J. Symbolic Computation (1998) **26**, 607–619
Article No. sy980229

A Normal Form for Function Rings of Piecewise Functions

MARTIN VON MOHRENSCHILDT†

Computing and Software, McMaster University, Hamilton, Canada

Computer algebra systems often have to deal with piecewise continuous functions. These are, for example, the absolute value function, signum, piecewise defined functions but also functions that are the supremum or infimum of two functions. We present a new algebraic approach to these types of problems. This paper presents a normal form for a function ring containing piecewise polynomial functions of a real variable. We give a complete rule system to compute the normal form of an expression. The main result is that this normal form can be used to decide extensional equality of two piecewise functions. Also we define supremum and infimum for piecewise functions; in fact, we show that the function ring forms a lattice. Additionally, a method to solve equalities and inequalities in this function ring is presented. Finally, we give a "user interface" to the algebraic representation of the piecewise functions.

© 1998 Academic Press

1. Introduction

One of the fundamental problems in computer algebra is to decide equality of two functions defined by expressions. Many algorithms have been developed, using algebraic and even statistical (Gonnet, 1985) arguments in order to decide or test the equality of expressions representing continuous functions. On the other hand, we often have to compute with piecewise continuous functions, e.g. in control theory and often in engineering applications. Piecewise defined functions are functions consisting of conditions and corresponding functions; the value at a point is the value of the function where the corresponding condition is true. The problem of deciding equality of piecewise defined function is harder in the following sense than that of deciding equality of continuous functions. In contrast to continuous functions, even if two piecewise defined functions are equal on a dense set in a given interval, they are not necessarily the same; they could differ at only one point. In order to decide equality of piecewise defined functions we propose to use normal forms. Normal forms for expressions are one of the most fundamental and successful concepts in symbolic computation. A normal form allows us to decide equality of functions represented by expressions by defining a unique representation for expressions defining the same function. In this paper we define a normal form for function rings of piecewise continuous function which are closed under composition.

We define the function ring PPFR (Piecewise Polynomial Function Ring), a polynomial function ring of one real variable extended with the function symbol step, step(x) is 1

†E-mail: mohrens@mcmaster.ca

0747–7171/98/110607 + 13 $30.00/0

© 1998 Academic Press

for $x > 0$ and 0 for $x \leq 0$. This step function enables us to express piecewise functions foe example functions build using `if-then-else` construct. The expression `if` $x > 0$ `then` $f(x)$ `else` $g(x)$ can be translated into an equivalent PPFR expression using step: $g(x) + \text{step}(x)(f(x) - g(x))$.

The main result of this work is that we develop an algorithm to decide extensional equality of the PPFR expression using a normal form. The normal form contains no products or powers of step and is non-nested, no step has a step as an argument. We present a complete rule system to transform a PPFR expression into its normal form.

Using step we are able to express the supremum and the infimum of two functions. We will prove that PPFR is a lattice: sup and inf of two functions of PPFR belong to PPFR again. Also, we are able to solve equalities and inequalities in this function ring. To solve inequalities we profit from the lattice structure: the inequality $f(x) > 0$ translates to the equality $\text{step}(f(x)) = 1$ having the same solutions.

A "weak" normal form, a normal form without the equality property, for function fields containing a conditional operator can be found in Aberer (1990). For differential equations containing piecewise functions see Filippov (1988) and von Mohrenschildt (1994). In the latter different forms of Heaviside kind of functions are used to define solutions of differential equations having "piecewise continuous" coefficients. PPFR is implemented in the computer algebra system Maple and it is part of it since version 4.

The paper is outlined as follows. First we define the ring PPFR. In the next section we define the normal form and give a rule system to compute the normal form. Being equipped with the normal form, we study the lattice structure of PPFR and discuss algorithms to solve equations and inequalities. Last we show how to convert a function in piecewise notation to a step expression and vice versa. We close with some examples of computations with piecewise functions.

2. Function Ring of Piecewise Functions

We build PPFR, Piecewise Polynomial Function Ring, over the term algebra containing the real algebraic numbers $\mathcal{RA}$, the symbol x, the binary operations $+, -, \times, \circ$ and the unary function symbol step.

Examples of terms of PPFR are:

$$\text{step}(x),\ x + \text{step}(x - \text{step}(x)),\ \text{step}(x-2)\,\text{step}(x^2 - 9).^{\dagger}$$

We restrict ourselves to real algebraic numbers $\mathcal{RA}$, solutions of polynomials with integer coefficients. The real algebraic numbers form a field and are computable. Further, the sign of real algebraic numbers is decidable, this is essential to be able to compute the value of $\text{step}(a)$ and also needed in the presented rule system and algorithms. Even the complexity of computing with real algebraic numbers can, in the worse case, be high (Kung, 1973; Rump, 1976; Roy, 1990), most computer algebra systems represent them in normalized form and provide built in methods to perform the needed computations with real algebraic numbers efficiently. The presented theory can easily be extended to any computable, algebraically closed, ordered real field of constants, e.g. the elementary numbers defined in Richardson (1992).

We define an interpretation of the terms as functions of the real algebraic numbers

[†]As usual we write $2\ x$ for $2 \times x$ and x^n for $\underbrace{x \times x \times x \times \cdots \times x}_{n}$.

