Applications and Methods for Recognition of (Anti)-Symmetric Functions

Richard Fateman
Armando Solar-Lezama
Computer Science
University of California, Berkeley

October 16, 2008

Abstract

One of the important advantages held by computer algebra systems (CAS) over purely-numerical computational frameworks is that the CAS can provide a higher-level "symbolic" viewpoint for problem solving. Sometimes this can convert apparently impossible problems to trivial ones. Sometimes the symbolic perspective can provide information about questions which cannot be directly answered, or questions which might be hard to pose. For example, we might be able to analyze the asymptotic behavior of a solution to a differential equation even though we cannot solve the equation. One route to implicitly solving problems is the use of symmetry arguments. In this paper we suggest how, through symmetry, one can solve a large class of definite integration problems, including some that we found could not be solved by computer algebra systems. One case of symmetry provides for recognition of periodicity, and this solves additional problems, since removal of periodic components can be important in integration and in asymptotic expansions.

1 Introduction

Using symmetry arguments one can solve problems that may otherwise stymie computer algebra systems as well as numerical methods. For example, success in a computation may require that a program exploit the fact that a function is (say) antisymmetric, or periodic *before it raises an alarm about the function being (say) discontinuous or infinite.*

We give a few examples and explore these techniques to see how these ideas can be systematically exploited.

2 A motivating example

Consider

$$\int \frac{1}{\sin x + x}\, dx$$

A CAS will confirm that there is no neat closed-form expression for this integral. If we need to compute a value for this integral between particular bounds, we could resort to numerical integration ("quadrature") programs. They may not work very well if the bounds straddle the origin: as x approaches zero the integrand is unbounded. A good visual plot may help: in addition to showing the vertical asymptote at $x = 0$, it should display the fact that the integrand is antisymmetric: that is for $f(x) := 1/(\sin x + x)$ except possibly for

$x = 0$, we see that $f(x) = -f(-x)$. Also, $f(0) = \pm\infty$ One consequence of the antisymmetry is that one can notice that

$$\int_{-a}^{a} \frac{1}{\sin x + x}\, dx = 0$$

In our tests, one computer algebra system (Macsyma 2.4) figured this out, noting that it was true in a principal value sense. Our first attempt with Mathematica (5.0) resulted in a warning message about $x = 0$. On advice from D. Lichtblau at Wolfram Research, and setting an option `PrincipalValue` to `True` Mathematica 5.0 produced the answer 0. Mathematica 6.0 doesn't seem to work though, returning a message of non-convergence. Maple (version 7) did not find a value even when given an option `CauchyPrincipalValue`. Testing suggests that Mathematica 5.0 was programmed to detect some symmetries about zero because it otherwise would not be able to do some otherwise implausible definite integrals which happen to be antisymmetric about zero. However, its detection mechanism seems fairly weak.

How hard is it to recognize this situation? Apparently neither Macsyma nor Mathematica 5.0 try *very* hard because a simple linear change of variable fools them: it keeps each of them from solving a similar problem below. (A Macsyma user can issue a command called `changevar` to set $y = x + c$. This converts the left to the right side and allows Macsyma to then solve the problem. We could find no equivalent built-in command in Mathematica, although clearly one could be constructed)

$$\int_{-c-a}^{a-c} \frac{1}{\sin(x+c) + x + c}\, dx = \int_{-a}^{a} \frac{1}{\sin y + y}\, dy$$

While we are examining such issues, we should also be able to recognize symmetric and periodic functions (or more general symmetries discussed later).

What is the point here? From the perspective of building a "world class" integration program, symmetric expressions about the point 0 integrated from $-a$ to a can be computed as twice the integral from 0 to a. Relevant to this application, we should be able to extract sub-parts of (anti) symmetric function ranges as well. That is, let I be the integral of $f(x)$ from a to b where f is antisymmetric around c, with $a < c < b$. The value of I could be reduced by a change of variable by subtracting off either the range $[a, 2c - a]$ or $[2c - b, b]$. Which range is used depends on whether c is closer to a (first expression) or closer to b. If this cannot be computed, then the integral can nevertheless be split into three pieces. The middle will be zero by antisymmetry, and at least one of the other two will be zero since it will have equal upper and lower bounds, $[a, c-r]$, $[c-r, c+r]$, $[c+r, b]$. The value of r is the "radius" of the antisymmetry and is $min(|c-a|, |c-b|)$. In the case of a periodic function, we can integrate over one period, extended to the number of complete periods, and then worry about the fractional periods that might be left over.

