INDUCTIVE REASONING IN MATHEMATICS

F. Malloy Brown
Department of Artificial Intelligence
University of Edinburgh
Scotland

a:

Sten-Ake Tarnlund
Dept. of Information Processing & Computer Science
University of Stockholm,
Royal Institute of Technology, Sweden.

Abstract

We investigate several methods of inductive reasoning in the domain of difference equations, including the method of generalization with beliefs, the method of successive refinement, and temporal methods based on comparisons with previously solved problems.

1. Introduction

We begin a study of inductive reasoning in mathematics. Inductive reasoning in mathematics differs from inductive reasoning in the empirical sciences in that there is an ultimate test although not necessarily a decision procedure, which can be used to determine what is a correct induction. Namely, that if what is induced is true, that is: if it can be deduced by one's deductive system then it must be correct. Because of this relationship between induction and deduction one of our goals is to classify what instances of reasoning are inductive and what are deductive.

Our main goal however, is to create a taxonomy of feasible inductive methods. Such a goal is not unrelated to the work of the mathematician: G. Polya, [1967, 1968A, 1968B], it is just more detailed. In particular, we try to carry out the analysis of each method to a level detailed enough so as to be programmed. Also, where necessary, we relate the inductive methods to the level and capabilities of contemporary deductive systems. Due to this requirement for detail we shall restrict our attention to a particular, but significant domain.

1.1 Our Problem Domain

We consider the problem of trying to find by inductive reasoning a closed-form solution, that is an algebraic solution, to a recursive function. That is, from a set K of equations of the form*:
ψ_k (n,fn,f(n-h) ,f(n-2h) . . .)=0 or rather:

fn = ψ_k (n,f(n-h),f(n-2h),...)
we wish to find an equation of the form:
Vn fn - øn where f does not occur in ø.

For example, given the recursive equation for the Fibonacci function:
F(n+2) = F(n+1)+Fn
Fl = 1
Fo = 0
We would like to find a theorem of the form:
Vn Fn = øm
where øn is a sentence constructed from algebraic symbols such as numerals, plus (+), times (*), power ( ), minus (-), division (/), logarithm (ln), sine (sin) and cosine (cos).

Or as another example, given the recursive equations for the minimum number of moves that must be made in the Tower of Hanoi puzzle of n discs (a description of this puzzle may be found in Luger [19761.)
H(n+1) = 2(Hn)+l
HO = 0
we would like to find a closed form solution:
Vn Hn = øn.

1.2 A definition of "Inductive Reasoning"

Of course if we, or rather our program already "knew" a closed form solution øn for a recursive function such as F or H then we would not want to call the method by which that øn was produced: "inductive reasoning". The problem then of defining just which methods of producing øn are examples of inductive reasoning, and which are not - resides in the question as to what it means for a system to "know" something. For example if the system had;
Vn Hn = øn
explicitly stored as an axiom we would clearly say that the system knew that Vn Hn = øn. Furthermore, if the system could apply a sequence of items (i.e. axioms, lemmas, axiom schemas and lemma schemas, written say in LISP) which transformed Hn into øn, then if the system knew that each application of an item in that sequence produced a correct result, then again we would say that the system knew that (Vn Hn = øn) .

So far this is an omnipotent sense of knowledge: We know whatever is deducible. We would like to modify it by the further requirement of feasibility. That is, any sequence of application of items must not be so long as to exhaust the system's resources or our patience. In regard to feasibility, it is pointed out that we don't strictly speaking have to "know" that each application of an item is correct, (for after all, proofs from axioms only are very long), but rather all that is needed is the possibility of "knowing" that our items are correct in the sense of an extensible deductive system (Brown [1976A]), coupled with the "belief"* that our items are correct in at least the sense of never having been refuted (i.e. shown to be incorrect) (Popper[1968]) .

In summary then we will call any method of producing a closed-form solution øn, which is not "known" to be the closed-form solution, inductive reasoning. For example, enumeration of all possible algebraic functions or random guessing would count as inductive reasoning (albeit poor inductive

* Our convention for parenthesizing complex subexpressions is to place the left parenthesis before the function symbol as is done in EVAL-LISP. Thus we write (F n) and (a + b) not F(n) and (a) + (b).

* We speak of "knowledge" as true "belief". The point is that although we hope that our deductive system is sound, there does not seem to be anyway to absolutely guarantee this fact.

reasoning) because as items of a deductive system they would quickly be refuted by producing false results.

2. Elementary Inductive Methods

We first describe two basic methods of inductive reasoning, and then compare them.

2.1 The Method of Generalization with Beliefs

One popular theory of inductive reasoning (Meltzer [1969], Plotkin[1969, 1971],Popplestone [1969], Feldman[1969], Hardy[1976]) suggests that it is basically generalization.* Consider for example the Tower of Hanoi problem. Using the recursive equations we can deduce closed-form solutions for numerical instances of Hn.

