I217: Functional Programming

11. Program Verification – Natural Numbers

Kazuhiro Ogata, Canh Minh Do

Roadmap

- Natural Numbers a la Peano
- Associativity of _+_
- Commutativity of _+_
- Associativity of _*_
- Commutativity of _*_
- Correctness of a Tail Recursive Factorial

Natural Numbers a la Peano

Natural numbers have been formalized by Giuseppe Peano (1858 – 1932), an Italian mathematician.

Inductively defined as follows:

(1) 0 is a natural number.

(2) If n is a natural number, then the successor of n, denoted $s(n)$, is also a natural number.

0	s(0)	s(s(0))	s(s(s(0)))	s(s(s(s(0))))
0	1	2	3	4

Natural Numbers a la Peano

Natural numbers can be specified in CafeOBJ as follows:

```
mod! PNAT1 {
  [PNat]
  op 0 : -> PNat {constr} .
  op s : PNat -> PNat {constr} .
  ...
}
```

Terms 0, s(0), s(s(0)), s(s(s(0))), s(s(s(s(0)))) denote 0, 1, 2, 3, 4.

Natural Numbers a la Peano

For every sort S, the following operator and equations are prepared in the built-in module EQL that is imported by BOOL:

op _=_ : S S -> Bool {**comm**} .
eq (X = X) = true .
eq (true = false) = false .

where X is a variable of S.

We declare the following equations in PNAT1 for PNat:

eq (0 = s(Y)) = false .
eq (s(X) = s(Y)) = (X = Y) .

where X and Y are variables of PNat.

Let X, Y and Z be variables of PNat in the rest of the slides.

Natural Numbers a la Peano

Addition of natural numbers is defined as follows:

op _+_ : PNat PNat -> PNat .
eq 0 + Y = Y . -- (+1)
eq s(X) + Y = s(X + Y) . -- (+2)

$\underline{s(s(s(0))) + s(s(s(s(0))))}$
$\to s(\underline{s(s(0)) + s(s(s(s(0))))})$ by (+2)
$\to s(s(\underline{s(0) + s(s(s(s(0))))}))$ by (+2)
$\to s(s(s(\underline{0 + s(s(s(s(0))))})))$ by (+2)
$\to s(s(s(s(s(s(s(0)))))))$ by (+1)

Natural Numbers a la Peano

Multiplication of natural numbers is defined as follows:

```
op _*_ : PNat PNat -> PNat .
eq 0 * Y = 0 .                -- (*1)
eq s(X) * Y = (X * Y) + Y .   -- (*2)
```

```
s(s(0)) * s(s(0))
→ (s(0) * s(s(0))) + s(s(0))          by (*2)
→ ((0 * s(s(0))) + s(s(0))) + s(s(0))  by (*2)
→ (0 + s(s(0))) + s(s(0))              by (*1)
→ s(s(0)) + s(s(0))                    by (+1)
→ s(s(0) + s(s(0)))                    by (+2)
→ s(s(0 + s(s(0))))                    by (+2)
→ s(s(s(s(0))))                        by (+1)
```


Natural Numbers a la Peano

Factorial function can be defined as follows:

```
op fact1 : PNat -> PNat .
eq fact1(0) = s(0) .                 -- (f1-1)
eq fact1(s(X)) = s(X) * fact1(X) .   -- (f1-2)
```

Another implementation (a tail recursive version) of Factorial function is as follows:

```
op fact2 : PNat -> PNat .
op sfact2 : PNat PNat -> PNat .
eq fact2(X) = sfact2(X,s(0)) .           -- (f2)
eq sfact2(0,Y) = Y .                     -- (sf2-1)
eq sfact2(s(X),Y) = sfact2(X,s(X) * Y) . -- (sf2-2)
```

Natural Numbers a la Peano

Let $l(X)$ and $r(X)$ be terms of a same sort (say PNat) in which a variable X of PNat occurs. Then, the following (A) and (B) are equivalent:

(A) $(\forall X{:}PNat)\ l(X) = r(X)$

(B) I. $l(0) = r(0)$

II. If $l(x) = r(x)$, then $l(s(x)) = r(s(x))$, where x is a fresh constant of PNat.

It suffices to prove (B) so as to prove (A). This is called proof by *structural induction* on natural number X. I is called the *base case*, II is called the *induction case*, and $l(x) = r(x)$ is called the *induction hypothesis*.

