The Problem of Hyperparamodulation

LARRY WOS
Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Ave., Argonne, IL 60439, U.S.A.

Abstract. This article is the thirty-second of a series of articles discussing various open research problems in automated reasoning. The problem proposed for research asks one to find a variant of the inference rule hyperparamodulation that avoids generating many of the binary paramodulants ordinarily deduced.

Key words. Automated reasoning, hyperparamodulation, inference rule, unsolved research problem.

Question: What is the appropriate definition of hyperparamodulation that avoids generating all paramodulants? (This question is the ninth of 33 problems proposed for research in [6] and will be referred to as Research Problem 9 throughout this article. All references to sections, chapters, test problems, and such also refer to [6].)

An automated reasoning program employing paramodulation [4, 11] treats equality as ‘‘understood’’ or ‘‘built in’’; this inference rule generalizes the usual notion of equality substitution. A reasoning program using hyperresolution [5] combines a number of small (binary resolution) deduction steps into one. Observing the successes obtained with these inference rules naturally suggested research whose goal was to find an inference rule that would possess the good features of both. The result was the formulation of the inference rule *hyperparamodulation* [8], a rule that combines a number of paramodulation steps into one. Therefore—as we shall see from the example given in this article—hyperparamodulation in effect enables an automated reasoning program to combine a number of equality substitution steps into one deduction step. The technique of identifying the valuable properties of different inference rules, and then seeking a single rule that possesses many or all of the properties, can—as it did with hyperparamodulation—lead to interesting and rewarding research. Specifically, in part because of the cited research, the inference rule negative hyperparamodulation was formulated, and an implementation of (positive) hyperparamodulation was produced [7] that is cleaner than the first version that was published and tested.

So that the nature of Research Problem 9 will be clear—this research problem is far more subtle and complicated than it might appear—let us examine an example of applying one variant of hyperparamodulation before giving a precise statement of the problem. For illustrative purposes only, we shall treat the example in terms of hyperresolution. To successfully apply hyperresolution, the program considers simultaneously a set of clauses such that one of the clauses (the nucleus) contains at least one negative literal, the remaining clauses (the satellites) each contain no negative literals, and the conclusion (the hyperresolvent) contains no negative literals. Whether the program applies the substitutions required to unify the appropriate literal pairs and removes each negative literal from the nucleus one at a time or all at once is not important here. What is important is that no intermediate clauses are considered for possible retention; only the final deduction, free of negative literals, is treated as a new clause to be processed further.

For example, if hyperresolution considers the clauses

left identity:

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

(GT-1) P(e,x,x)

right identity:
(GT-2) P(x,e,x)

left inverse:
(GT-3) P(inv(x),x,e)

associativity:
(GT-5) ¬P(x,y,u) | ¬P(y,z,v) | ¬P(u,z,w) | P(x,v,w)
(GT-6) ¬P(x,y,u) | ¬P(y,z,v) | ¬P(x,v,w) | P(u,z,w)

and, in particular, is applied to clause (GT-5) as nucleus and clauses (GT-3), (GT-3), and (GT-1) as satellites (paired with the negative literals left to right), the clause

P(inv(inv(x)),e,x)

is deduced. If, instead, clause (GT-5) is replaced by clause (GT-6) and clause (GT-1) is replaced by clause (GT-2), then the clause

P(e,x,inv(inv(x)))

is deduced. Although not precisely, the two deduced clauses, in effect, say that the inverse of the inverse of *x* is *x*.

If we apply essentially the same reasoning as is used in either of the preceding two deductions, and use the equality formulation with paramodulation as the inference rule, but do not stop until no *into* terms in the nucleus other than variables remain untouched, we can reach the precise conclusion we are after in four steps.

(GTEQ-1) EQUAL(prod(e,x),x)
(GTEQ-2) EQUAL(prod(x,e),x)
(GTEQ-3) EQUAL(prod(inv(x),x),e)
(GTEQ-5) EQUAL(prod(prod(x,y),z),prod(x,prod(y,z)))

from (GTEQ-3) into (GTEQ-5), into term prod(y,z)
(P-1) EQUAL(prod(prod(x,inv(z)),z),prod(x,e))

from (GTEQ-3) into (P-1), into term prod(x,inv(z))
(P-2) EQUAL(prod(e,z),prod(inv(inv(z)),e))

from (GTEQ-1) into (P-2), into term prod(e,z)
(P-3) EQUAL(z,prod(inv(inv(z)),e))

from (GTEQ-2) into (P-3), into term prod(inv(inv(z)),e)
(P-4) EQUAL(z,inv(inv(z)))

This last deduction differs from the preceding (in which hyperresolution was used) in a number of ways including the deduction of intermediate clauses, clauses (P-1), (P-2), and (P-3). Hyperparamodulation was formulated to yield the clause (P-4) in one step, without yielding any intermediate clauses. Hyperresolution yielded a clause by simultaneously considering clauses (GT-5), (GT-3), (GT-3), and (GT-1), for example. Hyperparamodulation yields clause (P-4) by simultaneously considering clauses (GTEQ-5), (GTEQ-3), (GTEQ-3), (GTEQ-1), and (GTEQ-2). Where hyperresolution removes all negative literals from the nucleus, hyperparamodulation applies equality substitution steps into all terms (other than variables) in the nucleus. With hyperresolution, each negative literal is unified with one positive literal in a satellite; with hyperparamodulation, each nonvariable term is unified with one argument of a satellite.

