose that Accountsfor is recursive. $\mathcal{M}'_2$ matches an explanation of I for $\mathcal{S}$ in the limit, if I has one. If $\gamma(S^+(I^t) \wedge \mathcal{S}, h)$ converges for all h explaining I for $\mathcal{S}$, then $\mathcal{M}'_2$ will eventually guess only hypotheses, h, which minimise $\lim_{t \to \infty} \gamma(S^+(I^t) \wedge \mathcal{S}, h)$.

Proof From theorem 3.1, $\mathcal{M}_2$ identifies some explanatory hypothesis h in the limit. So there is a τ such that if $t \geq \tau$, $\mathcal{M}_2(I^t) = h$. Hence, when $t \geq \tau$, the k calculated by $\mathcal{M}'_2$ will be independent of t and only a finite number of hypotheses will be considered by $\mathcal{M}'_2$. From the properties of P_2 developed in the proof of theorem 3.1, if h_i does not explain I for $\mathcal{S}$ then eventually $P_2(h_i, I^t)$ will always be false. Therefore in the limit $\mathcal{M}'_2$ will choose only hypotheses explaining I for $\mathcal{S}$. That is, $\mathcal{M}'_2$ matches an explanation of I for $\mathcal{S}$ in the limit. The

The second part of the theorem is obvious. This concludes the proof.

When both accountsfor and consistent are recursive, $\mathcal{M}_1'$ has the same limiting behaviour as $\mathcal{M}_2'$ although, of course, its local behaviour is better.

Theorem 2 Suppose I has an explanation, h_i for $\mathcal{S}$. $\mathcal{M}_3'$ approaches an explanation. If $\gamma(S^+(I^t) \wedge \mathcal{S}, h_i)$ is bounded as $t \to \infty$, then $\mathcal{M}_3'$ will only consider finitely many hypotheses and will match an explanation of I for $\mathcal{S}$ in the limit. If $\gamma(S^+(I^t) \wedge \mathcal{S}, h')$ converges for all h' explaining I for $\mathcal{S}$, then $\mathcal{M}_3'$ will eventually guess only hypotheses, h', which minimise $\lim_{t \to \infty} \gamma(S^+(I^t) \wedge \mathcal{S}, h')$.

Proof The properties of P_3 developed in the proof of theorem 3.3 show at once that $\mathcal{M}_3'$ approaches an explanation. If $\gamma(S^+(I^t) \wedge \mathcal{S}, h_i)$ is bounded, so is $d(S^+(I^t) \wedge \mathcal{S}, h_i)$. Suppose $t_1 \geq \max_{0<t<\infty} d(S^+(I^t) \wedge \mathcal{S}, h_i)$. Then $M_A(h_i, S^+(I^t) \wedge \mathcal{S}, t_1)$. Therefore, if $t \geq t_1, P_4(h_i, I^t)$. From the properties of γ we see that for some k, $l \geq k$ implies that $\gamma(S^+(I^t) \wedge \mathcal{S}, h_l) \geq \gamma(S^+(I^t) \wedge \mathcal{S}, h_i)$. Consequently $\mathcal{M}_3'$ only considers finitely many hypotheses. Since in general, it approaches an explanation, it must, in this case match one. The last part of the theorem is obvious and this concludes the proof.

5. Generalisation and hypothesis learning theory

The algorithms and theory developed in the previous chapters provide a class of induction machines each of which chooses a nicest explanatory hypothesis generalising a given H_0 and consistent with its knowledge. We will take a brief look at the behaviour in the limit of one such machine in a decidable case.

The hypothesis space, Hyp, is the set of finite sets of clauses containing no function symbols, other than constants.

Phen is the set of ground non-tautologous clauses containing no function symbols other than constants.

$\text{Accountsfor}(H,H_0)$ iff $H \leq H_0$.

$\text{Consistent}(H,H_0^+,H_0^-)$ iff $\forall H \wedge \bigwedge_{C \in H_0^+} C \wedge \bigwedge_{D \in H_0^-} D$ is consistent.

Since Accountsfor and Consistent are both recursive there is an algorithm which identifies an explanation in the limit. However, we will see that if we take $\mathcal{S} = \mathcal{S}_{cpg}$, any algorithm which chooses a nicest explanation need not match an explanation in the limit even on natural information sequences which arise from repeated presentations of the formal problem.