Associativity of _+_

$(X + Y) + Z$ equals $X + (Y + Z)$ for all natural numbers X,Y,Z. This is what we will prove by structural induction on X.

Theorem 1 [Associativity of _+_ (assoc+)]
$(\forall X{:}PNat)\ (\forall Y,Z{:}PNat)\ ((X + Y) + Z = X + (Y + Z))$

In the rest of the slides, the universal quantifiers, such as $(\forall X{:}PNat)$ and $(\forall Y,Z{:}PNat)$, are omitted, and the above formula is written as follows:

$$(X + Y) + Z = X + (Y + Z)$$

Associativity of _+_

Proof of Theorem 1 By structural induction on X.

Let x,y,z be fresh constants of PNat.

I. Base case

What to show is $(0 + y) + z = 0 + (y + z)$.

$(0 + y) + z \to y + z$ by (+1) $\quad$ $0 + (y + z) \to y + z$ by (+1)

II. Induction case

What to show is $(s(x) + y) + z = s(x) + (y + z)$

assuming the induction hypothesis $(x + Y) + Z = x + (Y + Z)$ -- (IH)

$(s(x) + y) + z \to s(x + y) + z$ by (+2)

$\to s((x + y) + z)$ by (+2)

$\to s(x + (y + z))$ by (IH)

$s(x) + (y + z) \to s(x + (y + z))$ by (+2)

End of Proof of Theorem 1

Associativity of _+_

Part of the proof is written in CafeOBJ. Proofs written in CafeOBJ are called *proof scores*.

Proof of Theorem 1 By structural induction on X.

I. Base case

```
open PNAT1 .
  -- fresh constants
  ops y z : -> PNat .
  -- check
  red (0 + y) + z = 0 + (y + z) .
close
```

II. Induction case

```
open PNAT1 .
  -- fresh constants
  ops x y z : -> PNat .
  -- IH
  eq (x + Y) + Z = x + (Y + Z) .
  -- check
  red (s(x) + y) + z = s(x) + (y + z) .
close
```

End of Proof of Theorem 1 Please see the appendices for the proof score.

Appendices

The proof score of Theorem 1 is as follows:

```
"Theorem 1 [associativity of _+_ (assoc+)]
(X + Y) + Z = X + (Y + Z)

Proof of Theorem 1. By structural induction on X.
I. Base case"

open PNAT1 .
  -- fresh constants
  ops y z : -> PNat .
  -- check
  red (0 + y) + z = 0 + (y + z) .
close
```


Appendices

```
"II. Induction case"

open PNAT1 .
  -- fresh constants
  ops x y z : -> PNat .
  -- IH
  eq (x + Y) + Z = x + (Y + Z) .
  -- check
  red (s(x) + y) + z = s(x) + (y + z) .
close

"End of Proof of Theorem 1"
```

Commutativity of _+_

Theorem 2 [Commutativity of _+_ (comm+)] $X + Y = Y + X$

Proof of Theorem 2 By structural induction on X.

Let x,y be fresh constants of PNat.

I. Base case

What to show is $0 + y = y + 0$.

$\underline{0 + y} \rightarrow y$ by (+1)

Since $y + 0$ cannot be rewritten, we need a lemma that makes it possible to conclude that $y + 0$ equals y. We conjecture the following lemma:

$\boxed{X + 0 = X \quad \text{-- (rz+)}}$

$\underline{y + 0} \rightarrow y$ by (rz+)

(The lemma will be proved later.)

