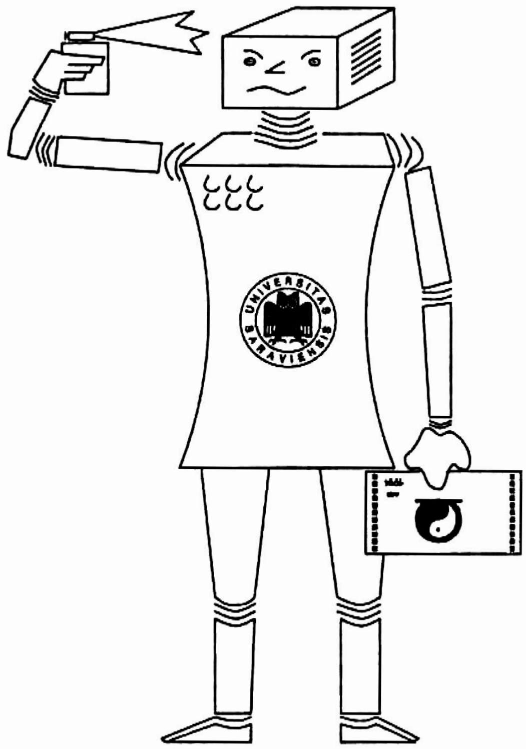
UNIVERSITÄT DES SAARLANDES
FACHBEREICH INFORMATIK
D-66041 SAARBRÜCKEN
GERMANY
WWW: http://jswww.cs.uni-sb.de/pub/www/

SEKI Report



The Mechanization of the Diagonalization Proof Strategy

Lassaad Cheikhrouhou

SEKI Report SR-96-14

The Mechanization of the Diagonalization Proof Strategy

Lassaad Cheikhrouhou
Fachbereich Informatik
Universität des Saarlandes
D-66041 Saarbrücken, Germany
lassaad@cs.uni-sb.de
http://jswww.cs.uni-sb.de/~lassaad

Abstract

We present an empirical study of mathematical proofs by diagonalization, the aim is their mechanization based on proof planning techniques. We show that these proofs can be constructed according to a strategy that (i) finds an indexing relation, (ii) constructs a diagonal element, and (iii) makes the implicit contradiction of the diagonal element explicit. Moreover we suggest how diagonal elements can be represented.

1 Introduction

In classical (automated) theorem proving the reasoning process is carried out at the object level, i.e. the level of the (first order) logic representation of the mathematical objects under study. Searching for a proof means applying calculus inference rules to manipulate the initial problem situation which at the beginning consists of the negated theorem to be proved and the given assertions (definitions, axioms, and other theorems) in order to find a final situation, for instance $\perp$. This guarantees that the theorem is a logical consequence of the given assertions. Tactical theorem proving applies tactics, i.e. composition of calculus inference rules. The reasoning remains however at the object level.

Proof planning [Bun91] is the search for a sequence of tactics (a proof plan) which can be applied to construct an object level proof. The used operators (methods) are specifications of tactics represented in a meta-language. They state in this meta-language when a tactic can be applied and what its effects are. Reasoning is therefore carried out at a meta level. The two main aspects, that make this approach interesting, can be demonstrated for inductive theorem proving in CLAM [BvHHS90] as follows.

The first aspect of proof planning is that the search for a proof plan is often done in the context of a well known mathematical proof technique such as induction or diagonalization. Such a proof technique characterizes a whole proof schema which is then instantiated to a sequence of planning steps (which in turn generate object level proofs). Similar to specifications of basic tactics, these proof schemata are called (proof) methods in the terminology of CLAM. As a mathematical proof technique implicitly comprises instructions on how to globally perform the associated part of a proof, we want to extend the proof schema in the representation of a technique with additional knowledge which expresses such instructions. In our approach we call these structures for the representation of mathematical proof techniques *proof strategies*, whereas specifications of basic tactics which correspond to ground proof plan steps are called proof methods as in CLAM. For instance an induction proof strategy consists of

- the induction method which computes an induction schema and reduces the theorem to the well-known subgoals of the base and the step case,
- some basic methods, for instance the symbolic evaluation method, and eventually the induction proof strategy to prove the subgoals of the base case,
- the rippling proof strategy to rewrite the subgoal of a step case, which corresponds to an induction conclusion, so that an induction hypothesis can be used to close this proof path. Wave methods choose the appropriate rewriting rules.
- and some basic methods, for instance the fertilize method, to close the proof path of the step case employing induction hypothesis.

The second aspect of proof planning is the abstraction from mere logical manipulation of formulae by calculus inference rules. For instance the task of proving an induction conclusion in CLAM is treated as reducing the syntactical differences to an induction hypothesis by the rippling proof strategy with the intention of employing this to close the proof path.

The point of proof planning is to analyze proof techniques in order to determine their typical proof steps and to find a suitable control to perform these steps within the proof planning process. In this report we present some properties of the diagonalization strategy which we noticed from an empirical study of several well-known proofs, that are based on the diagonalization principle. We give the essential proof steps of the diagonalization technique and suggest how to implement these steps in a proof planning environment.

2 Cantor Diagonalization

In order to show the main principles of the diagonalization technique, consider the Cantor theorem. This is where the diagonalization technique was first invented and it is therefore often called Cantor diagonalization [Kle43]. This theorem states that the power set of each set M has greater cardinality than the set itself, which is equivalent to the conjecture that there is no surjective function from the set into its power set:

$$\forall M. \neg\exists f. \mathrm{surj}(f, M, 2^M)$$

To prove the above conjecture, we assume that there is a surjective function f_0 from some set M_0 into its power set 2^{M_0} and deduce a contradiction by diagonalization. In [DSW94] a proof by diagonalization is described as follows:

> *The diagonalization method turns on the demonstration of two assertions of the following sort:*
>
> 1. *A certain set A can be enumerated in a suitable fashion.*
> 2. *It is possible, with the help of the enumeration, to define an object b that is different from every object in the enumeration, i.e. $b \notin A$.*

Below is the diagonalization part of the Cantor proof, where 2^{M_0} is *the enumerable set*. This set can be enumerated with the help of *the indexing relation* f_0 and *the diagonal element* D is the object which is defined with the help of the enumeration. It is different from every object $f_0(x)$ in the enumeration:

> *The set* $D = \{x \in M_0 | x \notin f_0(x)\}$ *belongs to* 2^{M_0}*, there is also an element* y_0 *of* M_0 *which is the index of* D *in* M_0 *(*$D = f_0(y_0)$ *with* $y_0 \in M_0$*). By the definition of* D y_0 *belongs to* D *iff* y_0 *is in* M_0 *and does not belong to* $f_0(y_0)$*. This is obviously a contradiction to* $D = f_0(y_0)$*.*

In order to formulate the characteristic proof steps of the above diagonalization proof, we consider the formal proof in Figure 1 of the Cantor theorem which was interactively constructed in the Ω-MKRP environment [HKK⁺94] using the problem description in Table 1 [1]. This proof was interactively constructed at the level of the natural deduction (ND) calculus, i.e. was generated by the application of ND rules [Gen35]. It is then abstracted to the so-called assertion level [Hua94], where assertions, in addition to ND rules, can be used as justifications.

TND	$\forall x_{o}.\, x \lor \neg x$
=-Refl	$\forall x_{o}.\, x = x$
=-Equiv	$\forall x_{o}.\, \forall y_{o}.\, x = y \to [x \leftrightarrow y]$
Surj-Def	$\forall f_{\iota\to(\iota\to o)}.\, \forall a_{\iota\to o}.\, \forall b_{(\iota\to o)\to o}.\, \mathrm{surj}(f,a,b) \leftrightarrow$ $\forall x_{\iota\to o}.\, x \in b \to \exists y_{\iota}.\, y \in a \land x = f(y)$
PSet-Def	$\forall a_{\iota\to o}.\, \forall x_{\iota\to o}.\, x \in P(a) \leftrightarrow x \subseteq a$
⊆-Def	$\forall a_{\iota\to o}.\, \forall b_{\iota\to o}.\, a \subseteq b \leftrightarrow \forall x_{\iota}.\, x \in a \to x \in b$
Powerset	$\forall M_{\iota\to o}.\, \neg\exists f_{\iota\to(\iota\to o)}.\, \mathrm{surj}(f, M, P(M))$

Table 1: A formulation of the '**Powerset**' problem

The key steps in the diagonalization part of the proof in Figure 1 are:

- the property, that the diagonal element belongs to the power set, is stated in line 9,
- the application of the definition of surjectivity ('**Surj-Def**') in line 10 to prove the existence of an index for the diagonal element, which is assumed to be y_0, is stated in line 11,
- applying the diagonal element, which is a function, on the index y_0 is done in line 14 to obtain an implicit contradiction in line 16,
- the contradiction is made explicit by a case analysis in lines 17 .. 25.

Analyzing the above key proof steps we now want to suggest a systematic way, how to search for a diagonalization proof:

The central point of diagonalization is the construction of the diagonal element. In Figure 1 the diagonal element is represented by a lambda expression that has the indexing function f_0 as a sub-term (see line 9). It is therefore convenient to search for the indexing function first before trying to construct the diagonal element.

[1] This example is taken from [HKC95].