How can we determine constructively if f is antisymmetric about zero? Two techniques come to mind.

- Simplify $f(x) + f(-x)$ by some method we have in our armamentum of simplification programs.
- Expand f in a symbolic power series about 0. If only odd-degree terms appear, the function is antisymmetric. Only even-degree terms implies symmetric.

  But symmetry about other points may be of interest, in which case it may not always be so obvious.

  $$3\,x^4 - 48\,x^3 + 293\,x^2 - 808\,x + 855 \qquad (*)$$

  is symmetric, but about $x = 4$, since it can be expressed as

  $$7 + 5\,(x-4)^2 + 3\,(x-4)^4 + \cdots.$$

  For a non-polynomial example, Macsyma knows that $\sin x$ expanded about 0 is

  $$\sum_{k=0}^{\infty} \frac{(-1)^k\, x^{2\,k+1}}{(2\,k+1)!}$$

and some work can show all the terms are odd: $\sin x$ is antisymmetric about 0. Mathematica does not have a built-in command, but the `RSolve` package provides `SeriesTerm[Sin[x],{x,0,n}] with answer:`
`(I^(-1 + n)*KroneckerDelta[Mod[-1 + n, 2]]*UnitStep[-1 + n])/Gamma[1 + n]`. This is the coefficient of x^n in the power series.

Neither of the proposed tests is entirely algorithmic for the *entire class of expressions available in a typical CAS*[1]. However, detecting a polynomial for symmetry around an (unknown) point can be done by polynomial decomposition. Return to our earlier example (*), to identify a symmetry, we can use a polynomial decomposition procedure (e.g. `polydecomp` in Macsyma) to express a polynomial in a nested fashion, $f_1(f_2(\cdots))$. The command `polydecomp` expresses its results as a list $\{f_1, \ f_2, \cdots\}$ and in this particular case,

$$\left\{\frac{x^2+59}{12}, 6\,x^2+5, x-4\right\}.$$

Observe that the right-most linear term expresses the center of the symmetry, and the next term, consisting of only even powers of x namely $\{0, 2\}$, shows that it is symmetric in $x-4$. A polynomial which is symmetric about zero will end with an all-even-powered term. (Incidentally, `polydecomp` works for polynomials with arbitrary symbolic coefficients: $a * x^2 + b$ is symmetric about $x = 0$.)

What are the difficulties for the non-polynomial case?

- Simplification may not be clever enough. Thus for the Bessel function of the first kind, J, Macsyma 2.4 realizes that $J_3(x) + J_3(-x)$ is always zero. Mathematica 4.1 does not unless `FullSimplify` is called, but Mathematica 6.0 manages that simplification without extra commands.

- The power series command may be unable to derive a general form for the series expression. A far easier task (still not always possible in the presence of singularities that are non-obvious) for a CAS is computing an expansion around a point of an expression into a formal *truncated power series* or TPS. A TPS calculation can show there are only odd- or even-degree terms *up to some chosen degree*. It cannot show that all the (as yet uncomputed) non-zero terms have the same character. (For example, as a TPS, $x + x^3 + x^{200}$ will look antisymmetric around 0 until the expansion includes terms of at least degree 200.) Macsyma or Mathematica easily computes the terms to degree 3 of $1/(x+\sin(x))$ as $\frac{1}{2\,x} + \frac{x}{24} + \frac{x^3}{720} + \cdots$, suggesting it might be antisymmetric. Neither could compute "all" the terms as a symbolic power series (and it is somewhat unlikely that a solution exists that is shorter and neater than the command to compute it), so approximating the series by computing the truncated series to "high enough" order may be the best current approach.

2.1 More on other points of symmetry

Even if these tests worked infallibly at $x = 0$, they are not sufficient since they assume that we know the potential point of symmetry is at $x = 0$. How can we mechanize the recognition of a more general situation?

The prescription is again fairly obvious, but again the devil is in the details.

