A treatise on elementary equation solving

Citation for published version:
Bundy, A 1979 'A treatise on elementary equation solving' Working Paper , no. 51, pp. 1-44.

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Other version

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Subject: A Tretise on Elementary Equation Solving

Author: Alan Bundy

Sections
1. Abstract
2. Introduction
3. Legal moves
4. Isolation
5. Standard methods
6. Collection and Attraction
7. Polynomial Equations
8. Rational Equations
9. Simple trigonometric equations
10. Change of unknown
11. Factorization
12. Simultaneous Equations
13. Pattern matching
14. Conclusion
15. Appendices
16. References

1.0 ABSTRACT

This paper was originally started in April 1975 but progress was interrupted. It describes our ideas on equation solving prior to the publication of Bundy 1975 and the writing of the FRES equation solving process. Following a revival of our interest in equation solving we have decided to finish the paper and produce it as a working paper. It deals with some areas not covered in Bundy 1975 and is complementary to that paper. Our ideas on some of these areas have never been implemented, especially those on polynomial, rational and trigonometric equations.
2.0 INTRODUCTION

This memo is an intermediate report of a study of algebraic equation solving, and is intended as a vehicle for discussion. The study began with an analysis of human mathematical behaviour in this problem domain, using introspection and both written and verbal protocols. It continued with the design of normative models of this behaviour using the techniques of mathematical logic and computer programming. Our eventual aim is to incorporate these models in a computer program which can solve equations with the abilities of an expert. The equations whose solutions we are going to study are constructed from real numbers, arbitrary constants, variables and all the normal functions of school algebra i.e. including trigonometric, exponential, logarithmic and inverse trigonometric functions. We call these the R Elementary Functions (see appendix A for a formal definition). The equations may be solved under certain assumptions (or preconditions) about the arbitrary constants. Only real number solutions to these equations are sought. In the first instance we confine our attention to equations in one variable. We later extend this to systems of simultaneous equations in several variables, and to equations constructed from a wider range of functions. Appendix C contains a proof that the problem of solving a single R elementary equation in one unknown is undecidable.

The normal, normative, computational model of this task would be of the derivation of an appropriate theorem in some real number, axiom system. However, a typical search space for the solution of even a simple equation would be explosively large. In fact the task is clearly beyond the abilities of a conventional, unguided, automatic theorem prover. In contrast an experienced mathematician can move almost directly to a solution of even a moderately difficult equation. The motivation of this work is to understand how he does this, and to try to draw conclusions about the general problem of searching a large search space.

In his written and verbal protocols, the experienced mathematician uses various high-level descriptions of the equations he is solving, and various high-level strategies to describe his progress. For instance:

"I have two equations in two unknowns."
"there are two occurences of x."
"I will eliminate x between equations 1 and 2"
"I will collect coefficients of x, factorize the polynomial and split into factors"

Clearly these descriptions and strategies are important clues as to how the search for a solution is being guided.

In this memo we begin to show how to build a normative model of the experienced mathematician's equation solving abilities in which a system of high-level strategies is used to guide the search for a solution. Each strategy will be a program which can make moves in some proof checking system or direct some other strategy to do so. It will decide which move or strategy to call, and whether these calls have achieved its objective, by calling for high-level descriptions of the equations in question. The proof checker will be a conventional, theorem-proving program, which will contain the axioms of real number theory (although most of these axioms will be what are normally thought of as theorems) and will be responsible for keeping records and making
sure that all moves are legal etc.

This separation of inference system and search strategy (see Kowalski 1974) has a number of advantages (see Hayes 1974). For instance:

1. Provided the proof checker is sound, the correctness of any solution is guaranteed. This is sometimes in doubt in an ad-hoc heuristic program.

2. The hour-glass effect between strategies is avoided. i.e. all the facts can be accessible to all the strategies, so a new fact need only be learnt once.

3. The learning of both new facts and new strategies is made easier.

3.0 LEGAL MOVES

In this section we discuss the formalism we will need to represent the equation solving process in a computer program. We start by considering an example and then use this as a vehicle to discuss the problems which arise.

3.1 Example

1. Solve $\sqrt{5x-25}-\sqrt{x-1} = 2$

2. $\sqrt{5x-25} = 2+\sqrt{x-1}$

3. $5x-25 = 4+4\sqrt{x-1}+x-1$

4. $4\sqrt{x-1} = 4x-28$

5. $\sqrt{x-1} = x-7$

6. $x-1 = x^2-14x+49$

7. $x^2-15x+50 = 0$

8. $(x-5)(x-10) = 0$

9. $x=5$ or $x=10$

10. Checking
    
    If $x=5$ then $lhs = \sqrt{25-25}-\sqrt{5-1}$
        $= 0-2 =/= rhs$
    (assuming $\sqrt{x}$ means the positive square root of x)
    If $x=10$ then $lhs = \sqrt{50-25}-\sqrt{10-1}$
        $= 5-3 = rhs$
11. The general solution is $x=10$.

3.2 First Formalism

Consider the above example (taken from Tranter 1972) of an equation being solved. On first sight the original problem appears to be, to find real numbers as values for $x$ such that $\sqrt{5x-25}-\sqrt{x-1}=2$. We might represent this problem to a traditional theorem prover, as the problem of deciding whether the formula

$$\text{some } x \ [ \ \sqrt{5x-25}-\sqrt{x-1} = 2]$$

is a theorem of the 1st order theory of real numbers. We would, of course, be interested in more than just a proof of this formula, we would also want to know what substitutions were applied to $x$ during the course of the proof.

In general we might represent the problem of solving the equation $e(x, a)$ for $x$, where $a$ is a vector of arbitrary constants obeying the precondition $p(x, a)$, as the problem of deciding whether the formula

$$\text{all } a \ \text{some } x \ [p(x, a) \rightarrow e(x, a)]$$

is a theorem of the first order theory of real numbers.

3.3 General Solutions

However this first sight is misleading. We want not just a solution to the equation, but the most general solution. To discover what form the most general solution of an equation takes let us look at some examples.

1. the most general solution of

$$20x^2+13x-21=0$$

is $x=\frac{3}{4} \lor x=-\frac{7}{5}$

So we may be able to find several real numbers which are solutions of an equation.

2. the most general solution of

$$a 
eq 0 \land ax^2+bx+c = 0$$

is $x = \frac{-b+\sqrt{b^2-4ac}}{2a} \lor x = \frac{-b-\sqrt{b^2-4ac}}{2a}$

So each solution may be given not as a particular real number, but as a symbolic expression, which evaluates to a real number if values are given for its arbitrary constants.

3. the most general solution of

$$\sin(x)=0$$

is $x=n\pi$ where $n$ is an integer, or more formally

$$\text{some } n \ [\text{in}(n, \text{integers}) \land x=n\pi]$$

So there may be a countably infinite number of solutions. Alternatively, we may think of the solution as containing a typed parameter, $n$.

4. there are no solutions of

$$x+1 = x$$

We will denote its general solution by the empty disjunction sign (or contradiction sign), false.
3.4 Second Formalism

Let us adopt the notation

\[ \text{some } n, s, f \ (\text{in}(n,s) \land x = f(n,a)) \]

for the general solution of an equation \( e(x,a) \) for \( x \), under precondition \( p(a) \). This notation can be thought of as representing a disjunction of equations of the form, \( x = f(n,a) \), where \( n \) varies over the set \( s \), and \( f(n,a) \) is a term which does not contain \( x \). So the problem of solving the equation \( e(x,a) \) for \( x \) can be represented as the problem of finding the \( f(n,a) \) such that

\[ (1) \ \text{all } a, x \ (p(a) \rightarrow \text{some } n, s, f \ (\text{in}(n,s) \land (x = f(n,a) \leftrightarrow e(x,a)))) \]

is a theorem of real number theory, where \( p(a) \) is a precondition, similar to that in example 2.

\[ (1) \text{ can be expressed as a conjunction of two formulae} \]

\[ (1.1) \ \text{all } a, x \ [(p(a) \land e(x,a)) \rightarrow \text{some } n, s, f \ (\text{in}(n,s) \land x = f(n,a))] \]

and

\[ (1.2) \ \text{all } a, x \ \text{some } n, s, f \ [(p(a) \land x = f(n,a)) \rightarrow (\text{in}(n,s) \land e(x,a))] \]

Note that formula (1.2) is closely related (see appendix B) to the formula

\[ \text{all } a \ \text{some } x \ [(p(a) \rightarrow e(x,a)] \text{ which was our first attempt at representing the problem of solving } e(x,a) \text{ for } x. \]

3.5 The Solution

By considering protocols of human problem solving, like the example at the beginning of this section it is possible to see that humans actually try to prove the formula (1.1) first. They do this by assuming

\[ p(a) \text{ and } e(x,a) \]

where \( x \) and the \( a \) are arbitrary constants, and deducing consequences of the form

\[ \text{in}(N,S) \land x = F(N,a) \]

where \( N, S \) and \( F \) are free variables. (This corresponds to using the dual skolemization of (1.1) as a goal (see appendix B).) The consequences usually take the form of a sequence of equations (or disjunction of equations) which starts with \( e(x,a) \) and ends with \( N, S \) and \( F \) being instantiated to particular values, e.g. \( S \) to the integers, \( F \) to some arctrig formula etc.

Note that after skolemization neither the hypothesis nor the conclusion of this sequence contain variables. This is because universally quantified variables in the formula we are trying to prove, skolemize into skolem functions (see Bledsoe 1972). These skolem functions are all constants because there are no variables for them to have as arguments. In fact no formulae in the sequence will contain variables because new variables are never introduced. This is extremely convenient when it comes to designing a pattern matching

\[ [1] \text{ Gauss (1966) (talking about congruences) expressed this point in the following way.} \]

"This conclusion needs demonstration but we have suppressed it here. Nothing more follows from our analysis than that the proposed congruences cannot be solved by other values of the unknown \( x, y \) etc. We have not shown that these values do satisfy them. It is even possible that there is no solution at all. A similar paralogism occurs in treating linear equations."
routine (see Plotkin 1972), and for solving the equality problem (see Bundy 1973 p.131). To distinguish $x$ from the other constants, $a$, we will call it the unknown.

Most of the steps in the sequence tend to be reversible. When they are all reversible we have proved the theorem

$$(1) \text{all } \alpha, x \{ p(\alpha) \rightarrow \text{some } n, s, f \{ \text{in}(n,s) \wedge (x=f(n,\alpha) \leftrightarrow e(x,\alpha)) \} \}
$$

however when some steps are not, for instance the steps from 2 to 3 and from 5 to 6 in our example above (section 3.1), then some of the $f(n,\alpha)$ may not be solutions. So for each $n$ we must check to see if

$$\text{all } \alpha \{ p(\alpha) \rightarrow e(f(n,\alpha), \alpha) \}
$$

is a theorem. This checking is necessary in our example, and takes place in step 10.

