

# The Curious Inference of Boolos in Mizar and OMEGA\*

Christoph E. Benz Müller<sup>1,2</sup> and Chad E. Brown<sup>2</sup>

<sup>1</sup>The University of Cambridge, Cambridge, UK

<sup>2</sup>Universität des Saarlandes, Saarbrücken, Germany  
chris|cebrown@ags.uni-sb.de

To Andrzej Trybulec

**Abstract.** We examine Boolos' curious inference and formalize it in a system based on set theory (Mizar) and a system based on classical higher-order logic (OMEGA). The Boolos example is interesting because while it can in principle be proven using a complete first-order calculus, it is impractical to do so. In our case study we are interested in aspects such as how natural and at what level of granularity Boolos' short second-order proof sketch can be formalized in Mizar and OMEGA.

## 1 Introduction

In an article in [2, 3], Boolos described a simple theorem of first-order logic which cannot practically be proven by a first-order calculus (even if the calculus includes a cut rule). The idea of the example is as follows:

- Assume we have a constant 1 and a successor function  $s$ .
- Axiomatize the definition of a binary function  $f(n, x)$  which would grow very rapidly on the natural numbers (analogous to the Ackermann function).
- Assume a predicate  $D$  contains 1 and is closed under successor.
- Prove  $f(5, 5)$  satisfies the predicate (where 5 is  $s(s(s(1)))$ ).

First-order calculi must essentially compute the value of  $f(5, 5)$  in order to prove  $D(f(5, 5))$  holds.

Boolos further argues that in a second-order calculus (with comprehension principles), one can prove the theorem very easily. One simply uses the existence of a least set  $N$  containing 1 and closed under successor. Since  $N$  is the least such set, one obtains an induction principle. Using this induction principle, one can prove that for any  $n$  and  $x$  in  $N$ ,  $f(n, x)$  must be in  $\{x \in N \mid D(x)\}$ . In particular, proving  $D(f(5, 5))$  reduces to proving  $5 \in N$  – a trivial task.

---

\* This work has been partly funded by the EPSRC project LEO-II (grant EP/D070511/1) at Cambridge University and the SFB 378 project OMEGA at Saarland University.

Since second-order logic is sufficient for performing the short proof, one can clearly also perform the argument in either higher-order logic or first-order set theory. One can perform the argument so long as one works in a strong enough meta-theory to define the set  $N$  and the sets used while applying the induction principle. To get a concrete idea of how such an argument looks in modern proof assistant systems, we look at Boolos’ curious inference formalized in a system based on set theory (Mizar [1, 11, 7]) and a system based on classical higher-order logic (OMEGA [8, 9]).

Questions that are of interest to us in this case study include

1. How natural can Boolos’ proof script be mapped into proof developments in the two systems? How strong is the influence of the logical basis of these systems to the naturalness of these mappings?
2. How much detailed knowledge about each proof assistant is required? How many different commands are needed?
3. Can the logical peculiarities left implicit in Boolos’ proof sketch be left implicit in the formal proofs as well? What is the de Bruijn factor of the formal proofs?
4. Are structural changes needed in the formal proofs or can Boolos’ proof outline be replayed sequentially step by step? Generally, how far are we away (factoring out the natural language aspects) from a fully automatic verification of Boolos’ proof sketch in modern proof assistants?

## 2 The Curious Inference

We briefly describe the “curious inference” given in detail in [2] and reprinted in Figure 1 with handcrafted proof step annotations.

Boolos constructs a first-order theorem of the form

$$((1) \wedge (2) \wedge (3) \wedge (4) \wedge (5)) \supset (6)$$

where (1)-(6) are given by

- (1)  $\forall n f(n, 1) = s(1)$
- (2)  $\forall x f(1, s(x)) = s(s(f(1, x)))$
- (3)  $\forall n \forall x f(s(n), s(x)) = f(n, f(s(n), x))$
- (4)  $D(1)$
- (5)  $\forall x (D(x) \supset D(s(x)))$
- (6)  $D(f(s(s(s(1))), s(s(s(1))))))$

Intuitively, it is clear that (6) follows from (4) and (5) if one can use (1)-(3) to express  $f(s(s(s(1))), s(s(s(1))))$  as a term formed by applying  $s$  to 1 a finite number of times. However, this “finite number of times” would be an astronomically large number of times. Boolos argues that the number of  $s$ ’s required to represent the simpler term  $f(s(s(s(1))), s(s(s(1))))$  (intuitively,  $f(4, 4)$ ) in terms of  $s$  and 1 is an exponential stack containing 64K 2s.

Boolos gives the following second-order proof of the first-order theorem. By comprehension (Step 1 in Fig. 1), there is a predicate  $N$  such that

$$N(z) \equiv \forall X (X(1) \wedge (\forall y (X(y) \supset X(s(y)))) \supset X(z))$$

<sup>1</sup>By the comprehension principle of second order logic,  $\exists N\forall z(Nz \leftrightarrow \forall X[X1 \& \forall y(Xy \rightarrow Xsy) \rightarrow Xz])$ , <sup>2</sup>and then for some  $N$ ,  $\exists E\forall z(Ez \leftrightarrow Nz \& Dz)$ .  
<sup>3</sup>**Lemma 1:**  $\frac{3.1}{N1}; \frac{3.2}{\forall y(Ny \rightarrow Nsy)}; \frac{3.3}{Nssss1}; \frac{3.4}{E1}; \frac{3.5}{\forall y(Ey \rightarrow E sy)}; \frac{3.6}{Es1}$ ;  
<sup>4</sup>**Lemma 2:**  $\forall n(Nn \rightarrow (\forall x(Nx \rightarrow Efnx)))$   
*Proof:* <sup>4.1</sup>By comprehension,  $\exists M\forall n(Mn \leftrightarrow \forall x(Nx \Rightarrow Efnx))$ . <sup>4.2</sup>We want  $\forall n(Nn \rightarrow Mn)$ . <sup>4.3</sup>Enough to show  $\frac{4.3.1}{M1}$  and  $\frac{4.3.2}{\forall n(Mn \rightarrow Msn)}$ , for then if  $\frac{4.4}{Nn}, \frac{4.5}{Mn}$ .  
<sup>4.3.1</sup> $M1$ : <sup>4.3.1.1</sup>Want  $\forall x(Nx \rightarrow Efx)$ . <sup>4.3.1.2</sup>By comprehension,  $\exists Q\forall x(Qx \leftrightarrow Efx)$ .  
<sup>4.3.1.3</sup>Want  $\forall x(Nx \rightarrow Qx)$ . <sup>4.3.1.4</sup>Enough to show  $\frac{4.3.1.4.1}{Q1}$  and  $\frac{4.3.1.4.2}{\forall x(Qx \rightarrow Qsx)}$ .  
<sup>4.3.1.4.1</sup> $Q1$ : <sup>4.3.1.4.1.1</sup>Want  $Efx$ . <sup>4.3.1.4.1.2</sup>But  $f11 = s1$  by (1) and <sup>4.3.1.4.1.3</sup> $Es1$  by Lemma 1.  
<sup>4.3.1.4.2</sup> $\forall x(Qx \rightarrow Qsx)$ : <sup>4.3.1.4.2.1</sup>Suppose  $Qx$ , <sup>4.3.1.4.2.2</sup>i.e.  $Efx$ . <sup>4.3.1.4.2.3</sup>By (2)  $f1sx = ssf1x$ ; <sup>4.3.1.4.2.4</sup>by Lemma 1 twice,  $Efx$ . <sup>4.3.1.4.2.5</sup>Thus  $Qsx$  and <sup>4.3.1.4.2.6</sup> $M1$ .  
<sup>4.3.2</sup> $\forall n(Mn \rightarrow Msn)$ : <sup>4.3.2.1</sup>Suppose  $Mn$ , <sup>4.3.2.2</sup>i.e.  $\forall x(Nx \rightarrow Efnx)$ .  
<sup>4.3.2.3</sup>Want  $Msn$ , <sup>4.3.2.4</sup>i.e.  $\forall x(Nx \rightarrow Efsnx)$ . <sup>4.3.2.5</sup>By comprehension,  $\exists P\forall x(Px \leftrightarrow Efsnx)$ . <sup>4.3.2.6</sup>Want  $\forall x(Nx \rightarrow Px)$ . <sup>4.3.2.7</sup>Enough to show  $\frac{4.3.2.7.1}{P1}$  and  $\frac{4.3.2.7.2}{\forall x(Px \rightarrow P sx)}$ .  
<sup>4.3.2.7.1</sup> $P1$ : <sup>4.3.2.7.1.1</sup>Want  $Efsn1$ . <sup>4.3.2.7.1.2</sup>But  $fsn1 = s1$  by (1) and <sup>4.3.2.7.1.3</sup> $Es1$  by Lemma 1.  
<sup>4.3.2.7.2</sup> $\forall x(Px \rightarrow P sx)$ : <sup>4.3.2.7.2.1</sup>Suppose  $Px$ , <sup>4.3.2.7.2.2</sup>i.e.  $Efsnx$ ; <sup>4.3.2.7.2.3</sup>thus  $Nfsnx$ . <sup>4.3.2.7.2.4</sup>Want  $Efsnsx$ . <sup>4.3.2.7.2.5</sup>Since  $Nfsnx$  and  $Mn, Efnfsnx$ .  
<sup>4.3.2.7.2.6</sup>But by (3)  $fnfsnx = fsnsx$ ; <sup>4.3.2.7.2.7</sup>thus  $Efsnsx$ .  
<sup>5</sup>By Lemma 1,  $Nssss1$ . <sup>6</sup>By Lemma 2,  $Efssss1ssss1$ . <sup>7</sup>Thus,  $Dfssss1ssss1$ , as desired.

**Fig. 1.** Boolos' original proof sketch from [2, 3]; we have identified single proof steps and annotated them for reference in this paper.

and a predicate  $E$  (Step 2) such that

$$E(z) \equiv N(z) \wedge D(z)$$

Boolos states Lemma 1 (Steps 3.1-6) without proof and proves Lemma 2 (Step 4):

**LEMMA 1:**  $N(1); \forall y(N(y) \supset N(s(y))); N(s(s(s(1))))); E(1); (\forall y(E(y) \supset E(s(y))))); E(s(1))$

**LEMMA 2:**  $\forall n(N(n) \supset \forall x(N(x) \supset E(f(n, x))))$

The proof of LEMMA 2 makes use of the comprehension principle several times (Steps 4.1, 4.3.1.2, and 4.3.2.5) in order to obtain predicates ( $M$ ,  $Q$ , and  $P$ ) for use in the induction principle implicit in the definition of  $N$ . We outline the proof and give the instances of comprehension:

- (Steps 4.1-5) To prove  $\forall x(N(x) \supset E(f(n, x)))$  for any  $n$  satisfying  $N(n)$ , Boolos uses a predicate  $M$  satisfying

$$M(n) \equiv \forall x(N(x) \supset E(f(n, x)))$$

- (Steps 4.3.1.1-4) To prove  $M(1)$ , Boolos uses a predicate  $Q$  satisfying

$$\forall x((Qx) \equiv E(f(1, x)))$$

and easily proves  $Q(1)$  (Steps 4.3.1.4.1-3) and closure of  $Q$  under  $s$  (Steps 4.3.1.4.2.1-6).

- (Steps 4.3.2.1-7) To prove  $M$  is closed under successor, suppose  $n$  satisfies  $N(n)$  and  $M(n)$ . The goal is now to prove  $M(s(n))$ . Equivalently, we should prove  $\forall x(N(x) \supset E(f(s(n), x)))$ . We can prove this by induction on  $x$  using a predicate  $P$  satisfying

$$\forall x((Px) \equiv E(f(s(n), x)))$$

Boolos argues  $P(1)$  (Steps 4.3.2.7.1.1-3) and  $P$  is closed under successor (Steps 4.3.2.7.2.1-7), completing the argument. Then he completes the overall proof by using LEMMA 1 and LEMMA 2 (Steps 5-7).

In summary, Boolos uses comprehension to obtain five predicates:  $N$ ,  $E$ ,  $M$ ,  $Q$ , and  $P$ .

In our set theory version below, we will use a separation principle to obtain sets corresponding to these five predicates. In the higher-order version, we will use  $\lambda$ -abstraction to define corresponding predicate terms (sets) of type  $\iota \rightarrow o$ .

### 3 Mizar Version

We first describe a version of the proof in Mizar [1, 11, 7]. We begin by reserving  $A$  to be a non empty set corresponding to the domain of the first-order problem.

**reserve A for non empty set;**

We reserve  $a$  to be an element of  $A$ , playing the role of 1 in the first order problem. Also,  $n$ ,  $x$  and  $y$  will play the role of first-order variables.

**reserve a,n,x,y for Element of A;**

We reserve  $s$  as a function from  $A$  to  $A$  and  $f$  as a function from  $A \times A$  to  $A$ , playing the roles of  $s$  and  $f$  in the first-order problem.

**reserve s for Function of A,A;**

**reserve f for Function of [: A,A :],A;**

The predicate  $D$  in the first-order problem corresponds to a subset of the set  $A$ .

**reserve D for (Subset of A);**

Second-order predicates will correspond to subsets of  $A$  as well. However, we reserve

$w$ ,  $X$ , and  $z$  to denote generic sets (without assuming they are subsets of  $A$ ).

`reserve w,X,z for set;`

For variables, we will always use one of the alphabetic characters above. Starting from the variable names reserved above, we can form terms and formulas according to the following grammar:

```
term ::= variable | 's.'term | 'f.[' term ',' term ']' | '$1' | '(' term ')'
```

```
formula ::= 'thesis' | term '=' term | term 'in' term | predicate '[' term ']'
```

```
          | formula '&' formula | formula 'implies' formula
```

```
          | 'for' variable-list 'holds' formula | '(' formula ')'
```

A *predicate* will be an identifier corresponding to a `defpred` reasoning item (described below). A *variable-list* is a list of variables separated by commas.

**Warning:** This is a simplification of the Mizar syntax for terms and formulas. Terms and formulas as given above are sufficient for the Mizar encoding of the Boolos example.

We can now declare the main theorem in Mizar as follows:

```
theorem BoolosCuriousInference:
  (for n holds ((f.[n,a] = s.a) &
    (for x holds ((f.[a,s.x] = s.(s.(f.[a,x]))) &
      (for n,x holds f.[s.n,s.x] = f.[n,f.[s.n,x])) &
      (a in D) &
      (for x st (x in D) holds (s.x in D))
    implies
    f.[s.(s.(s.(s.a))),s.(s.(s.(s.a)))] in D
```

In Mizar, such a theorem in an article should be followed by a *Justification*. We want this justification to correspond closely to Boolos' proof sketch. Boolos outlines a proof of this theorem by applying comprehension to obtain  $N$  and  $E$ , stating Lemma 1 (without proof), proving Lemma 2, and finally concluding the main result. In order to understand how to translate this proof sketch to a Mizar justification, we describe a subset of the full Mizar syntax sufficient for this example.

- A *Justification* can either be
  - empty,
  - a simple justification of the form 'by' *References*,
  - a simple justification of the form 'from' *Scheme-Reference*, or
  - a *Proof*.
- A *Proof* is a list of *Reasoning Items* between the keywords 'proof' and 'end'.
- A *Reasoning Item* is of one of the following forms:
  - 'assume' *Conditions* ';'
  - 'let' *Identifier* 'such' *Conditions* ';'
  - 'defpred' *Identifier* '[set]' 'means' *formula* ';'
  - 'consider' *Identifier* 'such' *Conditions* 'from' *Scheme-Reference* ';'
  - ['then' | 'hence' | 'thus'] [*Identifier* ':' ] *formula* *Justification* ';'
- *Conditions* is always of the form
 

```
'that' [Identifier ':' ] formula { 'and' [Identifier ':' ] formula }
```
- Every *Identifier* we use will be a string of alphanumeric characters starting with an alphabetic character.

- *References* is a list of separated by commas. Each member of this list is either an *Identifier* (a local reference to something labeled in the article) or a reference to the a theorem or definition in the Mizar Mathematical Library (MML). The only references to the MML we will use are 'ZFMISC\_1:106', 'XBOOLE\_0:def 1', and 'FUNCT\_2:7'.
- The only *Scheme-Reference* we will use is 'XBOOLE\_0:sch 1' (see below).

**Warning:** The description above can be used to create valid Mizar syntax, but is a simplification of the actual Mizar grammar. For the full Mizar grammar, see [11] or the grammar given on the Mizar web site.

We followed Boolos' outline to write a Mizar version of the proof (see Appendix A.1). The resulting file contains 177 non-comment lines and consisted of 78 reasoning items. This corresponds closely to the 60 steps we identified in Boolos' proof in Figure 1. We now briefly consider some particular aspects of the Mizar version.

The first two uses of comprehension in Boolos' proof are to obtain  $N$  and  $E$ . In Mizar, we can obtain the existence of these sets using a form of separation, referenced as a scheme from the MML article `XBOOLE_0` [5]. In order to use this scheme, we define local predicates  $Np$  and  $Ep$ . Obtaining references to such schemes from the vast MML can be a barrier to writing proofs in Mizar, though some recent work by Urban [10] should help in this respect. In our case, we used `grep` in the MML directory to locate examples of set separation. Once we know such a set exists, we can use `consider` to give it the appropriate name. (Double colons indicate comments.)

```

:: 1. existence of N
  defpred Np[set] means (for X holds
    ((a in X) &
      (for y holds (y in X) implies (s.y in X)))
    implies ($1 in X));
  consider N such that
  Ndef:for z holds z in N iff z in A & Np[z]
  from XBOOLE_0:sch 1;
:: 2. existence of E
  defpred Ep[set] means (($1 in N) & ($1 in D));
  consider E such that
  Edef:for z holds z in E iff z in A & Ep[z]
  from XBOOLE_0:sch 1;

```

Lemma 1 is actually six short lemmas. We split Lemma 1 into six lines in the main proof, each given without justification. Mizar does not accept these inferences. Instead, Mizar reports that the inferences are not accepted and continues processing the file. This is a very useful feature of Mizar since it allows us to complete the outline of the proof and then fill in any missing justifications afterwards.

```

:: 3. Lemma 1
:: 3.1
LEMMA1a: (a in N);
:: 3.2

```

```

LEMMA1b: (for y holds (y in N) implies (s.y in N));
:: 3.3
LEMMA1c: s.(s.(s.(s.a))) in N;
:: 3.4
LEMMA1d: a in E;
:: 3.5
LEMMA1e: (for y holds (y in E) implies (s.y in E));
:: 3.6
LEMMA1f: s.a in E;

```

The main part of the proof is the proof of Lemma 2. We include the proof in the Mizar article at the same level of detail Boolos gives. The only significant complication involves steps such as 4.2 (see Figure 1) in which Boolos states “We want...” or 4.3 in which Boolos states “Enough to show...” In each of these steps, Boolos is implicitly asserting that instead of showing the current goal (the “thesis” in Mizar), we can reduce the current goal to the new, explicitly given, goal or goals. One can simulate such steps in Mizar as follows. Suppose we must show  $G$ , but we want to show  $A$ . The justification that we can reduce  $G$  to  $A$  is a justification of  $A \supset G$ . Using a proof of  $A$  and  $A \supset G$ , we can complete the proof of  $G$ . The following outlines how to perform such a simulation of “enough to show” in Mizar:

```

enoughtoshow: A implies thesis;
A
proof
...
end;
hence thesis by enoughtoshow;

```

The fragment above corresponds to a single “subgoal reduction” step. For a concrete example, consider proof Step 4.2:

```

:: 4.2 Begin We want...
    wewant42: (for n holds (n in N) implies (n in M))
                implies thesis;
    (for n holds (n in N) implies (n in M))
    proof
        :: 4.3 Begin
        ...
        :: 4.3 End
    end;
    hence thesis by wewant42;
:: 4.2 End

```

Mizar could not verify the file with the proof outline since some justifications were not explicit in Boolos’ proof. In particular, there were 25 inferences Mizar did not accept. However, starting with the Mizar article containing the outline, it was not difficult to fill in the remaining inferences to obtain a proof Mizar accepts. Sometimes, this was simply a matter of making justifications explicit. For example, the final part of Lemma 1 could be justified simply by referring to the previous

parts:

```
LEMMA1f: s.a in E by LEMMA1d,LEMMA1e;
```

For most other parts of Lemma 1, we simply gave an explicit proof.

In a few cases, we needed auxiliary lemmas. For example, Boolos often reduces showing every member of  $N$  is a member of another set  $X$  to showing the base case and the induction case. In the proof outline, the first example is Step 4.3:

```
ets43: a in M & (for n holds (n in M) implies (s.n in M))
      implies thesis;
```

where the thesis is that every  $n$  in  $N$  is also in  $M$ . Mizar considers this step to be unjustified. It is not difficult justify such statements using the definition of  $N$ . However, we did not want to add too many steps to the outline in order to complete the proof. Instead, we included (and proved) a lemma `Nindprinc`:

```
theorem Nindprinc:
```

```
(for z holds z in N iff z in A &
 (for X holds ((a in X) &
 (for y holds (y in X) implies (s.y in X))) implies (z in X)))
 &
 (a in X)
 &
 (for y st (y in X) holds (s.y in X))
 implies
 (for n holds (n in N) implies (n in X))
```

Using this lemma, Step 4.3 can be justified as follows:

```
ets43: a in M & (for n holds (n in M) implies (s.n in M))
      implies thesis by Nindprinc,Ndef;
```

Another of the lemmas concludes that terms of the form  $f.[x,y]$  are elements of  $A$ . To prove this particular lemma, three references to the MML were made. These three references plus the set separation scheme are the only references to the MML in the article.

In total, we stated and proved 7 such lemmas.

The Mizar article with the complete proof contains 313 non-comment lines (see Appendix A.2). The main proof contains 104 reasoning items, and the proofs of the lemmas contain an additional 35 reasoning items. The file is not only short, but also quite readable.

## 4 OMEGA Version

In this section we discuss a solution of Boolos' problem in OMEGA [8,9], a proof assistant based on classical higher-order logic (Church's simple type theory [4]).

In order to construct the proof in OMEGA the user essentially only needs to know 7 quite general proof commands, they are:

- `call-otter-on-node` (proof line justification is 'Otter'; see Appendix B.1): calls the first order theorem prover `otter` in order to close a subgoal; these calls are performed whenever Boolos leaves out the logical details in his sketch. In



each call to OTTER the problem is implicitly mapped to first-order logic using an applicational transformation; using OMEGA's TRAMP subsystem [6] these proofs can be, upon request, translated to OMEGA's natural natural deduction calculus, integrated in OMEGA's proof object, and verified.

- **support** (no extra proof line justification needed) this command restricts the available hypotheses for an open proof line; we here use **support** exclusively in connection with **call-otter-on-node** in order to precisely specify before each call the hypotheses OTTER may use in its proof attempt. Both commands could easily be (and in fact should be) combined into one single command.
- **local-def-intro** (proof line justification is 'Local-Def'): introduces a local definition; each time Boolos uses the comprehension principle to introduce a new definition, we use 'local-def-intro' command to introduce the corresponding definition in OMEGA.
- **defn-contract-local-def** and **defn-expand-local-def** (proof line justifications are 'CDef' and 'EDef'): these commands allow to expand (unfold) and contract (fold) definitions; note that the clever use of definitions is an essential aspect in Boolos' proof.
- **impi** (proof line justification is 'IMPI'): the natural deduction rule for implication introduction.
- **foralli** (proof line justification is 'FORALLI'): the natural deduction command for introduction of a universal quantification.
- **ande** (proof line justification is 'ANDE'): the natural deduction rule for conjunction elimination.

Furthermore, the commands **load-problems** and **prove** for loading and initializing the problem are needed. The command **show-pds** (for displaying the current partial proof in OMEGA's emacs interface and the graphical interface LOUI (see Appendix B.4)) is useful but not mandatory.

The problem is initially defined in OMEGA's Post syntax as follows:

```
(th`defproblem boolos-curious-inference
  (in boolos)
  (constants (s (i i)) (f ((i i) i)) (one i) (D (o i)))
  (assumption a1 (forall (lam (n i) (= (f n one) (s one))))))
  (assumption a2
    (forall (lam (x i) (= (f one (s x)) (s (s (f one x)))))))
  (assumption a3
    (forall (lam (n i)
      (forall (lam (x i) (= (f (s n) (s x)) (f n (f (s n) x)))))))
  (assumption a4 (D one))
  (assumption a5 (forall (lam (x i) (implies (D x) (D (s x))))))
  (conclusion conc (D (f (s (s (s (s one)))))) (s (s (s (s one))))))
```

This problem specification is then added to OMEGA's theory library.

The complete interactive session is given in Appendix B.5, the proof script obtained from this interactive session in Appendix B.2, and the final proof object in ASCII, L<sup>A</sup>T<sub>E</sub>X, and GUI representation in Appendices B.3, B.1, and B.4. As for Mizar, we here discuss only some specific fragments of the formal development of Boolos' proof sketch in OMEGA.

After initialization of the OMEGA system with the above proof problem, we proceed step by step along Boolos' proof sketch. In Step 1 Boolos defines property *N* which introduces, as mentioned before, an induction principle wrt. the constructors

1 and  $s$ . Step 2 introduces  $E$  which connects  $N$  and  $D$ . In OMEGA these definition introduction steps (as well as all further definition introduction steps) are handled with `local-def-intro` which takes as single argument the definiens of the new definition. For instance, the set  $N$  is defined by the definiens ( $\lambda$ -term of type  $\iota \leftarrow o$ ):

$$\lambda z_\iota[(\forall X_{\iota \leftarrow o} X(One) \wedge (\forall y_\iota X(y) \Rightarrow X(s(y)))) \Rightarrow X(z)]$$

The name (definiendum) of each local definition is automatically chosen by OMEGA, so that  $N$  becomes  $LD1$  and  $E$  becomes  $LD2$ . Here is the protocol excerpt of the first steps of the proof in OMEGA:

```

OMEGA: prove boolos-curious-inference
  Changing to proof plan BOOLOS-CURIOUS-INFERENC-1
OMEGA: LOCAL-DEF-INTRO (LAM (Z I) (FORALL (LAM (X (O I)) (IMPLIES (AND (X ONE)
  (FORALL (LAM (Y I) (IMPLIES (X Y) (X (S Y)))))) (X Z))))))
OMEGA: LOCAL-DEF-INTRO (LAM (Z I) (AND (LD1 Z) (D Z)))
OMEGA: show-pds
A1 (A1)      ! (FORALL [N:I] (= (F N ONE) (S ONE)))          HYP
A2 (A2)      ! (FORALL [X:I]
              (=
              (F ONE (S X))
              (S (S (F ONE X))))))
A3 (A3)      ! (FORALL [N:I,X:I]
              (=
              (F (S N) (S X))
              (F N (F (S N) X))))          HYP
A4 (A4)      ! (D ONE)                                         HYP
A5 (A5)      ! (FORALL [X:I]
              (IMPLIES (D X) (D (S X))))          HYP
LD1 (LD1)    ! (=DEF                                           LOCAL-DEF
              LD1
              ([Z].
              (FORALL [X:(O I)]
              (IMPLIES
              (AND
              (X ONE)
              (FORALL [Y:I]
              (IMPLIES (X Y) (X (S Y))))))
              (X Z))))))
LD2 (LD2)    ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z))))          LOCAL-DEF
...
CONC (A1 A2 A3 ! (D                                           OPEN
      A4 A5)      (F
                  (S (S (S ONE))))
                  (S (S (S (S ONE))))))

```

In Steps 3.1-6 Boolos introduces the six statements of Lemma 1. As in Mizar we instead introduce six individual lemmas for the final goal line 'conc'. Then we subsequently prove them with the help of OTTER after appropriately choosing the support nodes and unfolding the definitions. We illustrate only the Step 3.1 and omit the others since they are analogous (see also the lines l1-l12 in the final proof object in the Appendix B.1).

```

OMEGA: LEMMA CONC (LD1 ONE)
OMEGA: DEFN-CONTRACT-LOCAL-DEF L1 () LD1 (O)
OMEGA: SUPPORT L2 ()
...
L2 (A1 A2 A3 ! (FORALL [DC-13:(O I)]
      A4 A5)  (IMPLIES
              (AND
              (DC-13 ONE)

```

```

      (FORALL [DC-17:I]
      (IMPLIES
      (DC-13 DC-17)
      (DC-13 (S DC-17))))))
      (DC-13 ONE)))
OMEGA: CALL-OTTER-ON-NODE L2 ...
...
----- PROOF -----
...
OMEGA: show-pds
...
L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)]
      A4 A5) (IMPLIES
      (AND
      (DC-13 ONE)
      (FORALL [DC-17:I]
      (IMPLIES
      (DC-13 DC-17)
      (DC-13 (S DC-17))))))
      (DC-13 ONE)))
      OTTER: (NIL)

L1 (A1 A2 A3 ! (LD1 ONE)
      DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
...

```

Next, we illustrate how the Steps 4.3.2.1-7 are treated; an interesting aspect here is that the definition of  $P$  has to refer to the locally introduced Eigenvariable  $n$  (named  $N1$  in the OMEGA proof). We start with Step 4.3.2.1 is (note that OMEGA has previously chosen the name  $LD3$  for predicate  $M$ ):

```

...
L16 (A1 A2 A3 ! (FORALL [N:I]
      A4 A5) (IMPLIES (LD3 N) (LD3 (S N))))
      OPEN

```

We extract hypothesis and conclusion and unfold the definition of  $LD3$ .

```

OMEGA: FORALLI L16 n1 ()
OMEGA: IMPI L26
OMEGA: DEFN-EXPAND-LOCAL-DEF () L27 LD3 (0)
OMEGA: DEFN-CONTRACT-LOCAL-DEF L28 () LD3 (0)
OMEGA: show-pds
LD3 (LD3) ! (=DEF
      LD3
      ([N].
      (FORALL [X:I]
      (IMPLIES
      (LD1 X)
      (LD2 (F N X))))))
      LOCAL-DEF

L27 (L27) ! (LD3 N1)
      HYP
L29 (L27) ! (FORALL [DC-217:I]
      (IMPLIES
      (LD1 DC-217)
      (LD2 (F N1 DC-217))))
      DEFN-EXPAND-LOCAL-DEF: ((0)) (L27 LD3)

L30 (L27 A1 ! (FORALL [DC-225:I]
      A2 A3 A4 (IMPLIES
      A5) (LD1 DC-225)
      (LD2 (F (S N1) DC-225))))
      OPEN

L28 (L27 A1 ! (LD3 (S N1))
      A2 A3 A4
      A5)
      DEFN-CONTRACT-LOCAL-DEF: ((0)) (L30 LD3)

L26 (A1 A2 A3 ! (IMPLIES (LD3 N1) (LD3 (S N1)))
      A4 A5)
      IMPI: (L28)

L16 (A1 A2 A3 ! (FORALL [N:I]
      A4 A5) (IMPLIES (LD3 N) (LD3 (S N))))
      FORALLI: (N1) (L26)

```

Now the predicate  $P$  (here automatically named  $LD5$ ) is introduced and the problem is then reduced to showing that  $P$  holds for  $One$  and is closed under  $s$ .

```

OMEGA: LOCAL-DEF-INTRO (LAM (X I) (LD2 (F (S N1) X)))
OMEGA: LEMMA L30 (FORALL (LAM (X I) (IMPLIES (LD1 X) (LD5 X))))
OMEGA: DEFN-EXPAND-LOCAL-DEF L30 L31 LD5 (1 0 2 0)
OMEGA: LEMMA L31 (LD5 ONE)
OMEGA: LEMMA L31 (FORALL (LAM (X I) (IMPLIES (LD5 X) (LD5 (S X)))))
OMEGA: DEFN-CONTRACT-LOCAL-DEF L31 () LD1 (1 0 1 0)
OMEGA: show-pds
...
LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z)))) LOCAL-DEF
L30 (L27 A1 ! (FORALL [DC-225:I] DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L31 LD5)
A2 A3 A4 (IMPLIES
A5) (LD1 DC-225)
(LD2 (F (S N1) DC-225))))
L31 (L27 A1 ! (FORALL [X:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L34 LD1)
A2 A3 A4 (IMPLIES (LD1 X) (LD5 X)))
A5)
L32 (L27 A1 ! (LD5 ONE) OPEN
A2 A3 A4
A5)
L33 (L27 A1 ! (FORALL [X:I] OPEN
A2 A3 A4 (IMPLIES (LD5 X) (LD5 (S X))))
A5)
L34 (L27 A1 ! (FORALL [DC-239:I] OPEN
A2 A3 A4 (IMPLIES
A5) (FORALL [DC-250:(0 I)]
(IMPLIES
(AND
(DC-250 ONE)
(FORALL [DC-254:I]
(IMPLIES
(DC-250 DC-254)
(DC-250 (S DC-254))))))
(DC-250 DC-239)))
(LD5 DC-239)))

```

OTTER can be employed to verify the reduction of lines  $L34$  to lines  $L32$  and  $L33$ . Thus, as for lemmas 3.1-6, OTTER can be employed to check the logical details not mentioned explicitly in Boolos proof sketch and to treat these steps implicit in OMEGA as well.

```

OMEGA: SUPPORT L34 (L32 L33)
OMEGA: CALL-OTTER-ON-NODE L34 ...
...
----- PROOF -----
...
OMEGA:

```

In fact, the idea to keep logical details implicit with the help of OTTER allows to further shorten the proof sketch of Boolos. We exemplarily demonstrate this for proof Steps 4.3.1.4.2.1-6 of Boolos original proof sketch. Here we avoid the explicit extraction of hypothesis and conclusion (with  $FORALLI$  and  $IMPI$ ) and only unfold the definition of  $LD4$  (resp.  $Q$ ) before calling OTTER (cf. the proof lines  $L21$ ,  $L24$ , and  $L25$  in Appendix B.1).

