FORMALIZED MATHEMATICS
Vol. 34, No. 1, pp. 111–121, 2026
DOI: 10.2478/forma-2026-0008
e-ISSN: 1898–9934

Paradigm
reference-global.com

The Ackermann Function and Primitive Recursive Functions

Artur Korniłowicz
Faculty of Computer Science
University of Białystok
Poland

Karol Pąk
Faculty of Computer Science
University of Białystok
Poland

Summary. This paper formalizes the classical proof that the Ackermann function is not primitive recursive. The intermediate results establish the recursive equations, monotonicity, and growth estimates needed to show that every primitive recursive function is majorized by a suitable level of the Ackermann function. The final diagonal argument then proves that the binary Ackermann function cannot itself be primitive recursive.

MSC: 68V20

Keywords: Ackermann function; primitive recursive function; majorization;

MML identifier: `ACKERMAN`, version: `8.1.15 5.105.1515`

Introduction

The Ackermann function is a classical example separating total computability from primitive recursiveness [1], [7]. Later presentations usually place the argument inside a hierarchy of fast-growing functions [11]. Our formalization follows the same mathematical idea, but packages it as a concrete majorization class: the functions bounded by a suitable level of the binary Ackermann function. After proving that this class contains the primitive recursive functions, the final diagonal step shows that the binary Ackermann function cannot itself be primitive recursive.

The Ackermann function and computability theory have also been formalized in proof assistants such as Isabelle/HOL and Lean [8], [10]. Our contribution is complementary to these developments: the emphasis is not computation of

© 2026 The Author(s) / AMU
(Association of Mizar Users)
under CC BY-SA 4.0 license

values or a general computability infrastructure, but the classical separation theorem itself. The formal work is therefore organized around the majorization invariant that survives constants, successor functions, projections, composition, and primitive recursion.

A point that is almost invisible in informal presentations is the construction of the Ackermann function itself. The displayed recursion does not immediately define a Mizar object: the value $\mathcal{A}(m, n)$ depends on the whole preceding row and, within a row, on the previously computed value in the same row. In the formal proof of existence we therefore first construct, for each finite prefix of rows, an operator that appends the next row. This appended row is itself obtained by ordinary recursion on n. These local constructions are then assembled into a sequence of finite approximations to the Ackermann table, and only at the end is the required binary operation on $\mathbb{N}$ extracted from the diagonal entries of these approximations. Technically, this uses several Mizar schemes for recursion and function construction, including recursion over natural numbers, choice of functions with prescribed values, assembly of indexed families of such functions, and finally the binary-function existence scheme. Thus even before proving that Ackermann's function is not primitive recursive, the article formalizes the nontrivial fact that the recursive equations determine an actual binary operation on natural numbers.

In Section 1, we collect the elementary facts about iterated exponentiation needed later to identify the first levels of the Ackermann function. These facts are mostly computational, but they give useful concrete landmarks such as the formula $\mathcal{A}(4, n) = 2 \uparrow\uparrow (n + 3) - 3$ (Theorem

Section 2 introduces the binary Ackermann function and develops its growth properties. The first task in this section is the existence and uniqueness of the function satisfying the Ackermann equations; this is formalized by building finite approximations to the table and then extracting the binary operation. After that, the central results are monotonicity in both arguments and the estimates that allow nested Ackermann calls, sums, and doubled arguments to be absorbed into a higher level of the same function. In particular, we prove $\mathcal{A}(k, \mathcal{A}(m, n)) < \mathcal{A}(k + m + 2, n)$ (Theorem

Section 3 provides the technical interface between ordinary operations and the Mizar representation of primitive recursive functions. Since primitive recursive functions in the library are functions on finite tuples, unary and binary operations have to be converted to tuple-valued domains.

Section 4 formalizes the levels of the Ackermann function as unary functions and shows that each fixed level is primitive recursive. The key result is Theorem

In the final section, we introduce the majorization relation and the class of functions majorized by the Ackermann function. First we prove the diagonal

observation that no binary operation majorizes its own tuple representation (Theorem

1. PRELIMINARIES

From now on D denotes a set and a, b, i, k, m, n, n_1, n_2 denote natural numbers.