For each $n$ for which this check fails we have shown that

$$(1') \text{all } \alpha, x \{ p(\alpha) \rightarrow \text{some } n, s, f \{ \text{in}(n,s) \wedge (x=f(n,\alpha) \leftrightarrow e(x,\alpha)) \} \}
$$

### 3.6 A Deductive System

We now turn to the problem of designing an algebraic manipulation system within which this sequence of steps can be generated. This system will be called the legal move maker. It will be responsible for ensuring the correctness of each step. But not for deciding which steps to make, this problem will be dealt with in succeeding sections. Each of the steps of the sequence will be justified by an axiom (or theorem) and a rule of real number theory. We will divide these axioms into 2 classes

1. identities i.e. Universally quantified equations (but we will usually omit the quantifiers), sometimes preceded by a precondition, e.g. 
   
   $$\sin(x) - \sin(y) = 2 \cos((x+y)/2) \sin((x-y)/2)$$
   $$\log xy = \log x + \log y$$

2. implications between an equation and a disjunction of equations. Sometimes preceded by a precondition, e.g.

   $$a=0 \rightarrow [ax^2+bx+c=0 \rightarrow (x=(-b+\sqrt{b^2-4ac})/2a \text{ or } x=(-b-\sqrt{b^2-4ac})/2a)]$$
   $$\sin(x)=y \rightarrow \text{some } n \{ \text{in}(n, \text{integers}) \wedge x = n \pi + (-1)^n \arcsin(y) \}$$

If the reverse implication (between the disjunction and the equation) is also a theorem then we will call the implication "reversible",

$$x=y \rightarrow x^2=y^2$$

is not reversible since $x^2=y^2 \rightarrow x=y$ is not a theorem, but

$$x.y=0 \rightarrow (x=0 \text{ or } y=0)$$

is, since $(x=0 \text{ or } y=0) \rightarrow x.y=0$ is a theorem.

Corresponding to each class of axioms we will have a rule of inference to show how axioms from this class are to be used. The rule for using identities is called paramodulation and the rule for using implications is called modus ponens. Their definitions follow.
1. **paramodulation**: let the current disjunction of equations be some \( n \) in \((n,s)\) & \( b(n) \) where each \( b(n) \) is an equation. Let \( c \) be a subterm of \( b(m) \), say. Let \( c' = d' \) (or \( d' = c' \)) be some identity. If \( \sigma \) is some substitution which matches \( c' \) to \( c \), i.e. \( c' \sigma = c \), then we may replace \( c \) by \( d' \sigma \) in \( b(m) \) to make \( b'(m) \), provided \( d' \sigma \) is variable free. Let \( b'(n) = b(n) \) for all \( n \neq m \), then some \( n \) in \((n,s)\) & \( b'(n) \) is the next disjunction in the sequence. Note that the substitution \( \sigma \) need only be applied to the identity, since the current disjunction of equations contains no variables. We defer discussion of the pattern matcher until later. Note that paramodulation steps are always reversible.

2. **modus ponens**: let \( b(m) \) be some equation in the current disjunction . Let \( p \rightarrow (e \rightarrow d) \) be an implication where \( p \) is the precondition, \( e \) is an equation and \( d \) is a disjunction of equations. We are going to try to match \( e \) to \( b(m) \) with a substitution \( \sigma \) (i.e. \( e \sigma = b(m) \)), and replace \( b(m) \) with \( d \sigma \) to form the next sequence. We will assume that we know, in advance, which variables in the equation, namely the vector of variables \( y \), we are going to match to subterms containing the unknowns in \( b(m) \), namely the vector of constants \( x \). For the application to succeed we are going to insist that \( p \sigma \) and \( d \sigma \) are variable free, and that \( p \sigma \) is provable. We defer discussion of how \( p \sigma \) is to be proved.

The equations \( d \sigma \) are said to be immediate descendants of \( b(m) \). An immediate descendant is a descendant and so is an immediate descendant of a descendant. If \( p \rightarrow (d \rightarrow e) \) then the application of the axiom will be called a reversible step, otherwise the equations in \( d \sigma \) and all their descendants are marked "suspect". A series of reversible steps will be called a reversible branch. [1]

The only part of the equation solving procedure which is not within the main sequence is the final checking of all solutions. i.e. Those descended from a non-reversible step. This involves substituting suspect solutions into the original equations and checking that the resulting identities are theorems. This checking could be done within a system similar to the one just described. This system will also be used for precondition testing and for the dynamic production of new identity axioms which is needed in the factorization strategy.

---

[1] These definitions essentially describe a 'rewrite rule' system for \( \rightarrow \) and \( \rightarrow \), with the additional complication that some of the \( \rightarrow \) steps are not reversible, i.e. \( \leftrightarrow \) does not hold.
3. 7 Change Of Unknown

The deductive system outlined here is extremely limited and will almost certainly have to be extended in the light of new types or uses of axioms as these are discovered. Meanwhile we have deliberately limited the system in the hope of attracting constructive criticism.

One possible extension, for instance, would be to allow an explicit "change of unknown", e.g.

\[ 3(\cos(x))^2 + \sin(x) \cdot \cos(x) - 4(\sin(x))^2 = 0 \]

becomes

\[ 3c^2 + s \cdot c - 4s^2 = 0 \]

where \( c = \cos(x) \) and \( s = \sin(x) \). c and s become the new unknowns and (temporarily) replace \( x \). This ability would be used by the change of unknown strategy to simplify the existing equation (in this case to a polynomial). It is justified by the 2nd order axiom:

all \( f, p \) all \( x \) [ \( p(x, f(x)) \iff \) some \( y \) (\( y = f(x) \) & \( p(x, y) \)) ] \( p(x, f(x)) \)

will be called the "old equation" and \( x \) the "old unknown". \( p(x, y) \) will be called the "new equation" and \( y \) the "new unknown". \( f(x) \) will be called the "replaced term".

Note that this ability admits the possibility of introducing several unknowns into the equation and we have had to allow for this in our definition of modus ponens. We will also have to extend our main sequence from being a simple disjunction of equations to a disjunction of conjunctions of equations, where each conjunction is a main equation followed by the definition of new unknowns in terms of old unknowns.

Some subjects would regard the equation:

\[ 3(\cos x)^2 + \sin x \cdot \cos x - 4(\sin x)^2 = 0 \]

simply as a polynomial in \( \cos x \) and \( \sin x \), and would not introduce new unknowns. We will call this an "implicit" change of unknown. It could be realized by making an explicit change of unknown but using the strings "\( \cos x \)" and "\( \sin x \)" for all output purposes.

Without some mechanism like explicit change of unknown it is difficult to understand the preparatory steps which subjects make to express terms containing \( x \) in terms of the prospective replaced term, \( f(x) \). For instance, consider the following example from Briggs and Bryan 1962 p 103:

1. \( 6x^4 + 35x^3 + 62x^2 + 35x + 6 = 0 \)
2. \( 6(x^2 + 1/x^2) + 35(x + 1/x) + 62 = 0 \)
3. \( 6(x + 1/x)^2 - 12 + 35(x + 1/x) + 62 = 0 \)
4. \( 6y^2 + 35y + 50 = 0 \) where \( y = x + 1/x \)
3.8 Further Extensions

Further extensions of our system might include anti-equations and evaluation. An anti-equation \( a=\neq b \), where \( a \) and \( b \) are terms, behaves in the same way as the implication \( a=b \implies \text{false} \), where \( \text{false} \) is the empty disjunction. Examples are:

\[
x^2=\neq y^2 \text{ and } 1=\neq 0
\]

The effect of a successful application of one of these to an equation in the current disjunction is to delete it from the disjunction.

There are an infinite number of anti-equations of the form \( i=\neq j \), where \( i \) and \( j \) are different numbers, so that these and similar axioms might best be handled by an evaluation mechanism.

3.9 Side Calculations

We will frequently need to carry out several deductions at the same time. For instance, while factorizing a polynomial we may leave deduction of the main sequence, temporarily, in order to calculate suitable factors (see polynomial equations section). Again in the change of unknowns strategy we may leave the main deduction in order to try to express some term containing \( x \) in terms of the prospective reduced term. These minor deductions will be called "side calculations". They will probably be handled by a theorem proving system similar to the one already outlined, but deductions made under different hypotheses would be scaled off from one another by something like Conniver contexts (see McDermott and Sussman 1972).

3.10 Tricky Legalities

There are some tricky problems about just what is, and what is not, an axiom of real number theory, e.g.

\[
0<y \land x^2=y \implies (x=\sqrt{y}) \lor x=-\sqrt{y})
\]

but

\[
x^2=y \implies x=\sqrt{y}, y \quad (\text{note that } x \text{ is real})
\]

What is the correct generalization of the above formulae for the equation \( x^2=y \)?

My conjecture is:

\[
[0=y \land \text{rational}(z) \land \text{even}(z) \land x^2=y] \implies [x=\sqrt{z}, y] \lor x=-\sqrt{z}, y]
\]

\[
[0>y \land \text{rational}(z) \land \text{even}(z) \land x^2=y] \implies \text{false}
\]

\[
[\text{rational}(z) \land \text{odd}(z) \land x^2=y] \implies x=\sqrt{z}, y
\]

\[
[\text{irrational}(z) \land x^2=y] \implies x=\sqrt{z}, y
\]

where \( \text{even}(z) \) iff \( \text{not odd}(z) \) iff \( p \) is even, where \( p/q \) is the fraction in lowest terms equal to \( z \).
Another problem is that it is not always clear what the definition of some of the functions should be. For instance, some functions e.g. sqrt and arcsin, are not naturally single-valued. However, we can easily make a multi-valued function, single-valued by distinguishing one of its values. For instance, the one which lies in a certain range, e.g. by distinguishing the positive square root and the value of arcsin which lies in the range -pi/2 to pi/2.

Again, some functions are not naturally defined everywhere. E.g. Division and the inverse trig functions. (consider 2/0 and arccos(2)). We may overcome this problem in at least two ways. We could assign all undefined terms some arbitrary value, say 0, or we could say that undefined terms are not well formed formula. The former solution does not seem very natural and could lead to unpredictable and unpleasant changes to the axioms, e.g. arcsin(x)=0 -> x=0

Might become
arcsin(x)=0 -> (x=0 v x>1 v x<-1)
The latter solution seems more natural, but means that the question of whether a formula is well-formed becomes recursively unsolvable.

3.11 Strategies

We now turn to the problem of running the legal move maker in order to find the most general solution to an equation. My contention is that human mathematicians use a hierarchy of strategies to guide their search for a solution, and that evidence for this can be found in the language they use, in both verbal and written protocols, to describe this process. These strategies can be thought of as macro-moves or as programs which can call lower level strategies and legal moves, but which can contain loops and conditionals. Typically these conditionals use a high level description of the current equation, as a test, to decide which branch they should take.

In succeeding sections we will investigate some of these strategies. The plan of the sections will be the same as the preceding one. First we will look at some examples of the strategy and analyse them. Then we will explore mechanisms for explaining them with an eye on our goal of implementing these mechanisms on a computer.

4.0 ISOLATION

If an equation contains only one occurrence of the unknown there is a strategy which always succeeds for the class of functions we are considering. It consists of stripping away the functions which surround the single occurrence, and leaving it isolated on one side of the equation (hence the name of the strategy: isolation), e.g.

1. Solve log(e, 3x^2+2) + pi = e for x
2. log(e, 3^x+2+2) = e-pi
3. $3x^2 + 2 = e^{(e-pi)}$

4. $3x^2 = e^{(e-pi)} - 2$

5. $x^2 = (e^{(e-pi)}) - 2)/3$

6. $x = \sqrt{((e^{(e-pi)} - 2)/3)}$ or $x = -\sqrt{((e^{(e-pi)} - 2)/3)}$

Clearly what is happening is an application of the following procedure:

1. Let $f(al, \ldots, an) = b$ be the current equation, where $x$ is contained in $ai$, but not in $aj$ for $i \neq j$, or in $b$.