Research Problem 9 asks for a variant of hyperparamodulation that does not yield all clauses yielded by paramodulation—a variant that avoids generating many of the (binary) *paramodulants* that are ordinarily deduced. The notion is that the absence (in the database of retained clauses) of a large number of binary paramodulants might increase the effectiveness of reasoning programs.

(Throughout this article, we focus on one variant of hyperparamodulation. In contrast to this variant, which requires all nonvariable terms to be unified with an argument of some satellite, the variant published in 1980 [8] requires the user to choose for each nucleus the precise terms into which paramodulation is applied. We mention the earlier variant only so that researchers who are acquainted with it are totally clear about the nature of Research Problem 9.)

Although at first the variant of hyperparamodulation on which we are focusing here might appear to be the sought-after inference rule, in fact, it is not. Except for paramodulants obtained by paramodulating into a variable—and we ordinarily avoid those—this variant permits deducing all binary paramodulants. For example, clause (P-1) will be deduced by applying (this variant of) hyperparamodulation to clause (GTEQ-5) as nucleus and, as satellites, clause (GTEQ-3) and three copies of the clause

(GTEQ-6) EQUAL(x,x)

which is the axiom of reflexivity. The individual paramodulation steps from reflexivity do nothing, of course. More generally, regardless of the choice of nucleus, to have this variant of hyperparamodulation yield a given binary paramodulant, the program simply uses the same *from clause* that was used in the binary paramodulation step, and uses for the remaining satellites the required number of copies of reflexivity. In short, one of the main obstacles to the formulation of the sought-after variant of hyperparamodulation focuses on how reflexivity is to be treated as a satellite. Specifically, when should it be acceptable for one of the paramodulation steps, combined into the single larger hyperparamodulation step, to be a paramodulation *from* reflexivity?

Note that we are not in any way referring to the functional reflexive axioms [4, 11, 10, 2, 3] in this discussion. We in fact recommend against their inclusion even though, in certain odd cases, the lack of those axioms can result in the loss of refutation completeness when using the set of support strategy [1]. Among the functional reflexive axioms are those instances of reflexivity of the form

EQUAL(inv(x),inv(x))
EQUAL(prod(x,y),prod(x,y))

where *inv* and *prod* are functions occurring in the set of clauses under consideration. For each function, the full set of functional reflexive axioms includes a clause of the type just displayed. An example of losing refutation completeness when the set of support strategy is employed and the appropriate functional reflexive axiom is omitted consists of the following four clauses.

Q(x,x)
¬Q(f(a),f(b))
EQUAL(a,b)
EQUAL(x,x)

If the only clause in the set of support is the first of the four, then no proof can be found. If all four clauses are in the set of support, then one can paramodulate from the third into the second to complete a proof. Finally, if the clause

EQUAL(f(x),f(x))

is present, the use of paramodulation quickly leads to a proof; however, two of the steps require paramodulation into a variable.

To understand the role of reflexivity in Research Problem 9 more fully, first recall that, for many theorems of mathematics, two rather different approaches to representation are frequently taken. In one, the equality relation is emphasized, and an inference rule such as paramodulation is employed. In the other, the equality relation is avoided where possible, and inference rules such as UR-resolution [9] and hyperresolution are employed. In a straightforward manner, proofs employing the second notational approach can be mapped to proofs in the first. For example, where the axiom of left identity in a group is represented with the clause

P(e,x,x)

in the second notation, this axiom is represented with the clause

EQUAL(prod(e,x),x)

in the first notation (see Section 6.1 for the axioms for a group in both notations). For a second example, where the associative law, in the notation avoiding (where possible) equality, is written as two clauses

¬P(x,y,u) | ¬P(y,z,v) | ¬P(u,z,w) | P(x,v,w)

and

¬P(x,y,u) | ¬P(y,z,v) | ¬P(x,v,w) | P(u,z,w)

the single clause

EQUAL(prod(prod(x,y),z),prod(x,prod(y,z)))

is used in the notation emphasizing equality.