In OMEGA there are altogether 83 proof steps needed (see the uncommented lines in OMEGA's proof script in Appendix B.2; it consists of 83 commands to OMEGA) to replay Boolos' proof idea as we sketched above. This number is quite close to the 60 annotations used to identify single proof steps in Boolos' original

proof sketch in Figure 1. Note that each call to OTTER (there are 16 altogether) comes with a previous call of `support` in order to specify which hypotheses OTTER shall use. By simply integrating both commands we would get  $83 - 16 = 67$  steps which comes actually very close to Boolos' original number of steps. The number in OMEGA can possibly further reduced by calling OTTER (or other external systems available to OMEGA) to larger subproblems. But this was not the idea of the exercise, since we aimed at replaying Boolos' proof as close as possible. The detailed proof-object is displayed in Appendix B.3; it has 357 lines and 2949 characters. The quite readable L<sup>A</sup>T<sub>E</sub>X conversion of this proof-object is presented in Appendix B.1. Note, that the justification 'OTTER' can be further expanded in which case the subproofs obtained from OTTER are transformed into natural deduction proof objects in OMEGA to be verified. We have not done this here. By doing so the proof object will clearly grow.

## 5 Evaluation

Boolos' famous curious inference has been formalized in two modern proof assistants: Mizar and OMEGA. We compare and discuss these formalizations according to the following aspects:

1. **How natural can Boolos' proof script be mapped into proof developments in the systems? How strong is the influence of logical basis of these systems to the 'naturalness' of these mappings?**

Boolos' original proof outline with its 60 steps can be very closely replayed in both Mizar and OMEGA. The different logical bases of the systems does not appear to play a significant role.

In Mizar, one can use the proof outline to create a Mizar article with 78 proof steps (reasoning items). Though this article cannot be verified by Mizar, the file can be completed by adding lemmas, justifications and further proof steps in order to obtain a verified Mizar article. The completed article contains 139 proof steps (reasoning items). The Mizar proof is quite readable.

In OMEGA, there are 83 proof steps needed (and this number could easily be further reduced to 67 by a natural merge of the `call-otter-on-node` and `support` commands). The formalization in POST is admittedly not very readable. In the GUI representation (see Appendix B.4) the readability is significantly better. OMEGA experts, however, seldom use OMEGA's LOUI interface in practice. The machine-oriented proof script of OMEGA is not readable/informative to human users at all. By replaying it step by step in OMEGA and by investigating the development of the partial proof in OMEGA's emacs or LOUI interface, however, a good understanding of the proof script can be obtained.

2. **How much detailed knowledge about the proof assistant is needed? How many different commands are needed?**

Anyone can write a proof in Mizar after perusing [11] or [7]. There are a few essential ingredients:

- One needs to know the concrete syntax for quantifiers and logical connectives.
- One needs to know the syntax for different kinds of reasoning items.
- One needs to know how to give simple justifications.

There are other issues as well (e.g., vocabularies and definitions) which one can learn as needed. Sometimes giving a justification can involve finding an appropriate definition, theorem or scheme in the MML (Mizar Mathematical Library). Finding such references can be challenging for a novice.

In order to construct the proof in OMEGA the user essentially only needs to have a basic understanding of the POST syntax and to know the 7(+2) rather general proof commands as mentioned before.

3. **Can the logical peculiarities left implicit in Boolos' argument be left implicit in these proof developments as well? What is the de Bruijn factor of the formal proofs?**

One can leave most of the logical details implicit in Mizar, but one sacrifices full verification. In order to obtain full verification, some of these logical details must be made explicit (though many logical rules are built into the Mizar verifier). This is, in fact, a natural way to write a Mizar article: one first writes an article in which some justifications are not accepted, then one fills in the logical details.

The steps that are left implicit by Boolos can be treated implicit in OMEGA as well. This is done by calling the first order theorem prover OTTER<sup>1</sup>. In order to automatically verify the proofs delivered by OTTER within OMEGA the TRAMP system can be employed. Generally we can say that Boolos' proof sketch is detailed enough such that all remaining logical peculiarities can already be automatically dealt with in OMEGA.

To obtain the de Bruijn factor for both formalizations, we consider the (relative) sizes of the following compressed (using gzip) files:

- A L<sup>A</sup>T<sub>E</sub>X file containing Figure 1 (Boolos' proof sketch) without annotations: 637 bytes.
- The completed Mizar article (without comments): 2310 bytes; de Bruijn factor 3.6.
- There are two files we may consider in OMEGA: (A) the proof script (see Appendix B.2) that can be employed to automatically reconstruct the final proof object in OMEGA, and (B) the ASCII representation of the final proof object itself (see Appendix B.3). For (A) we have 838 bytes and de Bruijn factor 1.3 and for (B) we obtain 2602 bytes and de Bruijn factor 4, 1. (Note, that by merging the commands `call-otter-on-node` and `support` commands in OMEGA we would reach de Bruijn factor close to 1.1 for file (A) – a very impressive figure.)

4. **Are structural changes needed in the formal proofs or can Boolos' proof outline be replayed sequentially step by step? Generally, how far are we away, from a fully automatic verification of Boolos' rather detailed proof script in modern proof assistants?**

---

<sup>1</sup> There is no particular reason why we have chosen OTTER; other first-order ATPs should be capable of proving all subgoals as well.

The only significant structural changes necessary to code the proof into Mizar resulted from Boolos' use of subgoals in the proof. One can imagine a program translating the L<sup>A</sup>T<sub>E</sub>X version of Boolos' proof sketch into the first Mizar article and then another program (using an automated theorem prover) filling in the remaining gaps to form the second Mizar article.

In OMEGA none (or only very minor) structural modifications of Boolos' original proof sketch are needed.<sup>2</sup> The proof can in fact be replayed very naturally step by step with a rather small amount of expertise about OMEGA. The main challenge for full automatic verification seems to be the bridge between the mixed use of natural language and mathematical formulas in Boolos' proof sketch and the respective proof commands in OMEGA.

## 6 Conclusion

We have shown that the (second-order) proof sketch of Boolos for his curious inference can be very naturally formalized and verified in Mizar and OMEGA. The fact that Mizar is based on first-order set theory and OMEGA on classical higher-order logic is of minor influence to the the naturalness of proof development. We have shown in particular that Boolos proof sketch can be replayed (in particular in OMEGA) at the same level of granularity (many details that Boolos leaves out can be omitted in the formal developments as well) and step by step, i.e., without need for major structural proof reorganisation. Assuming sufficient progress in the area of semantic analysis of natural language (in fact, mixed mathematical and natural language), this gives hope that we will one day be able to fully automatically analyze and verify human proof sketches such as the one by Boolos.

Furthermore, as we have already indicated in the paper, it should be possible to further raise the level of granularity. The stronger the automated subsystems (such as the ATPs in OMEGA) will get, the fewer details have to be provided to verify the proof. The full automation of Boolos curious inference seems not to be in reach and it will be a challenge problem to automated theorem proving for a long time to come. The key steps, namely the invention of the predicates  $N - P$ , the invention of the appropriate lemmata, and the clever unfolding of definitions are not yet sufficiently supported in todays theorem provers.

## References

1. Grzegorz Bancerek and Piotr Rudnicki. A Compendium of Continuous Lattices in MIZAR. *J. Autom. Reason.*, 29(3-4):189–224, 2002.
2. George Boolos. A curious inference. *Journal of Philosophical Logic*, 16:1–12, 1987.
3. George Boolos. *Logic, logic, and logic*. Harvard University Press, 1998.

---

<sup>2</sup> Note that we have slightly modified (shortened) Boolos' argument in Steps 4.3.1.4.2.1 – 6 for demonstration purposes only. As Steps 4.3.2.1 – 7 show, we could well have followed Boolos' original proof sketch more closely by explicitly extracting hypothesis and conclusion.

4. Alonzo Church. A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, 5:56–68, 1940.
5. Library Committee. Boolean properties of sets — definitions. *Journal of Formalized Mathematics*, EMM, 2002. [http://mizar.org/JFM/EMM/xboole\\_0.html](http://mizar.org/JFM/EMM/xboole_0.html).
6. Andreas Meier. System description: TRAMP: Transformation of machine-found proofs into natural deduction proofs at the assertion level. In David McAllester, editor, *Automated Deduction – CADE-17*, volume 1831 of *Lecture Notes in Artificial Intelligence*, pages 460–464, Pittsburgh, PA, USA, 2000. Springer-Verlag.
7. Michal Muzalewski. *An Outline of PC Mizar*. In R.Matuszewski, editor, Series of the Fondation Ph. le Hodey. Brussels, 1993.
8. Jörg Siekmann, Christoph Benz Müller, and Serge Autexier. Computer supported mathematics with omega. *Journal of Applied Logic*, 4(4):533–559, 2006.
9. Jörg Siekmann, Christoph Benz Müller, Armin Fiedler, Andreas Meier, Immanuel Normann, and Martin Pollet. *Proof Development in OMEGA: The Irrationality of Square Root of 2*, pages 271–314. Kluwer Applied Logic series (28). Kluwer Academic Publishers, 2003. ISBN 1-4020-1656-5.
10. Josef Urban. MoMM - fast interreduction and retrieval in large libraries of formalized mathematics. *International Journal on Artificial Intelligence Tools*, 2005. forthcoming, available online at <http://ktiml.mff.cuni.cz/~urban/MoMM/momm.ps>.
11. F. Wiedijk. Mizar: An impression, 1999.



## A Mizar Version

By Chad E. Brown.

In the following subsections we give the file corresponding to the outline of the proof (with 25 unaccepted inferences) and the file in which all inferences are accepted. Both of these Mizar articles begin by declaring the following environment.

```

environ

vocabularies FUNCT_1, FINSEQ_1, RELAT_1, FINSET_1, CARD_1,
  BOOLE, PARTFUN1, ARYTM_1, INT_1, RLSUB_2, TARSKI, FINSEQ_4,
  ORDINAL2, ARYTM, ZFMISC_1;
notations TARSKI, XBOOLE_0, ENUMSET1, SUBSET_1, ORDINAL1,
  ORDINAL2, RELAT_1, NUMBERS, XCMPLX_0, FINSEQ_1, FUNCT_1,
  PARTFUN1, CARD_1, FUNCT_2, FINSET_1, INT_1, NAT_1, FINSEQ_3,
  XXREAL_0, ZFMISC_1, RELSET_1;
constructors TARSKI, XBOOLE_0, ENUMSET1, WELLORD2, FUNCT_2,
  XXREAL_0, NAT_1, INT_1, MEMBERED, FINSEQ_3, ORDINAL2,
  ZFMISC_1;
registrations XBOOLE_0, FINSEQ_1, FUNCT_1, INT_1, FINSET_1,
  RELSET_1, MEMBERED, ARYTM_3;
requirements REAL, NUMERALS, SUBSET, BOOLE, ARITHM;
theorems XBOOLE_0, TARSKI, FUNCT_1, FUNCT_2, ZFMISC_1;
schemes XBOOLE_0;

```

### A.1 Proof Outline in Mizar

Below we give the Mizar article which closely corresponds to the outline given by Boolos. Mizar does not accept 25 of the inferences in this file.

```

begin

reserve A for non empty set;
reserve a,n,x,y for Element of A;
reserve s for Function of A,A;
reserve f for Function of [: A,A :],A;
reserve D for (Subset of A);
reserve X,z,N,E,M,P,Q for set;

theorem BoolosCuriousInference:
  (for n holds ((f.[n,a]) = s.a)) &
  (for x holds ((f.[a,s.x]) = s.(s.(f.[a,x]))) &
  (for n,x holds f.[s.n,s.x] = f.[n,f.[s.n,x]]) &
  (a in D) &
  (for x st (x in D) holds (s.x in D))
  implies
  f.[s.(s.(s.(s.a))),s.(s.(s.(s.a)))] in D
  proof

```

```

assume that
H1: (for n holds ((f.[n,a] = s.a) and
H2: (for x holds ((f.[a,s.x] = s.(s.(f.[a,x]))) and
H3: (for n,x holds f.[s.n,s.x] = f.[n,f.[s.n,x]]) and
H4: (a in D) and
H5: (for x st (x in D) holds (s.x in D));
:: 1. existence of N
defpred Np[set] means (for X holds
((a in X) &
(for y holds (y in X) implies (s.y in X)))
implies ($1 in X));
consider N such that
Ndef:for z holds z in N iff z in A & Np[z]
from XBOOLE_0:sch 1;
:: 2. existence of E
defpred Ep[set] means (($1 in N) & ($1 in D));
consider E such that
Edef:for z holds z in E iff z in A & Ep[z]
from XBOOLE_0:sch 1;
:: 3. Lemma 1
:: 3.1
LEMMA1a: (a in N);
:: 3.2
LEMMA1b: (for y holds (y in N) implies (s.y in N));
:: 3.3
LEMMA1c: s.(s.(s.(s.a))) in N;
:: 3.4
LEMMA1d: a in E;
:: 3.5
LEMMA1e: (for y holds (y in E) implies (s.y in E));
:: 3.6
LEMMA1f: s.a in E;
:: 4
LEMMA2: (for n holds (n in N) implies
(for x holds (x in N) implies (f.[n,x] in E)))
proof
:: 4.1 (existence of M)
defpred Mp[set] means (for x holds (x in N)
implies f.[$1,x] in E);
consider M such that
Mdef:(for z holds (z in M) iff z in A & Mp[z])
from XBOOLE_0:sch 1;
:: 4.2 Begin We want...
wewant42: (for n holds (n in N) implies (n in M))
implies thesis;
(for n holds (n in N) implies (n in M))

```

```

proof
  :: 4.3 Begin Enough to show...
  ets43: a in M & (for n holds (n in M) implies (s.n in M))
                                     implies thesis;

  :: 4.3.1 Begin
  aM: (a in M)
proof
  :: 4.3.1.1 Begin Want...
  want4311:(for x holds (x in N) implies f.[a,x] in E)
                                     implies thesis;
  (for x holds (x in N) implies f.[a,x] in E)
proof
  :: 4.3.1.2 Q exists
  defpred Qp[set] means f.[a,$1] in E;
  consider Q such that
  Qdef:(for z holds (z in Q) iff z in A & Qp[z])
  from XBOOLE_0:sch 1;
  :: 4.3.1.2 End
  :: 4.3.1.3 Begin Want...
  want4313: (for x holds (x in N) implies (x in Q))
                                     implies thesis;
  (for x holds (x in N) implies (x in Q))
proof
  :: 4.3.1.4 Begin Enough to show...
  ets4314: (a in Q) &
           (for x holds (x in Q) implies (s.x in Q))
                                     implies thesis;

  :: 4.3.1.4.1 Begin
  aQ: (a in Q)
proof
  :: 4.3.1.4.1.1 Begin Want
  want431411:(f.[a,a] in E) implies thesis;
  f.[a,a] in E
proof
  :: 4.3.1.4.1.2
  fact431412: f.[a,a] = s.a by H1;
  :: 4.3.1.4.1.3
  s.a in E by LEMMA1f;
  hence thesis by fact431412;
end;
  hence thesis by want431411;
  :: 4.3.1.4.1.1 End
end;
  :: 4.3.1.4.1 End
  :: 4.3.1.4.2 Begin
  sQ: (for x holds (x in Q) implies (s.x in Q))

```

```

proof
  :: 4.3.1.4.2.1
    let x such that xQ: x in Q;
  :: 4.3.1.4.2.2
    fact431422: f.[a,x] in E;
  :: 4.3.1.4.2.3
    fact431423: f.[a,s.x] = s.(s.(f.[a,x])) by H2;
  :: 4.3.1.4.2.4 Begin "By Lemma 1 twice"
    f.[a,s.x] in E;
  :: 4.3.1.4.2.4 End
  :: 4.3.1.4.2.5
    hence s.x in Q;
  end;
  :: 4.3.1.4.2 End
  thus thesis by ets4314,aQ,sQ;
  :: 4.3.1.4 End
end;
hence thesis by want4313;
:: 4.3.1.3 End
end;
hence thesis by want4311; :: 4.3.1.4.2.6
:: 4.3.1.1 End
end;
:: 4.3.1 End
:: 4.3.2 Begin
sM: (for n holds (n in M) implies (s.n in M))
proof
  :: 4.3.2.1
    let n such that nM: (n in M);
  :: 4.3.2.2
    (for x holds (x in N) implies f.[n,x] in E);
  :: 4.3.2.3 Begin Want
  want4323: (s.n in M) implies thesis;
  (s.n in M)
  proof
  :: 4.3.2.4 Begin IE
    ie4324:(for x holds (x in N) implies f.[s.n,x] in E)
            implies thesis;
    (for x holds (x in N) implies f.[s.n,x] in E)
    proof
  :: 4.3.2.5 P exists
    defpred Pp[set] means f.[s.n,$1] in E;
    consider P such that
    Pdef:(for z holds (z in P) iff z in A & Pp[z])
    from XBOOLE_0:sch 1;
  :: 4.3.2.5 End
  end
  end
  end
  end
  end

```

```

:: 4.3.2.6 Begin Want
  want4326: (for x holds (x in N) implies (x in P))
            implies thesis;
  (for x holds (x in N) implies (x in P))
proof
:: 4.3.2.7 Begin Enough to Show...
  ets4327: (a in P) &
            (for x holds (x in P) implies (s.x in P))
            implies thesis;

:: 4.3.2.7.1
  aP: (a in P)
proof
:: 4.3.2.7.1.1 Begin Want
  want432711: f.[s.n,a] in E implies thesis;
  f.[s.n,a] in E
proof
:: 4.3.2.7.1.2
  eh1: f.[s.n,a] = s.a by H1;
:: 4.3.2.7.1.3
  s.a in E by LEMMA1f;
  hence thesis by eh1;
end;
  hence thesis by want432711;
:: 4.3.2.7.1.1 End
end;
:: 4.3.2.7.2
  sP: (for x holds (x in P) implies (s.x in P))
proof
:: 4.3.2.7.2.1
  let x such that xP: x in P;
:: 4.3.2.7.2.2
  f.[s.n,x] in E;
:: 4.3.2.7.2.3
  then fact432723: f.[s.n,x] in N;
:: 4.3.2.7.2.4 Begin Want...
  want432724: f.[s.n,s.x] in E implies thesis;
  f.[s.n,s.x] in E
proof
:: 4.3.2.7.2.5
  fact432725: f.[n,f.[s.n,x]] in E
  by fact432723,nM;
:: 4.3.2.7.2.6
  fact432726: f.[n,f.[s.n,x]] = f.[s.n,s.x]
  by H3;
:: 4.3.2.7.2.7
  thus f.[s.n,s.x] in E;

```

```
        end;
        hence thesis by want432724;
    :: 4.3.2.7.2.4 End
    end;
    thus thesis by ets4327,aP,sP;
    :: 4.3.2.7 End
    end;
    hence thesis by want4326;
    :: 4.3.2.6 End
    end;
    hence thesis by ie4324;
    :: 4.3.2.4 End
    end;
    hence thesis by want4323;
    :: 4.3.2.3 End
    end;
    :: 4.3.2 End
    hence thesis by ets43,aM,sM;
    :: 4.3 End
    end;
    hence thesis by wewant42;
    :: 4.2 End
    end;
    :: 5
    s.(s.(s.(s.a))) in N by LEMMA1c;
    :: 6
    then f.[s.(s.(s.(s.a))),s.(s.(s.(s.a)))] in E
    by LEMMA2,H1,H2,H3,H4,H5;
    :: 7
    hence thesis by Edef;
    end;
```

## A.2 Complete Proof in Mizar

The following is the body of the file which Mizar accepts. The file is the proof outline with additional lemmas and justifications.

```
begin

reserve A for non empty set;
reserve a,n,x,y for Element of A;
reserve s for Function of A,A;
reserve f for Function of [: A,A :],A;
reserve D for (Subset of A);
reserve X,z,N,E,M,P,Q for set;

:: Typing lemma for s
theorem stype:
```

```

for x holds s.x is Element of A;

:: Typing lemma for f
theorem ftype:
  for x,y holds f.[x,y] is Element of A
  proof
    let x,y;
    xA: x in A;
    yA: y in A;
    A1: [x,y] in [:A,A:] by ZFMISC_1:106,xA,yA;
    then A2: [:A,A:] <> {} by XBOOLE_0:def 1;
    f.[x,y] in A by FUNCT_2:7,A1,A2;
    hence thesis;
  end;

theorem inference42:
  (for z holds (z in M) iff z in A & (for x holds (x in N)
    implies f.[z,x] in E))
  &
  (for n holds (n in N) implies (n in M))
  implies
  (for n holds (n in N) implies
  (for x holds (x in N) implies (f.[n,x] in E)))
  proof
    assume Mdef: (for z holds (z in M) iff
      z in A & (for x holds (x in N) implies f.[z,x] in E));
    assume NM: (for n holds (n in N) implies (n in M));
    let n such that nN: (n in N);
    let x such that xN: (x in N);
    (n in M) by nN,NM;
    hence thesis by Mdef,xN;
  end;

theorem inference4313:
  (for z holds (z in Q) iff z in A & f.[a,z] in E)
  &
  (for x holds (x in N) implies (x in Q))
  implies
  (for x holds (x in N) implies f.[a,x] in E)
  proof
    assume Qdef:(for z holds (z in Q) iff z in A & f.[a,z] in E);
    assume NQ: (for x holds (x in N) implies (x in Q));
    let x such that xN: x in N;
    x in Q by xN,NQ;
    hence thesis by Qdef;
  end;

```

```
theorem inference4326:
  (for z holds (z in P) iff z in A & f.[s.n,z] in E)
  &
  (for x holds (x in N) implies (x in P))
  implies
  (for x holds (x in N) implies f.[s.n,x] in E)
proof
  assume Pdef: (for z holds (z in P) iff
                z in A & f.[s.n,z] in E);
  assume NP: (for x holds (x in N) implies (x in P));
  let x such that xN: x in N;
  x in P by xN,NP;
  hence thesis by Pdef;
end;

theorem inference432725:
  f.[s.n,x] in N
  &
  n in M
  &
  (for z holds (z in M) iff z in A & (for x holds (x in N)
                                        implies f.[z,x] in E))
  implies
  f.[n,f.[s.n,x]] in E
proof
  assume fsnxN: f.[s.n,x] in N;
  assume nM: n in M;
  assume Mdef: (for z holds (z in M) iff
                  z in A & (for x holds (x in N) implies f.[z,x] in E));
  ftA: f.[s.n,x] is Element of A by ftype;
  (for x holds (x in N) implies f.[n,x] in E) by Mdef,nM;
  hence thesis by fsnxN,ftA;
end;

theorem Nindprinc:
  (for z holds z in N iff z in A &
   (for X holds ((a in X) &
                 (for y holds (y in X) implies (s.y in X)))
                 implies (z in X)))
  &
  (a in X)
  &
  (for y st (y in X) holds (s.y in X))
  implies
  (for n holds (n in N) implies (n in X))
proof
```



```

assume Ndef: (for z holds z in N iff z in A &
(for X holds ((a in X) &
(for y holds (y in X) implies (s.y in X)))
implies (z in X)))));
assume aX: a in X;
assume sX: (for y st (y in X) holds (s.y in X));
let n such that nN: n in N;
(for X holds
((a in X) &
(for y holds (y in X) implies (s.y in X))) implies (n in X))
by nN,Ndef;
hence thesis by aX,sX;
end;

theorem BoolosCuriousInference:
(for n holds ((f.[n,a] = s.a) &
(for x holds ((f.[a,s.x] = s.(s.(f.[a,x]))) &
(for n,x holds f.[s.n,s.x] = f.[n,f.[s.n,x]]) &
(a in D) &
(for x st (x in D) holds (s.x in D))
implies
f.[s.(s.(s.(s.a))),s.(s.(s.(s.a)))] in D
proof
  assume that
  H1: (for n holds ((f.[n,a] = s.a) and
  H2: (for x holds ((f.[a,s.x] = s.(s.(f.[a,x]))) and
  H3: (for n,x holds f.[s.n,s.x] = f.[n,f.[s.n,x]]) and
  H4: (a in D) and
  H5: (for x st (x in D) holds (s.x in D));
:: 1. existence of N
defpred Np[set] means (for X holds
((a in X) &
(for y holds (y in X) implies (s.y in X)))
implies ($1 in X));
consider N such that
Ndef:for z holds z in N iff z in A & Np[z]
from XBOOLE_0:sch 1;
:: 2. existence of E
defpred Ep[set] means (($1 in N) & ($1 in D));
consider E such that
Edef:for z holds z in E iff z in A & Ep[z]
from XBOOLE_0:sch 1;
:: 3. Lemma 1
:: 3.1
LEMMA1a: (a in N)
proof

```

```

    Np[a];
    hence thesis by Ndef;
end;
:: 3.2
LEMMA1b: (for y holds (y in N) implies (s.y in N))
proof
  let y such that yN: (y in N);
  Npy: Np[y] by Ndef,yN;
  Np[s.y]
  proof
    let X such that aX: a in X and
    sX: (for y st (y in X) holds (s.y in X));
    y in X by aX,sX,Npy;
    hence thesis by sX;
  end;
  hence thesis by Ndef;
end;
:: 3.3
LEMMA1c: s.(s.(s.(s.a))) in N
proof
  a in N by LEMMA1a;
  then s.a in N by LEMMA1b;
  then s.(s.a) in N by LEMMA1b;
  then s.(s.(s.a)) in N by LEMMA1b;
  hence thesis by LEMMA1b;
end;
:: 3.4
LEMMA1d: a in E by LEMMA1a,H4,Edef;
:: 3.5
LEMMA1e: (for y holds (y in E) implies (s.y in E))
proof
  let y such that yE: y in E;
  y in N by Edef,yE;
  then syN: s.y in N by LEMMA1b;
  y in D by Edef,yE;
  then s.y in D by H5;
  hence thesis by syN,Edef;
end;
:: 3.6
LEMMA1f: s.a in E by LEMMA1d,LEMMA1e;
:: 4
LEMMA2: (for n holds (n in N) implies
  (for x holds (x in N) implies (f.[n,x] in E)))
proof
  :: 4.1 (existence of M)
  defpred Mp[set] means (for x holds (x in N)

```

```

implies f.[ $\$1,x$ ] in E);
consider M such that
Mdef:(for z holds (z in M) iff z in A & Mp[z])
from XBOOLE_0:sch 1;
:: 4.2 Begin We want...
wewant42: (for n holds (n in N) implies (n in M))
           implies thesis by inference42,Mdef;
(for n holds (n in N) implies (n in M))
proof
  :: 4.3 Begin Enough to show...
  ets43: a in M & (for n holds (n in M) implies (s.n in M))
         implies thesis by Nindprinc,Ndef;
  :: 4.3.1 Begin
  aM: (a in M)
proof
  :: 4.3.1.1 Begin Want...
  want4311:(for x holds (x in N) implies f.[a,x] in E)
           implies thesis by Mdef;
  (for x holds (x in N) implies f.[a,x] in E)
proof
  :: 4.3.1.2 Q exists
  defpred Qp[set] means f.[a, $\$1$ ] in E;
  consider Q such that
  Qdef:(for z holds (z in Q) iff z in A & Qp[z])
  from XBOOLE_0:sch 1;
  :: 4.3.1.2 End
  :: 4.3.1.3 Begin Want...
  want4313: (for x holds (x in N) implies (x in Q))
            implies thesis by inference4313,Qdef;
  (for x holds (x in N) implies (x in Q))
proof
  :: 4.3.1.4 Begin Enough to show...
  ets4314: (a in Q) &
           (for x holds (x in Q) implies (s.x in Q))
           implies thesis by Nindprinc,Ndef;
  :: 4.3.1.4.1 Begin
  aQ: (a in Q)
proof
  :: 4.3.1.4.1.1 Begin Want
  want431411:(f.[a,a] in E) implies thesis by Qdef;
  f.[a,a] in E
proof
  :: 4.3.1.4.1.2
  fact431412: f.[a,a] = s.a by H1;
  :: 4.3.1.4.1.3
  s.a in E by LEMMA1f;

```

```

        hence thesis by fact431412;
    end;
    hence thesis by want431411;
    :: 4.3.1.4.1.1 End
end;
:: 4.3.1.4.1 End
:: 4.3.1.4.2 Begin
sQ: (for x holds (x in Q) implies (s.x in Q))
proof
:: 4.3.1.4.2.1
    let x such that xQ: x in Q;
:: 4.3.1.4.2.2
    fact431422: f.[a,x] in E by xQ,Qdef;
:: 4.3.1.4.2.3
    fact431423: f.[a,s.x] = s.(s.(f.[a,x])) by H2;
:: 4.3.1.4.2.4 Begin "By Lemma 1 twice"
    faxA: f.[a,x] is Element of A by ftype;
    then sfaxA: s.(f.[a,x]) is Element of A by stype;
    s.(f.[a,x]) in E by fact431422,LEMMA1e,faxA;
    then s.(s.(f.[a,x])) in E by LEMMA1e,sfaxA;
    then f.[a,s.x] in E by fact431423;
:: 4.3.1.4.2.4 End
:: 4.3.1.4.2.5
    hence s.x in Q by Qdef;
end;
:: 4.3.1.4.2 End
thus thesis by ets4314,aQ,sQ;
:: 4.3.1.4 End
end;
hence thesis by want4313;
:: 4.3.1.3 End
end;
hence thesis by want4311; :: 4.3.1.4.2.6
:: 4.3.1.1 End
end;
:: 4.3.1 End
:: 4.3.2 Begin
sM: (for n holds (n in M) implies (s.n in M))
proof
:: 4.3.2.1
    let n such that nM: (n in M);
:: 4.3.2.2
    (for x holds (x in N) implies f.[n,x] in E) by nM,Mdef;
:: 4.3.2.3 Begin Want
    want4323: (s.n in M) implies thesis;
    (s.n in M)

```

```

proof
:: 4.3.2.4 Begin IE
  ie4324:(for x holds (x in N) implies f.[s.n,x] in E)
        implies thesis by Mdef;
  (for x holds (x in N) implies f.[s.n,x] in E)
proof
:: 4.3.2.5 P exists
  defpred Pp[set] means f.[s.n,$1] in E;
  consider P such that
  Pdef:(for z holds (z in P) iff z in A & Pp[z])
  from XBOOLE_0:sch 1;
:: 4.3.2.5 End
:: 4.3.2.6 Begin Want
  want4326: (for x holds (x in N) implies (x in P))
            implies thesis by inference4326,Pdef;
  (for x holds (x in N) implies (x in P))
proof
:: 4.3.2.7 Begin Enough to Show...
  ets4327: (a in P) &
            (for x holds (x in P) implies (s.x in P))
            implies thesis by Nindprinc,Ndef;
:: 4.3.2.7.1
  aP: (a in P)
proof
:: 4.3.2.7.1.1 Begin Want
  want432711: f.[s.n,a] in E implies thesis
  by Pdef;
  f.[s.n,a] in E
proof
:: 4.3.2.7.1.2
  eh1: f.[s.n,a] = s.a by H1;
:: 4.3.2.7.1.3
  s.a in E by LEMMA1f;
  hence thesis by eh1;
end;
  hence thesis by want432711;
:: 4.3.2.7.1.1 End
end;
:: 4.3.2.7.2
  sP: (for x holds (x in P) implies (s.x in P))
proof
:: 4.3.2.7.2.1
  let x such that xP: x in P;
:: 4.3.2.7.2.2
  f.[s.n,x] in E by xP,Pdef;
:: 4.3.2.7.2.3

```

```

        then fact432723: f.[s.n,x] in N by Edef;
    :: 4.3.2.7.2.4 Begin Want...
        want432724: f.[s.n,s.x] in E implies thesis
        by Pdef;
        f.[s.n,s.x] in E
        proof
    :: 4.3.2.7.2.5
        fact432725: f.[n,f.[s.n,x]] in E
        by fact432723,nM,inference432725,Mdef;
    :: 4.3.2.7.2.6
        fact432726: f.[n,f.[s.n,x]] = f.[s.n,s.x]
        by H3;
    :: 4.3.2.7.2.7
        thus f.[s.n,s.x] in E by fact432725,fact432726;
        end;
        hence thesis by want432724;
    :: 4.3.2.7.2.4 End
        end;
        thus thesis by ets4327,aP,sP;
    :: 4.3.2.7 End
        end;
        hence thesis by want4326;
    :: 4.3.2.6 End
        end;
        hence thesis by ie4324;
    :: 4.3.2.4 End
        end;
        hence thesis by want4323;
    :: 4.3.2.3 End
        end;
    :: 4.3.2 End
        hence thesis by ets43,aM,sM;
    :: 4.3 End
        end;
        hence thesis by wewant42;
    :: 4.2 End
        end;
    :: 5
        s.(s.(s.(s.a))) in N by LEMMA1c;
    :: 6
        then f.[s.(s.(s.(s.a))),s.(s.(s.(s.a)))] in E
        by LEMMA2,H1,H2,H3,H4,H5;
    :: 7
        hence thesis by Edef;
    end;

```

## B OMEGA Version

By Christoph Benzmüller.