2. Look for an implication axiom of the form:
   
   $f(xl, \ldots, xn) = y \rightarrow \text{some } m \text{ some } s [\text{in}(m, s) \& xi = g(m, xl, \ldots, xi-l, xi+1, \ldots, xn, y)]$

3. If there is no such axiom then fail, but if there is, apply it to get:
   
   $\text{some } m \text{ some } s [\text{in}(m, s) \& ai = g(m, al, \ldots, ai-1, ai+1, \ldots, an, b)]$

4. Then apply the procedure to each of the equations
   
   $ai = g(m, al, \ldots, ai-1, ai+1, \ldots, an, b)$

   until $ai$ is the single occurrence of $x$ and succeed.

   (Later we will consider other stopping conditions).

This strategy will always succeed for us because there is an axiom of the required form corresponding to each argument place of each function in the class of elementary functions. In fact we may suspect that the, otherwise rather unpleasant functions, arcsin, sqrt etc were introduced into algebra, by mathematicians, just to make this true.

Note that this procedure uses the following kinds of high level description:

"the single occurrence of $x$ occurs on the left (right) hand side of the equation". "the dominating function on the left (right) hand side is $f$". "the arguments of $f$ are $al, \ldots, an$". "$ai$ contains $x$, but $aj$ for $i \neq j$, does not".

We will have to implement these and many similar descriptions.

The axioms used by this strategy are very clearly defined, and it would obviously be a quite simple matter to design a procedure to vet incoming facts and earmark those which could be used by isolation. If we could do this for each strategy, we would have a criterion for identifying "useful theorems" and the means to employ them fruitfully. Looking even further ahead, it might be possible, on being given the definition of a new function, to generate a description of the kind of theorems we would like to prove about it and then set about proving them.
5.0 STANDARD METHODS

Even if there is more than one occurrence of the unknown in an equation, we may be able to use a method very similar to isolation to solve the equation, if we recognise that the equation is in some standard form (e.g. A polynomial of degree less than or equal to 4) for which we know the general solution, e.g.

1. Solve \( x^2 - 3x + 1 = 0 \) for \( x \)

2. \( x = (3 + \sqrt{9 - 4.1.1}) / 2 \) or \( x = (3 - \sqrt{9 - 4.1.1}) / 2 \)

3. \( x = (3 + \sqrt{5}) / 2 \) or \( x = (3 - \sqrt{5}) / 2 \)

or

1. Solve \( 3 \cos(x) + 4 \sin(x) = 5 \)

2. some \( n \) \([\text{in}(n, \text{integers}) \& x = n \pi + (-1)^n \arcsin(5 / \sqrt{3^2 + 4^2}) - \arcsin(3 / \sqrt{3^2 + 4^2})]\)

3. some \( n \) \([\text{in}(n, \text{integers}) \& x = n \pi + (-1)^n \pi / 2 - \arcsin(3 / 5)]\)

In either case we recognise that the equation is in the form:

\[ b(x, a) = c \]

where \( x \) may occur several times in term \( b \), but does not occur in \( c \), and that we have an implication axiom:

\[ b(y, z) = d \rightarrow \text{some } n, s \ [\text{in}(n, s) \& y = g(n, z, d)] \]

which can be applied to it to yield:

\[ \text{some } n, s \ [\text{in}(n, s) \& x = g(n, a, c)] \]

These two strategies obviously have a lot in common, and we might think of combining them to produce a powerful generalisation which uses axioms of the form:

\[ b(x_1, \ldots, x_n) = d \rightarrow \text{some } m, s \ [\text{in}(m, s) \& x_i = g(m, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n, d)] \]

(where \( b \) can be an arbitrary term and \( x_i \) can occur several times in \( b \), but not in \( d \)), on equations of the form:

\[ b(a_1, \ldots, a_n) = c \]

(\( x \) occurs only in \( a_1 \)).

6.0 COLLECTION AND ATTRACTION

The preceding strategies take some equation whose solution is desired and deliver the general solution. In fact they will be the only strategies designed to deliver general solutions. However, their domain is extremely limited and we will need to extend this if we are to solve interesting equations. So the succeeding strategies will be designed to manipulate equations into this domain, either by reducing the number of occurrences of the unknown to one, so that isolation can be applied, or by reducing the equation to some standard form, so that a standard method can be applied.

We now consider two very simple strategies designed to achieve the former goal of reducing the number of occurrences of the unknown. It is sometimes possible to use an identity axiom to replace a subterm of an equation containing several occurrences of the unknown, by a term containing fewer
1. Solve $\log(e, x+1) + 2 + \log(e, x-1) = 0$

2. $2 + \log(e, x+1) + \log(e, x-1) = 0$

3. $2 + \log(e, (x+1)(x-1)) = 0$

4. $2 + \log(e, x^2-1) = 0$

The two occurrences of $x$ are moved closer together in lines 2 and 3 and collected in line 4. We will call this strategy "attraction", because it attracts occurrences of the unknown to each other. For clarity of explanation we have separated attraction from collection, but they are obviously very similar, and could be implemented in a similar way, or combined.

The only difference between them would be in the identities they used. Attraction could use an identity:

$p \rightarrow (a=b)$

if there are variables $y$ in $a$ which have the same number of occurrences in $a$ as in $b$, but no occurrences in $p$, and the distances between the $y$ is strictly greater in $a$ than it is in $b$ and $a$ is the least common dominating term for its occurrences of the $y$. The $y$ will each be bound to subterms containing the unknown, and $a$ will be bound to the least common dominating term in the equation.

7.0 POLYNOMIAL EQUATIONS

The strategies we have considered so far are completely general, and like most general purpose strategies rather weak. In fact we can only develop strong strategies for particular classes of equations. In the literature of equation solving one particular class stands out for the strength of the strategies developed for it. This is the class of polynomial equations. Special attention has been lavished on the polynomial equations for several reasons, e.g.:

1. polynomials occur frequently in applications of equation solving.

2. many non-polynomial equations can be reduced to polynomials by change of unknown and other devices.

We have singled them out for special attention for the latter reason above, and because many of the strategies developed for polynomial equations can be generalized to wider classes of equations. definitions

We now give a formal definition of the class of polynomial equations.

\[
\text{<poly>} ::= \text{<const>} / \text{<unknown>} / \langle\text{poly}\rangle + \langle\text{poly}\rangle / \langle\text{poly}\rangle - \langle\text{poly}\rangle / \langle\text{poly}\rangle / \langle\text{poly}\rangle / \langle\text{poly}\rangle \times \langle\text{positive integer}\rangle \\
\text{<poly eqn>} ::= \langle\text{poly}\rangle = \langle\text{poly}\rangle
\]
1. Solve $\cos(2x) - 2\sin(x)\cos(x) = \sin(\pi/4)$

2. $\cos(2x) - \sin(2x) = 1/\sqrt{2}$

3. $\sqrt{2} \cdot \sin(x + \arcsin(1/\sqrt{2})) = 1/\sqrt{2}$

4. $\sin(x + \pi/4) = 1/2$

The number of occurrences of $x$ was reduced in both line 2 and line 3. Isolation could now be applied. We will call this strategy "collection", since its goal is to collect together occurrences of the unknown. How might it be mechanised. First we will need to decide which set of occurrences we are going to collect. We can only think that we must try all possibilities, starting with those closest together. (the distance between $n$ occurrences of subterms is the number of arcs in the smallest connected subgraph containing all $n$ occurrences, when the equation is regarded as a tree). The closest occurrences in line 1 of our example are the 2nd and 3rd. We now focus our attention on the smallest subterm containing all the occurrences in question (which we will call the least common dominating term). In our example this is the subterm $\sin(x)\cos(x)$. We now look for identities with a left (or right) hand side which matches this subterm, and whose application would result in a reduction in the number of unknowns. In our case the identity:

$$\sin(y)\cos(y) = (\sin(2y))/2$$

fits the bill. The identities used by collection are all of the form:

$$p \rightarrow (a=b)$$

where some collection of variables which occur in a (vector $y$ say) occur strictly less times in $b$, and not at all in $p$, and $a$ is the least, common dominating term for its occurrences of the $y$. The $y$ will each be bound to the unknown or some subterm containing it and $a$ will be matched to the least common dominating term in the equation .

This description of the class of axioms useful to collection, like the description of the class of axioms useful to isolation, is very exacting. A lot of axioms would just fail to fit them, e.g.

$$2\sin(y)\cos(y) = \sin(2y)$$

(because of factor 2)

A procedure which was able to assimilate new axioms into a strategy, would have to be able to employ a higher level description to recognise an axiom as useful to that strategy, and to use some deduction to get this axiom into the preferred form.

Because the identities, which may be useful to collection, can be singled out in advance, and even indexed by whether they contain trigonometric, logarithmic functions etc, the process of recovering an identity which can affect a successful collection, can be made quite efficient.

Following a successful collection we may call collection again, on the resulting equation, until it can be applied no longer.

We still have not enlarged our domain of solvable equations very much, because collection can be only rarely applied. It is often the case, however, that although we cannot collect several occurrences of an unknown, we can reduce the distance between them and make their eventual collection more likely, e.g.
7.1 Main Strategy

The main strategy in dealing with polynomial equations is to put them into normal form, factorize them, split into factors and solve each of the factors recursively. There are also a number of special strategies for subclasses of the polynomial equation e.g. For equations of degree less than 5 and for reciprocal equations. We deal with these after consideration of the main strategy. We now turn to a consideration of each part of the main strategy in detail.

7.1.1 Polynomial Normal Form -

1. \((3x^2 + 2)(x+1) + ax(2x+1) = 13\)
2. \(3x^3 + 2x + 3x^2 + 2 + 2ax^2 ax = 13\)
3. \(3x^3 + (2a+3)x^2 + (2+a)x - 11 = 0\)

The above is an example of a polynomial being put into normal form. The operation can be divided into two main parts: "multiplying out", illustrated by the transition from line 1 to line 2 in our example, and "collecting coefficients of terms", illustrated by the transition from line 2 to line 3. Many intermediate steps have been omitted from the above. For instance, after applying the distributive law to \(ax(2x+1)\), \(ax.2x\) is rewritten as \(2ax^2\) and \(ax.1\) as \(ax\). Human subjects usually omit such steps from their written protocols and do them in their heads.

"multiplying out" consists of the following steps:

1. the distributive law is applied repeatedly until both sides of the equation are terms in the class "sum" defined as follows:

\[
\begin{align*}
\text{<symbol>} &::= \text{any function symbol except } + \text{ or } - \\
\text{<constituent>} &::= \text{<unknown>} / \text{<number>} / \text{<arbitrary constant>} / \text{<symbol>(<sum>, ..., <sum>)} / -\text{<constituent>} \\
\text{<product>} &::= \text{<constituent>}, ..., \text{<constituent>} \\
\text{<sum>} &::= \text{<product>+, ..., +<product>}
\end{align*}
\]

[1] These grammars are represented in standard BNF form. <...> represents a class of expressions; / represents disjunction; * means that this class may be repeated any positive number of times; ? means that this class is optional; *? means either * or ?.
2. then each product is simplified and put in the normal form defined by the class \(<\text{new product}>\).

\(<\text{power of unknown}> =: <\text{unknown}> / <\text{unknown}>^{\text{positive integer}>\)

\(<\text{new product l}> =: <\text{number}?>.<\text{arbitrary constants}?>.

