Theory Discovery and Hypothesis Language

by

Kevin T. Kelly

June 1988

Report CMU-PHIL-7

Philosophy
Methodology
Logic

Pittsburgh, Pennsylvania 15213-3890

Theory Discovery and Hypothesis Language

by

Kevin T. Kelly

June 1988

Report CMU-PHIL-7

Philosophy
Methodology
Logic

Pittsburgh, Pennsylvania 15213-3890

Theory Discovery and the Hypothesis Language

Kevin T. Kelly

(kk3n@andrew.cmu.edu)

Department of Philosophy, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.

Abstract

This paper develops a precise framework in which to compare the discovery problems determined by a wide range of distinct hypothesis languages. Twelve theorems are presented which provide a comprehensive picture of the solvability of these problems according to four intuitively motivated criteria of scientific success.

1. Introduction

It is widely recognized in the artificial intelligence literature that the choice of syntax in data structures can be relevant to the performance of a system that employs them. This is especially true when selecting an hypothesis language for a machine learning system. Consider a learning device that is responsible for discovering only what it can express. Insofar as the hypothesis language of the system is richer, its discovery task is more difficult, for the enriched language permits it to formulate more possibilities and finer distinctions. These additional, refined possibilities are more difficult to distinguish on the basis of the same data.

This paper examines systematically the impact of various hypothesis language restrictions on the difficulty of the problem of discovering a complete, true theory in this language. First, a formal setting for the investigation is formulated. Then a series of theorems are presented that determine various senses in which the problem of discovering a complete, true theory in the language is solvable or not solvable. Finally, I state two questions about the framework that remain open. Due to limitations of space and the number of results to be presented, I can only hint at the details, so the presentation will be kept as informal as possible.[1]

---

[1]Details can be found in Kelly and Glymour (in press).

2. Two Senses of Success

Imagine two pictures of scientific inquiry. In the first, the scientific community labors over its best theory of the world until on one fortunate day science is complete. Happily, the scientists are not sent home, because nobody can be *sure* that the theory is complete (it might be refuted by the next observation). But as a matter of fact, it is true, and the scientists never again find any reasons to reject it. In this sort of situation, the complete truth hits us all at once and then never leaves. We call this kind of convergence to the truth *EA* convergence to indicate that *there is* a time after which *all* expressible questions are settled. EA convergence has been proposed as a criterion of successful inquiry by a number of philosophers, computer scientists, and methodologists (Gold 1967, Putnam 1963, Osherson and Weinstein 1986, Angluin and Smith 1982, and Shapiro, 1981).

Now consider a somewhat less joyous but nonetheless optimistic scenario. In this situation, the scientists add more and more truths to their body of beliefs, and weed out ever more falsehoods, but there need not be any single time at which the entire theory is true and complete. This is very much like the picture of inquiry proposed by the philosophers C.S. Peirce (Peirce 1965) and Karl R. Popper (Popper 1963). Since *every* expressible question is correctly answered by inquiry at *some* time, we call this kind of convergence *AE* convergence for short. Observe that from a practical point of view, not much is lost in weakening EA convergence to AE convergence. If we *use* our theory to answer general questions we have about the universe, it is still the case in an AE convergent inquiry that for each such question it is eventually settled correctly. For a particular question, this is all we get out of EA convergence. We are only denied the spiritual pleasure of knowing that there is a time after which all conceivable questions are correctly answered by the theory at once.

3. A Mathematical Framework

The following definitions merely clarify the picture just presented. Let L be the system's *hypothesis language*. We assume that L is some fragment of a first-order logical language. Let M be a countable relational structure for L. The *complete L-theory of M* is just the set of all L-sentences true in M. Let an *evidence presentation*[2] for M be an infinite tape of literals of L with the following property: there is an interpretation of variables so that all and only the literals true under this interpretation in M occur on the tape. If e is an evidence presentation, then e_n denotes the initial segment of length n of e.

---

[2]The following definition is due to Osherson and Weinstein (Osherson and Weinstein 1986).

An *investigator* is a function from finite sequences of literals to recursive axiomatizations of L-theories. Concretely, we can take the output of an investigator to be an index for a procedure that decides a (possibly infinite) set of axioms for his conjectured theory. In the special case of finite axiomatizations, the axioms can be output themselves. We distinguish *effective* and *ineffective* investigators. The former have programs that compute them, and the latter do not. Ideally, we want impossibility results for ineffective theorists and possibility results for effective theorists.