### B.1 OMEGA's final proof object in $\mathbb{L}\mathbb{T}\mathbb{E}\mathbb{X}$

OMEGA's final proof object (see Appendix B.3 can be automatically transformed in a human readable  $\mathbb{L}\mathbb{T}\mathbb{E}\mathbb{X}$  representation. In order to illustrate the very close correspondence to Boolos' original proof the author has annotated the beginning of each line with the corresponding proof label of Boolos' original proof as given in Figure 1.

A1.	A1	$\vdash \forall N_{[c]}. F_{[(\iota, \iota) \rightarrow c]}(N, One_{[c]}) = S_{[\iota \rightarrow c]}(One)$	(Hyp)
A2.	A2	$\vdash \forall X_{[c]}. F(One, S(X)) = S(S(F(One, X)))$	(Hyp)
A3.	A3	$\vdash \forall N_{[c]}, X_{[c]}. F(S(N), S(X)) = F(N, F(S(N), X))$	(Hyp)
A4.	A4	$\vdash D_{[\iota \rightarrow o]}(One)$	(Hyp)
A5.	A5	$\vdash \forall X_{[c]}. [D(X) \Rightarrow D(S(X))]$	(Hyp)
<u>1</u> Ld1.	Ld1	$\vdash = Def_{[(\iota \rightarrow o, \iota \rightarrow o) \rightarrow o]}(Ld1_{[\iota \rightarrow o]}, \lambda Z_{[c]}. \forall X_{[\iota \rightarrow o]}. [[X(One) \wedge \forall Y_{[c]}. [X(Y) \Rightarrow X(S(Y))]] \Rightarrow X(Z)])$	(Local-Def)
<u>2</u> Ld2.	Ld2	$\vdash Ld2_{[\iota \rightarrow o]} := \lambda Z_{[c]}. [Ld1(Z) \wedge D(Z)]$	(Local-Def)
<u>3.1</u> L2.	$\kappa_1$	$\vdash \forall Z_{13[\iota \rightarrow o]}. [[Z_{13}(One) \wedge \forall Z_{17[\iota]}. [Z_{13}(Z_{17}) \Rightarrow Z_{13}(S(Z_{17}))]] \Rightarrow Z_{13}(One)]$	(Otter)
<u>3.1</u> L1.	$\kappa_1$	$\vdash Ld1(One)$	(CDef L2, Ld1)
<u>3.2</u> L5.	$\kappa_1$	$\vdash \forall Z_{48[\iota]}. [\forall Z_{59[\iota \rightarrow o]}. [[Z_{59}(One) \wedge \forall Z_{63[\iota]}. [Z_{59}(Z_{63}) \Rightarrow Z_{59}(S(Z_{63}))]] \Rightarrow Z_{59}(Z_{48})] \Rightarrow \forall Z_{68[\iota \rightarrow o]}. [[Z_{68}(One) \wedge \forall Z_{72[\iota]}. [Z_{68}(Z_{72}) \Rightarrow Z_{68}(S(Z_{72}))]] \Rightarrow Z_{68}(S(Z_{48}))]]$	(Otter)
<u>3.2</u> L4.	$\kappa_1$	$\vdash \forall Z_{26[\iota]}. [\forall Z_{37[\iota \rightarrow o]}. [[Z_{37}(One) \wedge \forall Z_{41[\iota]}. [Z_{37}(Z_{41}) \Rightarrow Z_{37}(S(Z_{41}))]] \Rightarrow Z_{37}(Z_{26})] \Rightarrow Ld1(S(Z_{26}))]$	(CDef L5, Ld1)
<u>3.2</u> L3.	$\kappa_1$	$\vdash \forall Y_{[\iota]}. [Ld1(Y) \Rightarrow Ld1(S(Y))]$	(CDef L4, Ld1)
<u>3.3/5</u> L6.	$\kappa_1$	$\vdash Ld1(S(S(S(S(One))))))$	(Otter L1, L3)
<u>3.4</u> L8.	$\kappa_1$	$\vdash [Ld1(One) \wedge D(One)]$	(Otter L1, A4)
<u>3.4</u> L7.	$\kappa_1$	$\vdash Ld2(One)$	(CDef L8, Ld2)
<u>3.5</u> L11.	$\kappa_1$	$\vdash \forall Z_{93[\iota]}. [[Ld1(Z_{93}) \wedge D(Z_{93})] \Rightarrow [Ld1(S(Z_{93})) \wedge D(S(Z_{93}))]]$	(Otter L3, A5)
<u>3.5</u> L10.	$\kappa_1$	$\vdash \forall Z_{86[\iota]}. [[Ld1(Z_{86}) \wedge D(Z_{86})] \Rightarrow Ld2(S(Z_{86}))]$	(CDef L11, Ld2)
<u>3.5</u> L9.	$\kappa_1$	$\vdash \forall Y_{[\iota]}. [Ld2(Y) \Rightarrow Ld2(S(Y))]$	(CDef L10, Ld2)
<u>3.6</u> L12.	$\kappa_1$	$\vdash Ld2(S(One))$	(Otter L7, L9)
<u>4</u> L13.	$\kappa_1$	$\vdash \forall N_{[c]}. [Ld1(N) \Rightarrow \forall X_{[c]}. [Ld1(X) \Rightarrow Ld2(F(N, X))]]$	(EDef L14, Ld3)
<u>4.1</u> Ld3.	Ld3	$\vdash Ld3_{[\iota \rightarrow o]} := \lambda N_{[c]}. \forall X_{[c]}. [Ld1(X) \Rightarrow Ld2(F(N, X))]$	(Local-Def)
<u>4.3.1.1</u> L18.	$\kappa_1$	$\vdash \forall Z_{151[\iota]}. [Ld1(Z_{151}) \Rightarrow Ld2(F(One, Z_{151}))]$	(EDef L19, Ld4)
<u>4.3.1</u> L15.	$\kappa_1$	$\vdash Ld3(One)$	(CDef L18, Ld3)
<u>4.3.2.4</u> L30.	$\kappa_2$	$\vdash \forall Z_{225[\iota]}. [Ld1(Z_{225}) \Rightarrow Ld2(F(S(N_{1[\iota]}), Z_{225}))]$	(EDef L31, Ld5)
<u>4.3.2.3</u> L28.	$\kappa_2$	$\vdash Ld3(S(N_1))$	(CDef L30, Ld3)
<u>4.3.2.1</u> L26.	$\kappa_1$	$\vdash [Ld3(N_1) \Rightarrow Ld3(S(N_1))]$	( $\Rightarrow I$ L28)
<u>4.3.2</u> L16.	$\kappa_1$	$\vdash \forall N_{[c]}. [Ld3(N) \Rightarrow Ld3(S(N))]$	( $\forall I$ L26)
<u>4.4-5</u> L17.	$\kappa_1$	$\vdash \forall Z_{123[\iota]}. [\forall Z_{134[\iota \rightarrow o]}. [[Z_{134}(One) \wedge \forall Z_{138[\iota]}. [Z_{134}(Z_{138}) \Rightarrow Z_{134}(S(Z_{138}))]] \Rightarrow Z_{134}(Z_{123})] \Rightarrow Ld3(Z_{123})]$	$\wedge$ (Otter L15, L16)
<u>4.2</u> L14.	$\kappa_1$	$\vdash \forall N_{[c]}. [Ld1(N) \Rightarrow Ld3(N)]$	(CDef L17, Ld1)

<u>4.3.1.2</u> Ld4.	Ld4	$\vdash Ld4_{[\ell \rightarrow o]} := \lambda X_{[\ell]} \bullet Ld2(F(One, X))$	(Local-Def)
<u>4.3.2.1</u> L27.	L27	$\vdash Ld3(N_1)$	(Hyp)
<u>4.3.2.2</u> L29.	$\mathcal{H}_3$	$\vdash \forall Z_{217}[\ell] \bullet [Ld1(Z_{217}) \Rightarrow Ld2(F(N_1, Z_{217}))]$	(EDef L27,Ld3)
<u>4.3.1.4.1.1-3</u> L23.		$\vdash Ld2(F(One, One))$	(Otter L12,A1)
<u>4.3.1.4.1.(1)</u> L23.		$\vdash Ld4(One)$	(CDef L23,Ld4)
<u>4.3.1.4.2.1-6</u> L25.		$\vdash \forall Z_{201}[\ell] \bullet [Ld2(F(One, Z_{201}))$ $Ld2(F(One, S(Z_{201})))]$	$\Rightarrow$ (Otter L9,A2)
<u>4.3.1.4.2.1-6</u> L24.		$\vdash \forall Z_{194}[\ell] \bullet [Ld2(F(One, Z_{194})) \Rightarrow Ld4(S(Z_{194}))]$	(CDef L25,Ld4)
<u>4.3.1.4.2.(1)</u> L24.		$\vdash \forall X_{[\ell]} \bullet [Ld4(X) \Rightarrow Ld4(S(X))]$	(CDef L24,Ld4)
<u>4.3.1.4</u> L22.	$\mathcal{H}_1$	$\vdash \forall Z_{165}[\ell] \bullet [\forall Z_{176}[\ell \rightarrow o] \bullet [[Z_{176}(One)$ $\forall Z_{180}[\ell] \bullet [Z_{176}(Z_{180}) \Rightarrow Z_{176}(S(Z_{180}))]]$ $Z_{176}(Z_{165})] \Rightarrow Ld4(Z_{165})]$	$\wedge$ (Otter L20,L21) $\Rightarrow$
<u>4.3.1.3</u> L19.	$\mathcal{H}_1$	$\vdash \forall X_{[\ell]} \bullet [Ld1(X) \Rightarrow Ld4(X)]$	(CDef L22,Ld1)
<u>4.3.2.5</u> Ld5.	Ld5	$\vdash Ld5_{[\ell \rightarrow o]} := \lambda X_{[\ell]} \bullet Ld2(F(S(N_1), X))$	(Local-Def)
<u>4.3.2.7.2.1</u> L37L37		$\vdash Ld5(X_{1[\ell]})$	(Hyp)
<u>4.3.2.7.2.2</u> L39Ld5,		$\vdash Ld2(F(S(N_1), X_1))$	(EDef L37,Ld5)
	L37		
<u>4.3.2.7.2.3</u> L40 $\mathcal{H}_4$		$\vdash [Ld1(F(S(N_1), X_1)) \wedge D(F(S(N_1), X_1))]$	(EDef L39,Ld2)
<u>6</u> L45.	$\mathcal{H}_1$	$\vdash Ld2(F(S(S(S(S(One))))), S(S(S(S(One))))))$	(Otter L6,L13)
<u>7</u> L46.	$\mathcal{H}_1$	$\vdash [Ld1(F(S(S(S(S(One))))), S(S(S(S(One))))))$ $D(F(S(S(S(S(One))))), S(S(S(S(One))))))]$	$\wedge$ (EDef L45,Ld2)
<u>4.3.2.7.2.3</u> L42 $\mathcal{H}_4$		$\vdash D(F(S(N_1), X_1))$	( $\wedge$ E L40)
<u>4.3.2.7.2.3</u> L41 $\mathcal{H}_4$		$\vdash Ld1(F(S(N_1), X_1))$	( $\wedge$ E L40)
<u>4.3.2.7.1.1-3</u> L38.		$\vdash Ld2(F(S(N_1), One))$	(Otter L12,A1)
<u>4.3.2.7.1</u> L32.	$\mathcal{H}_2$	$\vdash Ld5(One)$	(CDef L35,Ld5)
<u>4.3.2.7.2.5</u> L44 $\mathcal{H}_5$		$\vdash Ld2(F(N_1, F(S(N_1), X_1)))$	(Otter L29,L41)
<u>4.3.2.7.2.4.6.7</u> L43.		$\vdash Ld2(F(S(N_1), S(X_1)))$	(Otter A3,L44)
<u>4.3.2.7.2.4</u> L38 $\mathcal{H}_5$		$\vdash Ld5(S(X_1))$	(CDef L43,Ld5)
<u>4.3.2.7.2.1</u> L36 $\mathcal{H}_2$		$\vdash [Ld5(X_1) \Rightarrow Ld5(S(X_1))]$	( $\Rightarrow$ I L38)
<u>4.3.2.7.2</u> L33.	$\mathcal{H}_2$	$\vdash \forall X_{[\ell]} \bullet [Ld5(X) \Rightarrow Ld5(S(X))]$	( $\forall$ I L36)
<u>4.3.2.7</u> L34.	$\mathcal{H}_2$	$\vdash \forall Z_{239}[\ell] \bullet [\forall Z_{250}[\ell \rightarrow o] \bullet [[Z_{250}(One)$ $\forall Z_{254}[\ell] \bullet [Z_{250}(Z_{254}) \Rightarrow Z_{250}(S(Z_{254}))]]$ $Z_{250}(Z_{239})] \Rightarrow Ld5(Z_{239})]$	$\wedge$ (Otter L32,L33) $\Rightarrow$
<u>4.3.2.6</u> L31.	$\mathcal{H}_2$	$\vdash \forall X_{[\ell]} \bullet [Ld1(X) \Rightarrow Ld5(X)]$	(CDef L34,Ld1)
<u>0/7</u> Conc.	$\mathcal{H}_1$	$\vdash D(F(S(S(S(S(One))))), S(S(S(S(One))))))$	(Otter L46)

$\mathcal{H}_1 = A1, A2, A3, A4, A5, Ld1, Ld2, Ld3, Ld4, Ld5$   
 $\mathcal{H}_2 = A1, A2, A3, A4, A5, Ld1, Ld2, Ld3, Ld4, Ld5, L27$   
 $\mathcal{H}_3 = Ld3, Ld5, L27$   
 $\mathcal{H}_4 = Ld2, Ld5, L37$   
 $\mathcal{H}_5 = A1, A2, A3, A4, A5, Ld1, Ld2, Ld3, Ld4, Ld5, L27, L37$   
 CDef = Defn-Contract-Local-Def  
 EDef = Defn-Expand-Local-Def  
 LD1 = N  
 LD2 = E  
 LD3 = M  
 LD4 = Q  
 LD5 = P

## B.2 The final proof script in OMEGA

The following (proof step annotated) OMEGA proof script has been obtained by automatically storing the commands of the interactive session presented in Ap-



pendix B.5 in a file. Replaying this script reproduces the final proof project presented in Appendix B.3.

```

;;; step 1
OMEGA-BASIC LOCAL-DEF-INTRO ((LAM (Z I)
  (FORALL
    (LAM
      (X (O I))
      (IMPLIES
        (AND
          (X ONE)
          (FORALL
            (LAM
              (Y I)
              (IMPLIES (X Y) (X (S Y))))))
          (X Z))))))

;;; step 2
OMEGA-BASIC LOCAL-DEF-INTRO ((LAM (Z I) (AND (LD1 Z) (D Z))))
;;; step 3.1
OMEGA-BASIC LEMMA (CONC ((LD1 ONE))
;;; step 3.1
RULES DEFN-CONTRACT-LOCAL-DEF (L1) (NIL) (LD1) ((0))
;;; step 3.1
OMEGA-BASIC SUPPORT (L2) (NIL)
;;; step 3.1
EXTERN CALL-OTTER-ON-NODE (L2) default default (TEST) default default default default default default default default default
default

;;; step 3.2
OMEGA-BASIC LEMMA (CONC ((FORALL
  (LAM (Y I) (IMPLIES (LD1 Y) (LD1 (S Y))))))
;;; step 3.2
RULES DEFN-CONTRACT-LOCAL-DEF (L3) (NIL) (LD1) ((1 0 1 0))
;;; step 3.2
RULES DEFN-CONTRACT-LOCAL-DEF (L4) (NIL) (LD1) ((1 0 2 0))
;;; step 3.2
OMEGA-BASIC SUPPORT (L5) (NIL)
;;; step 3.2
EXTERN CALL-OTTER-ON-NODE (L5) default default (TEST) default default default default default default default default default
default

;;; step 3.3
OMEGA-BASIC LEMMA (CONC ((LD1 (S (S (S ONE))))))
;;; step 3.3
OMEGA-BASIC SUPPORT (L6) ((L1 L3))
;;; step 3.3
EXTERN CALL-OTTER-ON-NODE (L6) default default (TEST) default default default default default default default default default
default

;;; step 3.4
OMEGA-BASIC LEMMA (CONC ((LD2 ONE))
;;; step 3.4
RULES DEFN-CONTRACT-LOCAL-DEF (L7) (NIL) (LD2) ((0))
;;; step 3.4
OMEGA-BASIC SUPPORT (L8) ((A4 L1))
;;; step 3.4
EXTERN CALL-OTTER-ON-NODE (L8) default default (TEST) default default default default default default default default default
default

;;; step 3.5
OMEGA-BASIC LEMMA (CONC ((FORALL
  (LAM (Y I) (IMPLIES (LD2 Y) (LD2 (S Y))))))
;;; step 3.5
RULES DEFN-CONTRACT-LOCAL-DEF (L9) (NIL) (LD2) ((1 0 1 0))
;;; step 3.5
RULES DEFN-CONTRACT-LOCAL-DEF (L10) (NIL) (LD2) ((1 0 2 0))
;;; step 3.5
OMEGA-BASIC SUPPORT (L11) ((A5 L3))
;;; step 3.5
EXTERN CALL-OTTER-ON-NODE (L11) default default (TEST) default default default default default default default default default
default

;;; step 3.6
OMEGA-BASIC LEMMA (CONC ((LD2 (S ONE)))
;;; step 3.6
OMEGA-BASIC SUPPORT (L12) ((L7 L9))
;;; step 3.6
EXTERN CALL-OTTER-ON-NODE (L12) default default (TEST) default default default default default default default default default
default

;;; step 4
OMEGA-BASIC LEMMA (CONC ((FORALL
  (LAM (N I)
    (IMPLIES (LD1 N)
      (FORALL
        (LAM (X I)
          (IMPLIES (LD1 X) (LD2 (F N X))))))))))
;;; step 4.1
OMEGA-BASIC LOCAL-DEF-INTRO ((LAM (N I)
  (FORALL
    (LAM
      (X I)
      (IMPLIES (LD1 X) (LD2 (F N X))))))
;;; step 4.2
OMEGA-BASIC LEMMA (L13) ((FORALL (LAM (N I) (IMPLIES (LD1 N) (LD3 N))))))
;;; step 4.2

```

```

RULES DEFN-EXPAND-LOCAL-DEF (L13) (L14) (LD3) ((1 0 2 0))
;; step 4.3.1
OMEGA-BASIC LEMMA (L14) ((LD3 ONE))
;; step 4.3.2
OMEGA-BASIC LEMMA (L14) ((FORALL
(LAM (N I) (IMPLIES (LD3 N) (LD3 (S N))))))
;; step 4.3 -- Enough to show ...
RULES DEFN-CONTRACT-LOCAL-DEF (L14) (NIL) (LD1) ((1 0 1 0))
;; step 4.3
OMEGA-BASIC SUPPORT (L17) ((L15 L16))
;; step 4.3
EXTERN CALL-UTTER-ON-NODE (L17) default default (TEST) default default default default default default default default default
default

;; step 4.3.1.1
RULES DEFN-CONTRACT-LOCAL-DEF (L15) (NIL) (LD3) ((0))
;; step 4.3.1.2
OMEGA-BASIC LOCAL-DEF-INTRO ((LAM (X I) (LD2 (F ONE X))))
;; step 4.3.1.3
OMEGA-BASIC LEMMA (L18) ((FORALL (LAM (X I) (IMPLIES (LD1 X) (LD4 X)))))
;; step 4.3.1.3
RULES DEFN-EXPAND-LOCAL-DEF (L18) (L19) (LD4) ((1 0 2 0))
;; step 4.3.1.4.1
OMEGA-BASIC LEMMA (L19) ((LD4 ONE))
;; step 4.3.1.4.2
OMEGA-BASIC LEMMA (L19) ((FORALL
(LAM (X I) (IMPLIES (LD4 X) (LD4 (S X))))))
;; step 4.3.1.4 -- Enough to show ...
RULES DEFN-CONTRACT-LOCAL-DEF (L19) (NIL) (LD1) ((1 0 1 0))
;; step 4.3.1.4
OMEGA-BASIC SUPPORT (L22) ((L20 L21))
;; step 4.3.1.4
EXTERN CALL-UTTER-ON-NODE (L22) default default (TEST) default default default default default default default default default
default

;; step 4.3.1.4.1.1
RULES DEFN-CONTRACT-LOCAL-DEF (L20) (NIL) (LD4) ((0))
;; step 4.3.1.4.1.2-3
OMEGA-BASIC SUPPORT (L23) ((A1 L12))
;; step 4.3.1.4.1.2-3
EXTERN CALL-UTTER-ON-NODE (L23) default default (TEST) default default default default default default default default default
default

;; step 4.3.1.4.2.1-6
RULES DEFN-CONTRACT-LOCAL-DEF (L21) (NIL) (LD4) ((1 0 1 0))
;; step 4.3.1.4.2.1-6
RULES DEFN-CONTRACT-LOCAL-DEF (L24) (NIL) (LD4) ((1 0 2 0))
;; step 4.3.1.4.2.1-6
OMEGA-BASIC SUPPORT (L25) ((A2 L9))
;; step 4.3.1.4.2.1-6
EXTERN CALL-UTTER-ON-NODE (L25) default default (TEST) default default default default default default default default default
default

;; step 4.3.2.1
RULES FORALLI default default default
;; step 4.3.2.1.3
RULES IMPI default
;; step 4.3.2.2
RULES DEFN-EXPAND-LOCAL-DEF (NIL) (L27) (LD3) ((0))
;; step 4.3.2.4
RULES DEFN-CONTRACT-LOCAL-DEF (L28) (NIL) (LD3) ((0))
;; step 4.3.2.5
OMEGA-BASIC LOCAL-DEF-INTRO ((LAM (X I) (LD2 (F (S N1) X))))
;; step 4.3.2.6
OMEGA-BASIC LEMMA default ((FORALL
(LAM (X I) (IMPLIES (LD1 X) (LD5 X)))))
;; step 4.3.2.6
RULES DEFN-EXPAND-LOCAL-DEF (L30) (L31) (LD5) ((1 0 2 0))
;; step 4.3.2.7.1
OMEGA-BASIC LEMMA (L31) ((LD5 ONE))
;; step 4.3.2.7.2
OMEGA-BASIC LEMMA (L31) ((FORALL
(LAM (X I) (IMPLIES (LD5 X) (LD5 (S X)))))
;; step 4.3.2.7 -- Enough to ...
RULES DEFN-CONTRACT-LOCAL-DEF (L31) (NIL) (LD1) ((1 0 1 0))
;; step 4.3.2.7
OMEGA-BASIC SUPPORT (L34) ((L32 L33))
;; step 4.3.2.7
EXTERN CALL-UTTER-ON-NODE (L34) default default (TEST) default default default default default default default default default
default

;; step 4.3.2.7.1
RULES DEFN-CONTRACT-LOCAL-DEF (L32) (NIL) (LD5) ((0))
;; step 4.3.2.7.2-3
OMEGA-BASIC SUPPORT (L35) ((A1 L12))
;; step 4.3.2.7.2-3
EXTERN CALL-UTTER-ON-NODE (L35) default default (TEST) default default default default default default default default default
default

;; step 4.3.2.7.2.1
RULES FORALLI default default default
;; step 4.3.2.7.2.1
RULES IMPI default
;; step 4.3.2.7.2.2
RULES DEFN-EXPAND-LOCAL-DEF (NIL) (L37) (LD5) ((0))
;; step 4.3.2.7.2.3
RULES DEFN-EXPAND-LOCAL-DEF (NIL) (L39) (LD2) ((0))

```

```

;;; step 4.3.2.7.2.3
TACTICS ANDE (L40) default default
;;; step 4.3.2.7.2.4
RULES DEFN-CONTRACT-LOCAL-DEF (L38) (NIL) (LD5) ((0))
;;; step 4.3.2.7.2.5
OMEGA-BASIC LEMMA (L43) ((LD2 (F N1 (F (S N1) X1))))
;;; step 4.3.2.7.2.5
OMEGA-BASIC SUPPORT (L44) ((L41 L29))
;;; step 4.3.2.7.2.5
EXTERN CALL-OTTER-ON-NODE (L44) default default (TEST) default default default default default default default default default
default

;;; step 4.3.2.7.2.6
OMEGA-BASIC SUPPORT (L43) ((L44 A3))
;;; step 4.3.2.7.2.7
EXTERN CALL-OTTER-ON-NODE (L43) default default (TEST) default default default default default default default default default
default

;;; step 6
OMEGA-BASIC LEMMA (CONC) ((ld2 (F (S (S (S ONE)))) (S (S (S ONE))))))
;;; step 6
OMEGA-BASIC SUPPORT (L45) ((L6 L13))
;;; step 6
EXTERN CALL-OTTER-ON-NODE (L45) default default (TEST) default default default default default default default default default
default

;;; step 7
RULES DEFN-EXPAND-LOCAL-DEF (NIL) (L45) (LD2) ((0))
;;; step 7
OMEGA-BASIC SUPPORT (CONC) ((L46))
;;; step 7
EXTERN CALL-OTTER-ON-NODE (CONC) default default (TEST) default default default default default default default default default
default

```

### B.3 OMEGA's final proof object in ASCII format

```

(PDS (problem BOOLOS-CURIOUS-INFERENCE)
  (in BOOLOS)
  (declarations (type-variables )(type-constants )
    (constants (X1 I)
      (LD5 (O I))
      (N1 I)
      (LD4 (O I))
      (LD3 (O I))
      (LD2 (O I))
      (LD1 (O I)))
    (meta-variables )(variables ))
  (conclusion CONC)
  (assumptions A1 A2 A3 A4 A5)
  (open-nodes)
  (support-nodes LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5)
  (nodes
    (A1 (A1) (FORALL (lam (VAR0 I) (= (F VAR0 ONE) (S ONE))))
      (O ("HYP" (O) () "grounded" (O) ()))
    )
    (A2 (A2) (FORALL (lam (VAR3 I) (= (F ONE (S VAR3)) (S (S (F ONE VAR3))))))
      (O ("HYP" (O) () "grounded" (O) ()))
    )
    (A3 (A3) (FORALL (lam (VAR4 I) (FORALL (lam (VAR5 I) (= (F (S VAR4) (S VAR5)) (F VAR4 (F (S VAR4) VAR5))))))
      (O ("HYP" (O) () "grounded" (O) ()))
    )
    (A4 (A4) (D ONE)
      (O ("HYP" (O) () "grounded" (O) ()))
    )
    (A5 (A5) (FORALL (lam (VAR6 I) (IMPLIES (D VAR6) (D (S VAR6))))
      (O ("HYP" (O) () "grounded" (O) ()))
    )
    (LD1 (LD1) (=DEF LD1 (lam (VAR7 I) (FORALL (lam (VAR8 (O I)) (IMPLIES (AND (VAR8 ONE) (FORALL (lam (VAR9 I)
      (IMPLIES (VAR8 VAR9) (VAR8 (S VAR9)))))) (VAR8 VAR7))))))
      (O ("LOCAL-DEF" (O) () "grounded" (O) ()))
    )
    (LD2 (LD2) (=DEF LD2 (lam (VAR10 I) (AND (LD1 VAR10) (D VAR10))))
      (O ("LOCAL-DEF" (O) () "grounded" (O) ()))
    )
    (L2 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR11 (O I)) (IMPLIES (AND (VAR11 ONE)
      (FORALL (lam (VAR12 I) (IMPLIES (VAR11 VAR12) (VAR11 (S VAR12)))))) (VAR11 ONE))))
      (1 ("OTTER" ([:pds-nil]) (O) "expanded" (O) ("EXISTENT"))
        ("OTTER" ([:pds-nil]) (O) "untested" (O) ()))
    )
    (L1 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD1 ONE)
      (O ("DEFN-CONTRACT-LOCAL-DEF" ([:pds-post-obj (position 0)]) (L2 LD1) "grounded"
        (O) ("EXISTENT" "NONEXISTENT" "EXISTENT")))
    )
    (L5 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR13 I) (IMPLIES (FORALL (lam (VAR14 (O I))
      (IMPLIES (AND (VAR14 ONE) (FORALL (lam (VAR15 I) (IMPLIES (VAR14 VAR15) (VAR14 (S VAR15)))))) (VAR14 VAR13))))
      (FORALL (lam (VAR16 (O I)) (IMPLIES (AND (VAR16 ONE) (FORALL (lam (VAR17 I) (IMPLIES (VAR16 VAR17)
      (VAR16 (S VAR17)))))) (VAR16 (S VAR13))))))
      (1 ("OTTER" ([:pds-nil]) (O) "expanded" (O) ("EXISTENT"))
        ("OTTER" ([:pds-nil]) (O) "untested" (O) ()))
    )
    (L4 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR18 I) (IMPLIES (FORALL (lam (VAR19 (O I))

```

```

(IMPLIES (AND (VAR19 ONE) (FORALL (lam (VAR20 I) (IMPLIES (VAR19 VAR20) (VAR19 (S VAR20)))))) (VAR19 VAR18))))
(LD1 (S VAR18))))
  (0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 1 0 2 0))) (L5 LD1) "grounded"
    () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
  )
(L3 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR21 I) (IMPLIES (LD1 VAR21) (LD1 (S VAR21))))))
  (0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 1 0 1 0))) (L4 LD1) "grounded"
    () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
  )
(L6 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD1 (S (S (S (S ONE))))))
  (1 ("OTTER" (:pds-nil)) (L3 L1) "expanded" ()
    ("EXISTENT" "CLOSED" "CLOSED"))
  ("OTTER" (:pds-nil)) (L1 L3) "untested" () ()))
  )
(L8 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (AND (LD1 ONE) (D ONE))
  (1 ("OTTER" (:pds-nil)) (A4 L1) "expanded" ()
    ("EXISTENT" "CLOSED" "CLOSED"))
  ("OTTER" (:pds-nil)) (L1 A4) "untested" () ()))
  )
(L7 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD2 ONE)
  (0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 0))) (L8 LD2) "grounded"
    () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
  )
(L11 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR22 I) (IMPLIES (AND (LD1 VAR22) (D VAR22))
  (AND (LD1 (S VAR22)) (D (S VAR22))))))
  (1 ("OTTER" (:pds-nil)) (A5 L3) "expanded" ()
    ("EXISTENT" "CLOSED" "CLOSED"))
  ("OTTER" (:pds-nil)) (L3 A5) "untested" () ()))
  )
(L10 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR23 I) (IMPLIES (AND (LD1 VAR23) (D VAR23))
(LD2 (S VAR23))))))
  (0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 1 0 2 0))) (L11 LD2) "grounded"
    () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
  )
(L9 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR24 I) (IMPLIES (LD2 VAR24) (LD2 (S VAR24))))))
  (0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 1 0 1 0))) (L10 LD2) "grounded"
    () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
  )
(L12 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD2 (S ONE))
  (1 ("OTTER" (:pds-nil)) (L9 L7) "expanded" ()
    ("EXISTENT" "CLOSED" "CLOSED"))
  ("OTTER" (:pds-nil)) (L7 L9) "untested" () ()))
  )
(L13 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR25 I) (IMPLIES (LD1 VAR25) (FORALL (lam (VAR26 I)
(IMPLIES (LD1 VAR26) (LD2 (F VAR26 VAR26)))))))))
  (0 ("DEFN-EXPAND-LOCAL-DEF" (:pds-post-obj (position 1 0 2 0))) (L14 LD3) "grounded"
    () ("EXISTENT" "EXISTENT" "EXISTENT"))))
  )
(LD3 (LD3) (=DEF LD3 (lam (VAR27 I) (FORALL (lam (VAR28 I) (IMPLIES (LD1 VAR28) (LD2 (F VAR27 VAR28))))))
  (0 ("LOCAL-DEF" () () "grounded" () ()))
  )
  )
(L18 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR29 I) (IMPLIES (LD1 VAR29) (LD2 (F ONE VAR29))))))
  (0 ("DEFN-EXPAND-LOCAL-DEF" (:pds-post-obj (position 1 0 2 0))) (L19 LD4) "grounded"
    () ("EXISTENT" "EXISTENT" "EXISTENT"))))
  )
(L15 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD3 ONE)
  (0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 0))) (L18 LD3) "grounded"
    () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
  )
(L30 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR30 I) (IMPLIES (LD1 VAR30)
(LD2 (F (S N1) VAR30))))))
  (0 ("DEFN-EXPAND-LOCAL-DEF" (:pds-post-obj (position 1 0 2 0))) (L31 LD5) "grounded"
    () ("EXISTENT" "EXISTENT" "EXISTENT"))))
  )
(L28 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD3 (S N1))
  (0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 0))) (L30 LD3) "grounded"
    () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
  )
(L26 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (IMPLIES (LD3 N1) (LD3 (S N1)))
  (0 ("IMP1" () (L28) "grounded" () ("EXISTENT" "NONEXISTENT"))))
  )
(L16 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR31 I) (IMPLIES (LD3 VAR31) (LD3 (S VAR31))))))
  (0 ("FURALLI" (:pds-term N1)) (L26) "grounded" ()
    ("EXISTENT" "NONEXISTENT"))))
  )
(L17 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR32 I) (IMPLIES (FORALL (lam (VAR33 (O I))
(IMPLIES (AND (VAR33 ONE) (FORALL (lam (VAR34 I) (IMPLIES (VAR33 VAR34) (VAR33 (S VAR34)))))) (VAR33 VAR32))))))
(LD3 VAR32))))
  (1 ("OTTER" (:pds-nil)) (L16 L15) "expanded" ()
    ("EXISTENT" "EXISTENT" "EXISTENT"))
  ("OTTER" (:pds-nil)) (L15 L16) "untested" () ()))
  )
(L14 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR35 I) (IMPLIES (LD1 VAR35) (LD3 VAR35))))
  (0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 1 0 1 0))) (L17 LD1) "grounded"
    () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
  )
(LD4 (LD4) (=DEF LD4 (lam (VAR36 I) (LD2 (F ONE VAR36))))
  (0 ("LOCAL-DEF" () () "grounded" () ()))
  )
  )
(L27 (LD2) (LD3 N1)
  (0 ("HYP" () () "grounded" () ()))
  )

