On algebraic simplification

J. P. Fitch

Computer Laboratory, University of Cambridge, Corn Exchange Street, Cambridge CB2 3QG

After a brief review of the current state of simplification this paper proposes a classification of simplification rules that may help in the practical implementation of simplification procedures. The last part of the paper is concerned in the mathematical theory of simplification, and the set of expressions formed from rational powers of polynomials is shown to have a canonical form.

(Received August 1972)

1. The central problems

It is well known that Richardson (1966), following the results of Davis, Putman and Robinson (1961), showed that there is no algorithm that can decide in a finite number of steps if an arbitrary expression from a sufficiently rich class of functions is identically zero. Richardson's class is built from the elementary constants; that is the smallest set of real numbers forming a closed field that contains 1 and π, and is also closed under the operation of exp (a), log $|a|$ and sin (a). He allows the action of addition, multiplication, division, substitution, sine, exponentiation and log modulus on these constants, together with the quantities x and log 2. This class is called the elementary function class. Recently Matijasevic (1970) has shown Hilbert's tenth problem to be undecidable, and so it can be shown that Richardson's class can be refined by removing log 2 and the log modulus function, and replacing the exponential function by the absolute value function (Caviness, 1970). As most algebraic manipulators include both these classes within their terms of reference, Richardson's theorem represents a theoretical restriction on all such systems. However, it is equally well known that there is a canonical representation for polynomials over the complex rational numbers, and Caviness (1970) has shown that there is a canonical form for the first order exponential expressions; that is polynomials over the complex rational numbers, together with the unnested exponential function.

Two fields of investigation therefore arise. The first involves the problem of discovering for a defined class of expressions whether we can decide if an arbitrary expression is identically zero, and if we can, of exhibiting the decision algorithm. This field of investigation includes the discovery of normal and canonical forms for expression classes. The second field is the practical problem of producing feasible algorithms or heuristics that can effect some improvement in the simplicity of general expressions. The first of these studies has produced little in the way of concrete facts as yet, but many conjectures. As well as the concrete results of Richardson and Caviness mentioned above we have the work of Richardson (1966), Johnson (1970), Brown (1969), and Caviness (1970). Both Richardson and Johnson have produced algorithms that reduce deciding if an expression is zero to deciding if a constant is zero. By assuming the algebraic independence of π and e, Brown was able to produce a normal form for rational exponential expressions, a normal form being one where zero is unique. Caviness, assuming the transcendence of e^e, produced a canonical form for polynomial expressions formed from arbitrarily nested exponentials. A canonical form is one where every expression is unique. In addition, there is an existence lemma of Caviness, Pollack and Rubald (1971) that states that if there exists a normal form, there is a canonical form, provided that subtraction is allowed. In Section 6 we extend these results a little further.

The second field has proved a much more prolific subject, with programs ranging from formalised systems (Fenichel, 1966; Fateman, 1971) to *ad hoc* methods. Some programs are free standing simplifiers; notable here are FAMOUS (Fenichel, 1966) and the early work of Hart (1961) and Wooldridge (1963). Others, such as those of Tobey, Bobrow and Zillies (1963) in FORMAC, Korsvold (1965) in SCRATCHPAD and MATHLAB, and Fitch (1971) in CAMAL, are embedded in general algebraic manipulators.

Before considering methods of simplification we first clarify why we wish to simplify, and consequently when we should simplify.

2. Why simplify?

When considering the timing and frequency of attempts to simplify an expression, it is necessary to bear in mind that there are at least three reasons for wanting to simplify, which while not being completely distinct, can be conveniently separated. These three reasons are:

1. To make the expression smaller in store, and to speed subsequent calculation. I will call this the problem of *compactness*, and the process *compactification*.
2. To put an expression into an intelligible form for printing, or otherwise presenting to the user. This will be called the problem of *intelligibility*.
3. To see if an expression is identically zero. This has already been called the *identity* problem (Richardson 1966).

Collectively these will be called the *simplification* problem.