Our two notions of convergence can now be expressed clearly:

- Investigator ϕ EA-converges to the complete theory of M with respect to L on evidence presentation e if and only if for all but finitely many n, for each sentence $s \varepsilon L$, $\phi(e_n|=s$ if and only if $M|=s$.
- Investigator ϕ AE-converges to the complete theory of M with respect to L on evidence presentation e if and only if for each $s \varepsilon L$, for all but finitely many n, $\phi(e_n|=s$ if and only if $M|=s$.

In the machine learning literature, it is usual to describe the discovery problem as finding and "unknown" true theory. The requirement that the theory be "unknown" is, presumably, intended to remove from contention as serious investigators machines that ignore the data, and always output the same canned, albeit true, conjecture. We can arrive at a clear theory if we take "unknown" in the sense of "arbitrarily selected from some class". That is, we can define an *inductive problem* to be a set of relational structures or "possible worlds" and we can require any solution to the problem to converge to the complete theory of an arbitrarily selected structure on the basis of its evidence presentation. Since the lookup-table device can only converge to one theory, it fails to solve the problem. Accordingly, let K be a class of structures. Then we define

- Investigator ϕ *AE (EA* identifies collection K with respect to L) if and only if ϕ AE (EA) converges to the complete theory of each structure M in K with respect to L on each data presentation for M.
- K is *AE (EA* solvable with respect to L) if and only if there is an investigator ϕ such that ϕ AE (EA) identifies collection K with respect to L.
- K is *effectively AE (EA* solvable with respect to L) if and only if there is a computable investigator ϕ such that ϕ AE (EA) identifies collection K with respect to L.

To identify a class of structures in the appropriate sense is to solve the inductive problem the class of structures poses. As the range of possible worlds in the problem is enriched, the problem facing the theorist is made correspondingly more difficult, and a solution to a harder problem will tend to be slower.

It is of the utmost importance that speed not be confused with efficiency. Efficiency is optimal performance *for the problem solved*, so a highly efficient solution to a hard

problem may be very slow, while a simple-minded, inefficient solution to an easy problem may be very fast. Unless we characterize the computational problem under consideration with mathematical precision, it is too easy to begin a paper with a slow solution to a hard problem, to end it with a fast, inefficient solution to a trivial problem, and to announce the result as progress on the original problem.

Our four senses of success are therefore interesting in that they provide a coarse scale of comparison for the *intrinsic* difficulties of "large" discovery problems. In (Kelly and Glymour) it is shown that the criteria have the following relative difficulties:

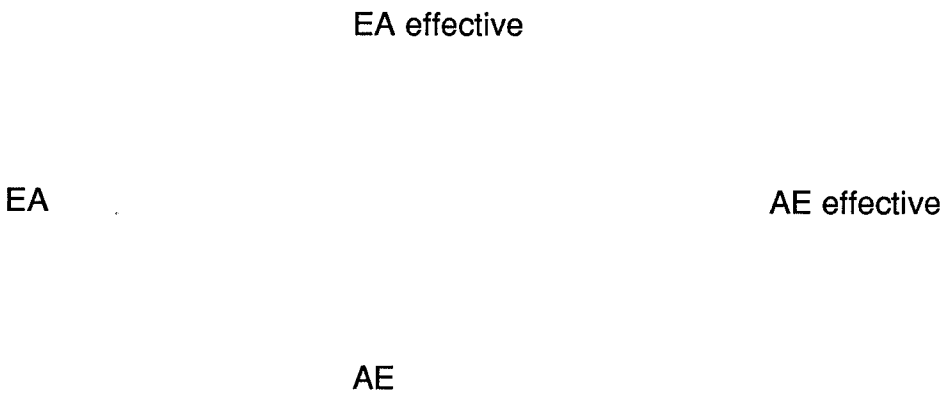

Figure 1: The Relative Difficulties of Four Criteria of Theory Identification

That is, any problem that is EA effectively identifiable is EA identifiable and AE effectively identifiable, and any problem that is either EA identifiable or effectively AE identifiable is AE identifiable. But no other implication holds. So for example, a problem that is AE effectively identifiable but not effectively EA identifiable is harder than a problem that is EA identifiable.

