The proof obligations of the project `mcfsi0-factorial1`

The factorial in four refinements: generation, comparison with Rodin, proof

Dominique Méry
Telecom Nancy, Université de Lorraine

November 23, 2026

dominique.mery@loria.fr

© Dominique Méry

Outline

1. The model
2. Overview of the obligations
3. Context `A-functions`
4. Machine `B-prepostok`
5. Machine `C-computing`
6. Machines `D-prealgo` and `E-algo`
7. Summary

Current Section Summary

1. The model
2. Overview of the obligations
3. Context `A-functions`
4. Machine `B-prepostok`
5. Machine `C-computing`
6. Machines `D-prealgo` and `E-algo`
7. Summary

The context `A-functions`: the factorial as the least relation

constants n_0, r_0, $factorial$

`req1`, `req2`	$n_0 \in \mathbb{N}, \quad r_0 \in \mathbb{Z}$
`axm2`	$factorial \in \mathbb{N} \leftrightarrow \mathbb{N}$
`axm3`	$0 \mapsto 1 \in factorial$
`axm4`	$\forall a, b \cdot a, b \in \mathbb{N} \wedge a \mapsto b \in factorial \Rightarrow a+1 \mapsto (a+1) * b \in factorial$
`axm5`	$\forall f \cdot f \in \mathbb{N} \leftrightarrow \mathbb{N} \wedge 0 \mapsto 1 \in f \wedge (\forall a, b \cdot \ldots \Rightarrow a+1 \mapsto (a+1) * b \in f) \Rightarrow factorial \subseteq f$

`th1`	$factorial \in \mathbb{N} \to \mathbb{N}$
`th2`	$factorial(0) = 1$
`th3`	$\forall u \cdot u \in \mathbb{N} \wedge u \neq 0 \Rightarrow factorial(u) = u * factorial(u-1)$

$factorial$ is defined as the least relation closed under `axm3`, `axm4` (`axm5` is the induction principle); the theorems derive a total function and its equations.

© Dominique Méry

The contract B-prepostok and its refinement C-computing

B-prepostok

variables r, n, ok

inv4 $n = n_0$

inv5 $ok = \text{TRUE} \Rightarrow r = factorial(n)$

INITIALISATION

$r :\in \mathbb{Z} \parallel ok := \text{FALSE} \parallel n := n_0$

computing1 when $ok = \text{FALSE}$ then

$ok := \text{TRUE} \parallel r := factorial(n)$

C-computing refines B-prepostok

variable fac: the history of the computed values

inv1 $fac \in \mathbb{N} \nrightarrow \mathbb{N}$ inv6 $\text{dom}(fac) = 0 \,..\, k$

inv4 $\forall i \cdot i \in \text{dom}(fac) \Rightarrow fac(i) = factorial(i)$

th3 (thm) $fac \subseteq factorial$

computing2 when $n \in \text{dom}(fac)$ then

$r := fac(n) \parallel ok := \text{TRUE}$

step2 (convergent) when $n \notin \text{dom}(fac)$ then $fac(k+1) := (k+1) * fac(k) \parallel k := k+1$

VARIANT $0 \,..\, n \setminus \text{dom}(fac)$ (a set)

The refinements D-prealgo and E-algo

D-prealgo refines C-computing

new variable $ffac$ inv2 $ffac = fac(k)$
inv5 $ok = \text{TRUE} \Rightarrow k = n$

computing3 when $k = n$ then
$r := ffac \parallel ok := \text{TRUE}$

step3 when $k \neq n$
then $fac(k+1) := (k+1) * fac(k)$
$\parallel k := k+1 \parallel ffac := (k+1) * ffac$

VARIANT vrn1 $n - k$

E-algo refines D-prealgo

fac disappears: variables r, k, n, $ffac$, ok

computing4 when $k = n$ then
$r := ffac \parallel ok := \text{TRUE}$

step4 when $k \neq n$
then $k := k+1 \parallel ffac := (k+1) * ffac$

theorems th1–th5, among which
th5 $ok = \text{TRUE} \Rightarrow r = factorial(n)$

E-algo is the algorithm: $ffac$ computes $k!$ without the history fac.