Commutativity of _+_

II. Induction case

What to show is $s(x) + y = y + s(x)$

assuming the induction hypothesis $x + Y = Y + x$ -- (IH)

$\underline{s(x) + y} \rightarrow s(\underline{x + y})$ by (+2)

$\rightarrow s(y + x)$ by (IH)

$y + s(x)$ cannot be rewritten, and then we need a lemma that makes it possible to conclude that $y + s(x)$ equals $s(y + x)$. We conjecture the following lemma:

$\boxed{X + s(Y) = s(X + Y) \quad \text{-- (rs+)}}$

$\underline{y + s(x)} \rightarrow s(y + x)$ by (rs+)

End of Proof of Theorem 2

(The lemma will be proved later.)

Commutativity of _+_

Lemma 1 [Right zero of _+_ (rz+)] $X + 0 = X$

Proof of Lemma 1 By structural induction on X.

Let x be fresh constants of PNat.

I. Base case

What to show is $0 + 0 = 0$.

$\underline{0 + 0} \rightarrow 0$ by (+1)

II. Induction case

What to show is $s(x) + 0 = s(x)$

assuming the induction hypothesis $x + 0 = x$ -- (IH)

$\underline{s(x) + 0} \rightarrow s(\underline{x + 0})$ by (+2)

$\rightarrow s(x)$ by (IH)

End of Proof of Lemma 1

Commutativity of _+_

Lemma 2 [Right successor of _+_ (rs+)] $X + s(Y) = s(X + Y)$

Proof of Lemma 2 By structural induction on X.

Let x,y be fresh constants of PNat.

I. Base case

What to show is $0 + s(y) = s(0 + y)$.

$\underline{0 + s(y)} \rightarrow s(y)$ by (+1) $s(\underline{0 + y}) \rightarrow s(y)$ by (+1)

II. Induction case

What to show is $s(x) + s(y) = s(s(x) + y)$

assuming the induction hypothesis $x + s(Y) = s(x + Y)$ -- (IH)

$\underline{s(x) + s(y)} \rightarrow s(\underline{x + s(y)})$ by (+2) $s(\underline{s(x) + y}) \rightarrow s(s(x + y))$ by (+2)

$\rightarrow s(s(x + y))$ by (IH)

End of Proof of Lemma 2

Commutativity of _+_

Lemma 1 [Right zero of _+_ (rz+)] X + 0 = X

Proof of Lemma 1 By structural induction on X.

I. Base case

```
open PNAT1 .
 -- check
 red 0 + 0 = 0 .
close
```

II. Induction case

```
open PNAT1 .
 -- fresh constants
 op x : -> PNat .
 -- IH
 eq x + 0 = x .
 -- check
 red s(x) + 0 = s(x) .
close
```

End of Proof of Lemma 1

Commutativity of _+_

Lemma 2 [Right successor of _+_ (rs+)] X + s(Y) = s(X + Y)

Proof of Lemma 2 By structural induction on X.

I. Base case

```
open PNAT1 .
 -- fresh constants
 op y : -> PNat .
 -- check
 red 0 + s(y) = s(0 + y) .
close
```

II. Induction case

```
open PNAT1 .
 -- fresh constants
 ops x y : -> PNat .
 -- IH
 eq x + s(Y) = s(x + Y) .
 -- check
 red s(x) + s(y) = s(s(x) + y) .
close
```

End of Proof of Lemma 2

Commutativity of _+_

Theorem 2 [Commutativity of _+_ (comm+)] X + Y = Y + X

Proof of Theorem 2 By structural induction on X.

I. Base case

```
open PNAT1 .
  -- fresh constants
  op y : -> PNat .
  -- lemmas
  eq X + 0 = X . -- (rz+)
  -- check
  red 0 + y = y + 0 .
close
```

II. Induction case

```
open PNAT1 .
  -- fresh constants
  ops x y : -> PNat .
  -- lemmas
  eq X + s(Y) = s(X + Y) . -- (rs+)
  -- IH
  eq x + Y = Y + x .
  -- check
  red s(x) + y = y + s(x) .
close
```

End of Proof of Theorem 2

Associativity of _*_

Theorem 3 [Associativity of _*_ (assoc*)] (X * Y) * Z = X * (Y * Z)

Proof of Theorem 3 By structural induction on X.

