Title

An Architecture For Mathematical Cognition

Permalink

https://escholarship.org/uc/item/72k6s7kz

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 8(0)

Author

Ammon, Kurt

Publication Date

1986

Peer reviewed

AN ARCHITECTURE FOR MATHEMATICAL COGNITION

Kurt Ammon[1]
Department of Computer Science
University of Hamburg

ABSTRACT

This paper presents the architecture of the discovery system SHUNYATA which models studies and research in higher mathematics. SHUNYATA analyzes mathematical proofs and produces concepts and proof strategies which form the basis for the discovery of more difficult proofs in other mathematical theories. Its architecture avoids combinatorial explosions and does not require search strategies. The proof strategies contain two categories of predicates. A predicate of the first category selects a small set of proof steps and the predicates of the second category evaluate partial proofs and decide which predicate of the first category should be applied next. Thus, the proof strategies include feedback loops. A detailed example is given. It contains a simple proof in group theory, the analysis of this proof, and the discovery of a proof in lattice theory whose degree of difficulty represents the state-of-the-art in automated theorem proving. The most important result of this work is the discovery of a holistic logic based on the concept that cognitive structures arise from simple perceptions, evolve by reflection and finally contain their own evolution mechanisms.

Keywords: Learning, knowledge acquisition, cognitive evolution, automated theorem proving.

1 INTRODUCTION

Traditional research in machine learning assumes the existence of domain-independent and objectivizable cognitive structures and discovery mechanisms (e.g., [4]). This approach entails the following difficulties:

- The learning system cannot change its representation language and its structure (e.g., [11]).
- After a period of time, the efficiency of the system decreases drastically (e.g., [8]).

SHUNYATA automatically changes its language and its structure on the basis of experience which increases its efficiency. A crucial consequence is that it is impossible to objectivize its architecture, i.e., it cannot be analyzed completely.

The organization of this paper is as follows. Section 2 gives an overview of SHUNYATA. Section 3 describes the reflection system which forms the core of SHUNYATA. Section 4 introduces the concept of analytical spaces and Section 5 presents holistic logic. Sections 6 and 7 give a simple proof in group theory and the analysis of this proof. Section 9 describes the discovery of a difficult proof in lattice theory.

[1] Copyright © 1986 by Kurt Ammon. All rights reserved. Further, storage and utilization of the described programs on data processing installations is forbidden without prior written permission of the author.

2 OVERVIEW OF SHUNYATA

SHUNYATA has the structure of a tree in functional representation, i.e., every object is either a symbol or else has the form

$$f(a_1, ..., a_n),$$

where f is a function and a_1, ..., a_n are arguments. At the most general level SHUNYATA does not consist of interacting components but has only this functional structure which is very suitable for representing reasoning processes. SHUNYATA contains three kinds of functions:

- *Calculable functions.* These are conventional effectively calculable functions, i.e., they are objectivizable because they have a precise representation and produce precisely describable values.
- *Strange functions.* The value of a strange function is composed of the name of the function and its arguments. Finite sets, tuples, and bags can be regarded as strange functions. Example: The set $\{a_1, ..., a_n\}$ is represented by

  $$fset(a_1, ..., a_n),$$

  where the value of $fset(a_1, ..., a_n)$ is $fset(a_1, ..., a_n)$.
- *Holistic functions.* Holistic functions evolve through experience and cannot be objectivized. Their development is accompanied by division and unification processes. Existential quantification, universal quantification, and the operator that constructs sets from predicates are holistic functions. The undecidability of predicate logic [5] indicates that the decision procedure is a holistic function.

SHUNYATA contains elementary knowledge for evaluating holistic functions. In particular, it uses a basic procedure to evaluate predicates that define finite sets. The central mechanism of this procedure is the replacement of element relations by disjunctions of equality relations. In SHUNYATA holistic functions presently produce the value *not-evaluable* if the system does not contain sufficient knowledge for evaluating the function.

The most important functions of SHUNYATA are the *analyze* function and the *proof* function which perform the proof analyses and generate proofs. The *analyze* function has the form

$$analyze(C, T, P),$$

where C is a predicate calculus, T a theorem, and P a proof for the theorem. It produces a proof strategy. The *proof* function has the form

$$proof(C, T, S),$$

where C is a predicate calculus, T a theorem, and S a proof strategy. It produces a proof for the theorem. The *analyze* and *proof* functions illustrate the spiral organization of SHUNYATA: The *analyze* function constructs powerful proof strategies from proofs and the *proof* function constructs complex proofs from proof strategies.

