Excluded Middle without Definite Descriptions in the Theory of Constructions

Jonathan P. Seldin*
Department of Mathematics
Concordia University
Montréal, Québec, Canada
seldin@alcor.concordia.ca

November 15, 1998

1 Introduction

In his posting to the TYPES network [3], Pottinger shows that if excluded middle and definite descriptions are added to Coquand's calculus of constructions, then any two terms in a small type (i.e., a type in Prop) are equal (in the sense of Leibniz equality). This conclusion is called "proof degeneracy". Although in general proof degeneracy does not imply inconsistency, it is still undesirable because it means that all terms in small types are identified by the logic, and hence there is no representation of a data type with more than one element.

Coquand [1] showed by a model theoretic proof that excluded middle in the calculus of constructions is consistent. Here I show a stronger result, that excluded middle without definite descriptions does not imply proof degeneracy. My method is proof theoretic, a variant of the $\neg\neg$-interpretation, and part of it provides an alternative proof of Coquand's consistency result. My main result will follow from a proof that an environment which implies

*The research which produced the information contained in this document was sponsored in part by the U.S. Air Force Systems Command, Rome Air Development Center, Griffiss AFB, New York 13441-5700 under Contract No. F30602-85-C-0098, in part by grant EQ41840 from the program Fonds pour la Formation de Chercheurs et l'aide a la Recherche (F.C.A.R.) of the Quebec Ministry of Education, and in part by grant OGP0023391 from the Natural Sciences and Engineering Research Council of Canada.

classical arithmetic is consistent. Since one of the Peano axioms asserts that zero is not a successor, proof degeneracy would imply the inconsistency of this environment.

Coquand [1] also proved that excluded middle and strong sums imply proof degeneracy. Garrel Pottinger has pointed out to me (private communication) that the strong sums, when interpreted under the Curry-Howard isomorphism, have the disjunction property. Since this property is known to be characteristic of constructive logic and incompatible with classical logic, this result of Coquand is really a confirmation of what we should expect of classical logic. The result of Pottinger [3], on the other hand, is unwelcome, since both excluded middle and definite descriptions are desirable in some circumstances. The result proved here shows that we are more likely to have to give up definite descriptions than excluded middle.

An ASCII version of this document was circulated on the TYPES mailing list in September 1990. A LaTeX version was prepared as the result of a request from some readers of TYPES in November 1990. I intend to incorporate this material in [6].

I would like to thank Garrel Pottinger for his helpful comments and suggestions.

2 TOC

I will follow Pottinger [3] in referring to the theory of constructions as TOC. The version of the theory of constructions used here will be the one of Seldin [5], which is more general (in its equality rules) than the versions used by Coquand and Pottinger. More specifically, I will use the formulation of TOC which is called TAC in Seldin [5, Chapter 4].

I will adopt most of the conventions of Pottinger [3]. However, I will not use parameters, so lower case Greek letters will be available for other uses. In particular, $\boldsymbol{\kappa}$ and $\boldsymbol{\kappa}'$ will each be either Prop or Type. Furthermore, '$[N/x]M$' will denote the result of substituting N for x in M.[1] The version of TOC used here is a natural deduction system in the sense of Gentzen and Prawitz. It has one axiom, namely

$$(\mathsf{P\ T}) \qquad\qquad \mathsf{Prop} : \mathsf{Type}$$

The rules are as follows:

[1] This was Curry's notation, and I am used to it.

($\boldsymbol{\kappa\kappa}'$Formation) *If x does not occur free in A or in any undischarged assumption,*

$$\frac{A : \boldsymbol{\kappa} \qquad \begin{array}{c}[x : A] \\ B : \boldsymbol{\kappa}'\end{array}}{(\forall x : A)B : \boldsymbol{\kappa}'},$$

(Eq$'\boldsymbol{\kappa}$)

$$\frac{A : \boldsymbol{\kappa} \qquad A =_* B}{B : \boldsymbol{\kappa}},$$

($\forall$ e)

$$\frac{M : (\forall x : A)B \qquad N : A}{MN : [N/x]B},$$

($\forall$ $\boldsymbol{\kappa}$i) *If x does not occur free in A or in any undischarged assumption*

$$\frac{\begin{array}{c}[x : A] \\ M : B\end{array} \qquad A : \boldsymbol{\kappa}}{\lambda x : A \,.\, M : (\forall x : A)B},$$

(Eq$''$)

$$\frac{M : A \qquad A =_* B}{M : B}.$$

Seldin [5, Theorem 4.14] proves the strong normalization theorem for deductions in this formulation. Since this result refers to a Prawitz-style reduction (see Prawitz [4]), it has as a consequence that in any normalized deduction whose conclusion is the conclusion of an inference by ($\forall$e), the main branch (which I am writing as the left branch in tree diagrams) consists of inferences by ($\forall$e) and (Eq$''$) and no other rules.

