FORMALIZED MATHEMATICS
Vol. 27, No. 2, Pages 181–187, 2019
DOI: 10.2478/forma-2019-0017

sciendo
https://www.sciendo.com/

Partial Correctness of a Factorial Algorithm

Adrian Jaszczak
Institute of Informatics
University of Białystok
Poland

Artur Korniłowicz
Institute of Informatics
University of Białystok
Poland

Summary. In this paper we present a formalization in the Mizar system [3],[1] of the partial correctness of the algorithm:

```
i := val.1
j := val.2
n := val.3
s := val.4
while (i <> n)
   i := i + j
   s := s * i
return s
```

computing the factorial of given natural number `n`, where variables `i, n, s` are located as values of a `V-valued Function`, `loc`, as: `loc/.1 = i`, `loc/.3 = n` and `loc/.4 = s`, and the constant `1` is located in the location `loc/.2 = j` (set `V` represents simple names of considered nominative data [16]).

This work continues a formal verification of algorithms written in terms of simple-named complex-valued nominative data [6],[8],[14],[10],[11],[12]. The validity of the algorithm is presented in terms of semantic Floyd-Hoare triples over such data [9]. Proofs of the correctness are based on an inference system for an extended Floyd-Hoare logic [2],[4] with partial pre- and post-conditions [13],[15],[7],[5].

MSC: 68Q60 68T37 03B70 03B35

Keywords: factorial; nominative data; program verification

MML identifier: `NOMIN_5`, version: `8.1.09 5.57.1355`

Let D be a set and f_1, f_2, f_3 be binominative functions of D. The functor PP-composition(f_1, f_2, f_3) yielding a binominative function of D is defined by the term

(Def. 1) $f_1 \bullet f_2 \bullet f_3$.

Let f_1, f_2, f_3, f_4 be binominative functions of D. The functor PP-composition (f_1, f_2, f_3, f_4) yielding a binominative function of D is defined by the term

(Def. 2) PP-composition$(f_1, f_2, f_3) \bullet f_4$.

From now on D denotes a non empty set, f_1, f_2, f_3, f_4 denote binominative functions of D, and p, q, r, t, w denote partial predicates of D.

Now we state the proposition:

(1) UNCONDITIONAL COMPOSITION RULE FOR 3 PROGRAMS: Suppose $\langle p, f_1, q\rangle$ is an SFHT of D and $\langle q, f_2, r\rangle$ is an SFHT of D and $\langle r, f_3, w\rangle$ is an SFHT of D and $\langle \sim q, f_2, r\rangle$ is an SFHT of D and $\langle \sim r, f_3, w\rangle$ is an SFHT of D. Then $\langle p, \text{PP-composition}(f_1, f_2, f_3), w\rangle$ is an SFHT of D.

(2) UNCONDITIONAL COMPOSITION RULE FOR 4 PROGRAMS: Suppose $\langle p, f_1, q\rangle$ is an SFHT of D and $\langle q, f_2, r\rangle$ is an SFHT of D and $\langle r, f_3, w\rangle$ is an SFHT of D and $\langle w, f_4, t\rangle$ is an SFHT of D and $\langle \sim q, f_2, r\rangle$ is an SFHT of D and $\langle \sim r, f_3, w\rangle$ is an SFHT of D and $\langle \sim w, f_4, t\rangle$ is an SFHT of D. Then $\langle p, \text{PP-composition}(f_1, f_2, f_3, f_4), t\rangle$ is an SFHT of D.

In the sequel d, v, v_1 denote objects, V, A denote sets, z denotes an element of V, d_1 denotes a non-atomic nominative data of V and A, f denotes a binominative function over simple-named complex-valued nominative data of V and A, and T denotes a nominative data with simple names from V and complex values from A.

Now we state the proposition:

(3) If V is without nonatomic nominative data w.r.t. A and $v \in V$ and $v \neq v_1$ and $v_1 \in \operatorname{dom} d_1$, then $(d_1 \nabla_a^v T)(v_1) = d_1(v_1)$.