HO = 0
HI = HO+1 = 2.04-1 = *1*
H2 - 2H1+1 = 2.1+1 = 3
H3 - 2H2+1 = 2.3+1 = 7
H4 = 2H3+1 =2.7+1-15
H5 - 2H4+1 = 2.15+1 = 31
H6 - 2H5+1 = 2.31+1 = 63

I suspect the reader has already induced a closed-form solution for Hn, but let us investigate how this might be done. From

H0=0 HI=1 H2=3 H3=7 H4=15 H5=31 H6=63

we wish to find**

Vn Hn = ?

First let us note that a least general generalixation method such as Plotkin [1969,1971] does not work because the least general generalization:

Vn Vm Hn = m is false.

Furthermore,even if we include an algebraic matching facility similar to that used in Feldman [1969] and Hardy [I976]***which allows us to try to relate the arguments of the Hn function to its values by adding,multiplying,subtracting,and dividing, we still would not obtain a solution. For example, adding one to the sequence of values of Hn gives:

HO+1=1 HI+1=2 H2+I=4 H3+I=8 H4+I=16 H5+I=32 H6+1=64

Vn Hn = ?

But still the remaining least general generalization is false.

So how can the closed form solution be induced? Let us suppose that the inductive system has available a belief that any function which produces the sequence of values 1, 2, 4, 8, 16, 32, for the arguments O, 1, 2, 3, 4, 5, is probably equal to the exponential function of base 2:

(Belief 1, 2, 4, 8, 16, 32 is probably 2^n)

Then given this belief a system might try to compare the sequence 1, 2, 4, 8, 16, 32 to the sequence of values given by the recursive function by applying various algebraic operations pairwise to the elements of each sequence. For example,in the case of Hn we might subtract from each element of the belief sequence, the corresponding element generated by Hn, obtaining

	1	2	4	8	16	32
-	0	1	3	7	15	31
	1	1	1	1	1	1

a sequence of ones. This would give us the knowledge that

$H0 = 2^0 - 1$
$H1 = 2^1 - 1$
$H2 = 2^2 - 1$
$H3 = 2^3 - 1$
$H4 = 2^4 - 1$
$H5 = 2^5 - 1$

and via our belief, we induce that:

$Hn = 2^n - 1$

which in fact is true.

There is, however, another inductive method which, other than for possibly the triggering of the belief that the closed form solution might involve an expenential function of base 2, does not really need to produce those instances of the Hn function. Nor does it need to go through any generalization steps. We call this method, the Method of Successive Refinement.

2.2 The .Method of Successive Refinement

The basic idea of the Method of Success Refinement is to make an initial (incorrect) guess as to what is the closed form solution, let a theorem prover try to prove this guess, and when it fails to prove it, go back and try to modify each branch of the protocol on which ^{n} had been produced. For example, let us suppose that we believe a solution to Hn might involve an exponential function to a power of two:

(Belief Hn involves 2^n)

In other words, let us first form the hypothesis that;

$Hn - 2^n$

and see what a deductive system will do to this expression. Inducting on n we get*:

$Hn = 2^n$

$HO = 2^0$ ← → $Hn = 2^n$ >- $H(n+1) = 2^{n+1}$
HO ■ 1   :
O ■ 1   :
¤   ■

Already we are in trouble as the deductive system shows that our hypothesis is false, on the base case. Can we modify our hypothesis to induce a better one? Analysing our proof we see that if the expression 0 = 1 were replaced by 0 = 1 - 1, then o and not n would be produced. This means that

HO ■ 1 would have to be HO = 1 - 1,
HO = 2 would have to be HO = 2 - 1, and
$Hn = 2^n$ would have to be $hn = 2^n - 1$.

Giving our new hypothesis to a deductive system we get:

* We consider analogy to be an essential component of the method of generalization as here described, and hence do not define a separate: method of analogy.

** V is our symbol for all.

*** In our case, however, we work directly on the numbers themselves rather than applying algebraic operations to the number of occurrences of symbols such as: the successor symbol.

* ■ is our symbol for true; a is our symbol for false, and >- is our symbol for implication. The left branch is the base of the induction, and the right branch is the induction step.

$$Hn = 2^n - 1$$

$$H0 = 2^0 - 1 \qquad (Hn)=2^n-1 \rightarrow H(n+1)=2^{n+1}-1$$

$$H0 = 1 - 1 \qquad 2^n=(Hn)+1 \rightarrow H(n+1)=2 \cdot 2^n-1$$

$$H0 = 0 \qquad H(n+1)=2((Hn)+1)-1$$

$$0 = 0 \qquad H(n+1)=2(Hn)+2-1$$

$$\blacksquare \qquad H(n+1)=2(Hn)+1$$

$$2(Hn)+1=2(Hn)+1$$

$$\blacksquare$$

which we deduce to be true.

