### 3.10.1 Factorial

The first example books on programming start with is that of factorial. Besides the fact that it is very elementary and easy to explain and understand, it may be used as a standard example for comparing tools and strategies for program verification.

The emphasis we want to put here is on the similarities of the program definition, the logical formulae expressed as the verification conditions and the properties of the mathematical function factorial.

Consider the program $Fact$, for computing the factorial function:

$$Fact[x] = \textbf{ If } x = 0 \textbf{ then } 1 \textbf{ else } x * Fact[x-1], \tag{3.177}$$

with the specification:

$$(\forall x)\ (I_{Fact}[x] \iff x \in \mathbb{N}),$$

$$(\forall x, y)\ (O_{Fact}[x, y] \iff y = x!).$$

Before starting with the essential part of the verification, we first check if $Fact$ is coherent with respect to its specification, the auxiliary programs and their specifications. In order to perform the coherence check, we instantiate the relevant conditions:

$$(\forall x : x \in \mathbb{N})\ (x = 0 \implies \mathbb{T}) \tag{3.178}$$

$$(\forall x : x \in \mathbb{N})\ (x \neq 0 \implies x - 1 \in \mathbb{N}) \tag{3.179}$$

$$(\forall x : x \in \mathbb{N})\ (x \neq 0 \implies \mathbb{T}) \tag{3.180}$$

$$(\forall x, y : x \in \mathbb{N})\ (x \neq 0 \wedge [x-1]! = y) \implies \mathbb{T}). \tag{3.181}$$

As we can see, the conditions (3.178), (3.180) and (3.181) are trivial to prove, because we have $\mathbb{T}$ at the right hand side of an implication. The origin of these $\mathbb{T}$ are the preconditions of the auxiliary functions $\lambda x.1$, the minus one function $\lambda x.x - 1$, and multiplication $\lambda x, y.x * y$.

In fact, only (3.179) requires a proofs, however, it is easily tractable in the theory of natural numbers.

After we are convinced that $Fact$ is coherent, we instantiate the relevant verification conditions for proving correctness:

$$(\forall x : x \in \mathbb{N})\ (x = 0 \implies 1 = x!) \tag{3.182}$$

$$(\forall x, y : x \in \mathbb{N})\ (x \neq 0 \wedge y = [x-1]! \implies x * y = x!) \tag{3.183}$$

We see, that (3.182), (3.183) and are tractable in the theory of natural numbers.

Now we need to prove the termination of $Fact$, that is:

$$(\forall x : x \in \mathbb{N})\ (Fact'[x] = \mathbb{T}) \tag{3.184}$$

where:

$$Fact'[x] = \textbf{ If } x = 0 \textbf{ then } \mathbb{T} \textbf{ else } Fact'[x-1]. \tag{3.185}$$

We have arrived to the most popular simplified version, namely the primitive recursive one and thus we are done.

### 3.10.2 Summation of a number

The next example we consider is the summation of a number:

$$\sum_{i=1}^{x} i.$$

The emphasis we want to put here is on the similarities of the verification conditions we obtain here and the verification conditions from the previous example (3.10.1).

In our opinion, many practical programs, when verifying, will have similar verification conditions. This fact could be considered as a research problem by the Mathematical Knowledge Management community.

Consider the program $Sum$, for computing the summation of a number function:

$$Sum[x] = \textbf{ If } x = 0 \textbf{ then } 0 \textbf{ else } x + Sum[x-1], \tag{3.186}$$

with the specification:

$$(\forall x)\ (I_{Sum}[x] \iff x \in \mathbb{N}),$$

$$(\forall x, y)\ (O_{Sum}[x, y] \iff y = \frac{x * (x+1)}{2}).$$

Even though, this program is suppose to compute the summation of a number, we provide here an alternative output specification.

Before starting with the essential part of the verification, we first check if $Sum$ is coherent with respect to its specification, the auxiliary programs and their specifications. In order to perform the coherence check, we instantiate the relevant conditions:

$$(\forall x : x \in \mathbb{N})\ (x = 0 \implies \mathbb{T}) \tag{3.187}$$

$$(\forall x : x \in \mathbb{N})\ (x \neq 0 \implies x - 1 \in \mathbb{N}) \tag{3.188}$$

$$(\forall x : x \in \mathbb{N})\ (x \neq 0 \implies \mathbb{T}) \tag{3.189}$$

$$(\forall x, y : x \in \mathbb{N})\ (x \neq 0 \wedge \frac{(x-1) * ((x-1)+1)}{2} = y) \implies \mathbb{T}). \tag{3.190}$$

As we can see, the conditions (3.187), (3.189) and (3.190) are trivial to prove, because we have $\mathbb{T}$ at the right hand side of an implication. The origin of these $\mathbb{T}$ are the preconditions of the auxiliary functions $\lambda x.0$, the minus one function $\lambda x.x - 1$, and addition $\lambda x, y.x + y$.