```

```

)
(L29 (LD5 LD3 L27) (FORALL (lam (VAR37 I) (IMPLIES (LD1 VAR37) (LD2 (F N1 VAR37))))))
(0 ("DEFN-EXPAND-LOCAL-DEF" (:pds-post-obj (position 0))) (L27 LD3) "grounded"
  () ("NONEXISTENT" "EXISTENT" "EXISTENT"))))
)
(L23 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD2 (F ONE ONE))
(1 ("OTTER" (:pds-nil)) (A1 L12) "expanded" ()
  ("EXISTENT" "CLOSED" "CLOSED"))
  ("OTTER" (:pds-nil)) (L12 A1) "untested" () ()))
)
(L20 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD4 ONE)
(0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 0))) (L23 LD4) "grounded"
  () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
)
(L25 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR38 I) (IMPLIES (LD2 (F ONE VAR38))
(LD2 (F ONE (S VAR38))))))
(1 ("OTTER" (:pds-nil)) (A2 L9) "expanded" ()
  ("EXISTENT" "CLOSED" "CLOSED"))
  ("OTTER" (:pds-nil)) (L9 A2) "untested" () ()))
)
(L24 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR39 I) (IMPLIES (LD2 (F ONE VAR39))
(LD4 (S VAR39))))))
(0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 1 0 2 0))) (L25 LD4) "grounded"
  () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
)
(L21 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR40 I) (IMPLIES (LD4 VAR40) (LD4 (S VAR40))))))
(0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 1 0 1 0))) (L24 LD4) "grounded"
  () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
)
(L22 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR41 I) (IMPLIES (FORALL (lam (VAR42 (0 I))
(IMPLIES (AND (VAR42 ONE) (FORALL (lam (VAR43 I) (IMPLIES (VAR42 VAR43) (VAR42 (S VAR43)))))) (VAR42 VAR41))))))
(LD4 VAR41))))))
(1 ("OTTER" (:pds-nil)) (L21 L20) "expanded" ()
  ("EXISTENT" "EXISTENT" "EXISTENT"))
  ("OTTER" (:pds-nil)) (L20 L21) "untested" () ()))
)
(L19 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR44 I) (IMPLIES (LD1 VAR44) (LD4 VAR44))))))
(0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 1 0 1 0))) (L22 LD1) "grounded"
  () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
)
(LD5 (LD5) (=DEF LD5 (lam (VAR45 I) (LD2 (F (S N1) VAR45))))
(0 ("LOCAL-DEF" () () "grounded" () ()))
)
(L37 (L37) (LD5 X1)
(0 ("HYP" () () "grounded" () ()))
)
(L39 (LD5 L37) (LD2 (F (S N1) X1))
(0 ("DEFN-EXPAND-LOCAL-DEF" (:pds-post-obj (position 0))) (L37 LD5) "grounded"
  () ("NONEXISTENT" "EXISTENT" "EXISTENT"))))
)
(L40 (LD2 LD5 L37) (AND (LD1 (F (S N1) X1)) (D (F (S N1) X1))))
(0 ("DEFN-EXPAND-LOCAL-DEF" (:pds-post-obj (position 0))) (L39 LD2) "grounded"
  () ("NONEXISTENT" "EXISTENT" "EXISTENT"))))
)
(L45 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD2 (F (S (S (S ONE)))) (S (S (S ONE))))))
(1 ("OTTER" (:pds-nil)) (L13 L6) "expanded" ()
  ("EXISTENT" "CLOSED" "CLOSED"))
  ("OTTER" (:pds-nil)) (L6 L13) "untested" () ()))
)
(L46 (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (AND (LD1 (F (S (S (S ONE)))) (S (S (S ONE))))))
(D (F (S (S (S ONE)))) (S (S (S ONE))))))
(0 ("DEFN-EXPAND-LOCAL-DEF" (:pds-post-obj (position 0))) (L45 LD2) "grounded"
  () ("NONEXISTENT" "EXISTENT" "EXISTENT"))))
)
(L42 (LD2 LD5 L37) (D (F (S N1) X1))
(0 ("ANDE" () (L40) "unexpanded" ()
  ("L41" "NONEXISTENT" "EXISTENT"))))
)
(L41 (LD2 LD5 L37) (LD1 (F (S N1) X1))
(0 ("ANDE" () (L40) "unexpanded" ()
  ("NONEXISTENT" "L42" "EXISTENT"))))
)
(L35 (LD5 L27 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD2 (F (S N1) ONE))
(1 ("OTTER" (:pds-nil)) (A1 L12) "expanded" ()
  ("EXISTENT" "CLOSED" "CLOSED"))
  ("OTTER" (:pds-nil)) (L12 A1) "untested" () ()))
)
(L32 (LD5 L27 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD5 ONE)
(0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 0))) (L35 LD5) "grounded"
  () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
)
(L44 (L37 LD5 L27 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD2 (F N1 (F (S N1) X1)))
(1 ("OTTER" (:pds-nil)) (L41 L29) "expanded" ()
  ("EXISTENT" "CLOSED" "CLOSED"))
  ("OTTER" (:pds-nil)) (L29 L41) "untested" () ()))
)
(L43 (L37 LD5 L27 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD2 (F (S N1) (S X1)))
(1 ("OTTER" (:pds-nil)) (L44 A3) "expanded" ()
  ("EXISTENT" "CLOSED" "CLOSED"))
  ("OTTER" (:pds-nil)) (A3 L44) "untested" () ()))
)

```

```

(L38 (L37 LD5 L27 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (LD5 (S X1))
  (0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 0))) (L43 LD5) "grounded"
    () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
)
(L36 (LD5 L27 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (IMPLIES (LD5 X1) (LD5 (S X1)))
  (0 ("IMPI" () (L38) "grounded" () ("EXISTENT" "NONEXISTENT"))))
)
(L33 (LD5 L27 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR46 I) (IMPLIES (LD5 VAR46) (LD5 (S VAR46)))))
  (0 ("FORALLI" (:pds-term X1)) (L36) "grounded" ()
    ("EXISTENT" "NONEXISTENT"))))
)
(L34 (LD5 L27 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR47 I) (IMPLIES (FORALL (lam (VAR48 (O I))
  (IMPLIES (AND (VAR48 ONE) (FORALL (lam (VAR49 I) (IMPLIES (VAR48 VAR49) (VAR48 (S VAR49)))))) (VAR48 VAR47))))))
  (LD5 VAR47))))
  (1 ("OTTER" (:pds-nil)) (L33 L32) "expanded" ()
    ("EXISTENT" "EXISTENT" "EXISTENT"))
  ("OTTER" (:pds-nil)) (L32 L33) "untested" () ()))
)
(L31 (LD5 L27 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (FORALL (lam (VAR50 I) (IMPLIES (LD1 VAR50) (LD5 VAR50))))
  (0 ("DEFN-CONTRACT-LOCAL-DEF" (:pds-post-obj (position 1 0 1 0))) (L34 LD1) "grounded"
    () ("EXISTENT" "NONEXISTENT" "EXISTENT"))))
)
(CONC (LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5) (D (F (S (S (S (S ONE)))))) (S (S (S (S ONE))))))
  (1 ("OTTER" (:pds-nil)) (L46) "expanded" ()
    ("EXISTENT" "CLOSED"))
  ("OTTER" (:pds-nil)) (L46) "untested" () ()))
)
(lemmata)
(agenda)
(controls
  (A1 (O O O O))
  (A2 (O O O O))
  (A3 (O O O O))
  (A4 (O O O O))
  (A5 (O O O O))
  (LD1 (O O O O))
  (LD2 (O O O O))
  (L2 (O (LD2 LD1 A1 A2 A3 A4 A5) (O) (O) (O)))
  (L1 (O O O O))
  (L5 ((L1) (L1 LD2 LD1 A1 A2 A3 A4 A5) (O) (O) (O)))
  (L4 ((L1) (O O O)))
  (L3 ((L1) (O O O)))
  (L6 ((L3 L1) (A5 A4 A3 A2 A1 LD1 LD2) (O) (O) (O O O)))
  (L8 ((L6 L3 L1) (A5 A3 A2 A1 LD1 LD2 L6 L3) (O) (O) (O)))
  (L7 ((L6 L3 L1) (O O O)))
  (L11 ((L7 L6 L3 L1) (A4 A3 A2 A1 LD1 LD2 L7 L6 L1) (O) (O) (O)))
  (L10 ((L7 L6 L3 L1) (O O O)))
  (L9 ((L7 L6 L3 L1) (O O O)))
  (L12 ((L9 L7 L6 L3 L1) (A5 A4 A3 A2 A1 LD1 LD2 L6 L3 L1) (O) (O)
    (O O O O)))
  (L13 ((L14 L12 L9 L7 L6 L3 L1) (O O O)))
  (LD3 (O O O O))
  (L18 ((L19 L12 L9 L7 L6 L3 L1) (O O O)))
  (L15 ((L12 L9 L7 L6 L3 L1) (O O O)))
  (L30 ((L31 L27 L15 L12 L9 L7 L6 L3 L1) (O O O)))
  (L28 ((L27 L15 L12 L9 L7 L6 L3 L1) (O O O)))
  (L26 ((L15 L12 L9 L7 L6 L3 L1) (O O O)))
  (L16 ((L15 L12 L9 L7 L6 L3 L1) (O O O)))
  (L17 ((L16 L15 L12 L9 L7 L6 L3 L1) (A5 A4 A3 A2 A1 LD1 LD2 LD3 L12 L9 L7 L6 L3 L1) (O) (O)
    (O)))
  (L14 ((L16 L15 L12 L9 L7 L6 L3 L1) (O O O)))
  (LD4 (O O O O))
  (L27 (O O O O))
  (L29 (O O O O))
  (L23 ((L12 L9 L7 L6 L3 L1) (A5 A4 A3 A2 LD1 LD2 LD3 LD4 L9 L7 L6 L3 L1) (O) (O)
    (O)))
  (L20 ((L12 L9 L7 L6 L3 L1) (O O O)))
  (L25 ((L20 L12 L9 L7 L6 L3 L1) (A5 A4 A3 A1 LD1 LD2 LD3 LD4 L20 L12 L7 L6 L3 L1) (O) (O)
    (O)))
  (L24 ((L20 L12 L9 L7 L6 L3 L1) (O O O)))
  (L21 ((L20 L12 L9 L7 L6 L3 L1) (O O O)))
  (L22 ((L21 L20 L12 L9 L7 L6 L3 L1) (A5 A4 A3 A2 A1 LD1 LD2 LD3 LD4 L12 L9 L7 L6 L3 L1) (O) (O)
    (O)))
  (L19 ((L21 L20 L12 L9 L7 L6 L3 L1) (O O O)))
  (LD5 (O O O O))
  (L37 (O O O O))
  (L39 (O O O O))
  (L40 ((L13 L37 L32 L27 L15 L12 L9 L7 L6 L3 L1) (O O O)))
  (L45 ((L40 L13 L12 L9 L7 L6 L3 L1) (A5 A4 A3 A2 A1 LD1 LD2 LD3 LD4 LD5 L40 L12 L9 L7 L3 L1) (O) (O)
    (O O O O)))
  (L46 (O O O O))
  (L42 (O O O O))
  (L41 (O O O O))
  (L35 ((L27 L15 L12 L9 L7 L6 L3 L1) (A5 A4 A3 A2 LD1 LD2 LD3 LD4 LD5 L27 L15 L9 L7 L6 L3 L1) (O) (O)
    (O)))
  (L32 ((L27 L15 L12 L9 L7 L6 L3 L1) (O O O)))
  (L44 ((L29 L41 L1 L3 L6 L7 L9 L12 L15 L27 L32 L37 L40) (A5 A4 A3 A2 A1 LD1 LD2 LD3 LD4 LD5 L1 L3 L6 L7 L9 L12
  L15 L27 L32 L37 L40) (O O)
    (O O O O)))
  (L43 ((L44 L40 L37 L32 L27 L15 L12 L9 L7 L6 L3 L1) (L1 L3 L6 L7 L9 L12 L15 L27 L32 L37 L40 LD5 LD4 LD3 LD2 LD1
  A1 A2 A4 A5) (O O)
    (O)))
)

```

*The Curious Inference of Boolos in Mizar and OMEGA*

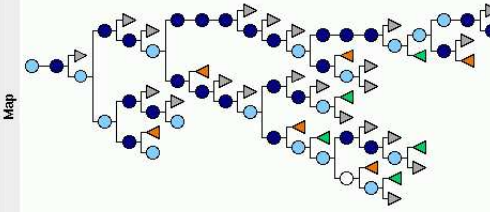
```
( )
(L38 ((L40 L37 L32 L27 L15 L12 L9 L7 L6 L3 L1) ( ) ( )))
(L36 ((L32 L27 L15 L12 L9 L7 L6 L3 L1) ( ) ( )))
(L33 ((L32 L27 L15 L12 L9 L7 L6 L3 L1) ( ) ( )))
(L34 ((L33 L32 L27 L15 L12 L9 L7 L6 L3 L1) (A5 A4 A3 A2 A1 LD1 LD2 LD3 LD4 LD5 L27 L15 L12 L9 L7 L6 L3 L1)
( ) ( ))
( )
(L31 ((L33 L32 L27 L15 L12 L9 L7 L6 L3 L1) ( ) ( )))
(CONC ((L46 L45 L40 L13 L12 L9 L7 L6 L3 L1) (L1 L3 L6 L7 L9 L12 L13 L40 L45 LD5 LD4 LD3 LD2 LD1 A1 A2 A3 A4 A5)
( ) ( ))
( ))
(plan-steps (L1 0 L2 0 LD1 0) (L2 0) (L3 0 L4 0 LD1 0)
(L4 0 L5 0 LD1 0) (L5 0) (L6 0 L3 0 L1 0) (L7 0 L8 0 LD2 0)
(L8 0 A4 0 L1 0) (L9 0 L10 0 LD2 0) (L10 0 L11 0 LD2 0)
(L11 0 A5 0 L3 0) (L12 0 L9 0 L7 0) (L13 0 L14 0 LD3 0)
(L14 0 L17 0 LD1 0) (L17 0 L16 0 L15 0) (L15 0 L18 0 LD3 0)
(L18 0 L19 0 LD4 0) (L19 0 L22 0 LD1 0) (L22 0 L21 0 L20 0)
(L20 0 L23 0 LD4 0) (L23 0 A1 0 L12 1) (L21 0 L24 0 LD4 0)
(L24 0 L25 0 LD4 0) (L25 0 A2 0 L9 0) (L16 0 L26 0)
(L26 0 L27 0 L28 0) (L29 0 L27 0 LD3 0) (L28 0 L30 0 LD3 0)
(L30 0 L31 0 LD5 0) (L31 0 L34 0 LD1 0) (L34 0 L33 0 L32 0)
(L32 0 L35 0 LD5 0) (L35 0 A1 0 L12 1) (L33 0 L36 0)
(L36 0 L37 0 L38 0) (L39 0 L37 0 LD5 0) (L40 0 L39 0 LD2 0)
(L41 0 L40 0) (L42 0 L40 0) (L38 0 L43 0 LD5 0)
(L44 0 L41 0 L29 0) (L43 0 L44 1 A3 0) (L45 0 L13 0 L6 1)
(L46 0 L45 1 LD2 0) (CONC 0 L46 0) )
```

B.4 The final proof in OMEGA's graphical user interface LOUI

Lovely Omega User Interface@ceb88 (Proof Plan: BOLOS-CURIOUS-INFERENCE-2)

File Presentation Edit View Go Theories Planner Agents Misc: Presentation Examples Extrem Analogy Omega Basic Tactics Verify Miscse Rules Options Help

Map



Label	Hypothesis	Term	Method	Premises
A2		$\forall x. \langle f 1 (s x) \rangle = \langle s (f 1 x) \rangle$	(f 1 HYP	
A3		$\forall n. \langle \forall x. \langle (f (s n) (s x)) \rangle = \langle f HYP$	HYP	
A4		$d 1$	HYP	
A5		$\forall x. \langle (d x) \rangle = \langle d (s x) \rangle$	HYP	
CONC		$\forall l 1 \forall l 2 \forall d (f (s (s (s 1)))) (s (s OTTER$	L46	
LD1		$= \text{def } l d 1 (Az, \langle \forall x. \langle (x 1) \rangle \wedge \langle LOCAL-DEF$		
LD2		$= \text{def } l d 2 (Az, \langle (l d 1) z \rangle \wedge \langle d z LOCAL-DEF$		
L1		$l d 1$	DEFN-CONTRACT-L2	LD1
L2		$\forall dc-220. \langle (dc-220 1) \rangle \wedge \langle \forall dc- OTTER$		
L3		$l d 1 \forall y. \langle (l d 1 y) \rangle = \langle l d 1 (s y) \rangle$	DEFN-CONTRACT-L4	LD1
L4		$\forall dc-233. \langle \langle \forall dc-244. \langle \langle (dc-244$	DEFN-CONTRACT-L5	LD1
L5		$\forall dc-255. \langle \langle \forall dc-266. \langle \langle (dc-266$	OTTER	
L6		$l d 1 (s (s (s 1)))$	OTTER	L1 L3
L7		$l d 2 1$	DEFN-CONTRACT-L8	LD2
L8		$l d 1 \wedge \langle d 1 \rangle$	OTTER	L1 A4
L9		$\forall y. \langle (l d 2 y) \rangle = \langle l d 2 (s y) \rangle$	DEFN-CONTRACT-L10	LD2
L10		$\forall dc-293. \langle \langle (l d 1) dc-293 \rangle \rangle \wedge \langle d$	DEFN-CONTRACT-L11	LD2
L11		$\forall dc-300. \langle \langle (l d 1) dc-300 \rangle \rangle \wedge \langle d$	OTTER	L3 A5
L12		$l d 2 (s 1)$	OTTER	L7 L9
L13		$\forall n. \langle (l d 1 n) \rangle = \langle \forall x. \langle (l d 1 x) \rangle = \text{DEFN-EXPAND-L114}$	LD3	LD1
L14		$\forall n. \langle (l d 1 n) \rangle = \langle l d 1 n \rangle = \text{LOCAL-DEF$	DEFN-CONTRACT-L17	LD1
L15		$l d 3 1$	DEFN-CONTRACT-L18	LD3

Output Message Error Warning Trace

Pretty Term

```

 $\forall n. \langle (l d 1 n) \rangle = \langle \forall x. \langle (l d 1 x) \rangle = \langle l d 2 \langle f n x \rangle \rangle \rangle$ 
 $\forall n. \langle (l d 1 n) \rangle = \langle l d 3 n \rangle$ 
 $= \text{def } l d 3 (\lambda n. \langle \forall x. \langle (l d 1 x) \rangle = \langle l d 2 \langle f n x \rangle \rangle \rangle)$ 
 $\forall dc-300. \langle \langle (l d 1) dc-300 \rangle \rangle \wedge \langle d dc-300 \rangle = \langle (l d 1 (s dc-300)) \rangle \wedge \langle d (s dc-300) \rangle \rangle$ 
 $\forall x. \langle (f 1 (s x)) \rangle = \langle s (f 1 x) \rangle$ 
 $\forall dc-293. \langle \langle (l d 1) dc-293 \rangle \rangle \wedge \langle d dc-293 \rangle = \langle l d 2 (s dc-293) \rangle$ 
 $l d 2 1$ 

```

0 0 0 18 0 0 27 7 0 5 0 Total: 57 Depth: 0 Command: Execute-Log Time: 1.48s



## B.5 Protocol of the interactive session in OMEGA

We present the complete protocol of the interactive session with OMEGA. All proof relevant commands are stored in a proof script (see Appendix B.2). This proof script contains all the information to replay this interactive proof. Note also that interactive proof development is supported by the graphical user interface LOUI (see Appendix B.4).

```

OMEGA: show-pds
OMEGA: prove boolos-curious-inference
Changing to proof plan BOOLOS-CURIOUS-INFERECE-1

;;; step 1
OMEGA: LOCAL-DEF-INTRO

TERM (TERM) a term that should be used as definiens: (LAM (Z I)
  (FORALL
    (LAM
      (X (O I))
      (IMPLIES
        (AND
          (X ONE)
          (FORALL
            (LAM
              (Y I)
              (IMPLIES (X Y) (X (S Y))))))
          (X Z))))))

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP
A2 (A2) ! (FORALL [X:I]
  (=
    (F ONE (S X))
    (S (S (F ONE X))))) HYP
A3 (A3) ! (FORALL [N:I,X:I]
  (=
    (F (S N) (S X))
    (F N (F (S N) X)))) HYP
A4 (A4) ! (D ONE) HYP
A5 (A5) ! (FORALL [X:I]
  (IMPLIES (D X) (D (S X)))) HYP
LD1 (LD1) ! (=DEF LOCAL-DEF
  LD1
  ([Z].
  (FORALL [X:(O I)]
  (IMPLIES
  (AND
  (X ONE)
  (FORALL [Y:I]
  (IMPLIES (X Y) (X (S Y))))
  (X Z))))))
  ...
CONC (A1 A2 A3 ! (D OPEN
  A4 A5) (F
  (S (S (S (S ONE))))
  (S (S (S (S ONE))))))

;;; step 2
OMEGA: LOCAL-DEF-INTRO (LAM (Z I) (AND (LD1 Z) (D Z)))

;;; step 3.1
OMEGA: LEMMA CONC (LD1 ONE)

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP
A2 (A2) ! (FORALL [X:I]
  (=
    (F ONE (S X))
    (S (S (F ONE X))))) HYP
A3 (A3) ! (FORALL [N:I,X:I]
  (=
    (F (S N) (S X))
    (F N (F (S N) X)))) HYP
A4 (A4) ! (D ONE) HYP
A5 (A5) ! (FORALL [X:I]
  (IMPLIES (D X) (D (S X)))) HYP

```

```

LD1 (LD1) ! (=DEF
            LD1
            ([Z].
            (FORALL [X:(0 I)]
            (IMPLIES
            (AND
            (X ONE)
            (FORALL [Y:I]
            (IMPLIES (X Y) (X (S Y))))))
            (X Z)))))) LOCAL-DEF

LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z)))) LOCAL-DEF
...

L1 (A1 A2 A3 ! (LD1 ONE) OPEN
    A4 A5)
...

CONC (A1 A2 A3 ! (D OPEN
    A4 A5) (F
            (S (S (S (S ONE))))
            (S (S (S (S ONE))))))

;; step 3.1
OMEGA: DEFN-CONTRACT-LOCAL-DEF L1 () LD1 (0)
;;CSM Arbitrary [2]: 0 provers have to be killed

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP
A2 (A2) ! (FORALL [X:I] HYP
            (=
            (F ONE (S X))
            (S (S (F ONE X))))))
A3 (A3) ! (FORALL [N:I,X:I] HYP
            (=
            (F (S N) (S X))
            (F N (F (S N) X))))
A4 (A4) ! (D ONE) HYP
A5 (A5) ! (FORALL [X:I] HYP
            (IMPLIES (D X) (D (S X))))
LD1 (LD1) ! (=DEF LOCAL-DEF
            LD1
            ([Z].
            (FORALL [X:(0 I)]
            (IMPLIES
            (AND
            (X ONE)
            (FORALL [Y:I]
            (IMPLIES (X Y) (X (S Y))))))
            (X Z))))))
LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z)))) LOCAL-DEF
...

L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)] OPEN
    A4 A5) (IMPLIES
            (AND
            (DC-13 ONE)
            (FORALL [DC-17:I]
            (IMPLIES
            (DC-13 DC-17)
            (DC-13 (S DC-17))))))
            (DC-13 ONE)))
L1 (A1 A2 A3 ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
    A4 A5)
...

CONC (A1 A2 A3 ! (D OPEN
    A4 A5) (F
            (S (S (S (S ONE))))
            (S (S (S (S ONE))))))

;; step 3.1
OMEGA: SUPPORT L2 ()

...

L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)] OPEN
    A4 A5) (IMPLIES
            (AND
            (DC-13 ONE)
            (FORALL [DC-17:I]
            (IMPLIES

```

The Curious Inference of Boolos in Mizar and OMEGA

```

(DC-13 DC-17)
(DC-13 (S DC-17))))
(DC-13 ONE)))

;; step 3.1
OMEGA: CALL-OTTER-ON-NODE L2 ...
...
----- PROOF -----
...

;; step 3.2
OMEGA: LEMMA CONC (FORALL
                  (LAM (Y I) (IMPLIES (LD1 Y) (LD1 (S Y))))))

;; step 3.2
OMEGA: DEFN-CONTRACT-LOCAL-DEF L3 () LD1 (1 0 1 0)
;;CSM Arbitrary [2]: 0 provers have to be killed

;; step 3.2
OMEGA: DEFN-CONTRACT-LOCAL-DEF L4 () LD1 (1 0 2 0)
;;CSM Arbitrary [2]: 0 provers have to be killed

;; step 3.2
OMEGA: support L5 ()

...

L5 (A1 A2 A3 ! (FORALL [DC-48:I]                                OPEN
  A4 A5) (IMPLIES
          (FORALL [DC-59:(0 I)]
            (IMPLIES
              (AND
                (DC-59 ONE)
                (FORALL [DC-63:I]
                  (IMPLIES
                    (DC-59 DC-63)
                    (DC-59 (S DC-63))))))
              (DC-59 DC-48))))
          (FORALL [DC-68:(0 I)]
            (IMPLIES
              (AND
                (DC-68 ONE)
                (FORALL [DC-72:I]
                  (IMPLIES
                    (DC-68 DC-72)
                    (DC-68 (S DC-72))))))
              (DC-68 (S DC-48))))))

;; step 3.2
OMEGA: CALL-OTTER-ON-NODE L5 ...
...
----- PROOF -----
...

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE)))           HYP
A2 (A2) ! (FORALL [X:I]
          (=
            (F ONE (S X))
            (S (S (F ONE X)))))                           HYP
A3 (A3) ! (FORALL [N:I,X:I]
          (=
            (F (S N) (S X))
            (F N (F (S N) X))))                           HYP
A4 (A4) ! (D ONE)                                         HYP
A5 (A5) ! (FORALL [X:I]
          (IMPLIES (D X) (D (S X))))                       HYP
LD1 (LD1) ! (=DEF
            LD1
            ([Z].
              (FORALL [X:(0 I)]
                (IMPLIES
                  (AND
                    (X ONE)
                    (FORALL [Y:I]
                      (IMPLIES (X Y) (X (S Y))))))
                  (X Z))))))                               LOCAL-DEF
LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z))))           LOCAL-DEF
L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)]                       OTTER: (NIL)
  A4 A5) (IMPLIES
          (AND
            (DC-13 ONE)
            (FORALL [DC-17:I]
              (IMPLIES

```

```

(DC-13 DC-17)
(DC-13 (S DC-17))))
(DC-13 ONE)))

L1 (A1 A2 A3 ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
A4 A5)

L5 (A1 A2 A3 ! (FORALL [DC-48:I] OTTER: (NIL)
A4 A5) (IMPLIES
(FORALL [DC-59:(0 I)]
(IMPLIES
(AND
(DC-59 ONE)
(FORALL [DC-63:I]
(IMPLIES
(DC-59 DC-63)
(DC-59 (S DC-63))))))
(DC-59 DC-48)))
(FORALL [DC-68:(0 I)]
(IMPLIES
(AND
(DC-68 ONE)
(FORALL [DC-72:I]
(IMPLIES
(DC-68 DC-72)
(DC-68 (S DC-72))))))
(DC-68 (S DC-48))))))

L4 (A1 A2 A3 ! (FORALL [DC-26:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
A4 A5) (IMPLIES
(FORALL [DC-37:(0 I)]
(IMPLIES
(AND
(DC-37 ONE)
(FORALL [DC-41:I]
(IMPLIES
(DC-37 DC-41)
(DC-37 (S DC-41))))))
(DC-37 DC-26)))
(LD1 (S DC-26))))

L3 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y))))
...

CONC (A1 A2 A3 ! (D OPEN
A4 A5) (F
(S (S (S ONE))))
(S (S (S ONE))))))

;; step 3.3
OMEGA: LEMMA CONC (LD1 (S (S (S ONE))))

;; step 3.3
OMEGA: SUPPORT L6 (L1 L3)

L3 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y))))

L1 (A1 A2 A3 ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
A4 A5)
...

L6 (A1 A2 A3 ! (LD1 (S (S (S ONE)))) OPEN
A4 A5)

;; step 3.3
OMEGA: CALL-OTTER-ON-NODE L6 ...
...
----- PROOF -----
...

;; step 3.4
OMEGA: LEMMA CONC (LD2 ONE)

;; step 3.4
OMEGA: DEFN-CONTRACT-LOCAL-DEF L7 () LD2 (0)
;;CSM Arbitrary [2]: 0 provers have to be killed

;; step 3.4
OMEGA: SUPPORT L8 (A4 L1)

A4 (A4) ! (D ONE) HYP

L1 (A1 A2 A3 ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
A4 A5)
...

L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE)) OPEN)

```

The Curious Inference of Boolos in Mizar and OMEGA

```

A4 A5)

;; step 3.4
OMEGA: CALL-OTTER-ON-NODE L8 ...
...
----- PROOF -----
...

;; step 3.5
OMEGA: LEMMA CONC (FORALL
                  (LAM (Y I) (IMPLIES (LD2 Y) (LD2 (S Y))))))

;; step 3.5
OMEGA: DEFN-CONTRACT-LOCAL-DEF L9 () LD2 (1 0 1 0)
;;CSM Arbitrary [2]: 0 provers have to be killed

;; step 3.5
OMEGA: DEFN-CONTRACT-LOCAL-DEF L10 () LD2 (1 0 2 0)
;;CSM Arbitrary [2]: 0 provers have to be killed

;; step 3.5
OMEGA: SUPPORT L11 (A5 L3)

A5 (A5)      ! (FORALL [X:I]                                HYP
              (IMPLIES (D X) (D (S X))))

L3 (A1 A2 A3 ! (FORALL [Y:I]                                DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
  A4 A5)      ! (IMPLIES (LD1 Y) (LD1 (S Y))))
  ...

L11 (A1 A2 A3 ! (FORALL [DC-93:I]                            OPEN
  A4 A5)      ! (IMPLIES
                (AND (LD1 DC-93) (D DC-93))
                (AND
                 (LD1 (S DC-93))
                 (D (S DC-93)))))

;; step 3.5
OMEGA: CALL-OTTER-ON-NODE L11 ...
...
----- PROOF -----
...

OMEGA: show-pds
A1 (A1)      ! (FORALL [N:I] (= (F N ONE) (S ONE)))          HYP
A2 (A2)      ! (FORALL [X:I]                                HYP
              (=
               (F ONE (S X))
               (S (S (F ONE X)))))
A3 (A3)      ! (FORALL [N:I,X:I]                            HYP
              (=
               (F (S N) (S X))
               (F N (F (S N) X))))
A4 (A4)      ! (D ONE)                                       HYP
A5 (A5)      ! (FORALL [X:I]                                HYP
              (IMPLIES (D X) (D (S X))))

LD1 (LD1)    ! (=DEF                                         LOCAL-DEF
  LD1
  ([2].
  (FORALL [X:(0 I)]
  (IMPLIES
  (AND
  (X ONE)
  (FORALL [Y:I]
  (IMPLIES (X Y) (X (S Y))))))
  (X Z))))))

LD2 (LD2)    ! (=DEF LD2 ([2].(AND (LD1 Z) (D Z))))          LOCAL-DEF

L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)]                          OTTER: (NIL)
  A4 A5)      ! (IMPLIES
                (AND
                 (DC-13 ONE)
                 (FORALL [DC-17:I]
                  (IMPLIES
                   (DC-13 DC-17)
                   (DC-13 (S DC-17))))))
                (DC-13 ONE)))

L1 (A1 A2 A3 ! (LD1 ONE)                                       DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
  A4 A5)

L5 (A1 A2 A3 ! (FORALL [DC-48:I]                              OTTER: (NIL)
  A4 A5)      ! (IMPLIES
                (FORALL [DC-59:(0 I)]
                 (IMPLIES
                  (DC-48 (S DC-59))
                  (DC-59))))))