$\mathcal{RA} \to \mathcal{RA}$ by defining the function symbols. The binary operations $+, -, \times$ have the usual meaning being addition, subtraction, and multiplication of the real numbers. The function symbol step is defined by

$$\text{step}(a) = \begin{cases} 0 & a \in \mathcal{RA} \text{ and } a \leq 0 \\ 1 & a \in \mathcal{RA} \text{ and } a > 0, \end{cases}$$

$\circ$ represents the composition of terms and is defined by:

Definition 2.1. (Composition) Let f, g, h be PPFR terms and a be a real algebraic number.

$$\begin{aligned} a \circ h &\to a \quad a \in \mathcal{RA} \\ x \circ h &\to h \quad x \text{ is identity of composition} \\ (f + g) \circ h &\to f \circ h + g \circ h \\ &\quad \text{same for } - \text{ and } \times \\ \text{step}(f) \circ h &\to \text{step}(f \circ h) \\ (f \circ g) \circ h &\to f \circ (g \circ h) \end{aligned}$$

We do not show that the rules given in 2.1 are well defined because this definition is common.

Definition 2.2. (Evaluation, Function) A term f of PPFR is evaluated at the point p, $p \in \mathcal{RA}$, by computing the composition of f and p, $f \circ p$ resulting in a real algebraic number.

A term f of PPFR is interpreted as a real valued function by

$$f : \mathcal{RA} \to \mathcal{RA} \quad f(p) := f \circ p \; p \in \mathcal{RA}.$$

For example, $(x \;\; \text{step}(x)) \circ 2 = (x \circ 2)\,\text{step}(x \circ 2) = 2 \times 1 = 2$.

We conclude that by construction of:

Corollary 2.3. *PPFR forms a function ring and is closed under composition.*

From now on, for most of the time, we use the functional notation, e.g. $f(2)$, $\text{step}(x+2)$ instead of $f \circ 2$, $\text{step} \circ (x + 2)$.

Note that the piecewise functions do not form a field, they contain zero divisors. But in the ring of the piecewise functions we allow division by a number $a \neq 0$.

3. The Normal Form

Expressions of PPFR represent functions of the real algebraic numbers. Clearly there exist several terms defining the same function. For example, $x \;\; \text{step}(x) - x \;\; \text{step}(-x)$ and $-x + 2x \;\; \text{step}(x)$ have the same extension, the absolute value function. Our main interest is to decide if two terms represent the same function.

Definition 3.1. Two functions defined by PPFR terms f, g are extensionally equal if

$$f \simeq g \;\; \text{iff} \;\; \forall p \in \mathcal{RA} \;\; f \circ p = g \circ p.$$

Next we define the normal form, which enables us to decide extensional equality. We observe that step satisfies several extensional equalities:

1. as step is a 0-1 valued function $\mathrm{step}(f)$ to any power reduces to $\mathrm{step}(f)$, e.g. $\mathrm{step}(f)^2 \simeq \mathrm{step}(f)$;
2. step is idempotent: $\mathrm{step}(\mathrm{step}(x)) \simeq \mathrm{step}(x)$;
3. a product of two steps can be represented as a sum of steps. $\mathrm{step}(x+2)\,\mathrm{step}(-x+1) \simeq \mathrm{step}(x+2)-1+\mathrm{step}(-x+1)$ and $\mathrm{step}(x-2)\ \mathrm{step}(x-3) \simeq \mathrm{step}(x-3)$

These properties enable us to define our normal form. The idea is the following: any product of steps can be reduced to a sum of steps, hence the normal form contains no product of steps. Using the property that step is a 0–1 valued function we can get rid of nested steps.

Definition 3.2. (The PPFR Normal Form) A term f of PPFR is in normal form iff

$$f = f_0 + \sum_{i=1}^{N} f_i \,\mathrm{step}(x - a_i) + \sum_{i=1}^{M} h_i \,\mathrm{step}(-x + b_i)$$

f_i and h_i are step-free terms, that is polynomials, and the steps are collected: $a_i = a_j \rightarrow f_i = f_j$ same for b_i, $h_i(a_i) \neq 0$, and $a_i, b_i, \in \mathcal{RA}$.

Rules for PPFR:
f is a step-free term, g is a PPFR term and $a, b \in \mathcal{RA}$

$$\mathrm{step}(ax - b) \rightarrow \mathrm{step}\left(\frac{ax-b}{|a|}\right) \quad \text{if } a \neq -1, 1, 0 \text{ division of constants} \tag{3.1}$$

$$f(x)\,\mathrm{step}(-x + a) \rightarrow f(x)(1 - \mathrm{step}(x - a)) \text{ if } f(a) = 0, \tag{3.2}$$

$$\mathrm{step}(x - a)\,\mathrm{step}(x - b) \rightarrow \mathrm{step}(x - a) \text{ if } a \geq b \quad \text{else} \quad \mathrm{step}(x - b), \tag{3.3}$$

$$\mathrm{step}(-x + a)\,\mathrm{step}(x - b) \rightarrow \mathrm{step}(x - b) - \mathrm{step}(x - a) \quad \text{if } a > b \text{ else } 0, \tag{3.4}$$

$$\mathrm{step}(-x + a)\,\mathrm{step}(-x + b) \rightarrow \mathrm{step}(-x + b) \text{ if } b \leq a \quad \text{else} \quad \mathrm{step}(-x - a), \tag{3.5}$$

