

7857

# Lecture Notes in Mathematics

A collection of informal reports and seminars

Edited by A. Dold, Heidelberg and B. Eckmann, Zürich

125

---

## Symposium on Automatic Demonstration

Held at Versailles/France, December 1968

Edited by

M. Laudet, IRIA, Rocquencourt/France,

D. Lacombe, L. Nolin and M. Schützenberger, Faculté  
des Sciences, Paris/France



Springer-Verlag

Berlin · Heidelberg · New York 1970

SYMPOSIUM ON AUTOMATIC DEMONSTRATION  
COLLOQUE DEMONSTRATION AUTOMATIQUE

C O N T E N T S

LAUDET Michel	Allocution d'ouverture .....	1
ARNOLD André	Présentation d'un langage de formalisation des démonstrations mathématiques naturelles. ....	6
de BRUIJN N. G.	The mathematical language AUTOMATH, its usage, and some of its extensions .....	29
ENGELER Erwin	Proof theory and the accuracy of computations ....	62
FRAISSE Roland	Aspects du Théoreme de complétude selon Herbrand	73
GRZEGORCZYK A.	Decision procedure for theories categorical in Alef	87
HAO WANG	On the long-range prospects of automatic theorem proving .....	101
KOWALSKI Robert	The case for using equality axioms in automatic demonstration .....	112
KREISEL G.	Hilbert's programme and the search for automatic proof procedures .....	128
LOVELAND D. W.	A linear format for resolution .....	147
LUCKHAM David	Refinement theorems in resolution theory .....	163
PAWLAK Z.	Definitional approach to automatic demonstration	191
PITRAT Jacques	Heuristic interest of using metatheorems .....	194
PRAWITZ Dag	A proof procedure with matrix reduction .....	207
ROBINSON G. and WOS L.	Axiom systems in automatic theorem proving ....	215
SCOTT Dana	Constructive validity .....	237
WOS L. and ROBINSON G.	Paramodulation and set of support .....	276

## DEFINITIONAL APPROACH TO AUTOMATIC DEMONSTRATION

If one speaks about the application of a computer to numerical calculations we understand exactly the idea he has in mind. But speaking about theorem proving by means of a computer seems to be not clear enough. This raises many discussions on automatic demonstration - some times caused by misunderstanding the notion of a computer and /or the task which the computer has to perform by doing theorem proving. The aim of the presented note is to give the main fields where the computer can be used as an instrumental aid in mathematical creativity. First we shall define the notion of a computer.

Let  $T$  be finite or infinite set and let  $\pi$  be a partial function  $\pi : T \rightarrow T$ . Sequence  $t_0, t_1, \dots$  such that for all  $i, t_i \in T$  and  $t_{i+1} = \pi(t_i)$  will be called process. The process  $t_0, \dots, t_k$  is called finite if  $t_k \notin D$ , where  $D_\pi$  denotes the domain of the function  $\pi$ . Let us introduce binary relation  $M \subset T \times T$  defined as follows:  $\langle t, t' \rangle \in M$  if and only if there exist finite process  $t_0, \dots, t_k$  such that  $t_0 = t$  and  $t_k = t'$ . It is easily to show that relation  $M$  is a function. This function is called computer.  $T$  is referred to as a memory of a computer  $M$  and  $\pi$  is called the control of a computer  $M$ . It can be easily shown that each digital computer may be presented as a function  $M$ . We shall say that the computer  $M$  computes the function  $f : X \rightarrow X$  if and only if for all  $x \in X, f(x) = \delta \{M[\alpha(x)]\}$ , where  $\alpha : X \rightarrow T$  and  $\delta : T \rightarrow X$  are called coding and decoding functions respectively. For the sake of simplicity we shall omit the coding and decoding functions and write  $y = M(x)$ , which is to mean that we supply data  $x$  to the computer  $M$  and as a result of computation we obtain  $y$ .

In a similar way we may define the main tasks of computer application in automatic demonstration. Before we define the fields where the computer can be used in theorem proving let us first introduce some notations:

$S$  - sentence in mathematical language

$T$  - theorem

$P_t$  - proof of the theorem  $T$

$Cn T$  - the set of all consequences of  $T$  /by fixed set of axioms/

$Pr T$  - set of all premisses of  $T$

$Z$  - set of sentences

$Ak$  - set of axioms.

By means of the above notations we can introduce the following definitions:

1 - Computation of truth value

$$M(S) = \begin{cases} 1, & \text{if } S \text{ is a theorem,} \\ 0, & \text{if } S \text{ is not a theorem.} \end{cases}$$

2 - Production of formal proof

$$M(S) = \begin{cases} P_S, & \text{if } S \text{ is a theorem,} \\ 0, & \text{if } S \text{ is not a theorem.} \end{cases}$$

3 - Search for semantic proof

$$M(S, I) = \begin{cases} 1, & \text{if } S \text{ is truth by the interpretation } I, \\ 0, & \text{if } S \text{ is not truth by the interpretation } I. \end{cases}$$

4 - Production of counterexample

$$M(S) = \begin{cases} 1, & \text{if } S \text{ is a theorem} \\ \text{counterexample,} & \text{if } S \text{ is not a theorem.} \end{cases}$$

5 - Production of consequences

$$M(Ak, T) = C \subset Cn(Ak, T) .$$

6 - Production of premisses

$$M(T) = P \subset Pr(T) .$$

7 - Investigation of independence

$$M(T, T') = \begin{cases} 0, & \text{if } T \text{ and } T' \text{ are independent} \\ 1, & \text{if } T \text{ and } T' \text{ are not independent.} \end{cases}$$

8 - Simplification of proof

$$M(T, P) = P'$$

9 - Verification of proof

$$M(T, Z) = \begin{cases} 0, & \text{if } Z = P \\ 1, & \text{if } Z \neq P \end{cases}$$

10 - Equivalence of axioms

$$M(Ak) = Ak',$$

where  $Ak$  and  $Ak'$  are equivalent sets of axioms and  $Ak'$  is in some sense simpler than  $Ak$ .

It seems that these are the main tasks which can be solved by means of a computer in automatic demonstration. It would be interesting to discuss which of these tasks are most important in mathematical reasoning and which are most promising in successful computer application. If it turns out that there is a gap between these two fields the question arises how to bridge the gap.

This seems to be one of the most important problems in developing automatic demonstration. According to my opinion the future of the application of computers in mathematical work lies not in batch processing but in conversational mode of using the computer in theorem proving. In other words it means that points 1 and 2 are less promising than points 5 or 6 for example. There is a little hope that the computer can be useful in production the whole proof for some theorem. We may rather expect some positive results by application of a machine to produce some partial results in the process of theorem proving-which may approach the main problem the whole proof of a theorem.