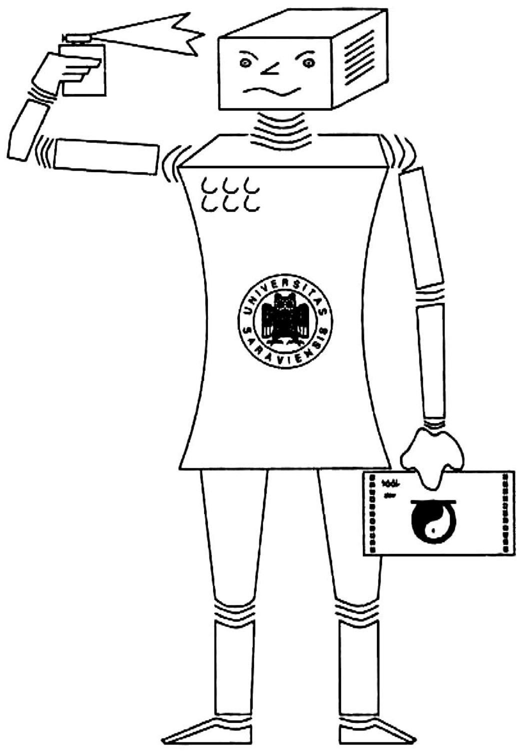

Planning Diagonalization Proofs

Lassaad Cheikhrouhou

SEKI Report SR-97-06

Planning Diagonalization Proofs

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

May 30, 1997

Abstract

This report is a first attempt of formalizing the diagonalization proof technique. We give a strategy how to systematically construct diagonalization proofs: (i) finding an indexing relation, (ii) constructing a diagonal element, and (iii) making the implicit contradiction of the diagonal element explicit. We suggest a declarative representation of the strategy and describe how it can be realized in a proof planning environment.

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 proven and the given assertions (definitions, axioms, and other theorems) in order to find a final situation, for instance $\perp$ in a resolution theorem prover. 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. Two main aspects make this approach interesting, since they provide some guidance while searching for a proof:

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. 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, e.g. methods which choose appropriate rewriting rules, these proof schemata are called (proof) methods in the terminology of CLAM [BvHHS90]. 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.

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 it 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. This report is a first attempt of formalizing the diagonalization proof technique. In the next section, we introduce the main idea of diagonalization by considering a formal proof for the Cantor theorem. Thereafter, we give a strategy how to construct systematically a diagonalization proof. This strategy is successfully applied by hand on some examples that we studied in [Che96]. We suggest a declarative representation of the strategy and describe how it can be realized in a proof planning environment.

2 Cantor Diagonalization

In order to show the main principles of proving by diagonalization, consider the Cantor theorem. For its proof, the diagonalization technique was first invented and it is therefore often called Cantor diagonalization [Kle43]. The 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 a set into its power set:

$$\forall m.\, \neg \exists f.\, \mathrm{surj}(f, m, \mathrm{pset}(m))$$

To prove the above conjecture, we assume that there is a surjective function f_0 from some set m_0 into its power set $\mathrm{pset}(m_0)$ 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 E can be enumerated in a suitable fashion.*
2. *It is possible, with the help of the enumeration, to define an object d that is different from every object in the enumeration, i.e. $d \notin E$.*

Below is the diagonalization part of the Cantor proof, where $\mathrm{pset}(m_0)$ is *the enumerated set*. This set can be enumerated with the help of *the indexing relation* f_0 and the set 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 $\mathrm{pset}(m_0)$, 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 $D = f_0(y_0)$.

