DRAFT: Comments on Infinite Indexed Sets in Computer Algebra Systems

Richard Fateman

October 18, 2006

1 Introduction

Computing with "sets" is well-explored in the programming-language and data-structure literature. Many languages have one or more set representations as well as operations for these sets. Unfortunately, the notion of set in mathematics is far more elaborate than the supported notion. The programming languages' notations and operations work for explicit finite sets only, not (for example), the set of all odd primes.

Representing and manipulating *all* kinds of sets and set descriptions, including infinite set notations, or cases where the set is described algebraically, geometrically, logically, functionally or using some other "implicit" method is an open-ended challenge.

In this paper we address a subset of the problem prompted by an application of a computer algebra system (CAS). We show how we can "automatically" provide some simplification of set descriptions and other operations. Our motivation comes from problems in which we obtained infinite sets such as $\{n\pi\}$ for integer n, and must compute with these further. These sets can be easily generated by a CAS asked to solve $\sin x = 0$ for x. To date, simplification of operations on such resulting sets are generally unavailable from CAS, making the immediate result only marginally suitable for human consumption ; worse, the result is almost useless to pass along to other automated manipulations.

2 Set notation

There are many common notations for designating sets. Large sections of various font alphabets are consumed by special conventions for naming sets in algebra and analysis. We do not intend to close off future extension or further discussions, but to keep our discussion finite, we must make some choices to simplify our discussion. We hope to navigate a territory that contains some useful non-trivial results but does not require us to solve the unsolvable.

2.1 Basic named sets used for constructing others

The set of natural numbers $\{0, 1, 2, 3, \cdots\}$, will be denoted $\mathbb{N}$. We use it in the *index set* designation of other sets.

An alternative index set, of all integers, $\{\cdots, -2, -1, 0, 1, 2, 3, \cdots\}$, will be denoted $\mathbb{Z}$. We will generally use identifiers n, m, k to mean variables restricted to $\mathbb{Z}$. Thus $n > -1$ means non-negative integers. (that is, $n > -1$ also means the same as $n \in \mathbb{N}$). Mathematica allows the following built-in names to be used for basic sets: `Algebraics, Booleans, Complexes, Integers, Primes, Rationals, Reals.` It would be a mistake to assume that the program effectively encodes any but the most trivial properties attached to these names.

2.2 Finite Lists

While we focus on infinite sets, we must also allow for the "easy" cases.

- The empty set $\{\}$, sometimes written $\emptyset$.
- Singletons: sets with only a single value. Arguably all results for a CAS command could be generalizable as a *set* of answers. The familiar case is the singleton set with a single value. For example $1+1 \rightarrow 2$ might be $1+1 \rightarrow \{2\}$. This allows for consistent treatment of $\sqrt{9} = \{3, -3\}$ or $1^{1/3} = 1,\ -1/2 + 1/2\, i\sqrt{3},\ -1/2 - 1/2\, i\sqrt{3}\}$.

  An astute observer might say that if $\sqrt{9}$ has two values, so does $\sqrt{3}$ in that last expression. So what is meant by that triple of expressions? In fact, in a purely algebraic setting we can choose either the popular positive interpretation approximated by 1.732... or its negation, as long as we use it consistently. In other settings we may need to distinguish (say) the positive root. If that set of roots is separated and the three values interpreted in different contexts (and with inconsistent signs for $\sqrt{3}$) we may not be so safe!
- Other explicit lists: all elements are provided: E.g. {a,b,c}.
- Multisets: sets including possible multiple occurrences of an element. E.g. $\{a, b, a, c\}$. (The kinds of applications of multisets that come to mind for computer algebra "analysis" are not very compelling. Consider the set of zeros of a polynomial. It is more likely that representing them as a multiset would not be as useful as a representation for a set of pairs (*value, multiplicity*), or two lists, of roots and multiplicities. Either would be easier for a program to handle.) We will generally try to ignore extra duplicated elements and will try not to distinguish $\{a, b, a, c\}$ from $\{a, b, c\}$.

2.3 Plus-minus Expressions

This notation (as input or output) is not commonly accepted in computer algebra systems, but is widely used in applied mathematics texts.

$-b/2 \pm \sqrt{d}$ is a set of two expressions.

$a \pm b \pm c$ is a set of 4 expressions, though $a \pm b \mp c$ is a set of 2 expressions.

2.4 Indexed Sets

The common notation is $\{f(n) \ \mid\ n \in \mathbb{N}\}$

A better notation would make clear the relationship between the two several occurrences of n. In particular, only when n occurs "free" on the left, it should be bound to the n on the right. (A non-free, i.e. bound n occurs in the integrand "$n^2 dn$" here: $\{n + \int_0^n n^2 dn \ \mid\ n \in \mathbb{N}\}$) and only the "external" n ranges over $\mathbb{N}$.

A clearer notation which uses explicit ideas is available rom the lambda calculus, which is simple and familiar to anyone who has learned Lisp or logic: We suggest something like $\{\lambda(n) f(n) \ \mid \mathbb{N}\}$ if appropriate to be more formal. We will not necessarily use this notation for the simple examples of this paper.

2.5 Compound Sets

Given any of the representations for sets A, B, C, it may be infeasible to simplify expressions like $A \cap B$ or $A \cup B$ or the complement $\bar{A}$. Our representation is too general to allow explicit computation of unions or intersections in all cases. The typical explicit sets provided in "set" packages are different in this respect.

Set simplification can use various identity laws, distribution, deMorgan's laws. While useful around the margins and dealing with left-over pieces of sets, we suspect that for our applications these are not going to be very effective.

3 Operations

Some of these may not be algorithmically computable *in general.* In most cases this does not prevent us from writing a program to perform these operations *sometimes*, and in other cases leave the results unresolved.

- Test for empty set.
- Choose an element. (and remove it from a set.) Generate the next element, etc. (This may impose an order on the set. This may be done in an indexed set by choosing the element with the least index, if that is possible.) A (generally not computable) alternative is to extract smallest element.
- Membership. Is $x \in S$? e.g. is $0 \in \{\lambda(n)\ 2 \times n \ \mid \mathbb{N}\}$?
- Intersection: Find an expression simpler than $A \cap B$ for an intersection. This is not possible in general, but it is helpful to be able to compute, for example that $\{\lambda(n)(2 \times n + 1) \ \mid \mathbb{N}\} \ \cap \ \{\lambda(n)2 \times n \ \mid \mathbb{N}\}$ can be expressed more simply, namely as the empty set $\emptyset$.
- Union: Find a simple expression than $A \cup B$ for a union. For $\{\lambda(n)2 \times (n+1) \ \mid \mathbb{N}\} \cup \{\lambda(n)2 \times n \ \mid \mathbb{N}\}$ we can use the set $\mathbb{N}$.
- Cardinality: the number of elements, which may be infinite. Unique cardinality (counting repeated elements once). For some descriptions the cardinality may not be computable, or we may not be clever enough to compute it. Sets defined by pipes typically fall in this category. It is not always possible to tell whether we have a setor a multiset (with repeated elements). When such a distinction is required in a given application, it would be an additional (and perhaps not computable) burden to assure that a set does not have repeated elements.
- Cross product: $A \times B$, the set of ordered pairs from sets A and B.
- Arithmetic: Arguably, $A + B$ could be the pairwise summing of elements from A and B, that is $\{a + b | a \in A, \quad b \in B\}$. If (say) A is not a set but a scalar quantity s, the sum is presumably $\{a + s | a \in A\}$. A conundrum follows, for $A \times B$, if A is the scalar 0, and B is the empty set. A commonly applied rule of thumb in algebra systems is that $0 \times$ *anything* is 0, and yet by this definition, we get not 0, but the empty set, violating our rule of thumb. A possible resolution is to forbid the use or generation of the empty set in arithmetic, or substitution of the empty set into expressions.
- Other operations may be added to this list.

4 Techniques

- Reduction to canonical index names. e.g. operations on sets should *not* be able to distinguish $\{\lambda(n)(2 \times n + 1) \ \mid \mathbb{N}\}$ from $\{\lambda(m)(2 \times m + 1) \ \mid \mathbb{N}\}$, each of which is the set of non-negative odd integers.
- Black box generators. In the sense of pipes or streams, consider a set which allows you to examine one of its elements and then, if necessary on can run a program generating the next element in the set. This may ordinarily require a pipe that generates elements in order. An implementation in Lisp might look like

  ```
  (define (integers-from n)(make-pipe n (integers-from (1+ n 1))))
  (define non-neg-integers (integers-from 0))
  ```

  From this can be created, using other programs working as filters or mapping functions, almost any computational series. For example, pipes that contain only primes are illustrated in a well-known computing text [1].

- Compare to well-known sets: $\mathbb{Z}$, $\mathbb{N}$. $\mathbb{Q}$, (rationals) $\emptyset = \{\}$ to see if a relationship (subset, equality?) holds.
- Simplifications include all operations (distributivity, deMorgan's laws) defined for sets generally in the same CAS.

5 Requirements, benchmarks

We would like to be able to consider solving the problems in this (growing) list.

1. $\{2n \mid n \in \mathbb{N}\} = \{2 * m \mid m \in \mathbb{N}\} = \{2 * k + 4 \mid k > -3\}$

   To notice this equality, the usual approach in a CAS is to find a *canonical form* to which each of these can be simplified. (See lambda-binding above).

2. $\{2n + 1 \mid n \in \mathbb{N}\} \cup \{2 * n \mid n \in \mathbb{N}\} = \mathbb{N}$

   Even and odd numbers are all there can be in $\mathbb{N}$.

3. $\{2n + 1 \mid n \in \mathbb{N}\} \cap \{2n \mid n \in \mathbb{N}\} = \{\}$

   Even and odd numbers don't overlap.

4. $\{n - 1 \mid n \geq 0\} \cap \{m \mid m \leq 0\} = \{-1,\ 0\}$

   Finite sets can result from intersections of infinite sets.

5. $\{2n \mid n \in \mathbb{N}\} \cup \{3m \mid m \in \mathbb{N}\} = \{n \mid ((n \bmod 3 = 0) \vee (n \bmod 2 = 0)) \ \wedge \ n \in \mathbb{N}\}$. This right-hand expression is probably not more attractive than the left-side expression. Arbitrary manipulation of the index set is clearly going to lead to problems.

   Let $P = \{n \mid \mathrm{prime}(n)\}$ and $R = \{(p + q) | (p \in P) \wedge (q \in P)\}$ If we can prove the set of even numbers greater than 4 is a subset of R we have proved Goldbach's conjecture[1].

6. $\{2n \mid n \in \mathbb{N}\} \cap \{3m \mid m >\in \mathbb{N}\} = \{n \mid n \bmod 6 = 0) \mid n \in \mathbb{N}\}$

   How can we deal with this?

   $\{n^2 \mid n \in \mathbb{N}\} = \{m^2 - 2m + 1 \mid m \in \mathbb{N}\}$ Each includes all integers that are squared.

6 Applications

6.1 Symmetry, periodicity in integration

Our initial motivation for this exploration was an application that requires detecting periodicity of a function. In this application [2], in the process of computing a definite integral, we first check for symmetry or antisymmetry of an integrand around some point (to be found). Our method depends on solving a related shifted equation whose solution can be an infinite set of the nature required above. Continuing the computation in a CAS unattended, that is, with the human taken *out of the loop*, requires further operations to be done sight unseen. In particular, the intersection of two sets, each an infinite set of points, might be finite (even empty) or infinite. An algorithmic approach is required, even if it is clear that all problems will not be solvable. Similar questions arise about set unions.

[1] Namely all positive even integers $n \geq 4$ can be expressed as the sum of two primes. A reward of $1,000,000 for such a proof was offered by a publisher (Faber and Faber), although it is now expired.

6.2 Other related problems

The notational and simplification issues here have parallels in dealing with other mathematical objects. For example, consider $\sum_{n=1}^{\infty} f(n) + \sum_{m=1}^{\infty} g(m)$. Simplification of this expression requires renaming of one (or both) of the index sets; the same lambda-binding idea holds. It requires aligning the index sets in $\sum_{n=1}^{\infty} f(n)x^n + \sum_{m=0}^{\infty} g(m)x^m$.

7 Plan of attack

As stated initially, we must make some judicious choices and fix some of the many variabilities.

We are not interested in discussing simplification of (other kinds of) sets that may already be included in a CAS. Thus we assume $A \cup A = A \cap A \rightarrow A$ and similar rules are known and applied when possible. Since our motivation was to be able to compute further with cases as returned from CAS solve programs, let us look at them again.

Consider the set we got from solving $\sin x = 0$ for x. The expression $\arcsin(0)$ is actually a set, $n\pi$ meaning $\{n\pi \mid n \in \mathbb{N}\}$[2]

Here is the kind of computations might we wish to do with this set:

What solutions does the equation $\sin(x) = 0$ have in common with $\sin(x/2) = 0$? The two solution sets are $\{n\pi \mid n \in \mathbb{N}\}$ and $\{2m\pi \mid m \in \mathbb{N}\}$. Their intersection is found by (a) assuring that the index range is identical, and (b) solving $n\pi = 2m\pi$ for (say) n, resulting in $n = 2m$. The intersection result is thus $\{2m\pi \mid m \in \mathbb{N}\}$. It is not appropriate to solve for m to get $m = n/2$ because m must be in $\mathbb{N}$, and half-integer values are not permitted. We must compute this explicitly, as shown below.

7.1 What is the union of the two sets?

This can be rather complicated since we are searching for a simplified form. Here is a heuristic that sometimes helps, given that A and B are simplified set notations. If $A \subseteq B$ then $A \cup B = B$. Constructively can we show $x \in A \;\Rightarrow\; x \in B$? In order to show that for each value $2m\pi$ there is a value $n\pi$, set them equal and solve for n. We must conclude $(n = 2m \;\wedge\; m \in \mathbb{N}) \Rightarrow n \in \mathbb{N}$ by noting that multiplication by 2 maps $\mathbb{N}$ into $\mathbb{N}$.

7.2 When are two sets equal?

Consider a problem that will turn out to be too hard. We encountered the question `solve(sin(x+c)=sin(x),c);` Macsyma says:

$$\left\{c = 2\,\arctan\left(\frac{\cos x}{\sin x}\right) + 2\,\pi\,\mathrm{n}_1, c = 2\,\pi\,\mathrm{n}_2\right\}.$$

This is kind of hard to fathom, so substituting $y - c/2$ for x and then expanding the sines of sums, we get:

$$2\,\sin\left(\frac{c}{2}\right)\,\cos y \;=\; 0$$

The first factor leads to solutions of the form $c = 2\arcsin 0$. The second factor leads to $x + c/2 = \arccos 0$ or $c = 2 \cdot (\arccos 0 - x)$. Combining these we can claim the solution is

$$\{n\pi\} \cup \{(2m+1)\pi - 2x\}.$$

[2]A possible quibble: if $n\pi$ is an angle, then can we say that—as angles—$0 = 2\pi$ in which case the set of unique solutions is the set of only two: $\{0,\ \pi\}$. We reject this because there are other ways of looking at arcsin where the angle continues to evolve indefinitely to higher numbers.

Here is another approach to the same question. Let us convert to exponential form first, then use the `radcan` simplification command. In this case Macsyma gives:

$$\{-2\,x+2\,\pi\,n+\pi\} \cup \{0\}.$$

Actually, that $c = 0$ solution is defective; it is claiming that $e^{ic} = 1$ has the (sole) solution $c = 0$ instead of $c = 2k\pi$. A corrected solution would then be

$$\{-2\,x+2\,\pi\,n+\pi\} \cup \{\ 2k\pi\}.$$

Mathematica says

$$\{-x + \arcsin(\sin(x))\}.$$

This expression is a piecewise-continuous function of x that is periodically constant. When $-\pi/2 \leq x \leq \pi/2$ then $c = 0$
When $\pi/2 \leq x \leq 3\pi/2$ then $c = -2x + \pi$
When $3\pi/2 \leq x \leq 5\pi/2$ then $c = -2\pi$. etc.

Maple finds only the solution is $c = 0$. True, but not very complete.

Comparing the above sets for all $x \in D$ (D might be real or complex numbers) is complicated, and not something we have programmed. In fact, the Mathematica set does not include solutions like $c = -2x + 5\pi$ which are provided in the Macsyma solution, so the sets are *not* equal.

The sets differ on a grosser level. The Macsyma solutions comprise an infinite set of lines with slope -2 spaced 2π apart. The Mathematica solution is single-valued (one member for each real x).

8 RAW NOTES

8.1 How to represent sets

We propose that a *basic set* is a pair: The first element is $\lambda(n)f(n)$. e.g. $\lambda(n)n^2$ for a set of squares. The second element is a domain for n, default is $n \in \mathbb{N}$. Other possibilities include $n > a$, $n < a$. The expression a must be integer-valued, although not necessarily an explicit integer. As will be evident shortly, we do not guarantee uniqueness of representation, and there may be sets that cannot be represented (conveniently) this way.

8.2 Operations

```
1. change of index, >  (lambda(n)f(n); n>a)
  if a is -1,  change to (lambda(n)f(n); n in N)
  if a is not -1, change  to  ((lambda(n)(lambda(q) f(q))(n+a)) n in N)
   ; q is not free in f.

2. change of index, <  (lambda(n)f(n); n<a)
  if a is 1,  change to (lambda(n)f(-n); n in N)
  if a is not 1,  change to ((lambda(n)(lambda(q) f(q))(-n+a)) n in N)
 ; q is not free in f.

(lambda(n)(lambda(q) f(q))(n+q)) can ordinarily be rewritten as (lambda(n)(f(n+q))).
 [assuming purely functional expression f.]

3. element test
   e is in ( lambda(n)f(n), n in N) if there is a solution n=k to
```

```
  f(n)=e and k is in N. Requires commands solve, and natnump.
Natnump(x):= if x is explicit natural number
             or x is declared to be a natural number
             or (x= y+z and natnump(y) and natnump(z))
             or (x= y*z and natnump(y) and natnump(z))
             or (x= y-z and natnump(y) and natnump(z))
             or (x= y^z and natnump(y) and natnump(z))

  may fail to identify some natnums.

fails on (n+1)*n/2 which is a natural number. either n or n+1
is even and thus the 2 always divides one or the other


heuristic:  for each indeterminate {v1,v2,..) substitute
a random integer. If the result is  an integer, suggest
that the expression  might be natnump. Test repeatedly to be more
confident..  haha.


4. subset test
  is  ( lambda(n)f(n), n in N)  a subset of
      ( lambda(m)g(m), m in N) ?

  solve the equation f(n)=g(m) for n=k
  test natnump(k) assuming m in N?

5. equality  A \subset B and B \subset A.

6. Intersection

   Let A=  ( lambda(n)f(n), n in N)
       B= ( lambda(m)g(m), m in N).

A \cap B is the set of all e such that there is an n in N
and an m in N such that
e=f(n)=g(m).

example,  A = set of n^2,  B = set of 2*m.    0,1,4,9,16,25,...  \cap 0,4,8,16, .. = 0,4,16,..

the common elements  can be found by setting
n^2=2*m
and solving the polynomial diophantine equation for n,m in N.

Since we cannot expect to solve arbitrary polynomial equations,
how about restricting the functions f, g, to linear functions
of n?

Example: what can we do with {2n+1}\cap{2n}   or {2n+5}\cap{2m}?
```

```
for x in intersection, x = 2n+1 for some n in \N
and also x = 2m for
some m in \N.

Thus $2n+1=2m$, and {\tt solve(2*n+1=2*m,m)} gives $m = (2n+1)/2$ which cannot be
shown to be a natnum by our previous method; substitution of random
integers also shows, at least some of the time,  it is {\em not} natnum.

{\tt solve(2*n+5=2*m,m)} gives $m=(2n+5)/2$, same argument.

How about $12kn$ and $20km$ intersection, where {\tt solve} gives $m=12/20n$,
certainly not a natnum. Caution is required here: assuming there is no solution
because we can't find it with this weak test leads to an incorrect
conclusion.

Find Least Common Multiple (LCM) of 60.
when n=0, m=0,    0
     n=5, m=3,   60
     n=10, m=6, 120
    etc

In general, if we want to find the intersection, for variables
n and m, with constants a,b,c,d
for a*n+b =  c*m+d  do this..

a*n=c*m+ (d-b)  ==> all converted to mod c.

 e.g. if d-b is negative, add multiples of c

so we need to find out if there are
solutions for n:   a*n mod c = d-b
.....
Example
         12kn+3       20km+1

when k=1, n=0,1,...  gives 3, 15, 27, 40, 51, 63, 75, 87, 99, 111, 123
          m=0,1,...  gives 1, 21, 41, 61, 81, 101, 121, 141, 161,181,

The question then is to solve for n: (12n+3) mod 20 = 1,
or rewrite as 12n mod 20 = 18

Mathematica allows this:

Solve[{12n ==18,Modulus==20},n,Mode-->Modular]
%though it gives the wrong answer in version 4.1 and probably 5.
%Solutions for linear diophantine equations can be programmed
```

References

[1] H. Abelson and G. Sussman, *Structure and Interpretation of Computer Programs,* MIT Press/ McGraw Hill.

[2] R. Fateman "Methods for integration of (anti)-symmetric functions" Draft.

[3] 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. (poster session at ISSAC 2000.)