3 THE REFLECTION SYSTEM

The reflection system is the core of SHUNYATA. It contains over one hundred simple functions and their relations which are represented in a language closely related to predicate logic. The universe of this language is the set of all symbols and all trees. The functions are meta-level concepts such as the subset relation, the union of sets, and the addition of natural numbers. Thus, the reflection system contains precise knowledge about finite trees.

An inference can be represented by

$$m = i(a_1, ..., a_n),$$

where m is a meta-level theorem, i is an rule of inference, and $a_1, ..., a_n$ are the arguments. The system uses predicates to construct small sets of inferences. These predicates are modified dynamically. The activity of the reflection system changes rapidly because it is controlled by the results of the inferences. The set of meta-level theorems generated by the reflection system is called reflection space. The architecture described avoids combinatorial explosions because the reflexion space can be pruned by additional predicates. This mechanism produces a combinatorial reduction of the size of the reflection space with regard to the number of predicates.

4 ANALYTICAL SPACES

Cognition permanently reduces huge amounts of information to a few concepts. Miller [10] argues that only some seven concepts are contained in short-term memory simultaneously. This reduction forces cognition to divide the world into analytical spaces which consist of a few concepts and the environment these concepts refer to. An analytical space represents a view of the world. Examples are geometry and space in mathematics, properties of programs and programs in computer science, and proof strategies and proofs in SHUNYATA. The principle of complementarity [3] states that quantum theory requires the existence of different analytical spaces. Gödel's Theorem [6] implies that there are different analytical spaces for the theory of natural numbers. Traditional epistemology regards the world and cognition as completely separable entities. From a holistic point of view the world and cognition are divided into different analytical spaces which contain essential but incomplete knowledge.

5 HOLISTIC LOGIC

The SHUNYATA system is the first step towards an implementation of holistic logic. The central hypotheses of this logic are:

- The basis or kernel of cognition is formless.
- Cognitive evolution creates new analytical spaces (*division*) and integrates existing analytical spaces (*unification*).

New information in a cognitive system generates elementary analytical spaces which cause perturbations in its structures. This process can be considered as low-level perception. In integrating the new analytical space the system must preserve its previous efficiency. It first tries to assimilate the new space into existing structures. If this is impossible, the new analytical space begins its own evolution, i.e., the system performs a division process. The evolution is controlled by the activity of the reflection system. Cognition achieves the most important advancements in its development by integrating different analytical spaces, i.e., by unification processes. Even the high-level structures of the reflection system are permanently revised and improved by divisions and unifications on the basis of its low-level structures. Because cognitive structures arise from simple perception, their origin can be regarded as an empty or formless kernel, i.e., the kernel lacks properties and structures. Thus, the theory of holistic logic implies that it is impossibe to objectivize cognition and discovery mechanisms completely. The integration of new information can be regarded as an induction process which creates new structures. These are modified by future experience, i.e., by feedback processes. Therefore, cognitive structures evolve by induction and feedback [1].

6 A SIMPLE PROOF

This section gives a formalization of the elementary theory of groups, a theorem, and a complete formalization of a proof which includes the reasons for the theorems. This approach to formalization forms the basis for the analysis of the proof. It can be applied to predicate logic and meta-level reasoning. The symbols and functions used in this section are defined in the appendix.

1. *One binary predicate letter:* p. We write $x = y$ for $p(x, y)$.
2. *Three binary function letters:* f, g, h. We write xy for $f(x, y)$.
3. *Constant:* c.
4. *Axioms*: $(xy)z = x(yz)$, $g(x, y)x = y$, $xh(x, y) = y$.
5. *Rules of inference:*
   - (a) Substitution Rule: $r = u \in E,\ s \in subs(r = u) \Longrightarrow sub(r = u, s) \in E$. Function: $sub(r = u, s)$.
   - (b) Reflexivity Rule: $r \in R \Longrightarrow r = r \in E$. Function: $ref(r)$.
   - (c) Symmetry Rule: $r = u \in E \Longrightarrow u = r \in E$. Function: $sym(r = u)$.
   - (d) Transitivity Rule: $r = u \in E,\ u = v \in E \Longrightarrow r = v \in E$. Function: $tran(r = u, u = v)$.
   - (e) Replacement Rule: $r \in R,\ d \in D,\ u = v \in E,\ d(r) = u \Longrightarrow rep(r, d, u = v) \in E$. Function: $rep(r, d, u = v)$.
   - (f) Chain Rule: $r \in R,\ d \in D,\ u = v \in E,\ s \in subs(u = v), d(r) = st(1, sub(u = v, s)) \Longrightarrow rep(r, d, sub(u = v, s)) \in E$. Function: $chain(r, d, u = v, s)$.
6. *Theorem:* $g(c, c)x = x$.
7. *Proof:*