3 Additional assumptions in TOC

The strong normalization theorem for deductions in TOC implies the consistency of the basic system. However, environments Γ may still be inconsistent in the sense that

$$\Gamma \;\vdash\; M : \mathsf{void}$$

for some term M. To prove an environment Γ consistent, we need to show that this cannot happen. Note that since void is $(\forall x : \mathsf{Prop})x$, from

$$\Gamma \;\vdash\; M : \mathsf{void},$$

it follows for a variable x not free in Γ that

$$\Gamma, x : \mathsf{Prop} \vdash Mx : x.$$

Furthermore, the last inference in this latter deduction cannot be by $(\forall\boldsymbol{\kappa}\mathrm{i})$. Hence, the last inference must be by $(\forall\mathrm{e})$.[2] Furthermore, the formula at the top of the main (left) branch cannot be discharged; for the only rule which would allow the discharge of an assumption over the left premise is ($\boldsymbol{\kappa\kappa}'$Formation), and this rule cannot occur in the left branch if the type of the conclusion is a variable. Hence, to show that there is no deduction of

$$\Gamma, x : \mathsf{Prop} \vdash Mx : x,$$

it is often sufficient to show that no formula of Γ can occur at the top of the left branch.

One class of environments which satisfies this requirement, and is hence consistent, is given by the following definition:

Definition 1 (Strongly consistent environment) Let Γ be a valid environment (in the sense of [5]) of the form

$$x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n.$$

For each $i = 1, 2, \ldots, n$, let A_i be

$$(\forall y_{i1} : B_{i1})(\forall y_{i2} : B_{i2}) \ldots (\forall y_{im_i} : B_{im_i})S_i,$$

where S_i does not convert to the form $(\forall x : A)B$. This S_i is called the *tail* of the type A_i. (It follows that each S_i converts to the form $z_i M_{i1} M_{i2} \ldots M_{ip}$, where z_i is either an atomic constant or a variable.) Then Γ is *strongly consistent* if for each z_i which is a variable (and is hence one of the x_j or y_{jk}), if the number of universal prefixes of A_i is greater than 0, then the tail of the type of z_i (which is the tail of A_j or B_{jk}) does not convert to Prop. (Since we are dealing with terms with normal forms, convertibility is decidable here.)

It is not difficult to see that it follows from this definition that no formula of a strongly consistent environment Γ can occur at the top of the left branch of a deduction of the form

$$\Gamma, \Gamma' \vdash M : x.$$

this proves the following immediate consequence of Definition 1:

[2] Or by (Eq''), but in this case the same conclusion follows.

Theorem 1 *A strongly consistent environment is consistent.*

Note how weak this result is: no type in a strongly consistent environment can be the universal closure of any type of the form void, $A \wedge B$, $A \vee B$, $(\exists x : A)B$, or $\mathsf{Q}AMN$ (see [5, Chapter 5] for the definitions of these types).

If the proofs of [5, Theorem 5.2 and Corollary 5.2.1] are properly rewritten, they show the consistency of an environment of the form Γ, Γ', where Γ is as in the theorem and Γ' is strongly consistent. The same sort of modification can be made in [5, Theorem 5.3 and Corollary 5.3.1].

Since $\neg A$ is $A \to \mathsf{void}$, no formula of the form $\neg A$ can occur as the type of an assumption in a strongly consistent environment. However, certain environments with negations of formulas as types can be proved consistent:

Definition 2 (Strongly negation consistent environment) Let Γ_0 be a strongly consistent environment. Let Γ_1 consist of assumptions of the form $u : \neg B$, where, under the assumptions of Γ_0, B is a small simple type but B does not convert to the type of an assumption in Γ_0. Let Γ_2 consist of assumptions of the form $v : \neg\neg B$, where, under the assumptions of Γ_0, B is a small simple type and where $\neg B$ does not convert to the type of an assumption in Γ_1 (but B may convert to the type of an assumption in Γ_0). If Γ is $\Gamma_0, \Gamma_1, \Gamma_2$, then Γ is said to be *strongly negation consistent*.