We see then that we can obtain a proof by modifying a previous unsuccessful attempt at proving a theorem. This leads naturally to the question as to what is the space of all such modifications? Although, potentially this space may be quite large, we will only consider two simple modifications:

1) addition of a constant to an equation in order to make it true
2) multiplication of a constant to an additive part of an equation in order to make it true.

We have just seen an example of (1) in obtaining a solution to the Hanoi function. An example of (2) applied to the false equation

1 - /5 would be to multiply /5 by 1//5
and an example of (2) applied to an additive part of the false equation
0 - 1 + /5
would be to multiply the additive part 1 by - 1//5

Note that modifications are only made to the right hand side of an equation. The reason for this is that that side contains the closed-form solution whereas the other side merely contains the recursive function.

The reader will see in the method of successive refinement an echo of Meltzer's [1969] hypothesis that inductive reasoning is inverse deduction. But in our case, this is not so much a semantical hypothesis: if A>B then A may be induced from B, as it is a notion of search strategy in the sense that if B is deduced from A by applying the item* A -> B to A, then A is induced from B by applying the item in the opposite direction.

We have seen how the closed-form solution for Hn may be obtained from the belief that the solution involves an exponential function of base two. Note that this belief is stronger than what is necessary, and that the belief that some arbitrary exponential function a is involved suffices: That is, if 2 is replaced by a in the previous protocols, essentially the same result will be obtained.

2.3 Comparison of the two Methods

So far, we have found that simple generalization methods are not sufficient and that sophisticated beliefs seem to be used in inductive reasoning. We have cast doubt on the suggestion that inductive reasoning is only generalization by describing another technique, the method of successive refinement, which seems to be more powerful (because it needs a weaker belief to produce the solution).

One advantage of the generalization method is the possibility of triggering the 2^n belief by an algebraic matching of the 1, 2, 4, 8, 16, 32, sequence to the sequence generated by the Hanoi function: Hn. A second advantage is that it is quite easy to explain how the belief, that a function whose initial instances are 1, 2, 4, 8, 16, 32, is the exponential of base 2:
(Belief 1, 2, 4, 8, 16, 32 is probably 2^n) n
was produced by simply instantiating n in 2 to successively 0, 1, 2, 3, 4, 5 whereas it does not seem quite as easy to explain the production of the initial hypothesis that Hn involved some exponential function:
(Belief Hn involves a^n)
Are we for example to believe that the closed-form solution of every recursive function involves an exponential function or what?

Although it is easier to explain the origins of a belief of the form:
(Belief 1, 2, 4, 8, 16, 32 is probably 2^n)
than of the form:
(Belief Hn involves a)
we note that if a recursive function is not actually equal to a simple algebraic function such as 2 then it is quite improbable that the necessary belief of this first form could be available to be used in obtaining the closed-form solution. We will now give an example of such a function and suggest that the closed-form solutions of most recursive functions are of this character. This example is interesting because it supports the view that inductive reasoning based only on the method of generalization even with sophisticated beliefs, is not sufficient, and that other methods are also needed.

Our example is the Fibonacci function. Using its recursive equations we deduce:
FO = 0
F1 = 1
F2 = FI + FO = 1 + 0 = 1
F3 = F2 + FI = 1 + 1 = 2
F4 = F3 + F2 = 2 + 1 = 3
F5 = F4 + F3 - 3 + 2 = 5
F6 = F5 + F4 = 5 + 3 = 8
F7 = F6 + F5 = 8 + 5 = 13
Thus from
F0=0 FI=I F2-1 F3=2 F4=3 F5=5 F6=8 F7=13
we wish to find:

Fn = ?

But there is no simple algebraic function which produces an initial sequence anything like 0, 1,1, 2, 3, 5, 8, 13. In other words there could not be any belief of the form:
(Belief $a_1....a_m$ is probably a)

which would be helpful in solving this problem. For after all we cannot expect to have beliefs about every algebraic function in our system; only the simple ones. We can see that the closed-form solution is not simple, by actually exhibiting it. Concurrently we will take the opportunity to show how this closed-form solution could be obtained by the method of successive refinement using the belief that the solution involves two exponential functions:
(Belief Fn involves a , b)
where a # 0 can b # 0.