Theorems	Reasons
$g(c,c)x = g(c,c)(ch(c,x))$	$chain(g(c,c)x,\ 2,\ y = xh(x,y),\ \{(x,c),(y,x)\})$
$g(c,c)(ch(c,x)) = (g(c,c)c)h(c,x)$	$chain(g(c,c)(ch(c,x)),\ (),\ x(yz) = (xy)z,$ $\{(x,g(c,c)),(y,c),(z,h(c,x))\})$
$(g(c,c)c)h(c,x) = ch(c,x)$	$chain((g(c,c)c)h(c,x),\ 1,\ g(x,y)x = y,\ \{(x,c),(y,c)\})$
$ch(c,x) = x$	$chain(ch(c,x),\ (),\ xh(x,y) = y,\ \{(x,c),(y,x)\})$

The proof steps containing the Symmetry Rule or the Transitivity Rule are omitted. The proof in ordinary scientific notation is

$$g(c,c)x = g(c,c)(ch(c,x)) = (g(c,c)c)h(c,x) = ch(c,x) = x.$$

7 THE ANALYSIS OF THE PROOF

This section describes the automatic analysis of the previous proof. The result of this analysis is a proof strategy.

The proof in the previous section contains four steps Y_1, Y_2, Y_3, and Y_4. These proof steps are tuples

$$(r = t, chain(r, d, u = v, s)),$$

where $r \in R$, $d \in D$, $u = v \in E$, and $d(r) = st(1, sub(u = v, s))$. SHUNYATA analyzes the four proof steps successively. The analysis produces two categories of predicates which represent the proof strategies. These predicates are generated from the functions of the reflection system like the terms and well-formed formulas of a predicate calculus. The predicates of the first category have the form

$$p(C, T, X, Y),$$

where C is a predicate calculus, T the theorem to be proved, X a tuple of proof steps, and Y a proof step. At the beginning of the analysis the tuple X is empty. SHUNYATA tests whether these predicates satisfy the following requirements:

- The evaluation of $p(C, T, X, Y_i)$ yields the truth value *true* where Y_i, $i \in \{1, 2, 3, 4\}$, is the proof step to be analyzed.
- The evaluation of $\{Y \mid p(C, T, X, Y)\}$ yields a small set of proof steps in a limited space of time.

The predicates of the second category have the form

$$q(C, T, X).$$

where C is a predicate calculus, T the theorem to be proved, and X a tuple of proof steps. They decide which predicate of the first category should be applied next. Therefore, the proof strategies include feedback loops. The proof steps generated by the predicates of the first category are appended to the tuple X until X contains the complete proof, i.e., the proof steps Y_1, Y_2, Y_3, and Y_4.

1. *The analysis of the first proof step.* SHUNYATA generates predicates and tests whether they satisfy the requirements described. It discovers the proof strategy:

   $$\begin{array}{|l|}\hline r = st(1, T) \wedge \\ u = v \in axms(C) \cup \{b \mid (\exists a)(a \in axms(C) \wedge b = sym(a))\} \wedge \\ pr(2, s) \subseteq con(T) \cup var(T) \\ \hline \end{array}$$

   This proof strategy produces twelve proof steps.

2. *The analysis of the second proof step (division).* The previous strategy does not generate the second proof step. Thus, the second proof step can be considered as a perturbation with regard to the previous strategy. SHUNYATA tests further predicates but they fail to produce the two proof steps. Therefore, it has to perform a division process, i.e., it contructs a second predicate that produces the second proof step and a criterion that decides when the first and the second predicate should be applied. The division yields the strategy:

   $$\begin{array}{|l|}\hline \ldots \\ \hline r \in pr((1, 2), X) - pr((1, 1), X) \wedge \\ u = v \in axms(C) \cup \{b \mid (\exists a)(a \in axms(C) \wedge b = sym(a))\} \wedge \\ ne(lvs(u)) = ne(lvs(v)) \\ \hline \end{array}$$

The dots represent the first predicate. The criterion states that the first predicate is applied only if the second predicate generates no proof steps, i.e.,

$$\{Y \mid p(C,T,X,Y)\} = \emptyset,$$

where p denotes the second predicate. Thus, the second predicate has priority. This strategy produces four additional proof steps.