$$\mathrm{step}(\mathrm{step}(\pm x \mp a) f + g) \rightarrow \mathrm{step}(\pm x \mp a)\,\mathrm{step}(f + g) + (1 - \mathrm{step}(\pm x \mp a))\,\mathrm{step}(g), \tag{3.6}$$

$$\mathrm{step}(f) \rightarrow a_0 + \sum_{i=1}^{N} a_i \,\mathrm{step}(\pm x \mp b_i) \quad f \notin \mathcal{RA}, \quad a_i = \pm 1 \text{ by algorithm } ^{\ddagger} \tag{3.7}$$

$$\mathrm{step}(a) \rightarrow 1 \text{ if } a > 0 \text{ else } 0 \quad a \in \mathcal{RA}. \tag{3.8}$$

[‡]The algorithm is given in the proof.

Example 3.3. Three different definitions for the absolute value function can be given using piecewise functions:

$$abs_1 := \begin{cases} x & x > 0 \\ -x & x \geq 0 \end{cases} \quad abs_2 := \begin{cases} x & x \geq 0 \\ -x & x < 0 \end{cases} \quad abs_3 := \begin{cases} x & x > 0 \\ -x & x < 0 \\ 0 & x = 0 \end{cases}.$$

In Section 6 we describe the conversion of piecewise expressions to step-terms. We give the PPFR representation of these functions and compute the normal form for each: $abs_1 = x\ \mathrm{step}(x) - x\ \mathrm{step}(-x)$. The only rule which can be applied is rule 2, as $-x$ is 0 at $x = 0$ hence: $x\ \mathrm{step}(x) - x\ (1 - \mathrm{step}(x))$ resulting in $-x + 2\,x\,\mathrm{step}(x)$. For the second $abs_2 = -x + 2x\,\mathrm{step}(x)$ is in normal form. And $abs_3 = x\,\mathrm{step}(x) - x\ \mathrm{step}(-x) + 0\ (1 - \mathrm{step}(x) - \mathrm{step}(-x))$ resulting in the normal form $-x + 2x\,\mathrm{step}(x)$. We see that all three normal forms are identical.

Note that deciding extensional equality is not a trivial problem. Let f be some PPFR term. Consider the term

$$f\ (1 - \mathrm{step}(x - a) - \mathrm{step}(-x + a)),$$

its interpretation is extensional equivalent to 0 if $f(a) = 0$, but the term is not identical 0. For these situations we have rule (2). If $f(a) = 0$, we apply rule 2 resulting in

$$f - f\,\mathrm{step}(x - a) - f(1 - \mathrm{step}(x - a)) = 0.$$

Theorem 3.4. (Normal Form) *For any term of PPFR we can compute its normal form. The PPFR normal form is well defined.* $\mathbf{normal}(\mathbf{normal}(f)) \equiv \mathbf{normal}(f)$. *The normal form decides extensional equality.* $f \simeq g$ *iff* $\mathbf{normal}(f - g) \equiv 0$.

Remark. It is possible to define a canonical form; a canonical form is a somehow stronger definition of a normal form having the additional property: if $f \simeq g$ then $\mathbf{normal}(f) \equiv \mathbf{normal}(g)$ (compare with, e.g., disjunctive normal forms in boolean algebra and canonical forms for polynomials). In this possible canonical form the coefficients h_i are constants. This can be accomplished by changing rule 2 to $f(x)\,\mathrm{step}(-x + a) \to f(x)(1 - \mathrm{step}(x - a)) - f(a)(1 - \mathrm{step}(x - a) - \mathrm{step}(-x + a))$, all proofs, with some minor modifications, would stay the same. But the canonical form would change the relation of PPFR to the piecewise representation, any condition would be represented as $x < a, x = a, x > a$, no $\leq$ or $\geq$. Even after we worked out all these details, we have chosen to work with the normal, not the canonical form.

Proof. We prove the theorem in four parts. (1) We show by induction on the structure that any PPFR term can be transformed to its normal form by performing a finite sequence of rule applications. (2) If a term is in normal form then none of the rules can be applied. (3) The rule system is confluent, meaning that the order in which the rules are applied does not matter. And, finally, (4) the normal form decides extensional equality: $f \simeq g$ iff $\mathbf{normal}(f - g) \equiv 0$. Also all occurrences of the composition operator $\circ$ are removed using the rules given in 2.1 before we start applying the normal form rules. As stated we do not show that the rules in 2.1 terminate because they are common.

(1) Induction over the signature of a term. If f and g are PPFR terms in normal form, then $f + g, f \times g, f \circ g$ can be converted to their normal form using a finite number of rule applications.

Atomic Expressions. Atomic expressions are the expressions which are not "created" by one of the operation $+, -, \times, \circ$. These are the numbers, x and step. f step free is in normal form. [§]

step$(a\ x - b)$ $a, b \in \mathcal{RA}$ is transformed into normal form by rule 1 or 8. Strictly, only step(x) is atomic since for example step$(x + 2)$ is step$(x) \circ (x + 2)$, but for reasons of simplicity we cover the general step$(a\ x - b)$ case. If $a = 0$ then the normal form is 0 for $b > 0$ and 1 for $b \leq 0$. If $a \neq 1, a \neq -1$ we use rule 1 step $\left(\frac{ax-b}{|a|}\right)$.

The normal form of step(g) where g is step-free is examined in the composition part.

Sum $f+g$. The sum of two terms in normal form will not produce any multiplication of step and no nested steps. To convert the sum to normal form the steps are first collected, then we check for rule 2. For any term of the form $h(x)\,\text{step}(-x+b)$ we verify if $h(b) \neq 0$ else we replace step$(-x + b)$ by $1 - \text{step}(x - b)$ and collect the steps again.

Product $f \times g$. The product of two terms in normal form can produce a finite number of products of steps with steps. Using rules 3, 4, 5 we can reduce each product of steps to a sum of steps. The result is then reduced using rule 2 as described above. We converted $f \times g$ to its normal form using a finite number of rule applications.

Composition $f \circ g$. Computing the composition of two PPFR functions in normal form can result in nested steps. We only have to deal with arguments of step, e.g. step$(x^2 + 4)$ since we removed all other compositions, e.g. $x^2 \circ \text{step}(x) = \text{step}(x)\,\text{step}(x) = \text{step}(x)$. The zero-one rule 6 removes a step inside a step:

$$\begin{aligned}
&\text{step}\left(f_0 + \sum_{i=1}^{N} f_i\,\text{step}(x - a_i) + \sum_{i=1}^{M} h_i\,\text{step}(-x + b_i)\right) \\
&\quad = \text{step}(x - a_1)\ \text{step}\left(f_0 + \sum_{i=1}^{N} f_i\,\text{step}(x - a_i) + \sum_{i=1}^{M} h_i\,\text{step}(-x + b_i)\right) \\
&\qquad + (1 - \text{step}(x - a_1))\ \text{step}\left(f_0 + f_1 + \sum_{i=2}^{N} f_i\,\text{step}(x - a_i) + \sum_{i=1}^{M} h_i\,\text{step}(-x + b_i)\right).
\end{aligned}$$

After a finite number of applications we obtain a term with no nested steps. It will contain products of steps which are reduced using rule 3, 4, 5 as above.

If g is a step-free function, a polynomial, then step(g) is reduced to a sum of steps. The idea is the following, we create a step term of the form

$$t := a_0 + \sum_{i=1}^{N} a_i\,\text{step}(\pm x \mp b_i))$$

where $a_i \in \{1, -1\}$ and $b_i \in \mathcal{RA}$ with $t = 1$ if $p > 0$ and $t = 0$ if $p \leq 0$. See Figure 1.

The pseudo code algorithm computing the normal form of step(g):

```
snormal(p:Polynomial) ==
  a1, . . . ,an := all real roots of p sorted smallest first
  count the multiplicity of the roots
  all roots with odd multiplicity are only once in the list
  all roots with even multiplicity are twice in the list
  i:=0;o:=0;
```

[§]The normal form of a polynomial is $a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$.

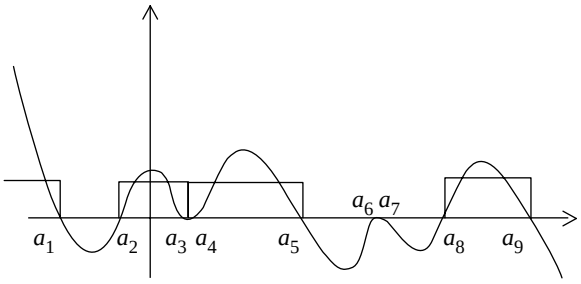

Figure 1. Normal form of step of a polynomial.

```
  if p("-infinity")< 0 then where:=-1 else where:=1 end;
            /* test the highest coefficient and exponent */
  if where=1 then o:=step(-x+a[i]);i:=i+1; endif;
  repeat
    if where=-1 and a[i]=a[i+1] then  /* nothing*/
    else o:=o+step(x-a[i])-step(-x+a[i+1]); where:=-where;
    endif
    i:=i+2;
  until no more two a[i]
    if one more a[i] then o:=o+step(x-a[i])
 return o
```

The roots of polynomials over the real algebraic numbers are computable. We consider only real roots of g. For example, the normal form of $\text{step}(x^2+2)$ is 1, that of $\text{step}(-x^2-2)$ is 0 and $\text{step}(x^2)$ has the normal form $\text{step}(-x)+\text{step}(x)$.

(2) No rule. can be applied to a term in normal form. We verify this property rule by rule. Let f be a term in normal form then:

1. Rule 1, 6, 7, 8 cannot be used because the argument of all step in the normal form is $\pm x \mp a$ with $a \in \mathcal{RA}$.
2. Rule 2 cannot be used, this is stated in the definition of the normal form: $h_i(b_i) \neq 0$.
3. Rules 3, 4 and 5 cannot be used because in the normal form we have no product of step with step.

(3) Confluence. We show that in whichever order the rules are applied we end up with the same result, the unique normal form. Confluence if proven using the classical lemma of Newman (1942) "A terminating rule system is confluent iff it is locally confluent". Hence we have to verify local confluence of the rules, meaning: if to a term t two rules can be applied to the same expression resulting in t_1 and t_2, then there exists a sequence of rules which reduces t_1 and t_2 to the same normal form. t_1, t_2 is called a *critical pair*.

Rule 1 cannot produce any critical pairs.