For a third example—and here we have one of the salient points—the image of the closure (or totality) axiom

P(x,y,prod(x,y))

(where *prod* is an appropriate function) is the clause

EQUAL(prod(x,y),prod(x,y))

under the mapping under discussion. But the clause

EQUAL(prod(x,y),prod(x,y))

which is the functional reflexive axiom for the function *prod*, is merely an instance of the clause for reflexivity

EQUAL(x,x)

and is, therefore, subsumed by the clause for reflexivity. Since, for reasons of efficiency, one does not ordinarily include a clause in the input when a second input clause subsumes it, the image of closure would typically be absent when using the equality-oriented notation. The obvious alternative to this practice is to include the functional reflexive axioms. This action, however, often results in a sharp decrease in the effectiveness of a reasoning program. Therefore, when a proof is mapped from the second notational approach to the first, the use of the clause

P(x,y,prod(x,y))

must be mapped to the use of the clause

EQUAL(x,x)

which brings us to the following summary.

For the inference rule hyperparamodulation, permitting unrestricted paramodulation from reflexivity generates all (binary) paramodulants; completely blocking such applications of paramodulation, on the other hand, presents the obstacle of a possible loss of refutation completeness because reflexivity is in part the image of the closure axiom in representations that do not emphasize equality. An examination of various proofs employing hyperresolution reveals the frequent use of a closure axiom as one of the satellites. For example, in the proof of the theorem that states that Boolean rings (rings in which the square of x is itself) are commutative (Test Problem 8, Section 6.2.1), one of the steps commonly used is the deduction of the fact that $x(x+x) = x+x$. That deduction, in the notation that does not emphasize equality, uses two occurrences of the closure axiom as a satellite. Therefore, if one maps proofs—in particular, the proof that Boolean rings are commutative—from one notation to the other as illustrated here, the variant of hyperparamodulation in focus throughout this article will occasionally include steps that correspond to paramodulating from reflexivity.

We can provide additional insight for this problem with the following observations. In most cases, any negative literals that must be removed to complete an application of hyperresolution can be removed with the closure axiom. The exceptions occur when, for example, some constant has replaced a variable and, therefore, the corresponding unification fails. In contrast, when attempting to complete an application of (this variant of) hyperparamodulation, and a nonvariable term must be considered for an equality substitution, reflexivity can be used; the corresponding unification never fails. Since some applications of hyperresolution, therefore, cannot be completed but all applications of hyperparamodulation can always be completed, we encounter an additional complication. However, a thorough analysis of this difference might provide an important clue to solving Research Problem 9.

To test a proposed solution to Research Problem 9, one might study the theorem from ring theory just discussed, where the representation is in terms of equality unit clauses. The exponent 3 theorem (Test Problem 9, Section 6.2.2) from ring theory cited in the preceding section, if phrased as equality unit clauses, is a more significant test for a proposed solution to this problem. The latter suggestion provides a better test for a proposed solution to Research Problem 9 because that test problem is substantially harder, causing the reasoning program to deduce a large number of clauses when attempting to solve it. Among a large set of deduced clauses, one is more likely to find different variations of the use of the image of closure.

References

1. Anderson, R., ''Completeness results for E-resolution'', pp. 653-656 in *Proceedings of the AFIPS 1970 Spring Joint Computer Conference, Vol. B36* (1970).

2. Brand, D., ''Proving theorems with the modification method'', *SIAM J. Computing* **4,** pp. 412-430 (1975).

3. Peterson, G. E., ''A technique for establishing completeness results in theorem proving with equality'', *SIAM J. Computing* **12,** pp. 82-100 (1983).

4. 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).

5. Robinson, J., ''Automatic deduction with hyper-resolution'', *International J. Computer Math.* **1,** pp. 227-234 (1965).

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

7. Wos, L., and McCune, W., ''Negative hyperparamodulation'', pp. 229-239 in *Proceedings of the Eighth International Conference on Automated Deduction, Springer-Verlag Lecture Notes in Computer Science, Vol. 230,* ed. J. Siekmann, New York, Springer-Verlag (1986).

8. Wos, L., Overbeek, R., and Henschen, L., ''Hyperparamodulation: A refinement of paramodulation'', pp. 208-219 in *Proceedings of the Fifth International Conference on Automated Deduction, Springer-Verlag Lecture Notes in Computer Science, Vol. 87,* ed. R. Kowalski and W. Bibel, Springer-Verlag, New York (1980).

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

10. Wos, L., and Robinson, G., ''Maximal models and refutation completeness: Semidecision procedures in automatic theorem proving'', pp. 609-639 in *Word Problems: Decision Problems and the Burnside Problem in Group Theory,*ed. W. Boone, F. Cannonito, and R. Lyndon, North-Holland, New York (1973).

11. Wos, L., and Robinson, G., ''Paramodulation and set of support'', pp. 276-310 in *Proceedings of the IRIA Symposium on Automatic Demonstration, Springer-Verlag Lecture Notes in Mathematics, Vol. 125,* Springer-Verlag, New York (1970).