From this belief we form an initial hypothesis that:
$Fn = a^n + b^n$
and see what our deductive system will do to this expression. Inducting on n twice, because Fn

*<-> is our symbol for iff (if and only if).

does not recurse on Fn+1, we obtain;

$\exists a \exists b \forall n \quad Fn = a^n + b^n$

$F0=a^0+b^0$	$Fn=a^n+b^n \rightarrow F(n+1)=a^{n+1}+b^{n+1}$
$F0=1+1$	$a^n=Fn-b^n \rightarrow F(n+1)=aa^n+b^{n+1}$
$F0=2$	$F(n+1)=a(Fn-b^n)+b^{n+1}$
$0=2$	$F(n+1)=a(Fn)+(b-a)b^n$
□	

$F(n+1)=aFn+(b-a)b^n \rightarrow F(n+2)=aF(n+1)+(b-a)b^{n+1}$

$(b-a)b^n=F(n+1)-aFn \rightarrow F(n+2)=aF(n+1)+(b-a)b^n b$

	$F(n+2)=aF(n+1)+F(n+1)-aFn)b$
$F1=aF0+(b-a)b^0$	$F(n+2)=aF(n+1)+bF(n+1)-abFn$
$F1=a\cdot 0+(b-a)\cdot 1$	$F(n+2)=(a+b)F(n+1)-abFn$
$F1=0+b-a$	$F(n+1)+Fn=(a+b)F(n+1)$
$F1=b-a$	$(a+b-1)F(n+1)-(a\cdot b+1)Fn=0$
$1=b-a$*	$a+b-1=0 \wedge a\cdot b+1=0$
$1=\frac{1-\sqrt{5}}{2}-\frac{1+\sqrt{5}}{2}$	
$1=\frac{2\sqrt{5}}{2}$	$a+b-1=0 \qquad a\cdot b+1=0$
	$a=1-b \quad \ldots \quad (1-b)\cdot b+1=0$
$1=\sqrt{5}$	■
□	$a=\frac{1+\sqrt{5}}{2} \qquad b-b^2+1=0$
	$b^2-b-1=0$
	$b=\frac{1-\sqrt{5}}{2}$
	■

The induction step seems to be true, but the two base cases are false. Can we modify our hypothesis to induce a better one?

Analysing the Fl base case we see that if the in (1 = /5) had been divided by √5 then ■ would have been produced. This means that the base case would have had to have looked something like:

Fl=aFO+((b-a)//5)b
Fl=a-0+((b-a)/5)-l
Fl-0+(b-a)/√5
Fl=(b-a)//5
1= (b-a)/√5
1= ((l-√5)/2-(l+√5)/2)/√5
1= 2√5/2√5
1-1
D

* In contemporary deductive systems (Brown [1976B, 1977], Boyer [1975]) the equality or unification items would end up replacing all occurrences of say "a" by b-1. A system used in successive refinement should not do this because if $Fn = a^n + b^n$ is false a might not equal b - 1, and then we would not want to propagate this falsity into the branch of the proof which is the induction step which may after all be true as in this example. The reason we prefer to have n derived on the branch of the proof which is the induction base rather than the induction step is that it is a simpler branch and will be easier to analyse why it produced n, in order to induce a hypothesis.

Working upwards we see that the five previous lines in the proof must have been something like:

$Fn=a^n+b^n/\sqrt{5}$

$Fn=a^n+b^n/\sqrt{5} \rightarrow F(n+1)=a^{n+1}+b^{n+1}/\sqrt{5}$

$a^n=Fn-b^n/\sqrt{5} \rightarrow F(n+1)=a\cdot a^n+b^{n+1}/\sqrt{5}$

$F(n+1)=a(Fn-b^n/\sqrt{5})+b^{n+1}/\sqrt{5}$

$F(n+1)=aFn+((b-a)/\sqrt{5})b^n$

? ■ ?

So far we have been able to induce a new hypothesii which will probably make the Fl-base branch of the proof result in . But, does this change the result of we had previously obtained in the induction step branch of the proof? Since inducting on $F(n+1)=aFn+(b-a)/\sqrt{5}-b^n$ gives:

$Fn+1=aFn+(b-a)/\sqrt{5}\,b^n \rightarrow Fn+2=aFn+1+(b-a)/\sqrt{5}\,b^{n+1}$

$(b-a)/\sqrt{5}\,b^n=Fn+1-Fn \rightarrow Fn+2=aFn+1+(b-a)/\sqrt{5}\,b^n\,b$

Fn+2=aFn+l+(Fn+l-aFn)b

and since the third line is identical to our previous third line we see that the induction step branch of our proof will still result in ■.

Of course we don't actually have to analyse our proofsteps downward, as we could simply apply our deductive system to such a step and see what it does. For example, to see that $Fn+1 = aFn +((b-a)// 5)b^n$ results in = all we need do is to apply our deductive system to it.