Now we state the proposition:

(1) $m \uparrow\uparrow (n+1) = m^{m \uparrow\uparrow n}$.

Let us consider m. Observe that $m \uparrow\uparrow 1$ reduces to m.

Now we state the propositions:

(2) $m \uparrow\uparrow 2 = m^m$.

(3) $m \uparrow\uparrow 3 = m^{m^m}$.

(4) $m \uparrow\uparrow 4 = m^{m^{m^m}}$.

(5) $m \uparrow\uparrow 5 = m^{m^{m^{m^m}}}$.

2. ACKERMANN FUNCTION

The functor AckermannFunction yielding a binary operation on $\mathbb{N}$ is defined by

(Def. 1) for every natural numbers m, n, $it(0,n) = n+1$ and if $m > 0$, then $it(m,0) = it(m-1,1)$ and if $m > 0$ and $n > 0$, then $it(m,n) = it(m-1, it(m,n-1))$.

Let m, n be natural numbers. The functor Ackermann(m,n) yielding an element of $\mathbb{N}$ is defined by the term

(Def. 2) (AckermannFunction)(m,n).

Now we state the propositions:

(6) Ackermann$(m+1,0) =$ Ackermann$(m,1)$.

(7) Ackermann$(m+1,n+1) =$ Ackermann$(m,$ Ackermann$(m+1,n))$.

(8) Ackermann$(0,n) = n+1$.

(9) Ackermann$(1,n) = n+2$.

PROOF: Define $\mathcal{P}$[natural number] $\equiv \mathcal{A}(1,\$_1) = \$_1 + 2$. $\mathcal{P}[0]$. For every k such that $\mathcal{P}[k]$ holds $\mathcal{P}[k+1]$. For every k, $\mathcal{P}[k]$ from [2, Sch. 2]. □

(10) Ackermann$(2,n) = 2 \cdot n + 3$.

PROOF: Define $\mathcal{P}$[natural number] $\equiv \mathcal{A}(2,\$_1) = 2 \cdot \$_1 + 3$. $\mathcal{P}[0]$. For every k such that $\mathcal{P}[k]$ holds $\mathcal{P}[k+1]$. For every k, $\mathcal{P}[k]$ from [2, Sch. 2]. □

(11) $\mathrm{Ackermann}(3, n) = 2^{n+3} - 3$.
PROOF: Define $\mathcal{P}$[natural number] $\equiv \mathcal{A}(3, \$_1) = 2^{\$_1+3} - 3$. $\mathcal{P}[0]$. For every k such that $\mathcal{P}[k]$ holds $\mathcal{P}[k+1]$ by (10), [9, (6)]. For every k, $\mathcal{P}[k]$ from [2, Sch. 2]. □

(12) $\mathrm{Ackermann}(4, n) = 2 \uparrow\uparrow (n + 3) - 3$.
PROOF: Define $\mathcal{P}$[natural number] $\equiv \mathcal{A}(4, \$_1) = 2 \uparrow\uparrow (\$_1 + 3) - 3$. $\mathcal{P}[0]$ by [3, (19)], (11). For every k such that $\mathcal{P}[k]$ holds $\mathcal{P}[k + 1]$ by [2, (38)], [3, (14)], (11). For every k, $\mathcal{P}[k]$ from [2, Sch. 2]. □

(13) $\mathrm{Ackermann}(4, 0) = 13$. The theorem is a consequence of (3) and (12).

(14) $\mathrm{Ackermann}(4, 1) = 65533$. The theorem is a consequence of (4) and (12).

(15) $\mathrm{Ackermann}(4, 2) = 2^{65536} - 3$. The theorem is a consequence of (5) and (12).