<<\text{symbol}>(<\text{sum}>, ..., <\text{sum}>) ?>.<\text{power of unknown}?>

\(<\text{new product}> =: <\text{new product l}> / -(<\text{new product l}>)

This operation uses the commutative and associative laws of multiplication, evaluation rules for numbers, the identities \((-x)(-y)=x.y\) and \(-x=x\) for minus and the identities \(x.x = x^2\) and \((x^n).x^n = x^{n+m}\).

"collecting coefficients of terms" consists of the steps:

1. implicitly redefining each occurrence of \(x^n\) as \(x^n\) (a new unknown), redefining \(x\) as \(x^1\) and multiplying each \(<\text{new product}>\) free of \(x\) by \(x^0\).
2. calling "collection" on each unknown \(x_i\) using the distributive law.
3. simplifying the coefficients of the \(x_i\)'s.
4. substituting back the values of the \(x_i\)'s.

7.1.2 Factorization -

1. \(20x^2+13x-21 = 0\)
2. \((5x+7)(4x-3) = 0\)
3. \(5x+7 = 0\) or \(4x-3 = 0\)

Above is an example of a simple polynomial being factored. How did we discover the factors of this polynomial? Obviously we used some kind of procedure to investigate the original polynomial and suggest suitable factors. Such a procedure is usually called a factorization algorithm. There is a factorization algorithm, due to Kronecker, which can factorize any multi-variable, integer polynomial into a product of polynomials irreducible over the integers. It is described, for instance, in Manove 1967 p 90. A different algorithm was used by the subjects whose protocols we studied (see also O'Shea 1973 p 4). Their algorithm was quite efficient at producing linear factors of single variable polynomials. It revolves around the observation

1. that the sum of the degrees of the factors is equal to the degree of the original polynomial.
Equation Solving

1. poly(p, s1) & poly(q, s2) & poly(r, s3) & p=q.r
   -> degre(p) = degree(q) + degree(r)

2. poly(p, s) -> in(degree(p), integers).

3. poly(p, s) & in(n, integers) & 0=<n=<degree(p)
   -> in(coeff(p, n), s).

4. poly(p, s1) & poly(q, s2) & poly(r, s3) & p=q.r & 0=<n=<degree(p)
   in(n, integers) -> coeff(p, n)
   = sigma(0=<i=<n, coeff(q, i).coeff(r, n-i)).

5. poly(p, s) & integer(n) & degree(p) < n -> coeff(p, n)=0

6. poly(p, s) -> p=sigma(0=<i=<degree(p), coeff(p, i).unknown(p^i)

7. poly(p, s) -> [reducible(p, t) <-> some q, r (p=q.r & poly(q, t)
   & poly(r, t) & degree(q)>0 & degree(r)>0].

Plus numerous axioms about arithmetic whose use we will acknowledge with the words 'by arith', and subroutines for evaluating functions and predicates over some of the arguments whose use we will acknowledge with the words 'by eval'.

7.1.2.2 The Interpreter -

Imitating the intuitive algorithm with these axioms seems to involve the theorem prover making the following steps.

1. Given a polynomial p, assert that it is reducible, which involves introducing the 2 variables q and r (with axiom 7.).

2. Deduce consequences of this assertion. In particular, find properties of q and r which restrict their values.

3. Find values for q and r within these restrictions (may involve case analysis) or deduce a contradiction showing that p was irreducible.

7.1.2.3 Example -

This algorithm can be used to factorise

20.x^2 + 13.x - 21 = 0

1. Assume reducible(20.x^2 +13.x -21 = 0, integers)

2. by 7. and eval

20.x^2 +13.x - 21 = q.r & poly(q, integers) & poly(r, integers) &
   degree(q)>0 & degree(r)>0.
2. that the products of all the constants in the factors is equal to $a_0$ and the product of all the coefficients of the leading terms is equal to $a_0$.

3. that if \((bk.x^{k+...+b0}).c1.x^{l+...+c0} = an.x^n+...+a0\) where \(n=k+l\) and \(an/=0/a0\), then \(\sigma(0=i<n, bi.cj-1) = aj\) for \(1<j<n\).

So that if we are trying to factorize the quadratic $20x^2+13x=21$ over the integers, for instance, we know that if it does factorize then it will have linear factors $ax+b$ and $cx+d$ where $a$, $b$, $c$, $d$ are integers, $a.c=20$, $b.d=-21$ and $ad+bc=13$. The possible values of $a$, $b$, $c$, $d$ are strictly limited (e.g. $b=1, 3, 7, 21, -1, -3, -7, -21$) so that we may rapidly exhaust them. We will call this algorithm the "intuitive algorithm". It can clearly be extended to a much wider class of factorization problems.

In a conventional algebraic manipulation program either of these algorithms would be self contained, opaque subroutines. Which would be difficult to alter and which would have limited range. However we would like the factorization algorithm to have very flexible range, e.g. We would like it, at least, to be able to factorize $x^2+(a+b)x+a.b=0$ where $a$ and $b$ are arbitrary constants.

We would also like it to be able to improve itself if new facts about the factorization of polynomials were made available to it, i.e. We want eventually to be able to extend learning to strategies. For instance we might add the fact that $x-a$ is a factor of polynomial $p(x)$ if and only if $p(a)=0$, enabling us to check potential linear factors more easily. The application of this fact is called the remainder theorem method, (see Briggs and Bryan 1962 p. 95). We therefore need some way of representing the factorization algorithm which makes it easily investigable and extendable by other programs. This facility is offered by predicate logic type programming languages (see Kowalski 1974), in which the knowledge used in the algorithm is represented as predicate logic formulae, and the interpreter of these formulae, is a theorem prover. To clarify ideas we will represent some of the knowledge needed by the intuitive algorithm to factorize polynomials in one unknown, as a predicate logic formulae and discuss the form of the theorem prover needed to direct their use.

7.1.2.1 'Intuitive' Factorization Algorithm Axioms -

poly(p, s) means p is a polynomial with coefficients in the set s.
in(n, s) means n is a member of the set s.
reducible(p, s) means p has factors which are polynomials over the set s.
degree(p) returns the degree of polynomial p.
unknown(p) returns the unknown of polynomial p.
coeff(p, n) returns the coefficient in p of unknown(p)^n.
3. by 1 and eval
   2 = degree(q) + degree(r).

4. by 2.
   in(degree(q), integers) & in(degree(r), integers).

5. by arith
   degree(q) = 1 & degree(r) = 1.

6. by 4, 5 and eval
   20 = coeff(q, 1).coeff(r, 1)
   13 = coeff(q, 1).coeff(r, 0) + coeff(r, 1).coeff(q, 0).
   -21 = coeff(q, 0).coeff(r, 0).

7. by 3 and eval
   in(coeff(q, 1), integers) & in(coeff(q, 0), integers)
   in(coeff(r, 1), integers) & in(coeff(r, 0), integers).

8. by arith
   [(coeff(q, 1) = 5 & coeff(q, 0) = 7 & coeff(r, 1) = 4 & coeff(r, 0) =
    -3) v (coeff(q, 1) = 4 & coeff(q, 0) = -3 & coeff(r, 1) = 5 & coeff(r,
    0) = 7)]

9. by 6 and eval and arith
   [q = 5.x+7 & r = 4.x-3] v [q = 4.x-3 & r = 5.x+7]

The kind of theorem prover we described in Bundy (1973) seems suitable for
this kind of deduction.

Note that even if the axioms used in this algorithm are false (e.g.
over-generalizations) this will not effect the soundness of the equation
solving process provided the output of the algorithm is only treated as advice
to the legal move maker. It may, however, cause the legal move maker to try
something which fails, or miss something which would have worked.

It should be possible to extend this algorithm to deal with polynomials of
many variables and to extend its efficiency simply by adding to or changing the
axioms. Eventually we would like to see the Knönecker algorithm coded in this
way.

7.1.3 The Knönecker Algorithm -

For the sake of completeness we record here the Knönecker polynomial
factorization algorithm.

To factorize, \( f(x) \) - a polynomial of degree \( n \)
if \( f \) is reducible
choose \( k \leq n \), find a factor of degree \( k \)
divide \( g \) into \( f \) to form \( h \)
recursively factorize \( h \)
To choose \( k \)
let \( p \) be a prime such that \( f(\text{mod } p) \) is of the same degree as \( f \).
if \( f(\text{mod } p) \) is irreducible then \( f \) is irreducible
else let \( D = \{ \text{degree}(h) : h \mid f(\text{mod } p) \} \)
choose \( k \) in \( D \)

To find a factor of degree \( k \)
choose \( k+1 \) integers \( n_0, \ldots, n_k \)
let \( B_i = \{ r : r \mid f(n_i) \} \)
loop:
if some \( B_i = \emptyset \) then \text{fail}
for all \( i \) choose \( r_i \) in \( B_i \)
let \( B_i = B_i \setminus \{ r_i \} \)
form unique \( g(x) \) of degree \( k \) such that \( g(n_i) = r_i \)
if \( g \mid f \) the succeed
else go to loop

7.2 Polynomials Of Degree Less Than 5

In theory it is always possible to factorize a polynomial with real coefficients (indeed with complex coefficients) into complex linear factors, and solve each one by isolation, rejecting non-real solutions. In practice it may not be easy to find the factors. The intuitive algorithm only works well at finding integer linear factors of integer polynomials. So alternative methods have been devised for polynomials irreducible over the integers.

The solutions to all polynomials less than 5 has been worked out in general and given by a radical formula for the unknowns in terms of the coefficients (see e.g. Burnside and Panton 1881 chap VI). The most familiar formula is the one for the quadratic equation.

\[
a x^2 + b x + c = 0
\]

which is

\[
x = \frac{-b \pm \sqrt{b^2-4.a.c.}}{2.a} \quad \text{or} \quad x = \frac{-b + \sqrt{b^2-4.a.c.}}{2.a}
\]

These formulae can be used to solve any polynomial equation of degree less than 5, even if it is reducible over the integers. Most of the subjects of our protocol analysis did not do this. They preferred to find integer factors if they could. In fact some people are so reluctant to use the quadratic formula that they prefer to use the strategy called 'completing the square', on which it is based.

We will also prefer to factorize into integer factors, where possible. A factorization algorithm is necessary for polynomials of high degree, and we will take the opportunity to develop it on simple examples.

Unfortunately, there can be no general solution, given by a radical formula, for equations of degree greater than 5. This result was first discovered by Galois (see Artin 1959 p 76). This means that we have to develop alternative methods of solving equations of degree greater than 4. Most of these methods will also be applicable to equations of degree less than 5 and so will yield an alternative method of solution to the main strategy.
7.3 Reciprocal Equations

A special method applies to the class of reciprocal equations. A polynomial equation $p$ is said to be reciprocal if, when it is written in normal form,

$$\sigma(0=<i<n, a_i x^{-i})$$

where $n=\text{degree}(p)$ then $a_i = a_{n-i}$ for all $0=<i<n$

The special method is based on the fact that for every root $b$ of the equation $1/b$ is also a root.

If the degree of $p$ is odd then, since the equation has an odd number of roots, 1 or -1 is a root. If we extract $x-1$ or $x+1$ as a linear factor we are left with a reciprocal equation of even degree.