Let x,y,z be fresh constants of PNat.

I. Base case

What to show is (0 * y) * z = 0 * (y * z).

$(\underline{0 * y}) * z \rightarrow \underline{0 * z}$ by (*1)

$\rightarrow 0$ by (*1)

$\underline{0 * (y * z)} \rightarrow 0$ by (*1)

Associativity of _*_

II. Induction case

What to show is (s(x) * y) * z = s(x) * (y * z)
assuming the induction hypothesis (x * Y) * Z = x * (Y * Z) -- (IH)

(s(x) * y) * z → ((x * y) + y) * z by (*2)

s(x) * (y * z) → (x * (y * z)) + (y * z) by (*2)

We cannot make the two terms rewritten to a same term and then need a lemma to do so. It seems sufficient to conjecture the following one:

(X + Y) * Z = (X * Z) + (Y * Z) -- (d*o+)

((x * y) + y) * z → ((x * y) * z) + (y * z) by (d*o+)
→ (x * (y * z)) + (y * z) by (IH)

End of Proof of Theorem 3

Associativity of _*_

Lemma 3 [Distributive law of _*_ over _+_ (d*o+)]
(X + Y) * Z = (X * Z) + (Y * Z)

Proof of Lemma 3 By structural induction on X.

Let x,y,z be fresh constants of PNat.

I. Base case

What to show is (0 + y) * z = (0 * z) + (y * z).

(0 + y) * z → y * z by (+1)

(0 * z) + (y * z) → 0 + (y * z) by (*1)
→ y * z by (+1)

Associativity of _*_

II. Induction case

What to show is (s(x) + y) * z = (s(x) * z) + (y * z)
assuming the induction hypothesis
(x + Y) * Z = (x * Z) + (Y * Z) -- (IH)

$$
\begin{aligned}
\underline{(s(x) + y)} * z &\rightarrow \underline{s(x + y) * z} && \text{by } (+2)\\
&\rightarrow (\underline{(x + y) * z}) + z && \text{by } (*2)\\
&\rightarrow \underline{((x * z) + (y * z)) + z} && \text{by (IH)}\\
&\rightarrow (x * z) + (\underline{(y * z) + z}) && \text{by (assoc+)}\\
&\rightarrow (x * z) + (z + (y * z)) && \text{by (comm+)}
\end{aligned}
$$

$$
\begin{aligned}
(\underline{s(x) * z}) + (y * z) &\rightarrow \underline{((x * z) + z) + (y * z)} && \text{by } (*2)\\
&\rightarrow (x * z) + (z + (y * z)) && \text{by (assoc+)}
\end{aligned}
$$

<u>End of Proof of Lemma 3</u>

Associativity of _*_

<u>Lemma 3</u> [Distributive law of _*_ over _+_ (d*o+)]
(X + Y) * Z = (X * Z) + (Y * Z)
<u>Proof of Lemma 3</u> By structural induction on X.

I. Base case

```
open PNAT2 .
  -- fresh constants
  ops y z : -> PNat .
  -- check
  red (0 + y) * z = (0 * z) + (y * z) .
close
```

<u>End of Proof of Lemma 3</u>

II. Induction case

```
open PNAT2 .
  -- fresh constants
  ops x y z : -> PNat .
  -- IH
  eq (x + Y) * Z = (x * Z) + (Y * Z) .
  -- check
  red (s(x) + y) * z = (s(x) * z) + (y * z) .
close
```

Note that PNAT2 is PNAT1 in which _+_ is declared as follows:

```
op _+_ : PNat PNat -> PNat {assoc comm} .
```

Associativity of _*_

Theorem 3 [Associativity of _*_ (assoc*)] (X * Y) * Z = X * (Y * Z)

Proof of Theorem 3 By structural induction on X.