Perhaps some justification is needed for differentiating between compactness and intelligibility. Consider the trigonometrical functions. One way of including all the trigonometrical identities is to map each such expression into its complex exponential form, using the identities

$$\sin \theta \equiv \frac{1i}{2}(e^{i\theta} - e^{-i\theta})$$

and

$$\cos \theta \equiv \frac{1}{2}(e^{i\theta} + e^{-i\theta}) .$$

Then the rules

$$\forall\, x, y \qquad e^x e^y \Rightarrow e^{x+y} ,$$

$$\forall\, x, y \qquad (e^x)^y \Rightarrow e^{xy} ,$$

$$e^0 \Rightarrow 1 ,$$

and

$$e^{i\pi} \Rightarrow -1 ,$$

embrace all the usual trigonometrical identities. However, a user would not normally be happy with the result

$$\frac{2 \exp [i\theta]}{(1 + \exp [i\theta])^2}$$

if he expects the equivalent expression

$$\operatorname{cosec}^2 (\theta) - \cot (\theta) \operatorname{cosec} (\theta) .$$

It seems likely that the requirements of compactification may demand the exponential form, while intelligibility asks for the more usual trigonometrical expressions.

We can use this division of intelligibility and compactification to throw some light on the question 'what is the simplest form of an expression?' This question posed by Fenichel (1966), reflects the difficulty of arranging an expression in an intelligible form. However, in the internal representation that concerns compactification, we can impose conditions of minimum store, or of avoiding structures that will lead to difficulty in subsequent calculation, or some amalgam of these based on the parameters of the system.

In trying to define an intelligible expression, we are thrown back on to such vague phrases as 'that which is acceptable to the user'. A practical implementation of this idea is of course difficult, but presumably it would ensure that the answer is expressed in terms similar to those of the problem. For example, if a factor $(x + y^2)$ occurs extensively in a question, it is a reasonable assumption that the user will prefer this factor to be exhibited in the answer. This in turn may mean some alteration to the algorithm for compactification because factorisation is so difficult. A similar problem is discussed in detail by Hearn (1969).

These remarks indicate that we should simplify for a number of different reasons. We should simplify before printing, to put the expression in an intelligible form; we should simplify to control intermediate expression swell, and we should simplify to solve the identity problem. Of course we may wish to make an expression into a canonical form for any of these reasons.

3. Classification of transformations

In this section we divide the identities which are used as simplification rules into three classes. The first class contains those transformations which can always be applied automatically, without appealing to a decision process, the second contains rules which in certain well defined circumstances are appropriate, and the third is the collection of all other transformations whose usefulness is ill-defined. It is not always easy to assign a given rule to one of the classes, but certain guidelines can be laid down.

The first class is easy to understand if we consider algebraic expressions as a ring A over a ground field F, when the elements of the field exhibit a canonical form. Class 1 then reflects the existence of the zero element and the unit element in this ring, and the structure of the ground field. **Fig. 1** is a summary of some of the more obvious identities in Class 1.

It should be understood that these rules are to be applied in a strict left to right manner, and that in this figure the symbols '0' and '1' stand for the explicit representations of zero and unity in A, not the identity class of zero and unity. This last point avoids the difficulty voiced by Fenichel (1966), namely that we cannot recognise zero. The reasons for saying that these rules should be applied mandatorily are fundamentally those of Sammet (1967), that the keeping of extra ones and zeros when trying to do large scale tedious algebra is misleading and wasteful. It is here contended that these Class 1 rules should be built into the basic system, preferably in the simple addition and multiplication routines.

As has been remarked in Section 1, for certain classes of expressions the identity problem is soluble, and algorithms exist that enable expressions from these classes to be reduced to a canonical form. Class 2 contains the simplification rules that are necessary to implement just these algorithms, and rules in this class are to be applied only if the expression belongs to one of these decidable classes. It is precisely the rules from Class 2 that are utilised in polynomial systems, in Poisson systems, or in first order exponential systems. Some Class 2 rules are shown in **Fig. 2.** The same comments about application from left to right apply as to Fig. 1.

$\forall x \in A$	$x + 0 = x$
$\forall x \in A$	$x.1 = x$
$\forall x \in A$	$x.0 = 0$
$\forall x \in A, x \neq 0$	$x^0 = 1$
$\forall m, n \in F$	$m + n = (m + n)$
$\forall x \in A$	$-(-x) = x$
$\forall x \in A$	$x - x = 0$
$\forall x, y \in A$	$x^y . x^{-y} = 1$

Fig. 1. Some Class 1 transformations

$\forall x, y, z \in$ polynomials $x(y + z) = xy + xz$
$\forall x, y \in$ polynomials $x(y) = xy$
$\forall x, y \in$ polynomials $\exp[x] \exp[y] = \exp[x + y]$
$\forall x, y \in$ angles $\sin[x] \sin[y] = (\cos[x - y] - \cos[x + y])/2$
$\forall x, y \in$ angles $\sin[x] \cos[y] = (\sin[x + y] + \sin[x - y])/2$
$\forall x, y \in$ angles $\cos[x] \cos[y] = (\cos[x + y] + \cos[x - y])/2$