3. *The analysis of the third proof step (unification).* It is very difficult to integrate the third proof step. The division process first fails because it yields a strategy that generates an infinite number of proof steps. Then, it produces a complex strategy with three predicates:

...
...
$r \in pr((1,2),X) - pr((1,1),X) \wedge$ $u = v \in axms(C) \cup \{b \mid (\exists a)(a \in axms(C) \wedge b = sym(a))\} \wedge$ $ne(lvs(u)) > ne(lvs(v)) \wedge$ $not(t \in pr((1,1),X) \cup pr((1,2),X))$

The dots represent the first and the second predicate. The third predicate has priority. Thus, the third proof step causes a strong perturbation in the previous strategy. Finally, SHUNYATA unifies the second and the third predicate and discovers a simple and efficient strategy:

...
$r \in pr((1,2),X) - pr((1,1),X) \wedge$ $u = v \in axms(C) \cup \{b \mid (\exists a)(a \in axms(C) \wedge b = sym(a))\} \wedge$ $ne(lvs(u)) \geq ne(lvs(v)) \wedge$ $not(t \in pr((1,1),X) \cup pr((1,2),X))$

The dots represent the first predicate. This strategy produces the third proof step.

4. *The analysis of the fourth proof step.* The previous strategy also produces the fourth proof step.

Thus, the proof strategy produces eighteen proof steps, i.e., eighteen theorems:

$g(c,c)x = g(g(c,c)c,c)x$			
$g(c,c)x = g(ch(c,c),c)x$			
$g(c,c)x = g(c,g(c,c)c)x$			
$g(c,c)x = g(c,ch(c,c))x$			
$g(c,c)x = g(g(x,c)x,c)x$			
$g(c,c)x = g(xh(x,c),c)x$			
$g(c,c)x = g(c,g(x,c)x)x$			
$g(c,c)x = g(c,xh(x,c))x$			
$g(c,c)x = g(c,c)(g(c,x)c)$	$... = (g(c,c)g(c,x))c$		
$\underline{g(c,c)x = g(c,c)(ch(c,x))}$	$\underline{... = (g(c,c)c)h(c,x)}$	$\underline{... = ch(c,x)}$	$\underline{... = x}$
$g(c,c)x = g(c,c)(g(x,x)x)$	$... = (g(c,c)g(x,x))x$		
$g(c,c)x = g(c,c)(xh(x,x))$	$... = (g(c,c)x)h(x,x)$		

The dots represent the terms on the right side of the preceding column. The proof is underlined. The first column is generated by the first predicate of the proof strategy and the other columns are generated by the second predicate.

The proof steps can be considered as new information for the system which can cause perturbations in its structures. The analysis of the proof steps produces predicates and proof strategies, i.e., new concepts. These predicates are generated like the terms and well-formed formulas of a predicate calculus and they are selected by experimentation. The predicates and proof strategies that were produced by analyzing the previous proof steps form the basis for the analysis of the next proof step. Thus, SHUNYATA changes its language and its structure on the basis of experience. The next section shows how the strategy for this simple proof in group theory can be used for discovering a difficult proof in lattice theory.

8 THE DISCOVERY OF A DIFFICULT PROOF

This section describes the discovery of a proof for SAM's Lemma which was an open problem until 1969. It was solved by Guard et al. [7] and subsequently by McCharen et al. [9]. The degree of difficulty for discovering a proof for SAM's Lemma represents the state-of-the-art in automated theorem proving [2]. The size of a proof in ordinary scientific notation is approximately one and a half pages [12].

1. *One binary predicate letter: p.* We write $x = y$ for $p(x, y)$.
2. *Two binary function letters: f, g.* We write xy for $f(x, y)$ and $x + y$ for $g(x, y)$. In order to omit parentheses, we assume that the first function has priority.
3. *Constants:* $0, 1, a, b, c, d$.
4. *Axioms*:
$$\begin{array}{llll} (xy)z = x(yz) & (x+y)+z = x+(y+z) & xy = yx & x+y = y+x \\ xx = x & x+x = x & x(x+y) = x & x+xy = x \\ 0x = 0 & 0+x = x & 1x = x & 1+x = 1 \\ (a+b)c = 0 & (a+b)+c = 1 & (ab)d = 0 & ab+d = 1 \\ \multicolumn{4}{l}{x+z = z \Rightarrow (x+y)z = x(y+z)} \end{array}$$
5. *Theorem*: $(c + da)(c + db) = c$.
6. *Proof Strategy*: The proof strategy for the theorem in group theory is applied but the axiom that contains the implication requires special treatment: The consequence is treated like the other equations if the condition can be proved by the repeated application of the second predicate of the strategy. This strategy generates seventy-four theorems. The application of the first predicate may be regarded as an extension step and the repeated application of the second predicate as a simplification step. The extension step produces seventy-two theorems and the simplification step two theorems.