1.	1	$\vdash$	$\exists f.\, \mathrm{surj}(f, M_0, P(M_0))$	(Hyp)
2.	1,2	$\vdash$	$\mathrm{surj}(f_0, M_0, P(M_0))$	(Hyp)
3.	3	$\vdash$	$x \in \lambda z.\, [z \in M_0 \wedge \neg[z \in f_0(z)]]$	(Hyp)
4.	3	$\vdash$	$[x \in M_0 \wedge \neg[x \in f_0(x)]]$	(LambdaE 3)
5.	3	$\vdash$	$x \in M_0$	(AndE 4)
6.		$\vdash$	$[x \in \lambda z.\, [z \in M_0 \wedge \neg[z \in f_0(z)]] \rightarrow x \in M_0]$	(ImpI 5 3)
7.		$\vdash$	$\forall x.\, x \in [\lambda z.\, [z \in M_0 \wedge \neg[z \in f_0(z)]] \rightarrow x \in M_0]$	(ForallI 6)
8.		$\vdash$	$\lambda z.\, [z \in M_0 \wedge \neg[z \in f_0(z)]] \subseteq M_0$	($\subseteq$-Def 7)
9.		$\vdash$	$\lambda z.\, [z \in M_0 \wedge \neg[z \in f_0(z)]] \in P(M_0)$	(PSet-Def 8)
			Proof of 16	
10.	1,2	$\vdash$	$\exists y.\, [y \in M_0 \wedge \lambda z.\, [z \in M_0 \wedge \neg[z \in f_0(z)]] = f_0(y)]$	(Surj-Def 2 9)
11.	1,2,11	$\vdash$	$[y_0 \in M_0 \wedge \lambda z.\, [z \in M_0 \wedge \neg[z \in f_0(z)]] = f_0(y_0)]$	(Hyp)
12.	1,2,11	$\vdash$	$\lambda z.\, [z \in M_0 \wedge \neg[z \in f_0(z)]] = f_0(y_0)$	(AndE 11)
13.		$\vdash$	$y_0 \in f_0(y_0) = y_0 \in f_0(y_0)$	(=-Refl)
14.	1,2,11	$\vdash$	$y_0 \in \lambda z.\, [z \in M_0 \wedge \neg[z \in f_0(z)]] = y_0 \in f_0(y_0)$	(=-Subst 12 13)
15.	1,2,11	$\vdash$	$[y_0 \in \lambda z.\, [z \in M_0 \wedge \neg[z \in f_0(z)]] \leftrightarrow y_0 \in f_0(y_0)]$	(=-Equiv 14)
16.	1,2,11	$\vdash$	$[[y_0 \in M_0 \wedge \neg[y_0 \in f_0(y_0)]] \leftrightarrow y_0 \in f_0(y_0)]$	(LambdaE 15)
			Case 1	
17.	1,2,11,17	$\vdash$	$y_0 \in f_0(y_0)$	(Case 1)
18.	1,2,11,17	$\vdash$	$\neg[y_0 \in f_0(y_0)]$	(16 17)
19.	1,2,11,17	$\vdash$	$\perp$	(NotE 18 17)
			Case 2	
20.	1,2,11,20	$\vdash$	$\neg[y_0 \in f_0(y_0)]$	(Case 2)
21.	1,2,11	$\vdash$	$y_0 \in M_0$	(AndE 11)
22.	1,2,11,20	$\vdash$	$y_0 \in f_0(y_0)$	(16 21 20)
23.	1,2,11,20	$\vdash$	$\perp$	(NotE 20 22)
24.		$\vdash$	$[y_0 \in f_0(y_0) \vee \neg[y_0 \in f_0(y_0)]]$	(TND)
25.	1,2,11	$\vdash$	$\perp$	(OrE 24 19 23)
			End of Case Analysis	
26.	1,2	$\vdash$	$\perp$	(ExistsE 10 25)
27.	1	$\vdash$	$\perp$	(ExistsE 1 26)
28.		$\vdash$	$\neg[\exists f.\, \mathrm{surj}(f, M_0, P(M_0))]$	(NotI 27)
29.		$\vdash$	$\forall M.\, \neg[\exists f.\, \mathrm{surj}(f, M, P(M))]$	(ForallI 28)

Figure 1: A formal proof of the '**Powerset**' example

In the Cantor proof, the function f_0 binds not only the diagonal element but also each element of the enumerable set $P(M_0)$ to an element (its index) in M_0. This property follows from the surjectivity of the function f_0 from M_0 into $P(M_0)$ and is represented by the formula:

$$\forall x_{\iota \rightarrow o}.\, x \in P(M_0) \rightarrow \exists y_{\iota}.\, y \in M_0 \wedge x = f_0(y)$$

The indexing property provides important information for the specification of the diagonal element: its type (a functional type corresponding to the element type of $P(M_0)$), and its domain type (same type as the element type of M_0).

In addition to these type constraints, the diagonal element must be different from each element of the enumerable set $P(M_0)$, i.e. from each $f_0(z)$. In the Cantor proof this is achieved by enforcing that for each z the diagonal element differs from the element $f_0(z)$ in some property. We call this property *the diagonal property* which is represented by $z \in f_0(z)$ in the Cantor proof. The diagonal element inverts this diagonal property (occurrence of $\neg z \in f_0(z)$ in the lambda expression representing the diagonal element in line 9).

In order to get a contradiction, the diagonal element is constructed in such a way, that it belongs to the enumerable set $P(M_0)$ (occurrence of $z \in M_0$ in the lambda expression representing the diagonal element in line 9). Consequently, the diagonal element has an index y_0 and the diagonal property for this element of M_0 ($y_0 \in f_0(y_0)$) is contradicted according to the construction principle of the diagonal element.

To summarize, a diagonalization proof can be carried out in the following way:

1. First we search for an indexing property by ensuring that the formula schema

   $$\forall x_{\alpha \to \beta \bullet} P[x] \to \exists y_{\alpha \bullet} Q[y] \land x = F[y]\ ^2$$

   matches a provable formula.

2. Then we construct a function D (the diagonal element) that belongs to P and inverts the diagonal property $F(x)(x)$. It is not necessary that the function D inverts the property $F(x)(x)$ for each x from the set of indices Q, but it is sufficient to invert the proposition $F(i)(i)$ where i is the index of D in Q. The inverting property of D can therefore be formulated as:

   $$D(i) \leftrightarrow \neg F(i)(i)$$

   The lambda expression schema $\lambda x_\bullet R[F(x), x]$ for a higher-order variable R partially specifies the diagonal element D. Its actual term structure is constructed by instantiating the meta-variable R so that it belongs to P and satisfies the inverting property.

3. Finally, we consider the index i of the diagonal element, which exists due to the indexing property. We make the implicit contradiction in D explicit by a case analysis with the cases $F(i)(i)$ and $\neg F(i)(i)$: One has to deduce $\neg F(i)(i)$ from $F(i)(i)$ and $F(i)(i)$ from $\neg F(i)(i)$ using the equality $D(i) = F(i)(i)$ and the inverting property $D(i) \leftrightarrow \neg F(i)(i)$.

Next let us now look at some other diagonalization examples in order to verify the observations of this section and patch the suggested proof construction.

3 Other Diagonalization Examples

In this section we consider other diagonalization proofs for which the diagonalization argument is somewhat different from that of the Cantor theorem. These differences are important, as we want to extend the diagonalization strategy as suggested in the previous section.

3.1 The Halting Problem

The Halting theorem states that there is no binary computable function (there is no h with $T_2(h)$) which decides for unary computable functions, whether they halt or not. Formally

[2] A term of the form $X[y_1, .., y_n]$ stays for the lambda expression schema $(..((\lambda z_1, .., z_{n \bullet} X)(y_1))..)(y_n)$, where the higher-order variable X denotes a not yet instantiated meta-variable. Whereas a term of the form $X(y_1, .., y_n)$ stays for the application $(..((\lambda z_1, .., z_{n \bullet} X)(y_1))..)(y_n)$, where X denotes a term of the object level.

TND	$\forall x_{o}\bullet x \vee \neg x$
Ext	$\forall f_{\mathbf{N}\to U}\bullet \forall g_{\mathbf{N}\to U}\bullet \forall x_{\mathbf{N}}\bullet f = g \to f(x) = g(x)$
Gödel	$\forall t_{\mathbf{N}\to U}\bullet T_1(t) \to \exists n_{\mathbf{N}}\bullet e(n) = t$
ifComp	$\forall f_{((\mathbf{N}\to U),\mathbf{N})\to B}\bullet T_2(f) \to$ $\forall x_{U}\bullet \forall y_{U}\bullet T_1(\lambda z_{\mathbf{N}}\bullet \mathrm{if}(f(e(z),z) = 0, x, y))$
ifDef	$\forall P_{o}\bullet \forall x_{U}\bullet \forall y_{U}\bullet P \to \mathrm{if}(P,x,y) = x \wedge$ $\neg P \to \mathrm{if}(P,x,y) = y$
defined	$\neg\mathrm{defined}(u) \wedge \mathrm{defined}(0)$
Halting	$\neg\exists h_{((\mathbf{N}\to U),\mathbf{N})\to B}\bullet T_2(h) \wedge \forall t_{\mathbf{N}\to U}\bullet T_1(t)$ $\to \forall x_{\mathbf{N}}\bullet \mathrm{defined}(t(x)) \leftrightarrow h(t,x) = 0$

Table 2: A formulation of the **Halting** problem

expressed: defined($t(x)$) iff $h(t,x) = 0$ for all t with $T_1(t)$ and for all x in **N**. The problem is formulated in Table 2 [3]. In this formalization we use the following sorts: **N** denotes the set of natural numbers. The symbol u represents the non-terminating function. U is the union of N and $\{u\}$. B denotes the set $\{0, 1\}$.