To a subterm of the form $f\,\text{step}(x-a)\,\text{step}(-x+b)$ with $f \notin \mathcal{RA}$ rules 2, 3, 4 or 5 can be applied. Applying first rule 2 results in

$$f(x)\,\text{step}(a-x)\;(1-\text{step}(x-b)) = f(x)\,\text{step}(x-a) - f(x)\,\text{step}(x-a)\,\text{step}(x-b).$$

This is 0 for $a >= b$ and $f\,\text{step}(x-a) - f\,\text{step}(x-b)$ for $a < b$. Applying first rule 4

we end up with the same result. The other cases, e.g. $\text{step}(-x+a)\,\text{step}(-x+b)f$, are similar.

To a term having a subterm of the form

$$\text{step}(x-a)\,\text{step}(x-b)\,\text{step}(-x+c)$$

3, 4 or 5 could be applied. w.o.l.g. $a >= b$ the term reduces to 0 if $c \leq a$ and to $\text{step}(x-a) - \text{step}(-x-c)$ if $a < c$ by all possible orders of application of rules 3,4 and 5. The other case $\text{step}(x-a)\,\text{step}(-x+b)\,\text{step}(-x+c)$ is similar.

Let f, g be step free and h a PPFR function. Then to a term having a subterm of the form

$$\text{step}(f\,\text{step}(x-a) + g\,\text{step}(x-b) + h)$$

rule 6 can be used in different ways to this subterm. Computing both ways (assuming $a \leq b$) we end up with the same normal form:

$$\begin{aligned}\text{step}(h) &- \text{step}(x-a)\,\text{step}(h) + \text{step}(x-a)\,\text{step}(f+h) - \text{step}(f+h)\,\text{step}(x-b)\\ &- \text{step}(x-b)\,\text{step}(f+g+h).\end{aligned}$$

Rules 7 and 8 cannot produce any critical pairs.

We conclude: the rule system is confluent. We showed that our rule system provides us with an unique normal form.

(4) Equality. Last, we are left to show that

$$f \simeq g \Leftrightarrow \mathbf{normal}(f-g) \equiv 0.$$

The implication $\mathbf{normal}(f-g) \equiv 0 \Rightarrow f \simeq g$ is the easy part, we have to verify rule by rule: applying a rule to a term t, the resulting term t' is extensionally equal $t \simeq t'$. We show this in example with rule 2. $f\ \text{step}(-x+a)$ is f iff $x < a$, and 0 iff $x \geq a$. We use rule 2: if $f(a) = 0$, resulting in $f\ (1 - \text{step}(x-a))$, which is f iff $x < a$, $f(a)$ iff $x = a$, and 0 iff $x > a$.

The other direction: if $f \simeq 0$ then $\mathbf{normal}(f) \equiv 0$.

Suppose f is a term of PPFR with $f(p) = 0$ for all $p \in \mathcal{RA}$. The normal form of f has the form:

$$f_0 + \sum_{i=1}^{N} f_i\,\text{step}(x-a_i) + \sum_{i=1}^{M} h_i\,\text{step}(-x+b_i).$$

We use the fact that the polynomials f_i and h_i have a maximal degree m, hence if $f_i(p)$=0 for $m+1$ different points $p \in \mathcal{RA}$ then $f_i \equiv 0$.

Definition 3.5. Let f be a PPFR function in normal form. Let α_i be the sorted list of a_i, b_i from the normal form. A number x is **close** to the number α iff

$$(x < \alpha \wedge \forall \alpha_j < \alpha :\ x > \alpha_j) \bigvee x = \alpha \bigvee (x > \alpha \wedge \forall \alpha_j > \alpha :\ x < \alpha_j).$$

We study the normal form of f close to points α and distinguish four cases depending on whether $\alpha = a_i \wedge \alpha \neq b_k$, $\alpha_i = b_k \wedge \alpha \neq a_j$, $\alpha = a_j = b_k$ (critical case) or $\alpha \neq a_j \wedge \alpha \neq b_k$ and show for all these cases that the corresponding sums of f_k and h_l are zero for an infinite number of points close to α:

1. $\alpha = a_i \wedge = \alpha \neq b_j$. Then f at points x close to point α has the form
$$l(x) + f_i(x)\,\mathrm{step}(x - a_i)$$
where $l(x) = f_0(x) + \sum_{i|a_i<\alpha} f_i(x) + \sum_{i|b_i>\alpha} h_i(x)$. This is $l(x) \simeq 0$ for $x \leq \alpha$ and $l(x) + f_i(x) \simeq 0$ for $x > \alpha$. Because these functions are polynomials we can conclude that $f_i(p) \simeq 0$ for an infinite number of points.
2. $\alpha = b_k\ \alpha \neq a_j$. Then f at points x close to point α has the form
$$l(x) + h_i(x)\,\mathrm{step}(-x - b_j)$$
where $l(x) = f_0(x) + \sum_{i|a_i<\alpha} f_i(x) + \sum_{i|b_i>\alpha} h_i(x)$. This is $l(x) + h_i(x) \simeq 0$ for $x \leq \alpha$ and $l(x) \simeq 0$ for $x > \alpha$. Again, using that these functions are polynomials we can conclude that $h_i(p) \simeq 0$ for an infinite number of points.
3. $\alpha = a_i = b_j$. for some i, j then f at points x close to α has the form
$$l(x) + f_i(x)\ \ \mathrm{step}(x - a_i) - h_j(x)\,\mathrm{step}(-x + a_j) \simeq 0$$
where $l(x) := f_0(x) + \sum_{i|a_i<\alpha} f_i(x) + \sum_{i|b_i>\alpha} h_i(x)$.
For $x < \alpha$ this is $l(x) - f_i(x) \simeq 0$, for $x > \alpha$ we have $l(x) - h_i(x) \simeq 0$, and for $x = \alpha$ we get $l(\alpha) = 0$. Now since $h_i(\alpha) \neq 0$, normal form, but $l(\alpha) = 0$ and $l + h_i \simeq 0$ we conclude using continuity of the $l(x)$ and $h_i(x)$ that $l(p) = 0$, $f_i(p) = 0$, and $h_i(p) = 0$ for an infinite number of points p. Note, for this proof the property $h_i(\alpha) \neq 0$ of the normal form is essential.
4. $\alpha \neq a_j \wedge \alpha \neq b_k$. This is the simplest case, for all p close to α $f(p) = 0$.

