Effective Set Membership in Computer Algebra and Beyond

James H. Davenport

Department of Computer Science
University of Bath, Bath BA2 7AY, United Kingdom
J.H.Davenport@bath.ac.uk

Abstract. In previous work, we showed the importance of distinguishing "I know that $X \neq Y$" from "I don't know that $X = Y$". In this paper we look at effective set membership, starting with Gröbner bases, where the issues are well-expressed in algebra systems, and going on to integration and other questions of 'computer calculus'.

In particular, we claim that a better recognition of the role of set membership would clarify some features of computer algebra systems, such as 'what does an integral mean as output'.

1 Introduction

In [7] we discussed the various ideas of equality that can be found in computer algebra, and showed the importance of distinguishing "I know that $X \neq Y$" from "I don't know that $X = Y$". In this paper (a fuller version of which is in [8]) we look at effective set membership. While sets can be defined in a variety of ways, we will be interested in sets defined as

$$S := \{x \in A \mid P(x)\} \tag{1}$$

where A is a set for which membership is "obvious", e.g. by construction, and P is some predicate, which will generally involve some existential quantifiers. The problem of **effective set membership**, then, is the following problem.

Problem 1. *Given some $x \in A$, produce*

either *an* effective *[5] proof of $P(x)$*
or *a proof of $\neg P(x)$.*

In general, it is the second part of the problem that is the hard one.

2 Ideals and Gröbner Bases

The classic definition of an *ideal* $(p_1, \ldots, p_m)$ in $k[x_1, \ldots, x_n]$ as

$$(p_1, \ldots, p_m) = \left\{ \sum_{i=1}^{m} f_i p_i : f_i \in k[x_1, \ldots, x_n] \right\} \tag{2}$$

means that exhibiting the f_i becomes a proof of **either**. But how to do so, and how to prove **or**? So in this case problem 1 becomes the following.

Problem 2. *For given $p_1, \ldots, p_m$ and given f*

either *exhibit $f_i \in k[x_1, \ldots, x_n]$ such that $f = \sum_{i=1}^m f_i p_i$*
or *demonstrate that none such exist.*

We have, of course, the process of polynomial reduction.

Algorithm 1 (Polynomial Reduction). *[1, Algorithm REDPOL]*
Input: *$f, p_1, \ldots, p_m \in k[x_1, \ldots, x_n]$, a monomial order $>$*
Output: *$\hat{f}, f_1, \ldots, f_m \in k[x_1, \ldots, x_n]$:*
$\hat{f} = f - \sum_{i=1}^m f_i p_i$: $\quad$ $\hat{f}$ irreducible by the p_i (with respect to $>$)

Clearly, *if* this process terminates with $\hat{f} = 0$, we have proved the **either**, and we have the f_i.

Theorem 1 (Buchberger [4]). *If the p_i are a Gröbner basis, then Algorithm 1 precisely solves problem 2, i.e. $\hat{f} \neq 0$ is a* proof *that $f \notin (p_1, \ldots, p_m)$.*

Since being a Gröbner basis is algorithmically testable, we have a complete process for solving problem 2 *if* we are given a Gröbner basis. Furthermore, Buchberger's algorithm lets us compute a Gröbner basis for any polynomial ideal starting from *any* finite set of generators. We have come to expect this of computer algebra systems, and would be rather surprised to see a system take a set of polynomial equations and just say "I can't solve these".

3 Integration in Elementary Terms

The problem of (indefinite) integration is not normally viewed as a set membership problem, but it can be. We refer the reader to [3] for the standard definitions, and we let $\mathcal{I}$ be some class of functions (elementary, Liouvillian, $\mathcal{EL}$ [14] etc.). When we say "given an $\mathcal{I}$ function", we mean that it is given *effectively*, i.e. it is given as a member of an effective field of $\mathcal{I}$ functions. Then the set of $\mathcal{I}$-integrable functions is

$$\{f \mid \exists g \in \mathcal{I} \quad g' = f\}$$

and the $\mathcal{I}$-integration problem (as perceived since [11,13]) becomes

Problem 3. *For given f* (normally $f \in \mathcal{I}$)

either *exhibit $g \in \mathcal{I}$ such that $f = g'$*
or *demonstrate that no such g exists.*

It was not always thus: [15] essentially perceived the problem as

Problem 4. *For given f*

either *exhibit $g \in \mathcal{I}$ such that $f = g'$*
or *return* `failed` *(g might exist, but hadn't been found),*

and a successful program was one which did not return `failed` when a freshman could see the answer.

The shift from problem 4 to problem 3 was essentially due to the rediscovery of Liouville's Theorem [10], which, in the case $\mathcal{I}$="elementary", reduced problem 3 to the following.