Fig. 2. Some Class 2 transformations

Class 3 is the class of all remaining identities in algebraic expressions. There may be classes of expressions which have a presently unknown canonical form. In this case, by the definition of Classes 2 and 3, the rules that operate this algorithm are in Class 3. When the algorithm is discovered, the appropriate rules can be transferred to Class 2. It should be noted that Class 3 is very large. In particular it is not recursive, containing such identities that, if known, would solve the identity problem for any given expression. In more detail, if we wished to know if an expression E were identical to zero, and it were, the rule

$$E \equiv 0$$

would be in Class 3. If E were non zero, then the rule would not be in any class. Finally the Class 2 rules must be taken with Class 3 for the expressions for which we do not have a canonical form, as they may yet be useful. We will consider this overlapping of classes in the next section.

4. User control over simplification

Consider a problem that contains only polynomials in n variables, z_i and $\sin(x)$ and $\cos(x)$. Then at least for calculations involving only addition, multiplication, division without remainder, differentiation, and restricted integration, the rule

$$\cos^2(x) \Rightarrow 1 - \sin^2(x)$$

can be used to produce a canonical form, were all even powers of $\cos(x)$ are replaced by an equivalent expression in $\sin(x)$. Thus for this expression class, this rule is a Class 2 rule. However, for a wider class of expressions it belongs to Class 3. Assuming that we do not want to write a special system to deal with this restricted set of functions, we must be able to adapt a more general simplification system to operate effectively for problems that permit this particular canonical form.

An entirely different difficulty is found in the familiar problem of putting an expression over a common denominator. This can be expressed by the Class 3 rule

$$\forall x, y, z, t \quad \frac{x}{y} + \frac{z}{t} = \frac{xt + yz}{yt}$$

or better by

$$\forall x, y, z, t \quad \frac{x}{y} + \frac{z}{t} = \frac{x \operatorname{lcm}(y, t)/y + z \operatorname{lcm}(y, t)/t}{\operatorname{lcm}(y, t)}$$

In many problems this rule is helpful, but in others it may be very retrograde, destroying a simple structure. A wrongful application of this rule could be extremely expensive in computer time, as partial fractional decomposition is not easy. In such cases the user needs the ability to control or inhibit the application of the rule.

From these two examples we can see that useful systems ought to be adaptable to differing types of problems. To facilitate this adaptation, the user must have some control over the simplification that is attempted by the system. Of course many existing algebra systems of the new left political classification of Moses (1971) have some degree of control. SYMBAL (Engeli, 1968) for example, has a vector of variables called MODE that controls such things as the removal of brackets and the truncation of series; REDUCE (Hearn, 1971) allows the introduction of simplification rules by the LET construction; while CAMAL (Fitch, 1971) has controls for the distribution of brackets and the ability to give simplification rules that can be either compulsory or optional to the simplify subroutine. What is advocated here, in the light of the two examples given above and of the separation of compactification and intelligibility is a new party to add to those of Moses; a breakaway group from the new left, with a flavour of the catholic party, a kind of International Socialists. Not only would this party provide the MODE-like controls, but also an extension of the LET command of REDUCE, that allows the specification of rules to include a statement as to whether it is to be considered as Class 2 or 3, and an indication of whether a rule is to be used for intelligibility or compactification or both. This can be related more closely to Moses' political classification of systems. We may identify the radical party with systems that always apply Class 1 and a subset of Class 2, while the conservatives only apply rules from any class that have explicitly been given. The liberals apply Class 1, but must have Class 2 and Class 3 rules specified. The new left implement Class 1 and some of Class 2, and allow other rules to be given, usually to supplement Class 2. In addition they allow control over the application of some of the built-in Class 2 rules. The new international socialist party take account of the separation of compactification and intelligibility. While remaining new left for internal work, they approach catholicism when propagandising.

The problems that are introduced by these remarks are the real ones of how to give syntactic descriptions of the various types of rule, while avoiding a proliferation of ad hoc notations and methods. One approach that might make this easier is to provide packages of simplification procedures that can be supplied to the user with names, like the switches in SCRATCHPAD (Greismer and Jenks, 1971) for example. This would require facilities for users to create new packages and to assimilate these into the algebraic manipulator.