```

```

(IMPLIES
  (AND
    (DC-59 ONE)
    (FORALL [DC-63:I]
      (IMPLIES
        (DC-59 DC-63)
        (DC-59 (S DC-63))))))
(FORALL [DC-68:(0 I)]
  (IMPLIES
    (AND
      (DC-68 ONE)
      (FORALL [DC-72:I]
        (IMPLIES
          (DC-68 DC-72)
          (DC-68 (S DC-72))))))
    (DC-68 (S DC-48))))))

L4 (A1 A2 A3 ! (FORALL [DC-26:I]
  A4 A5)      (IMPLIES
              (FORALL [DC-37:(0 I)]
                (IMPLIES
                  (AND
                    (DC-37 ONE)
                    (FORALL [DC-41:I]
                      (IMPLIES
                        (DC-37 DC-41)
                        (DC-37 (S DC-41))))))
                  (DC-37 DC-26)))
              (LD1 (S DC-26))))))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)

L3 (A1 A2 A3 ! (FORALL [Y:I]
  A4 A5)      (IMPLIES (LD1 Y) (LD1 (S Y))))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)

L6 (A1 A2 A3 ! (LD1 (S (S (S ONE))))
  A4 A5)
OTTER: (NIL) (L1 L3)

L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE))
  A4 A5)
OTTER: (NIL) (L1 A4)

L7 (A1 A2 A3 ! (LD2 ONE)
  A4 A5)
DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)

L11 (A1 A2 A3 ! (FORALL [DC-93:I]
  A4 A5)      (IMPLIES
              (AND (LD1 DC-93) (D DC-93))
              (AND
                (LD1 (S DC-93))
                (D (S DC-93))))))
OTTER: (NIL) (L3 A5)

L10 (A1 A2 A3 ! (FORALL [DC-86:I]
  A4 A5)      (IMPLIES
              (AND (LD1 DC-86) (D DC-86))
              (LD2 (S DC-86))))))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)

L9 (A1 A2 A3 ! (FORALL [Y:I]
  A4 A5)      (IMPLIES (LD2 Y) (LD2 (S Y))))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
...

CONC (A1 A2 A3 ! (D
  A4 A5)      (F
              (S (S (S ONE))))
              (S (S (S ONE))))))
OPEN

;; step 3.6
OMEGA: LEMMA CONC (LD2 (S ONE))

;; step 3.6
OMEGA: SUPPORT L12 (L7 L9)

L9 (A1 A2 A3 ! (FORALL [Y:I]
  A4 A5)      (IMPLIES (LD2 Y) (LD2 (S Y))))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)

L7 (A1 A2 A3 ! (LD2 ONE)
  A4 A5)
DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
...

L12 (A1 A2 A3 ! (LD2 (S ONE))
  A4 A5)
OPEN

;; step 3.6
OMEGA: CALL-OTTER-ON-NODE L12 ...
...
----- PROOF -----
...

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP
A2 (A2) ! (FORALL [X:I] HYP

```

The Curious Inference of Boolos in Mizar and OMEGA

```

(=
(F ONE (S X))
(S (S (F ONE X))))
A3 (A3) ! (FORALL [N:I,X:I] HYP
(=
(F (S N) (S X))
(F N (F (S N) X))))
A4 (A4) ! (D ONE) HYP
A5 (A5) ! (FORALL [X:I] HYP
(IMPLIES (D X) (D (S X))))
LD1 (LD1) ! (=DEF LOCAL-DEF
LD1
([Z.
(FORALL [X:(O I)]
(IMPLIES
(AND
(X ONE)
(FORALL [Y:I]
(IMPLIES (X Y) (X (S Y))))))
(X Z))))))
LD2 (LD2) ! (=DEF LD2 ([Z.(AND (LD1 Z) (D Z)))) LOCAL-DEF
L2 (A1 A2 A3 ! (FORALL [DC-13:(O I)] OTTER: (NIL)
A4 A5) (IMPLIES
(AND
(DC-13 ONE)
(FORALL [DC-17:I]
(IMPLIES
(DC-13 DC-17)
(DC-13 (S DC-17))))))
(DC-13 ONE)))
L1 (A1 A2 A3 ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
A4 A5)
L5 (A1 A2 A3 ! (FORALL [DC-48:I] OTTER: (NIL)
A4 A5) (IMPLIES
(FORALL [DC-59:(O I)]
(IMPLIES
(AND
(DC-59 ONE)
(FORALL [DC-63:I]
(IMPLIES
(DC-59 DC-63)
(DC-59 (S DC-63))))))
(DC-59 DC-48)))
(FORALL [DC-68:(O I)]
(IMPLIES
(AND
(DC-68 ONE)
(FORALL [DC-72:I]
(IMPLIES
(DC-68 DC-72)
(DC-68 (S DC-72))))))
(DC-68 (S DC-48))))))
L4 (A1 A2 A3 ! (FORALL [DC-26:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
A4 A5) (IMPLIES
(FORALL [DC-37:(O I)]
(IMPLIES
(AND
(DC-37 ONE)
(FORALL [DC-41:I]
(IMPLIES
(DC-37 DC-41)
(DC-37 (S DC-41))))))
(DC-37 DC-26)))
(LD1 (S DC-26))))
L3 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y))))
L6 (A1 A2 A3 ! (LD1 (S (S (S ONE)))) OTTER: (NIL) (L1 L3)
A4 A5)
L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE)) OTTER: (NIL) (L1 A4)
A4 A5)
L7 (A1 A2 A3 ! (LD2 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
A4 A5)
L11 (A1 A2 A3 ! (FORALL [DC-93:I] OTTER: (NIL) (L3 A5)
A4 A5) (IMPLIES
(AND (LD1 DC-93) (D DC-93))
(AND
(LD1 (S DC-93))

```

```

(D (S DC-93))))

L10 (A1 A2 A3 ! (FORALL [DC-86:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
A4 A5) (IMPLIES
      (AND (LD1 DC-86) (D DC-86))
      (LD2 (S DC-86))))

L9 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y))))

L12 (A1 A2 A3 ! (LD2 (S ONE)) OTTER: (NIL) (L7 L9)
A4 A5)
...

CONC (A1 A2 A3 ! (D OPEN
A4 A5) (F
      (S (S (S ONE))))
      (S (S (S ONE))))))

;; step 4
OMEGA: LEMMA CONC (FORALL
                  (LAM (N I)
                     (IMPLIES (LD1 N)
                                (FORALL
                                 (LAM (X I)
                                    (IMPLIES (LD1 X) (LD2 (F N X))))))))

;; step 4.1
OMEGA: LOCAL-DEF-INTRO (LAM (N I)
                        (FORALL
                         (LAM
                          (X I)
                           (IMPLIES (LD1 X) (LD2 (F N X))))))

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP
A2 (A2) ! (FORALL [X:I] HYP
          (=
           (F ONE (S X))
           (S (S (F ONE X)))))
A3 (A3) ! (FORALL [N:I,X:I] HYP
          (=
           (F (S N) (S X))
           (F N (F (S N) X))))
A4 (A4) ! (D ONE) HYP
A5 (A5) ! (FORALL [X:I] HYP
          (IMPLIES (D X) (D (S X))))

LD1 (LD1) ! (=DEF LOCAL-DEF
            LD1
            ([Z]
             (FORALL [X:(0 I)]
                    (IMPLIES
                     (AND
                      (X ONE)
                      (FORALL [Y:I]
                               (IMPLIES (X Y) (X (S Y))))))
                     (X Z))))))
LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z))) LOCAL-DEF

L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)] OTTER: (NIL)
A4 A5) (IMPLIES
      (AND
       (DC-13 ONE)
       (FORALL [DC-17:I]
                (IMPLIES
                 (DC-13 DC-17)
                 (DC-13 (S DC-17))))))
      (DC-13 ONE)))

L1 (A1 A2 A3 ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
A4 A5)

L5 (A1 A2 A3 ! (FORALL [DC-48:I] OTTER: (NIL)
A4 A5) (IMPLIES
      (FORALL [DC-59:(0 I)]
              (IMPLIES
               (AND
                (DC-59 ONE)
                (FORALL [DC-63:I]
                         (IMPLIES
                          (DC-59 DC-63)
                          (DC-59 (S DC-63))))))
               (DC-59 DC-48)))
      (FORALL [DC-68:(0 I)]
              (IMPLIES
               (AND
                (DC-68 ONE)
                (FORALL [DC-63:I]
                         (IMPLIES
                          (DC-68 DC-63)
                          (DC-68 (S DC-63))))))
               (DC-68 DC-48))))))

```



The Curious Inference of Boolos in Mizar and OMEGA

```

(DC-68 ONE)
(FORALL [DC-72:I]
  (IMPLIES
    (DC-68 DC-72)
    (DC-68 (S DC-72))))
(DC-68 (S DC-48))))))

L4 (A1 A2 A3 ! (FORALL [DC-26:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
  A4 A5) (IMPLIES
  (FORALL [DC-37:(0 I)]
    (IMPLIES
      (AND
        (DC-37 ONE)
        (FORALL [DC-41:I]
          (IMPLIES
            (DC-37 DC-41)
            (DC-37 (S DC-41))))))
        (DC-37 DC-26)))
    (LD1 (S DC-26))))))

L3 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
  A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y))))

L6 (A1 A2 A3 ! (LD1 (S (S (S ONE)))) OTTER: (NIL) (L1 L3)
  A4 A5)

L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE)) OTTER: (NIL) (L1 A4)
  A4 A5)

L7 (A1 A2 A3 ! (LD2 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
  A4 A5)

L11 (A1 A2 A3 ! (FORALL [DC-93:I] OTTER: (NIL) (L3 A5)
  A4 A5) (IMPLIES
  (AND (LD1 DC-93) (D DC-93))
  (AND
    (LD1 (S DC-93))
    (D (S DC-93))))))

L10 (A1 A2 A3 ! (FORALL [DC-86:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
  A4 A5) (IMPLIES
  (AND (LD1 DC-86) (D DC-86))
  (LD2 (S DC-86))))))

L9 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
  A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y))))

L12 (A1 A2 A3 ! (LD2 (S ONE)) OTTER: (NIL) (L7 L9)
  A4 A5)
...

L13 (A1 A2 A3 ! (FORALL [N:I] OPEN
  A4 A5) (IMPLIES
  (LD1 N)
  (FORALL [X:I]
    (IMPLIES
      (LD1 X)
      (LD2 (F N X))))))

LD3 (LD3) ! (=DEF LOCAL-DEF
  LD3
  ([N]
  (FORALL [X:I]
    (IMPLIES
      (LD1 X)
      (LD2 (F N X))))))
...

CONC (A1 A2 A3 ! (D OPEN
  A4 A5) (F
  (S (S (S ONE))))
  (S (S (S ONE))))))

;; step 4.2
OMEGA: LEMMA L13 (FORALL (LAM (N I) (IMPLIES (LD1 N) (LD3 N))))

;; step 4.2
OMEGA: DEFN-EXPAND-LOCAL-DEF L13 L14 LD3 (1 0 2 0)
;;CSM Arbitrary [2]: 0 provers have to be killed

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP
A2 (A2) ! (FORALL [X:I] HYP
  (=
    (F ONE (S X))
    (S (S (F ONE X))))))
A3 (A3) ! (FORALL [N:I,X:I] HYP
  (=
    (F (S N) (S X))
    (S (S (F ONE X))))))

```

```

(F N (F (S N) X)))

A4 (A4) ! (D ONE) HYP
A5 (A5) ! (FORALL [X:I] HYP
(IMPLIES (D X) (D (S X))))

LD1 (LD1) ! (=DEF LOCAL-DEF
LD1
([Z]
(FORALL [X:(0 I)]
(IMPLIES
(AND
(X ONE)
(FORALL [Y:I]
(IMPLIES (X Y) (X (S Y))))))
(X Z))))))

LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z)))) LOCAL-DEF

L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)] OTTER: (NIL)
A4 A5) (IMPLIES
(AND
(DC-13 ONE)
(FORALL [DC-17:I]
(IMPLIES
(DC-13 DC-17)
(DC-13 (S DC-17))))))
(DC-13 ONE)))

L1 (A1 A2 A3 ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
A4 A5)

L5 (A1 A2 A3 ! (FORALL [DC-48:I] OTTER: (NIL)
A4 A5) (IMPLIES
(FORALL [DC-59:(0 I)]
(IMPLIES
(AND
(DC-59 ONE)
(FORALL [DC-63:I]
(IMPLIES
(DC-59 DC-63)
(DC-59 (S DC-63))))))
(DC-59 DC-48)))
(FORALL [DC-68:(0 I)]
(IMPLIES
(AND
(DC-68 ONE)
(FORALL [DC-72:I]
(IMPLIES
(DC-68 DC-72)
(DC-68 (S DC-72))))))
(DC-68 (S DC-48))))))

L4 (A1 A2 A3 ! (FORALL [DC-26:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
A4 A5) (IMPLIES
(FORALL [DC-37:(0 I)]
(IMPLIES
(AND
(DC-37 ONE)
(FORALL [DC-41:I]
(IMPLIES
(DC-37 DC-41)
(DC-37 (S DC-41))))))
(DC-37 DC-26)))
(LD1 (S DC-26))))

L3 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y))))

L6 (A1 A2 A3 ! (LD1 (S (S (S ONE)))) OTTER: (NIL) (L1 L3)
A4 A5)

L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE)) OTTER: (NIL) (L1 A4)
A4 A5)

L7 (A1 A2 A3 ! (LD2 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
A4 A5)

L11 (A1 A2 A3 ! (FORALL [DC-93:I] OTTER: (NIL) (L3 A5)
A4 A5) (IMPLIES
(AND (LD1 DC-93) (D DC-93))
(AND
(LD1 (S DC-93))
(D (S DC-93))))))

L10 (A1 A2 A3 ! (FORALL [DC-86:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
A4 A5) (IMPLIES
(AND (LD1 DC-86) (D DC-86))
(LD2 (S DC-86))))

```

The Curious Inference of Boolos in Mizar and OMEGA

```

L9 (A1 A2 A3 ! (FORALL [Y:I]
A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)

L12 (A1 A2 A3 ! (LD2 (S ONE)) OTTER: (NIL) (L7 L9)
A4 A5)

L13 (A1 A2 A3 ! (FORALL [N:I]
A4 A5) (IMPLIES
(LD1 N)
(FORALL [X:I]
(IMPLIES
(LD1 X)
(LD2 (F N X)))))) DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)

LD3 (LD3) ! (=DEF LOCAL-DEF
LD3
([N].
(FORALL [X:I]
(IMPLIES
(LD1 X)
(LD2 (F N X))))))
...

L14 (A1 A2 A3 ! (FORALL [N:I] OPEN
A4 A5) (IMPLIES (LD1 N) (LD3 N)))
...

CONC (A1 A2 A3 ! (D OPEN
A4 A5) (F
(S (S (S (S ONE))))
(S (S (S (S ONE))))))

;; step 4.3.1
OMEGA: LEMMA L14 (LD3 ONE)

;; step 4.3.2
OMEGA: LEMMA L14 (FORALL
(LAM (N I) (IMPLIES (LD3 N) (LD3 (S N)))))

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP
A2 (A2) ! (FORALL [X:I] HYP
(=
(F ONE (S X))
(S (S (F ONE X)))))
A3 (A3) ! (FORALL [N:I,X:I] HYP
(=
(F (S N) (S X))
(F N (F (S N) X))))
A4 (A4) ! (D ONE) HYP
A5 (A5) ! (FORALL [X:I] HYP
(IMPLIES (D X) (D (S X))))

LD1 (LD1) ! (=DEF LOCAL-DEF
LD1
([Z].
(FORALL [X:(0 I)]
(IMPLIES
(AND
(X ONE)
(FORALL [Y:I]
(IMPLIES (X Y) (X (S Y))))))
(X Z))))
LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z))) LOCAL-DEF

L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)] OTTER: (NIL)
A4 A5) (IMPLIES
(AND
(DC-13 ONE)
(FORALL [DC-17:I]
(IMPLIES
(DC-13 DC-17)
(DC-13 (S DC-17))))))
(DC-13 ONE)))

L1 (A1 A2 A3 ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
A4 A5)

L5 (A1 A2 A3 ! (FORALL [DC-48:I] OTTER: (NIL)
A4 A5) (IMPLIES
(FORALL [DC-59:(0 I)]
(IMPLIES
(AND
(DC-59 ONE)
(FORALL [DC-63:I]
(IMPLIES

```

```

(DC-59 DC-63)
(DC-59 (S DC-63))))))
(DC-59 DC-48)))
(FORALL [DC-68:(0 I)]
(IMPLIES
(AND
(DC-68 ONE)
(FORALL [DC-72:I]
(IMPLIES
(DC-68 DC-72)
(DC-68 (S DC-72))))))
(DC-68 (S DC-48))))))
L4 (A1 A2 A3 ! (FORALL [DC-26:I]
A4 A5) (IMPLIES
(FORALL [DC-37:(0 I)]
(IMPLIES
(AND
(DC-37 ONE)
(FORALL [DC-41:I]
(IMPLIES
(DC-37 DC-41)
(DC-37 (S DC-41))))))
(DC-37 DC-26)))
(LD1 (S DC-26))))))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
L3 (A1 A2 A3 ! (FORALL [Y:I]
A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y))))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
L6 (A1 A2 A3 ! (LD1 (S (S (S ONE))))
A4 A5) OTTER: (NIL) (L1 L3)
L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE))
A4 A5) OTTER: (NIL) (L1 A4)
L7 (A1 A2 A3 ! (LD2 ONE)
A4 A5) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
L11 (A1 A2 A3 ! (FORALL [DC-93:I]
A4 A5) (IMPLIES
(AND (LD1 DC-93) (D DC-93))
(AND
(LD1 (S DC-93))
(D (S DC-93))))))
OTTER: (NIL) (L3 A5)
L10 (A1 A2 A3 ! (FORALL [DC-86:I]
A4 A5) (IMPLIES
(AND (LD1 DC-86) (D DC-86))
(LD2 (S DC-86))))))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
L9 (A1 A2 A3 ! (FORALL [Y:I]
A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y))))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
L12 (A1 A2 A3 ! (LD2 (S ONE))
A4 A5) OTTER: (NIL) (L7 L9)
L13 (A1 A2 A3 ! (FORALL [N:I]
A4 A5) (IMPLIES
(LD1 N)
(FORALL [X:I]
(IMPLIES
(LD1 X)
(LD2 (F N X))))))
DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)
LDS (LD3) ! (*DEF
LDS
([N]
(FORALL [X:I]
(IMPLIES
(LD1 X)
(LD2 (F N X))))))
...
LOCAL-DEF
L15 (A1 A2 A3 ! (LDS ONE)
A4 A5) OPEN
...
L16 (A1 A2 A3 ! (FORALL [N:I]
A4 A5) (IMPLIES (LDS N) (LDS (S N))))
...
OPEN
L14 (A1 A2 A3 ! (FORALL [N:I]
A4 A5) (IMPLIES (LD1 N) (LDS N)))
...
OPEN
CONC (A1 A2 A3 ! (D
A4 A5) (F
(S (S (S ONE))))
(S (S (S ONE))))))
OPEN

```

OMEGA: show-subproblem

The Curious Inference of Boolos in Mizar and OMEGA

```

PLAN (NDPLANLINE) Plan line to show: 114

L1 (A1 A2 A3 ! (LD1 ONE)                                DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
  A4 A5)

L3 (A1 A2 A3 ! (FORALL [Y:I]                             DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
  A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y))))

L6 (A1 A2 A3 ! (LD1 (S (S (S ONE))))                    OTTER: (NIL) (L1 L3)
  A4 A5)

L7 (A1 A2 A3 ! (LD2 ONE)                                DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
  A4 A5)

L9 (A1 A2 A3 ! (FORALL [Y:I]                             DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
  A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y))))

L12 (A1 A2 A3 ! (LD2 (S ONE))                            OTTER: (NIL) (L7 L9)
  A4 A5)
...

L15 (A1 A2 A3 ! (LD3 ONE)                                OPEN
  A4 A5)
...

L16 (A1 A2 A3 ! (FORALL [N:I]                             OPEN
  A4 A5) (IMPLIES (LD3 N) (LD3 (S N))))

LD3 (LD3) ! (=DEF                                        LOCAL-DEF
  LD3
  ([N].
  (FORALL [X:I]
  (IMPLIES
  (LD1 X)
  (LD2 (F N X))))))

LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z))))        LOCAL-DEF

LD1 (LD1) ! (=DEF                                        LOCAL-DEF
  LD1
  ([Z].
  (FORALL [X:(0 1)]
  (IMPLIES
  (AND
  (X ONE)
  (FORALL [Y:I]
  (IMPLIES (X Y) (X (S Y))))))
  (X Z))))))

A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE)))          HYP

A2 (A2) ! (FORALL [X:I]
  (=
  (F ONE (S X))
  (S (S (F ONE X)))))                                HYP

A3 (A3) ! (FORALL [N:I,X:I]
  (=
  (F (S N) (S X))
  (F N (F (S N) X))))                                HYP

A4 (A4) ! (D ONE)                                       HYP

A5 (A5) ! (FORALL [X:I]
  (IMPLIES (D X) (D (S X))))                          HYP
...

L14 (A1 A2 A3 ! (FORALL [N:I]                             OPEN
  A4 A5) (IMPLIES (LD1 N) (LD3 N)))

;; step 4.3 -- Enough to show ...
OMEGA: DEFN-CONTRACT-LOCAL-DEF L14 () LD1 (1 0 1 0)
;;CSM Arbitrary [2]: 0 provers have to be killed

;; step 4.3
OMEGA: SUPPORT L17 (L15 L16)

...

L16 (A1 A2 A3 ! (FORALL [N:I]                             OPEN
  A4 A5) (IMPLIES (LD3 N) (LD3 (S N))))
...

L15 (A1 A2 A3 ! (LD3 ONE)                                OPEN
  A4 A5)
...

L17 (A1 A2 A3 ! (FORALL [DC-123:I]                       OPEN
  A4 A5) (IMPLIES

```

```

(FORALL [DC-134:(0 I)]
(IMPLIES
(AND
(DC-134 ONE)
(FORALL [DC-138:I]
(IMPLIES
(DC-134 DC-138)
(DC-134 (S DC-138))))))
(DC-134 DC-123)))
(LD3 DC-123)))

;;; step 4.3 -- Enough to show ...
OMEGA: CALL-UTTER-ON-NODE L17 ...
...
----- PROOF -----
...

;;; step 4.3.1.1
OMEGA: DEFN-CONTRACT-LOCAL-DEF L15 () LD3 (0)
;;;CSM Arbitrary [2]: 0 provers have to be killed

;;; step 4.3.1.2
OMEGA: LOCAL-DEF-INTRO (LAM (X I) (LD2 (F ONE X)))

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP
A2 (A2) ! (FORALL [X:I] HYP
(=
(F ONE (S X))
(S (S (F ONE X)))))
A3 (A3) ! (FORALL [N:I,X:I] HYP
(=
(F (S N) (S X))
(F N (F (S N) X))))
A4 (A4) ! (D ONE) HYP
A5 (A5) ! (FORALL [X:I] HYP
(IMPLIES (D X) (D (S X))))
LD1 (LD1) ! (=DEF LOCAL-DEF
LD1
([Z].
(FORALL [X:(0 I)]
(IMPLIES
(AND
(X ONE)
(FORALL [Y:I]
(IMPLIES (X Y) (X (S Y))))))
(X Z))))))
LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z)))) LOCAL-DEF
L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)] OTTER: (NIL)
A4 A5) (IMPLIES
(AND
(DC-13 ONE)
(FORALL [DC-17:I]
(IMPLIES
(DC-13 DC-17)
(DC-13 (S DC-17))))))
(DC-13 ONE)))
L1 (A1 A2 A3 ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
A4 A5)
L5 (A1 A2 A3 ! (FORALL [DC-48:I] OTTER: (NIL)
A4 A5) (IMPLIES
(FORALL [DC-59:(0 I)]
(IMPLIES
(AND
(DC-59 ONE)
(FORALL [DC-63:I]
(IMPLIES
(DC-59 DC-63)
(DC-59 (S DC-63))))))
(DC-59 DC-48)))
(FORALL [DC-68:(0 I)]
(IMPLIES
(AND
(DC-68 ONE)
(FORALL [DC-72:I]
(IMPLIES
(DC-68 DC-72)
(DC-68 (S DC-72))))))
(DC-68 (S DC-48))))))
L4 (A1 A2 A3 ! (FORALL [DC-26:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
A4 A5) (IMPLIES

```

The Curious Inference of Boolos in Mizar and OMEGA

```

(FORALL [DC-37:(0 I)]
(IMPLIES
(AND
(DC-37 ONE)
(FORALL [DC-41:I]
(IMPLIES
(DC-37 DC-41)
(DC-37 (S DC-41))))))
(DC-37 DC-26)))
(LD1 (S DC-26))))

L3 (A1 A2 A3 ! (FORALL [Y:I]
A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)

L6 (A1 A2 A3 ! (LD1 (S (S (S (S ONE))))))
A4 A5) OTTER: (NIL) (L1 L3)

L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE)))
A4 A5) OTTER: (NIL) (L1 A4)

L7 (A1 A2 A3 ! (LD2 ONE))
A4 A5) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)

L11 (A1 A2 A3 ! (FORALL [DC-93:I]
A4 A5) (IMPLIES
(AND (LD1 DC-93) (D DC-93))
(AND
(LD1 (S DC-93))
(D (S DC-93))))) OTTER: (NIL) (L3 A5)

L10 (A1 A2 A3 ! (FORALL [DC-86:I]
A4 A5) (IMPLIES
(AND (LD1 DC-86) (D DC-86))
(LD2 (S DC-86)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)

L9 (A1 A2 A3 ! (FORALL [Y:I]
A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)

L12 (A1 A2 A3 ! (LD2 (S ONE)))
A4 A5) OTTER: (NIL) (L7 L9)

L13 (A1 A2 A3 ! (FORALL [N:I]
A4 A5) (IMPLIES
(LD1 N)
(FORALL [X:I]
(IMPLIES
(LD1 X)
(LD2 (F N X)))))) DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)

LD3 (LD3) ! (=DEF
LOCAL-DEF
LD3
([N]
(FORALL [X:I]
(IMPLIES
(LD1 X)
(LD2 (F N X))))))
...

L18 (A1 A2 A3 ! (FORALL [DC-151:I]
A4 A5) (IMPLIES
(LD1 DC-151)
(LD2 (F ONE DC-151)))) OPEN

L15 (A1 A2 A3 ! (LD3 ONE))
A4 A5) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L18 LD3)
...

L16 (A1 A2 A3 ! (FORALL [N:I]
A4 A5) (IMPLIES (LD3 N) (LD3 (S N)))) OPEN

L17 (A1 A2 A3 ! (FORALL [DC-123:I]
A4 A5) (IMPLIES
(FORALL [DC-134:(0 I)]
(IMPLIES
(AND
(DC-134 ONE)
(FORALL [DC-138:I]
(IMPLIES
(DC-134 DC-138)
(DC-134 (S DC-138))))))
(DC-134 DC-123)))
(LD3 DC-123))) OTTER: (NIL) (L15 L16)

L14 (A1 A2 A3 ! (FORALL [N:I]
A4 A5) (IMPLIES (LD1 N) (LD3 N))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L17 LD1)

LD4 (LD4) ! (=DEF LD4 (([X].(LD2 (F ONE X))))
LOCAL-DEF
...

CONC (A1 A2 A3 ! (D
A4 A5) (F) OPEN

```

```

(S (S (S (S ONE))))
(S (S (S (S ONE))))))

;;; step 4.3.1.3
OMEGA: LEMMA L18 (FORALL (LAM (X I) (IMPLIES (LD1 X) (LD4 X))))

;;; step 4.3.1.3
OMEGA: DEFN-EXPAND-LOCAL-DEF L18 L19 LD4 (1 0 2 0)
;;;CSM Arbitrary [2]: 0 provers have to be killed

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP
A2 (A2) ! (FORALL [X:I]
  (=
    (F ONE (S X))
    (S (S (F ONE X))))) HYP
A3 (A3) ! (FORALL [N:I,X:I]
  (=
    (F (S N) (S X))
    (F N (F (S N) X)))) HYP
A4 (A4) ! (D ONE) HYP
A5 (A5) ! (FORALL [X:I]
  (IMPLIES (D X) (D (S X)))) HYP
LD1 (LD1) ! (=DEF
  LD1
  ([Z]
    (FORALL [X:(0 I)]
      (IMPLIES
        (AND
          (X ONE)
          (FORALL [Y:I]
            (IMPLIES (X Y) (X (S Y))))))
        (X Z)))) LOCAL-DEF
LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z)))) LOCAL-DEF
L2 (A1 A2 A3
  A4 A5) ! (FORALL [DC-13:(0 I)]
  (IMPLIES
    (AND
      (DC-13 ONE)
      (FORALL [DC-17:I]
        (IMPLIES
          (DC-13 DC-17)
          (DC-13 (S DC-17))))))
    (DC-13 ONE))) OTTER: (NIL)
L1 (A1 A2 A3
  A4 A5) ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
L5 (A1 A2 A3
  A4 A5) ! (FORALL [DC-48:I]
  (IMPLIES
    (FORALL [DC-59:(0 I)]
      (IMPLIES
        (AND
          (DC-59 ONE)
          (FORALL [DC-63:I]
            (IMPLIES
              (DC-59 DC-63)
              (DC-59 (S DC-63))))))
        (DC-59 DC-48)))
    (FORALL [DC-68:(0 I)]
      (IMPLIES
        (AND
          (DC-68 ONE)
          (FORALL [DC-72:I]
            (IMPLIES
              (DC-68 DC-72)
              (DC-68 (S DC-72))))))
        (DC-68 (S DC-48)))))) OTTER: (NIL)
L4 (A1 A2 A3
  A4 A5) ! (FORALL [DC-26:I]
  (IMPLIES
    (FORALL [DC-37:(0 I)]
      (IMPLIES
        (AND
          (DC-37 ONE)
          (FORALL [DC-41:I]
            (IMPLIES
              (DC-37 DC-41)
              (DC-37 (S DC-41))))))
        (DC-37 DC-26)))
    (LD1 (S DC-26)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
L3 (A1 A2 A3
  A4 A5) ! (FORALL [Y:I]
  (IMPLIES (LD1 Y) (LD1 (S Y)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)

```



The Curious Inference of Boolos in Mizar and OMEGA

```

L6 (A1 A2 A3 ! (LD1 (S (S (S (S ONE)))))) OTTER: (NIL) (L1 L3)
A4 A5)

L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE))) OTTER: (NIL) (L1 A4)
A4 A5)

L7 (A1 A2 A3 ! (LD2 ONE)) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
A4 A5)

L11 (A1 A2 A3 ! (FORALL [DC-93:I] OTTER: (NIL) (L3 A5)
A4 A5) (IMPLIES
      (AND (LD1 DC-93) (D DC-93))
      (AND
        (LD1 (S DC-93))
        (D (S DC-93)))))

L10 (A1 A2 A3 ! (FORALL [DC-86:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
A4 A5) (IMPLIES
      (AND (LD1 DC-86) (D DC-86))
      (LD2 (S DC-86))))

L9 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y))))

L12 (A1 A2 A3 ! (LD2 (S ONE))) OTTER: (NIL) (L7 L9)
A4 A5)

L13 (A1 A2 A3 ! (FORALL [N:I] DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)
A4 A5) (IMPLIES
      (LD1 N)
      (FORALL [X:I]
        (IMPLIES
          (LD1 X)
          (LD2 (F N X)))))

LD3 (LD3) ! (=DEF LOCAL-DEF
LD3
([N].
(FORALL [X:I]
(IMPLIES
(LD1 X)
(LD2 (F N X)))))

L18 (A1 A2 A3 ! (FORALL [DC-151:I] DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L19 LD4)
A4 A5) (IMPLIES
      (LD1 DC-151)
      (LD2 (F ONE DC-151))))

L15 (A1 A2 A3 ! (LD3 ONE)) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L18 LD3)
A4 A5)
...

L16 (A1 A2 A3 ! (FORALL [N:I] OPEN
A4 A5) (IMPLIES (LD3 N) (LD3 (S N))))

L17 (A1 A2 A3 ! (FORALL [DC-123:I] OTTER: (NIL) (L15 L16)
A4 A5) (IMPLIES
      (FORALL [DC-134:(0 I)]
        (IMPLIES
          (AND
            (DC-134 ONE)
            (FORALL [DC-138:I]
              (IMPLIES
                (DC-134 DC-138)
                (DC-134 (S DC-138)))))
            (DC-134 DC-123)))
        (LD3 DC-123)))

L14 (A1 A2 A3 ! (FORALL [N:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L17 LD1)
A4 A5) (IMPLIES (LD1 N) (LD3 N)))

LD4 (LD4) ! (=DEF LD4 ([X].(LD2 (F ONE X)))) LOCAL-DEF
...

L19 (A1 A2 A3 ! (FORALL [X:I] OPEN
A4 A5) (IMPLIES (LD1 X) (LD4 X)))
...

CONC (A1 A2 A3 ! (D OPEN
A4 A5) (F
      (S (S (S (S ONE))))
      (S (S (S (S ONE)))))

;; step 4.3.1.4.1
OMEGA: LEMMA L19 (LD4 ONE)

;; step 4.3.1.4.2
OMEGA: LEMMA L19 (FORALL
      (LAM (X I) (IMPLIES (LD4 X) (LD4 (S X)))))