The hardest problem an investigator using hypothesis language L could possibly solve is to find the complete theory of an arbitrary structure whose particular facts can all be surveyed in the limit. Accordingly, we take the *unrestricted theorizing problem* for L to be the set of all *countable* relational structures for L.

The task remains to formulate a syntactic categorization of first-order hypothesis languages that is related naturally to expressive power and that is also simple, useful, and familiar. Expressive power and intractability are two sides of the same coin. The more a language can express, the more difficult it is to decide logical relations (e.g. entailment, consistency, and validity) over its sentences. Hence, it makes sense to look to the decidability theorems of mathematical logic for properties of hypothesis languages that are related to expressive power (i.e. that make entailment harder to decide). As it turns out, there are four simple syntactic features of an hypothesis language that have a major impact on the decidability of entailment, validity, and

consistency (and hence upon expressive power).

First, there is the maximum number of arguments for any predicate in the vocabulary. A language with only unary predicates is said to be *monadic*, and validity, entailment, and consistency are all decidable for such languages. Add a binary predicate, and these problems all turn out not to be decidable. So predicate arity can be expected to make an important difference in discovery problems as well. For simplicity in stating our results, we say that L is P_n if and only if the maximum arity of any predicate in the vocabulary of L is n. L is P_0, then L has no predicate symbols.

Second, we should also consider the maximum permissible arity of function symbols in the language. We say that L is F_n if the maximum arity of any predicate in the vocabulary of L is n. Again, if L is F_0, then L has no function symbols.

Third, a language L can either have or fail to have the identity predicate "=". If it does, we say that L is I_1. Otherwise L is I_0. Identity is important to consider separately because it adds to the expressive power of monadic languages, but not to the extent that an arbitrary, binary predicate does.

Finally, an important determinant of the power of an hypothesis language is the complexity of quantifier prefixes. An hypothesis is in *prenex normal form* just in case all its quantifiers are out front. It is a familiar fact that any first-order hypothesis has a logically equivalent formulation in prenex normal form. A prenex hypothesis is said to be Σ_n just in case its prefix begins with an existential quantifier and there are at most n+1 alternations between blocks of existential and universal quantifiers. So for example, the hypothesis

ExEyAz(f(x,y)=y)

is Σ_2. An hypothesis in prenex normal form is Π_n just in case its prefix begins with a universal quantifier and there are at most n+1 alternations between blocks of universal and existential quantifiers. An hypothesis language L is Π_n (or Σ_n) if and only if each of its sentences is Π_n (or Σ_n, respectively).

We characterize a language by clamping together its properties. So for example, L is $P_1F_1I_0\Pi_1$ if and only if it has at most unary predicates, it has no function symbols or identity, and its sentences are all purely universal (i.e. involves only quantifier prefixes consisting of all universal quantifiers). Notice that a language with no limitation on the number of quantifier alternations can fail to be either Π_n or Σ_n, for all n. In this case, we just drop the quantifier alternation bound from its description (e.g. $P_3F_4I_0$).

To acquire a feel for this notation, consider some examples familiar in machine learning applications. Consider *Boolean concepts* (Mitchell 1982), such as

Brat(x) <---> Disorderly(x) & Child(x)

A Boolean concept language is an hypothesis language in $P_1F_0I_0\Pi_1$. That is, a Boolean concept may involve, no function symbols, no identity, no existential quantifiers, and only unary predicates. *Structural descriptions* (Winston 1975), on the other hand, are hypotheses like the following:

```
(Ax)(Ew)(Ey)(Ez)
    [Arch(x)<--->
        [Part_of(w,x) &
        Part_of(y,x) &
        Part_of(z,x) &
        -Touch(w,y) &
        On_top(z,w) &
        On_top(z,y)]]
```

Hence, the structural descriptions are an hypothesis language in $P_nF_0I_0\Pi_2$, for some n typically greater than one. In the "learning to plan" literature, the learning agent is often provided with "precondition-postcondition rules" (Carbonell 1987) of the form

(Ax)[Glass(x) & Polished(x) ---> Reflective(polish(x))]

"Precondition-postcondition rules" are evidently in an hypothesis language in $P_1F_1I_0\Pi_1$. Ehud Shapiro's (1981) model inference system can generate sophisticated axioms such as

(Ax)(Append(nil,x,x)

(Aw)(Ax)(Ay)(Az)(Append(x,y,z) ---> Append(cons(w,x),y,cons(w,z))

Hypotheses of this sort are in $P_3F_2I_0\Pi_1$ languages. These typical cases are just a tiny fraction of the syntactic possibilities caught in our classification scheme.

