Proving termination with multiset orderings in PVS: theory, methodology and applications *

José A. Alonso, María J. Hidalgo, and Francisco J. Martín–Mateos
{jalonso, mjoseh, fjesus}@us.es

Departamento de Ciencias de la Computación e Inteligencia Artificial.
Escuela Técnica Superior de Ingeniería Informática, Universidad de Sevilla
Avda. Reina Mercedes, s/n. 41012 Sevilla, Spain

Abstract. There exist a number of non-trivial termination proofs of functions (or algorithms) which are carried out more naturally and simpler using well-founded multiset orderings. We present in this paper a methodology to organize and simplify these kind of termination proofs in the PVS specification and verification system. This methodology uses a well–known result due to Dershowitz and Manna, which states that every well–founded relation on a set T can be extended to a relation on finite multisets over T which is also well-founded. Therefore, we also present a formalization of this theorem in PVS. We think this methodology can be very useful to develop non-trivial termination proofs in PVS. To illustrate this, we have applied our methodology to formalize in PVS some termination proofs, like an iterative version of the Ackermann's function, the McCarthy's 91 function and also a tail-recursive definition of a schema of functions defined by double recursion.

1 Introduction

The use of well–founded orders for proving termination of recursive functions was suggested by Floyd in [5]. The idea is to find a set T, with a well–founded order $<$ and a measure function m mapping the arguments of the function into the elements of T, such that the measure of the arguments is reduced in each recursive call. Due the well–foundedness of $<$, this measure can not decrease indefinitely and hence, the termination of the function is assured.

The most used well–founded order is the usual order on natural numbers and the lexicographic order on n–tuples of natural numbers. However, Dershowitz and Manna [4] showed that every well–founded relation on a set T can be extended to a well–founded relation on the finite multisets over T. They also proved that the use of multiset orderings allows to construct simple and intuitive measure functions to carry out non–trivial proofs of termination. In particular, they showed that the multiset ordering can be used to prove the termination of Ackermann's function, McCarthy's 91 function and production systems, programs defined in term of rewriting rules.

* This research was partially funded by Spanish Ministry of Education and Science under project MTM2009-13842-C02-02 and TIN2009-09492

On the other hand, in the field of formal verification, we often have to tackle the problem of proving termination of of programs, logic reasoning systems or rewriting systems by using one of the current systems like ACL2, COQ, HOL, Isabelle, PVS,.... These proofs of termination are non–trivial and, in many cases, the use of multiset orderings can be very useful. Thus, in order to mechanize proofs of termination in a formal system by using multiset orderings, it would be appropriated to formalize these orderings in the corresponding system. In [16] we developed a formalization in ACL2 of the well–foundedness of multiset orders. We have used this result in several works of formalization in ACL2: Newman's lemma [15], tableaux algorithms [10, 2, 6], Dickson's lemma [9] and Higmann's lemma [11]. We have developed other works of formalization in other formal systems more appropriated to the new problems, due mainly to its expresiveness. In some of these works it has been necessary to carry out difficult proofs of termination, in which we have used multiset orderings. In fact, when we formalized in PVS a tableaux algorithm for the $\mathcal{ALC}$ description logic [2], multisets were a key tool for proving its termination. Thus, we think it is useful to hold in PVS a theory of wellfoundeness of multiset orderings, in order to be able to carry out this kind of termination proofs in this system.

In this paper we present the PVS theory about the well-foundedness of the multiset order induced by a well-founded order over a set T. To do this, we have extended the multiset library of PVS in order to include well–foundedness of the multiset order relation. We have proved the Dershowitz and Manna theorem [4] in an abstract way, allowing its instantiation to prove the well–foundedness of particular multiset relations. We also present a methodology to systematize the proofs of termination of recursive functions using multiset orderings. Finally, we show three case studies where we have used this methodology for proving termination of different tail recursive functions.

The paper is structured as follows. The second section presents how we have proved the well–foundedness of multiset relations induced by well–founded relations, reusing the PVS multiset library. In the third section, we show a detailed outline of the methodology suggested for proving termination properties. The fourth section shows the application of this methodology to prove, in PVS, termination of several recursive functions: a tail–recursive definition of Ackermann's function, an iterative definition of McCarthy's 91 function and an iterative function to compute a generic schema for double recursion. Finally, in the last section we show some conclusions and outline future lines of work.

We will remark the main features of the PVS system according as we will use them and we will explain both the definitions and the PVS proofs in a form that can be understood without being an expert in PVS. However, a detailed description of this system can be seen in [13]. Moreover, due to the lack of space, we will skip details of the proofs. Nevertheless, the whole formalization is available at `http://www.cs.us.es/~mjoseh/PROTEMO/`

2 Well–founded ordering of multisets

A *multiset* M over a set T is a function from T to the set of natural numbers. If $M(x) > 0$ for only finitely many $x \in T$, then we say that the multiset is finite. The set of all finite multisets over T is denoted as $\mathcal{M}(T)$. Multisets are a formalization of the intuitive idea of "sets with repeated elements", being $M(x)$ the number of occurrences of x in the multiset M. In [4], Dershowitz and Manna proved that every well-founded relation on a set T induces a well-founded relation on $\mathcal{M}(\mathcal{T})$. In this section we present how we have extended the PVS multiset library to include the formalization of thi result.