In order to prove the theorem, we need the Gödel enumeration theorem which states that there is an enumeration function e so that for every unary computable function t there is a natural number n so that $e(n)$ corresponds to t. The application of e to any natural number is always a computable function. Furthermore, we use some obvious definitions and the lemma that for a total and computable function f, the function $\lambda z_{\mathbf{N}}\bullet \mathrm{if}(f(e(z),z) = 0, x, y)$ is computable too, where "if(condition,then,else)" has the usual semantics.

Figure 2 shows a formal proof at the assertion level of the Halting problem as formalized in Table 2. This proof was interactively constructed in Ω-MKRP. In the first proof steps (lines 3,4) we assume that there is a computable function *halt* which returns 0 iff a function t halts on an input x. The rest of the proof consists of inferring a contradiction by diagonalization. We want to examine this part of the proof to find the key proof steps noticed in the Cantor diagonalization:

1. The indexing relation is given by the **Gödel** lemma. This delivers the enumerable set T_1, the indexing function e, and the set of indices **N**.

2. The diagonal element in line 6 is represented by a lambda expression that has $e(z)$ and z as sub-terms. Here the term $e(x)(x)$ does not denote a proposition, the diagonal property is therefore a predicate defined on this term (defined($e(x)(x)$)). The inverting property of the diagonal element is guaranteed by the conventional semantics of if and the properties:

   - $\forall x\bullet \mathrm{halt}(e(x),x) = 0 \leftrightarrow \mathrm{defined}(e(x)(x))$ which implies after substituting y_0 for x the conjecture $\mathrm{halt}(e(y_0), y_0) = 0 \leftrightarrow \mathrm{defined}(e(y_0)(y_0))$ in line 13,
   - $\neg$defined(u) (in line 1), and
   - defined(0) (in line 2).

3. With the help of the last three properties and of the equality in line 10, the implicit contradiction in the diagonal element is made explicit in the case analysis (lines 14 .. 29).

[3] This formalization is taken from [HKC95].

1.	1	$\vdash$	$\neg[\text{defined}(u)]$	(Hyp)
2.	2	$\vdash$	$\text{defined}(0)$	(Hyp)
3.	3	$\vdash$	$\exists h.[T_2(h) \wedge \forall t.[T_1(t) \rightarrow \forall x.[h(t,x) = 0 \leftrightarrow \text{defined}(t(x))]]]$	(Hyp)
4.	4	$\vdash$	$[T_2(halt) \wedge \forall t.[T_1(t) \rightarrow \forall x.[\text{halt}(t,x) = 0 \leftrightarrow \text{defined}(t(x))]]]$	(Hyp)
5.	4	$\vdash$	$T_2(halt)$	(4)
6.	4	$\vdash$	$T_1(\lambda z.\text{if}(\text{halt}(e(z),z) = 0, u, 0))$	(ifComp 5)
			Proof of 10 and 13	
7.	4	$\vdash$	$\exists n.e(n) = \lambda z.\text{if}(\text{halt}(e(z),z) = 0, u, 0)$	(Gödel 6)
8.	8	$\vdash$	$e(y_0) = \lambda z.\text{if}(\text{halt}(e(z),z) = 0, u, 0)$	(Hyp)
9.	8	$\vdash$	$e(y_0)(y_0) = (\lambda z.\text{if}(\text{halt}(e(z),z) = 0, u, 0))(y_0)$	(Ext 8)
10.	8	$\vdash$	$e(y_0)(y_0) = \text{if}(\text{halt}(e(y_0),y_0) = 0, u, 0)$	(LambdaE 9)
11.	8	$\vdash$	$\lambda z.\text{if}(\text{halt}(e(z),z) = 0, u, 0) = e(y_0)$	(=Com 8)
12.	4,8	$\vdash$	$T_1(e(y_0))$	(=Subst 11 6)
13.	8,4	$\vdash$	$[\text{halt}(e(y_0),y_0) = 0 \leftrightarrow \text{defined}(e(y_0)(y_0))]$	(4 12)
			Case 1	
14.	14	$\vdash$	$\text{halt}(e(y_0),y_0) = 0$	(Case 1)
15.	14	$\vdash$	$\text{if}(\text{halt}(e(y_0),y_0) = 0, u, 0) = u$	(ifDef 14)
16.	14,8	$\vdash$	$e(y_0)(y_0) = u$	(=Trans 10 15)
17.	8,14	$\vdash$	$u = e(y_0)(y_0)$	(=Com 16)
18.	1,14,8	$\vdash$	$\neg[\text{defined}(e(y_0)(y_0))]$	(=Subst 17 1)
19.	14,4,8	$\vdash$	$\text{defined}(e(y_0)(y_0))$	($\leftrightarrow$SubE 13 14)
20.	8,4,14,1	$\vdash$	$\perp$	(NotE 18 19)
			Case 2	
21.	21	$\vdash$	$\neg[\text{halt}(e(y_0),y_0) = 0]$	(Case 2)
22.	21	$\vdash$	$\text{if}(\text{halt}(e(y_0),y_0) = 0, u, 0) = 0$	(ifDef 21)
23.	21,8	$\vdash$	$e(y_0)(y_0) = 0$	(=Trans 10 22)
24.	8,21	$\vdash$	$0 = e(y_0)(y_0)$	(=Com 23)
25.	2,21,8	$\vdash$	$\text{defined}(e(y_0)(y_0))$	(=Subst 24 2)
26.	21,4,8	$\vdash$	$\neg[\text{defined}(e(y_0)(y_0))]$	($\leftrightarrow$SubE 13 21)
27.	8,21,2,4	$\vdash$	$\perp$	(NotE 26 25)
28.		$\vdash$	$[\text{halt}(e(y_0),y_0) = 0 \vee \neg[\text{halt}(e(y_0),y_0) = 0]]$	(TND)
29.	4,2,8,1	$\vdash$	$\perp$	(OrE 28 20 27)
			End of Case Analysis	
30.	1,2,4	$\vdash$	$\perp$	(ExistsE 7 29)
31.	2,1,3	$\vdash$	$\perp$	(ExistsE 3 30)
32.	1,2	$\vdash$	$\neg[\exists h.[T_2(h) \wedge \forall t.[T_1(t) \rightarrow \forall x.[h(t,x) = 0 \leftrightarrow \text{defined}(t(x))]]]]$	(NotI 31)

Figure 2: A formal proof of the **Halting** example

Compared to the proof of the Cantor theorem, the inverting property of the diagonal element is more complicated here. The diagonal element is represented by an *if*-term, whose condition-sub-term $\text{halt}(e(z), z) = 0$ is equivalent to the diagonal property $\text{defined}(e(x)(x))$. The *else*-sub-term 0 belongs to the relation of the diagonal property predicate, but the *then*-sub-term u does not.

Consequently, we can specify the diagonal element, in case the term $F(x)(x)$ does not denote a proposition, by the lambda expression schema $\lambda x.\, \text{if}(R[F(x), x], Y[x], Z[x])$. The inverting property holds if the following three properties can be satisfied:

- $R[F(i), i] \leftrightarrow U[F(i)(i)]$ where $U[F(x)(x)]$ denotes the diagonal property,
- $\neg U[Y[i]]$, and
- $U[Z[i]]$.

In the above formulae, i denotes the index of the diagonal element. Note that $R[F(x), x]$ and $U[F(x)(x)]$ can be instantiated with the same object term.

3.2 The 'Total' Problem

Consider the theorem '**Total**':

> The set $TOT = \{x \in \mathrm{N} | \forall y_{\bullet} \, \mathrm{defined}(\Phi(x, y))\}$ of indices for total computable functions is not recursively enumerable. Φ denotes the universal function which takes two natural numbers x, and y as arguments and delivers the result of the call of the x^{th} computable function in the Gödel enumeration with y as argument.

The informal proof of this theorem is given in [DSW94], page 90. This theorem is formalized in Table 3 and a formal proof is given in Figure 3.