Let D be an arbitrary language description of the sort just described (e.g. $P_nF_mI_b\Sigma_k$). It is useful to think of D as denoting the class of all languages having the properties listed. We now define the following, compact notation for stating our results.

- $D\varepsilon$[[AE]] if and only if for each $L\varepsilon D$ the unrestricted theorizing problem for L is AE solvable with respect to L.
- $D\varepsilon$[[EA]] if and only if for each $L\varepsilon D$ the unrestricted theorizing problem for L is EA solvable with respect to L.
- $D\varepsilon$[[AEe]] if and only if for each $L\varepsilon D$ the unrestricted theorizing problem for L is effectively AE solvable with respect to L.
- $D\varepsilon$[[EAe]] if and only if for each $L\varepsilon D$ the unrestricted theorizing problem for L is effectively EA solvable with respect to L.

4. Theorems

Now it is possible to express the results.

Positive Results

Prop 1: $P_1F_0I_0\varepsilon$[[EAe]]

Prop 2: $P_1F_0I_1\varepsilon$[[AEe]]

Prop 3: $P_1F_1I_0\varepsilon$[[AE]]

Prop 4: $P_nF_0I_1\Pi_1\varepsilon$[[AEe]], for all n.

Prop 5: $P_nF_mI_1\Sigma_1\varepsilon$[[AEe]], for all n,m.

Prop 6: $P_nF_mI_1\Pi_1\varepsilon$[[AE]], for all n,m.

Negative Results

Prop 7: neither $P_2F_0I_0\Pi_1$ nor $P_2F_0I_0\Sigma_1$ is in [[EA]].

Prop 8: neither $P_0F_0I_1\Pi_1$ nor $P_0F_0I_1\Sigma_1$ is in [[EA]]

Prop 9: neither $P_1F_1I_0\Pi_1$ nor $P_1F_1I_0\Sigma_1$ is in [[EA]].

Prop 10: neither $P_1F_2I_0\Pi_2$ nor $P_1F_2I_0\Sigma_2$ is in [[AE]].

Prop 11: neither $P_2F_0I_0\Sigma_2$ nor $P_2F_0I_0\Pi_2$ is in [[AE]]

Prop 12: neither $P_0F_1I_1\Pi_2$ nor $P_0F_1I_1\Sigma_2$ is in [[AE]].

Notice that if the unrestricted problems for languages in $P_iF_jI_b\Pi_k$ or in $P_iF_jI_b\Sigma_k$ fail to be solvable in a given sense then no class whose parameter values are at least as great as i,j,b and k is solvable in this sense. And if the unrestricted problems for languages in a class are solvable in a given sense, then the unrestricted problems for all smaller parameter values are also solvable in this sense. Hence, the above results settle nearly all solvability questions in the framework defined in this paper. In fact, what is undetermiend can be summarized by the following two questions:

Open Question 1: $P_nF_mI_1\Pi_1\varepsilon$[[AEe]], for all n,m?

Open Question 2: $P_1F_1I_0\varepsilon$[[AEe]]?

The results reveal a rich and complicated relationship between hypothesis language syntax and the solvability of the unrestricted theorizing problem for that language.

Proposition 1 says that if we have just unary predicates but no function symbols or identity, then even with no restriction whatever on quantifier complexity, the unrestricted theorizing problem must be solvable by an effective agent in the EA or "all at once" sense. I believe that the discovery technique employed in the proof is novel. It works like this. Any sentence in such a language may be put into an equivalent version in *primary normal form* (i.e. a Boolean combination of literals, purely universal disjunctions and purely existential disjunctions (Hilbert 1968)). So for example, consider the following transformation:

AxEy[(Px & Qy) v (Py & Qx)]
Ax[Ey(Px & Qy) v Ey(Py & Qx))]
Ax[(Px & Ey(Qy)) v (Ey(Py) & Qx)]
Ax[(Px v Ey(Py)) & (Px v Qx) & (Ey(Qy) v Ey(Py)) & (Ey(Qy) v Qx)]
Ax(Px) v Ey(Py)) & Ax(Px v Qx) & (Ey(Qy) v Ey(Py)) & (Ey(Qy) v Ax(Qx)]

Let s be a sentence in primary normal form. On evidence segment σ, we mark each universal component of s with a 1 if it is not yet refuted by σ, and we mark it 0 otherwise. We mark an atom or an existential component of s with a 1 if it is already verified by σ and we mark it 0 otherwise. Now, we mark s with a 1 if and only if its boolean valuation with respect to the markings just defined evaluates to 1. Call the result the *evidential evaluation of s with respect to* σ. It turns out that a complete theory in a $P_1F_0I_0$ language can always be axiomatized by one sentence, and the set of such sentences is recursive. Let τ be a tape of chosen primary normal forms of these complete axiomatizations. Now on evidence σ, conjecture the first sentence on τ whose evidential valuation with respect to σ is 1. This effective procedure solves the unrestricted theorizing problem for an arbitrary $P_1F_0I_0$ language in the EA sense. As an immediate corollary, it discovers, all at once, *all boolean concepts expressible in the language* . Of course, the procedure is not very elegant, and the search for tractable subcases could generate some interesting research in machine learning.

Extensions of the procedure of Proposition 1 are hemmed in tightly by the negative Propositions 7 through 9. Proposition 7 says that the method of Proposition 1 won't work for binary predicates even when we restrict quantifier prefixes to purely universal or purely existential ones. Proposition 8 shows that the same thing happens even when the added binary predicate is identity. The reason is roughly this. Identity and universal quantification can express upper cardinality bounds. For example, the hypothesis AxAy(x=y) is satisfied only in domains of cardinality no greater than one. The unrestricted problem for such a language includes domains of each finite cardinality, together with an infinite domain. By an argument similar to Gold's (Gold 1967), we can find, for any investigator ϕ that is assumed to solve the problem, a data presentation for the infinite domain on which ϕ is fooled into committing itself to each finite upper bound, so that it never converges to the truth in the infinite domain.

Proposition 9 tells us that the method of Proposition 1 fails when we add unary function symbols to the language; even when quantifier prefixes are restricted to purely universal or purely existential. The argument is similar to the preceding one, except that we force the would-be theorist to run a different gamut. Instead of an infinite sequence of upper cardinality bounds, the theorist must contend with an infinite sequence of possibilities of the form Ax(S(f(x))), Ax(S(ff(x)), Ax(S(fff(x)), and so on. For each sentence in this sequence, there is a countable structure in which it is true but its predecessors are all false. Let the problem consist of the set of all these structures, along with a structure M in which each such sentence is false. Now on the assumption that theorist ϕ can solve the problem in the EA sense, we construct a vicious data presentation for M on which ϕ commits itself, in sequence, to each sentence in the sequence, so ϕ does not converge on a data presentation for M.

Proposition 10 tells us that the intrinsic difficulty of the problem of Proposition 1 skyrockets when we add binary function symbols to the language, even when just one alternation between universal and existential quantifiers is permitted. This shows that even so small a change as adding a binary function symbol to the hypothesis language can overturn methodological intuitions that are sound in a less difficult inductive problem. The machine learning community must therefore be ever vigilant against unwarranted extrapolations of popular techniques.

Propositions 7 through 9 relied upon an infinite "con game" argument that does not work against AE solutions to the same problems. Propositions 4 and 5 assure us that the arguments of Propositions 7 and 8 do not extend to exclude AE solutions, for predicates of arbitrarily high arity. Proposition 3 assures us that the "con game" argument of Proposition 9 does not extend to exclude AE solutions.

The algorithm required in the proof of Proposition 4 converges piece-meal to the complete theory in an hypothesis language with predicates of any given arity, no function symbols, identity, and purely universal quantifier prefixes. The program is quite simple. Let τ be an infinite tape upon which the hypothesis language L is listed. At stage n in reading the evidence, the program examines the first n sentences written on the tape, deletes all of those that have counterinstances in the data, and conjectures the remainder. It is easy to see that any true hypothesis is eventually entailed by each conjecture. What is not so obvious is that for each false hypothesis, there is a time after which it is no longer entailed by any set of axioms conjectured by the program. But by a simple model-theoretic argument, one can show that any evidence that provides a counterinstance to a sentence s in this language also provides a counterinstance to some element of any set that entails this sentence. Hence, such a set will never be conjectured once a counterinstance to s has been seen in the evidence presentation.

The proof of Proposition 5 is similar, but easier. Enumerate the purely existential hypothesis language, and at stage n of reading the evidence, conjecture each of the first n hypotheses on the tape that that has an instance in the evidence. A false sentence can never have an instance, and hence is never added to the tape. Hence, only true sentences are conjectured, and true sentences can never collectively entail a falsehood. Hence, each truth is eventually entailed by all subsequent conjectures, and no falsehood is ever added, so the procedure solves the unrestricted theorizing problem for $P_nF_mI_1\Sigma_1$ in the AE sense.

It is interesting that the enumeration method of Proposition 5 works for function symbols in the case of existential quantifiers while the similar enumeration method of Proposition 4 does not work for function symbols in the case of universal quantifiers. The reason is that the presence of function symbols blocks the argument that any finite set of hypotheses entailing an hypothesis with a counterinstance in the data also contains an hypothesis with a counterinstance in the data.[3] This situation is partly remedied by the method of Proposition 6, which makes use of an undecidable consistency test. Hence it establishes only that $P_nF_mI_1\Pi_1$ is in [[AE]], and not that it is in [[AEe]]. Again, we enumerate the hypothesis langauge on an infinite tape. At stage n in reading the data, we cut off the initial segment of length n of this tape, and delete any hypothesis that has a counterinstance in the data, just as before. But now, instead of conjecturing the entire tape segment that results from this process, we we conjecture only the greatest initial segment of the segment that is consistent with the current data. It is this test that is (very) uncomputable, and the question whether there exists a computable way to solve the same problem is exactly Open Question number 1.

This method seems simple enough, but it steers a subtle course between the twin errors of dropping a true hypothesis infinitely often and adding a false hypothesis infinitely often. For suppose s is a false sentence in the hypothesis language. By some stage n, a counterinstance to it appears in the data. Thereafter, the data is inconsistent with any set of sentences that entails s, so no conjecture entailing s will ever be made again. But what about throwing out too much? Suppose s is a true sentence in the hypothesis language. Let s appear in position k on the hypothesis tape. Then by some stage n, counterinstances to all hypotheses prior to s will have appeared in the data. At this stage, the greatest initial segment of the hypothesis tape that is consistent with the data must include s, since all non-deleted hypotheses prior to it are true, and hence cannot be inconsistent with the data.

---

[3]This argument of hinges on being able to find, for any sentence false in a structure, a finite substructure of that structure in which the sentence is still false. But a structure for a language with function symbols may have no substructures at all, since functions are assumed to be total.

The method of Proposition 2 recognizes that a finite number of unary predicates induce a finite partition on a structure's domain. At stage n in reading the evidence, it calculates, for each of these partitions, the least number of objects it can contain, according to the data. It then conjectures that each partition has exactly the cardinality that is currently observed. If the structure is finite, it EA converges to the complete true theory of the structure. If the structure is infinite, it AE converges to the complete true theory, since its conjectures eventually entail each lower bound on the cardinality of an infinite partition cell, and each false upper bound on such a cell is eventually rejected forever.

Last, but not least, the discovery procedure of Proposition 3 may be of greatest practical interest to machine learning. Recall that the popular "precondition-postcondition rules" for operators fall into this category. Since we have universal quantifiers and function symbols, we again have to worry about later false conjectures entailing falsehoods that already have counterinstances in the data. But we also have mixed quantifiers, and only unary predicates and functions, which means that features of the method of Proposition 1 will also be involved. The procedure works like this: Form a tape of all primary normal form representatives of hypotheses in the hypothesis language. At stage n in the evidence presentation, consider the initial segment of this tape containing the first n hypotheses. For each hypothesis on this tape segment, put it in PASS if its evidential valuation on the current evidence is zero, and put it in FAIL otherwise. Now conjecture the greatest initial segment of the tape segment such that the elements of PASS that occur in this segment do not collectively entail any element of FAIL that occurs in this segment. This method works because the evidential valuation of a primary normal form sentence converges to its truth value as the evidence increases. Hence, for any sentence s in the enumeration, this sentence and all prior sentences eventually have their truth values settled correctly, and thereafter, this sentence is added to the conjecture if it is true, and is withheld if it is false. The entailment test between PASS and the elements of FAIL may be uncomputable. The question whether there is a clearly computable discovery function with the same performance is exactly Open Question number 2.

Finally, there are the strong negative results of Propositions 10, 11, and 12. Each of these propositions involves mixed quantifier prefixes in the hypothesis language. The proof technique of Proposition 11 is to construct a data presentation for a structure in which $\forall x \exists y \Phi(x,y)$ is true on which any theorist that can allegedly AE identify all countable structures for the language concludes that $\forall x \exists y \Phi(x,y)$ is false infinitely often. This can be done by showing the theorist an object b and by providing counterexamples to $\Phi(b,y)$ until he rejects $\forall x \exists y \Phi(x,y)$. Then we are free to exhibit a c such that $\Phi(b,c)$, along with lots of other pairs d,d' such that $\Phi(d,d')$ until the theorist is once again convinced of the hypothesis $\forall x \exists y \Phi(x,y)$. The proofs of Propositions 10 and 12 are by

simple reductions of Proposition 11.

5. Conclusion

The framework of this paper puts teeth into the truism that the hypothesis language helps determine the difficulty of the learning problem. The above framework, together with the results concerning it, provide a coarse but comprehensive picture of the relative difficulties of the discovery problems posed by hypothesis languages of various types. We have seen that mixed quantification, binary function symbols and binary predicates can combine to yield a very difficult learning problem, while hypothesis languages with unary predicates only give rise to fairly simple ones.

A practical consequence of this investigation is that learning techniques that work for easy problems should be viewed with great caution until they are shown to work in a broader context. But more importantly, the results in this paper show that a formal characterization of the relationship between hypothesis language is not only possible, but feasible. This particular survey may involve assumptions that are unnatural in various applications (e.g. the completeness and truth of the theory to be discovered, the completeness and truth of the data presentation, or the absence of complexity restrictions on the theorizing functions). These assumptions are not reasons to reject analysis. They are reasons for new and more detailed analyses of a broad range of different formal settings for machine learning. Only through such investigations can a clear grasp of the intrinsic difficulties of learning problems be determined, and such knowledge is essential if we are to distuinguish truly efficient methods from faster methods that solve easier problems in uninteresting ways.

Acknowledgements

I would like to thank Dan Osherson and Scott Weinstein for useful comments on results in this paper. I am also grateful for Clark Glymour's many essential contributions to this project.

References

Angluin, Dana and Smith, Carl H. (1982). *A Survey of Inductive Inference Methods*, Technical Report 250, Department of Computer Science, Yale University.

Carbonelll, J. and Gil, Y. (1987). Learning by Experimentation. In *Proceedings of the Fourth International Workshop on Learning* (pp. 256-266). Irvine, CA.

Gold, E. Mark (1967). Language Identification in the Limit. In *Information and Control.* Volume 10 (pp. 447-474).

Hilbert D. and Bernays, P. (1968). *Grundlagen der Mathematik I.* Berlin: Springer-Verlag.

Kelly, K. and Glymour, C. (in press). Convergence to the Truth and Nothing but the Truth.
Forthcoming in *Philosophy of Science*.

Osherson, D. and Weinstein, S. (1986). Identification in the Limit of First Order Structures.
In *Journal of Philosophical Logic*, Volume 15 (pp. 55-81).

Peirce, C.S. (1965). *The Collected Papers of Charles Sanders Peirce*. Vol 5.
Cambridge: Belknap Press.

Popper, K. R. (1963) *Conjectures and Refutations*.
New York: Harper and Row.

Putnam, Hilary (1963). 'Degree of Confirmation' and Inductive Logic.
In *The Philosophy of Rudalph Carnap*, Arthur Schilpp ed.
Lasalle, IL: Open Court Press.

Shapiro, Ehud Y. (1981). *Inductive Inference of Theories from Facts*.
Research Report 192, Department of Computer Science, Yale University.

Winston, Patrick Henry (1975). Learning Structural Descriptions from Examples.
In *The Psychology of Computer Vision*.
New York: McGraw Hill.