In fact, only (3.188) requires a proofs, however, it is easily tractable in the theory of natural numbers.

Note that in the previous example (3.10.1), we have the same condition (3.179).

After we are convinced that $Sum$ is coherent, we instantiate the relevant verification conditions for proving correctness:

$$(\forall x : x \in \mathbb{N})\ (x = 0\ \Longrightarrow 0 = \frac{x * (x + 1)}{2}) \tag{3.191}$$

$$(\forall x, y : x \in \mathbb{N})\ (x \neq 0 \wedge y = \frac{(x - 1) * ((x - 1) + 1)}{2}\ \Longrightarrow x + y = \frac{x * (x + 1)}{2}) \tag{3.192}$$

We see, that (3.191), (3.192) and are tractable in the theory of natural numbers.

Note that these are the specific verification conditions. In fact, the proofs of all the other verification conditions, that is, coherence and termination, may be reusable, because they are the same for different programs.

Now we need to prove the termination of $Sum$, that is:

$$(\forall x : x \in \mathbb{N})\ (Sum'[x] = \mathbb{T}) \tag{3.193}$$

where:

$$Sum'[x] = \textbf{ If } x = 0 \textbf{ then } \mathbb{T} \textbf{ else } Sum'[x - 1]. \tag{3.194}$$

We again arrived at the most popular simplified version, namely the primitive recursive one and thus we are done.

### 3.10.3 Floor of a real number

The next example we consider is the floor of a real number.

The purpose of showing this example is to demonstrate the ability of our method in domains different from $\mathbb{N}$. In fact, we do not give any restrictions on the possible domains on which the programs are executed and verified. For this particular example we take $\mathbb{R}$ as our domain.

Consider the program $Floor$, for computing the floor of a real nonnegative number:

$$Floor[x] = \textbf{ If } 0 \leq x < 1 \textbf{ then } 0 \textbf{ else } 1 + Floor[x-1], \tag{3.195}$$

with the specification:

$$(\forall x)\ (I_{Floor}[x] \iff x \in \mathbb{R} \ \wedge\ x \geq 0),$$

$$(\forall x, y)\ (O_{Floor}[x, y] \iff y \in \mathbb{N} \ \wedge\ x - y < 1 \ \wedge\ y \leq x).$$

Before starting with the essential part of the verification, we first check if $Floor$ is coherent with respect to its specification, the auxiliary programs and their specifications. In order to perform the coherence check, we instantiate the relevant conditions:

$$(\forall x : x \in \mathbb{R} \ \wedge\ x \geq 0)\ (0 \leq x < 1 \implies \mathbb{T}) \tag{3.196}$$

$$(\forall x : x \in \mathbb{R} \ \wedge\ x \geq 0)\ (\neg(0 \leq x < 1) \implies x - 1 \in \mathbb{R} \ \wedge\ x - 1 \geq 0) \tag{3.197}$$

$$(\forall x : x \in \mathbb{R} \ \wedge\ x \geq 0)\ (\neg(0 \leq x < 1) \implies \mathbb{T}) \tag{3.198}$$

$$(\forall x : x \in \mathbb{R} \wedge x \geq 0)\ (\neg(0 \leq x < 1) \wedge y \in \mathbb{N} \wedge (x-1) - y < 1 \wedge y \leq (x-1)) \implies \mathbb{T}). \tag{3.199}$$

As we can see, the conditions (3.196), (3.198) and (3.199) are trivial to prove, because we have $\mathbb{T}$ at the right hand side of an implication. The origin of these $\mathbb{T}$ are the preconditions of the auxiliary functions $\lambda x.0$, the minus one function $\lambda x.x - 1$, and the plus one function $\lambda x.1 + x$.

In fact, only (3.197) requires a proofs, however, it is easily tractable in the theory of real numbers.