We have now produced a modification which results in a on both the Fl base, and induction step branches of our proof. Only the FO base step remains. The FO base branch resulted in o, so we know that some sort of modification is what was needed to make the FO base branch result in ■: We first check to see if the modification induced from the Fl base branch will suffice to make this branch result in :

$Fn = a^n + b^n/\sqrt{5}$

$F0 = a^0 + b^0/\sqrt{5}$

$F0 = 1 + 1/\sqrt{5}$

$0 = 1 + 1/\sqrt{5}$

□

So there is still a problem. If, however, $0 = 1 + 1/\sqrt{5}$ were $0 = 1/\sqrt{5} - 1/\sqrt{5}$ then $0 = 0$, ■ would have been deduced. For this to be the case $F0=1+1/\sqrt{5}$ would have to be $F0=1/\sqrt{5} - 1/\sqrt{5}$

$F0=a^0+b^0/\sqrt{5}$ would have to be $F0=a^0/\sqrt{5} - b^0/\sqrt{5}$ and finally:

$Fn=(a^n-b^n)/\sqrt{5}$ would have to be $Fn=a^n-b^n/\sqrt{5}$ and then our deductive system would easily be able to deduce that this new hypothesis:

$\exists a \exists b \forall n\ Fn = \frac{a^n - b^n}{\sqrt{5}}$

is true.

Replacing a and b by the algebraic terms that we have found them to be in our proof we find that the closed form solution for the Fibonacci function is:

$$\forall n\ Fn = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}}$$

Thus we see that although it is plausible that we might be able to induce this solution by the inductive method of successive refinement, it is not very plausible that we could have induced this solution by the generalization method. The reason for this, as will be recalled, is because the generalization method would require prior beliefs that:

(Belief:1,1.618,2.618,4,236,6.854,11.090 is probably $\left(\frac{1+\sqrt{5}}{2}\right)^n$

(Belief:1,-.618 .382 -.236 .146 -.090 is probably $\left(\frac{1-\sqrt{5}}{2}\right)^n$

And it simply does not seem plausible that such beliefs would be available.

2.4 The Supplementary Method of Existential Functions

We have one final point to make about the Fibonacci example: If we had a slightly more sophisticated belief as to what the closed-form solution might be, then our deductive system would be able to obtain the solution in a very direct manner. Let us suppose that we have the belief that the closed-form solution of the Fibonacci involves a linear combination of two non-zero functions:

(Belief Fn involves $\alpha\ a_n + \beta b^n$)

Then the hypothesis is: $\exists\alpha\exists\beta\exists a\exists b\ \forall n\ Fn = \alpha a^n + \beta b^n$

A solution is obtained by inducting twice on n:

$$\exists\alpha\exists\beta\exists a\exists b\ \forall n\ Fn = \alpha a^n + \beta b^n$$

$$F0 = \alpha a^0 + \beta b^0$$
$$F0 = \alpha \cdot 1 + \beta \cdot 1$$
$$F0 = \alpha + \beta$$
$$0 = \alpha + \beta$$
$$\alpha = -\beta$$
$$(\alpha = 1/\sqrt{5})$$

$$Fn = \alpha a^n + \beta b^n \rightarrow F(n+1) = \alpha a^{n+1} + \beta b^{n+1}$$
$$\alpha a^n = Fn - \beta b^n \rightarrow F(n+1) = \alpha a^n a + \beta b^{n+1}$$
$$F(n+1) = (Fn - \beta b^n)a + \beta b^{n+1}$$
$$F(n+1) = aFn + (b-a)\beta b^n$$

$$F(n+1) = aFn + (b-a)\beta b^n \rightarrow F(n+2) = aF(n+1) + (b-a)\beta b^{n+1}$$
$$(b-a)\beta b^n = F(n+1) - aFn \rightarrow F(n+2) = aF(n+1) + (b-a)\beta b^n b$$
$$F(n+2) = aF(n+1) + F(n+1) - aFn)b$$
$$F(n+2) = aF(n+1) + bF(n+1) - abFn$$
$$F(n+2) = (a+b)F(n+1) - abFn$$
$$F(n+1) + Fn = (a+b)F(n+1) - abFn$$
$$(a+b-1)F(n+1) - (a \cdot b+1)Fn = 0$$
$$a+b-1=0 \wedge a \cdot b+1=0$$

$$F1 + aF0 + (b-a)\beta b^0$$
$$F1 = a \cdot 0 + (b-a)\beta \cdot 1$$
$$F1 = 0 + (b-a)\beta$$
$$F1 = (b-a)\beta$$
$$1 = (b-a)\beta$$
$$\beta = \frac{1}{b-a}$$
$$\left(\beta = \frac{1}{\frac{1-\sqrt{5}}{2} - \frac{1+\sqrt{5}}{2}}\right)$$
$$\left(\beta = \frac{2}{-2\sqrt{5}}\right)$$
$$\left(\beta = -\frac{1}{\sqrt{5}}\right)$$

$$a+b-1=0 \qquad a \cdot b+1=0$$
$$a = 1-b \ldots (1-b) \cdot b+1$$
$$\left(a = \frac{1+\sqrt{5}}{2}\right)$$
$$b - b^2 + 1 = 0$$
$$b^2 - b - 1 = 0$$
$$b = \frac{1-\sqrt{5}}{2}$$

This technique of using existential functions has been used by Bibel [1976](in our case constants: a,b,α,β).Because it is so powerful that we shall raise it to the status of a third inductive method which we shall call: The supplementary method of existential functions. It should be noted that this technique is no substitute for the more general method of successive refinement, for its power depends on knowledge possessed by the deductive system. In the protocols using the method of existential functions we have assumed the ability to solve equations of the form:

f - 0

for an arbitrary existential constant. This knowledge was not assumed when not using this method. We shall see more of this method in section 3.