Solve the equation $S = f(x + c) + f(-x + c) = 0$ or $S = \infty$ for c.

How do our computer algebra systems fare with this task on our running example? Mathematica 4.1's `Solve` refuses to deal with our running example, and gives a message but no answer. However[2] `SolveAlways[TrigExpand[S]==0,{x,Sin[x],Cos[x]}]` gives some messages but also $c == 0$.

[1] Indeed, for that class there is not even an algorithmic test for equivalence to zero.

[2] Thanks to a reviewer of an earlier version of this paper who provided this suggestion. The reviewer also pointed out that conversion to exponential form, and a few forced transformations into a rational form in some set of variables is probably worth doing.

Macsyma 2.4's `solve` returns a simpler form, but it is not really solved. It says $c = -\sin(c)\cos(x)$, which is true and clearly has the one solution $c = 0$ which is expected. However, if $cos(x) > 1$, possible for imaginary x, there may be other solutions. Since we are looking for conditions on c *alone* and not dependent on x that will produce antisymmetry, the condition $c = 0$ is the only winner.

The command `Reduce` in Mathematica seems to have the specifications that would lead us to believe it would solve this problem, but the command in version 4.1, available when this paper was first drafted, appeared to work for polynomials only[3]. If it were to work, it might return the logical statement: Either $c = 0$ or $x = -\arccos(c/\sin(c)) \pm (2n+1)\pi$. One way[4] to convince Mathematica 4.1 to complete this computation is to convert the question to one about a polynomial in algebraically independent "variables" $\sin x$, $\cos x$, $\sin c$, $\cos c$ by using `TrigReduce`, and this indeed produces a logical statement somewhat more complicated that that given above.

If we are given a finite set of values for c, or a characterization for "all" c values we can attempt to simplify or expand in power series or TPS, as suggested above. An even nicer result can be obtained by conversion to complex exponentials.

Our conclusion here is that the center of *some* symmetries can be found with tools that are available, although not obviously so.

3 Symmetries via Rootfinding

Observe that

- $a(x) := p(x) - p(-x)$ is antisymmetric about $x = 0$. Proof: algebraically, $a(x) = -a(-x)$.
- $s(x) := p(x) + p(-x)$ is symmetric about $x = 0$. Proof: algebraically, $s(x) = s(-x)$.

Trivial consequences include

- $b(x) = x$ is antisymmetric – Proof: $b(x) := p(x) - p(-x)$ where $p(x) := x/2$. (This value for p is not unique. Add any constant to p.).
- $c(x) := x^2$ is symmetric: choose $p(x) := x^2/2$
- $c(x) := 0$ is both anti- and symmetric: choose $p(x) := 0$.

Sums of even-power monomials are symmetric at 0. Sums of odd-power monomials are antisymmetric at 0.

Non-trivial combinations of even and odd monomial polynomials are neither.

Consider symmetries around another point, c.

- $a(x) := p(x-c) - p(-(x-c))$ is antisymmetric about $x = c$
- $s(x) := p(x-c) + p(-(x-c))$ is symmetric about $x = c$ Proof: simple algebra, again.

While these statements hold more generally, we will for the moment be dealing with the case that $p(x)$ is a polynomial.

To find if either of these cases holds, we can try to find c and p. If we fail, we can conclude there is no symmetry. For antisymmetry we know that $a(c) = 0$. For symmetry, consider the derivative with respect to x of s at c. We require $s'(c) = 0$. In the polynomial case we know there are a finite number of real zeros, and thus a finite number of possible choices for c. We also know that at most one of these possibilities (anti- or symmetric) need be tested: If the degree d of the polynomial q under examination is even, test q for symmetry. If d is odd, test for antisymmetry.

[3]Mathematica 5.0 introduced a much improved `Reduce` program, but still inadequate at version 6.0 for this task.

[4]suggested by a reviewer

(Test for antisymmetry at a finite point) For each candidate c in the set of real roots of the polynomial equation $q = 0$ test to see if q is antisymmetric about c. If all candidates are exhausted and none claims "antisymmetric" then answer "not antisymmetric".

