



# THE UNIVERSITY *of* EDINBURGH

## Edinburgh Research Explorer

### **Poof Analysis: A technique for Concept Formation**

**Citation for published version:**

Bundy, A 1985, 'Poof Analysis: A technique for Concept Formation' Proceedings of AISB-85.

**Link:**

[Link to publication record in Edinburgh Research Explorer](#)

**Document Version:**

Author final version (often known as postprint)

**Published In:**

Proceedings of AISB-85

**General rights**

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



# 'POOF' ANALYSIS— A TECHNIQUE FOR CONCEPT FORMATION

**Alan Bundy** Department of Artificial Intelligence,  
Edinburgh University, Edinburgh EH1 2QL, U.K.

## ABSTRACT

We report the discovery of an unexpected connection between the invention of the concept of uniform convergence and the occurs check in the unification algorithm. This discovery suggests the invention of further interesting concepts in analysis and a technique for automated concept formation. Part of this technique has been implemented.

The discovery arose as part of an attempt to understand the role of proof analysis in mathematical reasoning, so as to incorporate it into a computer program. Silver (1986) and Mitchell (1983) have investigated the automatic analysis of model proofs in order to extract and learn knowledge about controlling search, including the knowledge of new concepts. We focus on the analysis and correction of faulty proofs or 'poofs'\* especially where that correction involves the invention of new mathematical concepts.

A classic example of where the analysis of a poof leads to a new concept is the invention, by Weierstrass, Seidel, Cauchy and others, of uniform convergence as a result of an analysis of Cauchy's poof that the limit of a

\* A 'poof', according to one of my mathematics lecturers, is a proof with something vital missing.

convergent series of continuous functions is itself continuous. The correction consists of substituting in the theorem the new concept of 'uniformly convergent' for 'convergent'. We will investigate this example. The bug in Cauchy's poof is a violation of the occurs check. This observation suggests a technique for automatically correcting the proof and this leads to the concept of uniform convergence. The correction technique involves search. Other branches in the search space lead to alternative corrections and alternative concepts, one of which seemed interesting to me and turns out to have been discovered by the mathematician, Ascoli.

## 1. INTRODUCTION

Lakatos (1976) emphasizes the importance of poof analysis for discovering hidden lemmas and hence correcting poofs, defining new concepts, generalizing theories, etc. For instance, he relates how the concept of a polyhedron was defined and refined by analysis of poofs of Euler's theorem and how the concept of uniform convergence was defined by analysing Cauchy's poof that the limit of a convergent series of continuous functions is itself continuous. Poof analysis ensures that the new concepts it suggests will be useful, because they will help change a poof into a proof. Thus a poof analyser provides information to guide the process of concept formation, which is otherwise an explosive one. Such guidance can be provided by heuristic search based on 'interestingness' values (Lenat, 1982), but human concept formation is often problem driven, and poof analysis offers an alternative, problem driven, concept formation technique.

Such a poof analyser could also be an important ingredient of an artificial mathematics teacher, i.e. a program which reads student's solutions and discovers errors. DEBUGGY, (Burton, 1982), is an example of such a diagnosis program in the area of elementary arithmetic.

As a prelude to building such a poof analyser we investigate the example discussed by Lakatos—the invention of uniform convergence from an analysis of Cauchy's poof. This lends itself to formal analysis more easily than the poofs of Euler's theorem. We first give the informal poof and then translate it into a resolution/paramodulation poof in order to locate the bug. This turns out to be a missing occurs check in a unification. We discuss how the poof can be fixed up by re-ordering quantifiers to remove the dependencies that caused the unification failure. The effect is to replace 'convergent' by 'uniformly convergent' in the hypothesis of the theorem. We discuss alternative, systematic ways of fixing the poof leading to other concepts. Then we describe the partial automation of the process. Finally, we discuss how the technique may be used to disambiguate the meaning of a sentence during natural language understanding.

## 2. THE INFORMAL POOF

First the 'gist of Cauchy's proof' that the limit of a convergent series of

continuous functions is itself continuous, quoted, slightly modified, from Lakatos (1976).

Let  $f(x) = \sum_{i=0}^{\infty} f_i(x)$  be a convergent series of continuous functions and, for any  $n$ , define  $s_n(x) = \sum_{i=0}^n f_i(x)$  and  $r_n(x) = \sum_{i=n+1}^{\infty} f_i(x)$ .