2.5 Summary

In summary then we have described two basic inductive methods and one supplementary method; and have shown that the effectiveness of each method depends on the availability of particular beliefs. We have suggested that the beliefs needed by the generalization method would only be available if the recursive function is equal to some very simple algebraic function, such as is the case with the Tower of Hanoi function. For most other recursive functions such as the Fibonacci functions we have suggested that the beliefs needed by the generalization method would not be available.

In section 3 we describe how the beliefs needed by the method of successive refinement might be automatically produced.

3. Temporal Inductive Methods

We now describe methods of inductive reasoning based on information obtained from the solutions of previously solved problems. Such methods are called temporal inductive methods. There are two temporal inductive methods, distinguished by the type of information on which they are based. The first temporal method is based solely on information obtained from the theorem which expresses a previously solved problem, whereas the second temporal method is based on information obtained from the proof of a previously solved problem.

Referring back to section 2.5 we shall see in section 3.2 how the second temporal method may be applied to producing the kind of beliefs needed by the method of successive refinement.

3.1 The Temporal Method based on Theorems

The basic idea of the temporal method based on theorems is for any given problem F to try to find a similar problem H which has already been solved, and then to use something similar to the solution for H as a hypothesis for the solution of F.

For example let H be the Tower of Hanoi problem whose solution we discovered in Section 2, and let F be the Fibonacci function whose solution we are trying to find. The theorem which expresses the solution to the Hanoi problem is then:

($\forall n\ H(n+l)=2(Hn)+lAHO0) \rightarrow Hn=2^n-1$

and the (as yet incomplete) theorem which expresses

the Fibonacci problem is then:

$$(\forall n F(n+2)=F(n+1)+Fn \wedge F1=1 \wedge F0=0) \rightarrow Fn=?$$

We now try to compare these two theorems in order to produce a guess as to what ? should be:

$$(\forall n\ H(n+1)=2Hn + 1 \quad \wedge\ H0 = 0) \rightarrow Hn=2^{n-1}$$
$$\forall m\ F(m+2)=F(m+1) + Fm \quad F1 = 1 \wedge F0 = 0 \rightarrow Fm = ?$$

We see that there is no reasonable analogy that we can make which will help us to solve F. For example, if we form the analogy that m is n - 1 from (Hn+1 Fm+2), then we might conjecture that:

$Fm=2^{m-1}$ -1

which is false. In any case, so much syntax has been left unexplained by this analogy (for example the 2. , +1, HO = 0, +Fm, and Fl = 1) that we can have no confidence in this conjecture anyway.

Furthermore even if our generalization system were sophisticated enough to rewrite the H theorem as the equivalent theorem:

(VnH(n+2)=2H(n+l)+l 'HI«l^HO=0) + $Hn=2^n$-1

(VroF(m+2)-F(ra+l)+Fm / Fl-1 > FO=0) + Fm=?

The closest analogy would still not explain the 2*, +1, and +Fm.

The problem here is that H and F are so dissimilar that no immediate analogy can be made.

3.2 Temporal Method based on Proofs

The basic idea of the temporal method based on proofs is as follows:

1) First using the temporal method based on theorems find a problem H which is similar to the problem F you are trying to solve and conjecture a solution for F.

2) Using the deductive system produce the proof of H and a protocol of F with that hypothesis.

3) Finally compare the proof with the protocol to find out why no solution was obtained, and form a new hypothesis.

Suppose, now that we are trying to find a solution to the Fibonacci function F, given that we already know the solution to the Tower of Hanoi function H. We shall use the proof of the solution of H which is given as described in section 2.2, starting with:

3a Vn Hn = a^n- 1

The hypothesis for the solution of F is

3a Vn Fn - a^n-l

(We could throw in a few existential constants, for example: Fn = a.a^n -3, but it won't make any difference at this stage.)