(16) $n < \mathrm{Ackermann}(m, n)$.
PROOF: Define $\mathcal{M}$[natural number] $\equiv$ for every n, $n < \mathcal{A}(\$_1, n)$. $\mathcal{M}[0]$ by [2, (16)]. For every m such that $\mathcal{M}[m]$ holds $\mathcal{M}[m + 1]$ by (6), [2, (13)], (7). For every k, $\mathcal{M}[k]$ from [2, Sch. 2]. □

(17) $\mathrm{Ackermann}(m, n) < \mathrm{Ackermann}(m, n + 1)$.
PROOF: Define $\mathcal{P}$[natural number] $\equiv \mathcal{A}(\$_1, n) < \mathcal{A}(\$_1, n + 1)$. $\mathcal{P}[0]$ by [2, (16)]. For every k such that $\mathcal{P}[k]$ holds $\mathcal{P}[k + 1]$. For every k, $\mathcal{P}[k]$ from [2, Sch. 2]. □

(18) If $a < b$, then $\mathrm{Ackermann}(m, a) < \mathrm{Ackermann}(m, b)$.
PROOF: Define $\mathcal{P}$[natural number] $\equiv$ for every a such that $a < \$_1$ holds $\mathcal{A}(m, a) < \mathcal{A}(m, \$_1)$. For every k such that $\mathcal{P}[k]$ holds $\mathcal{P}[k+1]$ by [2, (13)], (17). For every k, $\mathcal{P}[k]$ from [2, Sch. 2]. □

(19) If $a \leqslant b$, then $\mathrm{Ackermann}(m, a) \leqslant \mathrm{Ackermann}(m, b)$. The theorem is a consequence of (18).

(20) $m < \mathrm{Ackermann}(m, n)$.
PROOF: Define $\mathcal{M}$[natural number] $\equiv$ for every m, $m < \mathcal{A}(m, \$_1)$. $\mathcal{M}[0]$ by [2, (13)], (17), (6). For every n such that $\mathcal{M}[n]$ holds $\mathcal{M}[n + 1]$. For every k, $\mathcal{M}[k]$ from [2, Sch. 2]. □

(21) $\mathrm{Ackermann}(m, n + 1) \leqslant \mathrm{Ackermann}(m + 1, n)$.
PROOF: Define $\mathcal{P}$[natural number] $\equiv$ for every m, $\mathcal{A}(m, \$_1 + 1) \leqslant \mathcal{A}(m + 1, \$_1)$. $\mathcal{P}[0]$. For every k such that $\mathcal{P}[k]$ holds $\mathcal{P}[k + 1]$ by (16), [2, (13)], (19), (7). For every k, $\mathcal{P}[k]$ from [2, Sch. 2]. □

(22) $\mathrm{Ackermann}(m, n) < \mathrm{Ackermann}(m + 1, n)$.
PROOF: Define $\mathcal{P}$[natural number] $\equiv$ for every n, $\mathcal{A}(\$_1, n) < \mathcal{A}(\$_1 + 1, n)$. $\mathcal{P}[0]$. For every k such that $\mathcal{P}[k]$ holds $\mathcal{P}[k+1]$. For every k, $\mathcal{P}[k]$ from [2, Sch. 2]. □

(23) If $a < b$, then Ackermann$(a, n) <$ Ackermann(b, n).
PROOF: Define $\mathcal{P}$[natural number] $\equiv$ for every a such that $a < \$_1$ holds $\mathcal{A}(a, n) < \mathcal{A}(\$_1, n)$. For every k such that $\mathcal{P}[k]$ holds $\mathcal{P}[k+1]$ by [2, (13)], (22). For every k, $\mathcal{P}[k]$ from [2, Sch. 2]. □

(24) If $a \leqslant b$, then Ackermann$(a, n) \leqslant$ Ackermann(b, n). The theorem is a consequence of (23).

(25) Ackermann$(k,$ Ackermann$(m, n)) <$ Ackermann$(k + m + 2, n)$. The theorem is a consequence of (24), (23), (18), (7), and (21).

(26) Let us consider natural numbers r, s, y. Then Ackermann(r, y)+Ackermann$(s, y) <$ Ackermann$(4 + \max(r, s), y)$. The theorem is a consequence of (24), (10), and (25).

(27) Let us consider natural numbers r, n. Then Ackermann$(r, n + n) <$ Ackermann$(r + 4, n)$.
PROOF: $n + n < \mathcal{A}(2, n)$ by (10), [2, (16)]. $\mathcal{A}(r, n + n) < \mathcal{A}(r, \mathcal{A}(2, n))$. $\mathcal{A}(r, n + n) < \mathcal{A}(r + 2 + 2, n)$. □

Let us consider m and n. One can verify that Ackermann(m, n) is non zero.

The functor AckermannFunction(m) yielding a unary operation on $\mathbb{N}$ is defined by

(Def. 3) for every natural number n, $it(n) =$ Ackermann(m, n).

Observe that AckermannFunction(m) is positive yielding and AckermannFunction(m) is increasing.

Now we state the propositions:

(28) (AckermannFunction$(m + 1)$)$(0) =$ (AckermannFunction(m))(1). The theorem is a consequence of (6).

(29) (AckermannFunction$(m+1)$)$(n+1) =$ (AckermannFunction(m))((AckermannFunc 1))(n)). The theorem is a consequence of (7).

3. AUXILIARY OPERATIONS

Let f be a function. The functor dom2tuple(f) yielding a function is defined by

(Def. 4) (i) dom $it =$ dom f^1 and for every object x such that $x \in$ dom f holds $it(\langle x \rangle) = f(x)$, **if** f is not empty,

(ii) $it = \emptyset$, **otherwise**.

Let f be an empty function. One can check that dom2tuple(f) is empty.

Let f be a non empty function. One can check that dom2tuple(f) is non empty.

Let D be a set and f be a D-defined function. One can check that dom2tuple(f) is (D^*)-defined.

Let f be a D-valued function. One can verify that dom2tuple(f) is D-valued.

Let D be a non empty set and f be a D-defined function. Observe that dom2tuple(f) is homogeneous.

Now we state the proposition:

(30) Let us consider a unary operation f on D. Then dom2tuple(f) is a partial function from D^* to D.

Let D be a non empty set and f be a unary operation on D. Let us observe that dom2tuple(f) is 1-ary.

Let D be a set and f be a binary operation on D. The functor dom2tuple(f) yielding a function is defined by

(Def. 5) (i) $\operatorname{dom} it = D^2$ and for every objects x, y such that $\langle x, y\rangle \in \operatorname{dom} f$ holds $it(\langle x, y\rangle) = f(x, y)$, **if** D is not empty,

(ii) $it = \emptyset$, **otherwise**.

Let f be a binary operation on $\mathbb{N}$. Observe that dom2tuple(f) is $(\mathbb{N}^*)$-defined.

Let D be a set and f be a binary operation on D. One can verify that dom2tuple(f) is D-valued.

Now we state the proposition:

(31) Let us consider a binary operation f on D. Then dom2tuple(f) is a partial function from D^* to D.

Let D be a non empty set and f be a binary operation on D. Observe that dom2tuple(f) is homogeneous and dom2tuple(f) is 2-ary.

Now we state the proposition:

(32) dom2tuple(AckermannFunction) $\in$ HFuncs $\mathbb{N}$.

4. Primitive Recursiveness

Let f be a function yielding many sorted set indexed by $\mathbb{N}$. We say that f is homogeneous if and only if

(Def. 6) for every natural number n, $f(n)$ is homogeneous.

We say that $f$into $\mathbb{N}$ if and only if

(Def. 7) for every natural number n, $f(n)$ inherits into $\mathbb{N}$.

We say that f is primitive recursive if and only if

(Def. 8) for every natural number n, $f(n)$ is primitive recursive.

Let D be a set. We say that f is D-defined if and only if

(Def. 9) for every natural number n, $f(n)$ is D-defined.

We say that f is satisfying primrec conditions if and only if

(Def. 10) f is homogeneous and $(\mathbb{N}^*)$-defined.

Let f be a homogeneous function. One can verify that $\mathbb{N} \longmapsto f$ is homogeneous.

Let f be a function. One can check that $\mathbb{N} \longmapsto f$.

Let f be a primitive recursive function. Note that $\mathbb{N} \longmapsto f$ is primitive recursive.

Let D be a set and f be a D-defined function. Note that $\mathbb{N} \longmapsto f$ is D-defined and there exists a function yielding many sorted set indexed by $\mathbb{N}$ which is satisfying primrec conditions and every function yielding many sorted set indexed by $\mathbb{N}$ which is satisfying primrec conditions is also homogeneous and $(\mathbb{N}^*)$-defined and every function yielding many sorted set indexed by $\mathbb{N}$ which is homogeneous and $(\mathbb{N}^*)$-defined is also satisfying primrec conditions.

Let f be a homogeneous, function yielding many sorted set indexed by $\mathbb{N}$ and n be a natural number. Note that $f(n)$ is homogeneous.

Let f be a function yielding many sorted set indexed by $\mathbb{N}$. Let us note that $f(n)$ and there exists a function yielding many sorted set indexed by $\mathbb{N}$ which is primitive recursive.

Let f be a primitive recursive, function yielding many sorted set indexed by $\mathbb{N}$ and n be a natural number. Let us observe that $f(n)$ is primitive recursive.

Let D be a set. Observe that there exists a function yielding many sorted set indexed by $\mathbb{N}$ which is D-defined.

Let f be a D-defined, function yielding many sorted set indexed by $\mathbb{N}$ and n be a natural number. Observe that $f(n)$ is D-defined.

Let k be a natural number and f be a homogeneous, function yielding many sorted set indexed by $\mathbb{N}$. We say that f is k-ary if and only if

(Def. 11) for every natural number n, $f(n)$ is k-ary.

Let k be a non zero natural number. Let us observe that there exists a homogeneous, function yielding many sorted set indexed by $\mathbb{N}$ which is k-ary, primitive recursive, and $(\mathbb{N}^*)$-defined.

Let f be a k-ary, homogeneous, function yielding many sorted set indexed by $\mathbb{N}$ and n be a natural number. Let us observe that $f(n)$ is k-ary.

Let k be a natural number. The functor k-aryPrimRecFuncs yielding a subset of HFuncs $\mathbb{N}$ is defined by the term

(Def. 12) $\{f$, where f is an element of HFuncs $\mathbb{N} : f$ is k-ary and primitive recursive$\}$.

Now we state the proposition:

(33) $\mathrm{const}_k(n) \in k$-aryPrimRecFuncs.

Let k be a natural number. Let us note that k-aryPrimRecFuncs is non empty.

Now we state the proposition:

(34) If $1 \leqslant n \leqslant k$, then $\mathrm{succ}_k(n) \in k$-aryPrimRecFuncs.

PROOF: Set $f = \mathrm{succ}_k(n)$. f is k-ary. f is primitive recursive by [5, (70)]. □

The functor AckermannSequence yielding a 1-ary, satisfying primrec conditions, primitive recursive, function yielding many sorted set indexed by $\mathbb{N}$ is defined by

(Def. 13) $it(0) = \mathrm{succ}_1(1)$ and for every natural number m, $it(m+1) = \mathrm{primrec}(\mathrm{const}_0(\mathrm{Acker}$ $(\prod^* \langle \mathrm{proj}_2(2) \rangle), 1)$.

Note that AckermannSequence is (HFuncs $\mathbb{N}$)-valued.

Now we state the propositions:

(35) $\mathrm{arity}(\mathrm{AckermannSequence})(m) \cdot (\prod^* \langle \mathrm{proj}_2(2) \rangle) = 2$.

(36) $(\mathrm{AckermannSequence})(m)(\langle n \rangle) = \mathrm{Ackermann}(m, n)$.

PROOF: Define $\mathcal{P}[\text{natural number}] \equiv$ for every n, $AS(\$_1)(\langle n \rangle) = \mathcal{A}(\$_1, n)$. $\mathcal{P}[0]$ by [6, (98)]. For every k such that $\mathcal{P}[k]$ holds $\mathcal{P}[k+1]$ by [5, (80)], [12, (7)], (6), [5, (36)]. For every k, $\mathcal{P}[k]$ from [2, Sch. 2]. □

(37) $\mathrm{dom2tuple}(\mathrm{AckermannFunction}(0)) = \mathrm{succ}_1(1)$.

(38) $\mathrm{dom2tuple}(\mathrm{AckermannFunction}(m+1)) = \mathrm{primrec}(\mathrm{const}_0(\mathrm{Ackermann}(m, 1)), (\mathrm{Ack}$ $(\prod^* \langle \mathrm{proj}_2(2) \rangle), 1)$. The theorem is a consequence of (35) and (36).

(39) $\mathrm{dom2tuple}(\mathrm{AckermannFunction}(m)) = (\mathrm{AckermannSequence})(m)$. The theorem is a consequence of (37) and (38).

Let us consider m. Let us note that $\mathrm{dom2tuple}(\mathrm{AckermannFunction}(m))$ is primitive recursive.

Let k be a natural number. Observe that there exists a homogeneous, $(\mathbb{N}^*)$-defined function which is k-ary.

Let f be a binary operation on $\mathbb{N}$ and g be a function. We say that f majorizes g if and only if

(Def. 14) there exists an element N of $\mathbb{N}$ such that for every non empty finite sequence F of elements of $\mathbb{N}$ such that $F \in \mathrm{dom}\, g$ for every element a of $\mathbb{N}$ such that $a = \max \mathrm{rng}\, F$ holds $g(F) < f(N, a)$.

Now we state the proposition:

(40) Let us consider a binary operation f on $\mathbb{N}$. Then f does not majorize $\mathrm{dom2tuple}(f)$.

Let f be a binary operation on $\mathbb{N}$. The functor majorizedBy(f) yielding a subset of HFuncs $\mathbb{N}$ is defined by the term

(Def. 15) $\{g,$ where g is a homogeneous, $(\mathbb{N}^*)$-defined function : f majorizes $g\}$.

Now we state the propositions:

(41) Let us consider a homogeneous, ($\mathbb{N}^*$)-defined function g. If arity $g = 0$, then AckermannFunction majorizes g.

(42) Let us consider a finite sequence G of elements of HFuncs $\mathbb{N}$. Suppose rng $G \subseteq$ majorizedBy(AckermannFunction). Then there exists an element N of $\mathbb{N}$ such that for every homogeneous, ($\mathbb{N}^*$)-defined function g such that $g \in \operatorname{rng} G$ for every non empty finite sequence F of elements of $\mathbb{N}$ such that $F \in \operatorname{dom} g$ for every element a of $\mathbb{N}$ such that $a = \max \operatorname{rng} F$ holds $g(F) < \text{Ackermann}(N, a)$.

PROOF: Define $\mathcal{P}$[natural number] $\equiv$ for every finite sequence G of elements of HFuncs $\mathbb{N}$ such that len $G = \$_1$ and rng $G \subseteq$ majorizedBy(AckermannFuncti there exists an element N of $\mathbb{N}$ such that for every homogeneous, ($\mathbb{N}^*$)-defined function g such that $g \in \operatorname{rng} G$ for every non empty finite sequence F of elements of $\mathbb{N}$ such that $F \in \operatorname{dom} g$ for every element a of $\mathbb{N}$ such that $a = \max \operatorname{rng} F$ holds $g(F) < \text{Ackermann}(N, a)$. $\mathcal{P}[0]$. For every natural number k such that $\mathcal{P}[k]$ holds $\mathcal{P}[k+1]$ by [4, (18)], [13, (53)], [14, (70)], [2, (11)]. For every natural number k, $\mathcal{P}[k]$ from [2, Sch. 2]. □

(43) AckermannFunction majorizes $\text{const}_m(n)$. The theorem is a consequence of (20).

(44) $\text{const}_m(n) \in$ majorizedBy(AckermannFunction). The theorem is a consequence of (43).

(45) If $\text{const}_m(n_1) = \text{const}_m(n_2)$, then $n_1 = n_2$.

(46) If $1 \leqslant n_1 \leqslant m$ and $1 \leqslant n_2 \leqslant m$ and $\text{succ}_m(n_1) = \text{succ}_m(n_2)$, then $n_1 = n_2$.

(47) If $1 \leqslant n_1 \leqslant m$ and $1 \leqslant n_2 \leqslant m$ and $\text{proj}_m(n_1) = \text{proj}_m(n_2)$, then $n_1 = n_2$.

Let m be a natural number. The functor constF(m) yielding a function from $\mathbb{N}$ into HFuncs $\mathbb{N}$ is defined by

(Def. 16) for every natural number n, $it(n) = \text{const}_m(n)$.

Let us consider m. One can check that constF(m) is one-to-one.

Now we state the proposition:

(48) rng constF(m) $\subseteq$ PrimRec.

Let us note that PrimRec is infinite.

Now we state the propositions:

(49) If $1 \leqslant n \leqslant m$, then AckermannFunction majorizes $\text{succ}_m(n)$. The theorem is a consequence of (9) and (24).

(50) If $1 \leqslant n \leqslant m$, then $\text{succ}_m(n) \in$ majorizedBy(AckermannFunction). The theorem is a consequence of (49).

(51) If $1 \leqslant n \leqslant m$, then AckermannFunction majorizes $\mathrm{proj}_m(n)$. The theorem is a consequence of (24) and (16).

(52) If $1 \leqslant n \leqslant m$, then $\mathrm{proj}_m(n) \in$ majorizedBy(AckermannFunction). The theorem is a consequence of (51).

Let us observe that majorizedBy(AckermannFunction) is composition closed and majorizedBy(AckermannFunction) is primitive recursion closed and majorizedBy(AckermannFunction) is primitive recursively closed.

Now we state the proposition:

(53) PrimRec $\subseteq$ majorizedBy(AckermannFunction).

Let us note that majorizedBy(AckermannFunction) is infinite.

Now we state the propositions:

(54) dom2tuple(AckermannFunction) is not primitive recursive. The theorem is a consequence of (53) and (40).

(55) dom2tuple(AckermannFunction) $\notin$ PrimRec.

(56) Let us consider a unary operation d on $\mathbb{N}$. Suppose for every natural number n, $d(n) = \mathrm{Ackermann}(n, n)$. Then dom2tuple($d$) is not primitive recursive. The theorem is a consequence of (53).

References

[1] Wilhelm Ackermann. Zum hilbertschen aufbau der reellen zahlen. *Mathematische Annalen*, 99:118–133, 1928. doi:10.1007/BF01459088.

[2] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.

[3] Grzegorz Bancerek. Epsilon numbers and Cantor normal form. *Formalized Mathematics*, 17(**4**):249–256, 2009. doi:10.2478/v10037-009-0032-8.

[4] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.

[5] Grzegorz Bancerek and Piotr Rudnicki. The set of primitive recursive functions. *Formalized Mathematics*, 9(**4**):705–720, 2001.

[6] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.

[7] Cristian Calude, Solomon Marcus, and Ionel Tevy. The first example of a recursive function which is not primitive recursive. *Historia Mathematica*, 6(4):380–384, 1979. doi:10.1016/0315-0860(79)90024-7.

[8] Mario Carneiro. Formalizing computability theory via partial recursive functions. In John Harrison, John O'Leary, and Andrew Tolmach, editors, *10th International Conference on Interactive Theorem Proving, ITP 2019, Portland, OR, USA, September 9-12, 2019*, volume 141 of *LIPIcs*, pages 12:1–12:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICS.ITP.2019.12.

[9] Rafał Kwiatek. Factorial and Newton coefficients. *Formalized Mathematics*, 1(**5**):887–890, 1990.

[10] Lawrence C. Paulson. Ackermann's function in iterative form: a proof assistant experiment. *Bull. Symb. Log.*, 27(4):426–435, 2021. doi:10.1017/BSL.2021.47.

[11] Yngve Sundblad. The ackermann function. a theoretical, computational, and formula manipulative study. *BIT Numerical Mathematics*, 11:107–119, 1971. doi:10.1007/BF01935330.

[12] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1 (**2**):329–334, 1990.
[13] Wojciech A. Trybulec. Non-contiguous substrings and one-to-one finite sequences. *Formalized Mathematics*, 1(**3**):569–573, 1990.
[14] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1 (**1**):73–83, 1990.

Received September 6, 2026