RAPPORT DE RECHERCHE

L R I

A CONSTRUCTION OF A DEFINITION RECURSIVE WITH RESPECT TO THE SECOND VARIABLE FOR THE ACKERMANN'S FUNCTION

FRANOVA M

Unité Mixte de Recherche 8623
CNRS-Université Paris Sud – LRI

12/2008

Rapport de Recherche N° 1511

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 490
91405 ORSAY Cedex (France)

A construction of a definition recursive with respect to the second variable for the Ackermann's function

Marta Fraňová

mf@lri.fr, http://www.lri.fr/~mf/index.html

UMR 8623 CNRS - Université Paris Sud
LRI, Bât. 490, 91405 Orsay Cedex, France

1. Introduction

Let N be the system of natural numbers defined by Peano's axioms. Let us recall that "multiple recursion" (or "simultaneous recursion on several variables") is defined schematically

$$f(0,n) = g_1(n) \tag{1}$$
$$f(m+1,0) = g_2(m) \tag{2}$$
$$f(m+1,n+1) = g_3(m,n,f(m,h(m,n)),f(m+1,n)), \tag{3}$$

where g_1, g_2, g_3 and h are already defined functions. f(m,h(m,n)) and f(m+1,n) can be considered as "previous" values of f, in m and n respectively.

Let us consider now the Ackermann-Péter's function ack defined axiomatically as follows.

$$ack(0,n) = n+1 \tag{A1}$$
$$ack(m+1,0) = ack(m,1) \tag{A2}$$
$$ack(m+1,n+1) = ack(m,ack(m+1,n)) \tag{A3}$$

Even though the computation of ack(m+1,n+1) in (A3) calls the computation of ack(m+1,n), this definition does not fall under the syntactic definition of the "multiple recursion". Actually, (2) shows that, in multiple recursion, computing f(m+1,0), i.e., when the second argument of f is 0, does not call recursively f, but depends on a previously defined function g_2 which is not defined in terms of f. (A2) shows that computing the value of ack(m+1,0) makes a recursive call to ack. Therefore, (A2) and (A3) should rather be considered as a conditional axiom

$$ack(m+1,y) = \begin{cases} ack(m,1), & \text{if } y = 0 \\ ack(m,ack(m+1,n)), & \text{if } y = n+1 \end{cases} . \tag{4}$$

In other words, the above definition defines the function ack recursively with respect to the first argument. Therefore, it is meaningful to formulate the problem of finding a definition that defines the function ack recursively with respect to the second argument. In the following part we are going to construct such a definition.

2. Theorem and its proof

Theorem 1

For the function ack defined by the system { (A1), (A2), (A3) }, there is a definition recursive in the second argument.

In other words, the recursive case of this new definition is of the form

$$ack(x,y+1) = f(x,y,ack(x,y)). \tag{5}$$

Proof

First, let us give an outline of our proof. We shall construct a definition for ack recursive in the second argument. This construction is performed via inductive proofs of the theorems

$$x \quad z\ ack(x,0) = z \tag{6}$$

and

$$x \quad y \quad z\ ack(x,y+1) = ack(x,y) + z. \tag{7}$$

Let sf be the Skolem function for (6) and sf3 the Skolem function for (7), i.e., let

$$x\ ack(x,0) = sf(x) \tag{8}$$

and

$$x \quad y\ ack(x,y+1) = ack(x,y) + sf3(x,y). \tag{9}$$

hold. We shall prove (6) and (7) by induction. This will provide definitions for sf and sf3. Then, let us rename the function ack to ak in (8), (9) and in the definition of sf3. We can write

$$ak(x,0) \;=\; sf(x) \tag{A4}$$

$$ak(x,y+1) \;=\; ak(x,y) + sf3(x,y) \tag{A5}$$

We thus obtain a definition for ak that does not depend on the definition for ack, while the performed construction guarantees that

$$x \quad y\ ack(x,y) = ak(x,y). \tag{10}$$

Note 1: On independence of ak on ack

Let us note that (A4) is a candidate for the base case of a definition for ak recursive with respect to the second argument if and only if the definition for the function sf does not depend on ak (i.e., nor on ack). Therefore, in our search for a definition of sf we ***cannot*** put

$$sf(x) \;=\; \begin{cases} 1, & \text{if } x = 0 \\ ack(x-1,1), & \text{if } x \neq 0 \end{cases}$$

even though this trivially is implied by (A1) and (A2).