Suppose that f_i $(i \geq 0)$ is a sequence of ground literals and Ev is a map from $\{f_i \mid i \geq 0\}$ to conjunctions of ground literals such that $\overline{Ev(f_i)} \cup \{f_i\}$ is in Phen and $\{f_i \wedge Ev(f_i) \mid i=1,n\}$ is consistent. Let $Ev(f_i) = e_{i1} \wedge \ldots \wedge e_{ij(i)}$ where the e_{ij} are ground literals and we

let $\mathcal{S} = \{\overline{Ev(f_i)} \cup \{f_i\} \mid i \geq 0\}$; a natural complete and consistent information sequence for the f_i and Ev is one such that $S^+(I) \supseteq \mathcal{S} \cup \{e_{ij} \mid i \geq 0, 1 \leq j \leq j(i)\}$ and if $i' > i$, $+(\overline{Ev(f_i)} \cup \{f_i\})$ occurs, doing so before $+(\overline{Ev(f_{i'})} \cup \{f_{i'}\})$, in I.

Recollect the $D_2 j$ of chapter 3, section 3.3.2 which provided an example of an infinite strictly decreasing chain and the γ^i_j of the representation theorem, theorem 3.3.3.2.4 and their properties.

One can find f_i and an Ev satisfying the necessary conditions outlined above such that:

$$f_{2i} = Q(x_{[1,2^i]})\ \gamma^1_{n(i)}$$
$$f_{2i-1} = Q(x_{[1,2^i]})\ \gamma^2_{n(i)}$$
$$\overline{Ev(f_{2i})} = D_2{}^i\ \gamma^1_{n(i)}$$
$$\overline{Ev(f_{2i-1})} = D_2{}^i\ \gamma^2_{n(i)}$$

for suitably large n(i) and all $i \geq 1$.

Now such an f_i and Ev has a natural information sequence explained by $\{Q(x)\}$. Yet by the properties of the D_{2j} and the representation theorem 3.3.3.2.4 the nicest hypothesis, in the sense of $\mathcal{S}_{cpg}$, explaining $\{\overline{Ev(f_i)} \cup \{f_i\} \mid 1 \leq i \leq 2n\}$ and consistent with $S^+(I)$ and the set of negations of members of $S^-(I)$, is $D_2 n \cup \{Q(x_{[1,2^n]})\} = E_n$ say. Thus at time t_n, E_n will be chosen. As this is a strictly decreasing sequence no member of which explains I, our machine will not even match an explanation in the limit.

One might object that $\mathcal{S}$ does not contain enough instances of the explanation. If it contains all instances, we will see that choosing the nicest will match an explanation in the limit.

For, suppose H is an explanation of a natural, complete and consistent information sequence I and that $\mathcal{S} \supseteq H_0 = \bigcup \{H\ \gamma_j^i \mid i,j \geq 1\}$. Then for some t_0, H is equivalent to a subset of $\mathcal{L}_\phi(S^+(I^{t_0}) \wedge \mathcal{S})$ by the representation theorem. Therefore if $t \geq t_0$, the nicest explanatory hypothesis will have complexity less than or equal to that of H. Let C be in H. The set $\{C\ \gamma_j^i \mid i,j \geq 1,\ C\ \gamma_j^i$ is ground and if $1 \leq j' \leq j, a_{ij'}$ does not occur in C$\}$ is infinite. Therefore it must eventually be the case since the complexity of the nicest explanatory hypothesis is bounded that any nicest explanatory hypothesis must contain a clause subsuming at least two members of this set, and so subsuming C itself, by the representation theorem. Therefore after some time $t_1 \geq t_0$ any nicest explanatory hypothesis must generalise H. Further, after t_1 no clause can occur in such a hypothesis which does not subsume some clause of H, as H is an explanation. Therefore by the minimality, with respect to $\prec_{cpg}$ requirement, such a hypothesis will be equivalent to a subset of $\mathcal{L}_\phi(H)$, generalising H. There is a fixed collection of such subsets of $\mathcal{L}_\phi(H)$ of equal cardinality any member of which is consistent with $S^+(I)$ and the set of negations of members of $S^-(I)$ such that eventually the nicest explanation will always be equivalent to one of this set, the choice being determined solely by power. Which has the greatest power depends on, amongst other things, the order of occurrence of the $C\ \gamma_j^{i'}$ $(C \in H)$ and so, in general, any machine which

chooses a nicest explanation will match rather than identify an explanation in the limit.

These results are critically dependent on $\succ_{cpg}$. Let us look instead at $\succ_{s'g}$. $H \succ_{s'g} H'$ iff H has a smaller number of symbol occurrences than H or, if they have the same number of symbols, then $H' \leq H$.