After we are convinced that $Floor$ is coherent, we instantiate the relevant verification conditions for proving correctness:

$$(\forall x : x \in \mathbb{R} \ \wedge\ x \geq 0)\ (0 \leq x < 1 \implies 0 \in \mathbb{N} \ \wedge\ x - 0 < 1 \ \wedge\ 0 \leq x) \tag{3.200}$$

$$(\forall x : x \in \mathbb{R} \ \wedge\ x \geq 0)\ (\neg(0 \leq x < 1) \wedge y \in \mathbb{N} \ \wedge\ (x-1) - y < 1 \ \wedge\ y \leq (x-1) \tag{3.201}$$

$$\implies$$

$$1 + y \in \mathbb{N} \ \wedge\ x - (1+y) < 1 \ \wedge\ (1+y) \leq x).$$

We see, that (3.200) and (3.201) are tractable in the theory of real numbers.
Now we need to prove the termination of $Floor$, that is:

$$(\forall x : x \in \mathbb{R} \ \wedge \ x \geq 0) \ (Floor'[x] = \mathbb{T}) \tag{3.202}$$

where:

$$Floor'[x] = \ \textbf{If} \ 0 \leq x < 1 \ \textbf{then} \ \mathbb{T} \ \textbf{else} \ Floor'[x - 1]. \tag{3.203}$$

This is a new simplified version, and one has to prove its termination, which easily possible by induction. We split the $\mathbb{R}^+$ into intervals with length one and then perform normal induction over the naturals.

### 3.10.4 A wrong version of Floor

The next example we consider is a wrong version of the floor function. We introduce here a bug in order to explore the verification conditions in such a situation. Moreover, a distinctive feature of our approach is the hint on "what is wrong" in case of a verification failure.

Consider the program $WrFloor$, for computing the floor of a real nonnegative number:

$$WrFloor[x] = \textbf{ If } 0 \leq x < 1 \textbf{ then } 5 \textbf{ else } 1 + WrFloor[x-1], \tag{3.204}$$

with the specification:

$$(\forall x)\ (I_{WrFloor}[x] \iff x \in \mathbb{R} \ \wedge\ x \geq 0),$$

$$(\forall x, y)\ (O_{WrFloor}[x, y] \iff y \in \mathbb{N} \ \wedge\ x - y < 1 \ \wedge\ y \leq x).$$

The the specification of $WrFloor$ is same as the $Floor$, presented in (3.10.3), however, in the definition of $WrFloor$ we introduced a bug, namely when $0 \leq x < 1$, $WrFloor[x] = 5$, in contrast to $Floor[x] = 0$.

Before starting with the essential part of the verification, we first check if $WrFloor$ is coherent with respect to its specification, the auxiliary programs and their specifications. In order to perform the coherence check, we instantiate the relevant conditions:

$$(\forall x : x \in \mathbb{R} \ \wedge\ x \geq 0)\ (0 \leq x < 1 \implies \mathbb{T}) \tag{3.205}$$

$$(\forall x : x \in \mathbb{R} \ \wedge\ x \geq 0)\ (\neg(0 \leq x < 1) \implies x - 1 \in \mathbb{R} \ \wedge\ x - 1 \geq 0) \tag{3.206}$$

$$(\forall x : x \in \mathbb{R} \ \wedge\ x \geq 0)\ (\neg(0 \leq x < 1) \implies \mathbb{T}) \tag{3.207}$$

$$(\forall x : x \in \mathbb{R} \wedge x \geq 0)\ (\neg(0 \leq x < 1) \wedge y \in \mathbb{N} \wedge (x-1) - y < 1 \wedge y \leq (x-1)) \implies \mathbb{T}). \tag{3.208}$$

As we can see, the conditions (3.205), (3.207) and (3.208) are trivial to prove, because we have $\mathbb{T}$ at the right hand side of an implication. The origin of these $\mathbb{T}$ are the preconditions of the auxiliary functions $\lambda x.5$, the minus one function $\lambda x.x - 1$, and the plus one function $\lambda x.1 + x$.

Only (3.206) requires a proofs, however, it is easily tractable in the theory of real numbers.

In fact, all the conditions here are the same as in the correct version of $Floor$ (3.10.3).

After we are convinced that $WrFloor$ is coherent, we instantiate the relevant verification conditions for proving correctness:

$$(\forall x : x \in \mathbb{R} \ \wedge\ x \geq 0)\ (0 \leq x < 1 \implies 5 \in \mathbb{N} \ \wedge\ x - 5 < 1 \ \wedge\ 5 \leq x) \tag{3.209}$$

$$(\forall x : x \in \mathbb{R} \ \wedge\ x \geq 0)\ (\neg(0 \leq x < 1) \wedge y \in \mathbb{N} \ \wedge\ (x-1) - y < 1 \ \wedge\ y \leq (x-1) \tag{3.210}$$

$$\Longrightarrow$$

$$1 + y \in \mathbb{N} \ \wedge \ x - (1 + y) < 1 \ \wedge \ (1 + y) \leq x).$$

$$(\forall x : x \in \mathbb{R} \ \wedge \ x \geq 0) \, (WrFloor'[x] = \mathbb{T}) \tag{3.211}$$

where:

$$WrFloor'[x] = \ \textbf{If} \, 0 \leq x < 1 \, \textbf{then} \, \mathbb{T} \, \textbf{else} \, WrFloor'[x - 1]. \tag{3.212}$$

We see, that (3.210), (3.211) are tractable in the theory of real numbers and they are the same as in the correct version of $Floor$.