```

```
;; step 4.3.1.4 -- Enough to show
OMEGA: DEFN-CONTRACT-LOCAL-DEF L19 () LD1 (1 0 1 0)
;;CSM Arbitrary [2]: 0 provers have to be killed
```

```
OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP
A2 (A2) ! (FORALL [X:I]
  (=
    (F ONE (S X))
    (S (S (F ONE X))))) HYP
A3 (A3) ! (FORALL [N:I,X:I]
  (=
    (F (S N) (S X))
    (F N (F (S N) X)))) HYP
A4 (A4) ! (D ONE) HYP
A5 (A5) ! (FORALL [X:I]
  (IMPLIES (D X) (D (S X)))) HYP
LD1 (LD1) ! (=DEF
  LD1
  ([Z]
    (FORALL [X:(0 I)]
      (IMPLIES
        (AND
          (X ONE)
          (FORALL [Y:I]
            (IMPLIES (X Y) (X (S Y))))))
        (X Z))))) LOCAL-DEF
LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z)))) LOCAL-DEF
L2 (A1 A2 A3 A4 A5) ! (FORALL [DC-13:(0 I)]
  (IMPLIES
    (AND
      (DC-13 ONE)
      (FORALL [DC-17:I]
        (IMPLIES
          (DC-13 DC-17)
          (DC-13 (S DC-17))))))
    (DC-13 ONE))) OTTER: (NIL)
L1 (A1 A2 A3 A4 A5) ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
L5 (A1 A2 A3 A4 A5) ! (FORALL [DC-48:I]
  (IMPLIES
    (FORALL [DC-59:(0 I)]
      (IMPLIES
        (AND
          (DC-59 ONE)
          (FORALL [DC-63:I]
            (IMPLIES
              (DC-59 DC-63)
              (DC-59 (S DC-63))))))
        (DC-59 DC-48)))
    (FORALL [DC-68:(0 I)]
      (IMPLIES
        (AND
          (DC-68 ONE)
          (FORALL [DC-72:I]
            (IMPLIES
              (DC-68 DC-72)
              (DC-68 (S DC-72))))))
        (DC-68 (S DC-48)))))) OTTER: (NIL)
L4 (A1 A2 A3 A4 A5) ! (FORALL [DC-26:I]
  (IMPLIES
    (FORALL [DC-37:(0 I)]
      (IMPLIES
        (AND
          (DC-37 ONE)
          (FORALL [DC-41:I]
            (IMPLIES
              (DC-37 DC-41)
              (DC-37 (S DC-41))))))
        (DC-37 DC-26)))
    (LD1 (S DC-26)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
L3 (A1 A2 A3 A4 A5) ! (FORALL [Y:I]
  (IMPLIES (LD1 Y) (LD1 (S Y)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
L6 (A1 A2 A3 A4 A5) ! (LD1 (S (S (S ONE)))) OTTER: (NIL) (L1 L3)
L8 (A1 A2 A3 A4 A5) ! (AND (LD1 ONE) (D ONE)) OTTER: (NIL) (L1 A4)
```

The Curious Inference of Boolos in Mizar and OMEGA

```

L7 (A1 A2 A3 ! (LD2 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
A4 A5)

L11 (A1 A2 A3 ! (FORALL [DC-93:I] OTTER: (NIL) (L3 A5)
A4 A5) (IMPLIES
(AND (LD1 DC-93) (D DC-93))
(AND
(LD1 (S DC-93))
(D (S DC-93))))))

L10 (A1 A2 A3 ! (FORALL [DC-86:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
A4 A5) (IMPLIES
(AND (LD1 DC-86) (D DC-86))
(LD2 (S DC-86))))

L9 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y))))

L12 (A1 A2 A3 ! (LD2 (S ONE)) OTTER: (NIL) (L7 L9)
A4 A5)

L13 (A1 A2 A3 ! (FORALL [N:I] DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)
A4 A5) (IMPLIES
(LD1 N)
(FORALL [X:I]
(IMPLIES
(LD1 X)
(LD2 (F N X))))))

LD3 (LD3) ! (=DEF LOCAL-DEF
LD3
([N].
(FORALL [X:I]
(IMPLIES
(LD1 X)
(LD2 (F N X))))))

L18 (A1 A2 A3 ! (FORALL [DC-151:I] DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L19 LD4)
A4 A5) (IMPLIES
(LD1 DC-151)
(LD2 (F ONE DC-151))))

L15 (A1 A2 A3 ! (LD3 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L18 LD3)
A4 A5)
...

L16 (A1 A2 A3 ! (FORALL [N:I] OPEN
A4 A5) (IMPLIES (LD3 N) (LD3 (S N))))

L17 (A1 A2 A3 ! (FORALL [DC-123:I] OTTER: (NIL) (L15 L16)
A4 A5) (IMPLIES
(FORALL [DC-134:(0 I)]
(IMPLIES
(AND
(DC-134 ONE)
(FORALL [DC-138:I]
(IMPLIES
(DC-134 DC-138)
(DC-134 (S DC-138))))))
(DC-134 DC-123)))
(LD3 DC-123)))

L14 (A1 A2 A3 ! (FORALL [N:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L17 LD1)
A4 A5) (IMPLIES (LD1 N) (LD3 N)))

LD4 (LD4) ! (=DEF LD4 ((X).(LD2 (F ONE X))) LOCAL-DEF
...)

L20 (A1 A2 A3 ! (LD4 ONE) OPEN
A4 A5)
...

L21 (A1 A2 A3 ! (FORALL [X:I] OPEN
A4 A5) (IMPLIES (LD4 X) (LD4 (S X))))
...

L22 (A1 A2 A3 ! (FORALL [DC-165:I] OPEN
A4 A5) (IMPLIES
(FORALL [DC-176:(0 I)]
(IMPLIES
(AND
(DC-176 ONE)
(FORALL [DC-180:I]
(IMPLIES
(DC-176 DC-180)
(DC-176 (S DC-180))))))
(DC-176 DC-165)))
(LD4 DC-165)))

L19 (A1 A2 A3 ! (FORALL [X:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L22 LD1)
A4 A5)

```

```

A4 A5)      (IMPLIES (LD1 X) (LD4 X)))
...
CONC (A1 A2 A3 ! (D                                     OPEN
      A4 A5)      (F
                  (S (S (S ONE))))
                  (S (S (S ONE))))))

;;; step 4.3.1.4
OMEGA: SUPPORT L22 (L20 L21)
...
L21 (A1 A2 A3 ! (FORALL [X:I]                             OPEN
      A4 A5)      (IMPLIES (LD4 X) (LD4 (S X))))
...
L20 (A1 A2 A3 ! (LD4 ONE)                                 OPEN
      A4 A5)
...
L22 (A1 A2 A3 ! (FORALL [DC-165:I]                         OPEN
      A4 A5)      (IMPLIES
                  (FORALL [DC-176:(0 I)]
                    (IMPLIES
                     (AND
                      (DC-176 ONE)
                      (FORALL [DC-180:I]
                        (IMPLIES
                         (DC-176 DC-180)
                         (DC-176 (S DC-180))))))
                     (DC-176 DC-165)))
                  (LD4 DC-165)))

;;; step 4.3.1.4
OMEGA: CALL-OTTER-ON-NODE L22 ...
...
----- PROOF -----
...

;;; step 4.3.1.4.1.1
OMEGA: DEFN-CONTRACT-LOCAL-DEF L20 () LD4 (0)
;;;CSM Arbitrary [2]: 0 provers have to be killed

OMEGA: show-pds
A1 (A1)      ! (FORALL [N:I] (= (F N ONE) (S ONE)))      HYP
A2 (A2)      ! (FORALL [X:I]
              (=
               (F ONE (S X))
               (S (S (F ONE X)))))                       HYP
A3 (A3)      ! (FORALL [N:I,X:I]
              (=
               (F (S N) (S X))
               (F N (F (S N) X))))                       HYP
A4 (A4)      ! (D ONE)                                     HYP
A5 (A5)      ! (FORALL [X:I]
              (IMPLIES (D X) (D (S X))))                 HYP
LD1 (LD1)    ! (=DEF
              LD1
              ([2].
               (FORALL [X:(0 I)]
                 (IMPLIES
                  (AND
                   (X ONE)
                   (FORALL [Y:I]
                     (IMPLIES (X Y) (X (S Y))))))
                 (X Z))))                                LOCAL-DEF
LD2 (LD2)    ! (=DEF LD2 ([2].(AND (LD1 Z) (D Z))))      LOCAL-DEF
L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)]                       OTTER: (NIL)
      A4 A5)      (IMPLIES
                  (AND
                   (DC-13 ONE)
                   (FORALL [DC-17:I]
                     (IMPLIES
                      (DC-13 DC-17)
                      (DC-13 (S DC-17))))))
                  (DC-13 ONE)))
L1 (A1 A2 A3 ! (LD1 ONE)                                     DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
      A4 A5)
L5 (A1 A2 A3 ! (FORALL [DC-48:I]                             OTTER: (NIL)
      A4 A5)      (IMPLIES

```

```

(FORALL [DC-59:(0 I)]
(IMPLIES
(AND
(DC-59 ONE)
(FORALL [DC-63:I]
(IMPLIES
(DC-59 DC-63)
(DC-59 (S DC-63))))))
(DC-59 DC-48)))
(FORALL [DC-68:(0 I)]
(IMPLIES
(AND
(DC-68 ONE)
(FORALL [DC-72:I]
(IMPLIES
(DC-68 DC-72)
(DC-68 (S DC-72))))))
(DC-68 (S DC-48))))))

L4 (A1 A2 A3 ! (FORALL [DC-26:I]
A4 A5) (IMPLIES
(FORALL [DC-37:(0 I)]
(IMPLIES
(AND
(DC-37 ONE)
(FORALL [DC-41:I]
(IMPLIES
(DC-37 DC-41)
(DC-37 (S DC-41))))))
(DC-37 DC-26)))
(LD1 (S DC-26))))

DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)

L3 (A1 A2 A3 ! (FORALL [Y:I]
A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y))))

DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)

L6 (A1 A2 A3 ! (LD1 (S (S (S ONE))))
A4 A5) OTTER: (NIL) (L1 L3)

L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE))
A4 A5) OTTER: (NIL) (L1 A4)

L7 (A1 A2 A3 ! (LD2 ONE)
A4 A5) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)

L11 (A1 A2 A3 ! (FORALL [DC-93:I]
A4 A5) (IMPLIES
(AND (LD1 DC-93) (D DC-93))
(AND
(LD1 (S DC-93))
(D (S DC-93))))))

OTTER: (NIL) (L3 A5)

L10 (A1 A2 A3 ! (FORALL [DC-86:I]
A4 A5) (IMPLIES
(AND (LD1 DC-86) (D DC-86))
(LD2 (S DC-86))))

DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)

L9 (A1 A2 A3 ! (FORALL [Y:I]
A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y))))

DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)

L12 (A1 A2 A3 ! (LD2 (S ONE))
A4 A5) OTTER: (NIL) (L7 L9)

L13 (A1 A2 A3 ! (FORALL [N:I]
A4 A5) (IMPLIES
(LD1 N)
(FORALL [X:I]
(IMPLIES
(LD1 X)
(LD2 (F N X))))))

DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)

LD3 (LD3) ! (=DEF LOCAL-DEF
LD3
([N]
(FORALL [X:I]
(IMPLIES
(LD1 X)
(LD2 (F N X))))))

L18 (A1 A2 A3 ! (FORALL [DC-151:I]
A4 A5) (IMPLIES
(LD1 DC-151)
(LD2 (F ONE DC-151))))

DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L19 LD4)

L15 (A1 A2 A3 ! (LD3 ONE)
A4 A5) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L18 LD3)

...

L16 (A1 A2 A3 ! (FORALL [N:I]
A4 A5) (IMPLIES (LD3 N) (LD3 (S N))))

OPEN

L17 (A1 A2 A3 ! (FORALL [DC-123:I]
A4 A5) OTTER: (NIL) (L15 L16)

```

```

A4 A5) (IMPLIES
        (FORALL [DC-134:(0 I)])
        (IMPLIES
          (AND
            (DC-134 ONE)
            (FORALL [DC-138:I]
              (IMPLIES
                (DC-134 DC-138)
                (DC-134 (S DC-138))))))
          (DC-134 DC-123)))
        (LD3 DC-123)))

L14 (A1 A2 A3 ! (FORALL [N:I]
A4 A5) (IMPLIES (LD1 N) (LD3 N))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L17 LD1)

LD4 (LD4) ! (=DEF LD4 ([X].LD2 (F ONE X))) LOCAL-DEF
...

L23 (A1 A2 A3 ! (LD2 (F ONE ONE)) OPEN
A4 A5)

L20 (A1 A2 A3 ! (LD4 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L23 LD4)
A4 A5)
...

L21 (A1 A2 A3 ! (FORALL [X:I]
A4 A5) (IMPLIES (LD4 X) (LD4 (S X)))) OPEN

L22 (A1 A2 A3 ! (FORALL [DC-165:I]
A4 A5) (IMPLIES
        (FORALL [DC-176:(0 I)])
        (IMPLIES
          (AND
            (DC-176 ONE)
            (FORALL [DC-180:I]
              (IMPLIES
                (DC-176 DC-180)
                (DC-176 (S DC-180))))))
          (DC-176 DC-165)))
        (LD4 DC-165))) OTTER: (NIL) (L20 L21)

L19 (A1 A2 A3 ! (FORALL [X:I]
A4 A5) (IMPLIES (LD1 X) (LD4 X))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L22 LD1)
...

CONC (A1 A2 A3 ! (D
A4 A5) (F
        (S (S (S (S ONE))))
        (S (S (S (S ONE)))))) OPEN

;;; step 4.3.1.4.1.2-3
OMEGA: SUPPORT L23 (A1 L12)

A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP

L12 (A1 A2 A3 ! (LD2 (S ONE)) OTTER: (NIL) (L7 L9)
A4 A5)
...

L23 (A1 A2 A3 ! (LD2 (F ONE ONE)) OPEN
A4 A5)

;;; step 4.3.1.4.1.2-3
OMEGA: CALL-UTTER-ON-NODE L23 ...
...
----- PROOF -----
...

;;; step 4.3.1.4.2.1-6
OMEGA: DEFN-CONTRACT-LOCAL-DEF L21 () LD4 (1 0 1 0)
;;;CSM Arbitrary [2]: 0 provers have to be killed

;;; step 4.3.1.4.2.1-6
OMEGA: DEFN-CONTRACT-LOCAL-DEF L24 () LD4 (1 0 2 0)
;;;CSM Arbitrary [2]: 0 provers have to be killed

;;; step 4.3.1.4.2.1-6
OMEGA: SUPPORT L25 (A2 L9)

A2 (A2) ! (FORALL [X:I]
        (=
          (F ONE (S X))
          (S (F ONE X)))) HYP

L9 (A1 A2 A3 ! (FORALL [Y:I]
A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
...

L25 (A1 A2 A3 ! (FORALL [DC-201:I]
A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y)))) OPEN

```

The Curious Inference of Boolos in Mizar and OMEGA

```

A4 A5)      (IMPLIES
            (LD2 (F ONE DC-201))
            (LD2 (F ONE (S DC-201))))

;; step 4.3.1.4.2.1-6
OMEGA: CALL-OTTER-ON-NODE L25 ...
...
----- PROOF -----
...

OMEGA: show-pds
A1 (A1)      ! (FORALL [N:I] (= (F N ONE) (S ONE)))           HYP
A2 (A2)      ! (FORALL [X:I]
              (=
               (F ONE (S X))
               (S (S (F ONE X)))))                             HYP
A3 (A3)      ! (FORALL [N:I,X:I]
              (=
               (F (S N) (S X))
               (F N (F (S N) X))))                             HYP
A4 (A4)      ! (D ONE)                                         HYP
A5 (A5)      ! (FORALL [X:I]
              (IMPLIES (D X) (D (S X))))                       HYP

LD1 (LD1)    ! (=DEF
              LD1
              ([Z].
               (FORALL [X:(O I)]
                (IMPLIES
                 (AND
                  (X ONE)
                  (FORALL [Y:I]
                   (IMPLIES (X Y) (X (S Y))))))
                (X Z))))                                         LOCAL-DEF

LD2 (LD2)    ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z))))             LOCAL-DEF

L2 (A1 A2 A3
   A4 A5)    ! (FORALL [DC-13:(O I)]
              (IMPLIES
               (AND
                (DC-13 ONE)
                (FORALL [DC-17:I]
                 (IMPLIES
                  (DC-13 DC-17)
                  (DC-13 (S DC-17))))))
              (DC-13 ONE)))                                     OTTER: (NIL)

L1 (A1 A2 A3
   A4 A5)    ! (LD1 ONE)                                         DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)

L5 (A1 A2 A3
   A4 A5)    ! (FORALL [DC-48:I]
              (IMPLIES
               (FORALL [DC-59:(O I)]
                (IMPLIES
                 (AND
                  (DC-59 ONE)
                  (FORALL [DC-63:I]
                   (IMPLIES
                    (DC-59 DC-63)
                    (DC-59 (S DC-63))))))
                (DC-59 (S DC-48)))
              (FORALL [DC-68:(O I)]
               (IMPLIES
                (AND
                 (DC-68 ONE)
                 (FORALL [DC-72:I]
                  (IMPLIES
                   (DC-68 DC-72)
                   (DC-68 (S DC-72))))))
              (DC-68 (S DC-48))))))                             OTTER: (NIL)

L4 (A1 A2 A3
   A4 A5)    ! (FORALL [DC-26:I]
              (IMPLIES
               (FORALL [DC-37:(O I)]
                (IMPLIES
                 (AND
                  (DC-37 ONE)
                  (FORALL [DC-41:I]
                   (IMPLIES
                    (DC-37 DC-41)
                    (DC-37 (S DC-41))))))
                (DC-37 DC-26)))
              (LD1 (S DC-26))))                             DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)

L3 (A1 A2 A3
   A4 A5)    ! (FORALL [Y:I]
              (IMPLIES (LD1 Y) (LD1 (S Y))))                   DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)

```

L6	(A1 A2 A3 ! (LD1 (S (S (S (S ONE)))))) A4 A5)	OTTER: (NIL) (L1 L3)
L8	(A1 A2 A3 ! (AND (LD1 ONE) (D ONE)) A4 A5)	OTTER: (NIL) (L1 A4)
L7	(A1 A2 A3 ! (LD2 ONE) A4 A5)	DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
L11	(A1 A2 A3 ! (FORALL [DC-93:I] A4 A5) (IMPLIES (AND (LD1 DC-93) (D DC-93)) (AND (LD1 (S DC-93)) (D (S DC-93))))))	OTTER: (NIL) (L3 A5)
L10	(A1 A2 A3 ! (FORALL [DC-86:I] A4 A5) (IMPLIES (AND (LD1 DC-86) (D DC-86)) (LD2 (S DC-86))))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
L9	(A1 A2 A3 ! (FORALL [Y:I] A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
L12	(A1 A2 A3 ! (LD2 (S ONE)) A4 A5)	OTTER: (NIL) (L7 L9)
L13	(A1 A2 A3 ! (FORALL [N:I] A4 A5) (IMPLIES (LD1 N) (FORALL [X:I] (IMPLIES (LD1 X) (LD2 (F N X))))))	DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)
LD3	(LD3) ! (=DEF LD3 ([N]. (FORALL [X:I] (IMPLIES (LD1 X) (LD2 (F N X))))))	LOCAL-DEF
L18	(A1 A2 A3 ! (FORALL [DC-151:I] A4 A5) (IMPLIES (LD1 DC-151) (LD2 (F ONE DC-151))))	DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L19 LD4)
L15	(A1 A2 A3 ! (LD3 ONE) A4 A5) ...	DEFN-CONTRACT-LOCAL-DEF: ((0)) (L18 LD3)
L16	(A1 A2 A3 ! (FORALL [N:I] A4 A5) (IMPLIES (LD3 N) (LD3 (S N))))	OPEN
L17	(A1 A2 A3 ! (FORALL [DC-123:I] A4 A5) (IMPLIES (FORALL [DC-134:(0 I)] (IMPLIES (AND (DC-134 ONE) (FORALL [DC-138:I] (IMPLIES (DC-134 DC-138) (DC-134 (S DC-138)))))) (DC-134 DC-123)) (LD3 DC-123))))	OTTER: (NIL) (L15 L16)
L14	(A1 A2 A3 ! (FORALL [N:I] A4 A5) (IMPLIES (LD1 N) (LD3 N)))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L17 LD1)
LD4	(LD4) ! (=DEF LD4 ([X].(LD2 (F ONE X))))	LOCAL-DEF
L23	(A1 A2 A3 ! (LD2 (F ONE ONE)) A4 A5)	OTTER: (NIL) (L12 A1)
L20	(A1 A2 A3 ! (LD4 ONE) A4 A5)	DEFN-CONTRACT-LOCAL-DEF: ((0)) (L23 LD4)
L25	(A1 A2 A3 ! (FORALL [DC-201:I] A4 A5) (IMPLIES (LD2 (F ONE DC-201)) (LD2 (F ONE (S DC-201))))))	OTTER: (NIL) (L9 A2)
L24	(A1 A2 A3 ! (FORALL [DC-194:I] A4 A5) (IMPLIES (LD2 (F ONE DC-194)) (LD4 (S DC-194))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L25 LD4)
L21	(A1 A2 A3 ! (FORALL [X:I] A4 A5) (IMPLIES (LD4 X) (LD4 (S X))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L24 LD4)



The Curious Inference of Boolos in Mizar and OMEGA

```

L22 (A1 A2 A3 ! (FORALL [DC-165:I]
A4 A5) (IMPLIES
(FORALL [DC-176:(0 I)]
(IMPLIES
(AND
(DC-176 ONE)
(FORALL [DC-180:I]
(IMPLIES
(DC-176 DC-180)
(DC-176 (S DC-180))))))
(DC-176 DC-165)))
(LD4 DC-165)))
OTTER: (NIL) (L20 L21)

L19 (A1 A2 A3 ! (FORALL [X:I]
A4 A5) (IMPLIES (LD1 X) (LD4 X)))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L22 LD1)
...

CONC (A1 A2 A3 ! (D
A4 A5) (F
(S (S (S (S ONE))))
(S (S (S (S ONE))))))
OPEN

;;; step 4.3.2.1
OMEGA: FORALL L16 n1 ()

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE)))
HYP
A2 (A2) ! (FORALL [X:I]
(=
(F ONE (S X))
(S (S (F ONE X))))))
HYP
A3 (A3) ! (FORALL [N:I,X:I]
(=
(F (S N) (S X))
(F N (F (S N) X))))
HYP
A4 (A4) ! (D ONE)
HYP
A5 (A5) ! (FORALL [X:I]
(IMPLIES (D X) (D (S X))))
HYP

LD1 (LD1) ! (=DEF
LD1
([Z]
(FORALL [X:(0 I)]
(IMPLIES
(AND
(X ONE)
(FORALL [Y:I]
(IMPLIES (X Y) (X (S Y))))))
(X Z))))))
LOCAL-DEF

LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z))))
LOCAL-DEF

L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)]
A4 A5) (IMPLIES
(AND
(DC-13 ONE)
(FORALL [DC-17:I]
(IMPLIES
(DC-13 DC-17)
(DC-13 (S DC-17))))))
(DC-13 ONE)))
OTTER: (NIL)

L1 (A1 A2 A3 ! (LD1 ONE)
A4 A5)
DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)

L5 (A1 A2 A3 ! (FORALL [DC-48:I]
A4 A5) (IMPLIES
(FORALL [DC-59:(0 I)]
(IMPLIES
(AND
(DC-59 ONE)
(FORALL [DC-63:I]
(IMPLIES
(DC-59 DC-63)
(DC-59 (S DC-63))))))
(DC-59 DC-48)))
(FORALL [DC-68:(0 I)]
(IMPLIES
(AND
(DC-68 ONE)
(FORALL [DC-72:I]
(IMPLIES
(DC-68 DC-72)
(DC-68 (S DC-72))))))
(DC-68 (S DC-48))))))
OTTER: (NIL)

```

L4	(A1 A2 A3 ! (FORALL [DC-26:I] A4 A5) (IMPLIES (FORALL [DC-37:(0 I)] (IMPLIES (AND (DC-37 ONE) (FORALL [DC-41:I] (IMPLIES (DC-37 DC-41) (DC-37 (S DC-41)))))) (DC-37 DC-26)))) (LD1 (S DC-26))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
L3	(A1 A2 A3 ! (FORALL [Y:I] A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
L6	(A1 A2 A3 ! (LD1 (S (S (S (S ONE)))))) A4 A5)	OTTER: (NIL) (L1 L3)
L8	(A1 A2 A3 ! (AND (LD1 ONE) (D ONE)) A4 A5)	OTTER: (NIL) (L1 A4)
L7	(A1 A2 A3 ! (LD2 ONE) A4 A5)	DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
L11	(A1 A2 A3 ! (FORALL [DC-93:I] A4 A5) (IMPLIES (AND (LD1 DC-93) (D DC-93)) (AND (LD1 (S DC-93)) (D (S DC-93))))))	OTTER: (NIL) (L3 A5)
L10	(A1 A2 A3 ! (FORALL [DC-86:I] A4 A5) (IMPLIES (AND (LD1 DC-86) (D DC-86)) (LD2 (S DC-86))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
L9	(A1 A2 A3 ! (FORALL [Y:I] A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
L12	(A1 A2 A3 ! (LD2 (S ONE)) A4 A5)	OTTER: (NIL) (L7 L9)
L13	(A1 A2 A3 ! (FORALL [N:I] A4 A5) (IMPLIES (LD1 N) (FORALL [X:I] (IMPLIES (LD1 X) (LD2 (F N X))))))	DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)
LD3	(LD3) ! (=DEF LD3 ([N]. (FORALL [X:I] (IMPLIES (LD1 X) (LD2 (F N X))))))	LOCAL-DEF
L18	(A1 A2 A3 ! (FORALL [DC-151:I] A4 A5) (IMPLIES (LD1 DC-151) (LD2 (F ONE DC-151))))	DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L19 LD4)
L15	(A1 A2 A3 ! (LD3 ONE) A4 A5) ...	DEFN-CONTRACT-LOCAL-DEF: ((0)) (L18 LD3)
L26	(A1 A2 A3 ! (IMPLIES (LD3 N1) (LD3 (S N1))) A4 A5)	OPEN
L16	(A1 A2 A3 ! (FORALL [N:I] A4 A5) (IMPLIES (LD3 N) (LD3 (S N))))	FORALLI: (N1) (L26)
L17	(A1 A2 A3 ! (FORALL [DC-123:I] A4 A5) (IMPLIES (FORALL [DC-134:(0 I)] (IMPLIES (AND (DC-134 ONE) (FORALL [DC-138:I] (IMPLIES (DC-134 DC-138) (DC-134 (S DC-138)))))) (DC-134 DC-123)))) (LD3 DC-123))))	OTTER: (NIL) (L15 L16)
L14	(A1 A2 A3 ! (FORALL [N:I] A4 A5) (IMPLIES (LD1 N) (LD3 N)))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L17 LD1)
LD4	(LD4) ! (=DEF LD4 ([X].(LD2 (F ONE X))))	LOCAL-DEF

The Curious Inference of Boolos in Mizar and OMEGA

```

L23 (A1 A2 A3 ! (LD2 (F ONE ONE)))                                OTTER: (NIL) (L12 A1)
    A4 A5)

L20 (A1 A2 A3 ! (LD4 ONE))                                        DEFN-CONTRACT-LOCAL-DEF: ((0)) (L23 LD4)
    A4 A5)

L25 (A1 A2 A3 ! (FORALL [DC-201:I]                                OTTER: (NIL) (L9 A2)
    A4 A5) (IMPLIES
            (LD2 (F ONE DC-201))
            (LD2 (F ONE (S DC-201)))))

L24 (A1 A2 A3 ! (FORALL [DC-194:I]                                DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L25 LD4)
    A4 A5) (IMPLIES
            (LD2 (F ONE DC-194))
            (LD4 (S DC-194))))

L21 (A1 A2 A3 ! (FORALL [X:I]                                     DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L24 LD4)
    A4 A5) (IMPLIES (LD4 X) (LD4 (S X))))

L22 (A1 A2 A3 ! (FORALL [DC-165:I]                                OTTER: (NIL) (L20 L21)
    A4 A5) (IMPLIES
            (FORALL [DC-176:(0 I)]
              (IMPLIES
                (AND
                  (DC-176 ONE)
                  (FORALL [DC-180:I]
                    (IMPLIES
                      (DC-176 DC-180)
                      (DC-176 (S DC-180))))))
                (DC-176 DC-165))))
            (LD4 DC-165)))

L19 (A1 A2 A3 ! (FORALL [X:I]                                     DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L22 LD1)
    A4 A5) (IMPLIES (LD1 X) (LD4 X)))
    ...

CONC (A1 A2 A3 ! (D                                             OPEN
    A4 A5) (F
            (S (S (S ONE))))
            (S (S (S ONE)))))

;; step 4.3.2.1,3
OMEGA: IMPI

IMPLICATION (NDLINE) Implication to justify: [L26]L26
;;CSM Arbitrary [2]: 0 provers have to be killed

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE)))                    HYP
A2 (A2) ! (FORALL [X:I]                                          HYP
    (=
      (F ONE (S X))
      (S (S (F ONE X)))))
A3 (A3) ! (FORALL [N:I,X:I]                                       HYP
    (=
      (F (S N) (S X))
      (F N (F (S N) X))))
A4 (A4) ! (D ONE)                                                  HYP
A5 (A5) ! (FORALL [X:I]                                          HYP
    (IMPLIES (D X) (D (S X))))
LD1 (LD1) ! (=DEF                                               LOCAL-DEF
    LD1
    ([2].
      (FORALL [X:(0 I)]
        (IMPLIES
          (AND
            (X ONE)
            (FORALL [Y:I]
              (IMPLIES (X Y) (X (S Y))))))
          (X Z))))))
LD2 (LD2) ! (=DEF LD2 ([2].(AND (LD1 Z) (D Z))))                  LOCAL-DEF
L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)]                                OTTER: (NIL)
    A4 A5) (IMPLIES
            (AND
              (DC-13 ONE)
              (FORALL [DC-17:I]
                (IMPLIES
                  (DC-13 DC-17)
                  (DC-13 (S DC-17))))))
            (DC-13 ONE)))
L1 (A1 A2 A3 ! (LD1 ONE))                                        DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)

```

```

A4 A5)
L5 (A1 A2 A3 ! (FORALL [DC-48:I]                                OTTER: (NIL)
A4 A5)      (IMPLIES
              (FORALL [DC-59:(0 I)]
                (IMPLIES
                  (AND
                    (DC-59 ONE)
                    (FORALL [DC-63:I]
                      (IMPLIES
                        (DC-59 DC-63)
                        (DC-59 (S DC-63))))))
                  (DC-59 DC-48)))
              (FORALL [DC-68:(0 I)]
                (IMPLIES
                  (AND
                    (DC-68 ONE)
                    (FORALL [DC-72:I]
                      (IMPLIES
                        (DC-68 DC-72)
                        (DC-68 (S DC-72))))))
                  (DC-68 (S DC-48))))))
L4 (A1 A2 A3 ! (FORALL [DC-26:I]                                DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
A4 A5)      (IMPLIES
              (FORALL [DC-37:(0 I)]
                (IMPLIES
                  (AND
                    (DC-37 ONE)
                    (FORALL [DC-41:I]
                      (IMPLIES
                        (DC-37 DC-41)
                        (DC-37 (S DC-41))))))
                  (DC-37 DC-26)))
              (LD1 (S DC-26))))
L3 (A1 A2 A3 ! (FORALL [Y:I]                                    DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
A4 A5)      (IMPLIES (LD1 Y) (LD1 (S Y))))
L6 (A1 A2 A3 ! (LD1 (S (S (S (S ONE))))))                       OTTER: (NIL) (L1 L3)
A4 A5)
L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE)))                          OTTER: (NIL) (L1 A4)
A4 A5)
L7 (A1 A2 A3 ! (LD2 ONE))                                        DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
A4 A5)
L11 (A1 A2 A3 ! (FORALL [DC-93:I]                                OTTER: (NIL) (L3 A5)
A4 A5)      (IMPLIES
              (AND (LD1 DC-93) (D DC-93))
              (AND
                (LD1 (S DC-93))
                (D (S DC-93))))))
L10 (A1 A2 A3 ! (FORALL [DC-86:I]                                DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
A4 A5)      (IMPLIES
              (AND (LD1 DC-86) (D DC-86))
              (LD2 (S DC-86))))
L9 (A1 A2 A3 ! (FORALL [Y:I]                                    DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
A4 A5)      (IMPLIES (LD2 Y) (LD2 (S Y))))
L12 (A1 A2 A3 ! (LD2 (S ONE)))                                  OTTER: (NIL) (L7 L9)
A4 A5)
L13 (A1 A2 A3 ! (FORALL [N:I]                                    DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)
A4 A5)      (IMPLIES
              (LD1 N)
              (FORALL [X:I]
                (IMPLIES
                  (LD1 X)
                  (LD2 (F N X))))))
LDS (LD3) ! (*DEF                                             LOCAL-DEF
          LDS
          ([N]
            (FORALL [X:I]
              (IMPLIES
                (LD1 X)
                (LD2 (F N X))))))
L18 (A1 A2 A3 ! (FORALL [DC-151:I]                               DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L19 LD4)
A4 A5)      (IMPLIES
              (LD1 DC-151)
              (LD2 (F ONE DC-151))))
L15 (A1 A2 A3 ! (LD3 ONE))                                     DEFN-CONTRACT-LOCAL-DEF: ((0)) (L18 LD3)
A4 A5)
...