To test a particular candidate c, set up the hypothetical polynomial

$$p(x) := p_d x^d + ... + p_0$$

Symbolically express the equality we wish to enforce:

$$q(x) - (p(x-c) - p(-x+c)) = \sum_{0 \le i \le d} K_i x^i.$$

(by matching coefficients, we extract the definition for each K_i). If there is a solution to the linear system $\{K_i = 0\}$ for all the $\{p_i\}$, then we have found p and c, and we conclude q is antisymmetric around c. Otherwise the linear system will be found inconsistent.

The test for symmetry is similar, using as candidates the real roots of the derivative of q with respect to x.

Example: Test the odd polynomial q(x) below. See if

$$q(x) \;:=\; -\frac{1504\,x^5 - 3760\,x^4 + 1424\,x^3 + 1624\,x^2 - 2050\,x + 629}{32}$$

is antisymmetric about some point. Note that $q(x)$ has 3 real roots: $\{x = -0.85349,\ x = 0.5,\ x = 1.85349\}$. It can also be seen to have a factorization

$$-\frac{(2\,x-1)\,\left(752\,x^4 - 1504\,x^3 - 40\,x^2 + 792\,x - 629\right)}{32}$$

showing the exact real root at 1/2.

Running the algorithm described above at the different roots shows that only at $c = 1/2$ can we obtain consistent values for $p(x)$:

$$p(x) - \frac{47\,x^5 - 73\,x^3 - 24\,x - 2\,p_0}{2}$$

where we note that p_0 is arbitrary, but could easily be forced to zero at the beginning of the algorithm.

4 Rational Functions

We can extend the discussion to symmetry of rational functions N/D. First assume that N/D is in lowest terms, since this is done by default in most CAS (by polynomial GCD computation).

Next suppose that N is antisymmetric at c. Test the symmetry of D at c If D is symmetric about c then N/D is antisymmetric at c. D cannot also be antisymmetric about c because then N/D is not in lowest terms, having a common factor of $(x - c)$.

Alternatively, suppose that N is symmetric about some point c. If D is symmetric at c, so is N/D. If D is antisymmetric, so is N/D.

A reminder: the third (most likely) alternative is that N/D is neither.

5 Periodic functions are symmetric. Maybe antisymmetric too

Non-zero polynomials are not periodic, but some common functions are.

As suggested earlier, we can also try to simplify an expression in preparation for integration if we suspect that it is periodic. That is, use via similar use of solving programs we find the smallest positive period

p such that $f(x) = f(x+p)$. Blindly using a numerical integration program on $\int_0^{2000} \sin x \, dx$ has a high probability of inaccuracy (or failure) unless we notice something special about the integral. One possibility is that we observe we can compute the indefinite integral $(-\cos x)$ and thus the integral is symbolically $-\cos(2000)+1$. Another possibility is that we notice the periodicity with period 2π and that therefore the numerical integral can be broken up into a section of full periods plus the integral of the leftovers. In this case it is $[0, 636\pi]$ and $[636\pi, 2000]$. We have chosen 636π because it is about 1998.05. The first of these integrals is $318 \cdot \int_0^{2\pi} \sin x \, dx$ which is precisely 0. The second can be computed numerically as the integral over $[636\pi,\ 2000]$ or even better, $[0,\ 2000 - 636\pi]$ which is about 1.36746.

Solve programs today tend to find roots in different ways. The programs may return none, some or all. Sometimes the roots are unnecessarily repeated, or on the other hand, sometimes their multiplicities are concealed. If there are an infinite number of identical or different roots, the programs use some uncomfortable locution. Try solving the equation $\sin(x) = \sin(x+c)$ for c. You probably need to find the smallest non-zero period. Macsyma provides a list of two answers, one of which is $c = 2n_1\pi$ with n_1 some integer. Mathematica 6.0's `Reduce` provides a better answer than previous versions, the answer provided having c being a function of x:

$$c_1 \in Z \wedge \left(c = -x - \sin^{-1}(\sin(x)) + 2\pi c_1 + \pi \vee c = -x + \sin^{-1}(\sin(x)) + 2\pi c_1\right)$$

shows that c can be any real number if one is allowed to find some suitable x. What we want is a value of c good for *all* x.

We can turn this issue to our advantage by observing that any value of c that is satisfactory must work for any particular values of x, say $x = 0$. Then we can solve $\sin(0) = \sin(c)$ for c. Mathematica says $c = 0$, but warns that values are lost by solve. Macsyma says $c = n\pi$, which is more appropriate[5].

If we choose some other values for x such as $x = 45$, Macsyma produces an answer which includes $c = n\pi$:

$$\left\{c = 2\,\pi\,\%\mathrm{n3} + 2\,\arctan\left(\frac{\cos 45}{\sin 45}\right), c = 2\,\pi\,\%\mathrm{n2}\right\}.$$

Mathematica 4.1 gives $c = -14\pi$ in this case, with some warnings.

The principle is that one could pick some number of values of x for which the solve command succeeds and find the intersection of the solution sets; perhaps checking the result in the expression for symbolic x would be appropriate.

Unfortunately, computing the simplest form for the intersection of infinite sets represented generally as shown above is not easy, and the result from Mathematica is insufficient. One would hope that simple cases can be done either in a simple principled way, or at least in a simple enough *ad hoc* fashion that covers many of the common cases.

We explore some of these issues in a separate paper [2].

6 Another approach

Can we look at a rule-driven approach?

6.1 Symmetry, Anti-Symmetry, Shifts

Algebraic expressions are, in the small, symmetric, antisymmetric or neither.

Symmetric about zero: sums of monomials of even powers. Cosine.

Antisymmetric about zero: sums of monomials of odd powers. Sine.

$(x-1)^2$ is symmetric about 1. $x^4 + x^2$ is a sum of expressions symmetric about 0.

[5]Actually, Macsyma returns a newly generated name for n, starting with `%n1`. This can be a boon as well as a sticky point when combining expressions with arbitrary integer constants.

If $f,\ g$ are real-valued functions ($R \rightarrow R$) of a single real argument each symmetric around zero, their composition $f \cdot g$, their product, their sum are symmetric and real also. $1/f$ is symmetric. Constants are symmetric. Well known symmetric functions include cos. Exponential of a symmetric function is symmetric.

If $f,\ g$ are real-valued functions of a single real argument each antisymmetric around zero, their composition $f \cdot g$, and their sum are antisymmetric and real also. Their product is *symmetric.* $1/f$ is antisymmetric. Functions sin, tan and arctan are antisymmetric. Exp of an antisymmetric function is neither symmetric nor antisymmetric.

Shifts:

If $f(x)$ is (anti)symmetric around $x = 0$, then $f(x - c)$ is (anti)symmetric around $x = c$.

We expect that shift-free decomposition or difference calculus could work to find symmetries more generally (beyond polynomial cases).

However, there is a direct method, if we are good enough at solving equations[6]: Let T be the result of substituting $x + c$ for x in P. Let U be the result of substituting $-x$ for x in T. Solve $T = U$ for c. If a solution exists which is a constant (free of x), then P is symmetric around c. If $T = -U$ has a solution for c, it is antisymmetric. The typical solve program may take longer than necessary to find such a solution, since it may find the other irrelevant ones as well. An alternative is to factor $T - U$ and look for a factor $c - p$, identifying p as the point of symmetry.

Example. Consider the antisymmetric polynomial $x^3 - 16x$ which has zeros at 0 and ± 4. If we integrate numerically from -4 to $4 + 2^{-25}$, the integral should be the same as from 4 to $4 + 2^{-25}$, since the integral from -4 to 4, or for that matter any $-r$ to $-r$ will be zero. Mathematica 4.1 gives 2.13×10^{-14} for the first integral, but the correct answer is about 1.42×10^{-14}. This is in fact obtained from the second interval. To Mathematica's credit it warns of loss of precision in the first instance[7].

6.2 Neither symmetric nor antisymmetric

We need not give up entirely upon learning that an integrand fits neither of our categories: it may be a sum of symmetric and antisymmetric parts, in which case one can separate them and return the sum of their separate integrals. It is also plausible to find subexpressions which are symmetric about different points, and still make this separation. It is possible, in the case of a polynomial with high-order term x^{2n+1} to cancel out the contribution of this term by subtracting off $(x - c)^{2n+1}$ where c is the center of the interval of integration; the contribution of the added term to the integral is zero. This approach is reminiscent of "completing the square" in integration.