As for any choice of α we showed that the corresponding polynomials are zero for an infinite number of points p and we covered all possible combinations of the f_i, h_i we conclude that $f_i \equiv 0$ and $h_i \equiv 0$, hence the normal form of f is identical 0, $\mathbf{normal}(f) \equiv 0$.

The application of the rules is not really restricted to functions containing no step. $\mathrm{step}(f\ \mathrm{step}(x) + h)$ can be reduced to $\mathrm{step}(x)\ \mathrm{step}(f + h + (1 - \mathrm{step}(x))\,\mathrm{step}(h)$ even if f or h contain steps. (von Mohrenschildt, 1994; Engeler and von Mohrenschildt, 1995). □

Note. The existence of this normal form depends only on the computability of the zeros of the "ground function ring", the functions free of steps, and the continuity of these functions. It is not difficult to extend PPFR with function symbols, e.g. *exp*, as long as we can compute the zeros in the extended ring and the functions represented by the function symbols are continuous.

4. Lattice Structure

Using step, we can express the supremum and the infimum of two functions. We show that PPFR is a lattice, which means supremum and infimum of two functions of the ring belong to the ring. For example, $\sup(x, -x)$ represents the piecewise function $-x$ for $x \leq 0$ and x for $x > 0$, the absolute value function.

Definition 4.1. We define the sup and the inf of two PPFR functions f, g:
$$\sup(f, g) = f\,\mathrm{step}(f - g) + g\,\mathrm{step}(g - f) + \frac{f + g}{2}(1 - \mathrm{step}(f - g) - \mathrm{step}(g - f)),$$
$$\inf(f, g) = g\,\mathrm{step}(f - g) + f\,\mathrm{step}(g - f) + \frac{f + g}{2}(1 - \mathrm{step}(f - g) - \mathrm{step}(g - f)).$$

THEOREM 4.2. sup *and* inf *are well defined. PPFR is a lattice.*

PROOF. We have to show that **(1)** sup and inf are compatible with the ordering relation of the field of constants $\mathcal{RA}$. Further, we have to show that **(2)** $\sup(f,f) \simeq f$, **(3)** $\sup(f,g) \simeq \sup(g,f)$, and **(4)** $\sup(f,\sup(g,h)) \simeq \sup(\sup(f,g),h))$ for any f,g,h.

(1) The supremum and infimum are compatible with the ordering relation of the ground ring. That is, if $a < b$ then $\sup(a,b) = a$. $\sup(a,b) = a\,\mathrm{step}(a-b) + b\,\mathrm{step}(b-a) + \frac{a+b}{2}(1 - \mathrm{step}(a-b) - \mathrm{step}(b-a)) = a$

(2) $\sup(f,f) \simeq f\,\mathrm{step}(f-f) + f\,\mathrm{step}(f-f) + \frac{f+f}{2}(1 - \mathrm{step}(f-f) - \mathrm{step}(f-f)) \simeq f$.

(3) $\sup(f,g) \simeq f\,\mathrm{step}(f-g) + g\,\mathrm{step}(g-f) + \frac{f+g}{2}(1 - \mathrm{step}(f-g) - \mathrm{step}(g-f)) \simeq \sup(g,f)$.

(4) To show transitivity we can use (1)

$$\sup(f(p),\sup(g(p),h(p))) \simeq \sup(\sup(f(p),g(p)),h(p))$$

for any point p. □

EXAMPLE 4.3.

$$\sup(-x,x) = 2x\,\mathrm{step}(x) - x,$$
$$\sup(x^2-2,x) = (x - x^2 + 2)\,\mathrm{step}(x+1) + (x - x^2 + 2)\,\mathrm{step}(x-2) - x + 2\,x^2 - 4.$$

5. Computing Zeros and Solving Inequalities

We study equations and inequalities in PPFR. We show that:

THEOREM 5.1. *The solutions of PPFR equations are computable.*

THEOREM 5.2. *The solutions of PPFR inequalities are computable.*

We state these theorems separately because Theorem 5.2 depends on 5.1.