There is now a machine which will identify a nicest explanation in the limit. Let $x_1,\ldots,x_i,\ldots$ be an infinite list of distinct variables. Let $\mathcal{H} = \{H \mid$ If H contains n variables these are precisely $x_1,\ldots,x_n\}$. Notice that any H has an alphabetic variant in $\mathcal{H}$. Now there are only finitely many members of $\mathcal{H}$ with a fixed number of symbols, and $\leq$ is a quasi-ordering. Hence $\mathcal{H}$ can be enumerated as $H_1,H_2,\ldots$ where $H_1,\ldots,H_{i_1}$ have one symbol, $H_{i_1},\ldots,H_{i_2}$ have two symbols and so on, and where if H_j and $H_{j'}$ have the same number of symbols then if $H_j \leq H_{j'}$ either $j \leq j'$ or $H_{j'} \leq H_j$. This follows easily from the fact that any finite partial ordering can be enlarged to a linear one. Consequently, if $j < j'$, $H_j \succ_{s'g} H_{j'}$. Since every H has an alphabetic variant in $\mathcal{H}$, if there is an explanation of some given I there will be one in the enumeration. Consequently that machine which picks the first explanation in the enumeration which consistently explains the phenomena will identify an explanation in the limit and will always pick the nicest.

These examples show that much work remains to be done in picking niceness relations.

5. Conclusions

We have presented a generalisation of Feldman's (1970) work. Feldman remarks that one of the more interesting theoretical problems was the inference of systems with semantics. In so far as our general theory covers systems using the predicate calculus which has a semantics, we have covered this problem. Here however we see that the notion of complexity seems inadequate to apply to some of the niceness relations developed earlier.

The various machines used and developed do not behave at all in accordance with any hypothesis discovery procedure employed by practising scientists. One could look for reasons in two general directions. A better description of normal scientific practice, including the discovery methods used would lead to more realistic machines. For example one might study how old theories are modified to obtain new ones. This is a descriptive approach.

On the other hand, it may be that the machines do not behave in the way they ought to. There is no formulation of any notions of justification of criticism of hypotheses. This is a normative approach.

Leaving these general points aside, the machines all have one deficiency, they are extremely inefficient. Each one would take so long to operate that the process of hypothesis discovery would lag irretrievably far behind the process of information acquisition. We believe therefore that it would be illuminating to formalise and prove

the conjecture:

Suppose Accountsfor and Consistent are recursive. Then there is a "natural" model of the axioms and choice of $\mathcal{S}$ such that no machine can efficiently identify an explanation in the limit on every (or almost every) explainable information sequence I.

Such a proof would show that it is necessary to consider special cases and methods, even from this simple point of view.

References

Amarel, S. (1962) On the automatic formation of a computer program which represents a theory. Self-Organizing Systems (eds. M.C. Yovits, G.T. Jacobi and G.D. Goldstein) Washington: Spartan Books, pp. 107-176.

Amarel, S. (1971) Representations and modeling in problems of program formation. Machine Intelligence 6 (eds. B. Meltzer and D. Michie) Edinburgh: Edinburgh University Press, pp. 411-466.

Andrews, P.B. (1968) Resolution with merging. J.A.C.M., Vol. 15, No. 3, 367-381.

Barrow, H.G. and Salter, S.H. (1969) Design of low-cost equipment for cognitive robot research. Machine Intelligence 5 (eds. B. Meltzer and D. Michie) Edinburgh: Edinburgh University Press, pp. 555-566.

Barrow, H.G. and Popplestone, R.J. (1971) Relational descriptions in picture-processing. Machine Intelligence 6 (eds. B. Meltzer and D. Michie) Edinburgh: Edinburgh University Press, pp. 377-396.

Becker, J.D. (1969) The modelling of simple analogic and inductive processes in a semantic memory system. Proceedings of the International Joint Conference in Artificial Intelligence, pp. 655-668.

Becker, J.D. (1970) An information-processing model of intermediate-level cognition. Stanford A.I. Memo. No. 119. Computer Science Department, Stanford University.

Black, M. (1966) Notes on the "Paradoxes of Confirmation". Aspects of Inductive Logic (eds. J. Hintikka and P. Suppes) Amsterdam: North Holland.

Bruner, J.S., Goodnow, J.J. and Austin, G.A. (1956) A study of thinking. New York: Wiley.

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

Buchanan, B.G., Sutherland, G.L. and Feigenbaum, E.A. (1969) Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry. Machine Intelligence 4 (eds. B. Meltzer and D. Michie) Edinburgh: Edinburgh University Press, pp. 209-254.