TOTdef1	$\forall n_{\mathrm{N}^\bullet} \, \mathrm{TOT}(n) \rightarrow \mathrm{totcomp}(\lambda x_{\mathrm{N}^\bullet} \, \Phi(n, x))$
TOTdef2	$\forall f_{\mathrm{N}\rightarrow\mathrm{Res}^\bullet} \, \mathrm{totcomp}(f) \rightarrow \exists n_{\mathrm{N}^\bullet} \, \mathrm{TOT}(n) \wedge f = \lambda x_{\mathrm{N}^\bullet} \, \Phi(n, x)$
totcomp1	$\forall f_{\mathrm{N}\rightarrow\mathrm{Res}^\bullet} \, \mathrm{totcomp}(f) \leftrightarrow \forall x_{\mathrm{N}^\bullet} \, \mathrm{defined}(f(x))$
totcomp2	$\forall f_{\mathrm{N}\rightarrow\mathrm{Res}^\bullet} \, \mathrm{totcomp}(f) \rightarrow \mathrm{totcomp}(\lambda x_{\mathrm{N}^\bullet} \, f(x) + 1)$
r.e.Lem	$\forall s_{\mathrm{N}\rightarrow o^\bullet} \, \mathrm{r.e.}(s) \wedge \mathrm{nempty}(s) \rightarrow$ $\exists g_{\mathrm{N}\rightarrow\mathrm{N}^\bullet} \, \forall x_{\mathrm{N}^\bullet} \, s(g(x)) \wedge \forall y_{\mathrm{N}^\bullet} \, s(y) \rightarrow \exists z_{\mathrm{N}^\bullet} \, y = g(z)$
nempty	$\mathrm{nempty}(TOT)$
=Axiom	$\forall x_{\mathrm{Res}^\bullet} \, \neg(x + 1 = x)$
Total	$\neg \mathrm{r.e.}(\mathrm{TOT})$

Table 3: A formulation of the '**Total**' problem

After assuming that the set TOT is r.e. (line 2) and after applying the lemma **r.e.Lem** to get the conjecture in line 3, a contradiction is derived using the diagonalization technique as follows:

- Construct the diagonal element and show that this belongs to the set *totcomp* (line 15). The function $\lambda x_\bullet \, \lambda y_\bullet \, \Phi(g(y), x)$ is the indexing function.
- Prove the existence of an index i for the diagonal element (lines 16 .. 22).
- Deduce the equality in line 25 which contradicts the equality axiom **=Axiom** from the equality in line 22.

The diagonalization part of the proof in Figure 3 differs from that of the Cantor problem and Halting problem in two aspects: First, the actual indexing property in this example is represented by the conjecture

$$\forall f_{\mathrm{N}\rightarrow\mathrm{Res}^\bullet} \, \mathrm{totcomp}(f) \rightarrow \exists n_{\mathrm{N}^\bullet} \, f = \lambda x_{\mathrm{N}^\bullet} \, \Phi(g_0(n), x) \tag{1}$$

and therefore cannot be directly proved by the application of an assertion from the problem description. Although the assertion **TOTdef2** satisfies the property of an indexing relation, the second conjunct in line 4 of Figure 3 allows the deduction of a second possible indexing property (1). Consequently, we have to deal with problem situations when there is more than one indexing property. Moreover the examination of the hypotheses in the problem

1.	1	$\vdash$	nempty(TOT)	(Hyp)
2.	2	$\vdash$	r.e.(TOT)	(Hyp)
3.	1,2	$\vdash$	$\exists g.[\forall x.\mathrm{TOT}(g(x)) \land \forall y.[\mathrm{TOT}(y) \rightarrow \exists z.y = g(z)]]$	(r.e.Lem 2 1)
4.	4	$\vdash$	$[\forall x.\mathrm{TOT}(g_0(x)) \land \forall y.[\mathrm{TOT}(y) \rightarrow \exists z.y = g_0(z)]]$	(Hyp)
5.	4	$\vdash$	$\mathrm{TOT}(g_0(y_0))$	(4)
6.	4	$\vdash$	$\mathrm{totcomp}(\lambda x.\Phi(g_0(y_0), x))$	(TOTdef1 5)
7.	4	$\vdash$	$\forall y.\mathrm{totcomp}(\lambda x.\Phi(g_0(y), x))$	($\forall I$ 6)
8.	4	$\vdash$	$\mathrm{totcomp}(\lambda z.\Phi(g_0(x_0), z))$	($\forall E$ 7)
9.	4	$\vdash$	$\mathrm{defined}((\lambda z.\Phi(g_0(x_0), z))(x_0))$	(totcomp1 8)
10.	4	$\vdash$	$\mathrm{defined}(\Phi(g_0(x_0), x_0))$	(LambdaE 9)
11.	4	$\vdash$	$\mathrm{defined}((\lambda y.\Phi(g_0(y), y))(x_0))$	(LambdaI 10)
12.	4	$\vdash$	$\forall x.\mathrm{defined}((\lambda y.\Phi(g_0(y), y))(x))$	($\forall I$ 11)
13.	4	$\vdash$	$\mathrm{totcomp}(\lambda y.\Phi(g_0(y), y))$	(totcomp1 12)
14.	4	$\vdash$	$\mathrm{totcomp}(\lambda x.(\lambda y.\Phi(g_0(y), y))(x) + 1)$	(totcomp2 13)
15.	4	$\vdash$	$\mathrm{totcomp}(\lambda x.\Phi(g_0(x), x) + 1)$	(LambdaE 14)
			Proof of 24	
16.	4	$\vdash$	$\exists p.[\mathrm{TOT}(p) \land \lambda x.\Phi(g_0(x), x) + 1 = \lambda x.\Phi(p, x)]$	(TOTdef2 15)
17.	17	$\vdash$	$[\mathrm{TOT}(p_0) \land \lambda x.\Phi(g_0(x), x) + 1 = \lambda x.\Phi(p_0, x)]$	(Hyp)
18.	17	$\vdash$	$\mathrm{TOT}(p_0)$	($\land E$ 17)
19.	17	$\vdash$	$\lambda x.\Phi(g_0(x), x) + 1 = \lambda x.\Phi(p_0, x)$	($\land E$ 17)
20.	4,17	$\vdash$	$\exists z.p_0 = g_0(z)$	(L8 18)
21.	21	$\vdash$	$p_0 = g_0(i)$	(Hyp)
22.	17,21	$\vdash$	$\lambda x.\Phi(g_0(x), x) + 1 = \lambda x.\Phi(g_0(i), x)$	(=Subst 21 19)
23.	17,21	$\vdash$	$\forall x.(\lambda x.\Phi(g_0(x), x) + 1)(x) = (\lambda x.\Phi(g_0(i), x))(x)$	(Ext-I 22)
24.	17,21	$\vdash$	$(\lambda x.\Phi(g_0(x), x) + 1)(i) = (\lambda x.\Phi(g_0(i), x))(i)$	($\forall E$ 23)
			Explicit Contradiction	
25.	17,21	$\vdash$	$\Phi(g_0(i), i) + 1 = \Phi(g_0(i), i)$	(LambdaE 24)
26.		$\vdash$	$\neg[\Phi(g_0(i), i) + 1 = \Phi(g_0(i), i)]$	(=Axiom)
27.	17,21	$\vdash$	$\bot$	($\neg E$ 26 25)
			End of explicit Contradiction	
28.	4,17	$\vdash$	$\bot$	($\exists E$ 20 27)
29.	4	$\vdash$	$\bot$	($\exists E$ 16 28)
30.	1,2	$\vdash$	$\bot$	($\exists E$ 3 29)
31.	1	$\vdash$	$\neg$[r.e.(TOT)]	($\neg I$ 30)

Figure 3: A formal proof of the '**Total**' example

description to verify whether one of them can assert (can be used as assertion to prove) an indexing property is incomplete relative to this task. That is, it is not enough to check the availability of an indexing relation in the problem situation.

The second difference concerns the construction of the implicit contradiction in the diagonal element: In the first two examples the diagonal element should contradict the diagonal property (the term $U[F(x)(x)]$ with type o), but in this example the type of the diagonal term $F(x)(x)$ is different from the truth value type o. The diagonal element should be a term containing $F(x)(x)$ as a sub-term and never equal $F(x)(x)$. Actually, it is enough to satisfy the inequality of the diagonal element to the term $F(x)(x)$ only at the position $(F(i), i)$ in the diagonal, where i denotes the index of the diagonal element. This means the diagonal element should be represented by the lambda expression schema $\lambda x.\, G[F(x)(x)]$ and satisfy the inequality $G[F(i)(i)] \neq F(i)(i)$. In general, this alternative has to be taken into account for the construction of a diagonal element.

3.3 The 'Kset' problem

We consider the theorem '**Kset**':

> The set $K = \{x \in \mathrm{N} | \neg\mathrm{defined}(\Phi(x,x))\}$ is not recursively enumerable (an exercise from [DSW94], page 94).

The formalization of this theorem and the necessary assertions is given in Table 4.

Kdef	$\forall n_{\mathrm{N}}.\, K(n) \leftrightarrow \neg\mathrm{defined}(\Phi(n,n))$
compLem	$\forall f_{\mathrm{N}\to\mathrm{Res}}.\, \mathrm{comp}(f) \to \exists n_{\mathrm{N}}.\, f = \lambda x_{\mathrm{N}}.\, \Phi(n,x)$
r.e.Def	$\forall s_{\mathrm{N}\to o}.\, \mathrm{r.e.}(s) \leftrightarrow$ $\exists g_{\mathrm{N}\to\mathrm{Res}}.\, \mathrm{comp}(g) \wedge \forall x_{\mathrm{N}}.\, s(x) \leftrightarrow \mathrm{defined}(g(x))$
TND	$\forall \phi_o.\, \phi \vee \neg\phi$
Kset	$\neg\mathrm{r.e.}(K)$

Table 4: A formulation of the '**Kset**' problem

Figure 4 shows a formal proof of the '**Kset**' problem at the assertion level, that was constructed interactively in Ω-MKRP. After assuming that the set K is recursively enumerable (line 1) and applying the recursive enumerability definition (line 2) we show a contradiction by diagonalization as follows:

- We state that the diagonal element g_0 belongs to the set *comp* (line 4),
- We prove the existence of an index n_0 for the diagonal element (lines 5,6), and
- We deduce the obvious contradiction in line 11 with the help of the equality in line 6, the second conjunct in line 3, and the definition **Kdef**.