I. Base case

```
open PNAT2 .
  -- fresh constants
  ops y z : -> PNat .
  -- check
  red (0 * y) * z = 0 * (y * z) .
close
```

II. Induction case

```
open PNAT2 .
  -- fresh constants
  ops x y z : -> PNat .
  -- lemmas
  eq (X + Y) * Z = (X * Z) + (Y * Z) . -- (d*o+)
  -- IH
  eq (x * Y) * Z = x * (Y * Z) .
  -- check
  red (s(x) * y) * z = s(x) * (y * z) .
close
```

End of Proof of Theorem 3

Commutativity of _*_

Theorem 4 [Commutativity of _*_ (comm*)] X * Y = Y * X

Proof of Theorem 4 By structural induction on X.

Let x,y be fresh constants of PNat.

I. Base case

What to show is 0 * y = y * 0.

$\underline{0 * y} \rightarrow 0$ by (*1)

We need the following lemma to make progress in the proof:

$\boxed{X * 0 = 0 \quad \text{-- (rz*)}}$

$\underline{y * 0} \rightarrow 0$ by (rz*)

Commutativity of _*_

II. Induction case

What to show is $s(x) * y = y * s(x)$

assuming the induction hypothesis $x * Y = Y * x$ -- (IH)

$\underline{s(x) * y} \rightarrow (\underline{x * y}) + y$ by (*2)

$\rightarrow (y * x) + y$ by (IH)

We need the following lemma to make progress in the proof:

$\boxed{X * s(Y) = (X * Y) + X \quad \text{-- (rs*)}}$

$\underline{y * s(x)} \rightarrow (y * x) + y$ by (rs*)

<u>End of Proof of Theorem 4</u>

Commutativity of _*_

<u>Lemma 4</u> [Right zero of _*_ (rz*)] $X * 0 = 0$

<u>Proof of Lemma 4</u> By structural induction on X.

Let x be a fresh constants of PNat.

I. Base case

What to show is $0 * 0 = 0$.

$\underline{0 * 0} \rightarrow 0$ by (*1)

II. Induction case

What to show is $s(x) * 0 = 0$

assuming the induction hypothesis $x * 0 = 0$ -- (IH)

$\underline{s(x) * 0} \rightarrow (\underline{x * 0}) + 0$ by (*2)

$\rightarrow 0 + 0$ by (IH)

$\rightarrow 0$ by (+1)

<u>End of Proof of Lemma 4</u>

Commutativity of _*_

Lemma 5 [Right successor of _*_ (rs*)] $X * s(Y) = (X * Y) + X$

Proof of Lemma 5 By structural induction on X.

Let x,y be a fresh constants of PNat.

I. Base case

What to show is $0 * s(y) = (0 * y) + 0$.

$\underline{0 * s(y)} \to 0$ by (*1)

$(\underline{0 * y}) + 0 \to \underline{0 + 0}$ by (*1)

$\to 0$ by (+1)

Commutativity of _*_

II. Induction case

What to show is $s(x) * s(y) = (s(x) * y) + s(x)$

assuming the induction hypothesis $x * s(Y) = (x * Y) + x$ -- (IH)

$\underline{s(x) * s(y)} \to (\underline{x * s(y)}) + s(y)$ by (*2)

$\to \underline{((x * y) + x) + s(y)}$ by (IH)

$\to \underline{s(y) + ((x * y) + x)}$ by (comm +)

$\to s(y + ((x * y) + x))$ by (+2)

$(\underline{s(x) * y}) + s(x) \to \underline{((x * y) + y) + s(x)}$ by (*2)

$\to \underline{s(x) + ((x * y) + y)}$ by (comm+)

$\to s(\underline{x + ((x * y) + y)})$ by (+2)

$\to s((\underline{(x * y) + y}) + x)$ by (comm+)

$\to s(\underline{(y + (x * y)) + x})$ by (comm+)

$\to s(y + ((x * y) + x))$ by (assoc+)

End of Proof of Lemma 5

Commutativity of _*_

Lemma 4 [Right zero of _*_ (rz*)] X * 0 = 0

Proof of Lemma 4 By structural induction on X.