Buchanan, B.G., Sutherland, G.L. and Feigenbaum, E.A. (1970) Rediscovering some problems of artificial intelligence in the context of organic chemistry. Machine Intelligence 5 (eds. B. Meltzer and D. Michie) Edinburgh: Edinburgh University Press, pp. 253-280.

Bunge, M. (1961) The weight of simplicity in the construction and assaying of scientific hypotheses. Philosophy of Science, Vol. 28, 120-149.

Burstall, R.M., Collins, J.S. and Popplestone, R.J. (1971) Programming in POP-2. Edinburgh: Edinburgh University Press.

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

Carnap, R. (1952) The Continuum of Inductive Methods. Chicago.

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

Chang, C.L. (1970) Renameable paramodulation for automatic theorem proving with equality. Artificial Intelligence, Vol. 1, No. 4, 247-256.

Chomsky, N. and Miller, G. (1957) Pattern Conception Report No. AFCRC-TN-57-57.

Eberle, R., Kaplan, D. and Montague, R. (1961) Hempel and Oppenheim on explanation. Philosophy of Science, Vol. 28, 418-428.

Evans, T.G. (1968) A program for the solution of geometric-analogy intelligence test questions. Semantic Information Processing (ed. M. Minsky) Cambridge, Massachusetts: M.I.T. Press.

Feigenbaum, E.A., Buchanan, B.G. and Lederberg, J. (1971) On generality and problem solving: A case study using the DENDRAL program. Machine Intelligence 6 (eds. B. Meltzer and D. Michie) Edinburgh: Edinburgh University Press, pp. 165-190.

Feldman, J.D. (1967) First thoughts on grammatical inference. Stanford A.I. Memo. No. 55. Computer Science Department, Stanford University.

Feldman, J. (1970) Some decidability results on grammatical inference and complexity. Stanford A.I. Memo. No. 93.1. Computer Science Department, Stanford University.

Feldman, J.D. and Biermann, A.W. (1970) On the synthesis of finite-state acceptors. Stanford A.I. Memo. No. 114. Computer Science Department, Stanford University.

Feldman, J.A., Gips, J., Horning, J.J. and Reder, S. (1969) Grammatical complexity and inference. Stanford A.I. Memo. No. 89. Computer Science Department, Stanford University.

Gold, M. (1967) Language identification in the limit. Information and Control, Vol. 10, 447-474.

Goodman, N. (1959) Recent developments in the theory of simplicity. Philosophy and Phenomenological Research, Vol. 19, 429-446.

Goodman, N. (1961) Safety, strength, simplicity. Philosophy of Science, Vol. 28, 150-151.

Goodman, N. (1965) Fact, fiction and forecast. Indianapolis: Bobbs-Merrill.

Hayes, P.J. (1971) A logic of actions. Machine Intelligence 6 (eds. B. Meltzer and D. Michie) Edinburgh: Edinburgh University Press, pp. 495-520.

Hempel, C.G. (1945) Studies in the logic of confirmation. Mind, Vol. 54, 1-26, 91-121.

Hempel, C.G. (1962) Deductive-nomological versus statistical explanation. Minnesota Studies in the Philosophy of Science, Vol. 3 (eds. H. Feigl, and G. Maxwell) Minneapolis: University of Minnesota Press, pp. 98-169.

Hempel, C.G. and Oppenheim, P. (1948) Studies in the logic of explanation. Philosophy of Science, Vol. 15, 135-175.

Hewitt, C. (1968) Functional abstraction in LISP and PLANNER. Artificial Intelligence Memo. No. 151. M.I.T. (Project MAC).

Hilpinen, R. (1968) Rules of acceptance and inductive logic. Acta Philosophica Fennica, Vol. 22. Amsterdam: North Holland.

Hintikka, J. (1953) Distributive normal forms in the calculus of predicates. Acta Philosophica Fennica, Vol. 6. Amsterdam: North Holland.

Hintikka, J. (1965a) Towards a theory of inductive generalization. Proc. 1964 Intern. Congress for Logic, Methodology, and Philosophy of Science (ed. Y. Bar-Hillel) Amsterdam: North Holland, pp. 274-288.

Hintikka, J. (1965b) Distributive normal forms in first-order logic. Formal systems and recursive functions. Proceedings of the 1963 Logic Colloquium in Oxford (ed. J.N. Crosley) Amsterdam: North Holland.