Current Section Summary

1. The model
2. Overview of the obligations
3. Context `A-functions`
4. Machine `B-prepostok`
5. Machine `C-computing`
6. Machines `D-prealgo` and `E-algo`
7. Summary

The kinds of obligations

kind	rule	what must be proved
`WD`	R1, R8	a formula is well defined (domain of the applied functions)
`THM`	R2, R3	a theorem of the context or of the machine follows from the axioms, invariants
`INV`	R4, R6	the initialisation or an event preserves an invariant
`FIS`	R5	a non-deterministic action ($r :\in \mathbb{Z}$) is feasible
`SIM`	R13	the concrete action simulates the abstract action
`GRD`	R11	the concrete guard strengthens the abstract guard
`VAR`	R10	a convergent event decreases the variant (here, a set)
`FIN`	R9	the set variant is finite (once per machine)

56 obligations

component	WD	THM	INV	FIS	SIM	GRD	VAR	FIN	total	Rodin	solvers
`A-functions`	2	3	–	–	–	–	–	–	5	1/5	4/5
`B-prepostok`	2	–	3	1	–	–	–	–	6	6/6	6/6
`C-computing`	3	3	15	–	1	–	1	1	24	24/24	23/24
`D-prealgo`	2	2	5	–	1	2	–	–	12	12/12	12/12
`E-algo`	2	5	–	–	2	–	–	–	9	9/9	9/9
total	11	13	23	1	4	2	1	1	**56**	**52/56**	**54/56**

- **Rodin**: same names and same goals for the 56 POs (`--comparer`); 52 discharged, 3 of them by hand; 4 theorems of the context are not.
- z3 proves 52 of them, Alt-Ergo 2 more; **`A-functions/th1/THM`** (unknown) and **`FIN`** (not translated) remain.

© Dominique Méry

Current Section Summary

1. The model
2. Overview of the obligations
3. Context `A-functions`
4. Machine `B-prepostok`
5. Machine `C-computing`
6. Machines `D-prealgo` and `E-algo`
7. Summary

A-functions: the theorems of the factorial

Hypotheses: `req1`, `req2`, `axm2`–`axm5`, and the preceding theorems.

obligation	goal	Rodin / solvers
th1/THM	$factorial \in \mathbb{N} \to \mathbb{N}$	not discharged / unknown
th2/WD	$0 \in \mathrm{dom}(factorial) \wedge factorial \in \mathbb{Z} \pfun \mathbb{Z}$	discharged / z3
th2/THM	$factorial(0) = 1$	not discharged / z3
th3/WD	$\forall u \cdot u \in \mathbb{N} \wedge u \neq 0 \Rightarrow u \in \mathrm{dom}(factorial) \wedge \ldots$	not discharged / z3
th3/THM	$\forall u \cdot \ldots \Rightarrow factorial(u) = u * factorial(u-1)$	not discharged / z3

- **th1** requires a **proof by induction**: `axm5` quantifies over the relations f (higher order); the solvers do not receive `axm2`, `axm5` (hypotheses not translated).
- z3 proves **th2**, **th3** *assuming* **th1**, as Rodin does: the whole development relies on **th1**.

Current Section Summary

1. The model
2. Overview of the obligations
3. Context `A-functions`
4. Machine `B-prepostok`
5. Machine `C-computing`
6. Machines `D-prealgo` and `E-algo`
7. Summary

B-prepostok: the contract

Hypotheses: the context (axioms and theorems), the invariants $r \in \mathbb{Z}$, $n \in \mathbb{Z}$, $ok \in \mathrm{BOOL}$, $n = n_0$.

obligation	goal	remark
`inv5/WD`	$ok = \mathrm{TRUE} \Rightarrow n \in \mathrm{dom}(factorial) \wedge \ldots$	
`INITIALISATION/act1/FIS`	$\mathbb{Z} \neq \emptyset$	$r :\in \mathbb{Z}$
`INITIALISATION/inv4/INV`	$n_0 = n_0$	
`INITIALISATION/inv5/INV`	$\mathrm{FALSE} = \mathrm{TRUE} \Rightarrow r' = factorial(n_0)$	$ok' = \mathrm{FALSE}$
`computing1/act2/WD`*	$n \in \mathrm{dom}(factorial) \wedge factorial \in \mathbb{Z} \nrightarrow \mathbb{Z}$	$r := factorial(n)$
`computing1/inv5/INV`	$\mathrm{TRUE} = \mathrm{TRUE} \Rightarrow factorial(n) = factorial(n)$	

* proved by hand in Rodin, automatically by z3 (with `th1`).