I. Base case

```
open PNAT2 .
  -- check
  red 0 * 0 = 0 .
close
```

II. Induction case

```
open PNAT2 .
  -- fresh constants
  op x : -> PNat .
  -- IH
  eq x * 0 = 0 .
  -- check
  red s(x) * 0 = 0 .
close
```

End of Proof of Lemma 4

Commutativity of _*_

Lemma 5 [Right successor of _*_ (rs*)] X * s(Y) = (X * Y) + X

Proof of Lemma 5 By structural induction on X.

I. Base case

```
open PNAT2 .
  -- fresh constants
  op y : -> PNat .
  -- check
  red 0 * s(y) = (0 * y) + 0 .
close
```

II. Induction case

```
open PNAT2 .
  -- fresh constants
  ops x y : -> PNat .
  -- IH
  eq x * s(Y) = (x * Y) + x .
  -- check
  red s(x) * s(y) = (s(x) * y) + s(x) .
close
```

End of Proof of Lemma 5

Commutativity of _*_

Theorem 4 [Commutativity of _*_ (comm*)] X * Y = Y * X

Proof of Theorem 4 By structural induction on X.

I. Base case

```
open PNAT2 .
  -- fresh constants
  op y : -> PNat .
  -- lemmas
  eq X * 0 = 0 . -- (rz*)
  -- check
  red 0 * y = y * 0 .
close
```

II. Induction case

```
open PNAT2 .
  -- fresh constants
  ops x y : -> PNat .
  -- lemmas
  eq X * s(Y) = (X * Y) + X . -- (rs*)
  -- IH
  eq x * Y = Y * x .
  -- check
  red s(x) * y = y * s(x) .
close
```

End of Proof of Theorem 4

Correctness of a Tail Recursive Factorial

Theorem 5 [Correctness of a Tail Recursive Factorial (trf)]
fact1(X) = fact2(X)

Proof of Theorem 5 By structural induction on X.

Let x be fresh constants of PNat.

I. Base case

What to show is fact1(0) = fact2(0).

fact1(0) → s(0) by (f1-1)

fact2(0) → sfact2(0,s(0)) by (f2)
→ s(0) by (sf2-1)

Correctness of a Tail Recursive Factorial

II. Induction case

What to show is fact1(s(x)) = fact2(s(x))

assuming the induction hypothesis fact1(x) = fact2(x) -- (IH)

$\underline{\text{fact1(s(x))}} \rightarrow$ s(x) * $\underline{\text{fact1(x)}}$ by (f1-2)

$\rightarrow$ s(x) * $\underline{\text{fact2(x)}}$ by (IH)

$\rightarrow$ s(x) * sfact2(x,s(0)) by (f2)

$\underline{\text{fact2(s(x))}} \rightarrow$ $\underline{\text{sfact2(s(x),s(0))}}$ by (f2)

$\rightarrow$ sfact2(x,s(x) * s(0)) by (sf2-2)

We cannot make the two terms equal only by rewriting, and need a lemma.

The following is one candidate:

s(X) * sfact2(X,s(0)) = sfact2(X,s(X) * s(0))

This is so specific that the proof of this lemma needs another lemma.

Therefore, we make it more generic as follows:

Y * sfact2(X,Z) = sfact2(X,Y * Z) -- (p-sf2)

$\underline{\text{s(x) * sfact2(x,s(0))}} \rightarrow$ sfact2(x,s(x) * s(0)) by (p-sf2)

<u>End of Proof of Theorem 5</u>

Correctness of a Tail Recursive Factorial

<u>Lemma 6</u> [Property of sfact2 (p-sf2)] Y * sfact2(X,Z) = sfact2(X,Y * Z)

<u>Proof of Lemma6</u> By structural induction on X.

Let x,y,z be fresh constants of PNat.