End of the note.

Before presenting our proofs for (6) and (7), let us recall that we admit the possibility that the reader may find proofs that are different from those we are going to present. The proofs we are going to present here follow basically the algorithms of our *Constructive Matching* methodology (*CMM*) [franova30]. In this sense, these proofs can be seen as a wishful result of a user-independent inductive theorem prover. In other words, as soon as our methodology is completely implemented, our system *PRECOMAS* [franova19], [franova24], or another system built over our methodology, shall hopefully help the user to to synthesize ak just from the above given axiomatic definition of the function ack, specifying N as a constructible domain (see [franova38]) and the unusual (for *CMM*) constraint that ak must not depend on ack. Some of the steps in the following shall be familiar to many readers, some steps are more sophisticated and require a slight acquaintance with some specific parts of our *CMM* or of our Theory of Constructible Domains. Some of the steps are even a basis for suggesting new heuristics that have to be further worked on to become a legal part of *CMM*. We shall try here to make accessible the main idea of these specific parts. But let us insist on the user-dependency of this proof with respect to the constraint that ak has not to be expressed in terms of ack.

Proof for (6):

Let us denote by F(x) the formula   z ack(x,0) = z. We shall perform the proof of (6) by induction on x. This means, that in the base step, when x = 0, we have to prove

$$z \; ack(0,0) = z \tag{11}$$

and, in the induction step, when x = a+1, we have to prove

$$z \; ack(a+1,0) = z \tag{12}$$

assuming the induction hypothesis

$$e \; ack(a,0) = e, \tag{13}$$

as well as the induction hypothesis

$$t \; t<a+1 \qquad e \; ack(t,0) = e. \tag{14}$$

Here, the relation < is the Peano's well-founded relation (see [franova38]) induced by constructors on Constructible Domain specified by the constructors 0 and the successor function +1.
Note that, with respect to our notation, in (13)

$$e = sf(a) \tag{15}$$

and, in (12), we look for

$$z = sf(a+1). \tag{16}$$

Base step for (6)

By evaluation of the term ack(0,0) in (11) we trivially have

$$z = 1. \tag{17}$$

With respect to our notation, we thus have

$$sf(0) = 1. \tag{18}$$

Induction Step for (6)

Assuming (13) we have to prove (12). By evaluation of ack(a+1,0), (12) is replaced by

$$z \; ack(a,1) = z \tag{19}$$

From inductive theorem proving view, it is obvious that z = ack(a,1) is a trivial solution for (19). However, we have mentioned already that sf must not be defined in terms of the function ack. In other words, (19) has to be attempted to be solved applying the induction hypothesis (13) which, hopefully, might eliminate the occurrence of ack.
However, the *CMM* fails to apply the induction hypothesis (13) to (19) directly. Therefore, to succeed the application of the induction hypothesis, *CMM* generates the subproblem

$$z1 \; ack(a,1) = ack(a,0) + z1. \tag{20}$$

In fact, (20) has to do with the fact that *CMM* is proving the given theorem in a constructible domain, and for each constructible domain the so-called jump-constructor function is determined (+ in N) as well as the so-called jump-constructor representation of two simply comparable elements (in N: x   y {x<y   z x+z = y}).

CMM does not find a trivial solution for this problem. Therefore, the last formula is generalized to the lemma

$$a1 \quad z1 \; ack(a1,1) = ack(a1,0) + z1. \tag{21}$$

Note 2: On lemmas generation in *CMM*

In course of a proof by induction of a formula F, it is necessary to keep a trace of problems met, solved, as well as the links between the given formula F and lemmas generated in course of its proof. Taking into account the lemmas "dependence" on the environment in which they are proved is a very important feature of *CMM*. Thus, (20) is a sufficient condition for a success of proving the induction step for (6). While a1 in (21) is universally quantified, for the success of (20) not the validity of (21) in its generality is necessary, but, in the framework of the induction step for (6) it is sufficient to consider the formula

$$a1 \{ a1 < a+1 \quad z1\ ack(a1,1) = ack(a1,0) + z1\}. \qquad (22)$$

In general, proving formulae that are implications, such as it is the case of the last formula, is complex. In consequence, for simplicity of our presentation, we shall deal here with (21) instead of (22) and we shall take into account the condition a1 < a+1 if and whenever necessary.