TND	$\forall x_o\text{.}\, x \vee \neg x$
=-Refl	$\forall x_o\text{.}\, x = x$
=-Equiv	$\forall x_o\text{.}\, \forall y_o\text{.}\, x = y \rightarrow [x \leftrightarrow y]$
Surj-Def	$\forall f_{\iota\rightarrow(\iota\rightarrow o)}\text{.}\, \forall a_{\iota\rightarrow o}\text{.}\, \forall b_{(\iota\rightarrow o)\rightarrow o}\text{.}\, \mathrm{surj}(f,a,b) \leftrightarrow$ $\forall x_{\iota\rightarrow o}\text{.}\, x \in b \rightarrow \exists y_\iota\text{.}\, y \in a \wedge x = f(y)$
PSet-Def	$\forall a_{\iota\rightarrow o}\text{.}\, \forall x_{\iota\rightarrow o}\text{.}\, x \in \mathrm{pset}(a) \leftrightarrow x \subseteq a$
$\subseteq$-Def	$\forall a_{\iota\rightarrow o}\text{.}\, \forall b_{\iota\rightarrow o}\text{.}\, a \subseteq b \leftrightarrow \forall x_\iota\text{.}\, x \in a \rightarrow x \in b$
Cantor	$\forall m_{\iota\rightarrow o}\text{.}\, \neg\exists f_{\iota\rightarrow(\iota\rightarrow o)}\text{.}\, \mathrm{surj}(f, m, \mathrm{pset}(m))$

Table 1: A formulation of the '**Cantor**' theorem

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.

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

- the property, that the function $\lambda z\text{.}\, z \in m_0 \wedge \neg z \in f_0(z)$ 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 function $\lambda z\text{.}\, z \in m_0 \wedge \neg z \in f_0(z)$, which is assumed to be y_0, is stated in line 11,
- applying the function $\lambda z\text{.}\, z \in m_0 \wedge \neg z \in f_0(z)$ to 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 which is an element of the enumerated set that is different from every object in the enumeration. 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.

In the Cantor proof, the function f_0 binds not only the diagonal element but also each element of the enumerated set $\mathrm{pset}(m_0)$ to an element (its index) in m_0. This property follows from the surjectivity of the function f_0 from m_0 into $\mathrm{pset}(m_0)$ and is represented by the formula:

$$\forall x_{\iota\rightarrow o}\text{.}\, x \in \mathrm{pset}(m_0) \rightarrow \exists y_\iota\text{.}\, y \in m_0 \wedge x = f_0(y)$$

The indexing property provides important information for the specification of the diagonal element: its type (a functional type corresponding to the element type of $\mathrm{pset}(m_0)$), and its domain type (same type as the element type of m_0).

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

1.	1	$\vdash \exists f.\, \text{surj}(f, m_0, \text{pset}(m_0))$	(Hyp)
2.	2	$\vdash \text{surj}(f_0, m_0, \text{pset}(m_0))$	(Hyp)
3.	3	$\vdash x_1 \in \lambda z.\, [z \in m_0 \wedge \neg[z \in f_0(z)]]$	(Hyp)
4.	3	$\vdash [x_1 \in m_0 \wedge \neg[x_1 \in f_0(x_1)]]$	(LambdaE 3)
5.	3	$\vdash x_1 \in m_0$	(AndEL 4)
6.		$\vdash [x_1 \in \lambda z.\, [z \in m_0 \wedge \neg[z \in f_0(z)]] \rightarrow x_1 \in m_0]$	(ImplI 5)
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.		$\vdash \lambda z.\, [z \in m_0 \wedge \neg[z \in f_0(z)]] \in \text{pset}(m_0)$	(Pset-Def 8)
		Proof of 16	
10.	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.	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.	11	$\vdash \lambda z.\, [z \in m_0 \wedge \neg[z \in f_0(z)]] = f_0(y_0)$	(AndER 11)
13.		$\vdash y_0 \in f_0(y_0) = y_0 \in f_0(y_0)$	(=-Refl)
14.	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.	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.	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.	17	$\vdash y_0 \in f_0(y_0)$	(Hyp)
18.	11,17	$\vdash \neg[y_0 \in f_0(y_0)]$	(16 17)
19.	1,2,11,17	$\vdash \bot$	(NotE 18 17)
		Case 2	
20.	20	$\vdash \neg[y_0 \in f_0(y_0)]$	(Hyp)
21.	11	$\vdash y_0 \in m_0$	(AndEL 11)
22.	1,2,11,20	$\vdash y_0 \in f_0(y_0)$	(16 21 20)
23.	1,2,11,20	$\vdash \bot$	(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	$\vdash \bot$	(OrE 24 19 23)
		End of Case Analysis	
26.	1,2	$\vdash \bot$	(ExistsE 10 25)
27.	1	$\vdash \bot$	(ExistsE 1 26)
28.		$\vdash \neg[\exists f.\, \text{surj}(f, m_0, \text{pset}(m_0))]$	(NotI 27)
29.		$\vdash \forall m.\, \neg[\exists f.\, \text{surj}(f, m, \text{pset}(m))]$	(ForallI 28)

Figure 1: A formal proof of the '**Cantor**' theorem

In addition to these type constraints, the diagonal element must be different from each element of the enumerated set pset(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 a proposition that depends on the term $f_0(z)(z)$ which we call *the diagonal term.* In the Cantor proof, the diagonal property is represented by the conjecture $z \in f_0(z)$ which is a syntactic sugar of the diagonal term itself. 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 enumerated set pset(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.

In [Che96], we studied other proofs by diagonalization which are somewhat different from the Cantor proof in Figure 1. We exploited these differences and suggested a Di-

Strategy : `Diag`	
Declarations	—
Goal	8
Precondition	NIL
Postcondition	6, $newconst(\bar{i}, \alpha)$, $< \Vert \; differs(D, \lambda x_{\alpha}.\, x) \;\; occurs(x, \overline{D}(x))$ $differs(D(i), F(i)(i)) \;\; 5$ $inverts(\overline{D}, F(i)(i), \overline{IP}) >$
Proof Schema	1. 1 $\vdash$ $n(i) \land D = F(i)$ (Hyp) 2. 1 $\vdash$ $n(i)$ (AndEL 1) 3. 1 $\vdash$ $D = F(i)$ (AndER 1) 4. 1 $\vdash$ $\bot$ (MEC(D, IP) 2 3) 5. $\vdash$ $E(\overline{D})$ (Open) 6. $\vdash$ $\forall x_{\overline{\alpha} \to \overline{\beta}}.\, \overline{E}(x) \to \exists y_{\overline{\alpha}}.\, \overline{n}(y) \land x = \overline{F}(y)$ (Open) 7. $\vdash$ $\exists y_{\alpha}.\, n(y) \land D = F(y)$ (6 5) 8. $\vdash$ $\bot$ (ExistsE 7 4)

Figure 2: The `Diag` strategy

agonalization proof strategy that can be applied to prove the examples in [Che96]. This strategy is described in the next section.

3 A Diagonalization Proof Strategy

In this section, we give a declarative representation of the diagonalization strategy and explain how it can be applied.

3.1 The Representation of the Strategy

To represent the diagonalization proof strategy, we use the declarative framework for the representation of proof methods in Ω-MKRP[HKRS94] with some extensions. To clarify the meaning of the strategy in Figure 2, we will give some explanations to the different slots and to the important aspects of the syntax.

The *proof schema* consists of ND lines whose formulae are schematic, i.e. are propositions with *metavariables* (free variables). We distinguish *higher order* (HOV) and *first order* (FOV) *metavariables.* HOVs are represented by capital letters and can be unified with lambda expressions (by higher order unification (HOU)), whereas FOVs can only be unified with constant symbols or FOVs. For instance, in Figure 2 the metavariable n can only be instantiated to a predicate symbol that denotes a sort. A metavariable can also occur in the strategy both over-lined and un-annotated. This notation should be interpreted as an additional constraint on the metavariables: an over-lined metavariable can be bound to an object term with free variables, i.e. metavariables, and an un-annotated occurrence of a metavariable stands for a closed object term.

Each ND line has a justification which can be either open (annotated by `Open`) or closed. A closed justification consists of a tactic, e.g. a ND rule, an argument list, and a list of

ND lines (the premises). This means, the corresponding ND line can be proven from the premises by applying the tactic with the given arguments.

The *goal* of a strategy should match an open ND line on the object level, which is to be closed by the strategy. Before applying the strategy, its *precondition* must be fulfilled, where a precondition consists of ND lines from the proof schema and constraints. The ND lines have to match *support* ND *lines* of the goal and the constraints are evaluated. The *support lines* of an open ND line consist of its hypotheses and their derived consequences.

The application of the strategy would reduce the goal to new subgoals to be closed and some additional constraints to be satisfied. Both subgoals and constraints are given in the *postcondition* list.

The occurrence of constraints together with ND lines in the precondition and the postcondition list enables the representation of some control information, namely the order of closing the goals and satisfying the constraints. The ND lines and the constraints should be considered sequentially from the left to the right, if they are separated by commas in the list, and simultaneously by grouping them in a list marked with ||. Moreover, it becomes possible to interchange between goal and constraint satisfaction.

The goal of the diagonalization strategy is a contradiction, i.e. $\perp$. Its precondition list is empty [2], it can directly be applied to prove a contradiction as described in the next section. We are not concerned with the control problem, when to choose the diagonalization strategy among other applicable methods. We assume that this choice can be done either according to some control knowledge that depends on some aspects such as the theory of the problem, or interactively by the user.

3.2 Application of the Strategy

The diagonalization strategy reduces the proof of a contradiction to the main tasks: the search for an indexing relation, and the construction of a diagonal element.

3.2.1 The Search for an Indexing Relation

The indexing relation is determined by closing the subgoal 6, i.e. stating that the formula schema of the ND line 6 can be unified with a provable formula from the support lines. This amounts to the general and complex task whether a formula schema unifies a provable formula from some premises. It is difficult to obtain all provable formulae unifying the schema. We suggest therefore to restrict this by finding formulae that do not only unify the schema but also can be proven by assertion application from the premises. The methods to be used for planning assertion applications should specify whether some premise can be an assertion to prove a formula schema and should specify the resulted subgoals, i.e. the premises of this assertion application.

For instance, in the Cantor theorem, the sole premise that asserts an indexing property is the surjective definition **Surj-Def**. Its application to close the indexing schema delivers the subgoal $\mathrm{surj}(\overline{F}, \overline{n}, \overline{E})$ that can be unified with the hypothesis $\mathrm{surj}(f_0, m_0, \mathrm{pset}(m_0))$ (see line 2 in Figure 1). Thus, the metavariables F, n, and E are instantiated respectively to f_0, m_0, and $\mathrm{pset}(m_0)$.

[2]The postcondition 6 can be used as a precondition. This would restrict the application of the strategy to goals with indexing property as support. In section 3.2, it becomes clear that such a strategy is less general than the strategy in Figure 2.

Assertion application alone is not enough to determine all possible indexing properties. Sometimes, it is necessary to combine several assertions to get the right indexing property. For instance, the following formulae can be combined to determine an indexing relation:

1. $\forall x_{\overline{\alpha}\to\overline{\beta}}. \overline{E}(x) \to \exists y_{\overline{\alpha}}. \overline{S}(x,y)$
2. $\forall z_{\alpha\to\beta}. \forall u_{\alpha}. S(z,u) \to \overline{n}(u) \land z = \overline{F}(u)$

The process of proving formula schemata by assertion applications must be extended to manage such cases.

The next postcondition is the constraint $newconst(\overline{i},\alpha)$ whose evaluation binds the metavariable i to a new constant with the type set up for the metavariable α. The rest of the postcondition list should be evaluated simultaneously and leads to the construction of the diagonal element.

3.2.2 The Construction of the Diagonal Element

The diagonal element D is a function that belongs to the enumerated set E and inverts some property wrt. the diagonal term $F(i)(i)$. The first property can be stated by closing the subgoal 5. The second property can be fulfilled by some propositions which depend on β, the type of $F(i)(i)$, and on the instantiation of D. This is the reason why we represent the inverting property as a constraint. The satisfaction of the *inverts* constraint would deliver the proofs IP to the propositions that guarantee the inverting property of D.

To prevent nonsense instantiations of D, we use some restriction constraints. The constraint $occurs(x, \overline{D}(x))$ is evaluated, whenever the (partial) instantiation for the metavariable D is affected. The constraints $differs(D, \lambda x_{\alpha}. x)$ and $differs(D(i), F(i)(i))$ have only to be evaluated, when the instantiation of D becomes closed, i.e. contains no free variables. If one of these constraints cannot be satisfied, we have to backtrack by considering the next possible instantiation for D.

A vague specification of D can be given by its representation with the term schema $\lambda x_{\alpha}. \overline{G}(F(x),x)$, and by the inverting property $\overline{U}(\overline{G}(F(i),i))) \leftrightarrow \neg\overline{U}(F(i)(i))$. These schemata can be further instantiated by closing the subgoal 5 and the given inverting property. However, some alternative instantiations of these schemata would make this task easier, as they will provide more control. In the following, we consider the possible instantiations of these schemata that we obtain by investigating the examples in [Che96].

- When the diagonal term $F(i)(i)$ denotes a proposition, i.e. the metavariable β is instantiated with the type of truth values o, as in our formulation of the Cantor theorem in section 2, then it is worth considering the formula schema $\neg F(i)(i) \leftrightarrow \overline{D}(i)$ as inverting property.

- If the diagonal term does not denote a proposition, two important possible instantiations of the metavariable are distinguished:

  1. The value of $D(x)$ can be defined according to some condition $U(F(x),x)$. $D(x)$ equals $Y(x)$, if $U(F(x),x)$ holds, and it is $Z(x)$ otherwise. Thus, D must be instantiated by the schema $\lambda x_{\alpha}. if(\overline{U}(F(x),x), \overline{Y}(x), \overline{Z}(x))$, where the constraint $differs(Y(x), Z(x))$ must hold.

The inverting property of D would involve the arguments of the *if*-construct, i.e. the condition, the *then*-term, and the *else*-term. The inversion of the term $F(i)(i)$ is obtained, if the following subgoals can be proven:
When $U(F(i),i)$ is instantiated to a proposition of the form $Q(F(i)(i))$,i.e. that depends on the term $F(i)(i)$, then we consider the subgoals $\neg Q(\overline{Y}(i))$, and $Q(\overline{Z}(i))$. Otherwise, the condition term $U(F(i),i)$ must be attributed to a proposition that depends on the term $F(i)(i)$. We consider therefore the subgoal $\overline{U}(F(i),i) \rightleftharpoons \overline{Q}(F(i)(i))$, where the connector $\rightleftharpoons$ can be either a left-to-right implication or an equivalence symbol. If $\rightleftharpoons$ can be instantiated to an equivalence symbol, then the subgoals $\neg Q(\overline{Y}(i))$, and $Q(\overline{Z}(i))$ must be considered. Otherwise, we consider the subgoals $\neg Q(\overline{Y}(i))$, $\neg\overline{U}(F(i),i) \rightarrow \overline{R}(F(i)(i))$, and $\neg R(\overline{Z}(i))$.

2. After proving the subgoal 5, the metavariable D can be instantiated with the function $\lambda x_\alpha. C(F(x),x)$ which is different from an *if*-construct. The inverting property of D can be reached by closing one of the following subgoals: the inequality $C(F(i),i) \neq F(i)(i)$ and the formula schema $\overline{U}(C(F(i),i)) \leftrightarrow \neg\overline{U}(F(i)(i))$.

While closing subgoals represented by formula schemata, metavariables are progressively instantiated. This is done by middle out reasoning (MOR) [KBB93].

3.3 Middle Out Reasoning

We are concerned here with subgoals represented by formula schemata. In general, propositions with *rigid heads*, i.e. represented by an application $P(t_1,..,t_n)$ whose function P is fully instantiated, are not problematic. They can be closed by assertion application or by simplification tactics.

Applying an assertion to a formula schema involves the use of HOU. However, this process can deliver infinite number of solutions [Hue75]. We solve this problem by considering only the n first solutions. Moreover, we use heuristics (similar say to the use of HO colored unification in linguistic analysis [GK96]) to prevent nonsense solutions and to prefer the promising ones. For instance, the projection solution $\lambda x. x$ of the problem $Y(a) = a$ should be considered before the imitation solution $\lambda x. a$.

Moreover, there are conflict situations, where many assertions are applicable to close a formula schema. To make the reasonable choice, we use control heuristics based on the following preference criteria:

- the metavariable instantiations caused by the assertion application,
- the number of the new subgoals resulted from applying the assertion,
- and whether the assertion application involves *new inserted hypotheses*, i.e. hypotheses that do not belong to the original proof assumptions.

We prefer heuristically the assertion that inserts the least subgoals, instantiates the most metavariables, and involves the most new inserted hypotheses.

Simplification tactics may not be applied to *critical goals*, where a goal is critical, iff its splitting by a simplifiction tactic leads to a *flexible subgoal* and/or a flexible hypothesis, i.e. a formula with a flexible head. A flexible hypothesis can assert every subgoal, and a flexible subgoal follows from every hypothesis. For instance, the ND rule `ImpI` is not allowed to be applied as a simplification tactic to the goal schema $p \to \overline{D}(i)$, because this would deliver the flexible subgoal $\overline{D}(i)$. A critical subgoal is suspended at first and we consider simultaneous subgoals until the involving metavariable is instantiated.

In general, assertion application alone is not enough to prevent critical subgoals. It is possible to obtain a proof plan with many suspended critical subgoals. To raise such a blockade, we use techniques from [Ble77], where instantiations of set variables in higher order theorems are suggested to obtain a first order theorem that can be proven by an automated theorem prover. In this procedure, HOVs that occur as heads of atoms are interpreted as sets. According to the position of the corresponding atom wrt. other subformulae in the theorem, a HOV is associated to the maximal possible set. If this HOV occurs several times as the head of an atom in the theorem, then it is instantiated with the intersection of the sets resulted by considering each occurrence. For instance, if we had the goal $\forall x(\overline{D}(x) \to m_0(x)) \land (\neg f_0(x)(x) \to \overline{D}(x))$ then we would obtain for the first occurrence the set $\{x : m_0(x)\}$ and for the second occurrence the set $\mathcal{U}$, which denotes the whole individual set. The intersection should be clearly the set $\{x : m_0(x)\}$.

For each suspended critical subgoal, Bledsoe's procedure delivers the appropriate set to the metavariable, that causes the blockade. Each of these metavariables is instantiated to the intersection of the associated sets. After that, at least one subgoal will become non-critical and the proof planning process can be continued.

In the Cantor example, the subgoal $\mathrm{pset}(m_0)(\overline{D})$ can be reduced to the subgoal $\overline{D}(a) \to m_0(a)$ after applying first the power set definition **PSet-Def**, then the subset definition **$\subseteq$-Def**, and finally the tactic `ForallI`. Since the resulted subgoal is critical, we consider next the inverting property $\neg f_0(i)(i) \leftrightarrow \overline{D}(i)$. No support line can be found to deduce a term of the form $\neg f_0(i)(i) \leftrightarrow \bullet$, where $\bullet$ stands for any proposition. This subgoal is therefore critical too. Bledsoe procedure computes the set $\{x : m_0(x)\}$ for the former critical subgoal and the set $\{x : \neg f_0(x)(x)\}$ for the latter subgoal. Now, we must conjunct the two sets for instantiating D and check that the resulted instantiation $\lambda x_\iota.\, m_0(x) \land \neg f_0(x)(x)$ satisfies the two suspended subgoals. The first subgoal $(m_0(a) \land \neg f_0(a)(a)) \to m_0(a)$ is obviously satisfiable. The second subgoal $\neg f_0(i)(i) \leftrightarrow (m_0(i) \land \neg f_0(i)(i))$ can be closed straightforward using the hypothesis $m_0(i)$.

In the specification of the diagonal element, we pretended that the HOV D can be instantiated to an if-term, which is commonly used as the conditional expression in programming languages. It is therefore convenient to use techniques of program synthesis while considering subgoals, that involve the metavariable D. For instance, the subgoal $p \to r(\overline{D})$ can be reduced, after unifying D with $\lambda x.\, if(p, \overline{Y}(x), \overline{Z}(x))$, to $r(\overline{Y})$.

While constructing the diagonal element D, there can be conflict situations, where both simplification tactics and techniques of program synthesis can be applied. In such situations, we prefer the application of program synthesis techniques. We try first techniques of program synthesis for instantiating D, then simplification tactics, thereafter assertion application, and finally Bledsoe method. While searching for the indexing relation, we attempt first assertion application, then simplification tactics, and at last the Bledsoe procedure.

3.4 The Execution of the Strategy

If the application of the strategy succeeds, i.e. its postcondition is satisfied, we can execute it. This means the corresponding instantiated proof schema is inserted into the ND proof. Lines of the proof schema, that are not justified by ND rules, can further be expanded by applying their justification tactics. The expansion of the line number 4 in the proof schema of the diagonalization strategy is interesting. This corresponds to making the implicit contradiction of the diagonal element explicit. The tactic MEC generates a contradiction proof on ND level according to the instantiation of the diagonal element and of the proven properties that guarantee the inversion of the diagonal term $F(i)(i)$.

For instance, in the Cantor theorem, D is instantiated to the lambda term $\lambda x_\iota . m_0(x) \wedge \neg f_0(x)(x)$ and the inverting property corresponds to the proposition $\neg f_0(i)(i) \leftrightarrow (m_0(i) \wedge \neg f_0(i)(i))$. In such situation, the tactic MEC uses the conjectures $m_0(i)$ and $\lambda x . m_0(x) \wedge \neg f_0(x)(x) = f_0(i)$, i.e. the instantiated formulae of line 2 and line 3 in Figure 2, to deliver the following expansion of line 4:

4	$\Delta, 1$	$\vdash \bot$	(OrE L_1 L_2 L_3)
L_1		$\vdash f_0(i)(i) \vee \neg f_0(i)(i)$	(TND)
C_1	C_1	$\vdash f_0(i)(i)$	(Hyp)
L_2	$\Delta, 1, C_1$	$\vdash \bot$	(NotE L_4 C_1)
L_4	$\Delta, 1, C_1$	$\vdash \neg f_0(i)(i)$	(=Subst 3 L_5)
L_5	$\Delta, 1, C_1$	$\vdash \neg(\lambda x . m_0(x) \wedge \neg f_0(x)(x))(i)$	(LambdaI L_6)
L_6	$\Delta, 1, C_1$	$\vdash \neg(m_0(i) \wedge \neg f_0(i)(i))$	(NotPop L_7)
L_7	$\Delta, 1, C_1$	$\vdash \neg m_0(i) \vee f_0(i)(i)$	(OrIR C_1)
C_2	C_2	$\vdash \neg f_0(i)(i)$	(Hyp)
L_3	$\Delta, 1, C_2$	$\vdash \bot$	(NotE C_2 L_8)
L_8	$\Delta, 1, C_2$	$\vdash f_0(i)(i)$	(=Subst 3 L_9)
L_9	$\Delta, 1, C_2$	$\vdash (\lambda x . m_0(x) \wedge \neg f_0(x)(x))(i)$	(LambdaI L_{10})
L_{10}	$\Delta, 1, C_2$	$\vdash m_0(i) \wedge \neg f_0(i)(i)$	(AndI 2 C_2)

In other instantiations of D, e.g. an if-term, the tactic MEC uses the proofs IP for the inverting propositions, that are closed during the satisfaction of the constraint *inverts*, in the expansion of line 4.

4 Conclusion and Future Work

In this report, we suggest a systematic way for constructing diagonalization proofs. We propose some extensions of the declarative environment for method representation presented in [HKRS94] to capture the main steps in a diagonalization proof schema and allow planning partial instantiated goals. The success of the diagonalization proof strategy depends mainly on two tasks: the existence of an indexing property and the existence of a function (the diagonal element) that satisfies an inverting property relative to the diagonal term $F(i)(i)$ (i is the index of the diagonal element.) and belongs to the enumerated set E. We describe how to achieve each of these tasks.

The suggested proof strategy can be applied successfully by hand to the examples in [Che96]. Next, we want to extend the proof planning framework in Ω-MKRP to implement this strategy. Moreover, the process of assertion application to prove the indexing property and that of middle out reasoning to construct the diagonal element have to be further specified and implemented.

Central 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. To answer these questions, more examples and especially other problem descriptions should be empirically examined.

References

[Ble77] W. W. Bledsoe. A Maximal Method for Set Variables in Automatic Theorem Proving. Memo ATP-33, Math. Dept., Univ. of Texas, Februar 1977.

[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.

[Che96] Lassaad Cheikhrouhou. The Mechanization of the Diagonalization Proof Strategy. SEKI Report SR-96-14, Fachbereich Informatik, Universität des Saarlandes, Im Stadtwald, Saarbrücken, Germany, 1996.

[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.

[HKRS94] Xiaorong Huang, Manfred Kerber, Jörn Richts, and Arthur Sehn. Planning Mathematical Proofs with Methods. *Journal of Information Processing and Cybernetics, EIK*, **30**(5-6):277–291, 1994.

[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.

[Hue75] Gérard Huet. A Unification Algorithm for the Typed λ-Calculus. *Theoretical Computer Science*, **1**:27–57, 1975.

[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.