Using our deductive system we obtain the following protocol:

$$\exists a\ \forall n\ Fn=a^n-1$$

$$F0=a^0-1 \qquad Fn=a^n-1 \rightarrow F(n+1)=a^{n+1}-1$$
$$F0=1-1 \qquad a^n=(Fn)+1 \rightarrow F(n+1)=a\cdot a^n-1$$
$$F0=0 \qquad F(n+1)=a\cdot((Fn)+1)-1$$
$$0=0 \qquad F(n+1)=a(Fn)+a-1$$
$$\blacksquare$$

$$F1=aF0+a-1 \qquad F(n+1)=aFn+a-1 \rightarrow F(n+2)=aF(n+1)+a-1$$
$$F1=a\cdot 0+a-1 \qquad a(Fn)+1=F(n+1)+1 \rightarrow F(n+2)=aF(n+1)+a-1$$
$$F1=0+a-1 \qquad a=\frac{F(n+1)+1}{(Fn)+1} \rightarrow F(n+2)=a(F(n+1)+1)-1$$
$$F1=a-1 \qquad F(n+2)=\frac{((Fn+1)+1)^2}{(Fn)+1}-1$$
$$1=a-1$$
$$2=a \qquad \vdots$$

We now compare this protocol with the proof of Hn. We see that in both cases the induction principle was first applied. We then see that the steps in the base case of proof and protocol were the same, and that the first few steps on the induction step branch of the proof and protocol were the same. In fact we find that they differ exactly where H(n+1) is replaced by 2(Hn)+l

$$H(n+1)=a(Hn)+a-1 \qquad F(n+1)=a(Fn)+a-1$$
$$\downarrow$$
$$2\cdot H(n+1)=a(Hn)+a-1 \qquad F1=aF0+a-1$$
$$F(n+1)=a(Fn)+a-1 \rightarrow F(n+2)=a(Fn)+1+a-1$$

This then is the problem. On F the theorem prover inducts because it can't recurse, because F recurses on n+2 not n+1, whereas H immediately recurses to obtain a solution.

So granted that F recurses on n+2 and that the theorem proven inducts twice the question, is: why is not a solution obtained after the second induction? Let us compare the steps beginning with the second induction in F to the steps in Hn:

3a Vn Hn = a^n - 1 3a Vn. F(n+1) = aFn+a - 1

We see that Hn involves some term "a" exponentiated by the induction variable n. Comparing Hn to Fn+1 this leads us to suspect that Fn+1 might also involve some term exponentiated by the induction term n+1. Thus we would hypothesize that Fn+1 involves some term b or rather than Fn involves some term b . The reader should bear in mind that the hypothesis is not a for this a would clash with the bound variable a already occuring in the above Fibonacci expression. The point is that the "a" in the Hanoi expression and "a" in the Fibonacci expression are two distinct bound variables, which do not necessarily refer to the same number.

There are two other indications that Fn involves a term of the form b . Inducting on n in each ca^se we get respectively the base cases:

$$H0 = a^0 - 1 \qquad F1 = aF0 + a - 1$$
$$H0 = 1 - 1 \qquad F1 = a\cdot 0 + a - 1$$
$$H0 = 0 \qquad F1 = 0 + a - 1$$
$$0 = 0 \qquad F1 = a - 1$$
$$\blacksquare \qquad 1 = a - 1$$
$$2 = a$$

Note that in the proof of H a skolem function disappears by having a replaced by 1 whereas in the protocol of F no skolem function disappears in this manner. Here then is a second indication that a term of the form b or rather b is needed in the Fibonacci protocol.

Finally, by comparing the induction steps of H and F we note that in H the equality item replaced a by an expression, whereas in F it merely replaced "a" by an expression. Here then is a third indication that Fn involves an expression of the form b (where b is not necessarily equal to a).

$$F(n+1)=a(Fn)+a-1 \rightarrow F(n+2)=aF(n+1)+a-1$$
$$a(Fn)+1=F(n+1)+1 \rightarrow F(n+2)=aF(n+1)+a-1$$
$$a=\frac{(F(n+1)+1)}{(Fn)+1} \rightarrow F(n+2)=a(F(n+1)+1)-1$$
$$Hn=a^n-1 \rightarrow H(n+1)=a^{n+1}-1 \qquad F(n+2)=\frac{((Fn+1)+1)^2}{(Fn)+1}-1$$
$$a^n=(Hn)+1 \rightarrow H(n+1)=a\ a^n-1 \qquad \vdots$$
$$H(n+1)=a\ ((Hn)+1)-1$$
$$\vdots$$
$$\blacksquare$$

Thus, for three reasons we are lead to the belief that Fn involved some term b where b is not necessarily a. We now go back and modify our original

belief: that Fn involves a :
(Belief Fn involves a)
to the new belief that Fn involves both a and b :
(Belief Fn involves a + b)
And this new belief, as we have already seen in section 2.3, sufficies to find the closed-form solution to the Fibonacci function.

3.3 Summary

In summary we have described two temporal inductive methods, one based on comparing the statement of a problem with the statement of a similar previously solved problem, and the other based on comparing the unsuccessful attempts to solve a problem with the proof of an analogous problem. In either case we have found that inductive reasoning depends on the ability to make use of previous solutions.