Given any  $\varepsilon > 0$ :

there is a  $\delta$ , such that for any  $b$ , if  $|b| < \delta$ , then

$$|s_n(x+b) - s_n(x)| < \varepsilon$$

(there is such a  $\delta$  because of the continuity of  $s_n(x)$ ): (i)

there is an  $M$ , such that  $|r_n(x)| < \varepsilon$  for all  $n \geq M$

(there is such an  $M$  because of the convergence of  $\sum_{i=0}^{\infty} f_i(x)$ ): (ii)

there is an  $M'$ , such that  $|r_n(x+b)| < \varepsilon$  for all  $n \geq M'$

(there is such an  $M'$  because of the convergence of  $\sum_{i=0}^{\infty} f_i(x+b)$ ): (iii)

Hence we can infer that:

$$\begin{aligned} |f(x+b) - f(x)| &= |s_n(x+b) + r_n(x+b) - s_n(x) - r_n(x)| \\ &\leq |s_n(x+b) - s_n(x)| + |r_n(x)| + |r_n(x+b)| \\ &< 3\varepsilon, \text{ for all } b < \delta \end{aligned} \quad \text{Q.E.D.}$$

This kind of semi-formal, 'mathematese', is typical of the notation used in human proofs. In order to locate the bug automatically it is necessary to translate it into a formal notation which can be proof checked. We chose a resolution/paramodulation type proof for this, although other types would be equally suitable.

### 3. THE FORMAL POOF AND THE IDENTIFICATION OF THE BUG

In this formal version we use the PROLOG convention that variables begin with an upper case letter and constants with a lower case letter. When Skolemizing a variable we preserve the name, but demote the first letter from upper to lower case.

The definition of convergence for  $\sum_{i=0}^{\infty} f_i(x)$  is:

$$\forall X \forall E > 0 \exists M \forall N \geq M \left| \sum_{i=N+1}^{\infty} f_i(x) \right| < E \quad (\text{iv})$$

and the definition of continuity for  $f(x)$  is:

$$\forall X \forall E > 0 \exists \Delta > 0 \forall B \left| B \right| < \Delta \rightarrow |f(X+B) - f(X)| < E \quad (\text{v})$$

Thus the translations of (i), (ii) and (iii), to clausal form are:

$$E > 0 \rightarrow \delta(E, X, N) > 0 \quad (\text{vi})$$

$$E > 0 \ \& \ |B| < \delta(E, X, N) \rightarrow |s(N, X+B) - s(N, X)| < E \quad (\text{vii})$$

$$E > 0 \ \& \ N \geq m(E, X) \rightarrow |r(N, X)| < E \quad (\text{viii})$$

$$E > 0 \ \& \ N \geq m(E, X + B) \rightarrow |r(N, X + B)| < E \quad (\text{ix})$$

from which we can infer:

$$\begin{aligned} E > 0 \ \& \ |B| < \delta(E, X, N) \ \& \ N \geq m(E, X) \ \& \ N \geq m(E, X + B) \\ & |f(X + B) - f(X)| < 3.E \end{aligned} \quad (\text{x})$$

The goal is to prove  $f(X)$  continuous, which after negation and translation to clausal form gives:

$$\rightarrow \varepsilon > 0 \quad (\text{xi})$$

$$\Delta > 0 \rightarrow |b(\Delta)| < \Delta \quad (\text{xii})$$

$$\Delta > 0 \ \& \ |f(x + b(\Delta)) - f(x)| < \varepsilon \rightarrow \quad (\text{xiii})$$

After cosmetic paramodulation with  $U = 3 \cdot (U/3)$ , (xiii) resolves with (x) to produce

$$\begin{aligned} \delta(\varepsilon/3, x, N) > 0 \ \& \\ \varepsilon/3 > 0 \ \& \ |b(\delta(\varepsilon/3, x, N))| < \delta(\varepsilon/3, x, N) \ \& \\ N \geq m(\varepsilon/3, x) \ \& \ N \geq m(\varepsilon/3, x + b(\delta(\varepsilon/3, x, N))) \rightarrow \end{aligned} \quad (\text{xiv})$$

With the aid of  $U > 0 \rightarrow U/3 > 0$ , we can use (vi), (xi) and (xii) to resolve the first three literals away leaving the rump:

$$N \geq m(\varepsilon/3, x) \ \& \ N \geq m(\varepsilon/3, x + b(\delta(\varepsilon/3, x, N))) \rightarrow \quad (\text{xv})$$

These almost resolve with  $\rightarrow Y \geq Y$ , but the presence of  $N$  on the right-hand side of the second literal causes the occurs check to fail. In fact, counter-examples can be found to the second literal.