5. The identity problem and point evaluation

Most theoretical work so far has been directed towards the identity problem because it is a more definite problem than either compactification or intelligibility, as it only requires the answer yes or no. There are two mathematical techniques; one due to Richardson (1966) and one due to Johnson (1971). Richardson's method applies to elementary expressions that are totally defined in a finite interval; that is, having no singularities in that interval. By using an induction technique on the complexity of the expression Richardson finally reduces the problem to that of recognising the constant zero. This method is only of practical use if it is possible to decide whether the constants produced from the derived expressions when a rational number is substituted for the algebraic variable are zero, and it is known that the expression has no singularities for physical or other reasons. However, as it is not known if e^e is rational, it seems possible that this algorithm may not be much help. Despite this it would be interesting to see how this algorithm fares in practical problems. Johnson's algorithms are based on the study of eigenvectors of certain transformations such as differentiation, and rely on being able to recognise the eigenvectors. We are then back at the constant problem.

However, there is another group of methods that are available in the identity case which are of little or no value in compactification and intelligibility. That is the use of numbers in *point evaluation*: evaluating the expression at a point. Mathematically this can be viewed as a homomorphism from the field of algebraic expressions onto some numerical field which possesses a canonical form. There are two obvious choices for the numerical field: a finite field of integers and the pseudo field of floating point numbers. The former of these methods has been developed in detail by Martin (1971), under the name of *hash coding*. The principle of application is the same in both cases. One maps simple sub-expressions into the numerical field and then does the arithmetic. The hope is that a zero result from this mapping implies a zero expression, and a non zero result implies a non zero expression. However, as the numerical field is simpler there must be some loss of information, and we must treat the answer with care. It is usual to arrange the mapping so that a non-zero result is taken as true, and a zero result is verified by other techniques. One could use a variety of such mappings to reduce the probability of a misleading answer. Martin concluded that because of overflow, underflow and rounding errors in floating point calculations, a finite field was better. Unfortunately, his search for a suitable field and mapping was less than successful, and he did not consider the time spent doing the finite field arithmetic to be significant. Floating point numbers, on the other hand, while not forming a field exactly, do have the advantage of speed (on most computers), but we are left with rounding errors and underflow. Rounding error might be calculable by utilising interval analysis (Moore, 1966), this being a method of point evaluation that deserves more attention. Giving a fixed error bound is destined to failure, but the variable error of interval methods is less likely to mislead. Preliminary experiments by Wittig in Cambridge have indicated that overflow and underflow are very rare and that the interval techniques have possibilities.

6. Some theorems on canonical forms and the identity problem

In Fitch (1971) an investigation of certain metrics of general relativity due to Harrison (1959) is described. These calculations involve expressions formed from polynomials over the field Q of rational numbers raised to rational powers. The first two theorems show that the constant problem for this class of expressions is soluble, and the solution is given by explicitly exhibiting a canonical form and giving the canonicalising algorithm.

Theorem 1

The class of constants formed from rational powers of rational numbers, by addition, subtraction, multiplication and division, has a canonical form.

Proof

We are concerned with expressions of the form

$$\frac{\sum_{i=0}^{N}\left(\prod_{j=0}^{M_i} R_{ij}^{r_{ij}}\right)}{\sum_{k=0}^{K}\left(\prod_{l=0}^{L_k} S_{kl}^{t_{kl}}\right)} \quad R_{ij}, S_{kl}, r_{ij}, t_{kl} \in Q \qquad (1)$$

We can assume without loss of generality, that R_{ij} and S_{kl} are prime integers, for, if they are not, we can factor them. By introducing zero powers of primes, we can ensure that all the M_i and L_k are equal to some M, and the same primes occur in each multiplicative term. By introducing rational coefficients we can insist that all the r_{ij} and t_{kl} satisfy

$$0 < r_{ij}, t_{kl} < 1$$

Thus we have written the expression (1) in the form

$$\frac{\sum_{i=0}^{N}\left(q_i \prod_{j=0}^{M} p_j^{r_{ij}}\right)}{\sum_{k=0}^{K}\left(s_k \prod_{j=0}^{M} p_j^{t_{kl}}\right)} \quad \begin{array}{l} r_{ij}, t_{kj}, q_i, s_k \in Q \\ p_i \text{ prime} \in Z^+ \\ 0 \leqslant r_{ij}, t_{kl} < 1 \end{array} \tag{2}$$