```

The Curious Inference of Boolos in Mizar and OMEGA

```

L28 (L27 A1 ! (LD3 (S N1))
A2 A3 A4
A5) OPEN

L26 (A1 A2 A3 ! (IMPLIES (LD3 N1) (LD3 (S N1)))
A4 A5) IMPI: (L28)

L16 (A1 A2 A3 ! (FORALL [N:I]
A4 A5) (IMPLIES (LD3 N) (LD3 (S N)))) FORALLI: (N1) (L26)

L17 (A1 A2 A3 ! (FORALL [DC-123:I]
A4 A5) (IMPLIES
(FORALL [DC-134:(O I)]
(IMPLIES
(AND
(DC-134 ONE)
(FORALL [DC-138:I]
(IMPLIES
(DC-134 DC-138)
(DC-134 (S DC-138))))))
(DC-134 DC-123)))
(LD3 DC-123))) OTTER: (NIL) (L15 L16)

L14 (A1 A2 A3 ! (FORALL [N:I]
A4 A5) (IMPLIES (LD1 N) (LD3 N))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L17 LD1)

LD4 (LD4) ! (=DEF LD4 ([X].(LD2 (F ONE X)))) LOCAL-DEF

L27 (L27) ! (LD3 N1) HYP

L23 (A1 A2 A3 ! (LD2 (F ONE ONE))
A4 A5) OTTER: (NIL) (L12 A1)

L20 (A1 A2 A3 ! (LD4 ONE)
A4 A5) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L23 LD4)

L25 (A1 A2 A3 ! (FORALL [DC-201:I]
A4 A5) (IMPLIES
(LD2 (F ONE DC-201))
(LD2 (F ONE (S DC-201)))))) OTTER: (NIL) (L9 A2)

L24 (A1 A2 A3 ! (FORALL [DC-194:I]
A4 A5) (IMPLIES
(LD2 (F ONE DC-194))
(LD4 (S DC-194)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L25 LD4)

L21 (A1 A2 A3 ! (FORALL [X:I]
A4 A5) (IMPLIES (LD4 X) (LD4 (S X)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L24 LD4)

L22 (A1 A2 A3 ! (FORALL [DC-165:I]
A4 A5) (IMPLIES
(FORALL [DC-176:(O I)]
(IMPLIES
(AND
(DC-176 ONE)
(FORALL [DC-180:I]
(IMPLIES
(DC-176 DC-180)
(DC-176 (S DC-180))))))
(DC-176 DC-165)))
(LD4 DC-165))) OTTER: (NIL) (L20 L21)

L19 (A1 A2 A3 ! (FORALL [X:I]
A4 A5) (IMPLIES (LD1 X) (LD4 X))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L22 LD1)
...

CONC (A1 A2 A3 ! (D
A4 A5) (F
(S (S (S (S ONE))))
(S (S (S (S ONE)))))) OPEN

;; step 4.3.2.2
OMEGA: DEFN-EXPAND-LOCAL-DEF () L27 LD3 (0)
;;CSM Arbitrary [2]: 0 provers have to be killed

;; step 4.3.2.4
OMEGA: DEFN-CONTRACT-LOCAL-DEF L28 () LD3 (0)
;;CSM Arbitrary [2]: 0 provers have to be killed

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP
A2 (A2) ! (FORALL [X:I]
(=
(F ONE (S X))
(S (S (F ONE X)))))) HYP
A3 (A3) ! (FORALL [N:I,X:I]
(=
(F (S N) (S X))

```

```

(F N (F (S N) X)))

A4 (A4) ! (D ONE) HYP
A5 (A5) ! (FORALL [X:I] HYP
  (IMPLIES (D X) (D (S X))))

LD1 (LD1) ! (=DEF LOCAL-DEF
  LD1
  ([Z]
  (FORALL [X:(0 I)]
  (IMPLIES
  (AND
  (X ONE)
  (FORALL [Y:I]
  (IMPLIES (X Y) (X (S Y))))))
  (X Z))))))

LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z)))) LOCAL-DEF

L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)] OTTER: (NIL)
  A4 A5) (IMPLIES
  (AND
  (DC-13 ONE)
  (FORALL [DC-17:I]
  (IMPLIES
  (DC-13 DC-17)
  (DC-13 (S DC-17))))))
  (DC-13 ONE)))

L1 (A1 A2 A3 ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
  A4 A5)

L5 (A1 A2 A3 ! (FORALL [DC-48:I] OTTER: (NIL)
  A4 A5) (IMPLIES
  (FORALL [DC-59:(0 I)]
  (IMPLIES
  (AND
  (DC-59 ONE)
  (FORALL [DC-63:I]
  (IMPLIES
  (DC-59 DC-63)
  (DC-59 (S DC-63))))))
  (DC-59 DC-48)))
  (FORALL [DC-68:(0 I)]
  (IMPLIES
  (AND
  (DC-68 ONE)
  (FORALL [DC-72:I]
  (IMPLIES
  (DC-68 DC-72)
  (DC-68 (S DC-72))))))
  (DC-68 (S DC-48))))))

L4 (A1 A2 A3 ! (FORALL [DC-26:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
  A4 A5) (IMPLIES
  (FORALL [DC-37:(0 I)]
  (IMPLIES
  (AND
  (DC-37 ONE)
  (FORALL [DC-41:I]
  (IMPLIES
  (DC-37 DC-41)
  (DC-37 (S DC-41))))))
  (DC-37 DC-26)))
  (LD1 (S DC-26))))

L3 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
  A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y))))

L6 (A1 A2 A3 ! (LD1 (S (S (S ONE)))) OTTER: (NIL) (L1 L3)
  A4 A5)

L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE)) OTTER: (NIL) (L1 A4)
  A4 A5)

L7 (A1 A2 A3 ! (LD2 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
  A4 A5)

L11 (A1 A2 A3 ! (FORALL [DC-93:I] OTTER: (NIL) (L3 A5)
  A4 A5) (IMPLIES
  (AND (LD1 DC-93) (D DC-93))
  (AND
  (LD1 (S DC-93))
  (D (S DC-93))))))

L10 (A1 A2 A3 ! (FORALL [DC-86:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
  A4 A5) (IMPLIES
  (AND (LD1 DC-86) (D DC-86))
  (LD2 (S DC-86))))

```

The Curious Inference of Boolos in Mizar and OMEGA

L9	(A1 A2 A3 A4 A5)	! (FORALL [Y:I] (IMPLIES (LD2 Y) (LD2 (S Y))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
L12	(A1 A2 A3 A4 A5)	! (LD2 (S ONE))	OTTER: (NIL) (L7 L9)
L13	(A1 A2 A3 A4 A5)	! (FORALL [N:I] (IMPLIES (LD1 N) (FORALL [X:I] (IMPLIES (LD1 X) (LD2 (F N X))))))	DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)
LD3	(LD3)	! (=DEF LD3 ([N]. (FORALL [X:I] (IMPLIES (LD1 X) (LD2 (F N X))))))	LOCAL-DEF
L18	(A1 A2 A3 A4 A5)	! (FORALL [DC-151:I] (IMPLIES (LD1 DC-151) (LD2 (F ONE DC-151))))	DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L19 LD4)
L15	(A1 A2 A3 A4 A5)	! (LD3 ONE)	DEFN-CONTRACT-LOCAL-DEF: ((0)) (L18 LD3)
		...	
L30	(L27 A1 A2 A3 A4 A5)	! (FORALL [DC-225:I] (IMPLIES (LD1 DC-225) (LD2 (F (S N1) DC-225))))	OPEN
L28	(L27 A1 A2 A3 A4 A5)	! (LD3 (S N1))	DEFN-CONTRACT-LOCAL-DEF: ((0)) (L30 LD3)
L26	(A1 A2 A3 A4 A5)	! (IMPLIES (LD3 N1) (LD3 (S N1)))	IMPI: (L28)
L16	(A1 A2 A3 A4 A5)	! (FORALL [N:I] (IMPLIES (LD3 N) (LD3 (S N))))	FORALLI: (N1) (L26)
L17	(A1 A2 A3 A4 A5)	! (FORALL [DC-123:I] (IMPLIES (FORALL [DC-134:(0 I)] (IMPLIES (AND (DC-134 ONE) (FORALL [DC-138:I] (IMPLIES (DC-134 DC-138) (DC-134 (S DC-138)))))) (DC-134 DC-123))) (LD3 DC-123)))	OTTER: (NIL) (L15 L16)
L14	(A1 A2 A3 A4 A5)	! (FORALL [N:I] (IMPLIES (LD1 N) (LD3 N)))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L17 LD1)
LD4	(LD4)	! (=DEF LD4 ([X].(LD2 (F ONE X))))	LOCAL-DEF
L27	(L27)	! (LD3 N1)	HYP
L29	(L27)	! (FORALL [DC-217:I] (IMPLIES (LD1 DC-217) (LD2 (F N1 DC-217))))	DEFN-EXPAND-LOCAL-DEF: ((0)) (L27 LD3)
L23	(A1 A2 A3 A4 A5)	! (LD2 (F ONE ONE))	OTTER: (NIL) (L12 A1)
L20	(A1 A2 A3 A4 A5)	! (LD4 ONE)	DEFN-CONTRACT-LOCAL-DEF: ((0)) (L23 LD4)
L25	(A1 A2 A3 A4 A5)	! (FORALL [DC-201:I] (IMPLIES (LD2 (F ONE DC-201)) (LD2 (F ONE (S DC-201))))	OTTER: (NIL) (L9 A2)
L24	(A1 A2 A3 A4 A5)	! (FORALL [DC-194:I] (IMPLIES (LD2 (F ONE DC-194)) (LD4 (S DC-194))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L25 LD4)
L21	(A1 A2 A3 A4 A5)	! (FORALL [X:I] (IMPLIES (LD4 X) (LD4 (S X))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L24 LD4)
L22	(A1 A2 A3 A4 A5)	! (FORALL [DC-165:I]	OTTER: (NIL) (L20 L21)

```

A4 A5) (IMPLIES
        (FORALL [DC-176:(0 I)]
          (IMPLIES
            (AND
              (DC-176 ONE)
              (FORALL [DC-180:I]
                (IMPLIES
                  (DC-176 DC-180)
                  (DC-176 (S DC-180))))))
            (DC-176 DC-165)))
        (LD4 DC-165)))

L19 (A1 A2 A3 ! (FORALL [X:I]
A4 A5) (IMPLIES (LD1 X) (LD4 X))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L22 LD1)
...

CONC (A1 A2 A3 ! (D
A4 A5) (F
        (S (S (S (S ONE))))
        (S (S (S (S ONE)))))) OPEN

;;; step 4.3.2.5
OMEGA: LOCAL-DEF-INTRO (LAM (X I) (LD2 (F (S N1) X)))

;;; step 4.3.2.6
OMEGA: LEMMA L30 (FORALL
                  (LAM (X I) (IMPLIES (LD1 X) (LD5 X))))

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP
A2 (A2) ! (FORALL [X:I]
          (=
            (F ONE (S X))
            (S (S (F ONE X))))) HYP
A3 (A3) ! (FORALL [N:I,X:I]
          (=
            (F (S N) (S X))
            (F N (F (S N) X)))) HYP
A4 (A4) ! (D ONE) HYP
A5 (A5) ! (FORALL [X:I]
          (IMPLIES (D X) (D (S X)))) HYP

LD1 (LD1) ! (=DEF
LD1
          ([Z].
            (FORALL [X:(0 I)]
              (IMPLIES
                (AND
                  (X ONE)
                  (FORALL [Y:I]
                    (IMPLIES (X Y) (X (S Y))))))
                (X Z)))) LOCAL-DEF
LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z)))) LOCAL-DEF

L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)]
A4 A5) (IMPLIES
        (AND
          (DC-13 ONE)
          (FORALL [DC-17:I]
            (IMPLIES
              (DC-13 DC-17)
              (DC-13 (S DC-17))))))
        (DC-13 ONE))) OTTER: (NIL)

L1 (A1 A2 A3 ! (LD1 ONE)
A4 A5) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)

L5 (A1 A2 A3 ! (FORALL [DC-48:I]
A4 A5) (IMPLIES
        (FORALL [DC-59:(0 I)]
          (IMPLIES
            (AND
              (DC-59 ONE)
              (FORALL [DC-63:I]
                (IMPLIES
                  (DC-59 DC-63)
                  (DC-59 (S DC-63))))))
            (DC-59 DC-48)))
        (FORALL [DC-68:(0 I)]
          (IMPLIES
            (AND
              (DC-68 ONE)
              (FORALL [DC-72:I]
                (IMPLIES
                  (DC-68 DC-72)
                  (DC-68 (S DC-72))))))
          (DC-68 (S DC-72)))))) OTTER: (NIL)

```



The Curious Inference of Boolos in Mizar and OMEGA

		(DC-68 (S DC-48))))))	
L4	(A1 A2 A3 A4 A5)	! (FORALL [DC-26:I] (IMPLIES (FORALL [DC-37:(O I)] (IMPLIES (AND (DC-37 ONE) (FORALL [DC-41:I] (IMPLIES (DC-37 DC-41) (DC-37 (S DC-41)))))) (DC-37 DC-26))) (LD1 (S DC-26))))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
L3	(A1 A2 A3 A4 A5)	! (FORALL [Y:I] (IMPLIES (LD1 Y) (LD1 (S Y))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
L6	(A1 A2 A3 A4 A5)	! (LD1 (S (S (S (S ONE))))))	OTTER: (NIL) (L1 L3)
L8	(A1 A2 A3 A4 A5)	! (AND (LD1 ONE) (D ONE))	OTTER: (NIL) (L1 A4)
L7	(A1 A2 A3 A4 A5)	! (LD2 ONE)	DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
L11	(A1 A2 A3 A4 A5)	! (FORALL [DC-93:I] (IMPLIES (AND (LD1 DC-93) (D DC-93)) (AND (LD1 (S DC-93)) (D (S DC-93))))))	OTTER: (NIL) (L3 A5)
L10	(A1 A2 A3 A4 A5)	! (FORALL [DC-86:I] (IMPLIES (AND (LD1 DC-86) (D DC-86)) (LD2 (S DC-86))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
L9	(A1 A2 A3 A4 A5)	! (FORALL [Y:I] (IMPLIES (LD2 Y) (LD2 (S Y))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
L12	(A1 A2 A3 A4 A5)	! (LD2 (S ONE))	OTTER: (NIL) (L7 L9)
L13	(A1 A2 A3 A4 A5)	! (FORALL [N:I] (IMPLIES (LD1 N) (FORALL [X:I] (IMPLIES (LD1 X) (LD2 (F N X))))))	DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)
LD3	(LD3)	! (=DEF LD3 ([N]. (FORALL [X:I] (IMPLIES (LD1 X) (LD2 (F N X))))))	LOCAL-DEF
L18	(A1 A2 A3 A4 A5)	! (FORALL [DC-151:I] (IMPLIES (LD1 DC-151) (LD2 (F ONE DC-151))))	DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L19 LD4)
L15	(A1 A2 A3 A4 A5)	! (LD3 ONE) ...	DEFN-CONTRACT-LOCAL-DEF: ((0)) (L18 LD3)
L30	(L27 A1 A2 A3 A4 A5)	! (FORALL [DC-225:I] (IMPLIES (LD1 DC-225) (LD2 (F (S N1) DC-225))))	OPEN
L28	(L27 A1 A2 A3 A4 A5)	! (LD3 (S N1))	DEFN-CONTRACT-LOCAL-DEF: ((0)) (L30 LD3)
L26	(A1 A2 A3 A4 A5)	! (IMPLIES (LD3 N1) (LD3 (S N1)))	IMPI: (L28)
L16	(A1 A2 A3 A4 A5)	! (FORALL [N:I] (IMPLIES (LD3 N) (LD3 (S N))))	FORALLI: (N1) (L26)
L17	(A1 A2 A3 A4 A5)	! (FORALL [DC-123:I] (IMPLIES (FORALL [DC-134:(O I)] (IMPLIES (AND (DC-134 ONE)	OTTER: (NIL) (L15 L16)

```

      (FORALL [DC-138:I]
      (IMPLIES
      (DC-134 DC-138)
      (DC-134 (S DC-138))))))
      (DC-134 DC-123)))
      (LD3 DC-123)))
L14 (A1 A2 A3 ! (FORALL [N:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L17 LD1)
      A4 A5) (IMPLIES (LD1 N) (LD3 N)))
      (LD4) ! (=DEF LD4 ([X].(LD2 (F ONE X)))) LOCAL-DEF
L27 (L27) ! (LD3 N1) HYP
L29 (L27) ! (FORALL [DC-217:I] DEFN-EXPAND-LOCAL-DEF: ((0)) (L27 LD3)
      (IMPLIES
      (LD1 DC-217)
      (LD2 (F N1 DC-217))))
L23 (A1 A2 A3 ! (LD2 (F ONE ONE)) OTTER: (NIL) (L12 A1)
      A4 A5)
L20 (A1 A2 A3 ! (LD4 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L23 LD4)
      A4 A5)
L25 (A1 A2 A3 ! (FORALL [DC-201:I] OTTER: (NIL) (L9 A2)
      A4 A5) (IMPLIES
      (LD2 (F ONE DC-201))
      (LD2 (F ONE (S DC-201))))))
L24 (A1 A2 A3 ! (FORALL [DC-194:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L25 LD4)
      A4 A5) (IMPLIES
      (LD2 (F ONE DC-194))
      (LD4 (S DC-194))))
L21 (A1 A2 A3 ! (FORALL [X:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L24 LD4)
      A4 A5) (IMPLIES (LD4 X) (LD4 (S X))))
L22 (A1 A2 A3 ! (FORALL [DC-165:I] OTTER: (NIL) (L20 L21)
      A4 A5) (IMPLIES
      (FORALL [DC-176:(0 I)]
      (IMPLIES
      (AND
      (DC-176 ONE)
      (FORALL [DC-180:I]
      (IMPLIES
      (DC-176 DC-180)
      (DC-176 (S DC-180))))))
      (DC-176 DC-165)))
      (LD4 DC-165)))
L19 (A1 A2 A3 ! (FORALL [X:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L22 LD1)
      A4 A5) (IMPLIES (LD1 X) (LD4 X)))
LD5 (LD5) ! (=DEF LD5 ([X].(LD2 (F (S N1) X)))) LOCAL-DEF
      ...
L31 (L27 A1 ! (FORALL [X:I] OPEN
      A2 A3 A4 (IMPLIES (LD1 X) (LD5 X)))
      A5)
      ...
CONC (A1 A2 A3 ! (D OPEN
      A4 A5) (F
      (S (S (S (S ONE))))
      (S (S (S (S ONE))))))
      ;; step 4.3.2.6
      OMEGA: DEFN-EXPAND-LOCAL-DEF L30 L31 LD5 (1 0 2 0)
      ;;CSM Arbitrary [2]: 0 provers have to be killed
      ;; step 4.3.2.7.1
      OMEGA: LEMMA L31 (LD5 ONE)
      ;; step 4.3.2.7.2
      OMEGA: LEMMA L31 (FORALL
      (LAM (X I) (IMPLIES (LD5 X) (LD5 (S X))))
      ;; step 4.3.2.7 -- Enough to ...
      OMEGA: DEFN-CONTRACT-LOCAL-DEF L31 () LD1 (1 0 1 0)
      ;;CSM Arbitrary [2]: 0 provers have to be killed
      OMEGA: show-pds
      A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE))) HYP
      A2 (A2) ! (FORALL [X:I] HYP
      (=
      (F ONE (S X))
      (S (S (F ONE X))))))

```

The Curious Inference of Boolos in Mizar and OMEGA

A3	(A3)	!	(FORALL [N:I,X:I] (= (F (S N) (S X)) (F N (F (S N) X))))		HYP
A4	(A4)	!	(D ONE)		HYP
A5	(A5)	!	(FORALL [X:I] (IMPLIES (D X) (D (S X))))		HYP
LD1	(LD1)	!	(=DEF LD1 ([Z]. (FORALL [X:(O I)] (IMPLIES (AND (X ONE) (FORALL [Y:I] (IMPLIES (X Y) (X (S Y)))))) (X Z))))		LOCAL-DEF
LD2	(LD2)	!	(=DEF LD2 ([Z].(AND (LD1 Z) (D Z))))		LOCAL-DEF
L2	(A1 A2 A3 A4 A5)	!	(FORALL [DC-13:(O I)] (IMPLIES (AND (DC-13 ONE) (FORALL [DC-17:I] (IMPLIES (DC-13 DC-17) (DC-13 (S DC-17)))))) (DC-13 ONE)))		OTTER: (NIL)
L1	(A1 A2 A3 A4 A5)	!	(LD1 ONE)	DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)	
L5	(A1 A2 A3 A4 A5)	!	(FORALL [DC-48:I] (IMPLIES (FORALL [DC-59:(O I)] (IMPLIES (AND (DC-59 ONE) (FORALL [DC-63:I] (IMPLIES (DC-59 DC-63) (DC-59 (S DC-63)))))) (DC-59 DC-48))) (FORALL [DC-68:(O I)] (IMPLIES (AND (DC-68 ONE) (FORALL [DC-72:I] (IMPLIES (DC-68 DC-72) (DC-68 (S DC-72)))))) (DC-68 (S DC-48))))))		OTTER: (NIL)
L4	(A1 A2 A3 A4 A5)	!	(FORALL [DC-26:I] (IMPLIES (FORALL [DC-37:(O I)] (IMPLIES (AND (DC-37 ONE) (FORALL [DC-41:I] (IMPLIES (DC-37 DC-41) (DC-37 (S DC-41)))))) (DC-37 DC-26))) (LD1 (S DC-26))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)	
L3	(A1 A2 A3 A4 A5)	!	(FORALL [Y:I] (IMPLIES (LD1 Y) (LD1 (S Y))))	DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)	
L6	(A1 A2 A3 A4 A5)	!	(LD1 (S (S (S ONE))))	OTTER: (NIL) (L1 L3)	
L8	(A1 A2 A3 A4 A5)	!	(AND (LD1 ONE) (D ONE))	OTTER: (NIL) (L1 A4)	
L7	(A1 A2 A3 A4 A5)	!	(LD2 ONE)	DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)	
L11	(A1 A2 A3 A4 A5)	!	(FORALL [DC-93:I] (IMPLIES (AND (LD1 DC-93) (D DC-93)) (AND (LD1 (S DC-93)) (D (S DC-93))))))	OTTER: (NIL) (L3 A5)	
L10	(A1 A2 A3 A4 A5)	!	(FORALL [DC-86:I] (IMPLIES	DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)	

```

                (AND (LD1 DC-86) (D DC-86))
                (LD2 (S DC-86))))
L9  (A1 A2 A3 ! (FORALL [Y:I]
A4 A5)         (IMPLIES (LD2 Y) (LD2 (S Y)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)

L12 (A1 A2 A3 ! (LD2 (S ONE))
A4 A5)
                                           OTTER: (NIL) (L7 L9)

L13 (A1 A2 A3 ! (FORALL [N:I]
A4 A5)         (IMPLIES
                (LD1 N)
                (FORALL [X:I]
                 (IMPLIES
                  (LD1 X)
                  (LD2 (F N X)))))) DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)

LD3 (LD3) ! (=DEF
            LD3
            ([N].
             (FORALL [X:I]
              (IMPLIES
               (LD1 X)
               (LD2 (F N X)))))) LOCAL-DEF

L18 (A1 A2 A3 ! (FORALL [DC-151:I]
A4 A5)         (IMPLIES
                (LD1 DC-151)
                (LD2 (F ONE DC-151)))) DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L19 LD4)

L15 (A1 A2 A3 ! (LD3 ONE)
A4 A5)
                                           DEFN-CONTRACT-LOCAL-DEF: ((0)) (L18 LD3)

L30 (L27 A1 ! (FORALL [DC-225:I]
A2 A3 A4      (IMPLIES
A5)           (LD1 DC-225)
              (LD2 (F (S N1) DC-225)))) DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L31 LD5)

L28 (L27 A1 ! (LD3 (S N1))
A2 A3 A4
A5)
                                           DEFN-CONTRACT-LOCAL-DEF: ((0)) (L30 LD3)

L26 (A1 A2 A3 ! (IMPLIES (LD3 N1) (LD3 (S N1)))
A4 A5)
                                           IMPI: (L28)

L16 (A1 A2 A3 ! (FORALL [N:I]
A4 A5)         (IMPLIES (LD3 N) (LD3 (S N)))) FORALLI: (N1) (L26)

L17 (A1 A2 A3 ! (FORALL [DC-123:I]
A4 A5)         (IMPLIES
                (FORALL [DC-134:(0 I)]
                 (IMPLIES
                  (AND
                   (DC-134 ONE)
                   (FORALL [DC-138:I]
                    (IMPLIES
                     (DC-134 DC-138)
                     (DC-134 (S DC-138))))))
                  (DC-134 DC-123)))
                (LD3 DC-123))) OTTER: (NIL) (L15 L16)

L14 (A1 A2 A3 ! (FORALL [N:I]
A4 A5)         (IMPLIES (LD1 N) (LD3 N))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L17 LD1)

LD4 (LD4) ! (=DEF LD4 ([X].(LD2 (F ONE X)))) LOCAL-DEF

L27 (L27) ! (LD3 N1)
                                           HYP

L29 (L27) ! (FORALL [DC-217:I]
            (IMPLIES
             (LD1 DC-217)
             (LD2 (F N1 DC-217)))) DEFN-EXPAND-LOCAL-DEF: ((0)) (L27 LD3)

L23 (A1 A2 A3 ! (LD2 (F ONE ONE))
A4 A5)
                                           OTTER: (NIL) (L12 A1)

L20 (A1 A2 A3 ! (LD4 ONE)
A4 A5)
                                           DEFN-CONTRACT-LOCAL-DEF: ((0)) (L23 LD4)

L25 (A1 A2 A3 ! (FORALL [DC-201:I]
A4 A5)         (IMPLIES
                (LD2 (F ONE DC-201))
                (LD2 (F ONE (S DC-201)))) OTTER: (NIL) (L9 A2)

L24 (A1 A2 A3 ! (FORALL [DC-194:I]
A4 A5)         (IMPLIES
                (LD2 (F ONE DC-194))
                (LD4 (S DC-194)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L25 LD4)

L21 (A1 A2 A3 ! (FORALL [X:I]
A4 A5)         (IMPLIES (LD4 X) (LD4 (S X)))) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L24 LD4)

```

The Curious Inference of Boolos in Mizar and OMEGA

```

L22 (A1 A2 A3 ! (FORALL [DC-165:I]
A4 A5) (IMPLIES
(FORALL [DC-176:(O I)]
(IMPLIES
(AND
(DC-176 ONE)
(FORALL [DC-180:I]
(IMPLIES
(DC-176 DC-180)
(DC-176 (S DC-180))))))
(DC-176 DC-165)))
(LD4 DC-165)))
OTTER: (NIL) (L20 L21)

L19 (A1 A2 A3 ! (FORALL [X:I]
A4 A5) (IMPLIES (LD1 X) (LD4 X)))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L22 LD1)

LD5 (LD5) ! (=DEF LD5 ([X].(LD2 (F (S N1) X))))
LOCAL-DEF
...

L32 (L27 A1 ! (LD5 ONE)
A2 A3 A4
A5)
OPEN
...

L33 (L27 A1 ! (FORALL [X:I]
A2 A3 A4 (IMPLIES (LD5 X) (LD5 (S X))))
A5)
OPEN
...

L34 (L27 A1 ! (FORALL [DC-239:I]
A2 A3 A4 (IMPLIES
A5) (FORALL [DC-250:(O I)]
(IMPLIES
(AND
(DC-250 ONE)
(FORALL [DC-254:I]
(IMPLIES
(DC-250 DC-254)
(DC-250 (S DC-254))))))
(DC-250 DC-239)))
(LD5 DC-239)))
OPEN

L31 (L27 A1 ! (FORALL [X:I]
A2 A3 A4 (IMPLIES (LD1 X) (LD5 X)))
A5)
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L34 LD1)
...

CONC (A1 A2 A3 ! (D
A4 A5) (F
(S (S (S (S ONE))))
(S (S (S (S ONE))))))
OPEN

;;; step 4.3.2.7
OMEGA: SUPPORT L34 (L32 L33)
...

L33 (L27 A1 ! (FORALL [X:I]
A2 A3 A4 (IMPLIES (LD5 X) (LD5 (S X))))
A5)
OPEN
...

L32 (L27 A1 ! (LD5 ONE)
A2 A3 A4
A5)
OPEN
...

L34 (L27 A1 ! (FORALL [DC-239:I]
A2 A3 A4 (IMPLIES
A5) (FORALL [DC-250:(O I)]
(IMPLIES
(AND
(DC-250 ONE)
(FORALL [DC-254:I]
(IMPLIES
(DC-250 DC-254)
(DC-250 (S DC-254))))))
(DC-250 DC-239)))
(LD5 DC-239)))
OPEN

;;; step 4.3.2.7
OMEGA: CALL-OTTER-ON-NODE L34 ...
...
----- PROOF -----
...

;;; step 4.3.2.7.1
OMEGA: DEFN-CONTRACT-LOCAL-DEF L32 () LD5 (O)
;;;CSM Arbitrary [2]: 0 provers have to be killed

```

```

;;; step 4.3.2.7.2-3
OMEGA: SUPPORT L35 (A1 L12)

A1 (A1)      ! (FORALL [N:I] (= (F N ONE) (S ONE)))           HYP
L12 (A1 A2 A3 ! (LD2 (S ONE))
     A4 A5)
...
L35 (L27 A1  ! (LD2 (F (S N1) ONE))
     A2 A3 A4
     A5)                                           OPEN

;;; step 4.3.2.7.2-3
OMEGA: CALL-OTTER-ON-NODE L35 ...
...
----- PROOF -----
...

;;; step 4.3.2.7.2.1
OMEGA: FORALLI

UNIV-LINE (NDLINE) A Universal line to prove: [L33]

PARAMETER (TERMSYM) New parameter: [x1]

LINE (NDLINE) A line: [()]
;;CSM Arbitrary [2]: 0 provers have to be killed

;;; step 4.3.2.7.2.1
OMEGA: IMPI

IMPLICATION (NDLINE) Implication to justify: [L36]
;;CSM Arbitrary [2]: 0 provers have to be killed

OMEGA: show-pds
A1 (A1)      ! (FORALL [N:I] (= (F N ONE) (S ONE)))           HYP
A2 (A2)      ! (FORALL [X:I]
               (=
                (F ONE (S X))
                (S (S (F ONE X)))))           HYP
A3 (A3)      ! (FORALL [N:I,X:I]
               (=
                (F (S N) (S X))
                (F N (F (S N) X))))           HYP
A4 (A4)      ! (D ONE)                                           HYP
A5 (A5)      ! (FORALL [X:I]
               (IMPLIES (D X) (D (S X))))           HYP
LD1 (LD1)    ! (=DEF
               LD1
               ([Z].
                (FORALL [X:(0 I)]
                 (IMPLIES
                  (AND
                   (X ONE)
                   (FORALL [Y:I]
                    (IMPLIES (X Y) (X (S Y))))))
                 (X Z))))           LOCAL-DEF
LD2 (LD2)    ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z))))           LOCAL-DEF
L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)]
     A4 A5)    (IMPLIES
               (AND
                (DC-13 ONE)
                (FORALL [DC-17:I]
                 (IMPLIES
                  (DC-13 DC-17)
                  (DC-13 (S DC-17))))))
               (DC-13 ONE)))           OTTER: (NIL)
L1 (A1 A2 A3 ! (LD1 ONE)
     A4 A5)    DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
L5 (A1 A2 A3 ! (FORALL [DC-48:I]
     A4 A5)    (IMPLIES
               (FORALL [DC-59:(0 I)]
                (IMPLIES
                 (AND
                  (DC-59 ONE)
                  (FORALL [DC-63:I]
                   (IMPLIES
                    (DC-59 DC-63)
                    (DC-59 (S DC-63))))))
                 (DC-59 DC-48))))           OTTER: (NIL)