Current Section Summary

1. The model
2. Overview of the obligations
3. Context `A-functions`
4. Machine `B-prepostok`
5. Machine `C-computing`
6. Machines `D-prealgo` and `E-algo`
7. Summary

C-computing: the invariants of the history fac

Initialisation ($fac := \{0 \mapsto 1\}$, $k := 0$): 7

INV

inv1	$\{0 \mapsto 1\} \in \mathbb{N} \nrightarrow \mathbb{N}$
inv2	$\mathrm{dom}(\{0 \mapsto 1\}) \subseteq 0 \mathinner{..} n_0$
inv3	$\mathrm{dom}(\{0 \mapsto 1\}) \neq \emptyset$
inv4	$\forall i \cdot i \in \mathrm{dom}(\{0 \mapsto 1\}) \Rightarrow \ldots$
inv5	$0 \in \mathrm{dom}(\{0 \mapsto 1\})$
inv6	$\mathrm{dom}(\{0 \mapsto 1\}) = 0 \mathinner{..} 0$
inv8	$\mathrm{FALSE} = \mathrm{TRUE} \Rightarrow \ldots$

step2 ($fac' = fac \lhd \{k+1 \mapsto (k+1) * fac(k)\}$): 7

INV

inv1	$fac' \in \mathbb{N} \nrightarrow \mathbb{N}$
inv2	$\mathrm{dom}(fac') \subseteq 0 \mathinner{..} n$
inv3	$\mathrm{dom}(fac') \neq \emptyset$
inv4*	$\forall i \cdot i \in \mathrm{dom}(fac') \Rightarrow fac'(i) = factorial(i)$
inv5	$k+1 \in \mathrm{dom}(fac')$
inv6	$\mathrm{dom}(fac') = 0 \mathinner{..} k+1$
inv8	$ok = \mathrm{TRUE} \Rightarrow n \in \mathrm{dom}(fac')$

Plus **computing2/inv8/INV**: $n \in \mathrm{dom}(fac)$ (the guard). All proved by z3, except *__step2/inv4/INV__: **by hand** in Rodin, by **Alt-Ergo**.

C-computing: the central PO `step2/inv4/INV`

`step2/inv4/INV`

`context`	`axm3`, `axm4`, `th1`–`th3`: the equations of $factorial$
`C.inv4`	$\forall i \cdot i \in \mathrm{dom}(fac) \Rightarrow fac(i) = factorial(i)$
`C.inv6`	$\mathrm{dom}(fac) = 0 \mathinner{..} k$
`step2.grd13`	$n \notin \mathrm{dom}(fac)$

$$\vdash \quad \forall i \cdot i \in \mathrm{dom}(fac') \Rightarrow fac'(i) = factorial(i)$$

where $fac' = fac \lhd \{k+1 \mapsto (k+1) * fac(k)\}$ (the value after `step2`).

- For $i \leq k$: $fac(i) = factorial(i)$ by `inv4`.
- For $i = k+1$: $(k+1) * fac(k) = (k+1) * factorial(k) = factorial(k+1)$ by `th3`.
- This is the heart of the correctness; Rodin and z3 fail on their own, Alt-Ergo succeeds.

C-computing: well-definedness, theorems, simulation, variant

obligation	goal	status
`inv4/WD`, `computing2/act1/WD`	$i \in \mathrm{dom}(fac)$, $n \in \mathrm{dom}(fac)$	z3
`step2/act11/WD`	$k \in \mathrm{dom}(fac)$	z3
`th1/THM`, `th2/THM`	$n = n_0, \ \ k \in 0 \mathinner{..} n_0$	z3
`th3/THM`*	$fac \subseteq factorial$	z3
`computing2/act2/SIM`	$fac(n) = factorial(n)$	z3
`step2/VAR`	$0 \mathinner{..} n \setminus \mathrm{dom}(fac') \subset 0 \mathinner{..} n \setminus \mathrm{dom}(fac)$	Alt-Ergo
`FIN`	$\mathrm{finite}(0 \mathinner{..} n \setminus \mathrm{dom}(fac))$	not translated

- * `th3/THM`: proved by hand in Rodin.
- The variant is a *set*: `step2/VAR` requires a strict inclusion ($k + 1 \in 0 \mathinner{..} n$ leaves the set), `FIN` its finiteness.
- `FIN` holds (a subset of an interval is finite) but finite is not translated into SMT-LIB.