Now, for this buggy version of $WrFloor$ we see that all the verification conditions remain the same, except one, namely, (3.209). Therefore, according to the *completeness* of the method, we conclude that the program $WrFloor$ does not satisfy its specification.

Furthermore, in order to demonstrate how a bug might be located in an automatic manner, we have a broad discussion on that topic in (3.10.7).

### 3.10.5 Remainder Rem in division of integers

In arithmetic, when the result of the division of two integers cannot be expressed with an integer quotient, the remainder is the amount "left over." The next example we consider is the remainder function $Rem$.

The purpose of showing this example is to demonstrate the ability of our method in vector domains, that is, the arguments $x, y, z$ could be not only single variables but vectors (tuples) as well.

Consider the program $Rem$, for computing the remainder of the division of two natrurals:

$$Rem[x, y] = \textbf{ If } x < y \textbf{ then } x \textbf{ else } Rem[x - y, y], \tag{3.213}$$

with the specification:

$$(\forall x, y)\ (I_{Rem}[x, y] \iff x \in \mathbb{N} \ \wedge y \in \mathbb{N}^+),$$

$$(\forall x, y, z)\ (O_{Rem}[x, y, z] \iff (\exists q : q \in \mathbb{N})\ (x = z + y * q \ \wedge \ z < y)).$$

Before starting with the essential part of the verification, we first check if $Rem$ is coherent with respect to its specification, the auxiliary programs and their specifications. In order to perform the coherence check, we instantiate the relevant conditions:

$$(\forall x, y : \ x \in \mathbb{N} \ \wedge \ y \in \mathbb{N}^+)\ (x < y \implies \mathbb{T}) \tag{3.214}$$

$$(\forall x, y : \ x \in \mathbb{N} \ \wedge \ y \in \mathbb{N}^+)\ (\neg(x < y) \implies x - y \in \mathbb{N} \ \wedge \ y \in \mathbb{N}^+) \tag{3.215}$$

$$(\forall x, y : \ x \in \mathbb{N} \ \wedge \ y \in \mathbb{N}^+)\ (\neg(x < y) \implies \mathbb{T}) \tag{3.216}$$

$$(\forall x, y, z : \ x \in \mathbb{N} \wedge y \in \mathbb{N}^+)\ (\neg(x < y) \wedge (\exists q : q \in \mathbb{N})\ (x - y = z + y * q \wedge z < y)) \implies \mathbb{T}). \tag{3.217}$$

As we can see, the conditions (3.214), (3.216) and (3.217) are trivial to prove, because we have $\mathbb{T}$ at the right hand side of an implication. The origin of these $\mathbb{T}$ are the preconditions of the auxiliary functions: the identity $\lambda x.x$, the minus function $\lambda x, y.x - y$, and the projection $\lambda x, y.y$.

In fact, only (3.215) requires a proofs, however, it is easily tractable in the theory of natural numbers.

After we are convinced that $Rem$ is coherent, we instantiate the relevant verification conditions for proving correctness:

$$(\forall x, y : \ x \in \mathbb{N} \ \wedge \ y \in \mathbb{N}^+)\ (x < y \implies (\exists q : q \in \mathbb{N})\ (x = x + y * q \ \wedge \ x < y)) \tag{3.218}$$

$$(\forall x, y, z : \ x \in \mathbb{N} \ \wedge \ y \in \mathbb{N}^+)\ (\neg(x < y) \wedge (\exists q : q \in \mathbb{N})\ (x - y = z + y * q \ \wedge \ z < y) \tag{3.219}$$

$$\Longrightarrow$$

$$(\exists q : q \in \mathbb{N})\ (x = z + y * q\ \wedge\ z < y)).$$

We see, that (3.218) and (3.219) are tractable in the theory of natural numbers.
Now we need to prove the termination of $Rem$, that is:

$$(\forall x, y :\ x \in \mathbb{N}\ \wedge y \in \mathbb{N}^{+})\ (Rem'[x, y] = \mathbb{T}) \tag{3.220}$$

where:

$$Rem'[x, y] =\ \textbf{If}\ x < y\ \textbf{then}\ \mathbb{T}\ \textbf{else}\ Rem'[x - y, y]. \tag{3.221}$$

This is a new simplified version, and one has to prove its termination. For that proof, we would suggest to use induction on $q$, where $x = Rem[x, y] + y * q$ for any $x$ and $y$: $x \in \mathbb{N}\ \wedge\ y \in \mathbb{N}^{+}$.