where for each prime p_j there is at least one r_{ij} or t_{ij} which is not zero. Now suppose that

$$r_{ij} = \frac{m_{ij}}{n_{ij}} \text{ and } t_{ij} = \frac{u_{ij}}{v_{ij}}, u_{ij}, v_{ij}, m_{ij}, n_{ij} \in Z$$

and these are in their simplest form. Let $L_j = \underset{i}{\text{lcm}}\,(n_{ij}, v_{ij})$. L_j is not 1 for any j because $r_{ij}, t_{ij} < 1$, and not all zero. Now we consider the field of rational numbers extended by each $(p_j)^{1/L_j}$, forming

$$PQ = Q(p_0^{1/L_1}, p_1^{1/L_2}, \ldots, p_M^{1/L_M}) .$$

This is an algebraic extension of degree $N' = \prod_{j=0}^{M} L_j$, and has a basis that is the product of bases of the fields $Q(p_0^{1/L_0}), \ldots, Q(p_M^{1/L_M})$. That is,

$$B = \{p_0^{N_0/L_0} P_1^{N_1/L_1} \ldots, p^{N_M/L_M} \,|\, 0 \leqslant N_j \leqslant L_j - 1, \; j = 0, \ldots, M\}$$

is a basis (see Postnikov 1962). Any element of B can be written as

$$b_i = \prod_{j=0}^{M} p_j^{S_{ij}/L_j}, 0 \leqslant S_{ij} \leqslant L_j - 1$$

As the expression (2) is a member of the field PQ, it can be expressed in terms of the basis,

$$\sum_{l=0}^{N} W_i \left(\prod_{j=0}^{M} p_j^{S_{ij}/L_j}\right), 0 \leqslant S_{ij} \leqslant L_j - 1, W_i \in Q \tag{3}$$

By the properties of a basis any expression in the form (1) can be written in a unique form on the pattern of (3) as any finite expression can only require a finite number of extensions. The extensions are characterised by the pairs (p_i, L_i), and so can be ordered. Within each extension the expressions can be ordered by the coefficients w_i. Thus the expression (3) constitutes a canonical form and order for expressions of the form (1).

Theorem 2
The constant problem for rational powers of rational numbers is decidable.

Proof
We need to produce a finite algorithm that will produce the canonical form (3) from expressions of the type of (1). We begin by transforming the expression to the form (2), an operation that involves factorisation. Then both the numerator and denominator separately are in the form of (3). We replace

$$\frac{1}{\sum_{k=0}^{K} s_k \left(\prod_{j=0}^{M} p_j^{t_{kj}}\right)}$$

by an expression of the form (3). We then multiply the numerator by this expression, and reduce the powers of the p_j to ensure they are less than one, by absorbing the extra p_j into the rational coefficients if necessary.

Thus we need algorithms to factor an integer into its prime factors, and to replace $\frac{1}{\alpha}$ by β, where $\alpha, \beta \in PQ$, and $\alpha\beta = 1$.

The first of these algorithms is obviously possible by a finite trial and error method. It is the second algorithm that presents difficulties. It will be shown that a polynomial in $p_0^{1/L_0}, p_1^{1/L_1}, \ldots, p_M^{1/L_M}$ can be inverted by an inductive method on M.

Consider $g(\theta)$ as a polynomial in $\theta = p_M^{1/L_M}$. Let $L_M = \lambda$ and let $Q_M = Q(p_0^{1/L_0}, p_1^{1/L_1}, \ldots, p_{M-1}^{1/L_{M-1}})$; then $g(\theta)$ is a polynomial with coefficients in Q_M, and

$$PQ = Q_M(\theta)$$

and θ is a root of

$$x^\lambda - p_M = 0 \tag{4}$$

The induction assumption is that elements of Q_M can be inverted. The conjugates of θ are $w\theta, w^2\theta, \ldots, w^{\lambda-1}\theta$, where $w^\lambda = 1$. Now let us consider

$$G = g(\theta)\, g(w\theta) \ldots, g(w^{\lambda-1}\theta) .$$

This is symmetric in the conjugates, and thus when this expression is multiplied out, the conjugates appear as the sum of products of the conjugates taken r at a time: that is the coefficient of $x^{\lambda-r}$ in (4), and so G is independent of θ and w, and is therefore a member of Q_M. As

$$\frac{'1}{g(\theta)} = \frac{1}{G}\, g(w\theta)\, g(w^2\theta) \ldots, g(w^{\lambda-1}\theta)$$

we have expressed $\frac{1}{g(\theta)}$ in terms of $\frac{1}{G}$ and a polynomial in θ.