We have also seen how the type of belief needed to apply the method of succession refinement to the Fibonacci problem might be generated.

4. Conclusion

We have investigated and described several feasible methods of inductive reasoning which are useful in the domain of solving difference equations. We stress that these methods use no mathematical knowledge in terms of lemmas, strategies or rules other than that which would be available to a simple algebra theorem prover which has only the following rules*.

1) Rules to simplify algebraic expressions.
2) The logical rules needed to put formulas in conjunctive and disjunctive normal form.
3) The two equality rules:
   $(\forall x\ x = y \rightarrow \phi x) \leftrightarrow \phi y$
   $(\exists x\ x = y \wedge \phi x) \leftrightarrow \phi y$
4) A mathematical induction principle.
5) Rules to equate the coefficients of various types of expressions. (This ability is needed by the method of existential functions)

Of course, if we were to allow the use of sophisticated mathematical techniques we could easily solve the problems we have considered in this paper. In particular, we could implement a theorem prover based either on the rules of the finite difference calculus (Spiegel [1971], Jordan [1965]), or on the method of generating functions (Lin [1968], Bekenbach [1964], Knuth [1969], Jordan [1965]).

But this is just what we do not want to do,for this would not explain how the mathematical knowledge embedded in these techniques was actually created. In our view of mathematics, we see that this knowledge is basically created by trying to solve a number of problems using only one's general inductive abilities and the current status of one's deductive abilities. Then one generalizes and organizes the methods that one has found useful in solving these problems into some sort of deductive calculus, such as for example the theory of finite difference equations or the method of generating functions. One then adds this calculus to one's deductive abilities, henceforth making what were difficult problems quite easy, thus allowing one to try to solve problems in more difficult domains by repeating this process.

* These rules are embedded in an algebra theorem prover which one of the authors (Brown) implemented about a year ago.

Acknowledgements
We thank Professor B. Meltzer for criticizing a draft„of this paper and for arranging for Dr. Tarnlund to visit the Department of Artificial Intelligence so that we could collaborate on this research.

References

Beckenbach,E.F.(editor) Applied Combinatorial Mathematics, Univ. of Cal., Eng. and Physical Sciences Extentions Series, John Wiley & Sons, Inc. 1964.
Bibel,W. "Synthesis of Strategic Definitions„and their control", Report 7610. Tech. Univ. Munchen, 1976.
Boyer,R.S., Moore,JS. "Proving Theorems about LISP Functions", Proceedings IJCAI3 and J. of ACM,1975.
Brown,F.M. "The Role of Extensible Deductive Systems in Mathematical Reasoning", Proceedings of AISB Summer Conference, Edinburgh, 1976A.
Brown,F.M. "Doing Arithmetic without Diagrams", to appear in Jnl. of Artificial Intelligence, also DAI Research Report No. 16A,1976B.
Brown,F.M. "Towards the Automation of Set Theory and its Logic", DAI Research Report No. 34,1977.
Feldman, J., et al. "Grammatical Complexity and Inference", Tech. Report No. CS125,Stanford AI Prefect, Stanford Univ., 1969.
Hardy,S. "Synthesis of LISP Functions from Examples" Proceedings of AISB Summer Conference, 1976.
Jordan,C. Calculus of Finite Differences, Chelsea Publishing Co., New York, 17.Y. 1965, Library of Congress Catalog Card No.58-27786.
Knuth,D.E. Fundamental Algorithms: The Art of Computer Programming, Vol.1., p-86,Addison Wesley Publishing Company, Inc., 1969.
Lin,C.L. Introduction to Combinatorial Mathematics, McGraw-Hill Inc., 1968.
Luger,G. "The Use of the state space to record the behavioral effects of subproblems and symmetries in the Tower of Hanoi Problem",Int.J.Man Machine Studies, Vol.8, p 411-421, 1976.
Meltzer,B. "Power Amplification for Theorem-Proving, Part 2: A hypothetics deductive approach to programming induction". MI5,1969.
Plotkin,G.D. "A note on Inductive Generalization", MI5, 1969.
Plotkin,G.D. "A Further Note on Inductive Generalization", MI6, 1971.
Polya,G. Mathematical Discovery, Vol. I,II, John Wiley & Sons, Inc. 1967.
Polya,G. Induction and Analogy in Mathematics, Mathematics and Plausible Reasoning,Vol.I.Oxford University Press, London, 1968A.
Polya,G. Patterns of Plausible Inference, Mathematics and Plausible Reasoning, Vol.11. Oxford University Press, London, 1968B.
Popper,K.R. The Logic of Scientific Discovery , Hutchinson, 3rd edition, 1968.
Popplestone,R.J. "An Experiment in Automatic Induction", MIS, 1969.
Spiegel,M.R. Finite Differences and Difference Equations. McGraw-Hill Inc., 1971.