Extension	Simplification
$(c+da)(c+db) = (cc+da)(c+db)$	
...	
$\underline{(c+da)(c+db) = (c+da(a+b))(c+db)}$	$\underline{(c+da(a+b))(c+db) = c}$
...	
$(c+da)(c+db) = (c+da)(c+db(b+a))$	$(c+da)(c+db(b+a)) = c$
...	
$(c+da)(c+db) = (c+da)(c+d1b)$	

Some parentheses are omitted because of the associativity of the first function. The first proof is underlined.

7. *Proof*: The strategy produces the proof

$$(c + da)(c + db) = (c + da(a + b))(c + db) = c.$$

9 RELATED WORK

The first project concerned with automated mathematics research was AM [8, p. 137]. AM focuses on research in elementary mathematics, SHUNYATA on research in higher mathematics. AM discovers mathematical concepts, SHUNYATA additionally develops powerful cognitive structures, e.g., new ideas. AM uses rules to represent heuristic knowledge whereas SHUNYATA uses a modified predicate calculus, in particular predicates that generate finite sets of trees. AM contains sophisticated heuristic rules from the beginning and does not learn from experience. In contrast, SHUNYATA analyzes new information on the basis of a simple language closely related to predicate logic and develops concepts and discovery mechanisms on the basis of experience. AM does not generate proofs.

10 CONCLUSIONS

In this paper, I have described an idealized trace of cognitive evolution from the very beginning to a level that is suitable for research in higher mathematics. The origin of cognitive structures is a formless kernel. This formulation is a short paraphrase of the hypothesis that cognitive structures arise from simple perceptions, evolve by reflection and finally contain their own evolution mechanisms. Their development is accompanied by division and unification processes and creates an increasing objectivization of the environment. The discovery system SHUNYATA models these high-level cognitive processes and constructs more advanced theories from weaker ones. It is an evolving tree in functional representation. Its core is a reflection system which contains a language that is potentially universal because this language is permanently revised and improved. Cognitive structures are holistic in the sense that they cannot be reduced to simpler structures and that they cannot be completely separated from the objects they refer to, i.e., they are domain specific because they change on the basis of experience. The experiments suggest that artificial cognitive systems must have at least the computational capacity of human cognition and that they must use the same cognitive structures and the same organization.

SHUNYATA is written in ZETALISP and is presently running on a Symbolics 3600. The system has constructed many proofs from proof strategies, for example the previous proof for SAM's Lemma without special procedures for associativity and commutativity. If a representation similar to the representation humans use for associativity and commutativity is integrated into SHUNYATA, its efficieny increases by a factor of over two hundred. In generating finite sets of proof steps from predicates, the system produces more than one hundred simple LISP programs per minute and evaluates them. SHUNYATA has analyzed several proofs and has acquired concepts and strategies for these proofs, for example the strategy described in this paper. The proof analysis requires multiprocessing. The strategies are LISP programs which contain holistic functions and which are written and compiled by SHUNYATA. They can be regarded as heuristic knowledge which is modified on the basis of experience. Thus, the proof analysis involves the automatic and evolutionary development of programs by division and unification. The automatic revision and extension of the reflection system has not yet been implemented.

APPENDIX: NOTATIONS

$axms(C)$ The axioms of a predicate calculus.

$bag(a_1, ...a_n)$ The bag containing the elements $a_1, ..., a_n$. A bag is an unordered group of elements.
Example: $bag(x, y, x)$.

$con(u = v)$ The constants of an equation $u = v$.
Example: $con(g(c, c)x = x) = \{c\}$.