```

The Curious Inference of Boolos in Mizar and OMEGA

```

(FORALL [DC-68:(0 I)]
(IMPLIES
(AND
(DC-68 ONE)
(FORALL [DC-72:I]
(IMPLIES
(DC-68 DC-72)
(DC-68 (S DC-72))))))
(DC-68 (S DC-48))))))

L4 (A1 A2 A3 ! (FORALL [DC-26:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
A4 A5) (IMPLIES
(FORALL [DC-37:(0 I)]
(IMPLIES
(AND
(DC-37 ONE)
(FORALL [DC-41:I]
(IMPLIES
(DC-37 DC-41)
(DC-37 (S DC-41))))))
(DC-37 DC-26)))
(LD1 (S DC-26))))

L3 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y))))

L6 (A1 A2 A3 ! (LD1 (S (S (S ONE)))) OTTER: (NIL) (L1 L3)
A4 A5)

L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE)) OTTER: (NIL) (L1 A4)
A4 A5)

L7 (A1 A2 A3 ! (LD2 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
A4 A5)

L11 (A1 A2 A3 ! (FORALL [DC-93:I] OTTER: (NIL) (L3 A5)
A4 A5) (IMPLIES
(AND (LD1 DC-93) (D DC-93))
(AND
(LD1 (S DC-93))
(D (S DC-93))))))

L10 (A1 A2 A3 ! (FORALL [DC-86:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
A4 A5) (IMPLIES
(AND (LD1 DC-86) (D DC-86))
(LD2 (S DC-86))))

L9 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y))))

L12 (A1 A2 A3 ! (LD2 (S ONE)) OTTER: (NIL) (L7 L9)
A4 A5)

L13 (A1 A2 A3 ! (FORALL [N:I] DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)
A4 A5) (IMPLIES
(LD1 N)
(FORALL [X:I]
(IMPLIES
(LD1 X)
(LD2 (F N X))))))

LD3 (LD3) ! (=DEF LOCAL-DEF
LD3
([N]
(FORALL [X:I]
(IMPLIES
(LD1 X)
(LD2 (F N X))))))

L18 (A1 A2 A3 ! (FORALL [DC-151:I] DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L19 LD4)
A4 A5) (IMPLIES
(LD1 DC-151)
(LD2 (F ONE DC-151))))

L15 (A1 A2 A3 ! (LD3 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L18 LD3)
A4 A5)

L30 (L27 A1 ! (FORALL [DC-225:I] DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L31 LD5)
A2 A3 A4 (IMPLIES
A5) (LD1 DC-225)
(LD2 (F (S N1) DC-225))))

L28 (L27 A1 ! (LD3 (S N1)) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L30 LD3)
A2 A3 A4
A5)

L26 (A1 A2 A3 ! (IMPLIES (LD3 N1) (LD3 (S N1))) IMPI: (L28)
A4 A5)

L16 (A1 A2 A3 ! (FORALL [N:I] FORALLI: (N1) (L26)
A4 A5) (IMPLIES (LD3 N) (LD3 (S N))))

```

```

L17 (A1 A2 A3 ! (FORALL [DC-123:I]
A4 A5) (IMPLIES
(FORALL [DC-134:(0 I)]
(IMPLIES
(AND
(DC-134 ONE)
(FORALL [DC-138:I]
(IMPLIES
(DC-134 DC-138)
(DC-134 (S DC-138))))))
(DC-134 DC-123)))
(LD3 DC-123)))
OTTER: (NIL) (L15 L16)

L14 (A1 A2 A3 ! (FORALL [N:I]
A4 A5) (IMPLIES (LD1 N) (LD3 N)))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L17 LD1)

LD4 (LD4) ! (=DEF LD4 ([X].(LD2 (F ONE X))))
LOCAL-DEF

L27 (L27) ! (LD3 N1)
HYP

L29 (L27) ! (FORALL [DC-217:I]
(IMPLIES
(LD1 DC-217)
(LD2 (F N1 DC-217))))
DEFN-EXPAND-LOCAL-DEF: ((0)) (L27 LD3)

L23 (A1 A2 A3 ! (LD2 (F ONE ONE))
A4 A5)
OTTER: (NIL) (L12 A1)

L20 (A1 A2 A3 ! (LD4 ONE)
A4 A5)
DEFN-CONTRACT-LOCAL-DEF: ((0)) (L23 LD4)

L25 (A1 A2 A3 ! (FORALL [DC-201:I]
A4 A5) (IMPLIES
(LD2 (F ONE DC-201))
(LD2 (F ONE (S DC-201))))))
OTTER: (NIL) (L9 A2)

L24 (A1 A2 A3 ! (FORALL [DC-194:I]
A4 A5) (IMPLIES
(LD2 (F ONE DC-194))
(LD4 (S DC-194))))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L25 LD4)

L21 (A1 A2 A3 ! (FORALL [X:I]
A4 A5) (IMPLIES (LD4 X) (LD4 (S X))))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L24 LD4)

L22 (A1 A2 A3 ! (FORALL [DC-165:I]
A4 A5) (IMPLIES
(FORALL [DC-176:(0 I)]
(IMPLIES
(AND
(DC-176 ONE)
(FORALL [DC-180:I]
(IMPLIES
(DC-176 DC-180)
(DC-176 (S DC-180))))))
(DC-176 DC-165)))
(LD4 DC-165)))
OTTER: (NIL) (L20 L21)

L19 (A1 A2 A3 ! (FORALL [X:I]
A4 A5) (IMPLIES (LD1 X) (LD4 X)))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L22 LD1)

LD5 (LD5) ! (=DEF LD5 ([X].(LD2 (F (S N1) X))))
LOCAL-DEF

L37 (L37) ! (LD5 X1)
HYP

L35 (L27 A1 ! (LD2 (F (S N1) ONE))
A2 A3 A4
A5)
OTTER: (NIL) (L12 A1)

L32 (L27 A1 ! (LD5 ONE)
A2 A3 A4
A5)
DEFN-CONTRACT-LOCAL-DEF: ((0)) (L35 LD5)
...

L38 (L37 L27 ! (LD5 (S X1))
A1 A2 A3
A4 A5)
OPEN

L36 (L27 A1 ! (IMPLIES (LD5 X1) (LD5 (S X1)))
A2 A3 A4
A5)
IMPI: (L38)

L33 (L27 A1 ! (FORALL [X:I]
A2 A3 A4 (IMPLIES (LD5 X) (LD5 (S X))))
A5)
FORALLI: (X1) (L36)

L34 (L27 A1 ! (FORALL [DC-239:I]
A2 A3 A4 (IMPLIES
A5) (FORALL [DC-250:(0 I)]
(IMPLIES
(AND

```



The Curious Inference of Boolos in Mizar and OMEGA

```

(DC-250 ONE)
(FORALL [DC-254:I]
 (IMPLIES
 (DC-250 DC-254)
 (DC-250 (S DC-254))))))
(DC-250 DC-239)))
(LD5 DC-239)))

L31 (L27 A1 ! (FORALL [X:I]
A2 A3 A4 (IMPLIES (LD1 X) (LD5 X)))
A5)
...
CONC (A1 A2 A3 ! (D
A4 A5) (F
(S (S (S (S ONE))))
(S (S (S (S ONE))))))
OPEN

;;; step 4.3.2.7.2.2
OMEGA: DEFN-EXPAND-LOCAL-DEF () L37 LD5 (0)
;;;CSM Arbitrary [2]: 0 provers have to be killed

;;; step 4.3.2.7.2.3
OMEGA: DEFN-EXPAND-LOCAL-DEF () L39 LD2 (0)
;;;CSM Arbitrary [2]: 0 provers have to be killed

;;; step 4.3.2.7.2.3
OMEGA: ANDE

CONJUNCTION (NDLINE) Conjunction to split: [L40]
LCONJ (NDLINE) Left conjunct: [()]
RCONJ (NDLINE) Right conjunct: [()]
;;;CSM Arbitrary [2]: 0 provers have to be killed

;;; step 4.3.2.7.2.4
OMEGA: DEFN-CONTRACT-LOCAL-DEF L38 () LD5 (0)
;;;CSM Arbitrary [2]: 0 provers have to be killed

;;; step 4.3.2.7.2.5
OMEGA: LEMMA L43 (LD2 (F N1 (F (S N1) X1)))

;;; step 4.3.2.7.2.5
OMEGA: SUPPORT L44 (L41 L29)

L41 (L37) ! (LD1 (F (S N1) X1))
ANDE: (L40)

L29 (L27) ! (FORALL [DC-217:I]
(IMPLIES
(LD1 DC-217)
(LD2 (F N1 DC-217))))
...
L44 (L37 L27 ! (LD2 (F N1 (F (S N1) X1)))
A1 A2 A3
A4 A5)
OPEN

;;; step 4.3.2.7.2.5
OMEGA: CALL-UTTER-ON-NODE L44 ...
...
----- PROOF -----
...

OMEGA: show-pds
A1 (A1) ! (FORALL [N:I] (= (F N ONE) (S ONE)))
HYP
A2 (A2) ! (FORALL [X:I]
(=
(F ONE (S X))
(S (S (F ONE X)))))
HYP
A3 (A3) ! (FORALL [N:I,X:I]
(=
(F (S N) (S X))
(F N (F (S N) X))))
HYP
A4 (A4) ! (D ONE)
HYP
A5 (A5) ! (FORALL [X:I]
(IMPLIES (D X) (D (S X))))
HYP
LD1 (LD1) ! (=DEF
LD1
([Z].
(FORALL [X:(0 I)]
(IMPLIES
(AND
(X ONE)
(FORALL [Y:I]

```

```

      (IMPLIES (X Y) (X (S Y))))
      (X Z))))))
LD2 (LD2) ! (=DEF LD2 ([Z].(AND (LD1 Z) (D Z)))) LOCAL-DEF
L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)] OTTER: (NIL)
   A4 A5) (IMPLIES
           (AND
            (DC-13 ONE)
            (FORALL [DC-17:I]
             (IMPLIES
              (DC-13 DC-17)
              (DC-13 (S DC-17))))))
          (DC-13 ONE)))
L1 (A1 A2 A3 ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
   A4 A5)
L5 (A1 A2 A3 ! (FORALL [DC-48:I] OTTER: (NIL)
   A4 A5) (IMPLIES
           (FORALL [DC-59:(0 I)]
            (IMPLIES
             (AND
              (DC-59 ONE)
              (FORALL [DC-63:I]
               (IMPLIES
                (DC-59 DC-63)
                (DC-59 (S DC-63))))))
             (DC-59 DC-48)))
          (FORALL [DC-68:(0 I)]
           (IMPLIES
            (AND
             (DC-68 ONE)
             (FORALL [DC-72:I]
              (IMPLIES
               (DC-68 DC-72)
               (DC-68 (S DC-72))))))
            (DC-68 (S DC-48))))))
L4 (A1 A2 A3 ! (FORALL [DC-26:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
   A4 A5) (IMPLIES
           (FORALL [DC-37:(0 I)]
            (IMPLIES
             (AND
              (DC-37 ONE)
              (FORALL [DC-41:I]
               (IMPLIES
                (DC-37 DC-41)
                (DC-37 (S DC-41))))))
             (DC-37 DC-26)))
          (LD1 (S DC-26))))
L3 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
   A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y))))
L6 (A1 A2 A3 ! (LD1 (S (S (S (S ONE)))) OTTER: (NIL) (L1 L3)
   A4 A5)
L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE)) OTTER: (NIL) (L1 A4)
   A4 A5)
L7 (A1 A2 A3 ! (LD2 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
   A4 A5)
L11 (A1 A2 A3 ! (FORALL [DC-93:I] OTTER: (NIL) (L3 A5)
   A4 A5) (IMPLIES
           (AND (LD1 DC-93) (D DC-93))
          (AND
           (LD1 (S DC-93))
           (D (S DC-93))))))
L10 (A1 A2 A3 ! (FORALL [DC-86:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
   A4 A5) (IMPLIES
           (AND (LD1 DC-86) (D DC-86))
          (LD2 (S DC-86))))
L9 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
   A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y))))
L12 (A1 A2 A3 ! (LD2 (S ONE)) OTTER: (NIL) (L7 L9)
   A4 A5)
L13 (A1 A2 A3 ! (FORALL [N:I] DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)
   A4 A5) (IMPLIES
           (LD1 N)
          (FORALL [X:I]
           (IMPLIES
            (LD1 X)
            (LD2 (F N X))))))
LD3 (LD3) ! (=DEF LOCAL-DEF

```

The Curious Inference of Boolos in Mizar and OMEGA

```

LD3
([N].
 (FORALL [X:I]
  (IMPLIES
   (LD1 X)
   (LD2 (F N X))))))
L18 (A1 A2 A3 ! (FORALL [DC-151:I]
A4 A5)          (IMPLIES
                (LD1 DC-151)
                (LD2 (F ONE DC-151))))
DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L19 LD4)
L15 (A1 A2 A3 ! (LD3 ONE)
A4 A5)          DEFN-CONTRACT-LOCAL-DEF: ((0)) (L18 LD3)
L30 (L27 A1 ! (FORALL [DC-225:I]
A2 A3 A4)      (IMPLIES
A5)            (LD1 DC-225)
              (LD2 (F (S N1) DC-225))))
DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L31 LD5)
L28 (L27 A1 ! (LD3 (S N1))
A2 A3 A4)      DEFN-CONTRACT-LOCAL-DEF: ((0)) (L30 LD3)
A5)
L26 (A1 A2 A3 ! (IMPLIES (LD3 N1) (LD3 (S N1)))
A4 A5)          IMP1: (L28)
L16 (A1 A2 A3 ! (FORALL [N:I]
A4 A5)          (IMPLIES (LD3 N) (LD3 (S N))))
FORALL1: (N1) (L26)
L17 (A1 A2 A3 ! (FORALL [DC-123:I]
A4 A5)          (IMPLIES
                (FORALL [DC-134:(0 I)]
                 (IMPLIES
                  (AND
                   (DC-134 ONE)
                   (FORALL [DC-138:I]
                    (IMPLIES
                     (DC-134 DC-138)
                     (DC-134 (S DC-138))))))
                 (DC-134 DC-123))))
                (LD3 DC-123)))
OTTER: (NIL) (L15 L16)
L14 (A1 A2 A3 ! (FORALL [N:I]
A4 A5)          (IMPLIES (LD1 N) (LD3 N)))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L17 LD1)
LD4 (LD4) ! (=DEF LD4 ([X].(LD2 (F ONE X))))
LOCAL-DEF
L27 (L27) ! (LD3 N1)
HYP
L29 (L27) ! (FORALL [DC-217:I]
            (IMPLIES
             (LD1 DC-217)
             (LD2 (F N1 DC-217))))
DEFN-EXPAND-LOCAL-DEF: ((0)) (L27 LD3)
L23 (A1 A2 A3 ! (LD2 (F ONE ONE))
A4 A5)          OTTER: (NIL) (L12 A1)
L20 (A1 A2 A3 ! (LD4 ONE)
A4 A5)          DEFN-CONTRACT-LOCAL-DEF: ((0)) (L23 LD4)
L25 (A1 A2 A3 ! (FORALL [DC-201:I]
A4 A5)          (IMPLIES
                (LD2 (F ONE DC-201))
                (LD2 (F ONE (S DC-201))))))
OTTER: (NIL) (L9 A2)
L24 (A1 A2 A3 ! (FORALL [DC-194:I]
A4 A5)          (IMPLIES
                (LD2 (F ONE DC-194))
                (LD4 (S DC-194))))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L25 LD4)
L21 (A1 A2 A3 ! (FORALL [X:I]
A4 A5)          (IMPLIES (LD4 X) (LD4 (S X))))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L24 LD4)
L22 (A1 A2 A3 ! (FORALL [DC-165:I]
A4 A5)          (IMPLIES
                (FORALL [DC-176:(0 I)]
                 (IMPLIES
                  (AND
                   (DC-176 ONE)
                   (FORALL [DC-180:I]
                    (IMPLIES
                     (DC-176 DC-180)
                     (DC-176 (S DC-180))))))
                 (DC-176 DC-165))))
                (LD4 DC-165)))
OTTER: (NIL) (L20 L21)
L19 (A1 A2 A3 ! (FORALL [X:I]
A4 A5)          (IMPLIES (LD1 X) (LD4 X)))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L22 LD1)
LD5 (LD5) ! (=DEF LD5 ([X].(LD2 (F (S N1) X))))
LOCAL-DEF

```

```

L37 (L37) ! (LD5 X1) HYP
L39 (L37) ! (LD2 (F (S N1) X1)) DEFN-EXPAND-LOCAL-DEF: ((0)) (L37 LD5)
L40 (L37) ! (AND DEFN-EXPAND-LOCAL-DEF: ((0)) (L39 LD2)
  (LD1 (F (S N1) X1))
  (D (F (S N1) X1)))
L42 (L37) ! (D (F (S N1) X1)) ANDE: (L40)
L41 (L37) ! (LD1 (F (S N1) X1)) ANDE: (L40)
L35 (L27 A1 ! (LD2 (F (S N1) ONE)) OTTER: (NIL) (L12 A1)
  A2 A3 A4
  A5)
L32 (L27 A1 ! (LD5 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L35 LD5)
  A2 A3 A4
  A5)
L44 (L37 L27 ! (LD2 (F N1 (F (S N1) X1))) OTTER: (NIL) (L29 L41)
  A1 A2 A3
  A4 A5)
  ...
L43 (L37 L27 ! (LD2 (F (S N1) (S X1))) OPEN
  A1 A2 A3
  A4 A5)
L38 (L37 L27 ! (LD5 (S X1)) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L43 LD5)
  A1 A2 A3
  A4 A5)
L36 (L27 A1 ! (IMPLIES (LD5 X1) (LD5 (S X1))) IMPI: (L38)
  A2 A3 A4
  A5)
L33 (L27 A1 ! (FORALL [X:I] FORALLI: (X1) (L36)
  A2 A3 A4 ! (IMPLIES (LD5 X) (LD5 (S X))))
  A5)
L34 (L27 A1 ! (FORALL [DC-239:I] OTTER: (NIL) (L32 L33)
  A2 A3 A4 ! (IMPLIES
  A5) (FORALL [DC-250:(0 I)]
  (IMPLIES
  (AND
  (DC-250 ONE)
  (FORALL [DC-254:I]
  (IMPLIES
  (DC-250 DC-254)
  (DC-250 (S DC-254))))))
  (DC-250 DC-239)))
  (LD5 DC-239)))
L31 (L27 A1 ! (FORALL [X:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L34 LD1)
  A2 A3 A4 ! (IMPLIES (LD1 X) (LD5 X)))
  A5)
  ...
CONC (A1 A2 A3 ! (D OPEN
  A4 A5) (F
  (S (S (S (S ONE))))
  (S (S (S (S ONE))))))

;; step 4.3.2.7.2.6
OMEGA: SUPPORT L43 (L44 A3)

L44 (L37 L27 ! (LD2 (F N1 (F (S N1) X1))) OTTER: (NIL) (L29 L41)
  A1 A2 A3
  A4 A5)
A3 (A3) ! (FORALL [N:I,X:I] HYP
  (=
  (F (S N) (S X))
  (F N (F (S N) X))))
  ...
L43 (L37 L27 ! (LD2 (F (S N1) (S X1))) OPEN
  A1 A2 A3
  A4 A5)

;; step 4.3.2.7.2.7
OMEGA: CALL-OTTER-ON-NODE L43 ...
...
----- PROOF -----
....

;; step 6
OMEGA: LEMMA CONC (ld2 (F (S (S (S (S ONE)))) (S (S (S (S ONE))))))

```

The Curious Inference of Boolos in Mizar and OMEGA

```

;;; step 6
OMEGA: SUPPORT L45 (L6 L13)

L13 (A1 A2 A3 ! (FORALL [N:I]
  A4 A5)      (IMPLIES
              (LD1 N)
              (FORALL [X:I]
                (IMPLIES
                 (LD1 X)
                 (LD2 (F N X))))))
DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)

L6 (A1 A2 A3 ! (LD1 (S (S (S ONE))))
  A4 A5)
OTTER: (NIL) (L1 L3)
...

L45 (A1 A2 A3 ! (LD2
  A4 A5)      (F
              (S (S (S ONE)))
              (S (S (S ONE))))))
OPEN

OMEGA: CALL-OTTER-ON-NODE L45 ...
...
----- PROOF -----
...

;;; step 6
OMEGA: DEFN-EXPAND-LOCAL-DEF () L45 LD2 (0)
;;;CSM Arbitrary [2]: 0 provers have to be killed

;;; step 7
OMEGA: SUPPORT CONC (L46)

L46 (A1 A2 A3 ! (AND
  A4 A5)      (LD1
              (F
               (S (S (S ONE)))
               (S (S (S ONE))))))
              (D
               (F
                (S (S (S ONE)))
                (S (S (S ONE))))))
              ...
CONC (A1 A2 A3 ! (D
  A4 A5)      (F
              (S (S (S ONE)))
              (S (S (S ONE))))))
OPEN

;;; step 7
OMEGA: CALL-OTTER-ON-NODE ...
...
----- PROOF -----
...

OMEGA: show-pds
A1 (A1)      ! (FORALL [N:I] (= (F N ONE) (S ONE)))
HYP
A2 (A2)      ! (FORALL [X:I]
              (=
               (F ONE (S X))
               (S (S (F ONE X)))))
HYP
A3 (A3)      ! (FORALL [N:I,X:I]
              (=
               (F (S N) (S X))
               (F N (F (S N) X))))
HYP
A4 (A4)      ! (D ONE)
HYP
A5 (A5)      ! (FORALL [X:I]
              (IMPLIES (D X) (D (S X))))
HYP
LD1 (LD1)    ! (=DEF
              LD1
              ([2].
               (FORALL [X:(0 I)]
                 (IMPLIES
                  (AND
                   (X ONE)
                   (FORALL [Y:I]
                     (IMPLIES (X Y) (X (S Y))))))
                 (X Z))))))
LOCAL-DEF
LD2 (LD2)    ! (=DEF LD2 ([2].(AND (LD1 Z) (D Z))))
LOCAL-DEF
L2 (A1 A2 A3 ! (FORALL [DC-13:(0 I)]
  A4 A5)      (IMPLIES
              (AND
               (S (S (S ONE)))
               (S (S (S ONE))))))
OTTER: (NIL)

```

```

(DC-13 ONE)
(FORALL [DC-17:I]
  (IMPLIES
    (DC-13 DC-17)
    (DC-13 (S DC-17))))))
(DC-13 ONE)))

L1 (A1 A2 A3 ! (LD1 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L2 LD1)
A4 A5)

L5 (A1 A2 A3 ! (FORALL [DC-48:I] OTTER: (NIL)
A4 A5) (IMPLIES
  (FORALL [DC-59:(0 I)]
    (IMPLIES
      (AND
        (DC-59 ONE)
        (FORALL [DC-63:I]
          (IMPLIES
            (DC-59 DC-63)
            (DC-59 (S DC-63))))))
        (DC-59 DC-48)))
    (FORALL [DC-68:(0 I)]
      (IMPLIES
        (AND
          (DC-68 ONE)
          (FORALL [DC-72:I]
            (IMPLIES
              (DC-68 DC-72)
              (DC-68 (S DC-72))))))
          (DC-68 (S DC-48))))))

L4 (A1 A2 A3 ! (FORALL [DC-26:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L5 LD1)
A4 A5) (IMPLIES
  (FORALL [DC-37:(0 I)]
    (IMPLIES
      (AND
        (DC-37 ONE)
        (FORALL [DC-41:I]
          (IMPLIES
            (DC-37 DC-41)
            (DC-37 (S DC-41))))))
        (DC-37 DC-26)))
    (LD1 (S DC-26))))

L3 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L4 LD1)
A4 A5) (IMPLIES (LD1 Y) (LD1 (S Y))))

L6 (A1 A2 A3 ! (LD1 (S (S (S ONE)))) OTTER: (NIL) (L1 L3)
A4 A5)

L8 (A1 A2 A3 ! (AND (LD1 ONE) (D ONE)) OTTER: (NIL) (L1 A4)
A4 A5)

L7 (A1 A2 A3 ! (LD2 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L8 LD2)
A4 A5)

L11 (A1 A2 A3 ! (FORALL [DC-93:I] OTTER: (NIL) (L3 A5)
A4 A5) (IMPLIES
  (AND (LD1 DC-93) (D DC-93))
  (AND
    (LD1 (S DC-93))
    (D (S DC-93))))))

L10 (A1 A2 A3 ! (FORALL [DC-86:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L11 LD2)
A4 A5) (IMPLIES
  (AND (LD1 DC-86) (D DC-86))
  (LD2 (S DC-86))))

L9 (A1 A2 A3 ! (FORALL [Y:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L10 LD2)
A4 A5) (IMPLIES (LD2 Y) (LD2 (S Y))))

L12 (A1 A2 A3 ! (LD2 (S ONE)) OTTER: (NIL) (L7 L9)
A4 A5)

L13 (A1 A2 A3 ! (FORALL [N:I] DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L14 LD3)
A4 A5) (IMPLIES
  (LD1 N)
  (FORALL [X:I]
    (IMPLIES
      (LD1 X)
      (LD2 (F N X))))))

LD3 (LD3) ! (=DEF LOCAL-DEF
LD3
{[N].
(FORALL [X:I]
  (IMPLIES
    (LD1 X)
    (LD2 (F N X))))))

L18 (A1 A2 A3 ! (FORALL [DC-151:I] DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L19 LD4)

```

The Curious Inference of Boolos in Mizar and OMEGA

```

A4 A5) (IMPLIES
(LD1 DC-151)
(LD2 (F ONE DC-151))))

L15 (A1 A2 A3 ! (LD3 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0) (L18 LD3)
A4 A5)

L30 (L27 A1 ! (FORALL [DC-225:I] DEFN-EXPAND-LOCAL-DEF: ((1 0 2 0)) (L31 LD5)
A2 A3 A4 (IMPLIES
A5) (LD1 DC-225)
(LD2 (F (S N1) DC-225))))

L28 (L27 A1 ! (LD3 (S N1)) DEFN-CONTRACT-LOCAL-DEF: ((0) (L30 LD3)
A2 A3 A4
A5)

L26 (A1 A2 A3 ! (IMPLIES (LD3 N1) (LD3 (S N1))) IMPI: (L28)
A4 A5)

L16 (A1 A2 A3 ! (FORALL [N:I] FORALLI: (N1) (L26)
A4 A5) (IMPLIES (LD3 N) (LD3 (S N))))

L17 (A1 A2 A3 ! (FORALL [DC-123:I] OTTER: (NIL) (L15 L16)
A4 A5) (IMPLIES
(FORALL [DC-134:(O I)])
(IMPLIES
(AND
(DC-134 ONE)
(FORALL [DC-138:I]
(IMPLIES
(DC-134 DC-138)
(DC-134 (S DC-138))))))
(DC-134 DC-123)))
(LD3 DC-123)))

L14 (A1 A2 A3 ! (FORALL [N:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L17 LD1)
A4 A5) (IMPLIES (LD1 N) (LD3 N)))

LD4 (LD4) ! (=DEF LD4 ([X].(LD2 (F ONE X)))) LOCAL-DEF

L27 (L27) ! (LD3 N1) HYP

L29 (L27) ! (FORALL [DC-217:I] DEFN-EXPAND-LOCAL-DEF: ((0) (L27 LD3)
(IMPLIES
(LD1 DC-217)
(LD2 (F N1 DC-217))))

L23 (A1 A2 A3 ! (LD2 (F ONE ONE)) OTTER: (NIL) (L12 A1)
A4 A5)

L20 (A1 A2 A3 ! (LD4 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0) (L23 LD4)
A4 A5)

L25 (A1 A2 A3 ! (FORALL [DC-201:I] OTTER: (NIL) (L9 A2)
A4 A5) (IMPLIES
(LD2 (F ONE DC-201))
(LD2 (F ONE (S DC-201))))))

L24 (A1 A2 A3 ! (FORALL [DC-194:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 2 0)) (L25 LD4)
A4 A5) (IMPLIES
(LD2 (F ONE DC-194))
(LD4 (S DC-194))))

L21 (A1 A2 A3 ! (FORALL [X:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L24 LD4)
A4 A5) (IMPLIES (LD4 X) (LD4 (S X))))

L22 (A1 A2 A3 ! (FORALL [DC-165:I] OTTER: (NIL) (L20 L21)
A4 A5) (IMPLIES
(FORALL [DC-176:(O I)])
(IMPLIES
(AND
(DC-176 ONE)
(FORALL [DC-180:I]
(IMPLIES
(DC-176 DC-180)
(DC-176 (S DC-180))))))
(DC-176 DC-165)))
(LD4 DC-165)))

L19 (A1 A2 A3 ! (FORALL [X:I] DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L22 LD1)
A4 A5) (IMPLIES (LD1 X) (LD4 X)))

LD5 (LD5) ! (=DEF LD5 ([X].(LD2 (F (S N1) X)))) LOCAL-DEF

L37 (L37) ! (LD5 X1) HYP

L39 (L37) ! (LD2 (F (S N1) X1)) DEFN-EXPAND-LOCAL-DEF: ((0) (L37 LD5)

L40 (L37) ! (AND DEFN-EXPAND-LOCAL-DEF: ((0) (L39 LD2)
(LD1 (F (S N1) X1))
(D (F (S N1) X1))))

```

```

L45 (A1 A2 A3 ! (LD2
A4 A5) (F
(S (S (S (S ONE))))
(S (S (S (S ONE)))))) OTTER: (NIL) (L6 L13)

L46 (A1 A2 A3 ! (AND
A4 A5) (LD1
(F
(S (S (S (S ONE))))
(S (S (S (S ONE))))))
(D
(F
(S (S (S (S ONE))))
(S (S (S (S ONE))))))
DEFN-EXPAND-LOCAL-DEF: ((0)) (L45 LD2)

L42 (L37) ! (D (F (S N1) X1)) ANDE: (L40)
L41 (L37) ! (LD1 (F (S N1) X1)) ANDE: (L40)
L35 (L27 A1 ! (LD2 (F (S N1) ONE)) OTTER: (NIL) (L12 A1)
A2 A3 A4
A5)

L32 (L27 A1 ! (LD5 ONE) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L35 LD5)
A2 A3 A4
A5)

L44 (L37 L27 ! (LD2 (F N1 (F (S N1) X1))) OTTER: (NIL) (L29 L41)
A1 A2 A3
A4 A5)

L43 (L37 L27 ! (LD2 (F (S N1) (S X1))) OTTER: (NIL) (A3 L44)
A1 A2 A3
A4 A5)

L38 (L37 L27 ! (LD5 (S X1)) DEFN-CONTRACT-LOCAL-DEF: ((0)) (L43 LD5)
A1 A2 A3
A4 A5)

L36 (L27 A1 ! (IMPLIES (LD5 X1) (LD5 (S X1))) IMPI: (L38)
A2 A3 A4
A5)

L33 (L27 A1 ! (FORALL [X:1]
A2 A3 A4 (IMPLIES (LD5 X) (LD5 (S X))))
A5) FORALLI: (X1) (L36)

L34 (L27 A1 ! (FORALL [DC-239:1]
A2 A3 A4 (IMPLIES
A5) (FORALL [DC-250:(0 I)]
(IMPLIES
(AND
(DC-250 ONE)
(FORALL [DC-254:I]
(IMPLIES
(DC-250 DC-254)
(DC-250 (S DC-254))))))
(DC-250 DC-239)))
(LD5 DC-239)))
DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L34 LD1)

L31 (L27 A1 ! (FORALL [X:1]
A2 A3 A4 (IMPLIES (LD1 X) (LD5 X)))
A5) DEFN-CONTRACT-LOCAL-DEF: ((1 0 1 0)) (L34 LD1)

CONC (A1 A2 A3 ! (D
A4 A5) (F
(S (S (S (S ONE))))
(S (S (S (S ONE)))))) OTTER: (NIL) (L46)

```

OMEGA: