

What might “understand a function” mean?

James H. Davenport
Department of Computer Science
University of Bath
Bath BA2 7AY England
J.H.Davenport@bath.ac.uk

June 27, 2007

The claim is often made (these days generally informally) that a given computer algebra system “understands” \sin , or some other function. Here we ask three questions.

1. What does this mean?
2. What might it mean?
3. How should a system “understand” a new function?

Notation: throughout this paper, the term ‘function’ means a function from \mathbf{R} to \mathbf{R} or \mathbf{C} to \mathbf{C} . The principles apply to functions \mathbf{R}^n to \mathbf{R} or \mathbf{C}^n to \mathbf{C} , but we shall not consider such functions here. C will denote an arbitrary field of constants (of characteristic zero), and x will be the variable of differentiation/integration.

θ is said to be elementary over a differential field K if one of the following is true:

(a) θ is algebraic over K ;

(b) $\theta' = \eta'/\eta$ for some $\eta \in K$ (we write $\theta = \log \eta$);

(c) $\theta' = \eta'\theta$ for some $\eta \in K$ (we write $\theta = \exp \eta$).

The object f is said to be elementary over K if it can be expressed in some $K(\theta_1, \dots, \theta_n)$ with each θ_i elementary over $K(\theta_i \dots \theta_{i-1})$. If K is omitted, $C(x)$ is assumed.

θ is said to be Liouvillian over a differential field K if one of the following is true:

- (a) θ is algebraic over K ;
- (b) $\theta' = \eta$ for some $\eta \in K$ (we write $\theta = \int \eta$);
- (c) $\theta' = \eta'\theta$ for some $\eta \in K$ (we write $\theta = \exp \eta$).

The object f is said to be Liouvillian over K if it can be expressed in some $K(\theta_1, \dots, \theta_n)$ with each θ_i Liouvillian over $K(\theta_i \dots \theta_{i-1})$. If K is omitted, $C(x)$ is assumed.

So far
no functions!

Let K be a field of functions in $\mathbf{R} \rightarrow \mathbf{R}$ (or $\mathbf{C} \rightarrow \mathbf{C}$). $f(x)$, a function from $\mathbf{R} \rightarrow \mathbf{R}$ (or $\mathbf{C} \rightarrow \mathbf{C}$) is said to be an *elementary* (resp. *Liouvillian*) function if it lies in some elementary (resp. Liouvillian) extension $K(\theta_1, \dots, \theta_n)$ of K .

However, even this is not enough.

Let K be a field of functions in $\mathbf{R} \rightarrow \mathbf{R}$ (or $\mathbf{C} \rightarrow \mathbf{C}$). $f(x)$, a function from $\mathbf{R} \rightarrow \mathbf{R}$ (or $\mathbf{C} \rightarrow \mathbf{C}$) is said to be a *proper elementary* (resp. *Liouvillian*) function if it lies in some elementary (resp. Liouvillian) extension $K(\theta_1, \dots, \theta_n)$ of K , where each θ_i is proper elementary (resp. Liouvillian) over $K(\theta_1, \dots, \theta_{i-1})$, and, for each x where both are defined,

$$(f')(x) = \lim_{\epsilon \rightarrow 0} \frac{f(x + \epsilon) - f(x)}{\epsilon}. \quad (1)$$

Furthermore, we require that the right-hand side of (1) be defined almost everywhere.

As examples of the various pathologies that can occur, we give the following examples, where K is the field $\mathbb{Q}(x)$ of rational functions $\mathbb{C} \rightarrow \mathbb{C}$ equipped with the derivation induced by $x' = 1$.

1. $K(\theta)$ where $\theta' = \frac{1}{x}$. Here θ is merely an abstract symbol, not a function at all.
2. $K(\theta)$ where $\theta' = \frac{1}{x}$ and $\theta : x \mapsto 0$. Here θ is elementary, and a function, but not a proper elementary function since equation (1) is violated.

3. $K(\theta)$ where $\theta' = \frac{1}{x}$ and $\theta : x \mapsto \begin{cases} 1 & x \in \mathbf{Q} \\ 0 & x \notin \mathbf{Q} \end{cases}$.

Here equation (1) is satisfied, but this falls foul of the last clause.

4. $K(\theta = \log(x) + 42)$ where $\theta' = \frac{1}{x}$. This is a proper elementary function, even though it is not “what we all mean by” $\log x$.

5. $K(\theta)$ where $\theta' = \frac{1}{x}$ and $\theta : x \mapsto \begin{cases} 0 & x > 0 \\ -i\pi & x < 0 \end{cases}$. As a function $\mathbf{R} \rightarrow \mathbf{R}$ this is $\log|x|$, and is a proper elementary function in our sense. Whether it is “what we all mean by” $\log x$ has been debated elsewhere.