The diagonalization proof part in Figure 4 differs from the previous diagonalization proofs in that the diagonal element (the function g_0) is directly given by the problem situation and does not need to be constructed. Consequently such an alternative should be taken into account when searching for a diagonal element. The lambda expression schemata, that were suggested in the previous examples to specify the diagonal element D, can be instantiated to a function symbol which belongs to the enumerable set. The inverting property of D can be stated if one of the conjectures $U[D(i)] \leftrightarrow \neg U[F(i)(i)]$ and $D(i) \neq F(i)(i)$ can be proved from the current hypotheses. U is a meta-variable, F denotes the indexing function, and i is the index of the diagonal element D.

More examples are given in the appendix.

4 A Diagonalization Strategy

In this section we summarize the properties of the presented diagonalization proofs. First, we give the essential proof steps and then suggest how these steps can be performed and implemented within a proof planning process.

The goal of the diagonalization strategy is a contradiction, i.e. $\perp$. A diagonalization proof plan can now be constructed in the following way:

1.	1	$\vdash$	r.e.(K)	(Hyp)
2.	1	$\vdash$	$\exists g.[\text{comp}(g) \land \forall x.[K(x) \leftrightarrow \text{defined}(g(x))]]$	(r.e.Def 1)
3.	3	$\vdash$	$[\text{comp}(g_0) \land \forall x.[K(x) \leftrightarrow \text{defined}(g_0(x))]]$	(Hyp)
4.	3	$\vdash$	comp(g_0)	(3)
			Proof of 11	
5.	3	$\vdash$	$\exists n.g_0 = \lambda x.\Phi(n, x)$	(compLem 4)
6.	6	$\vdash$	$g_0 = \lambda x.\Phi(n_0, x)$	(Hyp)
7.	3	$\vdash$	$[K(n_0) \leftrightarrow \text{defined}(g_0(n_0))]$	(3)
8.	3,6	$\vdash$	$[K(n_0) \leftrightarrow \text{defined}((\lambda x.\Phi(n_0, x))(n_0))]$	(=Subst 6 7)
9.	3,6	$\vdash$	$[K(n_0) \leftrightarrow \text{defined}(\Phi(n_0, n_0))]$	(LambdaE 8)
10.		$\vdash$	$[K(n_0) \leftrightarrow \neg[\text{defined}(\Phi(n_0, n_0))]]$	(Kdef)
11.	3,6	$\vdash$	$[\text{defined}(\Phi(n_0, n_0)) \leftrightarrow \neg[\text{defined}(\Phi(n_0, n_0))]]$	($\leftrightarrow$SubE 9 10)
			Case 1	
12.	12	$\vdash$	$\text{defined}(\Phi(n_0, n_0))$	(Case 1)
13.	3,6,12	$\vdash$	$\neg[\text{defined}(\Phi(n_0, n_0))]$	($\leftrightarrow$SubE 11 12)
14.	3,6,12	$\vdash$	$\bot$	($\neg E$ 13 12)
			Case 2	
15.	15	$\vdash$	$\neg[\text{defined}(\Phi(n_0, n_0))]$	(Case 2)
16.	3,6,15	$\vdash$	$\text{defined}(\Phi(n_0, n_0))$	($\leftrightarrow$SubI 11 15)
17.	3,6,15	$\vdash$	$\bot$	($\neg E$ 15 16)
18.		$\vdash$	$[\text{defined}(\Phi(n_0, n_0)) \lor \neg[\text{defined}(\Phi(n_0, n_0))]]$	(TND)
19.	3,6	$\vdash$	$\bot$	($\lor E$ 18 14 17)
			End of Case Analysis	
20.	3	$\vdash$	$\bot$	($\exists E$ 5 19)
21.	1	$\vdash$	$\bot$	($\exists E$ 2 20)
22.		$\vdash$	$\neg$[r.e.(K)]	($\neg I$ 21)

Figure 4: A formal proof of the '**Kset**' example

1. Verify that the formula schema

$$\forall x_{\alpha \to \beta}.\, P[x] \to \exists y_{\alpha}.\, Q[y] \land x = F[y]$$

matches a provable formula from the hypotheses in order to obtain an indexing property,

2. Check whether a function D (the diagonal element) belongs to P and satisfies an inverting property relative to the term $F(x)(x)$. The specification of D and the corresponding inverting property for the index i of D depend on the type of P in this construction:

   - If the type of P is $(\alpha \to o) \to o$, i.e. the term $F(x)(x)$ denotes a proposition, then D is a predicate and must unify the lambda expression schema $\lambda x_{\alpha}.\, R[F(x), x]$. The inverting property of D is ensured by the formula $D(i) \leftrightarrow \neg F(i)(i)$.
   - Otherwise, D is a function and must unify either:
     - a lambda expression of the form $\lambda x_{\alpha}.\, \text{if}(R[F(x), x], Y[x], Z[x])$; The inverting property of D is guaranteed by the formulae: $R[F(i), i] \leftrightarrow U[F(i)(i)]$, $\neg U[Y[i]]$, and $U[Z[i]]$,
     - or a lambda expression of the form $\lambda x_{\alpha}.\, G[F(x)(x)]$, where G differs from the identity $\lambda x_{\beta}.\, x$; The inverting property is satisfied if one of the conjectures $D(i) \neq F(i)(i)$ and $U[D(i)] \leftrightarrow \neg U[F(i)(i)]$ can be proven.

3. Find the proof plan for making the implicit contradiction of the diagonal element explicit; the structure of this plan can be determined from the instantiation of the diagonal element and the corresponding inverting property,
4. Generate a proof plan for the whole diagonalization proof by using the partial plans computed in the previous three steps.

The success of the diagonalization proof strategy depends mainly on the first and the second proof step, i.e. on the existence of an indexing property and the existence of a function (the diagonal element) that satisfies an inverting property relative to the term $F(i)(i)$ (i is the index of this diagonal element.) and belongs to the enumerable set P. Depending on the task at hand we need special methods for the special planning task. Furthermore, we need control knowledge to solve conflict situations, i.e. situations with many applicable methods.

Verifying the existence of an indexing property amounts to the general and complex task whether a formula schema matches a provable formula from the hypotheses. It is difficult to obtain all provable formulae which match the schema. We suggest therefore to restrict this task to find the formulae which can be proved by assertion application of the hypotheses and which match the schema. The methods to be used for planning assertion applications should specify whether a hypothesis can be an assertion to prove a formula schema and should specify the resulted subgoals, i.e. the premises of this assertion application.

Assertion application alone is not enough to determine all possible indexing properties. For instance, in the '**Total**' example one has to combine several assertions to get the right indexing property. Therefore we must extend the procedure of assertion application with the possibility to combine assertions. For this purpose we must investigate how an indexing property could be proved by combining assertions. Moreover we must provide control knowledge to choose one indexing property, if several are available. A control rule could state that hypotheses which do not belong to the original proof assumptions (i.e. which are introduced during the proof) are more important in proving the current goal if this goal depends on them: Indexing properties whose proof involves such hypotheses should be preferred.

The second main step in a diagonalization proof is the construction of the diagonal element. In the given diagonalization strategy, the function that corresponds to the diagonal element is partially specified. It must be an element of the enumerable set P, it has to unify with some lambda expression schemata, and finally a proposition that depends on it, i.e. the inverting property, must hold. We suggest to use middle out reasoning [KBB93] for the construction of this function. The goals in this process are the formula schema $P(D)$, where P is the enumerable set and D is the meta-variable which represents the function to be constructed, and finally the conjectures that specify the inverting property.

We suggest to prove these goals by assertion application and higher order unification. For instance, in the Halting problem, the membership of the diagonal element D to the enumerable set T_1, i.e. the formula schema $T_1(D)$, can be reduced by applying the assertion **ifComp**

$$\forall f_{((\mathrm{N}\to U),\mathrm{N})\to B}. T_2(f) \to \forall x_U. \forall y_U. T_1(\lambda z_{\mathrm{N}}. \mathrm{if}(f(e(z), z) = 0, x, y))$$

after unifying D with $\lambda z_{\mathrm{N}}. \mathrm{if}(F_1(e(z), z) = 0, x, y)$, to the subgoal $T_2(F_1)$. F_1 is a meta-variable which can be instantiated while proving the resulted subgoal $T_2(F_1)$.

In general, assertion application is not enough to fully instantiate all the meta-variables that occur in the considered subgoals. There can be subgoals that cannot be proven from any assertion. In such situations, instantiation alternatives need to be suggested in order to satisfy the goal at hand and continue the search process. We suggest to provide the instantiation alternatives using special heuristics which satisfy a goal by proposing some possible bindings of its higher-order variables. For instance, one heuristic can satisfy the formula schema $\forall x. R[F(x), x] \rightarrow M_0(x)$ by instantiating the higher-order variable R to $\lambda x. \lambda y. M_0(y) \wedge R_1[x, y]$.

5 Conclusion and Future Work

In this report we presented an empirical study of proofs by diagonalization and exploited their similarities to suggest a diagonalization proof strategy. In order to effectively plan diagonalization proofs the following should be done:

- Methods have to be designed and implemented for the proof of an indexing property by application of assertions, and control knowledge has to be developed to apply these methods within a planning process. (Or a procedure has to be implemented to search for an indexing property using the current hypotheses.)
- Methods and heuristics for the construction of the diagonal element by middle out reasoning have to be designed and implemented. Planning with such methods involves the use of higher-order unification which in general delivers many solutions some of which are not useful at all. Preferred solutions should be formally described and specified in order to apply approaches which restrict the solutions of higher-order unification (similar say to the use of HOL-unification in linguistic analysis [GK96]).

Other questions that need to be answered are whether an indexing property could be formulated using other proof schemata and whether there is another specification for diagonal elements (In three other diagonalization examples, which are presented in the Appendix, the suggested diagonalization proof strategy can be successfully applied.). To answer these questions, more examples and especially other problem descriptions from the literature should be empirically examined.

Moreover, we want to design a general framework for proof planning where proof strategies in addition to methods can be declaratively represented (including control knowledge) in an interactive proof development system such as Ω-MKRP.

References

[Bun91] Alan Bundy. A Science of Reasoning. In *Computational Logic: Essays in honor of Alan Robinson.* MIT Press, 1991. also presented at the 10th CADE 1990 as extended abstract.

[BvHHS90] Alan Bundy, Frank van Harmelen, Christian Horn, and Alan Smaill. The OYSTER-CLAM system. In Mark E. Stickel, editor, *Proceedings of the 10th CADE,* pages 647–648, Kaiserslautern, Germany, 1990. Springer Verlag, Berlin, Germany, LNAI 449.

[DSW94] Martin D. Davis, Ron Sigal, and Elaine J. Weyuker. *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science*. Academic Press, second edition, 1994.

[Gen35] Gerhard Gentzen. Untersuchungen über das logische Schließen I. *Mathematische Zeitschrift*, **39**:176–210, 1935.

[GK96] Claire Gardent and Michael Kohlhase. Higher–order coloured unification and natural language semantics. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*. ACL, Santa Cruz, 1996.

[HKC95] Xiaorong Huang, Manfred Kerber, and Lassaad Cheikhrouhou. Adaptation of declaratively represented methods in proof planning. SEKI Report SR-95-12, Fachbereich Informatik, Universität des Saarlandes, Im Stadtwald, Saarbrücken, Germany, 1995.

[HKK$^+$94] Xiaorong Huang, Manfred Kerber, Michael Kohlhase, Erica Melis, Dan Nesmith, Jörn Richts, and Jörg Siekmann. Ω-MKRP: A Proof Development Environment. In Alan Bundy, editor, *Proceedings of the 12th CADE*, pages 788–792, Nancy, 1994. Springer Verlag, Berlin, Germany, LNAI 814.

[Hua94] Xiaorong Huang. Reconstructing proofs at the assertion level. In Alan Bundy, editor, *Proceedings of the 12th CADE*, pages 738–752, Nancy, France, 1994. Springer Verlag, Berlin, Germany, LNAI 814.

[KBB93] I. Kraan, D. Basin, and A. Bundy. Middle-out reasoning for program synthesis. In P. Szeredi, editor, *Proceedings of the 10-th International Conference on Logic Programming*. MIT Press, 1993.

[Kle43] Stephen C. Kleene. Recursive predicates and quantifiers. In Martin Davis, editor, *The Undecidable: Basic Papers On Undecidable Propositions, Unsolvable Problems And Computable Functions*, pages 254–287. Raven Press, Hewlett, New York, 1965, 1943.

Appendix A: Three additional examples

In this appendix we present three more theorems whose proofs are mainly based on the diagonalization technique and we reflect upon the proof steps of the suggested diagonalization strategy in the given formal proofs.

A.1 The 'NatReal' problem

The theorem in Table 5 whose formalization is taken from [HKC95] states that there is no surjective function from the natural numbers onto the interval $[0, 1]$. In its formal proof, which is shown in Figure 5, we assume that there is such a surjective function f_0 and then we prove a contradiction by diagonalization, where the indexing property follows from the surjectivity of f_0. The diagonal element is represented by an *if*-construct (line 17). The membership of the diagonal element to the interval $[0, 1]$ is shown in lines 4 .. 17. In the proof part from line 18 to line 23 an implicit contradiction is derived by first applying the indexing property (the surjectivity of f_0) in order to get an index y_0 for the diagonal element and then applying the diagonal element to this index to deliver the equality in line 23 which embodies the implicit contradiction. This contradiction is made explicit by a case analysis (lines 25 .. 37).

TND	$\forall x_o \centerdot x \vee \neg x$
surjDef	$\forall f_{\iota\rightarrow(\iota\rightarrow\iota)} \centerdot \forall D_{\iota\rightarrow o} \centerdot \forall C_{(\iota\rightarrow\iota)\rightarrow o} \centerdot$ $\text{surj}(f, D, C) \leftrightarrow$ $\forall y_{\iota\rightarrow\iota} \centerdot y \in C \rightarrow \exists x_\iota \centerdot x \in D \wedge y = f(x)$
[0,1]Def	$\forall h_{\iota\rightarrow\iota} \centerdot h \in [0, 1] \leftrightarrow (\forall n_\iota \centerdot n \in \mathbf{N} \rightarrow \text{dig}(h(n)))$
digits	$\text{dig}(0) \wedge \text{dig}(1)$
=Axiom	$0 \neq 1$
ifDef	$\forall P_o \centerdot \forall x_\iota \centerdot \forall y_\iota \centerdot P \rightarrow \text{if}(P, x, y) = x \wedge$ $\neg P \rightarrow \text{if}(P, x, y) = y$
NatReal	$\neg\exists f_{\iota\rightarrow(\iota\rightarrow\iota)} \centerdot \text{surj}(f, \mathbf{N}, [0, 1])$

Table 5: A formulation of the '**NatReal**' problem

The diagonalization proof in this example corresponds to the type of the diagonalization strategy as presented in section 4 where the diagonal element is represented by the lambda expression schema $\lambda x_\alpha \centerdot \text{if}(R[F(x), x], Y[x], Z[x])$. The meta-variables R, Y, and Z are to be instantiated respectively by $\lambda x \centerdot \lambda y \centerdot x(y) = 1$, $\lambda x \centerdot 0$, and $\lambda x \centerdot 1$. The meta-variable U in the inverting property $R[F(i), i] \leftrightarrow U[F(i)(i)]$ should be instantiated to $\lambda x \centerdot x = 1$.

A.2 The 'totFn' problem

We consider the theorem that there is no enumeration f_0, f_1, f_2, ... of all total unary functions on $\mathbf{N}$ (an exercise from [DSW94], page 94). This theorem is equivalent to the conjecture that there is no surjective function from the set of natural numbers $\mathbf{N}$ onto the set tfn of total unary functions on $\mathbf{N}$ which is formalized with the necessary assertions in Table 6. In the assertion level proof of this theorem, shown in Figure 6, we assume the existence of such a function g_0 and then prove a contradiction by diagonalization. The diagonal element corresponds to the function $\lambda z \centerdot s(g_0(z)(z))$ (line 10) and the indexing

1.	1	$\vdash$	$\neg[0 = 1]$	(Hyp)
2.	2	$\vdash$	$\exists f.\mathrm{surj}(f, N, [0,1])$	(Hyp)
3.	3	$\vdash$	$\mathrm{surj}(f_0, N, [0,1])$	(Hyp)
4.		$\vdash$	$[f_0(n)(n) = 1 \vee \neg[f_0(n)(n) = 1]]$	(TND)
5.	5	$\vdash$	$f_0(n)(n) = 1$	(Case 1)
6.	5	$\vdash$	$\mathrm{if}(f_0(n)(n) = 1, 0, 1) = 0$	(ifDef 5)
7.		$\vdash$	$\mathrm{dig}(0)$	(digits)
8.	5	$\vdash$	$\mathrm{dig}(\mathrm{if}(f_0(n)(n) = 1, 0, 1))$	(=Subst 6 7)
9.	9	$\vdash$	$\neg[f_0(n)(n) = 1]$	(Case 2)
10.	9	$\vdash$	$\mathrm{if}(f_0(n)(n) = 1, 0, 1) = 1$	(ifDef 9)
11.		$\vdash$	$\mathrm{dig}(1)$	(digits)
12.	9	$\vdash$	$\mathrm{dig}(\mathrm{if}(f_0(n)(n) = 1, 0, 1))$	(=Subst 10 11)
13.		$\vdash$	$\mathrm{dig}(\mathrm{if}(f_0(n)(n) = 1, 0, 1))$	($\vee E$ 4 8 12)
14.		$\vdash$	$\mathrm{dig}(\lambda z.\mathrm{if}(f_0(z)(z) = 1, 0, 1)(n))$	(LambdaI 13)
15.		$\vdash$	$[n \in N \rightarrow \mathrm{dig}(\lambda z.\mathrm{if}(f_0(z)(z) = 1, 0, 1)(n))]$	($\rightarrow I$ 14)
16.		$\vdash$	$\forall n.[n \in N \rightarrow \mathrm{dig}(\lambda z.\mathrm{if}(f_0(z)(z) = 1, 0, 1)(n))]$	($\forall I$ 15)
17.		$\vdash$	$\lambda z.\mathrm{if}(f_0(z)(z) = 1, 0, 1) \in [0,1]$	([0,1]Def 16)
			Proof of 24	
18.	3	$\vdash$	$\exists y.[y \in N \wedge \lambda z.\mathrm{if}(f_0(z)(z) = 1, 0, 1) = f_0(y)]$	(surjDef 3 17)
19.	19	$\vdash$	$[y_0 \in N \wedge \lambda z.\mathrm{if}(f_0(z)(z) = 1, 0, 1) = f_0(y_0)]$	(Hyp)
20.	19	$\vdash$	$\lambda z.\mathrm{if}(f_0(z)(z) = 1, 0, 1) = f_0(y_0)$	($\wedge E$ 19)
21.	19	$\vdash$	$\forall y.\lambda z.\mathrm{if}(f_0(z)(z) = 1, 0, 1)(y) = f_0(y_0)(y)$	(Ext-I 20)
22.	19	$\vdash$	$\lambda z.\mathrm{if}(f_0(z)(z) = 1, 0, 1)(y_0) = f_0(y_0)(y_0)$	($\forall E$ 21)
23.	19	$\vdash$	$\mathrm{if}(f_0(y_0)(y_0) = 1, 0, 1) = f_0(y_0)(y_0)$	(LambdaE 22)
24.	19	$\vdash$	$f_0(y_0)(y_0) = \mathrm{if}(f_0(y_0)(y_0) = 1, 0, 1)$	(=Com 23)
			Case 1	
25.	25	$\vdash$	$f_0(y_0)(y_0) = 1$	(Case 1)
26.	25	$\vdash$	$\mathrm{if}(f_0(y_0)(y_0) = 1, 0, 1) = 0$	(ifDef 25)
27.	25	$\vdash$	$1 = f_0(y_0)(y_0)$	(=Com 25)
28.	19,25	$\vdash$	$1 = \mathrm{if}(f_0(y_0)(y_0) = 1, 0, 1)$	(=Trans 27 24)
29.	19,25	$\vdash$	$1 = 0$	(=Trans 28 26)
30.	19,25	$\vdash$	$0 = 1$	(=Com 29)
31.	1,19,25	$\vdash$	$\perp$	($\neg E$ 1 30)
			Case 2	
32.	32	$\vdash$	$\neg[f_0(y_0)(y_0) = 1]$	(Case 2)
33.	32	$\vdash$	$\mathrm{if}(f_0(y_0)(y_0) = 1, 0, 1) = 1$	(ifDef 32)
34.	19,32	$\vdash$	$f_0(y_0)(y_0) = 1$	(=Trans 24 33)
35.	19,32	$\vdash$	$\perp$	($\neg E$ 32 34)
36.		$\vdash$	$[f_0(y_0)(y_0) = 1 \vee \neg[f_0(y_0)(y_0) = 1]]$	(TND)
37.	1,19	$\vdash$	$\perp$	($\vee E$ 36 31 35)
			End of Case Analysis	
38.	1,3	$\vdash$	$\perp$	($\exists E$ 18 37)
39.	1,2	$\vdash$	$\perp$	($\exists E$ 2 38)
40.	1	$\vdash$	$\neg[\exists f.\mathrm{surj}(f, N, [0,1])]$	($\neg I$ 39)

Figure 5: A formal proof of the '**NatReal**' example

property follows from the surjectivity of g_0. The membership of the diagonal element to the set tfn is proved in lines 3 .. 10. In lines 11 and 12 an index y_0 of the diagonal element is determined by applying the surjectivity definition to the formula in line 10. With the help of the extensionality property we obtain from the function equality in line 12 the equality in line 16 which consists of an implicit contradiction. This implicit contradiction is made explicit in lines 17 .. 21.

The diagonalization proof part here corresponds to the type of the diagonalization strat-

tfnDef	$\forall f_{\iota\to\iota}\boldsymbol{.}\, \mathrm{tfn}(f) \leftrightarrow (\forall x_{\iota}\boldsymbol{.}\, x \in \mathrm{N} \to f(x) \in \mathrm{N})$
surjDef	$\forall f_{\iota\to(\iota\to\iota)}\boldsymbol{.}\, \forall D_{\iota\to o}\boldsymbol{.}\, \forall C_{(\iota\to\iota)\to o}\boldsymbol{.}$ $\mathrm{surj}(f, D, C) \leftrightarrow$ $(\forall y_{\iota\to\iota}\boldsymbol{.}\, y \in C \to \exists x_{\iota}\boldsymbol{.}\, x \in D \wedge y = f(x)) \wedge$ $(\forall z_{\iota}\boldsymbol{.}\, z \in D \to f(z) \in C)$
succAx1	$\forall x_{\iota}\boldsymbol{.}\, x \in \mathrm{N} \to s(x) \in \mathrm{N}$
succAx2	$\forall x_{\iota}\boldsymbol{.}\, x \in \mathrm{N} \to s(x) \neq x$
totFn	$\neg\exists g_{\iota\to(\iota\to\iota)}\boldsymbol{.}\, \mathrm{surj}(g, \mathrm{N}, \mathrm{tfn})$

Table 6: A formulation of the '**totFn**' problem

egy given in section 4 where the diagonal element is represented by the lambda expression schema $\lambda x_{\alpha}\boldsymbol{.}\, G[F(x)(x)]$. The meta-variable G has to be instantiated to $\lambda x\boldsymbol{.}\, s(x)$.

1.	1	$\vdash$	$\exists g.\mathrm{surj}(g, N, \mathrm{tfn})$	(Hyp)
2.	2	$\vdash$	$\mathrm{surj}(g_0, N, \mathrm{tfn})$	(Hyp)
3.	3	$\vdash$	$y \in N$	(Hyp)
4.	2,3	$\vdash$	$\mathrm{tfn}(g_0(y))$	(surjDef 2 3)
5.	2,3	$\vdash$	$g_0(y)(y) \in N$	(tfnDef 4 3)
6.	2,3	$\vdash$	$s(g_0(y)(y)) \in N$	(succAx1 5)
7.	2,3	$\vdash$	$\lambda z.s(g_0(z)(z))(y) \in N$	(LambdaI 6)
8.	2	$\vdash$	$[y \in N \to \lambda z.s(g_0(z)(z))(y) \in N]$	($\to I$ 7)
9.	2	$\vdash$	$\forall y.[y \in N \to \lambda z.s(g_0(z)(z))(y) \in N]$	($\forall I$ 8)
10.	2	$\vdash$	$\mathrm{tfn}(\lambda z.s(g_0(z)(z)))$	(tfnDef 9)
			Proof of 16	
11.	2	$\vdash$	$\exists y.[y \in N \wedge \lambda z.s(g_0(z)(z)) = g_0(y)]$	(surjDef 2 10)
12.	12	$\vdash$	$[y_0 \in N \wedge \lambda z.s(g_0(z)(z)) = g_0(y_0)]$	(Hyp)
13.	12	$\vdash$	$y_0 \in N$	($\wedge E$ 12)
14.	12	$\vdash$	$\lambda z.s(g_0(z)(z)) = g_0(y_0)$	($\wedge E$ 12)
15.	12	$\vdash$	$\forall y.\lambda z.s(g_0(z)(z))(y) = g_0(y_0)(y)$	(Ext-I 14)
16.	12	$\vdash$	$\lambda z.s(g_0(z)(z))(y_0) = g_0(y_0)(y_0)$	($\forall E$ 15)
			Explicit Contradiction	
17.	2,12	$\vdash$	$\mathrm{tfn}(g_0(y_0))$	(surjDef 2 13)
18.	2,12	$\vdash$	$g_0(y_0)(y_0) \in N$	(tfnDef 17 13)
19.	2,12	$\vdash$	$\neg[s(g_0(y_0)(y_0)) = g_0(y_0)(y_0)]$	(succAx2 18)
20.	12	$\vdash$	$s(g_0(y_0)(y_0)) = g_0(y_0)(y_0)$	(LambdaE 16)
21.	2,12	$\vdash$	$\perp$	($\neg E$ 19 20)
			End of explicit Contradiction	
22.	2	$\vdash$	$\perp$	($\exists E$ 11 21)
23.	1	$\vdash$	$\perp$	($\exists E$ 1 22)
24.		$\vdash$	$\neg[\exists g.\mathrm{surj}(g, N, \mathrm{tfn})]$	($\neg I$ 23)

Figure 6: A formal proof of the '**totFn**' example

A.3 The 'Aset' problem

Let us finally look at the theorem which states that the set $A = \{x \in \mathrm{N} | \mathrm{defined}(\Phi(x, x)) \wedge \Phi(x, x) > x\}$ is not recursive (an exercise from [DSW94], page 94). The formalization of this problem is listed in Table 7. In the formal proof in Figure 7 we assume that the set A is recursive and then prove a contradiction by diagonalization. The diagonal element is represented by the *if*-construct $\lambda y\boldsymbol{.}\, \mathrm{if}(p_0(y) = \mathrm{yes}, \mathrm{id}(y), s(y))$ in line 12 and the indexing property follows from the definition of the total computability **'tcompDef'**. The total computability of the diagonal element is shown in lines 6 .. 12. In the proof part from line 13 to line 18 an implicit contradiction is obtained by first applying the total computability definition in order to get an index y_0 for the diagonal element and then applying the diagonal element to this index to deliver the equality in line 18. This equality embodies the implicit contradiction which is made explicit by a case analysis in the lines 19 .. 39.

TND	$\forall x_{o}.\, x \vee \neg x$
ADef	$\forall x_{\mathrm{N}}.\, x \in A \leftrightarrow \mathrm{def}(\Phi(x,x)) \wedge \Phi(x,x) > x$
recDef	$\forall u_{\mathrm{N}\to o}.\, \mathrm{rec}(u) \leftrightarrow \exists p_{\mathrm{N}\to \mathrm{dig}}.\, [\mathrm{tcomp\_p}(p) \wedge \forall x_{\mathrm{N}}.\, x \in u \leftrightarrow p(x) = \mathrm{yes}]$
grterAx	$\forall x_{\mathrm{N}}.\, \mathrm{id}(x) \not> x \wedge s(x) > x$
idAx	$\forall x_{\mathrm{N}}.\, \mathrm{def}(id(x))$
succAx	$\forall x_{\mathrm{N}}.\, \mathrm{def}(s(x))$
tcompDef	$\forall f_{\mathrm{N}\to \mathrm{res}}.\, \mathrm{tcomp}(f) \leftrightarrow [\exists n_{\mathrm{N}}.\, f = \lambda x_{\mathrm{N}}.\, \Phi(n,x) \wedge \forall y_{\mathrm{N}}.\, \mathrm{def}(f(y))]$
tcompIf	$\forall p_{\mathrm{N}\to \mathrm{dig}}.\, \forall f_{\mathrm{N}\to \mathrm{res}}.\, \forall g_{\mathrm{N}\to \mathrm{res}}.$ $[\mathrm{tcomp\_p}(p) \wedge \forall y_{\mathrm{N}}.\, \mathrm{def}(f(y)) \wedge \forall z_{\mathrm{N}}.\, \mathrm{def}(g(z))] \to$ $\mathrm{tcomp}(\lambda x_{\mathrm{N}}.\, \mathrm{if}(p(x) = \mathrm{yes}, f(x), g(x)))$
ifDef	$\forall P_{o}.\, \forall x_{\mathrm{res}}.\, \forall y_{\mathrm{res}}.\, P \to \mathrm{if}(P,x,y) = x \wedge$ $\neg P \to \mathrm{if}(P,x,y) = y$
Aset	$\neg\mathrm{rec}(A)$

Table 7: A formulation of the '**Aset**' problem

The diagonalization proof in this example corresponds to the type of the diagonalization strategy described in section 4 where the diagonal element is represented by the lambda expression schema $\lambda x_{\alpha}.\, \mathrm{if}(R[F(x),x], Y[x], Z[x])$. The meta-variables R, Y, and Z are to be instantiated respectively to $\lambda x.\, \lambda y.\, p_0(y) = \mathrm{yes}$, $\lambda x.\, \mathrm{id}(x)$, and $\lambda x.\, s(x)$. The meta-variable U in the inverting property $R[F(i), i] \leftrightarrow U[F(i)(i)]$ should be instantiated to $\lambda x.\, \mathrm{defined}(x) \wedge x > y_0$.

1.	1	⊢	$\forall y.\text{defined}(\text{id}(y))$	(Hyp)
2.	2	⊢	$\forall y.\text{defined}(s(y))$	(Hyp)
3.	3	⊢	$\text{rec}(A)$	(Hyp)
4.	3	⊢	$\exists p.[\text{tcomp\_p}(p) \wedge \forall y.[A(y) \leftrightarrow p(y) = \text{yes}]]$	(recDef 3)
5.	5	⊢	$[\text{tcomp\_p}(p_0) \wedge \forall y.[A(y) \leftrightarrow p_0(y) = \text{yes}]]$	(Hyp)
6.	5	⊢	$\text{tcomp\_p}(p_0)$	(5)
7.	1	⊢	$\forall y.\text{defined}(\text{id}(y))$	(Abu 1)
8.	2	⊢	$\forall z.\text{defined}(s(z))$	(Abu 2)
9.	1,2	⊢	$[\forall y.\text{defined}(\text{id}(y)) \wedge \forall z.\text{defined}(s(z))]$	($\wedge I$ 7 8)
10.	1,2,5	⊢	$[\text{tcomp\_p}(p_0) \wedge [\forall y.\text{defined}(\text{id}(y)) \wedge \forall z.\text{defined}(s(z))]]$	($\wedge I$ 6 9)
11.		⊢	$[[\text{tcomp\_p}(p_0) \wedge [\forall y.\text{defined}(\text{id}(y)) \wedge \forall z.\text{defined}(s(z))]] \rightarrow$ $\text{tcomp}(\lambda y.\text{if}(p_0(y) = \text{yes}, \text{id}(y), s(y)))]$	(tcompIf)
12.	1,2,5	⊢	$\text{tcomp}(\lambda y.\text{if}(p_0(y) = \text{yes}, \text{id}(y), s(y)))$	($\rightarrow E$ 10 11)
			Proof of 18	
13.	1,2,5	⊢	$\exists n.\lambda y.\text{if}(p_0(y) = \text{yes}, \text{id}(y), s(y)) = \lambda y.\Phi(n, y)$	(tcompDef 12)
14.	14	⊢	$\lambda y.\text{if}(p_0(y) = \text{yes}, \text{id}(y), s(y)) = \lambda y.\Phi(y_0, y)$	(Hyp)
15.	14	⊢	$\forall y.\lambda y.\text{if}(p_0(y) = \text{yes}, \text{id}(y), s(y))(y) = \lambda y.\Phi(y_0, y)(y)$	(Ext-I 14)
16.	14	⊢	$\lambda y.\text{if}(p_0(y) = \text{yes}, \text{id}(y), s(y))(y_0) = \lambda y.\Phi(y_0, y)(y_0)$	($\forall E$ 15)
17.	14	⊢	$\text{if}(p_0(y_0) = \text{yes}, \text{id}(y_0), s(y_0)) = \Phi(y_0, y_0)$	(LambdaE 16)
18.	14	⊢	$\Phi(y_0, y_0) = \text{if}(p_0(y_0) = \text{yes}, \text{id}(y_0), s(y_0))$	(=Com 17)
			Case1	
19.	19	⊢	$A(y_0)$	(Case 1)
20.	5,19	⊢	$p_0(y_0) = \text{yes}$	(5 19)
21.	5,19	⊢	$\text{if}(p_0(y_0) = \text{yes}, \text{id}(y_0), s(y_0)) = \text{id}(y_0)$	(ifDef 20)
22.	5,14,19	⊢	$\Phi(y_0, y_0) = \text{id}(y_0)$	(=Trans 18 21)
23.		⊢	$\neg[\text{id}(y_0) > y_0]$	(grterAx)
24.	5,14,19	⊢	$\neg[\Phi(y_0, y_0) > y_0]$	(=Subst 22 23)
25.	5,14,19	⊢	$\neg[A(y_0)]$	(ADef 24)
26.	1,2,3,5,14,19	⊢	$\bot$	($\neg E$ 25 19)
			Case2	
27.	27	⊢	$\neg[A(y_0)]$	(Case 2)
28.	1,2,5	⊢	$\text{defined}(\lambda y.\text{if}(p_0(y) = \text{yes}, \text{id}(y), s(y))(y_0))$	(tcompDef 12)
29.	1,2,5	⊢	$\text{defined}(\text{if}(p_0(y_0) = \text{yes}, \text{id}(y_0), s(y_0)))$	(LambdaE 28)
30.	1,2,5,14	⊢	$\text{defined}(\Phi(y_0, y_0))$	(=Subst 17 29)
31.	5,27	⊢	$\neg[p_0(y_0) = \text{yes}]$	(5 27)
32.	5,27	⊢	$\text{if}(p_0(y_0) = \text{yes}, \text{id}(y_0), s(y_0)) = s(y_0)$	(ifDef 31)
33.	5,14,27	⊢	$\Phi(y_0, y_0) = s(y_0)$	(=Trans 18 32)
34.		⊢	$s(y_0) > y_0$	(grterAx)
35.	5,14,27	⊢	$\Phi(y_0, y_0) > y_0$	(=Subst 33 34)
36.	1,2,3,5,14,27	⊢	$A(y_0)$	(ADef 30 35)
37.	1,2,3,5,14,27	⊢	$\bot$	($\neg E$ 27 36)
38.		⊢	$[A(y_0) \vee \neg[A(y_0)]]$	(TND)
39.	1,2,3,5,14	⊢	$\bot$	($\vee E$ 38 26 37)
			End of Case Analysis	
40.	1,2,3,5	⊢	$\bot$	($\exists E$ 13 39)
41.	1,2,3	⊢	$\bot$	($\exists E$ 4 40)
42.	1,2	⊢	$\neg[\text{rec}(A)]$	($\neg I$ 41)

Figure 7: A formal proof of the '**Aset**' example