In the literature there are different equivalent definitions of well–founded relation (see [14] for a formalization of these definitions in MIZAR). Specifically, if $<$ is a binary relation on T, the following assertions have been proved equivalent:

- Every non–empty subset of T has a $<$–minimal element.
- Well founded induction holds for $(T, <)$.
- For every set V, there exist recursively defined functions from T into V.
- Recursively defined functions on T are unique.
- There are no descending w chains in $(T, <)$.

The first of these characterizations is just the definition of well–founded relation included in the prelude of PVS. However, the characterization of well–foundedness that we will use is not exactly any of the above. We will use a characterization, provided in [1], based on the concept of well founded part of a relation $<$. Given a binary relation $<$ on a set T, the well-founded part of $<$ is the set of elements $a \in T$ such that there is no infinite descending sequence $...a_1 < a_0 < a$. Following [1], a relation $<$ is well-founded if and only if its well-founded part is T. In [12], T. Nipkow shows an inductive proof of the well–foundedness of the multiset reduced relation, due to Wilfried Buchholz. He defines the well founded part of a binary relation $W(<)$ inductively by the rule

$$\frac{\forall y < x . y \in W(<)}{x \in W(<)}$$

We have used this definition in PVS due to the system's ability to define sets inductively. The sketch of our PVS formalization of the Dershowitz and Manna theorem is the following:

1. We have defined inductively the well–founded part of a binary relation $(T, <)$ and we have proved that $(T, <)$ is well–founded if and only if $W(T) = T$.
2. We have proved that well–foundedness is preserved by transitive closure.
3. We have proved that the *multiset reduced relation* $<_1$ (where $N <_1 M$ if N is obtained by replacing an element a of M by a finite multiset K of elements less than a) induced by a well–founded relation is well–founded. The proof formalized in PVS is based on the proof described by T. Nipkow [12].

4. Finally, we have proved that the *multiset relation* $<_{mult}$ (where $N <_{mult} M$ if N is obtained by replacing a multiset K_1 of elements of M by a multiset K_2 of elements less than any of K_1) is the transitive closure of $<_1$ if the relation $<$ is transitive. So, we proved that the *multiset relation* $<_{mult}$ of a well–founded relation $<$ is well–founded.

In the subsequent subsections we will explain with more detail these four points.

2.1 A characterization of well–foundedness

The definition of well–foundedness included in the PVS prelude is the usual one: a relation $<$ on T is well–founded if every non-empty subset of T has a $<$–minimal element. Let us start by describing the PVS proof of the equivalence between this definition and the alternative definition of well–foundedness presented in [1].

Definition 1. *Given a binary relation $<$ defined on a set T, the well-founded part of T with respect to $<$, denoted as $W(T,<)$, is the smallest subset of T closed under the set of rules* $(\forall a \in T)[(\forall y < a)[y \in W(T,<)] \rightarrow a \in W(T,<)]$.

In order to formalize this definition in PVS, we introduce T and $<$ as theory parameters and we use the support that PVS provides for constructing inductive definitions of sets or predicates to make the following definition:

```
wf_cs[T: TYPE+, <: pred[[T,T]]]: THEORY

well_founded_part(x): INDUCTIVE bool =
  FORALL y: y < x IMPLIES well_founded_part(y)
```

It is important to note that the above inductive definition of the well–founded part generates, automatically, two induction axioms, which allow us to prove properties by induction on the defined set:

- Weak induction axiom for the well–founded part:

$$\frac{(\forall x)[(\forall y)[y < x \rightarrow P(y)] \rightarrow P(x)]}{(\forall x)[x \in W(T,<) \rightarrow P(x)]}$$

- Induction axiom for the well–founded part:

$$\frac{(\forall x)[(\forall y)[y < x \rightarrow y \in W(T,<) \wedge P(y)] \rightarrow P(x)]}{(\forall x)[x \in W(T,<) \rightarrow P(x)]}$$

The following theorem, that characterizes the well–foundedness of a relation by means of its well–founded part is proved by using these schemes.

Theorem 1. *$(T,<)$ is well–founded if and only if $W(T,<) = T$.*

```
well_founded_part_nsc: THEOREM
  well_founded?[T](<) IFF (FORALL x: well_founded_part(x))
```

The PVS proof of the necessary condition is carried out by induction on $(T,<)$ applied to the predicate $x \in W(T,<)$. On the other hand, the sufficient condition is proved using the weak induction axiom for the well–founded part of T with respect to $<$.

2.2 Well–foundedness of the transitive closure

In order to prove that well–foundedness is preserved by transitive closure, it would be noted that the transitive closure of a relation can be also specified in PVS as an inductive relation. Thus, the induction axioms associated to $<^+$ will be automatically generated.