Current Section Summary

1. The model
2. Overview of the obligations
3. Context `A-functions`
4. Machine `B-prepostok`
5. Machine `C-computing`
6. Machines `D-prealgo` and `E-algo`
7. Summary

D-prealgo: the current value $ffac$

obligation	goal	remark
`inv2/WD`, `inv4/WD`	$k \in \mathrm{dom}(fac)$, …	
`inv3/THM`, `inv4/THM`	$\mathrm{dom}(fac) = 0 \mathinner{..} k, \quad fac(i) = factorial(i)$	kept from `C`
`INITIALISATION/inv2/INV`	$1 = \{0 \mapsto 1\}(0)$	
`INITIALISATION/inv5/INV`	$\mathrm{FALSE} = \mathrm{TRUE} \Rightarrow 0 = n_0$	
`computing3/inv5/INV`	$\mathrm{TRUE} = \mathrm{TRUE} \Rightarrow k = n$	guard $k = n$
`computing3/grd1/GRD`	$n \in \mathrm{dom}(fac)$	$k = n$, $\mathrm{dom}(fac) = 0 \mathinner{..} k$
`computing3/act1/SIM`	$ffac = fac(n)$	`inv2`, $k = n$
`step3/inv2/INV`	$(k+1) * ffac = (fac \lhd \{\ldots\})(k+1)$	
`step3/inv5/INV`	$ok = \mathrm{TRUE} \Rightarrow k + 1 = n$	$ok = \mathrm{FALSE}$
`step3/grd13/GRD`	$n \notin \mathrm{dom}(fac)$	$k \neq n$

12 POs, all discharged by Rodin and by z3. The variant `vrn1` $n - k$ yields *no* PO: `step3` is ordinary (it refines the convergent event `step2`).

© Dominique Méry

E-algo: the algorithm and its postcondition

obligation	goal	remark
`th1/WD`, `th1/THM`	$k \in \mathrm{dom}(fac),\ \ ffac = fac(k)$	fac: abstract variable
`th2/THM`	$k \in 0 \,..\, n$	
`th3/THM`	$n = n_0$	
`th4/THM`	$ok = \mathrm{TRUE} \Rightarrow k = n$	
`th5/WD`, `th5/THM`	$ok = \mathrm{TRUE} \Rightarrow r = factorial(n)$	the postcondition
`INITIALISATION/act3/SIM`	$0 = 0$	$k := 0$
`step4/act2/SIM`	$k + 1 = k + 1$	$k := k + 1$

- 9 POs, all discharged by Rodin and by z3.
- `step4` no longer assigns fac: the theorems relate $ffac$ to the abstract variable fac, and `th5` gives back the contract of `B-prepostok`.

© Dominique Méry

Current Section Summary

1. The model
2. Overview of the obligations
3. Context `A-functions`
4. Machine `B-prepostok`
5. Machine `C-computing`
6. Machines `D-prealgo` and `E-algo`
7. Summary

Summary and remarks

- 56 POs: 11 **WD**, 13 **THM**, 23 **INV**, 1 **FIS**, 4 **SIM**, 2 **GRD**, 1 **VAR**, 1 **FIN**;
- Rodin: 52 discharged (3 by hand: **computing1/act2/WD**, **C.th3/THM**, **step2/inv4/INV**); solvers: 54 (z3 52, Alt-Ergo 2).
- **Open point:** **A-functions/th1/THM** (*factorial* is a total function). Discharged neither by Rodin nor by the solvers, it requires an induction with `axm5`; everything else uses it as a hypothesis. To be done in Rodin (interactive proof) or with Isabelle/Rocq (`make prouver MODEL=mcfsi0-factorial1`).
- **th2/THM**, **th3/WD**, **th3/THM** are not discharged by Rodin, but z3 proves them from **th1**.
- **FIN** remains to be proved outside SMT (finiteness of a subset of $0 \,..\, n$).

© Dominique Méry