If the degree of $p$ is even, say $2m$ then we can make a change of unknown of $y$ for $x + 1/x$ which yields a polynomial equation of degree $m$. We first make the following transformation to the equation:

rewrite

$$\sigma(0=<i<2m, a_i x^{-i})$$

as

$$\sigma(0=<i<m, a_i (x^{-i} + x^{-2m+i})) = 0$$

then as

$$\sigma(0=<i<m, a_i (x^{-m+i} + 1/x^{-m+i})) = 0$$

then starting at $i=0$ eliminate the term

$$(x^{-m+i} + 1/x^{-m+i})$$

using the binomial expansion:

$$(x+1/x)^{-m+i} = \sigma(0=<r<(m-i), c(m-i, r) x^{-2r+i-m})$$

where $c(i, j)$ is the number of ways of choosing $j$ things from $i$ things.

At each stage $(x^{-m+i} + 1/x^{-m+i})$ will be replaced by the term $(x+1/x)^{-m+i}$ and a reciprocal polynomial of degree $m-2$, which will combine with the remaining reciprocal polynomial. Finally we will get a polynomial of degree $m$ ($x+1/x$).

i.e. $\sigma(0=<i<m) b_i (x+1/x)^{-m+i} = 0$

and we can make our change of unknown.

7.3.1 Example -

1. Solve $3x^7 + 2x^6 - x^5 + 3x^4 + 3x^3 - x^2 + 2x + 3 = 0$

2. Since this reciprocal equation is of odd degree it has as a root either 1 or -1. We use the remainder theorem method to check these roots.

   when $x=1$ LHS = $3+2-1+3+3-1+2+3 =/= 0 = \text{RHS}$

   when $x=-1$ LHS = $-3+2+1+3-3-1-2+3 = 0 = \text{RHS}$

   so -1 is a root and $x+1$ a factor.

3. Dividing the LHS by $x+1$ we get

$$3x^6 - x^5 + 3x^3 - x + 3 = 0$$

4. Collecting terms and dividing by $x^3$

$$3(x^3 + 1/x^3) - (x^2 + 1/x^2) + 3 = 0$$
Equation Solving

5. Using the binomial theorem
   \[3(x^3 + 1/x)^3 - 9/x - 9.x - (x^2 + 1/x^2) + 3 = 0\]

6. Again
   \[3(x^3 + 1/x)^3 - (x^3 + 1/x)^2 + 2 - 9.(x^3 + 1/x) + 3 = 0\]

7. Changing unknown
   \[3.y^3 - y^2 - 9y + 5 = 0\]

7.3.2 Extensions –

This method may be extended in various ways. Equations of the form
\[\sum(0=<i=<n, a_i x^i) = 0, \quad a_n /= 0\]
where \(a_i = a_n - i\) for even \(i\)
and \(a_i = -a_n - i\) for odd \(i\) also have roots occurring in pairs.

The roots of each pair are reciprocals of opposite sign. Odd equations have a complex root of \(i\) or \(-i\). Even equations can be solved with a change of unknown to \(y = x - 1/x\).

Equations of odd degree of the form:
\[\sum(0=<i=<n, a_i x^i) = 0, \quad a_n /= 0\]
where \(a_i = a_n - i\) for all \(i\) are satisfied by \(x = 1\) and reduce to reciprocal equations on division by \(x - 1\).

8.0 RATIONAL EQUATIONS

Rational equations are those made from equations between Rational functions, where Rational functions are terms made from polynomials by division and exponentiation with an integer, i.e.

\(<\text{ratn func}> = :: <\text{poly}> / <\text{ratn func}>/<\text{ratn func}> / <\text{ratn func}>^<\text{integer}>\)

An example is:
\[(x^2 + 1)^2 / [(1+x)^2 - 2 / (x^4 + 1)^2 - 3]\]

Rational equations can be readily solved by converting them into polynomial equations and solving these. To convert a rational equation into a polynomial equation we make repeated use of the following axioms:

1. \(y = z \rightarrow y - z = 0\)
2. \(y = 0 \rightarrow y.z = 0\)
3. \(w = z.z' \& w = v.v' \rightarrow y/z + u/v = (y.z' + u.v')/w\)
4. \(w = z.z' \& w = v.v' \rightarrow y/z + u/v = (y.z' - u.v')/w\)
5. \( y = y/1 \)
6. \( y^{-z} = 1/y^z \)
7. \( y.(u/v) = (y.u)/v \)
8. \( (y/z)/w = y/(z.w) \)
9. \( y/(z/w) = (y.w)/z \)
10. \( (y/z)^w = (y^w)/(z^w) \)
11. \( y/z = w \rightarrow y = w.z \)

All applied as rewrite rules left to right.

For example:
1. \( (1+1/x)^2 / [(3.x+2)^-2 / (2.x^3-3)^-3] = 1 \)
2. \( ((x+1)/x)^2 / [(2.x^3-3)^3 / (3.x+2)^2] = 1 \) by 3., 6., 7., 8. and 9.
3. \( (x+1)^2.(3.x+2)^2 / [x^2.(2.x^3-3)^3] = 1 \) by 10., 8. and 9.
4. \( (x+1)^2.(3.x+2)^2 = x^2.(2.x^3-3)^3 \) by 11.

What is the right strategy for applying these axioms? Here are two observations.

1. Application of axioms like 3. and 4. requires a small strategy of its own. Faced with a term of the form:
   \( A/B + C/D \)
   we would like to use axiom 3 to combine both subterms under a common denominator. This means finding terms \( E, B' \) and \( D' \) such that \( B.D' = E = D.B' \). Ideally we would like \( E \) to be the least common multiple of \( B \) and \( D \), although we would be satisfied with any common multiple if the least one proved difficult to calculate. If \( A, B, C \) and \( D \) are not already polynomials we should first convert \( A/B \) and \( C/D \) into expressions of the form:
   \(<poly>/<poly>\)
   by applying 3-10. recursively. Suppose \( A, B, C \) and \( D \) are all polynomials. Factorize \( B \) and \( D \) completely and delete any common factors from each to form \( B' \) and \( D' \). Let \( E = B.D' \) and therefore \( E = D.B' \).

2. It would be nice to have a general mechanism which given:
   1. A formula
   2. The Backus-Naur form of a normal form for the formula.
   3. A set of axioms (rewrite rules).

Could convert the formula into the normal form defined by the BNF, using the axioms. This could be done by
1. Parsing the formula bottom up.

2. Finding the first place at which it differed from the normal form.

3. Finding a rewrite rule whose left hand side matched this place and applying it.

4. Repeating the process.

9.0 SIMPLE TRIGONOMETRIC EQUATIONS

There are several classes of equations, which have been singled out for special attention, because equations from them occur so often in applications of equation solving. The most common of these classes is the polynomial equations, another is the simple trigonometric equations, which we deal with here.

9.1 Definition

By a simple trigonometric equation we mean a rational equation in several trigonometric functions, where a trigonometric function is a trigonometric symbol followed by an angle, and an angle is a linear combination of unknowns. A more precise definition follows:

<trig symbol> ::= sin/cos/tan/sec/cosec/cot
<angle> ::= <unknown> / <const>+<angle> / <angle>+<const> / <const>,<angle> / <angle>.<const>
<trig function> ::= <trig symbol>(<angle>)
<trig ratn> ::= <const> / <trig function> / <trig ratn>+<trig ratn> / <trig ratn>-<trig ratn> / <trig ratn>.<trig ratn> / <trig ratn>/<trig ratn> / <trig ratn>^<integer>

9.2 Main Strategy

The main strategy in solving trigonometric equations is to try to re-express the equation as a rational equation in a single trigonometric function, change the unknown to this function, solve the rational, resubstitute the trigonometric function and then solve for the original unknown using isolation, e.g.

1. Solve \(8(\sin x)^2 + 6 \cos x - 9 = 0\)
2. \(8(1-(\cos x)^2) + 6\cos x - 9 = 0\)
3. \(8(\cos x)^2 - 6\cos x + 1 = 0\)
4. \((4\cos x - 1)(2\cos x - 1) = 0\)
5. \(\cos x = 1/4 \text{ or } \cos x = 1/2\)
6. \(x = 2n\pi + \arccos(1/4) \text{ or } x = 2n\pi - \arccos(1/4) \text{ or } x = 2n\pi + \pi/3\)
   \(x = 2n\pi - \pi/3\)

(this example was taken from Hobson and Jessop 1896 p 244)

The above example illustrates one important technique for re-expressing the equation in terms of a single trigonometric function, that of making all the trigonometric symbols the same. In this case \((\sin x)^2\) is expressed in terms of \(\cos x\). In general we will also have to make all the angles the same. This requires a different technique, which is illustrated in our second example (also taken from Hobson and Jessop).

1. Solve \(\sin 3x = 3\sin x\) for \(x\)
2. \(3\sin x - 4(\sin x)^3 = 3\sin x\)
3. \((\sin x)^3 = 0\)
4. \(\sin x = 0\)
5. \(x = n\pi\)

Of these two techniques it seems preferable to perform the latter first, i.e. To make all the angles the same, then make all the symbols the same. This is because it is always possible to express one symbol in terms of another without changing the angle, but not vice versa.

**9.2.1 Equal Angle Strategy**

We will now investigate the task of changing the equations so that all the angles are equal. Suppose that we have written all the angles in polynomial normal form and that they are:
\[a_i x + b_i \text{ for } i \leq n\]
the identities that are useful in this task are:

1. the "sum of angle" formula
   e.g. \(\cos(y+z) = \cos y \cos z - \sin y \sin z\)
2. and the special cases of these:
   e.g. \(\cos 2y = 1 - 2(\sin y)^2\)
3. the "negation" formulae
   e.g. \(\sin(-y) = -\sin y\)
Equation Solving

4. and the other "symmetry" formulae
  e.g. sin(\(\pi/2 - y\)) = \cos y

If all the \(a_i\) are integers we can always make do with type 1. and 3. as follows:

1. first make all the \(a_i\) positive by applying a negation identity to any that are not.

2. find the greatest common divisor, \(d\), of the \(a_i\). Let \(c_i \cdot d = a_i\) for all \(i \leq n\). Express each angle in the form:
   \[d \cdot x + d \cdot x + \ldots + d \cdot x + b_i\]
   (where \(d \cdot x\) is repeated \(c_i\) times) then apply "sum of angle" identities repeatedly until all angles are of the form \(d \cdot x\).

The identities of type 2. and 4. can sometimes be used to cut corners. Even if all the \(a_i\) are not integers we may still be able to find some \(d\) which divides all of them and carry out step 2.

9.2.2 Equal Symbol Strategy -

There are a variety of ways of making all the trigonometric symbols in an equation the same, and which way is used will depend on the circumstances. The identities which are useful are:

1. the definitions of tan, sec, cosec and cot
   e.g. \(\cot y = \cos y / \sin y\) These can always be used to eliminate all but a pair of symbols. These do not have to be \(\cos\) and \(\sin\), we might prefer \(\tan\) and \(\sec\) for instance if this resulted in the smallest number of occurrences of trigonometric symbols. Preferred pairs are \(\cos\) and \(\sin\), \(\tan\) and \(\sec\), \(\cot\) and \(\cosec\), because of the formulae below.

2. the "square" formulae
   e.g. \((\tan y)^2 + 1 = (\sec y)^2\) These can be used to eliminate one member of the previous pair in favour of another. We prefer to eliminate trigonometric functions to an even power (e.g. \((\sec y)^6\) ), otherwise some square roots will be introduced and will have to be eliminated if the equations are to remain rational.