Definition 2. *The* transitive closure *of a binary relation $<$ on T, is the smallest relation $<^+$ such that* $(\forall x, y \in T)[(x < y \vee (\exists z)[x <^+ z \wedge z < y]) \rightarrow x <^+ y]$

```
tr_cl(<)(x,y): INDUCTIVE bool =
 x < y OR EXISTS z: tr_cl(<)(x,z) AND z < y
```

The main result about well–foundedness and transitive closure is the following:

Theorem 2. *If $(T, <)$ is well–founded, then $(T, <^+)$ is well–founded.*

```
well_founded_cl_tr: THEOREM
  well_founded?[T](<) IMPLIES well_founded?[T](tr_cl(<))
```

Using Theorem 1, this property is a consequence of the following property about the well-founded parts: $W(T, <) \subseteq W(T, <^+)$. We prove this last stament in PVS using the weak induction axiom generated by the definition of $W(T, <)$ applied to the predicate $P(x) \equiv x \in W(T, <^+)$.

2.3 Well–founded multiset relations in PVS

In order to specify in PVS the multiset relations, we have reused the PVS library about bags[1]. In this library, a multiset (bag) of elements in T is represented by means of a function with domain T and range the set of natural numbers. Let us start showing the specification of the `bag` and `finite_bag` types, and also the basic operations `insert` and `plus` (in the following denoted both as $U+$) included in the PVS library.

```
bag: TYPE = [T -> nat]
insert(x,b): bag = (LAMBDA t: IF x = t THEN b(t) + 1 ELSE b(t) ENDIF)
plus(a,b)  : bag = (LAMBDA t: a(t) + b(t))
bag_to_set(b): set[T] = {t: T | b(t) > 0}
is_finite(b): bool = is_finite(bag_to_set(b))
finite_bag: TYPE = {b: bag | is_finite(b)}
```

Given a relation $<$ in T, it induces two relations in the set of finite multisets over T, $\mathcal{M}(T)$:

- The *multiset reduction relation* denoted as $<_1$: $N <_1 M$ if there exists multisets M_0, K_2 and $b \in M$ such that $M = M_0 \uplus \{b\}$, $N = M_0 \uplus K_2$ and $(\forall a)[a \in K_2 \rightarrow a < b]$.

[1] Available at `http://shemesh.larc.nasa.gov/fm/ftp/larc/PVS-library/pvslib.html`

– The *multiset relation* denoted as $<_{mult}$: $N <_{mult} M$ if there exist multisets $M_0, K_1, K_2 \in \mathcal{M}(T)$ such that $K_1 \neq \emptyset$, $M = M_0 \uplus K_1$, $N = M_0 \uplus K_2$ and $(\forall a)[a \in K_2 \rightarrow (\exists b)[b \in K_1 \wedge a < b]]$. It can be proved that if $<$ is transitive, then $<_{mult}$ is the transitive closure of $<_1$.

We specify the relations $<_1$ and $<_{mult}$ in PVS by `less_1` and `less_mult`, respectively, as follows

```
less(K,a): bool = FORALL b: member(b,K) IMPLIES b < a

less_1(N,M): bool =
  EXISTS M_0,a,K: M = insert(a,M_0) AND N = plus(M_0,K) AND less(K,a)

less_mult(N,M): bool =
  EXISTS M_0,K1,K2: nonempty_bag?(K1) AND M = plus(M_0, K1) AND
                    N = plus(M_0, K2) AND
                    FORALL a: member(a,K2) IMPLIES
                              EXISTS b: member(b,K1) AND a < b
```

In order to prove that $<_{mult}$ is a well–founded relation, we prove first that if $<$ is transitive, then $<_{mult}$ is the transitive closure of $<_1$ ($<_{mult}$=$<_1^+$).

```
less_bag: PRED[[finite_bag[T],finite_bag[T]]] = tr_cl(less_1)

less_mult_equiv_less_bag: THEOREM
  transitive?[T](<) IMPLIES (less_mult(N,M) IFF less_bag(N,M))
```

Therefore, by the Theorem 2, it is enough to prove that the relation $<_1$ is well–founded. So, this is the main lemma we have stablished in PVS.

Lemma 1. *Let $<$ be a well–founded relation on T. Then $<_1$ is a well–founded relation on $\mathcal{M}(T)$.*

```
wf_less_1: THEOREM well_founded?[finite_bag[T]](less_1)
```

We would like to note that to prove this lemma, we do an extensive use of different induction schemes. First, using Theorem 1, it is sufficient to prove that $\mathcal{M}(T) \subseteq W(\mathcal{M}(T), <_1)$. The PVS proof is carried out by induction on finite multisets, according to the following scheme:

$$\frac{P(\emptyset) \wedge (\forall a)(\forall M)[P(M) \rightarrow P(M \uplus \{a\})]}{(\forall M)P(M)}$$

where the predicate $P(M)$ stands for $M \in W(\mathcal{M}(T), <_1)$. Therefore, we have to prove:

1. $\emptyset \in W(\mathcal{M}(T), <_1)$ (which is true by definition).
2. $(\forall a)[(\forall M)[M \in W(\mathcal{M}(T), <_1) \rightarrow M \uplus \{a\} \in W(\mathcal{M}(T), <_1)]]$.
   This result is proved by well–founded induction on $<$, with the following predicate $P(a)$:

$$(\forall M)[M \in W(\mathcal{M}(T), <_1) \rightarrow M \uplus \{a\} \in W(\mathcal{M}(T), <_1)]$$

With this, the proof is reduced to prove $(\forall b)[b < a \rightarrow P(b)] \rightarrow P(a)$, or equivalently, to prove that if

$$(\forall b)[b < a \rightarrow (\forall M)[M \in W(\mathcal{M}(T), <_1) \rightarrow M \uplus \{b\} \in W(\mathcal{M}(T), <_1)]]$$

then

$$(\forall M)[M \in W(\mathcal{M}(T), <_1) \rightarrow M \uplus \{a\} \in W(\mathcal{M}(T), <_1)] \qquad (1)$$

Finally, we prove (1) by using the weak induction axiom for the well–founded part $W(\mathcal{M}(T), <_1)$, applied to the predicate $Q(M)$:

$$M \uplus \{a\} \in W(\mathcal{M}(T), <_1) \vee M \notin W(\mathcal{M}(T), <_1).$$

Then, as consequence of Lemma 1 and Theorem 2, we obtain the main theorem of this section:

Theorem 3 (Dershowitz and Manna). *Let $<$ be a transitive and well–founded relation on T. Then the relation $<_{mult}$ is a well–founded relation on $\mathcal{M}(T)$.*

```
less_mult_is_wf: THEOREM
 transitive?[T](<) IMPLIES well_founded?[finite_bag[T]](less_mult)
```

A sketch of the proof is as follows. First, we show that the relation $<_{mult}$ is contained in $<_1^+$. Furthermore, if the relation $<$ is transitive, then $<_{mult}$ is transitive and contains the relation $<_1^+$. Therefore, if $<$ is transitive, $<_1^+ = <_{mult}$ and by Lemma 1 and Theorem 2, we conclude that $<_{mult}$ is well–founded.

3 Methodology for proving termination with multiset orderings

In order to define a recursive function in PVS with domain D and range R, a measure function must be provided, along with an optional well-founded relation. The measure should be a function whose signature matches that of the recursive function, but with range type the domain of the order, which defaults to $<$ on $\mathbb{N}$ or on the ordinals. If an ordering $<_D$ is provided, then it must be a binary well-founded relation. Thus, when a recursive function is defined in PVS by

```
f(x:D): RECURSIVE R =
 ...
 MEASURE x BY less
```

several proof obligations (called TCCs) are generated to prove that $<_D$ is well-founded and to prove that the arguments of f decrease with respect to $<_D$ in each recursive call.

According to the work of Dershowitz and Manna [4], multiset orders can be used to prove the termination of a recursive function f in the following way: a measure function should be defined mapping each element of the domain D to a finite multiset over a well-founded set $(T, <_T)$, and a relation on D should

be considered such that two elements are related if and only if their measures are related with respect to the multiset relation induced on $M(T)$ by $<_T$. The well-founded set $(T, <_T)$ will depend on each specific function f.

For the purpose of formalizing this kind of proofs in PVS, we have proved (as it is shown in section 2) the well-foundedness of the multiset relation induced by a well-founded relation, in a way that we can easily use it for any relation, simply by instantiating the parameters of the PVS theory. Previously, and using the definition of well-foundedness based on the notion of minimal element, we have proved in PVS the well-foundedness of every relation which can be embedded in a well-founded relation by a monotonic function.

Lemma 2. *If $f : (T, <) \to (T', <')$ is monotone and $(T', <')$ is well–founded, then $(T, <)$ is well–founded.*

Thus, the idea presented above can be clarified in the following methodology: given a recursive function $f : D \to R$

Step 1 Consider an appropiate set T and build a well-founded relation on it. If T is the set of natural numbers or the set of ordinals, then the relation could be the usual order $<$, whose well-foundedness property is ensured by the prelude of PVS. In other cases, we can prove that $(T, <_T)$ is well-founded by building a monotonic function over a known well-founded set $(T', <_{T'})$ and applying Lemma 2.

Step 2 Apply the Dershowitz–Manna theorem, assuring that $(\mathcal{M}(T), <_{mult})$ is well–founded. To do this in PVS, it is enough to instantiate the parameters of the PVS theory `finite_bags_order` with T and $<_T$. Then, the corresponding theorem `less_mult_is_wf` is automatically proved.

Step 3 Define an adequate measure function `f_measure` : $D \to \mathcal{M}(T)$.

Step 4 Define in D the relation $<_D$ induced by the measure function:

$$x <_D y \Leftrightarrow \texttt{f\_measure}(x) <_{mult} \texttt{f\_measure}(x)$$

It should be noted that, in this way, the measure function is monotone. Therefore, Lemma 2 proof automatiically that $<_D$ is a well–founded relation.