End of the note.

First, let us show, how solving (21) contributes to solving (20) , i.e., to (19).
Let us denote by sf1 the Skolem function of (21), i.e.,

$$a1\ ack(a1,1) = ack(a1,0) + sf1(a1) \qquad (23)$$

holds. As soon as (21) is proved by induction, and thus a definition for sf1 obtained, (19) is replaced by

$$z\ ack(a,0) + sf1(a) = z. \qquad (24)$$

Now, the induction hypothesis (13) is applied, which yields

$$z\ e + sf1(a) = z. \qquad (25)$$

Using (15) and (16), we thus have

$$sf(a) + sf1(a) = sf(a+1). \qquad (26)$$

Then, if the definition of sf1 does not make a reference to the function ack, (26) completes the proof of (6) and the following formulae

$$sf(0) = 1 \qquad (A6)$$

$$sf(a1+1) = sf(a1) + sf1(a1) \qquad (A7)$$

shall define axiomatically the function sf.

We are going to prove (21).

<u>Proof for (21)</u>

In the base step, a1 = 0, it is necessary to prove

$$z1\ ack(0,1) = ack(0,0) + z1. \qquad (27)$$

In the induction step, a1 = b+1, the formula

$$z1\ ack(b+1,1) = ack(b+1,0) + z1 \qquad (28)$$

has to be proved assuming the induction hypothesis

$$e1\ ack(b,1) = ack(b,0) + e1. \qquad (29)$$

Note that e1 is sf1(b). Since (21) is a lemma generated in the induction step of (6), *CMM* generates (see [franova38]) also particular induction hypotheses corresponding to (6) in the sense that b < b+1 = a1 < a1+1 < a+1, and thus, replacing b+1 for t in (14),

$$e2\ ack(b+1,0) = e2, \qquad (30)$$

and, substituting b for t in (14),

$$e3\ ack(b,0) = e3, \qquad (31)$$

are the assumptions that may be used while proving the induction step of (21). Note that

$$e2 = sf(b+1) \tag{32}$$

and

$$e3 = sf(b). \tag{33}$$

Base step for (21)

For base step, *CMM* finds trivially z1 = 1. In other words,

$$sf1(0) = 1. \tag{34}$$

Induction step for (21)

By a repeated evaluation, (28) changes to

$$z1\ ack(b,ack(b,1)) = ack(b,1) + z1. \tag{35}$$

The induction hypothesis (29) can now be applied. This leads to

$$z1\ ack(b,ack(b,0)+e1) = (ack(b,0)+e1) + z1. \tag{36}$$

No axioms can be applied to this last formula. However, (31) can be applied, which, with respect to (33) gives

$$z1\ ack(b,sf(b)+e1) = (sf(b)+e1) + z1. \tag{37}$$

This problem can be simplified no more. Therefore, *CMM* generates a new lemma to solve the problem specified by (37). First of all, by the *CMM*-lemma-generation analysis of (37) the obvious generalization (see [franova-kodratoff04]) is applied transforming (37) to

$$z1\ ack(b,y) = y + z1. \tag{38}$$

The generalization substitution y = sf(b) + e1 is kept in the memory of the *CMM*-environment. e1 = sf1(b), thus y = sf(b) + sf1(b).

The lemma generated from (38) is

$$x1\quad y\quad z3\ ack(x1,y) = y + z3. \tag{39}$$

Let us denote by sf2 the Skolem function corresponding to (39). In other words,

$$x1\quad y\ ack(x1,y) = y + sf2(x1,y) \tag{40}$$

holds.

Proof for (39)

We shall prove (39) by induction. In this way, we shall obtain a definition for sf2 as well.