3. the "collection" formulae
   e.g. \(a \cdot \cos y + b \cdot \sin y = \sqrt{a^2 + b^2} \cdot \sin(y + \arcsin(a / \sqrt{a^2 + b^2}))\)
   It is sometimes possible to use adapted versions of the sum of angles formulae to collect together occurrences of the unknown, and so avoid introducing square roots. However, these formulae change the angle of the trigonometric functions and are to be avoided unless there are only two occurrences of the unknown.

4. the "half angle" formulae
   e.g. \(\cos y = (1 - t^2) / (1 + t^2)\) where \(t = \tan(y/2)\) Another way of avoiding introducing the square root is to express all the trigonometric functions in terms of \(\tan(x/2)\), where \(x\) is the unique angle of the equation. This always works but complicates the equation.
a lot (for instance if it was previously a polynomial, it will become a rational) and is to be tried only as a last resort.

Although the above strategies will usually work they do not always yield the neatest solutions. We now consider some alternative strategies.

9.3 Factorization

Factorization is a strategy which can work on equations of several unknowns. So we do not need to express the equation in terms of a single trigonometric function to use it. In addition whereas factorization of polynomials is based mainly on the distributive law in trigonometry we have an additional class of axioms which it can use, namely the "sum and difference" formulae e.g. \( \sin y + \sin z = 2 \sin(y+z)/2 \cdot \cos(y-z)/2 \)

Like the distributive law these can be used to make multiplication the dominant function of a term. Consider the following example from Borchardt and Perrott 1904 p 270.

1. Solve \( 3 \sin 7x - 2 \sin 4x + 3 \sin x = 0 \)
2. \( 3(\sin 7x + \sin x) = 2 \sin 4x \)
3. \( 6 \sin 4x \cdot \cos 3x = 2 \sin 4x \)
4. \( \sin 4x = 0 \) or \( 3 \cos 3x - 1 = 0 \)
5. \( 4x = n \pi \) or \( 3x = 2n \pi + \arccos 1/3 \) or \( 3x = 2n \pi - \arccos 1/3 \)

One cannot help feeling that this question was "cooked" a bit in order to make it "work out". However the very unlikelihood of the strategy succeeding (and its neatness if it does) make it a good candidate to try first, provided its chances of success can be quickly ascertained.

10.0 CHANGE OF UNKNOWN

Several of the strategies developed for polynomial and simple trigonometric equations can be generalized to apply to a wider class of equations. For instance, in the main strategy for simple trigonometric equations we started with a rational equation in several trigonometric functions, converted this into a rational equation in a single trigonometric function and then changed the unknown into this single trigonometric function.
10.1 The Strategy

The generalized strategy is a process of continuous parsing. We parse the equation as if it were a member of some class of equations called the base class which we are good at solving e.g. The polynomial equations. If we come to a term which fails the parse we ask the following questions:

1. Would it help to extend the base class. E.g. If the term were of the form $a/b$ or $a^{-n}$ then extending the base class to the rational equations would enable the parse to continue to the next step.

2. Failing this, would the parse succeed if the offending term were treated as an unknown. E.g. If the term were of the form $a^b$ where $b$ was not an integer, then the parse would succeed if $a^b$ were regarded as an unknown.

In the first case we extend the base class and continue. In the second case we put the offending term in an offenders' set, record the current state of the parse, then continue the parse treating the offending term as an unknown. Otherwise we fail. If the actual unknown is ever parsed, as an isolated term, it must also be added to the offenders' set.

If the parse succeeds we will have an offenders' set:

$$(c_1(x), \ldots, c_n(x))$$

and an equation

$$e(c_1(x), \ldots, c_n(x))$$

which is a member of the base class in terms of the offending terms $c_i(x)$ $0 < i < n$.

If the offenders' set is a singleton $(c_1(x))$ we make a change of unknown, $y = c_1(x)$, solve

$$e(y)$$

as a member of the base class, to get solution

some $n$, $s$ $(in(n,s) \& y = g(n))$

resubstitute and solve

some $n$, $s$ $(in(n,s) \& c_1(x) = g(n))$

If the offenders' set is not a singleton then we try to transform the equation until it is. We pick one of the offending terms, say $c_i(x)$. We try to find an identity axiom which is applicable to $c_i(x)$ and which if applied would allow the parse to continue from where it was previously frozen, we have here exercised choices and must be prepared to come back and remake them. If we succeed in allowing the parse to continue we repeat the above procedure.

To illustrate this strategy consider the following example from Tranter 1972 p.28.
10.1.1 Example -
Solve $5^2x-5^7(x+1)+4=0$

We start by trying to parse this as an integer polynomial. The parse succeeds with offends set

$$(5^2x, 5^7(x+1))$$

Suppose we decide to work on $5^2x$.

The axiom

$$w^x(u,v)=(w^v)^x$$

is applicable to $5^2x$ and changes it to $(5^x)^2$. This allows the parse to continue with new offends set

$$(5^x, 5^x+1)$$

No axiom is applicable to $5^x$, so we try $5^7(x+1)$ the axiom

$$w^y(u,v)=(w^y)^u$$

is applicable to $5^7(x+1)$ and changes it to $(5^x)(5^7)$ which is simplified to $5.5^x$. This allows the parse to continue with new offends set $(5^x)$. So we change the unknown by setting $y=5^x$ and form the new equation

$$y^2-5y+4=0$$

Which is solved as a quadratic in $y$.

10.2 The Base Class

For a particular class of equation to be useful as a base class we must have some facility at solving equations in that class. In fact the more confident we are that we can solve an equation from that class the better candidate it is. What classes do we know something about? Polynomial equations with coefficients from various sets; rational equations; simple trigonometric equations.

These classes also have another desirable feature: they fit together into a simple structure which enables us to consider all these base classes in one parse. We can start by parsing the current equation as if it were an integer polynomial equation. If this breaks down we investigate the first term which it fails to parse as an integer polynomial. If this is a non-integer constant then we can extend the base class to the real polynomials. If its dominant function is $^+$ or $/$ then we extend to the rationals. If its dominant function is trigonometric we can extend to the simple trigonometric functions. Similarly if the rationals are the correct base class and we fail to parse a trigonometric function then we can extend to the simple trigonometric equations.

We cannot begin by using the simple trigonometric equations as a base class because then we would fail to recognise any non-constant rational equation. We could begin by adopting the rationals as base class, but then we would not be able to assess the chances of success as accurately as if we had kept the base class as small as possible. We need to spell out precisely the conditions under which one base class can be extended to another so that new potential base classes can be fitted into this structure by the system itself.
10.3 Re-expressing Offending Terms

This part of the strategy: the application of identity axioms to offending terms in an attempt to allow the parse to continue, requires a deductive system of its own. Although hopefully it will be very similar to the basic system. We may have to apply the identity axiom not to the offending term on its own, but to some subterm, for instance in our example we applied \[w^\w(u.v) = (w^w)^v\]

to
\[5^w(2x)^2\]

but suppose we only had the axiom
\[w^\w(u.v) = (w^w)\w v\]
to succeed we would first have had to apply the commutative law of multiplication to \[2x\]. This example could have been handled by the pattern matcher (see pattern matching section), but we can easily think of examples which could not, for instance, re-express \(\sin 2x\) using the axiom
\[\sin(u+v) = \sin u \cdot \cos v + \cos u \cdot \sin v\]

we may have several axioms which apply to the offending term and allow the parsing to continue. For instance
\[\sin u = -\sin(-u)\]
also applies to \(\sin 2x\). We will have to choose one to try, and back up if it fails, some of these alternatives may be essentially equivalent, for instance applying
\[\sin 2u = 2\sin u \cdot \cos u\] to \(\sin 2x\)
is equivalent to applying \(\sin(u+v) = \sin u \cdot \cos v + \cos u \cdot \sin v\)

There is a great deal of scope here for guidance and the strategy we have outlined is obviously incomplete without this guidance. We still do not seem to have used completely the fact that we are trying to re-express all the offending terms in terms of a single reduced term. We must link together the processes of re-expressing the different offending terms.

We also have to choose which of the offending terms to work on first. Not so much hangs on this, as all the terms must be worked on eventually. A good heuristic seems to be to work on the most complex first, as we may succeed in expressing it in terms of one of the less complex terms and thus save working on the latter at all. When we have solved the harder problem of linking together the processes of re-expressing the different offending terms, we may understand this easier problem better.

10.4 The Reduced Term

It is important to try and spot what is going to be the future reduced term as early as possible during the course of the strategy. This information can then be used to guide the re-expression of the offending terms. That is we will prefer to use identity axioms which allow the parse to continue and express the offending term in terms of the potential reduced term. We will also choose to work on terms which we do not suspect of being the reduced term using the following information:
1. It will eventually occur as one of the offending terms.
2. It will probably be the one of the least complex terms.
3. We will prefer those terms which occur the greatest number of times in the current equation.
4. We will reject isolated occurrences of the unknown.

Thus during the course of the strategy we will maintain a current hypothesis about the identity of the reduced term which we will use to guide our search.

11.0 FACTORIZATION

Another strategy which can be generalized from its specific uses in solving polynomial and simple trigonometric equations is factorization. Like the change of unknown strategies it can be use to simplify equations. We usually use it to replace an equation with several simpler equations. Consider the following example.

11.1 Example

1. \( \sin x \cdot \cos x + \cos x = (\cos x)^2 + \sin x \)
2. \( \sin x \cdot (\cos x - 1) = \cos x \cdot (\cos x - 1) = 0 \)
3. \( (\cos x -1) \cdot (\sin x - \cos x) = 0 \)
4. \( \cos x - 1 = 0 \) or \( \sin x - \cos x = 0 \)

This example was part of the written protocol of one of our experimental subjects. The accompanying verbal protocol was:

"Now it looks as if by collecting terms we might be able to factorize this. In fact taking the \( \sin x \cdot \cos x \) on the left with the \( \sin x \) on the right we get \( \sin x \cdot (\cos x + 1) \) and taking the \( \cos x \) on the left with the \( (\cos x)^2 \) on the right we get \(-\cos x \cdot (\cos x - 1)\) all equal to 0. Giving as our factors \( (\cos x - 1) \cdot (\sin x - \cos x) = 0.\)

What factorization algorithm could be responsible for discovering these factors? We may make a hypothesis.

1. Maybe the subject made some implicit changes of unknown by setting \( c=\cos x \) and \( s=\sin x \), then factorized the new equation as a polynomial equation in two unknowns.
2. It is possible to factorize the resulting polynomial as the subject did by regarding it as a quadratic in c, where s is a constant. However, if our subject did this he did not put the quadratic into normal form before factorizing it, or his written protocol would have included the step:

\[(\cos x)^2 - (\sin x + 1)\cos x + \sin x = 0\]

Instead the subject was able to group the polynomial into 2 sets of 2 pairs, apply the distributive law to each pair to get a common factor: \(\cos x - 1\), then extract this common factor with the distributive law again.

11.2 Matters Arising

All these considerations suggest the following:

1. That the change of unknown strategy be amended, so that when it cannot reduce its offenders' set to a singleton, it changes each member of the set to a new unknown and try to factorize the resulting equation.

2. That a factorization algorithm be developed for polynomials in several unknowns. As a first attempt we can use a recursive call of a single unknown algorithm. That is, following Manosev 1967 p87, we can express the multi-unknown polynomial as a polynomial in the other unknowns. To factorize a polynomial we first recursively factorize its coefficients, then factorize it using the single unknown algorithm.