Step 5 Use the relation $<_D$ as the well–founded relation needed to prove the termination of function f.

It should be noted that steps 2 and 4 can be considered totally mechanized, since it is only necessary to instantiate the parameters of a PVS theory. Nevertheless, steps 1, 3 and 5 are specific to each function.

4 Case studies

In this section, we present some examples in which the suggested methodology has been used in PVS to prove non–trivial termination properties.

4.1 Ackermann's function

Ackermann's function is defined recursively in PVS as follows

```
A(m,n): RECURSIVE nat =
 IF m = 0 THEN  n+1 ELSIF n = 0 THEN A(m-1,1)
                                 ELSE  A(m-1, A(m,n-1))
 ENDIF MEASURE lex2
```

Termination of this function is assured by the lexicographic order in $\mathbb{N} \times \mathbb{N}$. On the other hand, a tail recursive function which computes Ackermann's function is $\texttt{A\_it}(m,n) = \texttt{A\_it\_aux}((m),n)$, where

$$\texttt{A\_it\_aux}(S,z) = \begin{cases} z & \text{if } S = () \\ \texttt{A\_it\_aux}((s_2,\ldots,s_k),z+1) & \text{if } S = (0,s_2,\ldots,s_k) \\ \texttt{A\_it\_aux}((s_1-1,s_2,\ldots,s_k),1) & \text{if } s_1 \neq 0 \wedge z = 0 \\ \texttt{A\_it\_aux}((s_1,s_1-1,s_2,\ldots,s_k),z) & \text{in other case} \end{cases}$$

where $S = (s_1,\ldots,s_k)$ is a stack such that in every step

$$\texttt{A\_it\_aux}(S,z) = \texttt{A}(s_k,\texttt{A}(s_{k-1},\ldots,\texttt{A}(s_1,z)))$$

The PVS specification of this function is the following

```
A_it_aux (p): RECURSIVE nat =
 LET S = proj_1(p), z = proj_2(p) IN
 IF null?(S) THEN z
 ELSE LET s1 = car(S), S2 = cdr(S) IN
      IF s1 = 0 THEN A_it_aux ((S2,z+1))
                ELSIF z = 0 THEN A_it_aux ((cons (s1-1,S2),1))
                            ELSE A_it_aux ((cons(s1,cons(s1-1,S2)),z-1))
      ENDIF
 ENDIF MEASURE p BY less_measure

A_it(m,n): nat = A_it_aux(((: m :), n))
```

Let us note that, in this specification, the relation `less_measure` is not yet determined. In [4], a proof of termination of this function using a multiset measure has been shown. In this case, the measure function maps a pair $((s_1,\ldots,s_k),z)$ into the multiset of pairs of natural numbers $\dot{\{}(s_1,z),(s_2+1,0),\ldots,(s_k+1,0)\dot{\}}$. Note that in each case of the conditional, one or two pairs of the multiset are removed, and they are replaced by smaller pairs with respect to the lexicographic order in $\mathbb{N} \times \mathbb{N}$.

In order to build in PVS the appropriate relation `less_measure` and prove its properties, we carry out the following steps, according to the explained methodology.

Step 1 In this case, $T = \mathbb{N} \times \mathbb{N}$ and $<_T$ is the lexicographic order. We prove that it is a well–founded relation by defining a monotone function into ordinals.

```
well_founded_lex: LEMMA well_founded?[[nat,nat]](lex)
```

Step 2 We prove that the extension to $\mathcal{M}(\mathbb{N}\times\mathbb{N})$ of the lexicographic order on $\mathbb{N}\times\mathbb{N}$ is a well–founded ordering, by instantiting the parameters of the main theory by $\mathbb{N}\times\mathbb{N}$ and *lex*.

```
IMPORTING finite_bags_order[[nat,nat], lex]

well_founded_bags_pair_mult: COROLLARY
  well_founded?[finite_bag[[nat,nat]]](less_mult)
```

Step 3 We define the *measure function* from the pairs (S, z) in $\mathcal{M}(\mathbb{N}\times\mathbb{N})$

```
a_measure(p): finite_bag[[nat,nat]] =
  IF null?(S) THEN  emptybag
              ELSIF length(S)=1 THEN insert((car(S),z), emptybag)
              ELSE  insert((car(S),z), list_mult(cdr(S)))
  ENDIF
  WHERE S = proj_1(p), z = proj_2(p)
```

Step 4 From it, we have directly proved, by Lemma 2, that the ordering between the arguments of the function induced by this function, is well–founded

```
a_less(p1,p2): bool = less_mult(a_measure(p1),a_measure(p2))

a_less_wf: COROLLARY well_founded?[[list[nat],nat]](a_less)
```

Step 5 So, this is the function we need as well-founded order in the definition of `A_it_aux`. We prove the proof obligations automatically generated by PVS to ensure that the arguments of the function decrease in each recursive call:

$A\_it\_aux\_TCC2$: $a\_less(((s_2,\ldots,s_k),z+1),((s_1,\ldots,s_k),z))$
$A\_it\_aux\_TCC4$: $a\_less(((s_1-1,s_2,\ldots,s_k),1),((s_1,\ldots,s_k),z))$
$A\_it\_aux\_TCC6$: $a\_less(((s_1,s_1{-}1,s_2,\ldots,s_k),z{-}1),((s_1,\ldots,s_k),z))$

Finally, we prove in PVS that both functions are the same

```
a_it_eq_a: THEOREM A_it(m,n) = A(m,n)
```

4.2 McCarthy's 91 function

McCarthy's 91 function is a recursive function defined by John McCarthy in [7] as

$$M(n) = \begin{cases} n-10, & \text{if } n > 100 \\ M(M(n+11)), & \text{if } n \leq 100 \end{cases}$$

This function returns 91 for all $n \leq 101$ and $n-10$ for $n > 101$. There are some papers [4, 3, 8] that address termination proofs of this function. Our goal here is to show how we use the methodology of multiset orderings in PVS to prove termination of an iterative version of McCarthy's 91 function. We follow the same steps as for Ackermann's function. First, we specify the function in PVS in three ways

```
mc (x): nat = IF x > 100 THEN x - 10 ELSE 91 ENDIF

mc_original (x): RECURSIVE nat =
   IF x > 100 THEN x - 10 ELSE mc(mc(x+11)) ENDIF
   MEASURE x

mc_it_aux(p): RECURSIVE nat =
  LET n = proj_1(p), z = proj_2(p) IN
   IF n = 0 THEN z
            ELSIF z > 100 THEN mc_it_aux(n-1,z-10)
                          ELSE mc_it_aux(n+1,z+11)
   ENDIF MEASURE p BY less_measure

mc_it (x) :nat = mc_it_aux((1, x))
```

We will consider the termination problem of the function `mc_it_aux`. This task is not trivial due to the behavior of the second recursive call. In [4] a multiset measure is given to ensure termination of this function: every pair (n, z) is measured by the finite multiset $\dot{\{}z, mc(z), mc^2(z), \ldots, mc^{n-1}(z)\dot{\}}$. Let us note that `mc_it_aux`$(n, z) = mc^n(z)$, and that the relation to compare multisets is the multiset relation induced by the following well–founded relation in $\mathbb{N}$:

$$m <_{mc} n \Leftrightarrow n < m \leq 111$$

In the sequel, we describe how to prove termination of this function in PVS following the proposed methodology:

Step 1 We define in $\mathbb{N}$ the relation $m <_{mc} n = n < m \wedge m <= 111$ and we prove that $(\mathbb{N}, <_{mc})$ is well–founded, using Lemma 2.

```
mc_less (m,n): bool = n < m & m <= 111

f_mc_less(n): nat = IF n <= 111 THEN 111-n ELSE 0 ENDIF

f_mc_less_monotone: LEMMA
  mc_less(m,n) IMPLIES f_mc_less(m) < f_mc_less(n)

mc_less_well_founded: LEMMA well_founded?(mc_less)
```

Step 2 We prove that the multiset relation induced in $\mathcal{M}(N)$ by $<_{mc}$ is well-founded, by instantiating the parameters of the PVS theory `finite_bags_order` with $\mathbb{N}$ and $<_{mc}$

```
IMPORTING finite_bags_order[nat, mc_less]

less_mult_mc_less_is_wf: COROLLARY
  well_founded?[finite_bag[nat]](less_mult)
```

Step 3 We define the *measure function* from $\mathbb{N} \times \mathbb{N}$ to $\mathcal{M}(\mathbb{N})$

```
mc_it_measure(p): RECURSIVE finite_bag[nat] =
 LET n = proj_1(p), z = proj_2(p) IN
```

```
IF n = 0 THEN emptybag
         ELSIF n = 1
         THEN insert(z, emptybag)
         ELSE insert(iterate(mc,n-1)(z), mc_it_measure((n-1,z)))
ENDIF MEASURE proj_1(p)
```

Step 4 We define in $\mathbb{N} \times \mathbb{N}$ the order induced by this measure function. Then, we have automatically proved that it is a well–founded relation

```
mc_it_less(p1,p2): bool = less_mult(mc_it_measure(p1),mc_it_measure(p2))

mc_it_less_wf: THEOREM well_founded?[[nat,nat]](mc_it_less)
```

Step 5 We use the relation `mc_it_less` as the well-founded relation needed to ensure the termination of the function `mc_it_aux` and we prove the proofs obligations generated to ensure that the arguments of the function decrease in each recursive call.