#### 4. CORRECTING THE POOF

Now to fix up the poof. The occurs check would not fail if  $N$  were not contained in  $m(\varepsilon/3, x + b(\delta(\varepsilon/3, x, N)))$  and it would not be so contained if  $m$  were a unary Skolem function dependent only on  $\varepsilon/3$ . Locating the introduction of  $m$  in the definition of convergence of  $f$ , we see that  $m$  would have the required argument structure if the quantifier order were changed to:

$$\forall E > 0 \exists M \forall X \forall N \geq M \left| \sum_{i=N+1}^{\infty} f_i(X) \right| < E \quad (\text{xvi})$$

but this is the definition of *uniform convergence*. We will call this the *m unary' solution*.

Now the rump goals are:

$$N \geq m(\varepsilon/3) \ \& \ N \geq m(\varepsilon/3)$$

which will resolve with  $\rightarrow Y \geq Y$  to prove the theorem.

However,  $m$  is not the only Skolem function surrounding the offending occurrence of  $N$ , there are also  $\delta$  and  $b$ . Similar corrections can be made by making  $\delta$  a binary function, independent of its third argument,  $N$ , or by making  $b$  a constant, independent of  $\delta(\varepsilon/3, x, N)$ .

### 5. ALTERNATIVE CORRECTIONS—MAKING $b$ A CONSTANT

The Skolem function  $b$  arises from the definition of continuity.

$$\forall X \forall E > 0 \exists \Delta > 0 \forall B |B| < \Delta \rightarrow |f(X+B) - f(X)| < E$$

when it is negated and Skolemised as the conclusion of the theorem. The required correction involves putting the  $\forall B$  before the  $\exists \Delta$  to create a concept I shall call *weak continuity*, because it is a weaker property than continuity, i.e.

$$\forall X \forall E > 0 \forall B \exists \Delta > 0 |B| < \Delta \rightarrow |f(X+B) - f(X)| < E$$

However, this is a trivial property that holds for all functions. In particular the square wave (the normal counter-example to Cauchy's proof). This is because we can choose  $\Delta$  to depend on  $B$ . For instance, we can choose  $\Delta$  to be  $|B|/2$ , but anything guaranteed less than or equal to  $|B|$  will do. Now  $|B| < \Delta$  is always false, so weak continuity is always true.

In contrast to the make  $m$  unary case, the rump subgoals.

$$N \geq m(\varepsilon/3, x) \ \& \ N \geq m(\varepsilon/3, x+b) \rightarrow$$

are not identical and cannot be resolved away with  $\rightarrow Y \geq Y$ . However, they can be resolved away with  $\rightarrow \max(Y, Z) \geq Y$  and  $\rightarrow \max(Y, Z) \geq Z$ .

Hand checking indicates that this version of Cauchy's proof is legal, although the theorem it proves is trivial, since its conclusion is always true.

### 6. ALTERNATIVE CORRECTIONS—MAKING $\delta$ A BINARY FUNCTION

The Skolem function  $\delta$  arises from the lemma that  $s(N, X)$  is continuous for all  $N$ , i.e.

$$\forall N \in Z \forall X \forall E > 0 \exists \Delta > 0 \forall B |B| < \Delta \rightarrow |s(N, X+B) - s(N, X)| < E$$

This makes  $\delta$  dependent on  $N, X$  and  $E$ . We can remove the dependence on  $N$  by shifting  $\forall N$  after  $\exists \Delta$  to create a concept called *equi-continuity* by Ascoli (Simmons, 1963, p. 126), i.e.

$$\forall X \forall E > 0 \exists \Delta > 0 \forall N \in Z \forall B |B| < \Delta \rightarrow |s(N, X+B) - s(N, X)| < E$$

Tracing this lemma back in the proof we see that the family of  $f_n$ s must also be equi-continuous.

$$\forall X \forall E > 0 \exists \Delta > 0 \forall N \in Z \forall B |B| < \Delta \rightarrow |f(N, X+B) - f(N, X)| < E$$

The proof of the rump subgoals,

$$N \geq m(\varepsilon/3, x) \ \& \ N \geq m(\varepsilon/3, x + b(\delta(\varepsilon/3, x))) \rightarrow$$

is similar to that in the make- $b$ -constant solution, i.e. they are resolved away with  $\rightarrow \max(Y, Z) \geq Y$  and  $\rightarrow \max(Y, Z) \geq Z$ .