D The set of subtree descriptors. Subtree descriptors are tuples of natural numbers that denote subtrees of trees.
Examples: The subtree descriptor $()$ denotes the subtree $f(a, b)$ of the tree $f(a, b)$, the subtree descriptor (2) the subtree $g(b, c)$ of the tree $f(a, g(b, c))$, and the subtree descriptor $(2, 1)$ the subtree b of the tree $f(a, g(b, c))$. We write 1 for the subtree descriptor (1) and 2 for the subtree descriptor (2).

E The theorems of a predicate calculus.

$lvs(t)$ The bag of leaves of a tree t.
Example: $lvs(f(g(x, y), x)) = bag(x, y, x)$.

$ne(b)$ The number of elements in a bag b.
Example: $ne(bag(x, y, x)) = 3$.

$pr(d, t)$ The projection of a set of trees t described by the subtree descriptor d.
Examples: $pr(1, \{(a, b), (c, d)\}) = \{a, c\}$,
$pr((1, 2), \{(f(a, b), c), (f(d, e), c)\}) = \{b, e\}$.

R The terms of a predicate calculus.

$rep(r, d, u = v)$ The replacement of the subterm of a term r that is selected by the subtree descriptor d by the term v of the equation $u = v$.
Example: $rep(g(c, c)x, 2, x = ch(c, x)) = g(c, c)(ch(c, x))$.

$st(d, t)$ The subtree of the tree t that is selected by the subtree descriptor d.
Examples: $st((), f(a, b)) = f(a, b)$, $st(2, f(a, b)) = b$,
$st((1, 2), f(g(a, b), c)) = b$.

$sub(u = v, s)$ The application of the substitution s for the variables in the equation $u = v$ to the equation $u = v$.
Example: $sub(y = xh(x, y),\ \{(x, c), (y, x)\}) = x = ch(c, x)$.

$subs(u = v)$ The substitutions for the variables of an equation $u = v$.
Example: $\{(x, c), (y, x)\}$ is a substitution for the variables of the equation $y = xh(x, y)$.

$var(u = v)$ The variables of an equation $u = v$.
Example: $var(g(c, c)x = x) = \{x\}$.

ACKNOWLEDGEMENTS

I wish to thank Wilfried Brauer, Bernd Neumann, and Hans-Joachim Novak for providing constructive comments on an earlier draft of this paper. I also wish to thank Ernie Cohen and John McCarthy for helpful conversations. Special thanks to Russell Block for improving my English.

REFERENCES

1. Ammon, K. The Automatic Discovery of Concepts by Induction and Feedback: Towards a Theory of Intelligence. In H. Trost and J. Retti (Eds.) *Österreichische Artificial Intelligence-Tagung*. Springer-Verlag, Berlin and Heidelberg, 1985.

2. Antoniou, G., and Ohlbach, H. J. TERMINATOR. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, 1983.

3. Bohr, N. The Quantum Postulate and the Recent Development of Atomic Theory. *Nature* 121 (1928), pp. 580-590.

4. Carbonell, J. G., Michalski, R. S., and Mitchell, T. M. An Overview of Machine Learning. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (Eds.) *Machine Learning: An Artificial Intelligence Approach*. Tioga, Palo Alto, 1983.

5. Church, A. A Note on the Entscheidungsproblem. *The Journal of Symbolic Logic* 1 (1936), pp. 40-41. Correction, ibid., pp. 101-102.

6. Gödel, K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik* 38 (1931), pp. 173-198.

7. Guard, J. R., Oglesby, F. C., Bennett, J. H., and Settle, L. G. Semi-Automated Mathematics. *Journal of the ACM* 16 (1969), pp. 49-62.

8. Lenat, D. B. AM: Discovery in Mathematics as Heuristic Search. In R. Davis, and D. B. Lenat (Eds.) *Knowledge-Based Systems in Artificial Intelligence*. McGraw-Hill, New York, 1982.

9. McCharen, J. D., Overbeek, R. A., and Wos, L. A. Problems and Experiments for and with Automated Theorem-Proving Programs. *IEEE Transactions on Computers* C-25 (1976), pp. 773-782.

10. Miller, G. A. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Infomation. *Psychological Review* 63 (1956), pp. 81-97.

11. Mitchell, T. M., Utgoff, P. E., and Banerji, R. Learning by Experimentation: Acquiring and Refining Problem-Solving Heuristics. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (Eds.) *Machine Learning: An Artificial Intelligence Approach*. Tioga, Palo Alto, 1983.

12. Ohlbach, H. J. The Markgraf Carl Refutation Procedure: The Logic Engine. Interner Bericht 24/82, University of Karlsruhe, 1982.