Finally, we can prove in PVS the equivalence between both definitions

```
mc_it_equal_mc: THEOREM mc_it(n) = mc(n)
```

4.3 Program schema for double recursion

Let us consider the following schema for functions which are defined by double recursion. Let T a set and $<$ a well–founded relation in T. Let p be a predicate on T, $g, k, l : T \to T$ functions and h a binary operation with the associative, commutative properties, and an identity element ($e \in T$). We also assume that the property

$$\forall x(\neg p(x) \Rightarrow k(x) < x \wedge l(x) < x) \tag{1}$$

is satisfied. Then, we define the following schema of functions:

$$F(x) = \begin{cases} g(x) & \text{if } p(x) \\ h(F(k(x)), F(l(x))) & \text{if } \neg p(x) \end{cases}$$

It should be noted that by instantiating the predicate variable p and the variables g, k, l, h and e, we obtain an instance of this schema that define some particular function. Termination of F is assured by the property (1).

We specify in PVS the function F and a tail–recursive function defined in [4] to compute F in a more efficient way, and we will follow the suggested methodology for proving its termination. We introduce the variables over functions and relations as parameters of the theory and we specify its properties by means of PVS assumptions.

```
double_rec[T: TYPE+, <:(well_founded?[T]),p:pred[T], g,k,l:[T->T],
           h:[[T,T]->T], e:T]: THEORY
  BEGIN

 ASSUMING
   h_ax_1: ASSUMPTION commutative?(h)
   h_ax_2: ASSUMPTION associative?(h)
   h_ax_3: ASSUMPTION identity?(h)(e)
```

```
   p_ax_1: ASSUMPTION FORALL (x:T): not p(x) IMPLIES k(x) < x AND l(x) < x
   ax_4: ASSUMPTION transitive?[T](<)
  ENDASSUMING

f(x) : RECURSIVE T =
  IF p(x) THEN g(x) ELSE h(f(k(x)), f(l(x))) ENDIF
  MEASURE x BY <

f_it_aux(p1): RECURSIVE T =
  LET z = proj_1(p1), l = proj_2(p1) IN
   IF null?(l) THEN z
   ELSE LET s1 = car(l), ls = cdr(l) IN
            IF p(s1) THEN f_it_aux ((h(g(s1),z),ls))
                     ELSE f_it_aux ((z, cons(k(s1),cons(l(s1),ls))))
            ENDIF
   ENDIF MEASURE p1 by f_it_less

f_it(x): T = f_it_aux((e, (: x :)))
```

where the relation `f_it_less` must be a well–founded relation and the arguments of `f_it_aux` must decrease according to `f_it_less` in every recursive call. For constructing this relation and proving its properties in PVS we have followed the steps of the proposed methodology. In this case, the measure of the arguments $(z, (s_1, \ldots, s_k))$ is just the multiset $\dot{\{}s_1, \ldots, s_k\dot{\}}$. Note that, in each recursive call, the element s_1 is deleted or replaced by the minor elements $k(s_1), l(s_1)$.

Step 1 In this case, it suffices to consider $(T, <)$.

Step 2 We prove that the extension to $\mathcal{M}(T)$ of $<$ is a well–founded ordering, by instantiating the parameters of the theory `finite_bags_order` by T and $<$.

```
IMPORTING finite_bags_order[T,<]

less_mult_wf: COROLLARY well_founded?[finite_bag[T]](less_mult)
```

Step 3 We define the *measure function* from the pairs (z, S) into $\mathcal{M}(\mathcal{T})$

```
p1, p2: VAR [T,list[T]]

list2bag(l) : RECURSIVE finite_bag[T] =
 IF null?(l) THEN emptybag ELSE insert(car(l), list2bag(cdr(l)))
 ENDIF MEASURE length(l)

f_it_measure(p1): finite_bag[T] = list2bag(proj_2(p1))
```

Step 4 We define in $T \times \texttt{list}[T]$ the relation induced by the measure function. Then, by Lemma 2, we have proved that it is well–founded

```
 f_it_less(p1,p2): bool = less_mult(f_it_measure(p1),f_it_measure(p2))

 f_it_less_wf: THEOREM  well_founded?(f_it_less)
```

Step 5 We use the `f_it_less` as the termination measure of function `f_it_aux` and we prove the obligation proofs automatically generated by PVS to assure that the arguments of the function decrease in each recursive call.

Finally, we prove that both functions are the same

```
f_it_eq_f: THEOREM f(x) = f_it(x)
```

Note that the Fibonacci function can be defined as a particular case of this schema. We show here this specification and some results of computing it through the PVSio

```
 Fib(n):nat = f_it[nat,<,p,g,k,l,+,0](n)

<PVSio> Fib(20); ==> 6765 ; cpu time 10 msec user, 0 msec system

<PVSio> Fib(30); ==> 832040 ; cpu time 520 msec user, 0 msec system
```

where $p(x) \equiv x \leq 1$, $g(x) = x$, $k(x) = max(0, x-1)$ y $k(x) = max(0, x-2)$.

5 Conclusions and future work

We have presented a formalization of multiset relations in the PVS system and a methodology for proving non–trivial termination properties of recursive functions using multiset orderings in PVS. First, we have proved that the definition of well–foundedness based on the well–founded part of a relation is equivalent to this one based on the minimal element. Then, we have defined the multiset relation induced by a given relation and proved the theorem which establishes well–foundedness of the multiset relation that extends a well–founded relation.

We have also presented a methodology to organize and simplify termination proofs which use well–founded multiset orderings. The main utility of this methodology is given by the easy way to prove the well–foundedness of a multiset relation. It is enough to instantiate the parameters of a PVS theory and, automatically, a corollary with the expected result is obtained. The non–mechanized part is to define the measure function, that it is specific for each function, and to prove that the arguments of the function decrease in each recursive call. In [2] we have used this methodology to prove in PVS the termination of a tableau algorithm for the $\mathcal{ALC}$ logic. This measure function is more complex than the previous ones, since in this case, when a kind of rule (universal rules) is applied, this one is not disabled forever, but it can be reapplied due to the introduction of new individuals by subsequent applications of another kind of rule (existential rules).

Finally, we would like to point out two lines for future work. First, in order to make the methodology more automatic, we would like to develop PVS strategies to increase the mechanization of the process. Second, we would like to apply this methodology to prove termination of rewriting systems or tableau algorithms, in which a measure in multisets was required.

References

1. P. Aczel. An Introduction to Inductive Definitions. In J. Barwise, editor, *Handbook of Mathematical Logic*, pages 739–782. North–Holland Publishing Company, 1977.
2. J. A. Alonso, J. Borrego, M. J. Hidalgo, F. J. Martín-Mateos, and J. L. Ruiz-Reina. A Formally Verified Prover for the ALC Description Logic. In *Theorem Proving in Higher Order*, volume 4732 of *Lecture Notes in Computer Science*, pages 135–150. Springer, 2007.
3. J. Cowles. Knuth's Generalization of McCarthy's 91 Function. In *Computer–Aided reasoning: ACL2 case studies*, pages 283–299. Kluwer Academic Publishers, 2000.
4. N. Dershowitz and Z. Manna. Proving Termination with Multiset Orderings. *Communications of the ACM*, 22(8):465–476, 1979.
5. R. W. Floyd. Assigning Meanings to Programs. *Mathematical Aspects of Computer Science*, 19:19–31, 1967.
6. M. J., J. A. Alonso, F. J. Martín-Mateos, and J. L. Ruiz-Reina. Constructing Formally Verified Reasoners for the ALC Description Logic. *Electr. Notes Theor. Comput. Sci*, 200(3):87–102, 2008.
7. Z. Manna and J. McCarthy. Properties of Programs and Partial Function Logic. *Machine Intelligence*, 5:27–37, 1970.
8. F. J. Martín-Mateos, J. A. Alonso, M. J. Hidalgo, and J. L. Ruiz-Reina. A Generic Instantiation Tool and a Case Study: A Generic Multiset Theory. In *Third International Workshop on the ACL2 Theorem Prover and its Applications (ACL2 '02)*, April 2002.
9. F. J. Martín-Mateos, J. A. Alonso, M. Hidalgo, and J. L. Ruiz-Reina. A Formal Proof of Dickson's Lemma in ACL2. In *Logic for Programming, Artificial Intelligence, and Reasoning*, volume 2850 of *Lecture Notes in Computer Science*, pages 49–58. Springer, 2003.
10. F. J. Martín-Mateos, J. A. Alonso, M. J. Hidalgo, and J. L. Ruiz-Reina. Verifying an Applicative ATP Using Multiset Relations. In *Computer Aided Systems Theory*, volume 2178 of *Lecture Notes in Computer Science*, pages 616–626, 2001.
11. F. J. Martín-Mateos, J. L. Ruiz, J. A. Alonso, and M. J. Hidalgo. Proof Pearl: A Formal Proof of Higman's Lemma in ACL2. In *Higher Order Logic Theorem Proving and Its Applications*, volume 3603 of *Lecture Notes in Computer Science*, pages 368–372. Springer, 2005.
12. T. Nipkow. An Inductive Proof of the Wellfoundedness of the Multiset Order. (Available on the Web at `http://www4.informatik.tu-muenchen.de/~nipkow/misc/multiset.ps`), 1998. A proof due to W. Buchholz.
13. S. Owre, N. Shankar, J. M. Rushby, and D. W. J. Stringer-Calvert. PVS Language Reference. Technical report, Computer Science Laboratory, SRI International, 1999.
14. P. Rudnicki and A. Trybulec. On Equivalents of Well–Foundedness. *Journal of Automated Reasoning*, 23(3):197–234, 1999.
15. J. L. Ruiz, J. A. Alonso, M. J. Hidalgo, and F. J. Martín. Formal Proofs About Rewriting Using ACL2. *Ann. Math. Artif. Intell*, 36(3):239–262, 2002.
16. J. L. Ruiz, J. A. Alonso, M. J. Hidalgo, and F. J. Martín. Termination in ACL2 Using Multiset Relation. In F. D. Kamareddine, editor, *Thirty Five Years of Automating Mathematics*, pages 217–245. Kluwer Academic Publishers, 2003.