Hand checking of this version of Cauchy's proof suggests that it is legal. This version of the theorem seems to be a genuine and interesting addition to the standard one suggested by the make- $m$ -unary correction. It involves a different, but independently interesting, concept: equi-continuity. I have been unable to find any reference to it in the literature. Its proof is similar to parts of the proof of Ascoli's theorem. (Simmons, 1963, pp. 124–128), but Ascoli's theorem itself is different. This version of Cauchy's theorem seems too obvious not to have been previously discovered by someone, and I would be grateful for any help in tracking it down.

## 7. THE AUTOMATION OF POOF ANALYSIS

What are the requirements for automating this process?

- (a) Assuming that the initial representation of the poof is in mathematical English (or 'mathematese'), the program must compare predicate calculus definitions like (iv) and (v) above with the mathematese statements like (i), (ii) and (iii), to produce predicate calculus versions like (vi), (vii), (viii) and (ix).
- (b) the program must fill in gaps in the informal proof, e.g. derive the lemmas (xiv) and (xv).
- (c) The program must recognise failure, as at step (xv), and the reason for it, i.e. occurs check failure.
- (d) The program must be able to fix up failures by modifying the theorem, e.g. by reordering the quantifiers in the definition of convergence.

(d), above, has been implemented, by the author, in a Prolog program, SEIDEL. The program takes a description of an occurs check bug in a poof, and produces all those modifications of the original theorem which correct the poof. The description of the occurs check bug consists of the variable,  $Y$ , and term,  $f(Y)$ , on which the unification attempt should have failed in the original poof. SEIDEL finds a skolem function,  $sk$ , in  $f(Y)$ , and marks the argument place of  $sk$  which contains  $Y$  (suppose this is the  $n$ th place). It recovers the formula,  $F_m$ , whose skolemization gave risk to  $sk$  and modifies  $F_m$  to remove the dependence of  $sk$  on  $Y$ . SEIDEL searches, depth first, for all such modifications.

In the current version of SEIDEL,  $F_m$  has to be in prenex normal form, i.e. a list of quantifiers, Prefix, followed by a quantifier free formula, Matrix. Prefix must be of the following form:

$$\dots QX, \dots, RSk, \dots$$

where  $Q$  is the quantifier governing variable  $X$ , and  $R$  is the quantifier governing variable  $Sk$ .  $Sk$  is the variable which skolemizes to  $sk$ , and  $X$  is its  $n$ th argument. If  $F_m$  is skolemized positively then  $R$  will be an existential quantifier and  $Q$  a universal one. If  $F_m$  is skolemized negatively, then vice versa.

The idea of the modification is to move the quantifiers in Prefix so that  $QX$  no longer appears to the left of  $RSk$ . This will remove the dependence of  $sk$  on its  $n$ th argument place and thus on  $Y$ . We try to do this in a way that minimises the disruption to  $F_m$ , i.e. so that there is no unnecessary modification.

$F_m$  preserves its meaning under any permutation of similar quantifiers within Prefix, i.e. two adjacent existential or two adjacent universal quantifiers can be exchanged without affecting the meaning of the formula. This device is used to minimise the subsequent disruption to  $F_m$ . That is,  $RSk$  is moved as far leftwards as possible within its group of R-type quantifiers, and  $QX$  is moved as far rightwards as possible within its group of Q-type quantifiers.

At this point  $RSk$  and  $QX$  may be adjacent. This happens in all the examples in this paper. They may then be exchanged to remove the dependence of  $sk$  on  $x$ . Otherwise, there are two minimal solutions:  $RSk$  may be moved leftwards to immediately left of  $QX$ , or  $QX$  may be moved rightwards to immediately right of  $RSk$ .

SEIDEL works correctly on the Cauchy poof, producing the concepts of uniform convergence, weak continuity and equi-continuity.

This completes the description of the current state of SEIDEL. Further work required includes: the removal of the restriction to prenex normal form; the addition of other bug types and fixes; and the addition of mechanisms for (a), (b) and (c) above.

- (a) looks feasible—it requires fairly straightforward natural language understanding, perhaps using semantic grammars geared to mathematase.
- (b) could be done using theorem proving with some length of proof bound, say.
- (c) could be tricky—how do you know when failure is absolute and not some limitation of the proof bound? Perhaps counter-examples could be brought to bear here. If the poof comes with a global counterexample then this can be used to test gaps—à la Gelernter (1963).