7 A Generalization and a Reduction

This issue of detection of symmetries was proposed as a term project to a graduate class at UC Berkeley, Spring, 2004 (CS 282: Algebraic Algorithms). Student Armando Solar-Lezama generalized the problem somewhat and formulated a method to resolve this generalized question to (partially) solved problems in equation solving. The issue remains as to how much of the various methods can be used automatically, when reduced to a program in a computer algebra system. [available on request]

Here is the idea, whose description draws extensively from Solar-Lezama's paper.

7.1 Symmetry

Consider symmetry as invariance under a group of transformations (For example see Mumford et. al. for an introduction to the subject [4]). Thus, given a transformation T that maps some domain R to itself, we

[6] If you sense that this is a big "if" you are not mistaken!

[7] Mathematica 5.1 does better on this integration and gets about 1.42×10^{-14} in both cases. But Mathematica 6.0 does worse, giving a warning message in the first case, returning "$0. \times 10^{-14}$.

say that $f: R \to R$ is symmetric under T if and only if $f = T[f]$. Analyzing the set of *all* possible T in full generality is neither practical nor particularly helpful. Instead, let us focus on transformations T that can be expressed in a particular, but still interesting, form. Given $f(x)$, define $T[f](x)$ as a combination of linear transforms around the function f as follows:

$$T_{a,b,c,d}[f](x) = a \times f(b \times x + c) + d;$$

Such a family of transformations is parameterized by 4 constants. For our current discussion you may assume they are real numbers $\{a, \ , b, \ c, \ d\}$, though this could be generalized. The family of transformations is fairly powerful; in particular, it allows us to express the forms previously mentioned plus other traditional notions of symmetry. Thus the notion that a function f is even simply means that it is invariant under $T_{1,-1,0,0}$. Similarly, the notion that a function is odd simply means that it is invariant under $T_{-1,-1,0,0}$. This family of transformations also allows us to express periodicity, since a function periodic with period p will be invariant under $T_{1,0,p,0}$.

In general, one of the advantages of understanding symmetries as invariance under transformations is that the resulting transformations have some very nice algebraic properties. For example, whenever one has a family of reversible transformations, this family will constitute a group under the operation of composition. Furthermore, for a given function f, the family of transformations under which f is invariant will constitute a sub-group of the original group. This fact is easy to see, because if f is symmetric under T_1 and T_2, then $T_1[T_2[f]] = T_1[f] = f$, and similarly $T_2[T_1[f]] = T_2[f] = f$. Also, $T^{-1}[f] = T^{-1}[T[f]] = f$, so the set of transformations under which f is symmetric is closed under composition and inverse. This is a very general fact, and is true not only for functions, but for any kind of symmetry. In the case of this family of transformations, however, there is a nice additional property that the operations of composition and inverse can be expressed in terms of the $\{a, b, c, d\}$ parameters.

In the discussion below we will use the shorthand notation $\langle a, b, c, d\rangle$ to refer to the transformation $T_{a,b,c,d}$.

In particular, given $T1 = \langle a_1, b_1, c_1, d_1\rangle$, $T2 = \langle a_2, b_2, c_2, d_2\rangle$ and $T = \langle a, b, c, d\rangle$ we have the following algebraic rules:

$$T1 \circ T2 = \langle a_1a_2, \ b_1b_2, \ b_1c_2 + c_1, \ a_1d_2 + d_1\rangle \tag{1}$$
$$T^{-1} = \langle 1/a, \ 1/b, \ -c/b, \ -d/a\rangle \tag{2}$$
$$T^n = \langle a^n, \ b^n, \ c \times \frac{(b^n - 1)}{b - 1}, \ d \times \frac{(a^n - 1)}{a - 1}\rangle \tag{3}$$

The two rules are easy to verify just from the definition of $T_{a,b,c,d}$. The third rule can be proven by induction on n, using the first equation, and is a generalization of rule (2) for $n = -1$. Note that the last formula works also for negative powers of n. If $b_1 = 1$ use the alternate form $\sum_{0 \le i < n} b_1^i$.

Now that we have defined a family of transformations and the algebraic rules needed to manipulate them, we can begin to talk about inferring symmetries for a function.