Note that, in this notation, z1 that is a solution for (37) is sf2(b,sf(b) + sf1(b)), i.e.,

$$sf1(b+1) = sf2(b,sf(b) + sf1(b)). \tag{41}$$

Provided now the definition for sf2 does not refer to ack, both sf1 and sf shall not refer to ack, and thus (A4) will be an acceptable base step definition for ak. Note also that sf(b) + sf1(b) = sf(b+1), therefore, after a successful proof of (39), we shall have the following axiomatic definition for sf1.

$$sf1(0) = 1 \tag{A8}$$

$$sf1(b+1) = sf2(b,sf(b+1)) \tag{A9}$$

Before starting the proof of (39), it is necessary to prepare all the assumptions that can be taken into account during the proof of (39). (39) is a lemma generated in course of the proof (21), and thus, in course of the proof of (6) as well. (39) is a generalization of

(38), x1 in (39) corresponds to b of (38), which, in turn comes from the representation a1 = b+1. Thus, we have

$$x1 \leq b < b+1 = a1 \leq a < a+1 = x. \quad (42)$$

In consequence, in the framework of (6), it is logically justified to generate the assumption

$$e4\ ack(t,0) = e4, \quad (43)$$

and, in the framework of (21), it is logically justified to generate the assumption

$$e5\ ack(t,1) = ack(t,0) + e5. \quad (44)$$

(43) and (44) may be applied while proving (39) for each t verifying $t \leq x1$. In this environment, *CMM* starts to prove (39). This formula contains two universally quantified variables, the function ack is defined recursively with respect to the first argument. This is why the variable x1 is selected to be the induction variable. With respect to this choice, *CMM* formulates the base and induction steps.

Base step for (39)

In base step, for x1 = 0, the formula

$$y \quad z3\ ack(0,y) = y + z3 \quad (45)$$

has to be proved. By evaluation and simple equation solving, it is found

$$z3 = 1, \quad (46)$$

i.e.,

$$sf2(0,y) = 1. \quad (47)$$

Induction step for (39)

In induction step, $x1 = a_2+1$ and *CMM* generates the following induction hypotheses (see [franova38])

$$e5\ ack(a_2,y) = y + e5, \quad (48)$$

e5 is here $sf2(a_2,y)$);

$$u \quad e6\ ack(a_2,u) = u + e6, \quad (49)$$

e6 is $sf2(a_2,u)$;

$$t \quad q\ q < a_2+1 \qquad e7\ ack(q,t) = t + e7, \quad (50)$$

e7 is sf2(q,t), and finally

$$t\ t < y \qquad e8\ ack(a_2+1,t) = t + e8, \quad (51)$$

e8 is $sf2(a_2+1,t)$.

For our presentation it is not important that (48) and (49) are particular cases of (50). (51) corresponds to the recursive call of ack with respect to the second argument in the axiom (A3). *CMM* keeps on mind that an application of (51) requires the corresponding base step, namely proving

$$z3\ ack(a_2+1,0) = 0 + z3. \quad (52)$$

Summarizing, we have

$$e5 = sf2(a_2,y),\ e6 = sf2(a_2,u),\ e7 = sf2(q,t),\ e8 = sf2(a_2+1,t). \quad (53)$$

Assuming now (43), (44), and the above mentioned induction hypotheses, the goal is to prove

$$z3\ ack(a_2+1,y) = y + z3. \quad (54)$$

The value of z3 found will correspond to $sf2(a_2+1,y)$.

In order to solve the problem specified by (54), axioms (A2) and (A3) indicate that two cases specified by y = 0 and $y = b_2+1$, respectively, have to be considered. Thus, (54) is replaced by two subproblems, namely

$$z31\ ack(a_2+1,0) = 0 + z31 \quad (55)$$

and

$$z32\ ack(a_2+1,b_2+1) = (b_2+1) + z32. \quad (56)$$

We have here that z31 from (55) corresponds to $sf2(a_2+1,0)$; z32 from (56) corresponds to $sf2(a_2+1,b_2+1)$.

Let us treat the subproblem specified by (55). By evaluation, (55) is replaced by

$$z31\ ack(a_2,1) = z31. \quad (57)$$

In terms of the given definition for the function ack, $ack(a_2,1)$ is an evaluable expression for each concrete value of a_2. However, with respect to the constraint of defining sf2 independently of ack, (57) cannot be considered as an equation that suggests itself the value for z31. Nevertheless, since (57) has to be solved in the framework of the induction step for (39), we can check whether an induction hypothesis can be applied. This search is successful. Let us consider (49) with the instantiation u = 1, i.e.,

