Preprint MCS-P365-0693, Mathematics and Computer Science Division, Argonne National Laboratory

The Problem of Naming and Function Replacement*

LARRY WOS
Argonne National Laboratory, 9700 South Cass Ave., Argonne, IL 60439, U.S.A.

Abstract. This article is the twenty-eighth of a series of articles discussing various open research problems in automated reasoning. The problem proposed for research asks one to find criteria that an automated reasoning program can profitably use to remove functions present in the representation and replace them with appropriate predicates or constants that *name* the entities that were *named* by the functions. The notation used to present a problem to a reasoning program can have a profound effect on the likelihood of the program's success.

Key words. Automated reasoning, function replacement, naming, representation, unsolved research problem.

Question: *What criteria should be used to remove functions present in the representation of a problem and replace them with appropriate predicates or constants that name the entities that were named by the functions?*

(This question is the twenty-second of 33 problems proposed for research in [4] and will be referred to as Research Problem 22 throughout this article. All references to sections, chapters, test problems, and such also refer to [4].)

Ross Overbeek in the late 1960s and early 1970s designed and implemented a reasoning program in which two new procedures were introduced, *naming* [2] and *weighting* [1, 6]. He was in part motivated by the desire to study the commutator theorem (Test Problem 2, Section 6.1.2) cited in Section 5.3. In this theorem, one is asked to prove that when the cube of x is the identity e in a group, then $[[x,y],y] = e$; $[x,y]$ is the product of x, y, the inverse of x, and the inverse of y, and is called the commutator of x and y. He says that the proof of this theorem, published in [3], caught his interest (see Section 2.3.1). His effort was certainly rewarded, for his program did in fact succeed in proving the commutator theorem. Since this theorem is rather difficult to prove, immediate evidence existed of the power of his program.

Whenever such evidence is encountered, one naturally wonders what feature or features of the new program give the program its power. In particular, is the procedure of naming or the procedure of weighting the key? In Research Problem 22, we focus on naming and its possible value.

The procedure of naming replaces all occurrences of functions with appropriate predicates and constants, constants that name the various constructs referred to in the original functional representation. For example, the axiom of right inverse, which can be written as

P(x,inv(x),e)

for an appropriate function *inv*, is instead written (with ‘‘|’’ denoting **or** and ‘‘¬’’ denoting **not**) as

*This work was supported by the Office of Scientific Computing, U.S. Department of Energy, under Contract W-31-109-Eng-38.

$\neg I(y,x) \mid P(x,y,e)$

to avoid the use of function symbols. For a second example, rather than the term *prod*(*a,b*), naming would employ some new constant, say *d*, instead. Use of naming recasts the problems inherent in nested functions and can sharply reorder the search space of derivable conclusions. On the other hand, its use introduces extra literals, which may sharply detract from the effectiveness of a reasoning program.

The object of Research Problem 22 is to determine criteria for deciding when to *name*. The solution to this problem could permit an automated reasoning program to avoid many of the problems encountered with the more standard approaches. For example, the space of generated clauses to be searched before an answer to some specific question is found might be markedly smaller.

Statistics of the type discussed in Section 5.19 are an appropriate test for a proposed solution to this problem. In particular, one might study various theorems from ring theory and from Boolean algebra. If one wishes to study a hard problem, then one might consider the exponent 3 theorem (Test Problem 9, Section 6.2.2) in ring theory suggested in Section 5.3. That problem asks one to prove that when the cube of every element *x* is *x* in a ring, the ring (multiplication) is commutative. An excellent program for conducting experiments is William McCune's OTTER, see [6] for examples of its use and a diskette of the program, and see [5] for additional examples.

References

1. McCharen, J., Overbeek, R., and Wos, L., 'Complexity and related enhancements for automated theorem-proving programs', *Computers and Mathematics with Applications* **2,** pp. 1-16 (1976).

2. Overbeek, R., 'An implementation of hyper-resolution', *Computational Mathematics with Applications* **1,** pp. 201-214 (1975).

3. Robinson, G., and Wos, L., 'Paramodulation and theorem-proving in first-order theories with equality', pp. 135-150 in *Machine Intelligence 4,* ed. B. Meltzer and D. Michie, Edinburgh University Press, Edinburgh (1969).

4. Wos, L., *Automated Reasoning: 33 Basic Research Problems,* Prentice-Hall, Englewood Cliffs, New Jersey (1988).

5. Wos, L., Overbeek, R., Lusk, E., and Boyle, J., *Automated Reasoning: Introduction and Applications,* 2nd ed., McGraw-Hill, New York (1992).

6. Wos, L., 'Automated reasoning answers open questions', *Notices of the AMS* **5**, 15-26 (January 1993).