3. That the Intuitive Factorization Algorithm be modified so that it can imitate our example protocol.

11.3 General Factorization

All the previous example of factorization are based on the left/right use of the distributive law

\[x.y + x.z = x.(y+z)\]

and are essentially polynomial factorization. In fact terms can be factorized using any identity of the form

\[f(a1,...,an) = b.c\]

where \(f\) is not multiplicative. Consider for instance.

1. \(\log(e,x^x) = 0\)
   \(x.\log(e,x) = 0\)
   Therefore \(x=0\) or \(\log(e,x)=0\)

2. \(\sin(2.x) - \sin(pi/4 - x) = 0\)
   \(2.\cos(2.x + pi/4 -x)/2.\sin(2.x - pi/4 + x)/2 = 0\)
   Therefore \(\cos(x/2 + pi/8) = 0\) or \(\sin(3.x/2 - pi/8) = 0\)
Neither of these examples is very natural in that there are alternative methods of solution for each equation which do not involve factorization. So whether the notion of factorization should be extended to include the above examples is a moot point.

12.0 SIMULTANEOUS EQUATIONS

In this section we turn our attention to the solution of several equations in several unknowns simultaneously. To extend our ideas on single equations to simultaneous equations two further techniques are needed:

1. The ability to eliminate unknowns between equations.
2. and the ability to form an overall plan (the Grand Plan) of which unknowns to eliminate when.

12.1 Elimination

Elimination is the process of forming a new equation out of two (or more) existing equations, such that the new equation does not contain an unknown which is in the original equations. Consider the following example.

1. \( x + y = 7 \)
   \( x - y = 1 \)

2. Eliminating \( y \) by adding
   \( 2x = 8 \)

In this example \( y \) has been eliminated by 'adding' the two original equations so that the two occurrences of \( y \) cancel each other. Alternatively, \( x \) could have been eliminated by 'subtracting' the two equations.

1. \( x + y = 7 \)
   \( x - y = 1 \)

2. Eliminating \( x \) by subtracting
   \( 2y = 6 \)

A simple form of Elimination can be made from an equation solver for one equation in one unknown. Suppose \( p(x,y) \) and \( q(x,y) \) are two equations in two unknowns then if each of them are solved for \( y \) in terms of \( x \) to form:
   \[ y = p'(x) \quad \text{and} \quad y = q'(x) \]
then \( p'(x) = q'(x) \) is a new equation in \( x \) without \( y \). Let us call this method 'stripping'.

The methods of 'adding' and 'subtracting' illustrated above are no more powerful than 'stripping', but they do cut corners and lead to simpler solutions in some cases as illustrated by the following example.
1. \( \sin(x) + \tan(y) = 1 \)
   \( \sin(2x) - \tan(y) = 0 \)

2. Eliminating \( y \) by adding
   \( \sin(x) + \sin(2x) = 1 \)

The 'stripping' method would have wasted time here by isolating in the two equations and producing the more complex equation:
   \[ n \cdot \pi + \arctan(1 - \sin(x)) = n \cdot \pi + \arctan(\sin(2x)) \]

But 'stripping' is more powerful, as it applies in cases where 'adding' and 'subtracting' fail. Consider
   \( \sin(x) + \log(e, y) = 1 \)
   \( x^3 + \tan(y) = 2 \)

Neither \( x \) nor \( y \) can be Eliminated without 'stripping'.

Therefore, for best results, all three methods should be available: 'adding' or 'subtracting' should be applied first if possible, but if not 'stripping' should be used.

Simple versions of 'adding' and 'subtracting' will be easy to implement. The following examples illustrate some more sophisticated versions.

1. \( x + 2y + 3z = 0 \)
   \( 7x - 3y - 2z = 6 \)
   \( 2x + 4y - z = 7 \)

2. Eliminating \( z \) by adding
   \( 10x + 3y = 13 \)

Here \( z \) is Eliminated by 'adding' all three equations.

1. \( x^2 + \sin(y) = 4 \)
   \( x + \cos(y) = 3 \)

2. Isolating the trigonometric terms and squaring
   \( \sin(y)^2 = (4 - x^2)^2 \)
   \( \cos(y)^2 = (3 - x)^2 \)

3. Eliminating \( y \) by adding
   \( 1 = (4 - x^2)^2 + (3 - x)^2 \)

Here considerable preparation was put in before the 'adding' was done. Even the 'adding' step was not simple, since the two terms involved, \( \sin(y)^2 \) and \( \cos(y)^2 \), were not negations of each other, but combined to give a term not containing \( y \), using the identity:
   \( \cos(u)^2 + \sin(u)^2 = 1 \) We cannot at present see how to implement this more sophisticated version of 'adding'.

---

Equation Solving

Page 34
12.2 Forming A Plan

The two strategies, elimination and solve, can be combined to solve systems of simultaneous equations in several unknowns. But before they can be used a plan has to be made to show how they are combined. These plans can often be designed using only the knowledge of which unknowns occur in which equations (see Polya 1962 p129-130), e.g. Suppose we have three equations \( p \), \( q \) and \( r \), and that:

- \( p \) contains precisely \( x \) and \( y \);
- \( q \) contains precisely \( x \), \( y \) and \( z \);
- \( r \) contains precisely \( y \) and \( z \).

The kind of plan which immediately suggests itself is:

1. Eliminate \( x \) between \( p(x, y) \) and \( q(x, y, z) \) to form \( s(y, z) \).
2. Eliminate \( y \) between \( s(y, z) \) and \( r(y, z) \) to form \( t(z) \).
3. Solve \( t \) for \( z \).
4. Substitute the value of \( z \) into \( r(y, z) \) to form \( r'(y) \) and solve \( r'(y) \) for \( y \).
5. Substitute the value of \( y \) into \( p(x, y) \) to form \( p'(x) \) and solve \( p'(x) \) for \( x \).

This plan is certainly not unique, for instance, we could have reversed the roles of \( x \) and \( z \), and \( p \) and \( r \). This would have been an advantage if we were primarily interested in the value of \( x \), and not the values of \( y \) or \( z \). In this case we could have terminated the plan at step 3. Reversing these roles would also have been an advantage if the elimination of \( y \) between \( p(x, y) \) and \( q(x, y, z) \) were easier than the elimination of \( x \) between \( p(x, y) \) and \( q(x, y, z) \). We could also form different plans by eliminating \( z \) (instead of \( y \)) in step 2 or by using \( q \) (instead of \( p \)) in step 5 etc.

In fact there are an infinite number of alternative plans, most of which look less efficient than our original, for instance:

1. Eliminate \( y \) between \( p(x, y) \) and \( r(y, z) \) to form \( s(x, z) \).
2. Eliminate \( x \) between \( s(x, z) \) and \( q(x, y, z) \) to form \( t(y, z) \).
3. Eliminate \( z \) between \( t(y, z) \) and \( r(y, z) \) to form \( u(y) \).
4. Solve \( u(y) \) for \( y \).
5. Substitute the value of \( y \) into \( t(y, z) \) to form \( t'(z) \) and solve \( t'(z) \) for \( z \).
6. Substitute the values of \( y \) and \( z \) into \( q(x, y, z) \) to form \( q'(x) \) and solve \( q'(x) \) for \( x \).
If we are to design a procedure for constructing such plans we will have to state precisely the knowledge used to construct them, together with information about how to construct efficient plans and how to break ties. We used steps of the following kinds in constructing the above plans:

1. If \( p \) is an equation in the set of unknowns \( \{x\} \cup \text{uset1} \) and \( q \) is an equation in the set of unknowns \( \{x\} \cup \text{uset2} \) then we can eliminate \( x \) between \( p \) and \( q \) to form a new equation \( r \) (say) in the set of unknowns \( \text{uset1} \cup \text{uset2} \) (where \( \cup \) means disjoint union).

2. If \( p \) is an equation in the set of unknowns \( \{x\} \cup \text{uset} \) and we have a solution for each unknown in \( \text{uset} \) then we can substitute these solutions into \( p \) to get a new equation \( p' \) and solve \( p' \) for \( x \), to get a solution for \( x \).

Given a system of simultaneous equations these two steps define an, often infinite, search space. To each of the steps we may attach a cost i.e. the estimated difficulty of eliminating an unknown or of solving an equation. Thus the space can be generated in a "graph traverser" like fashion until a least costly, complete plan is discovered. A complete plan is a branch of the space on which there exists a solution for each unknown for which we sought a solution.

This is not the finish of it. We cannot compute the exact cost of a step until it has been performed, and we will not perform any steps until a plan has been found. Indeed at the time of forming the plan we may not know the exact form of some of the equations involved in it, i.e. Those equations formed by elimination or substitution, and may have extreme difficulty in making an estimate. We must, therefore, allow the possibility of revising the plan in the light of better estimates (especially if some step is found to be un-performable), and so must keep the partially generated search space for future use. Any step which has already been performed, will, of course, have zero cost.

This account is a bit simplistic because the general solution of an equation does not provide a single value, but a set of alternative values (see legal moves section). The substitution of all these values into an equation produces a disjunction of equations all of which must be solved. Elimination may also create not just a single equation but a disjunction of equations, and each of these may become involved in a different plan. Provided this does not happen, however, the above account can easily be adapted to cover disjunctions of equations.

12.3 A More Efficient Procedure

We would like to add to the above account the following conjecture. That there is a more specific and more efficient procedure, which is always adequate to find the most efficient plan, and is often used by humans. We have not proved these statements, but we can define the procedure:
1. initially let:
   1. The "current equation set" be the system of simultaneous equations we are trying to solve.
   2. The "unknowns set" be the set of all unknowns in the current equation set.
   3. The "saved equations list" be nil.
   4. The "solutions list" be nil.

2. Choose an unknown, x, (say) from the unknowns set, and remove it from the set.

3. Remove from current equation set all those equations which contain x, and make a set of them called the "x equation set".

4. Choose an equation from the x equation set and call it the "x equation".

5. For each of the equations, e, (say) in the x equation set minus the x equation, eliminate x between the x equation and e, and put the resulting equation in the current equation set.

6. Save the x equation on the saved equations list, but throw away the other equations in the x equation set as they will not be needed again.

7. If the unknowns set is non-empty, go to 2.

8. The top equation on the saved equations list contains only one unknown, x, (say). Remove this equation from the list, solve it for x and substitute the resulting value for every occurrence of x in the remaining equations on the saved equations list. Save the solution on the solutions list.

9. If we now have a solution for each unknown for which we sought a solution stop with success, otherwise go to 8.

This procedure is non-deterministic unless we also specify how to make the choices in steps 2 and 4, so we will provide some heuristics for making these choices:

In step 2 choose first (to eliminate) those unknowns whose value is not required. Otherwise choose first those unknowns which are easiest to eliminate.

In step 4 choose the equation which will be easiest to solve for x when values are substituted for the other unknowns.

Note: we could replace step 5 with a more exotic way of eliminating all occurrences of an unknown from a set of equations. E.g. Simultaneous elimination between three or more equations.
12.4 Reversibility

We can extend the notion of reversibility to cover systems of simultaneous equations. **Definition:** an axiom \((p \& q) \rightarrow r\) which can be used to justify the elimination of an unknown between the equations which match \(p\) and \(q\) to form the equation which matches \(r\) is said to be "left reversible" if
\[(p \& q) \leftrightarrow (p \& r)\]
"right reversible" if
\[(p \& q) \leftrightarrow (q \& r)\]
and "reversible" if it is both left and right reversible. An application of a (left, right) reversible axiom is said to be a (left, right) reversible step and a series of (left, right) reversible steps is said to be a (left, right) reversible branch. Note that applications of the simple elimination strategies of "adding" and "subtracting" always lead to reversible branches. Alternatively we could regard the elimination axiom
\[(p \& q) \rightarrow r\]
as the implication axiom
\[p \rightarrow (q \rightarrow r)\]
which is reversible if and only if the elimination axiom is left reversible. It is easy to prove (by induction) that if all the applications of elimination in step 5 are left reversible then the original system of simultaneous equations is equivalent to the conjunction of the equations which are in the saved equations list when step 8 is entered for the first time.