$$e61\ ack(a_2,1) = 1 + e61. \quad (58)$$

Obviously,

$$e61 = sf2(a_2,1). \quad (59)$$

In consequence, the application of (58) to **Erreur! Source du renvoi introuvable.** yields

$$z31\ 1+ e61 = z31, \quad (60)$$

i.e.,

$$z31\ 1+ sf2(a_2,1) = z31. \quad (61)$$

As noted above, z31 from (55) corresponds to $sf2(a_2+1,0)$. Thus, we obtain

$$sf2(a_2+1,0) = 1+ sf2(a_2,1). \quad (62)$$

Let us consider now the formula (56). By evaluation, we have

$$z32\ ack(a_2,ack(a_2+1,b_2)) = (b_2+1) + z32. \quad (63)$$

No axioms are applicable, however it is possible to apply induction hypotheses. First of all, the application of (49) with $u = ack(a_2+1,b_2)$, i.e.,

$$e62\ ack(a_2,ack(a_2+1,b_2)) = ack(a_2+1,b_2) + e62, \quad (64)$$

is performed, (here, $e62 = sf2(a_2,ack(a_2+1,b_2))$), and by the application of this instance,

$$ack(a_2,ack(a_2+1,b_2)) \quad (65)$$

in (63) is replaced by

$$ack(a_2+1,b_2) + e62, \quad (66)$$

i.e.,

$$ack(a_2+1,b_2) + sf2(a_2,ack(a_2+1,b_2)) = (b_2+1) + z32. \quad (67)$$

To eliminate the occurrence of ack in the last terms, to $ack(a_2+1,b_2)$, (51) is applied with $t = b_2$, which trivially verifies $t < b_2+1 = y$, i.e., we shall apply the induction hypothesis

$$b_2 < b_2+1 \qquad e81\ ack(a_2+1,b_2) = b_2 + e81. \quad (68)$$

Here, $e81 = sf2(a_2+1,b_2)$.

Thus (66) changes to

$$(b_2 + e81) + e62, \quad (69)$$

i.e.,

$$(b_2 + sf2(a_2+1,b_2)) + sf2(a_2,b_2 + sf2(a_2+1,b_2)). \quad (70)$$

In consequence, (63) changes to

$$z32\ (b_2 + e81) + e62 = (b_2+1) + z32, \tag{71}$$

i.e.,

$$z32\ (b_2 + sf2(a_2+1,b_2)) + sf2(a_2,b_2 + sf2(a_2+1,b_2)) = (b_2+1) + z32. \tag{72}$$

In thus expressed problem, the function ack is occurs no more, and the problem is expressed only in terms of the functional symbols + and sf2. Namely, since z32 is sf2(a_2 +1,b_2+1), by eliminating b_2 on both sides of the equation, we have

$$sf2(a_2+1,b_2) + sf2(a_2,b_2 + sf2(a_2+1,b_2)) = 1 + sf2(a_2+1,b_2+1). \tag{73}$$

In order to obtain the final line of the definition for sf2, we would need to subtract 1 from the both sides of the last equation in order to obtain

$$sf2(a_2+1,b_2+1) = (sf2(a_2+1,b_2) + sf2(a_2,b_2 + sf2(a_2+1,b_2))) - 1. \tag{74}$$

A sufficient condition to do so would be that sf2(m,n) ≠ 0 for all m and n. Since sf2 is a function in course of construction, we shall apply the heuristics of the introduction of a hypothetical program. This means that we introduce a new hypothetical program, say f, which is a simple rewriting of sf2 for which we do the subtraction in the last equation and we try to prove that f(m,n) ≠ 0 for all m and n. So, let us introduce f

$$f(0,y) = 1 \tag{R1}$$

$$f(a_2+1,0) = 1+ f(a_2,1) \tag{R2}$$

$$f(a_2+1,b_2+1) = (f(a_2+1,b_2) + f(a_2,b_2 + f(a_2+1,b_2))) - 1. \tag{R3}$$

Our goal is to prove that

$$m \quad n \ \ f(m,n) \neq 0. \tag{75}$$

It is not difficult to prove it it via an inductive proof of

$$m \quad n \quad z\ f(m,n) = z+1. \tag{76}$$

An inductive proof for this formula does not represent many technical challenges and so we omit it here.

This completes the proof of (39), thus the proof of (21), and, finally, the proof of (6). (47), (62) and (74) give the following axiomatic definition of sf2

$$sf2(0,y) = 1 \tag{A10}$$

$$sf2(a_2+1,0) = 1+ sf2(a_2,1). \tag{A11}$$

$$sf2(a_2+1,b_2+1) = sf2(a_2+1,b_2)+ sf2(a_2,b_2+sf2(a_2+1,b_2)) - 1 \tag{A12}$$

Proof for (7)

We shall now present the *CMM*-proof for (7), i.e., the formula

$$x \quad y \quad z\ ack(x,y+1) = ack(x,y) + z.$$

Recall that sf3 is our notation for the Skolem function of (7). The proof is by induction on x.

Base step for (7)

In the base step, x = 0. The goal is to prove

$$z\ ack(0,y+1) = ack(0,y) + z. \tag{77}$$

By evaluation this transforms to

$$z\ (y+1) + 1 = (y+1) + z. \tag{78}$$

This gives z = 1, i.e.,

$$sf3(0,y) = 1. \tag{79}$$

Induction step for (7)

In the induction step, $x = a_3+1$, the induction hypotheses are

$$u \quad e9\ ack(a_3,u+1) = ack(a_3,u) + e9, \tag{80}$$

its more general version

$$u \quad v\ v < a_3+1 \quad e10\ ack(v,u+1) = ack(v,u) + e10, \tag{81}$$

and

$$t\ t<y \qquad e11\ ack(a_3+1,t+1) = ack(a_3+1,t) + e11. \tag{82}$$

The goal is to prove

$$z\ ack(a_3+1,y+1) = ack(a_3+1,y) + z. \tag{83}$$

By evaluation, this gives

$$z\ ack(a_3,ack(a_3+1,y)) = ack(a_3+1,y) + z. \tag{84}$$

No induction hypothesis can be applied. Instead of forcing the application of an induction hypotheses, a new lemma is generated. The lemma-generation analysis recognizes that the obvious generalization [franova-kodratoff04] can be applied. This leads to

$$a_3 \quad m \quad z\ ack(a_3,m) = m + z \tag{85}$$

with $m = ack(a_3+1,y)$. But this lemma is recognized as identical to (39). In consequence, for z from (85), we have

$$z = sf2(a_3,m). \tag{86}$$

In consequence, for z from (84), i.e., for $sf3(a_3+1,y)$ we have

$$sf3(a_3+1,y) = sf2(a_3,ack(a_3+1,y)). \tag{87}$$

By this, the proof of (7) is completed. (79) and (87) give the following axiomatic definition for sf3

$$sf3(0,y) = 1 \tag{A13}$$

$$sf3(a_3+1,y) = sf2(a_3,ack(a_3+1,y)). \tag{A*}$$

Summarizing, as a by-product of the above proofs for (6) and (7) we have synthesized the functions sf, sf1, sf2 and sf3. Let us consider now the definition of the function ak given by the axioms (A4) and (A5) given above. These axioms correspond to the theorems (6) and (7), respectively. In other words, the symbol ak in (A4) and (A5) describes the same function as the symbol ack in (6) and (7). Therefore, it is possible to render the definition of ak independent of the definition for ack. To do this, it is sufficient to replace the symbol ack in (87), i.e., in the definition of sf3, by the symbol ak. In other words, we shall define sf3 by the axioms

$$sf3(0,y) = 1 \tag{A13}$$

$$sf3(a_3+1,y) = sf2(a_3,ak(a_3+1,y)) \tag{A14}$$

3. 3. Corollary

Let ack be the function defined axiomatically by

$$\begin{aligned} ack(0,n) &= n+1 && \text{(A1)} \\ ack(m+1,0) &= ack(m,1) && \text{(A2)} \\ ack(m+1,n+1) &= ack(m,ack(m+1,n)) && \text{(A3)} \end{aligned}$$

Let the function ak be defined axiomatically by

$$\begin{aligned} ak(x,0) &= sf(x) && \text{(A4)} \\ ak(x,y+1) &= ak(x,y) + sf3(x,y) && \text{(A5)} \end{aligned}$$

where
sf is defined by

$$\begin{aligned} sf(0) &= 1 && \text{(A6)} \\ sf(a1+1) &= sf(a1) + sf1(a1) && \text{(A7)} \end{aligned}$$

sf1 is defined by

$$\begin{aligned} sf1(0) &= 1 && \text{(A8)} \\ sf1(b+1) &= sf2(b,sf(b+1)) && \text{(A9)} \end{aligned}$$

sf2 is defined by

$$\begin{aligned} sf2(0,y) &= 1 && \text{(A10)} \\ sf2(a_2+1,0) &= 1+ sf2(a_2,1). && \text{(A11)} \\ sf2(a_2+1,b_2+1) &= sf2(a_2+1,b_2) + sf2(a_2,b_2+sf2(a_2+1,b_2)) - 1 && \text{(A12)} \end{aligned}$$

and
sf3 is defined by

$$\begin{aligned} sf3(0,y) &= 1 && \text{(A13)} \\ sf3(a_3+1,y) &= sf2(a_3,ak(a_3+1,y)) && \text{(A14)} \end{aligned}$$

Then,

$$x \quad y \; ack(x,y) = ak(x,y)$$

holds.

Finally, let us note that (A5) can be written in the form of the following two conditional axioms, eliminating sf3 from the definition of ak. We then have

$$\begin{aligned} ak(x,y+1) &= ak(0,y) + 1, \text{ if } x = 0 \\ ak(x,y+1) &= ak(x,y) + sf2(x-1,ak(x,y)), \text{ if } x \neq 0 \end{aligned}$$

The recursion of ak with respect to the second argument becomes in this way more transparent.

4. a LISP program

Just for the reader's convenience, let us give the corresponding LISP program.

```
(defun sf (x)
   (if    (= x 0)
          1
          (+ (sf (sub1 x)) (sf1 (sub1 x)))))

(defun sf1 (x)
   (if    (= x 0)
          1
          (sf2 (sub1 x) (sf x))))

(defun sf2 (x y)
   (if    (= x 0)
          1
          (if    (= y 0)
                 (+ 1 (sf2 (sub1 x) 1))
                 (sub1 (+ (sf2 x (sub1 y)) (sf2 (sub1 x) (+ (sub1 y) (sf2 x (sub1 y)))))))))


(defun sf3 (x y)
   (if    (= x 0)
          1
          (sf2 (sub1 x) (ak x y))))

(defun ak (x y)
   (if    (= y 0)
          (sf x)
          (+ (ak x (sub1 y)) (sf3 x (sub1 y)))))
```

References

[franova-kodratoff04] M. Franova, Y. Kodratoff: Practical Problems in the Automatization of Inductive Theorem Proving; Rapport de Recherche No.752, L.R.I., Université de Paris-Sud, Orsay, France, Mai, 1992.

[franova-kodratoff07] M. Franova, Y. Kodratoff: Predicate Synthesis from Formal Specifications: Using Mathematical Induction for Finding the Preconditions of Theorems; Rapport de Recherche No.781, L.R.I., Université de Paris-Sud, Orsay, France, October, 1992.

[franova08] M. Franova: *PRECOMAS* Challenge; Rapport de Recherche No.376, L.R.I., Université de Paris-Sud, Orsay, France, September, 1987.

[franova19] M. Franova: *PRECOMAS* 0.3 User Guide; Rapport de Recherche No.524, L.R.I., Université de Paris-Sud, Orsay, France, October, 1989.

[franova24] M. Franova: *PRECOMAS* - An Implementation of *Constructive Matching* Methodology; Proceedings of ISSAC'90 (Tokyo, Japan, August 20-24, 1990), ACM, New York, 1990, 16-23.

[franova30] M. Franova: *Constructive Matching* methodology: a standard way of proving user-independently theorems by induction; Rapport de Recherche No.973, L.R.I., Université de Paris-Sud, Orsay, France, Mai, 1995.

[franova36] M. Franova: A standard proof by *Constructive Matching* methodology for a theorem formulated by Skolem; Rapport de Recherche No.972, L.R.I., Université de Paris-Sud, Orsay, France, Mai, 1995.

[franova38] M. Franova: A Theory of Constructible Domains - a formalization of inductively defined systems of objects for a user-independent automation of inductive theorem proving, Part I; Rapport de Recherche No.970, L.R.I., Université de Paris-Sud, Orsay, France, Mai, 1995.