By the induction hypothesis, we can invert $G \in Q_M$. The induction process starts because if $M = 0$, then G is a rational number, for which inversion presents no difficulties. So we can invert an element of PQ in $M + 1$ steps.

Thus we have produced an algorithm to transform an expression of the form (1) to the canonical form (3).

Theorem 3
The class of expressions formed from rational powers of polynomials in n variables over the rational numbers by addition, subtraction, multiplication and division has a canonical form.

Proof
We remark that monic irreducible polynomials in the field $Q(x_0, \ldots, x_{n-1})$ are analogous to primes in the field Q. Thus, as $Q(x_0, \ldots, x_{n-1})$ has unique factorisation and a canonical form, Theorem 3 is proved by rewriting the proof of Theorem 1 with $Q(x_0, \ldots, x_{n-1})$ written for Q, and monic irreducible written for prime.

Theorem 4
The identity problem for rational powers of polynomials over the rational numbers is decidable.

Proof
As in Theorem 2 we need algorithms for factorisation and for inversion to produce the canonical form.

The factorisation algorithm of Kronecker (van der Waerden, 1949), extended by Jordan, Kain and Clapp (1964) is sufficient for that purpose. The inversion algorithm given in the proof of Theorem 2 is still valid, if we make textual changes to replace Q by $Q(x_0, \ldots, x_{n-1})$.

Thus an algorithm exists for reducing to canonical form polynomials over Q raised to rational powers, and consequently the identity problem for these expressions is decidable.

7. Possible extensions to these canonical forms

We can try to extend these canonical forms in a variety of ways. One possibility is to allow irrational powers as well as rational powers. When this is done in the constant case, of Theorem 1, we can apply the Gelfond-Schneider theorem (Niven, 1956) to produce a canonical form for $Q(n^\alpha)$ for some $n \in Q$ and an algebraic irrational α, but to extend by two such numbers is an unsolved problem of transcendental number theory (Schneider, 1957; Gelfond, 1949). Similar problems arise with algebraic powers of polynomials.

One would hope that the canonical form could be extended to polynomials to polynomial powers. This has not yet been done, but the following lemma, due to M. N. Huxley, extends the canonical form to some further cases.

Lemma
$(1 + x)^{1+x}$ is transcendental over $Q(x)$

Proof
Let $f = (1 + x)^{1+x}$. This is well defined as a formal power series in x. Suppose f satisfies the equation

$$a_n f^n + a_{n-1} f^{n-1} + \ldots, + a_0 = 0$$

for some n, and $a_i \in Q(x)$.

Now consider $f_0 = f(\sqrt{2} - 1) = (\sqrt{2})^{\sqrt{2}}$, which is transcendental by the Gelfond-Schneider theorem f_0 must satisfy

$$a_n(\sqrt{2} - 1) f_0^n + a_{n-1}(\sqrt{2} - 1) f_0^{n-1} + \ldots, + a_0(\sqrt{2} - 1) = 0;$$

that is an equation of degree n in $Q(\sqrt{2})$. Hence it satisfies an equation of degree $2n$ in Q, which is a contradiction. This proves the lemma.

Theorem 5
The field of expressions formed from rational powers of $Q(x, x^x)$ has a decidable canonical form.

Proof
By the lemma x^x is independent of x, and so can be written as a new transcendental quantity y.

Theorems 3 and 4 show that there is a decidable canonical form for rational powers of $Q(x, y)$, and so the theorem is proved.

Corollary
The field of expressions formed from linear polynomial powers of polynomials in one variable over the rationals has a decidable canonical form.

Conclusion
This paper has presented a philosophical attitude to algebraic simplification, which it is believed may be useful in the writing of efficient and effective algebraic manipulators. From this attitude a new direction for simplifiers is proposed. In addition some new canonical forms have been given, that advance by a little our knowledge of this subject. The classes of expression considered arise from an application in general relativity, but it is of interest to note that these canonical forms were not used in the calculations. There remain the problems of producing effective algorithms or heuristics to implement these canonical forms, and of extending these canonicalising theorems. Also it is useful to continue to investigate the rationale behind simplifiers, and this paper is offered as an addition to the work of Moses (1971).