I. Base case

What to show is y * sfact2(0,z) = sfact2(0,y * z).

y * $\underline{\text{sfact2(0,z)}} \rightarrow$ y * z by (sf2-1)

$\underline{\text{sfact2(0,y * z)}} \rightarrow$ y * z by (sf2-1)

Correctness of a Tail Recursive Factorial

II. Induction case

What to show is y * sfact2(s(x),z) = sfact2(s(x),y * z)

assuming the induction hypothesis

Y * sfact2(x,Z) = sfact2(x,Y * Z) -- (IH)

```
y * sfact2(s(x),z) → y * sfact2(x,s(x) * z)           by (sf2-2)
                   → sfact2(x.y * (s(x) * z))          by (IH)

sfact2(s(x),y * z) → sfact2(x,s(x) * (y * z))          by (sf2-2)
                   → sfact2(x,(s(x) * y) * z)          by (assoc*)
                   → sfact2(x,(y * s(x)) * z)          by (comm*)
                   → sfact2(x,y * (s(x) * z))          by (assoc*)
```

End of Proof of Lemma 6

Correctness of a Tail Recursive Factorial

Lemma 6 [Property of sfact2 (p-sf2)] Y * sfact2(X,Z) = sfact2(X,Y * Z)

Proof of Lemma 6 By structural induction on X.

I. Base case

```
open PNAT3 .
  -- fresh constants
  ops y z : -> PNat .
  -- check
  red y * sfact2(0,z)
      = sfact2(0,y * z) .
close
```

End of Proof of Lemma 6

II. Induction case

```
open PNAT3 .
  -- fresh constants
  ops x y z : -> PNat .
  -- lemmas
  eq (X + Y) * Z = (X * Z) + (Y * Z) . -- (d*o+)
  -- IH
  eq Y * sfact2(x,Z) = sfact2(x,Y * Z) .
  -- check
  red y * sfact2(s(x),z) = sfact2(s(x),y * z) .
close
```

Note that we need to use the lemma that was not used in the manual proof.

6

Note that PNAT3 is PNAT2 in which _*_ is declared as follows:

```
op _*_ : PNat PNat -> PNat {assoc comm} .
```

Correctness of a Tail Recursive Factorial

Theorem 5 [Correctness of a Tail Recursive Factorial (trf)]

fact1(X) = fact2(X)

Proof of Theorem 5 By structural induction on X.

I. Base case

```
open PNAT3 .
  -- check
  red fact1(0) = fact2(0) .
close
```

II. Induction case

```
open PNAT3 .
  -- fresh constants
  op x : -> PNat .
  -- lemmas
  eq Y * sfact2(X,Z) = sfact2(X,Y * Z) . -- (sf2-p)
  -- IH
  eq fact1(x) = fact2(x) .
  -- check
  red fact1(s(x)) = fact2(s(x)) .
close
```

End of Proof of Theorem 5

Exercises

1. Write the specifications and proof scores used in the slides and feed them to the CafeOBJ systems. Write all proofs on the on the slides by hand as well.

2. Write two functions sum1 and sum2 that calculate the sum of natural numbers up to a given natural numbers (inclusive) that correspond to fact1 and fact2, write manual proofs verifying that sum1(X) equals sum2(X) for all natural numbers X, and write proof scores formally verifying that sum1(X) equals sum2(X) for all natural numbers X.

Exercises

3. Do you think that humans understand natural numbers? If yes, describe why. If no, describe why not.

4. What is understanding somethings, such as natural numbers? Describe your opinions on it.

Appendices

The proof score of Theorem 1 is as follows:

```
"Theorem 1 [associativity of _+_ (assoc+)]
(X + Y) + Z = X + (Y + Z)

Proof of Theorem 1. By structural induction on X.
I. Base case"

open PNAT1 .
  -- fresh constants
  ops y z : -> PNat .
  -- check
  red (0 + y) + z = 0 + (y + z) .
close
```

Appendices

```
"II. Induction case"

open PNAT1 .
  -- fresh constants
  ops x y z : -> PNat .
  -- IH
  eq (x + Y) + Z = x + (Y + Z) .
  -- check
  red (s(x) + y) + z = s(x) + (y + z) .
close

"End of Proof of Theorem 1"
```