Let x, y be objects. Assume x is a complex number and y is a complex number. The functors: $x + y$ and $x * y$ yielding complex numbers are defined by conditions

(Def. 3) there exist complex numbers x_1, y_1 such that $x_1 = x$ and $y_1 = y$ and $x + y = x_1 + y_1$,

(Def. 4) there exist complex numbers x_1, y_1 such that $x_1 = x$ and $y_1 = y$ and $x * y = x_1 \cdot y_1$,

respectively. Let us consider A. Assume A is complex containing. The functors: addition(A) and multiplication(A) yielding functions from $A \times A$ into A are defined by conditions

(Def. 5) for every objects x, y such that $x, y \in A$ holds addition$(A)(x, y) = x + y$,

(Def. 6) for every objects x, y such that $x, y \in A$ holds multiplication$(A)(x, y) = x * y$,

respectively. Let us consider V. Let x, y be elements of V. The functors: addition

(A, x, y) and multiplication(A, x, y) yielding binominative functions over simple-named complex-valued nominative date of V and A are defined by terms

(Def. 7) lift-binary-op(addition$(A), x, y)$,

(Def. 8) lift-binary-op(multiplication$(A), x, y)$,

respectively.

Let us consider elements i, j of V and complex numbers x, y. Now we state the propositions:

(4) Suppose A is complex containing and $i, j \in \operatorname{dom} d_1$ and $d_1 \in$ dom(addition (A, i, j)). Then if $x = d_1(i)$ and $y = d_1(j)$, then (addition$(A, i, j))(d_1) = x + y$.

(5) Suppose A is complex containing and i, $j \in \operatorname{dom} d_1$ and $d_1 \in$ dom(multiplication(A, i, j)). Then if $x = d_1(i)$ and $y = d_1(j)$, then (multiplication$(A, i, j))(d_1) = x \cdot y$.

In the sequel loc denotes a V-valued function and val denotes a function.

Let us consider V, A, and loc. The functor factorial-loop-body(A, loc) yielding a binominative function over simple-named complex-valued nominative data of V and A is defined by the term

(Def. 9) $\mathrm{Asg}^{(loc_{/1})}$(addition$(A, loc_{/1}, loc_{/2})) \bullet (\mathrm{Asg}^{(loc_{/4})}$(multiplication$(A, loc_{/4}, loc_{/1})))$.

The functor factorial-main-loop(A, loc) yielding a binominative function over simple-named complex-valued nominative data of V and A is defined by the term

(Def. 10) WH($\neg$ Equality$(A, loc_{/1}, loc_{/3})$, factorial-loop-body(A, loc)).

Let us consider val. The functor factorial-var-init(A, loc, val) yielding a binominative function over simple-named complex-valued nominative data of V and A is defined by the term

(Def. 11) PP-composition$(\mathrm{Asg}^{(loc_{/1})}(val(1) \Rightarrow_a), \mathrm{Asg}^{(loc_{/2})}(val(2) \Rightarrow_a), \mathrm{Asg}^{(loc_{/3})}(val(3) \Rightarrow_a), \mathrm{Asg}^{(loc_{/4})}(val(4) \Rightarrow_a))$.

The functor factorial-main-part(A, loc, val) yielding a binominative function over simple-named complex-valued nominative data of V and A is defined by the term

(Def. 12) factorial-var-init$(A, loc, val) \bullet$ (factorial-main-loop(A, loc)).

Let us consider z. The functor factorial-program(A, loc, val, z) yielding a binominative function over simple-named complex-valued nominative data of V and A is defined by the term

(Def. 13) factorial-main-part$(A, loc, val) \bullet (\mathrm{Asg}^{z}((loc_{/4}) \Rightarrow_a))$.

In the sequel n_0 denotes a natural number.

Let us consider V, A, val, n_0, and d. We say that n_0 and d constitute a valid input for the factorial w.r.t. V, A and val if and only if

(Def. 14) there exists a non-atomic nominative data d_1 of V and A such that $d = d_1$ and $\{val(1), val(2), val(3), val(4)\} \subseteq \operatorname{dom} d_1$ and $d_1(val(1)) = 0$ and $d_1(val(2)) = 1$ and $d_1(val(3)) = n_0$ and $d_1(val(4)) = 1$.

The functor valid-factorial-input(V, A, val, n_0) yielding a partial predicate over simple-named complex-valued nominative data of V and A is defined by

(Def. 15) $\operatorname{dom} it = \mathrm{ND}_{\mathrm{SC}}(V, A)$ and for every object d such that $d \in \operatorname{dom} it$ holds if n_0 and d constitute a valid input for the factorial w.r.t. V, A and val, then $it(d) = true$ and if n_0 and d do not constitute a valid input for the factorial w.r.t. V, A and val, then $it(d) = false$.

Note that valid-factorial-input(V, A, val, n_0) is total.

Let us consider z and d. We say that n_0 and d constitute a valid output for the factorial w.r.t. A and z if and only if

(Def. 16) there exists a non-atomic nominative data d_1 of V and A such that $d = d_1$ and $z \in \operatorname{dom} d_1$ and $d_1(z) = n_0!$.

The functor valid-factorial-output(A, z, n_0) yielding a partial predicate over simple-named complex-valued nominative data of V and A is defined by

(Def. 17) $\operatorname{dom} it = \{d$, where d is a nominative data with simple names from V and complex values from $A : d \in \operatorname{dom}(z \Rightarrow_a)\}$ and for every object d such that $d \in \operatorname{dom} it$ holds if n_0 and d constitute a valid output for the factorial w.r.t. A and z, then $it(d) = true$ and if n_0 and d do not constitute a valid output for the factorial w.r.t. A and z, then $it(d) = false$.

Let us consider loc and d. We say that n_0 and d constitute a valid invariant for the factorial w.r.t. A and loc if and only if

(Def. 18) there exists a non-atomic nominative data d_1 of V and A such that $d = d_1$ and $\{loc_{/1}, loc_{/2}, loc_{/3}, loc_{/4}\} \subseteq \operatorname{dom} d_1$ and $d_1(loc_{/2}) = 1$ and $d_1(loc_{/3}) = n_0$ and there exist natural numbers I, S such that $I = d_1(loc_{/1})$ and $S = d_1(loc_{/4})$ and $S = I!$.

The functor factorial-inv(A, loc, n_0) yielding a partial predicate over simple-named complex-valued nominative data of V and A is defined by

(Def. 19) $\operatorname{dom} it = \mathrm{ND}_{\mathrm{SC}}(V, A)$ and for every object d such that $d \in \operatorname{dom} it$ holds if n_0 and d constitute a valid invariant for the factorial w.r.t. A and loc, then $it(d) = true$ and if n_0 and d do not constitute a valid invariant for the factorial w.r.t. A and loc, then $it(d) = false$.

One can check that factorial-inv(A, loc, n_0) is total.

Let us consider val. We say that loc and val are compatible w.r.t. 4 locations if and only if

(Def. 20) $val(4) \neq loc_{/3}$ and $val(4) \neq loc_{/2}$ and $val(4) \neq loc_{/1}$ and $val(3) \neq loc_{/2}$ and $val(3) \neq loc_{/1}$ and $val(2) \neq loc_{/1}$.

Now we state the propositions:

(6) Suppose V is not empty and V is without nonatomic nominative data w.r.t. A and $loc_{/1}$, $loc_{/2}$, $loc_{/3}$, $loc_{/4}$ are mutually different and loc and val are compatible w.r.t. 4 locations. Then $\langle$valid-factorial-input(V, A, val, n_0), factorial-var-init(A, loc, val), factorial-inv$(A, loc, n_0)\rangle$ is an SFHT of $\mathrm{ND_{SC}}$ (V, A).

PROOF: Set $i = loc_{/1}$. Set $j = loc_{/2}$. Set $n = loc_{/3}$. Set $s = loc_{/4}$. Set $i_1 = val(1)$. Set $j_1 = val(2)$. Set $n_1 = val(3)$. Set $s_1 = val(4)$. Set $I =$ valid-factorial-input(V, A, val, n_0). Set $i_2 =$ factorial-inv(A, loc, n_0). Set $D_1 = i_1 \Rightarrow_a$. Set $D_2 = j_1 \Rightarrow_a$. Set $D_3 = n_1 \Rightarrow_a$. Set $D_4 = s_1 \Rightarrow_a$. Set $S_1 = \mathrm{S_P}(i_2, D_4, s)$. Set $R_1 = \mathrm{S_P}(S_1, D_3, n)$. Set $Q_1 = \mathrm{S_P}(R_1, D_2, j)$. Set $P_1 = \mathrm{S_P}(Q_1, D_1, i)$. $I \models P_1$. $\square$

(7) Suppose V is not empty and A is complex containing and V is without nonatomic nominative data w.r.t. A and $loc_{/1}$, $loc_{/2}$, $loc_{/3}$, $loc_{/4}$ are mutually different. Then $\langle$factorial-inv(A, loc, n_0), factorial-loop-body(A, loc), factorial-inv$(A, loc, n_0)\rangle$ is an SFHT of $\mathrm{ND_{SC}}(V, A)$. The theorem is a consequence of (3), (4), and (5).

(8) $\langle\sim$ factorial-inv(A, loc, n_0), factorial-loop-body(A, loc), factorial-inv$(A, loc, n_0)\rangle$ is an SFHT of $\mathrm{ND_{SC}}(V, A)$.

(9) Suppose V is not empty and A is complex containing and V is without nonatomic nominative data w.r.t. A and $loc_{/1}$, $loc_{/2}$, $loc_{/3}$, $loc_{/4}$ are mutually different. Then $\langle$factorial-inv(A, loc, n_0), factorial-main-loop(A, loc), Equality$(A, loc_{/1}, loc_{/3})\wedge$factorial-inv$(A, loc, n_0)\rangle$ is an SFHT of $\mathrm{ND_{SC}}(V, A)$. The theorem is a consequence of (7) and (8).

(10) Suppose V is not empty and A is complex containing and V is without nonatomic nominative data w.r.t. A and $loc_{/1}$, $loc_{/2}$, $loc_{/3}$, $loc_{/4}$ are mutually different and loc and val are compatible w.r.t. 4 locations. Then $\langle$valid-factorial-input(V, A, val, n_0), factorial-main-part(A, loc, val), Equality$(A, loc_{/1}, loc_{/3})\wedge$factorial-inv$(A, loc, n_0)\rangle$ is an SFHT of $\mathrm{ND_{SC}}(V, A)$. The theorem is a consequence of (6) and (9).

(11) Suppose V is not empty and V is without nonatomic nominative data w.r.t. A and for every T, T is a value on $loc_{/1}$ and T is a value on $loc_{/3}$. Then Equality$(A, loc_{/1}, loc_{/3}) \wedge$ factorial-inv$(A, loc, n_0) \models$ $\mathrm{S_P}($valid-factorial-output$(A, z, n_0), (loc_{/4}) \Rightarrow_a, z)$.

PROOF: Set $i = loc_{/1}$. Set $j = loc_{/2}$. Set $n = loc_{/3}$. Set $s = loc_{/4}$. Set $D_4 = s \Rightarrow_a$. Consider d_1 being a non-atomic nominative data of V and A such that $d = d_1$ and $\{i, j, n, s\} \subseteq \operatorname{dom} d_1$ and $d_1(j) = 1$ and $d_1(n) = n_0$ and there exist natural numbers I, S such that $I = d_1(i)$ and $S = d_1(s)$ and $S = I!$. Reconsider $d_2 = d$ as a nominative data with simple names from

V and complex values from A. Set $L = d_2 \nabla_a^z D_4(d_2)$. n_0 and L constitute a valid output for the factorial w.r.t. A and z. $\square$

(12) Suppose V is not empty and V is without nonatomic nominative data w.r.t. A and for every T, T is a value on $loc_{/1}$ and T is a value on $loc_{/3}$. Then $\langle \text{Equality}(A, loc_{/1}, loc_{/3}) \wedge \text{factorial-inv}(A, loc, n_0), \text{Asg}^z((loc_{/4}) \Rightarrow_a)$, valid-factorial-output$(A, z, n_0)\rangle$ is an SFHT of $\text{ND}_{\text{SC}}(V, A)$. The theorem is a consequence of (11).

(13) Suppose for every T, T is a value on $loc_{/1}$ and T is a value on $loc_{/3}$. Then $\langle\sim (\text{Equality}(A, loc_{/1}, loc_{/3}) \wedge \text{factorial-inv}(A, loc, n_0)), \text{Asg}^z((loc_{/4}) \Rightarrow_a)$, valid-factorial-output$(A, z, n_0)\rangle$ is an SFHT of $\text{ND}_{\text{SC}}(V, A)$.

(14) Partial correctness of a FACTORIAL algorithm:
Suppose V is not empty and A is complex containing and V is without nonatomic nominative data w.r.t. A and $loc_{/1}$, $loc_{/2}$, $loc_{/3}$, $loc_{/4}$ are mutually different and loc and val are compatible w.r.t. 4 locations and for every T, T is a value on $loc_{/1}$ and T is a value on $loc_{/3}$. Then $\langle$valid-factorial-input(V, A, val, n_0), factorial-program(A, loc, val, z), valid-factorial-output$(A, z, n_0)\rangle$ is an SFHT of $\text{ND}_{\text{SC}}(V, A)$. The theorem is a consequence of (10), (12), and (13).

References

[1] Grzegorz Bancerek, Czesław Byliński, Adam Grabowski, Artur Korniłowicz, Roman Matuszewski, Adam Naumowicz, and Karol Pąk. The role of the Mizar Mathematical Library for interactive proof development in Mizar. *Journal of Automated Reasoning*, 61(1):9–32, 2018. doi:10.1007/s10817-017-9440-6.

[2] R.W. Floyd. Assigning meanings to programs. *Mathematical aspects of computer science*, 19(19–32), 1967.

[3] Adam Grabowski, Artur Korniłowicz, and Adam Naumowicz. Four decades of Mizar. *Journal of Automated Reasoning*, 55(3):191–198, 2015. doi:10.1007/s10817-015-9345-1.

[4] C.A.R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10): 576–580, 1969.

[5] Ievgen Ivanov and Mykola Nikitchenko. On the sequence rule for the Floyd-Hoare logic with partial pre- and post-conditions. In *Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops, Kyiv, Ukraine, May 14–17, 2018*, volume 2104 of *CEUR Workshop Proceedings*, pages 716–724, 2018.

[6] Ievgen Ivanov, Mykola Nikitchenko, Andrii Kryvolap, and Artur Korniłowicz. Simple-named complex-valued nominative data – definition and basic operations. *Formalized Mathematics*, 25(**3**):205–216, 2017. doi:10.1515/forma-2017-0020.

[7] Ievgen Ivanov, Artur Korniłowicz, and Mykola Nikitchenko. Implementation of the composition-nominative approach to program formalization in Mizar. *The Computer Science Journal of Moldova*, 26(1):59–76, 2018.

[8] Ievgen Ivanov, Artur Korniłowicz, and Mykola Nikitchenko. On an algorithmic algebra over simple-named complex-valued nominative data. *Formalized Mathematics*, 26(**2**):149–158, 2018. doi:10.2478/forma-2018-0012.

[9] Ievgen Ivanov, Artur Korniłowicz, and Mykola Nikitchenko. An inference system of an extension of Floyd-Hoare logic for partial predicates. *Formalized Mathematics*, 26(**2**): 159–164, 2018. doi:10.2478/forma-2018-0013.

[10] Ievgen Ivanov, Artur Korniłowicz, and Mykola Nikitchenko. On algebras of algorithms and specifications over uninterpreted data. *Formalized Mathematics*, 26(**2**):141–147, 2018. doi:10.2478/forma-2018-0011.

[11] Artur Kornilowicz, Andrii Kryvolap, Mykola Nikitchenko, and Ievgen Ivanov. Formalization of the algebra of nominative data in Mizar. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017, Prague, Czech Republic, September 3–6, 2017.*, pages 237–244, 2017. ISBN 978-83-946253-7-5. doi:10.15439/2017F301.

[12] Artur Kornilowicz, Andrii Kryvolap, Mykola Nikitchenko, and Ievgen Ivanov. Formalization of the nominative algorithmic algebra in Mizar. In Leszek Borzemski, Jerzy Świątek, and Zofia Wilimowska, editors, *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017 – Part II, Szklarska Poręba, Poland, September 17–19, 2017*, volume 656 of *Advances in Intelligent Systems and Computing*, pages 176–186. Springer, 2017. ISBN 978-3-319-67228-1. doi:10.1007/978-3-319-67229-8_16.

[13] Artur Korniłowicz, Andrii Kryvolap, Mykola Nikitchenko, and Ievgen Ivanov. An approach to formalization of an extension of Floyd-Hoare logic. In Vadim Ermolayev, Nick Bassiliades, Hans-Georg Fill, Vitaliy Yakovyna, Heinrich C. Mayr, Vyacheslav Kharchenko, Vladimir Peschanenko, Mariya Shyshkina, Mykola Nikitchenko, and Aleksander Spivakovsky, editors, *Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer, Kyiv, Ukraine, May 15–18, 2017*, volume 1844 of *CEUR Workshop Proceedings*, pages 504–523. CEUR-WS.org, 2017.

[14] Artur Korniłowicz, Ievgen Ivanov, and Mykola Nikitchenko. Kleene algebra of partial predicates. *Formalized Mathematics*, 26(**1**):11–20, 2018. doi:10.2478/forma-2018-0002.

[15] Andrii Kryvolap, Mykola Nikitchenko, and Wolfgang Schreiner. Extending Floyd-Hoare logic for partial pre- and postconditions. In Vadim Ermolayev, Heinrich C. Mayr, Mykola Nikitchenko, Aleksander Spivakovsky, and Grygoriy Zholtkevych, editors, *Information and Communication Technologies in Education, Research, and Industrial Applications: 9th International Conference, ICTERI 2013, Kherson, Ukraine, June 19–22, 2013, Revised Selected Papers*, pages 355–378. Springer International Publishing, 2013. ISBN 978-3-319-03998-5. doi:10.1007/978-3-319-03998-5_18.

[16] Volodymyr G. Skobelev, Mykola Nikitchenko, and Ievgen Ivanov. On algebraic properties of nominative data and functions. In Vadim Ermolayev, Heinrich C. Mayr, Mykola Nikitchenko, Aleksander Spivakovsky, and Grygoriy Zholtkevych, editors, *Information and Communication Technologies in Education, Research, and Industrial Applications – 10th International Conference, ICTERI 2014, Kherson, Ukraine, June 9–12, 2014, Revised Selected Papers*, volume 469 of *Communications in Computer and Information Science*, pages 117–138. Springer, 2014. ISBN 978-3-319-13205-1. doi:10.1007/978-3-319-13206-8_6.

Accepted May 27, 2019