Problem 5. *For given f in a differential field K*

either *exhibit f as $v_0' + \sum_{i=1}^{n} c_i \frac{v_i'}{v_i}$, with $v_0 \in K$, $c_i \in C = \overline{\{g \in K \mid g' = 0\}}$, $v_i \in CK$;*
or *prove that no such decomposition exists.*

When K is purely transcendental over its field of constants, this problem is soluble [13] and generally implemented[1]. Hence, when such a system returns an unevaluated integral, this *should be* a proof that no such elementary integral exists. However, the documentation may not say so: for example Maple 11 says merely

> If Maple cannot find a closed form expression for the integral, the function call is returned.

When K is algebraic, the problem is solved in principle [2], but not completely implemented. Hence, when such a system returns an unevaluated integral, this can mean any of:

1. there is no elementary integral, i.e. the [**or**] of problem 5;
2. the implementation is fundamentally inadequate, e.g. Reduce's integrator uses [6], which only works for quadratic algebraic functions of x;
3. the implementation has attempted to address the question, but has failed, which may be reported as "implementation incomplete"; ([9] reports this of Axiom), or just as an unevaluated integral;
4. the implementation may be of some (theoretically[2]) weaker algorithm, without a proof of completeness.

In general, the user does not know which of these applies, and the standard notation of computer algebra provides no convenient way of telling the user, though a warning (on the lines of the error reported in case (3) above) would at least be useful.

4 Other Areas

There are other areas in which set membership problems are, at least in principle, decidable. One obvious example is the solution of differential equations in terms of Liouvilian functions [16]. Again, it is not clear how these decision procedures can be effectively 'sold' to the user.

[1] Subject to undecidability problems over constants [12]. This is an important caveat in principle, but less so in practice, and we shall ignore it from now on.
[2] It may be stronger in practice, however, as reported in [9].

5 Conclusions

In one area of computer algebra (polynomial ideals) we are now used to the fact that we have a decision procedure for set membership, and would be surprised if anything other than a clear-cut answer were obtained. Elsewhere, e.g. integration, we have decision procedures, but the user community is apparently willing to settle for not knowing whether a decision procedure has been applied or not. Put bluntly, the user, no matter how expert, has no way of knowing what an unevaluated integral means, and in many ways the situation has gone back to the user expectations of [15], where we are merely asking "can the software find any answer".

References

1. Becker, T., Weispfenning, V., Kredel, H.: Groebner Bases. A Computational Approach to Commutative Algebra. Springer, Heidelberg (1993)
2. Bronstein, M.: Integration of elementary function. J. Symbolic Comp. 9, 117–173 (1990)
3. Bronstein, M.: Symbolic Integration I, 2nd edn. Springer, Heidelberg (2005)
4. Buchberger, B.: Ein Algorithmus zum Auffinden des basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. PhD thesis, Math. Inst. Universität Innsbruck (1965)
5. Davenport, J.H.: Effective Mathematics — the Computer Algebra viewpoint. In: Richman, F. (ed.) Proceedings Constructive Mathematics Conference 1980 [Springer Lecture Notes in Mathematics 873], pp. 31–43. Springer, Heidelberg (1981)
6. Davenport, J.H.: On the Integration of Algebraic Functions. Springer Lecture Notes in Computer Science vol.102 (1981)
7. Davenport, J.H.: Equality in computer algebra and beyond. J. Symbolic Comp. 34, 259–270 (2002)
8. Davenport, J.H.: Effective Set Membership in Computer Algebra and Beyond. Technical Report CSBU–2008–03, Dept. Computer Science, University of Bath (2008), `http://www.cs.bath.ac.uk/department/technical-report-series/technical-report-series/index.php`
9. Kauers, M.: Integration of Algebraic Functions: A Simple Heuristic for Finding the Logarithmic Part(2008), `http://www.risc.uni-linz.ac.at/publications/download/risc_3390/main.pdf`
10. Liouville, J.: Mémoire sur l'intégration d'une classe de fonctions transcendantes. Crelle's J., 13, 93–118 (1835)
11. Moses, J.: Symbolic Integration. PhD thesis M.I.T., & Project MAC TR-47 (1967)
12. Richardson, D.: Some Unsolvable Problems Involving Elementary Functions of a Real Variable. Journal of Symbolic Logic 33, 514–520 (1968)
13. Risch, R.H.: The Problem of Integration in Finite Terms. Trans. A.M.S. 139, 167–189 (1969)
14. Singer, M.F., Saunders, B.D., Caviness, B.F.: An Extension of Liouville's Theorem on Integration in Finite Terms. SIAM J. Comp. 14, 966–990 (1985)
15. Slagle, J.: A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus. PhD thesis, Harvard U. (1961)
16. van Hoeij, M., Ragot, J.-F., Ulmer, F., Weil, J.-A.: Liouvillian Solutions of Linear Differential Equations of Order Three and Higher. J. Symbolic Comp. 28, 589–609 (1999)