Towards Specifying Symbolic Computation*

Jacques Carette and William M. Farmer

Computing and Software, McMaster University, Canada
http://www.cas.mcmaster.ca/~carette
http://imps.mcmaster.ca/wmfarmer

Abstract. ??

1 Introduction

2 Simplifying Expressions denoting Rational Functions

2.1 The Problem

Let $f = \lambda\, x : \mathbb{Q} \,.\, R(x)$ be an expression that denotes a function of type $\mathbb{Q} \to \mathbb{Q}$. Furthermore, let $R(x)$, as a syntactic expression, denote a member of the field $\mathbb{Q}(x)$ consisting of *rational expressions* of the form $\frac{P(x)}{Q(x)}$ where $P(x)$ and $Q(x)$ are polynomials in x. A function that can be represented by an expression like f whose body is a rational expression is called a *rational function*.

$R(x)$ could be a very complicated expression build from the primitive components of the field $\mathbb{Q}(x)$: $0, 1, +, *, -, ^{-1}$. An obvious way to simplify f would be to simplify $R(x)$ syntactically as a rational expression in a meaning-preserving way. This operation, which we will call simpRatFun, is an example of a *syntax-based mathematical algorithm (SBMA)*. As an SBMA, simpRatFun works by manipulating syntactic expressions in a mathematically meaningful way.

SBMAs can be difficult to specify since they involve an interplay of syntax and semantics. We are interested in the following three questions:

1. What should be the specification of simpRatFun?
2. Can this specification be expressed in traditional logic?
3. How would this specification be expressed in a logic with undefinedness, quotation, and evaluation?

2.2 A Naive Specification

Let $R(x)$ be an expression that denotes a member of $\mathbb{Q}(x)$. The *normal form* of $R(x)$ is an expression $R'(x)$ of the form $\frac{P(x)}{Q(x)}$ such that $P(x)$ and $Q(x)$ are

* This research is supported by NSERC.

polynomials in standard form, $\frac{P(x)}{Q(x)}$ is in lowest terms, and $R(x)$ and $R'(x)$ both denote the same member of $\mathbb{Q}(x)$. For example, the normal form of

$$\frac{x^2+2x-1}{x^2-1} - \frac{x}{x-1}$$

is

$$\frac{1}{x+1}.$$

If we do not think too hard, we might be tempted to specify $\mathsf{simpRatFun}$ as follows:

> For all expressions $f = \lambda\, x : \mathbb{Q} \,.\, R(x)$ of type $\mathbb{Q} \rightarrow \mathbb{Q}$ where the expression $R(x)$ denotes a member of $\mathbb{Q}(x)$, $\mathsf{simpRatFun}(f) = \lambda\, x : \mathbb{Q} \,.\, R'(x)$ where $R'(x)$ is the normal form of $R(x)$.

Hence $\mathsf{simpRatFun}$ applied to

$$\lambda\, x : \mathbb{Q} \,.\, \frac{x^2+2x-1}{x^2-1} - \frac{x}{x-1}$$

should return

$$\lambda\, x : \mathbb{Q} \,.\, \frac{1}{x+1}.$$

This specification is essentially the same as Maple's normal operation that reduces the expression $\frac{x^2-2x-1}{x^2-1} - \frac{x}{x-1}$ to $\frac{1}{x+1}$.

Is this specification of $\mathsf{simpRatFun}$ correct? If so, f and $\mathsf{simpRatFun}(f)$ should denote the same function of type $\mathbb{Q} \rightarrow \mathbb{Q}$ for all expressions f that denote rational functions. Unfortunately, this is not the case. Let

$$f = \lambda\, x : \mathbb{Q} \,.\, \frac{x^2+2x-1}{x^2-1} - \frac{x}{x-1}.$$

Then $f(1)$ is obviously undefined, but $\mathsf{simpRatFun}(f)(1) = 1/2$. Hence $\mathsf{simpRatFun}$ is — as specified — not meaning preserving. What went wrong?