Theorem 2 *A strongly negation consistent environment is consistent.*

Remark Clearly, if B converts to C, then $u : \neg B, v : C \vdash uv : \mathsf{void}$. What this theorem says is that if B is a small simple type, this is essentially the only way to get a contradiction.

Proof Suppose Γ is strongly negation consistent and suppose that for some term M

$$\Gamma \vdash M : \mathsf{void}.$$

then for a variable w which is not free in Γ, we have for some term M'

$$\Gamma, w : \mathsf{Prop} \vdash M' : w.$$

Normalize this deduction and let the result be D. Suppose that there is no proper subdeduction of D which proves either

$$\Gamma' \vdash M'' : \mathsf{void} \qquad \text{or} \qquad \Gamma', w : \mathsf{Prop} \vdash M'' : w$$

for any other strongly negation consistent Γ'; otherwise we can begin with this proper subdeduction. (Here proper subdeduction means that there is more difference than the one inference necessary to go back and forth between a conclusion whose type is void and one whose type is w.) Now the last inference in D which differs from (Eq$''$) cannot be ($\forall\boldsymbol{\kappa}$i); thus it must be ($\forall$e). It follows that the left branch of D consists of inferences by ($\forall$e) and (Eq$''$), and hence the top of the left branch is not discharged. This assumption at the top of the left branch must be in Γ_1 or Γ_2.

Case 1. It is in Γ_1. Then it is $u : \neg B$ for a small simple type B not convertible to a type in Γ_0 or Γ_2, and D is

$$\dfrac{u : \neg B \qquad \begin{array}{c}\underbrace{w : \mathsf{Prop}, u : \neg B} \\ D_1 \\ M'' : B\end{array}}{uM'' : \mathsf{void}}\;(\forall\mathrm{e})$$
$$\begin{array}{c} D_2 \\ M' : w. \end{array}$$

Now clearly no assumption of D_1 is discharged in D_2. Hence, since B is a simple type, the top of the left branch of D_1 must be in Γ_1 or Γ_2. Hence, D_1 is

$$\dfrac{u' : \neg B' \qquad \begin{array}{c}\underbrace{w : \mathsf{Prop}, u : \neg B, u' : \neg B'} \\ D_3 \\ M''' : B'\end{array}}{u'M''' : \mathsf{void}}\;(\forall\mathrm{e})$$
$$\begin{array}{c} D_4 \\ M'' : B, \end{array}$$

where B' is simple or the negation of a simple type. But then

$$\dfrac{u' : \neg B' \qquad \begin{array}{c}\underbrace{w : \mathsf{Prop}, u : \neg B, u' : \neg B'} \\ D_3 \\ M''' : B'\end{array}}{u'M''' : \mathsf{void}}\;(\forall\mathrm{e})$$

is a proper subdeduction of D contradicting the assumption about D. Hence, the top of the left branch of D is not in Γ_1.

Case 2. It is in Γ_2. Then it is $u : \neg\neg B$, where B is a small simple type, and D is

$$
\dfrac{u : \neg\neg B \qquad \begin{array}{c} \underbrace{w : \mathsf{Prop}, u : \neg\neg B} \\ D_1 \\ M'' : \neg B \end{array}}{uM'' : \mathsf{void}} \ (\forall \mathrm{e})
\\
D_2
\\
M' : w.
$$

the argument of Case 1 shows that the last inference in D_1 which differs from (Eq$''$) is not ($\forall$e), so it must be by ($\forall$Pi), and D_1 is

$$
\dfrac{\begin{array}{c} \underbrace{w : \mathsf{Prop}, u : \neg\neg B, \overset{1}{[v : B]}} \\ D_3 \\ M''' : \mathsf{void} \end{array} \qquad \begin{array}{c} D_4 \\ B : \mathsf{Prop} \end{array}}{\lambda v : B \, . \, M''' : \neg B,} \ (\forall \mathsf{Pi} - 1)
$$

where M'' converts to $\lambda v : B \, . \, M'''$. But then D_3 is a proper subdeduction of D contradicting our assumption. Hence, the assumption at the top of the left branch of D is not in Γ_2.

This shows that Γ is consistent. ∎

4 Classical logic

Any environment with the assumption

$$\mathsf{cl} : (\forall u : \mathsf{Prop})(\neg\neg u \rightarrow u)$$

will imply classical logic. On the other hand, this assumption cannot occur in a strongly consistent environment.

To simplify the notation, let CL be an abbreviation for

$$(\forall u : \mathsf{Prop})(\neg\neg u \rightarrow u).$$

We want each occurrence of $\mathsf{cl} : \mathsf{CL}$ as an assumption to occur in a subdeduction of the form

$$\dfrac{\dfrac{\mathsf{cl} : \mathsf{CL} \qquad \overset{D_1}{A : \mathsf{Prop}}}{\mathsf{cl}A : \neg\neg A \to A}\,(\forall\mathrm{e}) \qquad \overset{D_2}{M : \neg\neg A}}{\mathsf{cl}AM : A.}\,(\forall\mathrm{e})$$

This is not a difficult restriction to satisfy, since we can replace

$$\dfrac{\mathsf{cl} : \mathsf{CL} \qquad \overset{D_1}{A : \mathsf{Prop}}}{\mathsf{cl}A : \neg\neg A \to A,}\,(\forall\mathrm{e})$$

where the conclusion is not a major premise for $(\forall\mathrm{e})$, by

$$\dfrac{\dfrac{\dfrac{\mathsf{cl} : \mathsf{CL} \quad \overset{D_1}{A : \mathsf{Prop}}}{\mathsf{cl}A : \neg\neg A \to A}\,(\forall\mathrm{e}) \qquad \overset{1}{[x : \neg\neg A]}}{\mathsf{cl}Ax : A}\,(\forall\mathrm{e}) \qquad \overset{D_2}{\neg\neg A : \mathsf{Prop}}}{\lambda x : \neg\neg A\ .\ \mathsf{cl}Ax : \neg\neg A \to A,}\,(\forall\mathsf{Pi}-1)$$

where D_2 is

$$\dfrac{\dfrac{\overset{D_1}{A : \mathsf{Prop}} \qquad \overset{D_\mathsf{v}}{\mathsf{void} : \mathsf{Prop}}}{\neg A : \mathsf{Prop}}\,(\mathsf{PP}\mathrm{Formation}-\mathrm{v}) \qquad \overset{D_\mathsf{v}}{\mathsf{void} : \mathsf{Prop}}}{\neg\neg A : \mathsf{Prop},}\,(\mathsf{PP}\mathrm{Formation}-\mathrm{v})$$

and where D_v is

$$\dfrac{\mathsf{Prop} : \mathsf{Type} \qquad \overset{n}{[x : \mathsf{Prop}]}}{\mathsf{void} : \mathsf{Prop};}\,(\mathsf{TP}\mathrm{Formation}-n)$$

also, if $\mathsf{cl} : \mathsf{CL}$ is not the major premise for an inference by $(\forall\mathrm{e})$, then we can replace it by

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{cl} : \mathsf{CL} \quad \overset{2}{[x : \mathsf{Prop}]}}{\mathsf{cl}x : \neg\neg x \to x}\,(\forall\mathrm{e}) \qquad \overset{1}{[y : \neg\neg x]}}{\mathsf{cl}xy : x}\,(\forall\mathrm{e}) \qquad \begin{matrix} 2 \\ [x : \mathsf{Prop}] \\ D_1 \\ \neg\neg x : \mathsf{Prop} \end{matrix}}{\lambda y : \neg\neg x\ .\ \mathsf{cl}xy : \neg\neg x \to x}\,(\forall\mathsf{Pi}-1) \qquad \mathsf{Prop} : \mathsf{Type}}{\lambda x : \mathsf{Prop}\ .\ \lambda y : \neg\neg x\ .\ \mathsf{cl}xy : \mathsf{CL},}\,(\forall\mathsf{Ti}-2)$$

where D_1 is

$$\dfrac{\dfrac{x : \mathsf{Prop} \quad \overset{D_{\mathsf{v}}}{\mathsf{void} : \mathsf{Prop}}}{\neg x : \mathsf{Prop}}\,(\mathsf{PPFormation} - \mathrm{v}) \qquad \overset{D_{\mathsf{v}}}{\mathsf{void} : \mathsf{Prop}}}{\neg\neg x : \mathsf{Prop}.}\,(\mathsf{PPFormation} - \mathrm{v})$$

A deduction in which both of these replacements have been made systematically in all possible places will be called *prepared*.[3]

Now consider in a prepared deduction a subdeduction of the form

$$\dfrac{\dfrac{\mathsf{cl} : \mathsf{CL} \quad \overset{D_1}{A : \mathsf{Prop}}}{\mathsf{cl}A : \neg\neg A \to A}\,(\forall \mathrm{e}) \qquad \overset{D_2}{M : \neg\neg A}}{\mathsf{cl}AM : A.}\,(\forall \mathrm{e})$$

Since there is a subdeduction of $A : \mathsf{Prop}$, A is a type; hence, it is either simple or compound.

The strategy is to follow the idea of [4] for classical logic, and eliminate occurrences of subdeductions like those above in which A is compound. Thus, assume that A is $(\forall y : B)C$. We need a lemma:

Lemma 1 *Let* Γ *be a valid environment, and suppose that*

$$\Gamma \vdash A : X,$$

where $A =_* (\forall y : B)C$ *and* $X =_* \boldsymbol{\kappa}$. *Then*

$$\Gamma \vdash B : \boldsymbol{\kappa}' \qquad \textit{and} \qquad \Gamma, y : B \vdash C : \boldsymbol{\kappa}.$$

Proof A straightforward induction on the normalized deduction of $\Gamma \vdash A : X$. ■

Now by this lemma and D_1 above, it follows that there are deductions

$$\begin{array}{c} D_3 \\ B : \boldsymbol{\kappa}, \end{array} \qquad \begin{array}{c} y : B \\ D_4(y) \\ C : \mathsf{Prop}. \end{array}$$

[3]Note that in preparing a deduction, we replace terms in which cl occurs by terms to which they are η-convertible.

Then we can transform

$$\dfrac{\dfrac{\mathsf{cl}:\mathsf{CL} \quad \begin{matrix}D_1\\ A:\mathsf{Prop}\end{matrix}}{\mathsf{cl}A:\neg\neg A\to A}\,(\forall\mathrm{e}) \qquad \begin{matrix}D_2\\ M:\neg\neg A\end{matrix}}{\mathsf{cl}AM:A}\,(\forall\mathrm{e})$$

into the following:

$$\dfrac{\dfrac{\dfrac{\mathsf{cl}:\mathsf{CL} \quad \begin{matrix}1\\ [y:B]\\ D_4\\ C:\mathsf{Prop}\end{matrix}}{\mathsf{cl}C:\neg\neg C\to C}\,(\forall\mathrm{e}) \qquad \begin{matrix}\underbrace{\begin{matrix}D_3\\ B:\boldsymbol{\kappa},\end{matrix}\quad \begin{matrix}D_5\\ \lambda y:B\,.\,C:(\forall y:B)\mathsf{Prop},\end{matrix}\quad \begin{matrix}D_2\\ M:\neg\neg A,\end{matrix}\quad \begin{matrix}1\\ [y:B]\end{matrix}}\\ D_6(B,\lambda y:B\,.\,C,M,y)\\ \mathcal{T}(B,\lambda y:B\,.\,C,M,y):\neg\neg C\end{matrix}}{\mathsf{cl}C\mathcal{T}(B,\lambda x:B\,.\,C,M,N):C}\,(\forall\mathrm{e}) \qquad \begin{matrix}D_3\\ B:\boldsymbol{\kappa}\end{matrix}}{\lambda y:B\,.\,\mathsf{cl}C\mathcal{T}(B,\lambda x:B\,.\,C,M,N):(\forall x:B)C,}\,(\forall\boldsymbol{\kappa}\mathrm{i}-1)$$

where D_5 is

$$\dfrac{\begin{matrix}2\\ [y:B]\\ D_4(y)\\ C:\mathsf{Prop}\end{matrix} \qquad \begin{matrix}D_3\\ B:\boldsymbol{\kappa}\end{matrix}}{\lambda y:B\,.\,C:(\forall y:B)\mathsf{Prop},}\,(\forall\boldsymbol{\kappa}\mathrm{i}-2)$$

$\mathcal{T}(u',v',u,v)$ is

$$\lambda w:\neg v'v\,.\,u(\lambda y:(\forall x:u')(v'x)\,.\,w(yv)),$$

and $D_6(u',v',u,v)$ is the obvious normalized deduction of

$$u':\boldsymbol{\kappa}, v':(\forall x:u')\mathsf{Prop}, u:\neg\neg(\forall x:u')(v'x), v:u' \;\vdash\; \mathcal{T}(u',v',u,v):\neg\neg v'v.$$

In the special case in which $A =_* \mathsf{void}$, we have a special transformation: we replace

$$\dfrac{\dfrac{\mathsf{cl}:\mathsf{CL} \qquad \begin{matrix}D_{\mathsf{v}}\\ \mathsf{void}:\mathsf{Prop}\end{matrix}}{\mathsf{cl}\ \mathsf{void}:\neg\neg\mathsf{void}\to\mathsf{void}}\,(\forall\mathrm{e}) \qquad \begin{matrix}D_1\\ M:\neg\neg\mathsf{void}\end{matrix}}{\mathsf{cl}\ \mathsf{void}M:\mathsf{void}}\,(\forall\mathrm{e})$$

by

$$\dfrac{\begin{matrix}D_1\\ M:\neg\neg\mathsf{void}\end{matrix} \qquad \dfrac{\begin{matrix}1\\ [x:\mathsf{void}]\end{matrix} \qquad \begin{matrix}D_{\mathsf{v}}\\ \mathsf{void}:\mathsf{Prop}\end{matrix}}{\lambda x:\mathsf{void}\,.\,x:\neg\mathsf{void}}\,(\forall\mathsf{Pi}-1)}{M(\lambda x:\mathsf{void}\,.\,x):\mathsf{void}.}\,(\forall\mathrm{e})$$

If we repeatedly apply these transformations to a deduction, we will eventually reach a point at which in all occurrences of a part of a deduction of the form

$$\dfrac{\dfrac{\mathsf{cl} : \mathsf{CL} \quad \begin{matrix} D_1 \\ A : \mathsf{Prop} \end{matrix}}{\mathsf{cl}A : \neg\neg A \to A}\,(\forall\mathrm{e}) \qquad \begin{matrix} D_2 \\ M : \neg\neg A \end{matrix}}{\mathsf{cl}AM : A,}\,(\forall\mathrm{e})$$

A is a simple type. Of course, this will have replaced terms of the form $\mathsf{cl}AM$ for compound types A of the form $(\forall y : B)C$ by

$$\lambda y : B \;.\; \mathsf{cl}C\mathcal{T}(B, \lambda x : B \;.\; C, M, N)$$

and $\mathsf{cl}\,\mathsf{void}M$ by $M(\lambda u : \mathsf{void} \;.\; u)$. If we repeat these replacements, we will eventually eliminate all occurrences of the assumption $\mathsf{cl} : \mathsf{CL}$ as the major premise for an inference by $(\forall\mathrm{e})$ in which the term of the minor premise is a compound type. We can go on to eliminate all occurrences of this assumption by changing some small simple types B to $\neg\neg B$; this will convert a deduction of $\Gamma, \mathsf{cl} : \mathsf{CL} \vdash M : A$ to a deduction of $\Gamma' \vdash M^* : A'$, where Γ' and A' are obtained from Γ and A by replacing some small simple types B by $\neg\neg B$ and changing some of the terms. Note that all the terms so changed have occurrences of cl in them; it follows from the subject-construction property[4] that if a term without an occurrence of cl occurs in a type in Γ or in A, then that term is unchanged, and so is any type to which it is proved to belong in the deduction. These terms occurring in the types of Γ or A (whether changed or not) will be called *type arguments*.

Since it is trivial to prove in constructive logic that $\neg\neg A \vdash A$, we can put all this in the form of the following theorem:

Theorem 3 *If there is a deduction of* $\Gamma, \mathsf{cl} : \mathsf{CL} \vdash M : A$, *and if* Γ' *and* A' *are obtained from* Γ *and* A *by 1) replacing every simple small type* B *by* $\neg\neg B$ *provided that* B *occurs in* Γ *or* A *but does not occur inside an occurrence of the type* void *or in the type of a type argument in which* cl *does not occur, and 2) by changing type arguments in which* cl *does occur, then for some term* M^* *there is a deduction of* $\Gamma' \vdash M^* : A'$.

Now suppose that we have a deduction $\mathsf{cl} : \mathsf{CL} \vdash M : \mathsf{void}$ (where Γ is empty). Then by the theorem, there is a deduction D of $\vdash M : \mathsf{void}$. Since

[4]See [2, Notes 14.18 and 15.12 and Remark 16.37] and [5, p. 301]. It says that a deduction follows the construction of the term.

there is no such deduction (by the normalization theorem), this gives us a proof of Coquand's consistency result:

Corollary 3.1 *Classical logic is consistent in the calculus of constructions.*

5 Classical arithmetic

In [5, Chapter 5] it is shown that intuitionistic arithmetic can be obtained by adding the two assumptions

$$\mathsf{peano1} : (\forall n : \mathsf{N})(\neg \boldsymbol{\sigma} n =_{\mathsf{N}} \mathbf{0}),$$

$$\mathsf{peano2} : (\forall n : \mathsf{N})(\forall m : \mathsf{N})(\boldsymbol{\sigma} n =_{\mathsf{N}} \boldsymbol{\sigma} m \to n =_{\mathsf{N}} m),$$

where "$P =_A Q$" is an abbreviation for "$\mathsf{Q}APQ$". There is no need for induction because of the predicate $\mathcal{N}$, which is defined to be

$$\lambda n : \mathsf{N} \,.\, (\forall A : \mathsf{N} \to \mathsf{Prop})((\forall m : \mathsf{N})(Am \to A(\boldsymbol{\sigma} m)) \to A\mathbf{0} \to An).$$

Then the induction axiom follows for terms of which $\mathcal{N}$ is true; in fact, this definition is essentially the way Dedekind proved induction. This is why $\mathsf{peano1}$ and $\mathsf{peano2}$ are sufficient for arithmetic.

In fact, $\mathsf{peano1}$ is sufficient. We can derive a version of $\mathsf{peano2}$ relative to $\mathcal{N}$ using the predecessor defined in [5, Definition 5.7] as follows:

Lemma 2 *For some term M,*

$$\vdash\ M : (\forall n : \mathsf{N})(\mathcal{N} n \to \boldsymbol{\pi}(\boldsymbol{\sigma} n) =_{\mathsf{N}} n).$$

Proof A direct calculation gives that $\boldsymbol{\pi}(\boldsymbol{\sigma}(\boldsymbol{\sigma} n)) =_* \boldsymbol{\sigma}(\boldsymbol{\pi}(\boldsymbol{\sigma} n))$. Hence, there is a term M_1 such that

$$n : \mathsf{N}, x : \boldsymbol{\pi}(\boldsymbol{\sigma} n) =_{\mathsf{N}} n \ \vdash\ M_1 : \boldsymbol{\pi}(\boldsymbol{\sigma}(\boldsymbol{\sigma} n)) =_{\mathsf{N}} \boldsymbol{\sigma} n.$$

Hence, by ($\forall\mathsf{Pi}$), there is a term M_2 such that

$$\vdash\ M_2 : (\forall n : \mathsf{N})(\boldsymbol{\pi}(\boldsymbol{\sigma} n) =_{\mathsf{N}} n \to \boldsymbol{\pi}(\boldsymbol{\sigma}(\boldsymbol{\sigma} n)) =_{\mathsf{N}} \boldsymbol{\sigma} n).$$

This is the induction step. The basis is easy, since $\boldsymbol{\pi}\mathbf{0} =_* \mathbf{0}$. Then induction (which follows from the definition of $\mathcal{N}$) gives us the lemma. ■

Lemma 3 *For some term M,*

$$\vdash\ M : (\forall n : \mathsf{N})(\forall n : \mathsf{N})(\mathcal{N} n \to \mathcal{N} m \to \boldsymbol{\sigma} n =_{\mathsf{N}} \boldsymbol{\sigma} m \to n =_{\mathsf{N}} m).$$

Proof We can easily formalize in this logic the following argument, where $n = m$ represents $n =_{\mathsf{N}} m$: $\boldsymbol{\sigma} n = \boldsymbol{\sigma} m$, therefore $\boldsymbol{\pi}(\boldsymbol{\sigma} n) = \boldsymbol{\pi}(\boldsymbol{\sigma} m)$, and so $n = m$. ■

Hence, to prove classical arithmetic consistent, it is enough to prove that $\mathsf{cl} : \mathsf{CL}, \mathsf{peano1} : (\forall n : \mathsf{N})(\neg \boldsymbol{\sigma} n =_{\mathsf{N}} \mathbf{0})$ is consistent.

Theorem 4 *Let Γ be a strongly consistent environment in which all simple types which have universal prefixes are large. Then*

$$\Gamma, \mathsf{cl} : \mathsf{CL}, \mathsf{peano1} : (\forall n : \mathsf{N})(\neg \boldsymbol{\sigma} n =_{\mathsf{N}} \mathbf{0})$$

is consistent.

Proof Suppose there is a term M such that

$$\Gamma, \mathsf{cl} : \mathsf{CL}, \mathsf{peano1} : (\forall n : \mathsf{N})(\neg \boldsymbol{\sigma} n =_{\mathsf{N}} \mathbf{0}) \ \vdash \ M : \mathsf{void}.$$

Then by Theorem 3 there is a term M' such that

$$\Gamma', \mathsf{peano1} : (\forall n : \mathsf{N})\neg(\forall z : \mathsf{N} \to \mathsf{Prop})(\neg\neg z(\boldsymbol{\sigma} n) \to \neg\neg z \mathbf{0}) \ \vdash \ M' : \mathsf{void}.$$

(Recall that $P =_A Q$ converts to $(\forall z : A \to \mathsf{Prop})(zP \to zQ)$.) It is not hard to see that Γ' is strongly negation consistent. Now in a normal deduction of

$$\Gamma', \mathsf{peano1} : (\forall n : \mathsf{N})\neg(\forall z : \mathsf{N} \to \mathsf{Prop})(\neg\neg z(\boldsymbol{\sigma} n) \to \neg\neg z \mathbf{0}) \ \vdash \ M' : \mathsf{void},$$

the top of the left branch must be

$$\mathsf{peano1} : (\forall n : \mathsf{N})\neg(\forall z : \mathsf{N} \to \mathsf{Prop})(\neg\neg z(\boldsymbol{\sigma} n) \to \neg\neg z \mathbf{0}),$$

and the minor premise for an inference by ($\forall$e) is

$$Q : (\forall z : \mathsf{N} \to \mathsf{Prop})(\neg\neg z(\boldsymbol{\sigma} U) \to \neg\neg z \mathbf{0}),$$

where there is an assumption $U : \mathsf{N}$. This is proved impossible by the following lemma. ■

Lemma 6 *If Γ is strongly negation consistent, and if for some term M*

$$\Gamma, \mathsf{peano1} : (\forall n : \mathsf{N})\neg(\forall z : \mathsf{N} \to \mathsf{Prop})(\neg\neg z(\boldsymbol{\sigma} n) \to \neg\neg z \mathbf{0}) \\ \vdash \ M : (\forall z : A \to \mathsf{Prop})(\neg\neg z R \to \neg\neg z S),$$

then $R =_* S$.

Proof Assume that D is a normalized deduction as in the lemma and that there is no proper subdeduction of something of this form whose undischarged assumptions are in some strongly negation consistent environment or are of the given form for $\mathsf{peano1}$. If the last inference in D[5] is ($\forall$e), then the top of the left branch must be

$$\mathsf{peano1} : (\forall n : \mathsf{N})\neg(\forall z : \mathsf{N} \rightarrow \mathsf{Prop})(\neg\neg z(\boldsymbol{\sigma} n) \rightarrow \neg\neg z\mathbf{0}),$$

and then the deduction ending in the minor premise violates the assumptions about D. In fact, this shows that

$$\mathsf{peano1} : (\forall n : \mathsf{N})\neg(\forall z : \mathsf{N} \rightarrow \mathsf{Prop})(\neg\neg z(\boldsymbol{\sigma} n) \rightarrow \neg\neg z\mathbf{0})$$

cannot occur anywhere in D at the top of the left branch, and, since D is normalized, this implies that it is not used in the deduction. It follows (by repeating this argument about subdeductions ending in ($\forall$e)) that we can decompose D until we have a deduction of

$$\Gamma, z : A \rightarrow \mathsf{Prop}, u : \neg\neg zR, v : \neg zS \ \vdash \ Q : \mathsf{void}.$$

By Theorem 2, this is only possible if $R =_* S$. ■

References

[1] Thierry Coquand. Metamathematical investigations of a calculus of constructions. In P. Odifreddi, editor, *Logic and Computer Science*, pages 91–122. Academic Press, 1990.

[2] J. Roger Hindley and Jonathan P. Seldin. *Introduction to Combinators and λ-calculus*. Cambridge University Press, 1986.

[3] Garrel Pottinger. Definite descriptions and excluded middle in the theory of constructions. Circulated electronically to TYPES mailing list, November 1989.

[4] Dag Prawitz. *Natural Deduction*. Almqvist & Wiksell, Stockholm, Göteborg, and Uppsala, 1965.

[5] Except for (Eq$''$).

[5] Jonathan P. Seldin. MATHESIS: The mathematical foundation for ULYSSES. Interim Report RADC-TR-87-223, Rome Air Development Center, Air force Systems Command, Griffiss Air Force Base, New York, November 1987. Actually published in 1988.

[6] Jonathan P. Seldin. On the proof theory of Coquand's calculus of constructions. *Annals of Pure and Applied Logic*, 83:23–101, 1997.