7.2 Inferring Symmetry

The strategy for finding symmetries in functions will be as follows. First, for any function f, we can define a set $S(f)$ of transformations such that

$$T \in S(f) \Leftrightarrow T[f] = f$$

As we saw before, $S(f)$ will be a group. Now the plan is to start with a set $S(f)$ for a set of basic functions, and to have a set of rules that allow you to find S for combinations of functions for which their S is known. Actually, even if we are generally successful, that that we will find S, but perhaps some $S' \subseteq S$. However, we will show that the algorithm is powerful enough to handle many interesting forms of symmetry.

We start with some set of basic functions, along with the definition of their S. (This set below is not minimal since for example $\cos x$ symmetries can be found from $\sin(x + \pi/2)$ symmetries. Adding tan is possible although its S can be derived.).

$$\begin{aligned}
S(C_0) &= \{\langle a,\ b,\ c,\ C_0 - aC_0\rangle \quad C_0 \text{ a constant}\} && (4)\\
S(x) &= \{\langle a,\ 1/a,\ c,\ -ac\rangle\} && (5)\\
S(\sin) &= \{\langle -1,\ 1, \pi,\ 0\rangle, \langle -1,\ -1,\ 0,\ 0\rangle\} && (6)\\
S(\cos) &= \{\langle -1,\ 1,\ \pi,\ 0\rangle, \langle 1,\ -1,\ 0,\ 0\rangle\} && (7)\\
S(\exp) &= \{\langle e^{-c},\ 1,\ c,\ 0\rangle\} && (8)\\
S(\log) &= \{\langle 1,\ b,\ 0,\ -\log(b)\rangle\} && (9)\\
S(x^n) &= \{\langle a^n,\ 1/a,\ 0,\ 0\rangle\} && (10)
\end{aligned}$$

Note that the representations given above for the sets S are not exhaustive. For example, $\langle 1,\ 1,\ 2\times\pi,\ 0\rangle$ does not appear in the symmetry representation of $S(\sin)$. However, this and all other symmetries for the above functions can be generated from the S sets given above. In the case of sine, for example, $\langle 1,\ 1,\ 2\pi,\ 0\rangle = \langle -1,\ 1,\ \pi,\ 0\rangle^2$ by the composition rule (3) given in the previous section, and in fact we could specify

$$S(\sin) = \{\langle -1^n,\ 1, n\pi,\ 0\rangle, \langle -1,\ -1,\ 0,\ 0\rangle\}.$$

This merely reflects that this (or any) function that has period π also has period $n\pi$ for positive integer n.

Now that we have the symmetries for some basic functions, we need a way to find symmetries for more complex functions built from the basic functions given above. To do this, we use the following composition rules. For all $\langle a_f, b_f, c_f, d_f\rangle \in S(f)$, and $\langle a_g, b_g, c_g, d_g\rangle \in S(g)$, the following apply

$$\begin{aligned}
\langle a_f, b_f, c_f, d_f + d_g\rangle &\in S(f+g) && \text{given} && a_f = a_g, b_f = b_g, c_f = c_g && (11)\\
\langle a_f \times a_g, b_f, c_f, 0\rangle &\in S(f\times g) && \text{given} && d_f, d_g = 0, b_f = b_g, c_f = c_g && (12)\\
\langle a_f, b_f, c_f, d_f \times d_g\rangle &\in S(f\times g) && \text{given} && a_g = 0 && (13)\\
\langle a_g, b_g, c_g, d_f \times d_g\rangle &\in S(f\times g) && \text{given} && a_f = 0 && (14)\\
\langle a_f/a_g, b_f, c_f, 0\rangle &\in S(f/g) && \text{given} && d_f, d_g = 0, b_f = b_g, c_f = c_g && (15)\\
\langle a_f, b_f, c_f, d_f/d_g\rangle &\in S(f/g) && \text{given} && a_g = 0 && (16)\\
\langle a_g, b_g, c_g, d_f/d_g\rangle &\in S(f/g) && \text{given} && a_f = 0 && (17)\\
\langle 1/a_f, b_g, c_g, -d_f/a_f\rangle &\in S(f\circ g) && \text{given} && b_f = a_g \text{ and } c_f = d_g && (18)
\end{aligned}$$

Using these rules, we can attempt to find a non-empty set S' for any function involving sums, products and divisions of our basic functions. The reason we cannot use these rules to find S, as opposed to S', is that they fail to take into account algebraic relationships that may hold between f and g. In particular, for every operation listed above (composition, multiplication, division and addition), there is a type of algebraic relationship that will cause *new* symmetries to appear even if f and g have no symmetries on their own. As the simplest example, if $g = -f$ (but presumably in a non-obvious way, so that it is not noticed), then $f + g$ will have all the symmetries of 0, regardless of the absence of symmetries in f.