2.3 A Correct Specification

2.4 A Formalized Specification

3 Rational Expressions, Rational Functions

3.1 The Problem

Let e be an expression in the language $\mathcal{L}$ of the field $\mathbb{Q}(x)$, that is, a well-formed expression built from the symbols $x, 0, 1, +, *, -, ^{-1}$, elements of $\mathbb{Q}$ and parentheses (as necessary). For greater readability, we will take the liberty of using fractional notation for $^{-1}$ and the exponential notation x^n for $x * \cdots * x$ (n times). e can be something simple like $\frac{x^4-1}{x^2-1}$ or something more complicated like

$$\frac{\frac{1-x}{3/2x^{18}+x+17}}{\frac{1}{9834*x^{19393874}-1/5}} + 3 * x - \frac{12}{x}.$$

The meaning of e, written $[\![e]\!]_{\mathbb{Q}(x)}$, is the element of $\mathbb{Q}(x)$ that e denotes. If e is an expression like $x/0$, then the meaning of e is undefined and e is said to be *undefined*. We will call members of $\mathcal{L}$ *rational expressions*. Two rational expressions e and e' are *equal*, written $e = e'$, if they are both defined and $[\![e]\!]_{\mathbb{Q}(x)} = [\![e']\!]_{\mathbb{Q}(x)}$ and are *quasi-equal*, written $e \simeq e'$, if either $e = e'$ or they are both undefined.

We are taught that, like for members of $\mathbb{Q}$ (such as 5/15), there is a *normal form* for rational expressions. This is typically defined to be an expression p/q for two polynomials $p, q \in \mathbb{Q}[x] \subseteq \mathbb{Q}(x)$ such that p and q are themselves in polynomial normal form and $\mathsf{gcd}\,(p, q) = 1$. The motivation for the latter property is that we usually want to write $\frac{x^4-1}{x^2-1}$ as $x^2 + 1$ just as we usually want to write 5/15 as 1/3. Thus, the normal forms of $\frac{x^4-1}{x^2-1}$ and $\frac{x}{x}$ are $x^2 + 1$ and 1, respectively. This definition of normal form is based on the characteristic that the elements of the *field of fractions* of a ring R can be written as quotients r/s of elements of R where $r_0/s_0 = r_1/s_1$ if and only if $r_0 * s_1 = r_1 * s_0$ in R.

Every computer algebra system implements a function that *normalizes* rational expressions that denote elements of $\mathbb{Q}$, $\mathbb{Q}[x]$, and $\mathbb{Q}(x)$. Let $\mathsf{normRatExpr}$ be the name of an algorithm that implements this function. It should certainly have the signature $\mathsf{normRatExpr} : \mathcal{L} \to \mathcal{L}$ and satisfy the invariant $\mathsf{normRatExpr}(e) \simeq e$ — in other words, be *meaning preserving* — for all $e \in \mathcal{L}$. $\mathsf{normRatExpr}$ is an example of an SBMA. Note that merely giving the signature of $\mathsf{normRatExpr}$ and saying that it is meaning preserving is a (tremendously) incomplete specification of the *computational behavior* of $\mathsf{normRatExpr}$.

SBMA should be defined in the Introduction.

A rational expression e can be interpreted as a *function* $f = \lambda\, x : \mathbb{Q}\ .\ e$. Such functions are typically called *rational functions*. One might think that the rational functions $\lambda\, x : \mathbb{Q}\ .\ x/x$ and $\lambda\, x : \mathbb{Q}\ .\ 1$ should be equal since x/x and 1 are equal as rational expressions. But they are not equal since $\lambda\, x : \mathbb{Q}\ .\ x/x$ is undefined at 0, while $\lambda\, x : \mathbb{Q}\ .\ 1$ is defined everywhere. Hence, if a rational expression e is interpreted as a function, then it is not valid to simplify the function by applying $\mathsf{normRatExpr}$ to e, but CASs let the user do exactly this because usually there is no distinction made between e as a rational expression and e as representing a rational function. We need to know the *mathematical meaning* of $\mathsf{normRatExpr}$ applied to rational functions to be able to avoid unsound applications of $\mathsf{normRatExpr}$.

I don't know why we need to say this: Of course, given some symbol y, $f(y)$ **is** in $\mathcal{L}$.

We are thus interested in the following questions:

1. What should the specification of the computational behavior of normRatExpr be?
2. What should the the mathematical meaning of normRatExpr be when normRatExpr is applied to the body of a rational function?
3. What features of a logic are needed to express normRatExpr's specification and mathematical meaning?
4. What features of a logic would make expressing normRatExpr's specification and mathematical meaning relatively straightforward?

3.2 The Specification of normRatExpr

3.3 The Mathematical Meaning of normRatExpr

4 Conclusion

Todo list