Hintikka, J. (1966) A two-dimensional continuum of inductive methods. Aspects of Inductive Logic (eds. J. Hintikka and P. Suppes) Amsterdam: North Holland.

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

Horning, J.J. (1969) A study of grammatical inference. Stanford A.I. Memo. No. 98. Computer Science Department, Stanford University.

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

Kemeny, J.G. (1953) A logical measure function. Journal of Symbolic Logic, Vol. 18, 289-308.

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

Kling, R.E. (1971) A paradigm for reasoning by analogy. Artificial Intelligence Group Technical Note 47R. Stanford Research Institute, Menlo Park, California.

Kowalski, R. (1969) Search strategies for theorem-proving. Machine Intelligence 5 (eds. B. Meltzer and D. Michie) Edinburgh: Edinburgh University Press, pp. 181-201.

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 (ed. I. Lakatos) Amsterdam: North Holland, pp. 138-157. Reprinted with a postscript in The Philosophy of Mathematics (ed. J. Hintikka) Oxford: Oxford University Press, (1969), pp. 78-94.

Kreisel, G. (1970) Hilbert's Programme and the search for automatic proof procedures. Lecture notes in mathematics (eds. A. Dold and B. Eckmann) Heidelberg and Zurich: Springer-Verlag, pp. 128-146.

Kyburg, H.E. (1961) Probability, rationality and a rule of detachment. Middleton, Conn.: Wesleyan University Press.

Kyburg, H.E. (1964) Recent work in inductive logic. American Philosophical Quarterly, 1, 249-287.

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 (eds. B. Meltzer and D. Michie) Edinburgh: Edinburgh University Press, pp. 165-179.

Minsky, M.M. and Papert, S.A. (1969) Perceptrons: An Introduction to Computational Geometry. Cambridge, Massachusetts: M.I.T. Press.

Nilsson, N.J. (1965) Learning Machines: Foundations of Trainable Pattern Classifying Systems. New York: McGraw-Hill.

Perryman, G. (1970) Discovering the structure of an automaton from partial information. M.Sc. Thesis. Department of Machine Intelligence and Perception, University of Edinburgh.

Plotkin, G.D. (1970) A note on inductive generalization. Machine Intelligence 5 (eds. B. Meltzer and D. Michie) Edinburgh: Edinburgh University Press, pp. 153-163.

Plotkin, G.D. (1971) A further note on inductive generalization. Machine Intelligence 6 (eds. B. Meltzer and D. Michie) Edinburgh: Edinburgh University Press, pp. 101-124.

Pólya, G. (1954) Mathematics and plausible reasoning. Volume 1. Induction and Analogy in Mathematics. Princeton: Princeton University Press.

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

Pólya, G. (1968) Mathematics and plausible reasoning. Volume 2. Patterns of Plausible Inference. Princeton: Princeton University Press.

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

Popplestone, R.J. (1970) An experiment in automatic induction. Machine Intelligence 5 (eds. B. Meltzer and D. Michie) Edinburgh: Edinburgh University Press, pp. 203-215.

Putnam, H. (1956) A definition of degree of confirmation for very rich languages. Philosophy of Science, Vol. 23, 58-62.

Putnam, H. (1963) Degree of confirmation and inductive logic. The Philosophy of Rudolf Carnap (ed. P.A. Schilpp) La Salle, Illinois: Open Court Publishing Co., pp. 761-783.

Putnam, H. (1967) Probability and confirmation. Philosophy of Science Today (ed. S. Morgenbesser) New York: Basic Books.

Quine, W. (1955) A way to simplify truth functions. Amer. Math. Month., 62, 627-631.

Reynolds, J.C. (1970) Transformational systems and the algebraic structure of atomic formulae. Machine Intelligence 5 (eds. B. Meltzer and D. Michie) Edinburgh: Edinburgh University Press, pp. 135-152.

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

Schilpp. P.A. (ed.) (1963) The Philosophy of Rudolf Carnap. La Salle.

Schoenfield, J.R. (1967) Mathematical Logic. Massachusetts: Addison-Wesley.

Shamir, E. (1962) A remark on discovery algorithms for grammars. Information and Control, Vol. 3, 246-251.

Solomonoff, P.J. (1964) A formal theory of inductive inference. Information and Control, Vol. 7, 1-22, 224-254.

Tredwill, R.F. (1965) The problem of counterfactuals. Philosophy of Science, Vol. 32, 310-323.

Winston, P.H. (1970) Learning structural descriptions from examples. Report MAC TR-76 (Thesis). Cambridge, Massachusetts, Project MAC, M.I.T.