In the case of addition, if f and g satisfy the relationship

$$f = T[g]$$

Where T satisfies the properties $T = T^-1$, and $T[f + g] = T[f] + T[g]$. If all of this is true, then $f = T[g]$ will imply

$$T^{-1}[f+g] = T^{-1}[T[g] + g] = g + T^{-1}[g] = g + f.$$

Now, it turns out that the only T that have this property are transformations of the form $\langle \pm 1, \pm 1, c, 0 \rangle$, where c is nonzero only if b is equal to one. Thus, functions that involve additions of functions satisfying this relationship will have symmetries that we will be unable to find.

Similarly, in the case of multiplication and division, if f and g satisfy the relationship $f = T[g]$, and T satisfies $T = T^{-}1$, and $T[f+g] = T[f] \times T[g]$, then $f \times g$ will be symmetric under T, but our algorithm will not know it.

Finally, in the case of composition, if $g = f^{-1}$, (but presumably in a non-obvious way so that it is not simplified away) then once again there may be symmetries that our system fails to catch.

The algorithm, described in more detail in a term paper [6], depends on using Mathematica's `Reduce` command to find out which of the requirements specified in the combination rules can be verified in an essentially syntactic decomposition of an expression.

While the principle is fairly clear, this tactic rephrases the symmetry problem in terms of the functionality of Mathematica's `Reduce` command. This functionality can, and has, changed in recent versions of Mathematica. Any new features of this program may benefit the symmetry detection. Any difficulties in `Reduce` can result in problems here. We have shown [6] that the simplification of the multiple constraints mandated by the combinations of the rules above to some minimal set from which useful properties may be extracted, is a challenge for computer algebra systems. Even fairly small expressions can lead to unnecessarily large constraint sets.

8 Conclusion

We have provided some suggestions for designers of computer algebra systems to get smarter with respect to symmetry, in general, and for specific tasks as well. In particular, we see that definite integration programs in computer algebra systems can take advantage of symmetry, sometimes removing the need to deal with poorly conditioned or even "impossible" integrals.

At least if one can come up with a symmetry test that is guaranteed to be reasonably fast, it should be done early in the processing of definite integrals, perhaps even numerical integrals. There is some interaction between our processing and testing for singularities in the range of integration (leading to Principal Value considerations) and thus one might claim that some integrals, including our running example, should be rejected. Using symmetry arguments exclusively might deprive you of the warning that could accompany the discovery of singularities, and one could argue that the system should warn of such answers.

Taking into account these and other issues is necessary for the design of a programmed "user command" for definite integration. There is a recent discussion of these issues for Mathematica by Lichtblau [5].

9 Acknowledgments

This research was supported in part by NSF grant CCR-9901933 administered through the Electronics Research Laboratory, University of California, Berkeley. Thanks to Daniel Lichtblau for useful conversations.

References

[1] R. Fateman and W. Kahan, "Improving Exact Integrals from Symbolic Computation Systems," Tech. Rept. Ctr. for Pure and Appl. Math. PAM 386, Univ. Calif. Berkeley. 1986. (also poster session at ISSAC 2000.)

[2] R. Fateman, "Infinite Indexed Sets in Computer Algebra Systems," draft.

[3] K.O. Geddes and G.J. Fee, "Hybrid symbolic-numeric integration in Maple." Proceedings of ISSAC'92, P.S. Wang (ed.), ACM Press, New York, 1992, pp. 36-41.

[4] Mumford.

[5] Daniel Lichtblau, "Issues in symbolic definite integration", Int'l Conf. on Applications of Computer Algebra (ACA 2003), Session on Symbolic Summation (July, 2003).

[6] A. Solar-Lezama, "Finding symmetries in Functions of One Variable" (term paper in CS282, Spring, 2004, UC Berkeley).