PROOF OF 5.1. Let $f = 0$ be the equation to be solved. First compute the normal form of

$$f = f_0 + \sum_{i=1}^{N} f_i\,\mathrm{step}(x - a_i) + \sum_{i=1}^{M} h_i\,\mathrm{step}(-x + b_i).$$

If in the normal form all f_i and h_i are constant then we have the solution, it is the union of the intervals where this piecewise constant function is 0. The solution can be translated to an interval representation using the step to piecewise conversion algorithm given in 6.

Otherwise, let α_i be the sorted list of a_i, b_i of the normal form and L be the set of all (step-free) functions l_j $l_j := f_0 + \sum_{i|a_i \le p_j} f_i + \sum_{i|b_i \ge p_j} h_i$ where p_j is $\frac{1}{2}(\alpha_{i+1} + \alpha_i)$ for $i = 1,\ldots,N+M$ and $p_0 = \alpha_1 - 1$ and $p_{M+N+1} = \alpha_{n+m} + 1$.

Next solve all these $N + M + 2$ equations, where $N + M$ is the number of steps in the normal form of f. Note that we can either solve all these equations over the hole real axis or on the intervals $[\alpha_i, \alpha_{i+1}]$. Doing the letter we have to include the α_i to the list of possible zeros. Having all these zeros we verify if they are solutions of $f = 0$. We obtain all solutions of $f = 0$. □

Theorem 5.1 depends only on the computability of the zeros of step-free terms. If we extend PPFR with additional function symbols, e.g. *exp*, then we have to verify that we can solve the step-free equations.

Proof of 5.2. Any inequality of the form $f > 0$ can be written as $\text{step}(f) = 1$ and we use Theorem 5.1 to solve the equation $\text{step}(f) - 1 = 0$. The inequality $f >= 0$ can be written as $\text{step}(-f) = 0$. Again Theorem 5.1 is used to solve these equations. □

Example 5.3. Solve $x^2 - 4\,\text{step}(x) = 0$. The equation contains a single step, hence we solve the two equations: $x^2 - 4 = 0$ and $x^2 = 0$. The solutions are $2, -2, 0$. Next, we verify these solutions and find that 2 and 0 are the solutions of the original equation.

Next we solve $x^2 - 4\,\text{step}(x) > 0$. According to 5.2 we have to solve $\text{step}(x^2 - 4\ \text{step}(x)) = 1$. Computing the normal form results in $\text{step}(-x) + \text{step}(x - 2)$. In the normal form the f_i are constant, hence we determine the intervals where $\text{step}(-x) + \text{step}(x - 2)$ is 0 using the algorithm in 6. We obtain $x < 0 \vee x > 2$. Hence the solution is $x \in (-\infty, 0) \cup (2, \infty)$.

6. Converting Piecewise and PPFR Functions

In this section, we describe the process of converting a piecewise-expression into a step-term and vice versa.

The `piecewise` function $\texttt{piecewise}(C_1, f_1, C_2, f_2, ...)$ is interpreted as an if .. elseif .. elseif .. else .. statement. If the first condition C_1 is true, then the `piecewise` expression evaluates to the first function f_1, otherwise, if the second condition C_2 is true, then f_2, and so on. It is possible to give a last argument without a condition for the else case. If all conditions are false and no else case is provided, the piecewise function evaluates to its default value 0.

6.1. piecewise to PPFR

The conversion of a `piecewise` expression into its step representation is straight forward. For every condition in the piecewise-expression we construct a step term which is 1 if and only if the condition is satisfied. We can describe the algorithm by a conversion function S:

$$
\begin{aligned}
S(x > a) &\to \text{step}(x - a)\\
S(x \le a) &\to 1 - \text{step}(x - a)\\
S(x < a) &\to \text{step}(-x + a)\\
S(x \ge a) &\to 1 - \text{step}(-x + a)\\
S(\texttt{not}\,E) &\to 1 - S(E)\\
S(E\ \texttt{and}\ B) &\to S(E)S(B)\\
S(E\ \ \texttt{or}\ B) &\to \text{step}(S(E) + S(B))\\
S(\texttt{piecewise}(E_1, f_1, E_2, f_2, E_3, \cdots)) &\to S(E_1)f_1 + S(E_2)(1 - S(E_1))f_2\\
&\quad + S(E_3)(1 - \text{step}(S(E_1) + S(E_2))f_3\\
&\quad + \cdots.
\end{aligned}
$$

Example 6.1. Let us apply S to the absolute value function,

$$\begin{aligned}&\texttt{piecewise}(x>0,x,x\le 0,-x)\\ &=S(x>0)(1-S(x\le 0))x+(1-S(x>0))S(x\le 0)(-x)\\ &=\operatorname{step}(x)(1-\operatorname{step}(-x))(x-(1-\operatorname{step}(x))\operatorname{step}(-x))-x=2x\operatorname{step}(x)-x.\end{aligned}$$

6.2. PPFR to piecewise

We give the algorithm to convert a PPFR term f into an semantical equivalent `piecewise` expression.

1. Compute the normal form of f.
2. Sort the a_i and b_i of the normal form in increasing order resulting in $\alpha_1, \alpha_2, \ldots, \alpha_n$.
3. We use the sorted α_i to determine the type of the conditions, which is one of $x<a, x\ge a$ or $x\le a, x>a$ or $x<a, x=a, x>a$, for the `piecewise` function according to:
   (a) if $\alpha_i=\alpha_{i+1}$, that is, there are $a_i=b_i$, then the piecewise function contains the condition $x<\alpha_i\ , x=\alpha_i\ , x>\alpha_i$;
   (b) if α_i is an a_j, then the piecewise function contains the condition $x\le\alpha_i, x>\alpha_i$;
   (c) if α_i is a b_j, then the piecewise function contains the condition $x<\alpha_i, x\ge\alpha_i$.
4. Determine the functions g_i in every piece. (See 5.1 for the construction of the l_i.)

$$\texttt{piecewise}(cond_1, l_1, cond_2, l_2, \cdots, cond_k, l_k).$$

Example 6.2. Let us convert $\operatorname{step}(-x-2)+x\operatorname{step}(x)-\operatorname{step}(x-3)$ to a piecewise expression:

1. The zero points α_i are $-2, 0, 3$.
2. The conditions are: $x\le -2, x\le 0, x\le 3, x>3$.
3. The piecewise function has the form
$$\texttt{piecewise}(x\le -2, g_1, x\le 0, g_2, x\le 3, g_3, x>4, g_4).$$
4. We determine the g_i by finding the sum of f_i and c_i in interval i: $g_i := f_0 + f_{i_1} + \cdots f_{i_n} + c_{j_1} + \cdots + c_{j_k}$. $\texttt{piecewise}(x\le -2, 1, x\le 0, 0, x\le 3, x, x>3, x-1)$.

Let us convert $\operatorname{step}(x)+2\operatorname{step}(-x)+\operatorname{step}(x-2)$ to a piecewise. As the normal form has $\operatorname{step}(x)$ and $\operatorname{step}(-x)$ we are in the (a) case for the 0 and in the (b) case for the $\operatorname{step}(x-2)$. Hence, we obtain the conditions $x<0, x=0, x\le 2, x>2$. This corresponds to the piecewise function $\texttt{piecewise}(x<0, 2, x=0, 0, x\le 2, 1, x>2, 2)$.

7. Some Examples

These examples were computed using an implementation of PPFR in Maple:

$$|2-|x||=\begin{cases}-2-x & x\le -2\\ 2+x & x\le 0\\ 2-x & x\le 2\\ x-2 & 2<x\end{cases}$$

$$f(x) := \begin{cases} x & x < -1 \\ x^2 & x \le 1 \\ x & 1 < x \end{cases} \qquad g(x) := \begin{cases} x+1 & 0 \le x \\ 3\,x & \text{otherwise} \end{cases}$$

$$f(x)+g(x) = \begin{cases} 1+2\,x & x < -1 \\ 3+2\,x & x \le 0 \\ 2+2\,x & x \le 2 \\ -2+4\,x & 2 < x \end{cases} \qquad f(x)g(x) = \begin{cases} x^2+x & x \le -1 \\ 2+3\,x+x^2 & x \le 0 \\ -3\,x^2+6\,x & x \le 2 \\ -6\,x+3\,x^2 & 2 < x \end{cases}$$

$$g(f(x)) = \begin{cases} 3\,x & x < -1 \\ x^2+1 & x \le 1 \\ x+1 & 1 < x \end{cases} \qquad f(g(x)) = \begin{cases} 3\,x & x < -1/3 \\ 9\,x^2 & x < 0 \\ x+1 & 0 \le x \end{cases} .$$

Other applications of PPFR can be found in Jeffrey *et al.* (1997) and von Mohrenschildt (1996). The step function with normal form is implemented in MapleV.4. step is called `unitstep` (not documented) and `'simplify/unitstep'(expression,var)` computes the normal form with respect to `var`. `convert(expression,piecewise,var)` converts to the `piecewise` representation and `convert(expression,unitstep)` converts to the PPFR representation.

References

Aberer, K. (1990). Normal forms in function fields. *Proc. ISSAC 90*, pp. 1–7.

Engeler, E., von Mohrenschildt, M. (1995). *The Combinatory Program.* Birkhäuser.

Filippov, A. F. (1988). *Differential Equations with Discontinuous Right-hand Sides.* Kluwer Academic Publishers.

Gonnet G. (1985). Determining equivalence of expressions in random polynomial time. *Proc. 16th ACM Symp. Theory of Computing*, 1984, pp. 334–341.

Jeffrey, D.J., Labahn, G., von Mohrenschildt, M., Rich, A.D. (1997). Integration of signum, piecewise and related functions. *ISSAC 97.*

Kung, H. T. (1973). The computational complexity of algebraic numbers. *Conf. Rec. Fifth Annual (ACM) Symp. on Theory of Computing 30.2.*

Newman, M. H. (1942). On theories with a combinatorial definition of equivalence. *Ann. Math.*, **43**, 223–268.

Richardson, D. (1992). The elementary constant problem. *Proc. ISSAC '92*, pp. 108–116.

Roy, M. F., Szpirlas, A. (1990). Complexity of computation on real algebraic numbers. *J. Symb. Comput.*, **10**, 39–51.

Rump, S. (1976). On the sign of a real algebraic number. *Proc. 1976 ACM on Symb. and Algeb. Computation. ACM76.*

von Mohrenschildt, M. (1994). Discontinuous ordinary differential equations, PhD Thesis, Department of Mathematics, ETH Zurich, 10768.

von Mohrenschildt, M. (1996). Using piecewise to solve classes of control theory problems. *MapleTech97.*

Originally Received 4 March 1997
Accepted 22 June 1998