What does it mean?

1. Numerical evaluation. Generally speaking, if the input is real, this means real evaluation where possible. To do numerical evaluation, one has to choose the branch cuts (if there are any) of the relevant function.
 2. Plotting — generally a consequence of the above, though more can in fact be done if the function is better “understood”.
- * So far, no algebra!

3. Differentiation. This property is generally hard-coded for some functions, with an extension mechanism for others, e.g. defining `diff/f` for a function `f` in Maple, or giving a symbol `a` a `!*DF` property in REDUCE.
4. Integration. This is the difficult one.
 - * This slide, no values, only algebra!

5. Special values. This is *not the same* as numerical evaluation: $\sin(\pi) = 0$, whereas $\sin(3.14159265358\dots26433) = 8.327 \times 10^{-26}$.

This is a case where the precise nature, and the adherence, of the branch cuts is critical: $\log(-1.0 + \epsilon i)$ might be near either of πi or $-\pi i$, but $\log(-1)$ *has* to be πi .

6. Simplification. Some of this is built in, e.g. for even/odd functions: other simplifications can be invoked via commands such as `expand` or `collect`, or by giving functions properties.

* Branch cuts!

So what do we need to know?

1. The algebraic properties — these are easy (**post Moses/Risch**);
2. Implicit, functional, knowledge of the branch cuts — even a Fortran programmer can do this;
3. Explicit knowledge of the branch cuts — so far not treated in any algebra system to my knowledge.

How is a function defined?

- * Hacked together by some guy(s)!
- 1. By explicit formulae, normally composition.
- 2. By indefinite integration.
- 3. By first order linear differential equations.
- 4. By higher order linear differential equations.
- 5. By functional equations.
- 6. By non-linear differential equations.
- 7. by definite integrals.

- By explicit formulae, normally composition.

! $\log \log \sin x : \mathbf{R} \rightarrow \mathbf{R}$ is in fact the ‘undefined’ function.

$$h(x) = \begin{cases} f(g(x)) & g(x) \text{ well-defined} \\ \lim_{y \rightarrow x} f(g(y)) & \text{otherwise.} \end{cases} \quad (2)$$

Expecting a system to perform (2) automatically is, in the author’s opinion, expecting too much.

By indefinite integration

1. This gives us evaluation **subject to not introducing a branch cut.**
2. Risch–Bronstein gives a test for ‘novelty’.
3. Integration is hard.
4. Special values are *definite* integration — hard.
5. Further Integration is hard — see paper.
6. Even/Odd simplification OK — rest harder.

By first-order l.o.d.e.

In general, one would consider a y defined by

$$y' + fy = g. \quad (3)$$

Let $F = \int f$ and $y = z \exp(-F)$. Then (3) becomes

$$z' \exp(-F) - fz \exp(-F) + fz \exp(-F) = g, \quad (4)$$

i.e. $z' = g \exp(F)$. Hence

$$y = \exp(-F) \int (g \exp(F)), \quad (5)$$

and the problem is reduced to the previous one.

By higher-order l.o.d.e.

- Even absolute ‘novelty’ is hard: the theory just about generalises to second-order equations.
- Relative novelty is a problem. J_ν are solutions of

$$x^2 y'' + xy' + (x^2 - \nu^2)y = 0, \quad (6)$$

whereas j_n is defined as solutions of

$$x^2 y'' + 2xy' + (x^2 - n(n+1))y = 0. \quad (7)$$

! No known general solution.

Function/Non-linear equations

For example, $ye^y = x$, whose solution is the Lambert W function. Not Liouvillian, but can also be defined by a non-linear differential equation: $W'(x) = \frac{W(x)}{(1+W(x))x}$.

Just as \log has infinitely many variants, separated by $2k\pi i$, which can be chosen to have a common branch cut, so W has infinitely many branches, but the description is somewhat hairier.

By definite integration

The classic example of this is the Γ function, defined by

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt. \quad (8)$$

This is continuous over the whole of the complex plane, except for $z = 0, -1, -2, \dots$

Many “simplification rules” are erroneous in places.

Conclusion

There has been comparatively little systematic work in this area: an early attempt was Moses, but little has been done in this direction. Perhaps the most interesting is MeunierSalvy.

To define functions completely, one has to know the branch cuts and their behaviour, and nothing has been done about automating this — largely because there is no consistent philosophy here. Indeed, it would be a significant step forward to have a system capable of checking that a definition of, say, a proper Liouvillian function and its branch cuts *was* consistent.