Substitution is always reversible in the old sense. So if all the solutions to the saved equations obtained in step 8 are most general solutions, it is easy to prove (again by induction) that the conjunction of equations which are in the saved equations list when step 8 is entered for the first time is equivalent to the conjunction of the equations in the final solution list and the equations in the final saved equations list.

Therefore if all eliminations are left reversible, and all solutions are most general, the original system of simultaneous equations is equivalent to the final solutions plus any saved equations which are left, and no checking of solutions is necessary.

13.0 Pattern Matching

The application of an axiom to an equation calls for terms in the axiom to be matched against terms in the equation, i.e. we must find values for the variables in the axiom which make the axiom’s terms equal to the equation’s terms.

For instance, we might apply:
\[
\cos(u+v) = \cos u \cos v - \sin u \sin v
\]
to
\[
4 \cdot \cos 3x + 3 \cdot \cos x = 0
\]
by matching \(\cos(u+v)\) to \(\cos 3x\) and getting the substitution \(\{2x/u, x/v\}\) then replacing \(\cos 3x\) by \(\cos 2x \cdot \cos x - \sin 2x \cdot \sin x\) to get
\[
4(\cos 2x \cdot \cos x - \sin 2x \cdot \sin x) + 3 \cdot \cos x = 0
\]
The process of doing this is called *pattern matching*. Such processes are very common in theorem proving - for instance, the Resolution unification algorithm. Our pattern matching process will have to differ from the unification algorithm in two respects, namely

1. We will never have variables in the equation we are trying to solve, only in the axioms. The unification algorithm with this restriction is sometimes called *left unification* (see Plotkin 1972).

2. The normal unification algorithm knows nothing about the properties of the theory it is working in, so it would fail to unify:
   \[ ax+b \] with \[ 3y+2 \]
because it does not know about the commutativity of addition or 'zero' role of \( o \). This can be extremely irritating and many authors have overcome the problem by building axioms into the unification algorithm (see e.g. Plotkin 1972) and dropping them from the axiom set.

   Such a step may be essential to us. A particular strategy may, on the basis of its description of the current equation, narrow down its choice of axioms to one or two. If the application of these axioms failed because of a trivial failure of the pattern matcher then the opportunity to use this strategy would be lost.

The ultimate extension of the idea of building-in axioms to the pattern matcher is to treat pattern matching as equation solving and call the equation solver recursively to match terms. This is a very powerful method of matching and can be very successful provided.

1. Care is taken that the process does not loop. This can be done easily by making sure the terms to be matched are not maximal, e.g. like \( A \) in \( A \in 0 \).

2. Attempts at unification are not undertaken lightly (since so much effort can be expended in their satisfaction). The various strategies are a great help here in narrowing down the number of matches to be attempted. [1]

14.0 **CONCLUSION**

In this memo we have explored methods of controlling the search involved in an area of Mathematics, namely equation solving. The methods we have discovered have used high level (meta-language) concepts for describing expressions and the strategies for guiding proofs. With the aid of these strategies the amount of search involved can be considerably reduced and the combinatorial explosion avoided.

Some of the techniques have already been successfully tested in a program, PRESS. Some remain to be tested - and this is the next step.

[1] This possibility is explored more fully in Bundy 1975.
15.0 APPENDICES

15.1 Appendix A - Definition Of The R Elementary Functions

We are restricting our attention to equations between R Elementary Functions. A definition of this class of terms follows:

1. \(<R\ \text{elem func}> =: <\text{constant}> / <\text{func sym 1}(<R\ \text{elem func}>) / <\text{func sym 2}(<R\ \text{elem func}>, <R\ \text{elem func}>)

2. <\text{func sym 1}> =: -/sqrt/sin/cos/tan/cosec/sec/cot/arcsin/arccos/arctan/arccosec/arcsec/arccot

3. <\text{func sym 2}> =: +/-/.///~/log/root

4. <\text{constant}> =: <\text{real}> / <\text{arb const}> / <\text{unknown}>

5. <\text{real}> =: <\text{integer}> / \pi / e

6. <\text{integer}> =: 1/2/3/4/5/........

7. <\text{arb const}> =: a/b/c/......

8. <\text{unknown}> =: x/y/z/......

15.2 Appendix B - General Equivalence Theorem

The problem of finding the most general solution to an equation \(e(x, a)\) for \(x\) in terms of \(a\), under the conditions, \(p(a)\), can be formalized as:

\[
(1) \quad \text{all } a, x \left[ p(a) \rightarrow \text{some } n, s, f \left\{ \text{in}(n, s) \& \left( x = f(n, a) \leftrightarrow e(x, a) \right) \right\} \right]
\]

where \(f(n, a)\) is a term not containing \(x\) and \(a\) is a vector of constants.

Formula (1) breaks into two sub-formulae:

\[
(1.1) \quad \text{all } a, x \left\{ p(a) \& e(x, a) \rightarrow \text{some } n, s, f \left\{ \text{in}(n, s) \& x = f(n, a) \right\} \right\}
\]

and

\[
(1.2) \quad \text{all } a, x \text{ some } n, s, f \left\{ p(a) \& x = f(n, a) \rightarrow \{ \text{in}(n, s) \& e(x, a) \} \right\}
\]

Formula (1.1) is used to find candidate solutions and formula (1.2) is used to check them. Dually skolemized (so that it can be used as a goal) (1.1) becomes:

\[
(1.1') \quad p(a) \& e(x, a) \rightarrow \text{in}(N, S) \& x = F(N, a)
\]

where upper case letters denote variables and lower case letters denote (skolem) constants.

The first formalization of the equation solving problem (from section 3.2) was:

\[
(2) \quad \text{all } a \text{ some } x \left[ p(a) \rightarrow e(x, a) \right]
\]
15.2.1 Theorem 1 (General Equivalence Theorem) - Formula (1.2) is equivalent to formula (2).

Proof

(1.2) => (2)
assume (1.2)

(1.2) all \( a, x \) some \( n, s, f \) \[
\{(p(a) & x=f(n,a)) \rightarrow \{in(n,s) & e(x,a)\}\}
\]
(1.2') \( p(A) & X=f(n0,A) \rightarrow \{in(n0,s0) & e(X,A)\} \)
by universal and existential instantiation

assume \( p(a0) \)

Therefore \( X=f(n0,a0) \rightarrow \{in(n0,s0) & e(X,a0)\} \) by (1.2')

hence in\( (n0,s0) & e(f(n0,a0), a0) \) by \( U=U \)

\( p(a0) \rightarrow e(f(n0,a0), a0) \) discharging assumption

all \( a \) some \( x \) \( \{p(a) \rightarrow e(x,a)\} \)
by universal and existential introduction

(2) => (1.2)
(2) all \( a \) some \( x \) \( \{p(a) \rightarrow e(x,a)\} \)
(2') \( p(A) \rightarrow e(x1,A) \)
by universal and existential instantiation

define \( \text{foo}(U,V)=x1 \)

\( \text{foo}(U,V)=x1 \)
\( p(\overline{1}) & x2=\text{foo}(1,\overline{1}) \)

Therefore \( e(x1,\overline{1}) \) by (2')

hence \( e(x2,\overline{1}) \) by assumption

also \( \text{in}(1,\{1\}) \) from set theory

\( \{p(\overline{1}) & x2=\text{foo}(1,\overline{1})\} \rightarrow \{\text{in}(1,\{1\}) & e(x2,\overline{1})\} \)
discharging assumption

(1.2) all \( a, x \) some \( n, s, f \) \[
\{(p(a) & x=f(n,a)) \rightarrow \{in(n,s) & e(x,a)\}\}
\]
by universal and existential instantiation

Q.E.D.

15.3 Appendix C - A Recursively Unsolvable Equation

We show that there is an unsolvable \( R \) Elementary equation by showing that the solution of any Diophantine equation can be expressed as the solution of an \( R \) Elementary equation. and then using the well known result of Matiyasevich, Davis, Putnum and Robinson that there is a recursively unsolvable Diophantine equation

\( Q(y,x1,\ldots,xn) = 0 \)
where the \( x_i \)'s are to be solved in terms of \( y \).

The only tricky bit of our proof is ensuring that all the solutions to our \( R \) Elementary equation are integers. For this we use the observation that

\( \sin(x,\pi) = 0 \) iff \( x \) is an integer

We also need to combine this information with the equation proper to make a new equation. This is done using the observation that

\( x^2 + y^2 = 0 \) iff \( x=0 \) and \( y=0 \)
15.3.1 Theorem 2 (Diophantine Equations In Algebra) - If
$P(y, x_1, \ldots, x_n) = 0$ is a Diophantine equation then it has solutions
$x_i = a_i(y)$ for $1 \leq i \leq n$, where the $a_i(y)$ are integers iff
$\sum_{1 \leq i \leq n} \sin(x_i \pi)^2 + P(y, x_1, \ldots, x_n)^2 = 0$
has solutions $x_i = a_i(y)$ $1 \leq i \leq n$ \[1\]

Proof
$\sum_{1 \leq i \leq n} \sin(x_i \pi)^2 + P(y, x_1, \ldots, x_n)^2 = 0$
iff
$\sin(x_i \pi) = 0$ for $1 \leq i \leq n$ and $P(y, x_1, \ldots, x_n) = 0$
iff
$x_i$ is an integer for $1 \leq i \leq n$ and $P(y, x_1, \ldots, x_n) = 0$

15.3.2 Corollary - Therefore,
$\sum_{1 \leq i \leq n} \sin(x_i \pi)^2 + Q(y, x_1, \ldots, x_n) = 0$
is recursively unsolvable, where $Q(y, x_1, \ldots, x_n) = 0$ is the recursively
unsolvable Diophantine equation given in Matiyasevich 1970.

15.3.3 Corollary - It is also interesting to note that
$\sin(x \pi)^2 + \sin(y \pi)^2 + ((e + \pi).x - y)^2 = 0$
iff
$e + \pi$ is rational
which is a well known open conjecture.

16.0 REFERENCES


Bledsoe, W. W., Boyer, R., Henneman, W. H. "Computer proofs of limit theorems"
Artificial Intelligence 3 (1972) p 27-60.

Borchardt, W. G., Perrott M. A. "New Trigonometry for schools" Bell and sons
ltd (1964).

(1973), P 130-8, Also D.C.L. Memo no. 61, Edinburgh.

Bundy, A. "Analysing Mathematical proofs (or reading between the lines)" DAI

1881.

1962.

[1] The results in this appendix are based on those in Richardson 1966.
Equation Solving


Kowalski, R. "Pre dic ate logic as a programming language" D.C.L. Memo no. 70, School of Artificial Intelligence, Edinburgh (1974).


O’Shea, T. "Some experiments with an adaptive, self-improving teaching system" technical report no 18, Dept of Computer Science, Univ. of Texas at Austin (1973).


Richardson, D. Doctoral dissertation, Univ. of Bristol, Bristol 1966.