Acknowledgements
I would like to acknowledge the help I have received from M. J. T. Guy and M. N. Huxley, in producing the theorems, and the many helpful suggestions of D. Barton and J. Larmouth.

References

Brown, W. S. (1969). Rational Exponential Expressions and a Conjecture Concerning π and e, *Amer. Math. Mon.*, Vol. 76, pp. 28-34

Caviness, B. F. (1970). On Canonical Forms and Simplification, *JACM*, Vol. 17, pp. 385-396.

Caviness, B. F., Pollack, P. L., and Rubald, C. M. (1971). An Existence Lemma for Canonical forms in Symbolic Mathematics, *Inform. Process. Lett.*, Vol. 1, pp. 45-46.

Davis, M., Putnam, H., and Robinson, J. (1961). The Decision Problem for Exponential Diophantine Equations, *Ann. Math.*, Vol. 74, pp. 425-436.

Engeli, M. (1968). *User's Manual for the Formula Manipulation Language SYMBAL*. University of Texas at Austin.

Fateman, R. J. (1971). The User Level Semantic Matching Capabilities in MACSYMA, *Proc. SYMSAM/2*, Los Angeles.

Fenichel, R. R. (1966). *An On-line System for Algebraic Manipulation*. Harvard University Ph.D. thesis, MAC-TR-35.

Fitch, J. P. (1971). *An Algebraic Manipulator*. University of Cambridge, Ph.D. thesis.

Gelfond, A. (1949). On the Algebraic Independence of Transcendental Numbers of Certain Classes, *Uspehi Matem. Nauk*, Vol. 4, pp. 14-18 (and AMS Translation No. 66 (1952) p. 46).

Griesmer, J., and Jenks, R. (1971). Scratchpad/1: An Interactive Facility for Symbolic Mathematics, *Proc. SYMSAM/2*, Los Angeles.

Harrison, B. K. (1959). Exact Three-variable Solutions of the Field Equations of General Relativity, *Phy. Rev.* Vol. 116, pp. 1285-1296.

Hart, T. (1961). *Simplify*, MIT AI Proj. Memo 27.

Hearn, A. C. (1968). *The Problem of Substitution*, Stanford University AI Proj. Memo 70.

Hearn, A. C. (1971). Reduce 2, *Proc. SYMSAM/2*, Los Angeles.

Johnson, S. C. (1971). On the Problem of Recognizing Zero, *JACM*, Vol. 18, pp. 559-565.

Jordan, D. E., Kain, R., and Clapp, L. C. (1968) Symbolic Factoring of Polynomials in Several Variables, *CACM*, Vol. 9, pp. 638-643.

Korsvold, K. (1965). *An On-line Algebraic Simplify Program*, Stanford University AI Proj. Memo 37.

Martin, W. A. (1971). Determining the Equivalence of Algebraic Expressions by Hash Coding, *JACM*, Vol. 18, pp. 549-558.

Matijasevic, Ju. V. (1970). Enumerable Sets are Diophantine, *Sov. Math. Dok.*, Vol. 11, pp. 453-458.

Moore, R. E. (1966). *Interval Analysis*, Prentice Hall.

Moses, J. (1971). Algebraic Simplification: A Guide to the Perplexed, *Proc. SYMSAM/2*, Los Angeles.

Niven, I. (1956). Irrational Numbers, *Carus Math. Monographs* 11.

Postnikov, M. M. (1962). *Foundations of Galois Theory*, Int. Series of Monographs on Pure and Applied Math., Pergamon.

Richardson, D. (1966). *Solvable and Unsolvable Problems Involving Elementary Functions of a Real Variable*, University of Bristol Ph.D. thesis.

Sammet, J. E. (1967). Formula Manipulation by Computer, *Advances in Computers*, Vol. 8, pp. 47-102.

Schneider, Th. (1957). Einführung in die Transzendenten Zahlen, *Mathematischen Wissenschaften*, 81.

Tobey, R. G., Bobrow, R. J., and Zilles, S. N. (1965). Automatic Simplification in FORMAC, *Proc. FJCC*, pp. 37-53.

van der Waerden, B. L. (1949). *Modern Algebra*, Vol. 1, Ungar.

Wooldridge, D. (1963). *An Algebraic Simplify Program in LISP*, Stanford University AI Proj. Memo 11.