## 8. AN APPLICATION TO NATURAL LANGUAGE UNDERSTANDING

The purpose of this section is to show that poof analysis, and even the particular technique of quantifier swapping described above, may find application outside of mathematics. We will try to demonstrate a use for the technique in a very different area: disambiguating the meaning of sentences in natural language understanding.

A classical ambiguous sentence in English is

Everybody loves somebody. (xviii)

This can be represented in Predicate Logic as:

$\exists X \forall Y \{ \text{person}(X) \ \& \ (\text{person}(Y) \rightarrow \text{loves}(Y, X)) \}$  (xix)

or

$\forall Y \exists X \{ \text{person}(X) \ \& \ (\text{person}(Y) \rightarrow \text{loves}(Y, X)) \}$  (xx)

Suppose (xix) was intended by a speaker but (xx) was understood by the hearer.

The speaker might go on to say

Therefore, someone loves themself. (xxi)

This does indeed follow from (xix) but not from (xx). The hearer may then detect the problem by realising that (xxi) does not follow from his/her understanding of (xviii). However, by relaxing the occurs check s/he can generate a poof of (xxi) from (xviii).

In clausal form the hearer's version of the poof is:

$\rightarrow \text{person}(x(Y))$  from (xx) (xxii)

$\text{person}(Y) \rightarrow \text{loves}(Y, x(Y))$  from (xx) (xxiii)

$\text{person}(Z) \ \& \ \text{loves}(Z, Z) \rightarrow$  from the negation of (xxi) (xxiv)

$\text{person}(x(Y)) \rightarrow$  from (xxiii) and (xxiv)

from (xxv) and (xxii)

The faulty step is the unification of  $\text{loves}(Y, x(Y))$  and  $\text{loves}(Z, Z)$ , since this violates the occurs check—one is not allowed to unify  $Y$  and  $x(Y)$ . The poof can be corrected by removing the dependance of the skolem function  $x$  on the variable  $Y$ . Using the technique of section 7 this is done by reversing the order of the quantifiers in (xx), which gives (xix). Thus using poof analysis to correct the faulty argument reconstructs the correct translation of the ambiguous sentence.

This example has been run successfully on SEIDEL.

## 9. CONCLUSION

Concept formation is an important area of research in Artificial intelligence and a vital ingredient of a program for modelling the cognitive process involved in mathematical reasoning. New concepts can easily be formed by manipulation of old ones, but the search space is combinatorially explosive. One way to guide the search through this space is to form concepts that facilitate the solving of problems. We call this problem-driven concept-formation.

Poof analysis is one technique for problem-driven concept-formation in mathematics. We have investigated its application in a classic example—the invention of uniform convergence. This investigation has suggested a systematic technique for poof correction and concept formation, and led to the discovery of alternative corrections and concepts. The technique has also been applied to the problem of disambiguation in natural language understanding. The technique has been partially implemented.

## ACKNOWLEDGMENTS

I am grateful for feedback on earlier versions of this paper to members of the mathematical reasoning group in the Edinburgh AI Department, and to Alan Sinclair of the Mathematics Department at Edinburgh.

This research was supported by SERC grant GR/B/20826.

## REFERENCES

- Burton, R. R. (1982). *Diagnosing bugs in a simple procedural skill*, Academic Press, London; pp 157–183.
- Gelernter, H. (1963). Realization of a Geometry theorem-proving machine. In Feigenbaum E and Feldman J (eds) *Computers and Thought*, pp 134–52. McGraw-Hill, New York.
- Lakatos, I. (1976). *Proofs and refutations; The logic of Mathematical discovery*, Cambridge University Press, Cambridge.
- Lenat, D. B. (1982). AM: An Artificial Intelligence approach to discovery in Mathematics as Heuristic Search. In *Knowledge-based systems in artificial intelligence*. McGraw-Hill. Also available from Stanford as TechReport AIM 286.
- Mitchell, T. M. (1983). Learning and Problem Solving. In Bundy A (ed) *Proc Eighth IJCAI*. pp 1139–1151, Morgan Kaufmann, Los Altos.
- Silver, B. (1986). Precondition Analysis: Learning Control Information, in Michalski R S, Carbonell J G and Mitchell T M (eds) *Machine Learning*, 2, Morgan Kaufmann, Los Altos. Earlier version available from Univ of Edinburgh as DAI Research Paper 220.
- Simmons, G. F. (1963). *Introduction to Topology and Modern Analysis*. McGraw-Hill, New York.