Interpreting HOL in the Calculus of Constructions ⋆

Jonathan P. Seldin

Department of Mathematics and Computer Science, University of Lethbridge, 4401 University Drive, Lethbridge, Alberta, Canada, T1K 3M4

Abstract

The purpose of this paper is to consider a representation of the HOL theorem-prover in the calculus of constructions with the property that consistency results from the calculus of constructions imply such results in HOL. This kind of representation is impossible using the propositions-as-types representation of logic and equality, but it is possible if a different representation is used.

Key words: Theorem-proving, Calculus of Constructions, HOL, Extensionality
1991 MSC: 03B15, 03B35, 03B40, 03B70

Since the basic formalism of higher-order logic (HOL) is weaker than that of the calculus of constructions, it seems natural to interpret the HOL theorem prover [1] in the latter. However, there are some problems in carrying out such an interpretation. The HOL theorem prover is based on classical logic, includes an ϵ-choice operator, and includes a principle of extensionality in the form

$$(\forall X : A)(\forall Y : A)((\forall x : A \to \mathsf{Prop})((Xx \leftrightarrow Yx) \supset X =_A Y).$$

The calculus of constructions is based on constructive logic, and although it can be consistently extended to classical logic by postulating excluded middle,

⋆ This paper represents a presentation entitled "Logic and Type Theory in Theorem Provers" which was presented to the conference "Thirty-Five Years of AUTOMATH" held at Heriot-Watt University, Edinburgh, 10–13 April 2002. This work was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada.

Email address: `jonathan.seldin@uleth.ca` (Jonathan P. Seldin).
URL: `http://www.cs.uleth.ca/~seldin` (Jonathan P. Seldin).

it has been known for some time that excluded middle together with an ϵ-choice operator leads to *proof irrelevance*, which says that all terms in a small type are Leibniz equal; see [2], which follows [3] and [4]. This would seem to make an interpretation of HOL in the calculus of constructions impossible.

However, these problems all involve representing logic in type theories using the *proposition-as-types* representation (also known as the *formulas-as-types* notion or the *Curry-Howard isomorphism*), in which the types are interpreted as logical formulas and the terms as proofs and, a type represents a *provable* formula if and only if it is inhabited. There is an alternative representation, which we might call the *Frege* representation, in which there is a type Bool in which there are distinct terms T and F, a term of type Bool represents a formula, and a formula in this sense A is provable if and only if $A =_{\mathsf{Bool}} \mathsf{T}$.

In this paper, I will use the Frege representation to interpret HOL in the calculus of constructions. To make the paper relatively self-contained, I will review the definition of the calculus of constructions in §1 and the propositions-as-types representation of logic with equality in §2. In §3, I will take up the Frege-style representation of logic with equality, and in §4 I will use this to interpret the HOL theorem-prover. In §5, the conclusion, I will discuss further work.

I would like to thank Martin Bunder, Roger Hindley, and the anonymous referees for their helpful comments and suggestions.

1 The Calculus of Constructions

Definition 1 *The* calculus of constructions *has syntax:*

$$M ::= x|c|\mathsf{Prop}|\mathsf{Type}|(MM)|(\lambda x : M \,.\, M)|(\forall x : M)M.$$

Here Prop *and* Type *are special constants called* sorts*; s, s', and s_1, etc., will be used for sorts.*[1] *Conversion will be β-conversion, generated by*

$$(\lambda x : A \,.\, M)N \rhd [N/x]M,$$

where $[N/x]M$ denotes the substitution of N for all free occurrences of x in M, with bound variables being changed to avoid conflicts. Judgements are of

[1] It is common to use * for Prop, $\Box$ for Type. I formerly referred to sorts as *kinds* [5–10].

the form $\Gamma \vdash M : A$, *where* Γ *is*

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

Here $dom(\Gamma) = \{x_1, x_2, \ldots, x_n\}$. *The system has one axiom, namely*

$$\vdash \mathsf{Prop} : \mathsf{Type}.$$

It has the following rules:

(start) If $x \notin \mathrm{dom}(\Gamma)$

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}$$

(weakening)

$$\frac{\Gamma \vdash A : B \quad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B}$$

(application)

$$\frac{\Gamma \vdash M : (\forall x : A)B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : [N/x]B}$$

(abstraction)

$$\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash (\forall x : A)B : s}{\Gamma \vdash \lambda x : A \,.\, M : (\forall x : A)B}$$

(conversion)

$$\frac{\Gamma \vdash A : B \quad \Gamma \vdash B' : s \quad B =_\beta B'}{\Gamma \vdash A : B'}$$

(product)

$$\frac{\Gamma \vdash A : s \quad \Gamma, x : A \vdash B : s'}{\Gamma \vdash (\forall x : A)B : s'},$$

A sequence of assumptions Γ is *legal* or *well-formed* if

$$\Gamma \vdash \mathsf{Prop} : \mathsf{Type},$$

A necessary and sufficient condition for he sequence

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

to be well-formed is that the variables $x_1, x_2, \ldots, x_n$ all be distinct and the sequence satisfy both of the following conditions:

(1) The variable x_i does not occur free in $A_1, A_2, \ldots, A_i$ (but it may occur free in $A_{i+1}, \ldots, A_n$), and
(2) $x_1 : A_1, x_2 : A_2, \ldots x_{i-1} : A_{i-1} \vdash A_i : s$ for some sort s.

From now on, all environments will be assumed to be well-formed.

2 Representing Logic with Equality, Propositions-as-Types

Let us review the representation of logic and equality in the calculus of constructions under the propositions-as-types representation. This material comes from [7, §6], where more details will be found.

If $x \notin \mathrm{FV}(B)$ in $(\forall x : A)B$, then we usually write this as $A \to B$ or, if $A, B : \mathsf{Prop}$, we write $A \supset B$.

It is easy to show special cases of (Application) and (Abstraction) are

$$\frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}$$

and

$$\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash A \to B : s}{\Gamma \vdash \lambda x : A \,.\, M : A \to B}$$

or

$$\frac{\Gamma \vdash M : A \supset B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}$$

and

$$\frac{\Gamma, x : A \ \vdash \ M : B \quad \Gamma \ \vdash \ A \supset B : s}{\Gamma \ \vdash \ \lambda x : A \,.\, M : A \supset B}$$

The *conjunction* of two propositions A and B is defined by

$$A \wedge B \equiv (\forall w : \mathsf{Prop})((A \to B \to w) \to w).$$

The terms of type $A \wedge B$ are ordered pairs whose first element is in A and whose second element is in B. The pairing operator and its projections give us the usual properties of conjunction, including

$$\begin{array}{l} A \to B \to A \wedge B, \\ A \wedge B \to A, \\ A \wedge B \to B. \end{array}$$

The *disjunction* of two propositions A and B is defined by

$$A \vee B \equiv (\forall w : \mathsf{Prop})((A \to w) \to ((B \to w) \to w)).$$

The terms of type $A \vee B$ are disjoint unions of A and B. These unions together with their injections and the case operator give us the usual properties of disjunction, including

$$\begin{array}{l} A \to A \vee B, \\ B \to A \vee B, \\ A \vee B \to (\forall w : \mathsf{Prop})((A \to w) \to ((B \to w) \to w)). \end{array}$$

The type void, which is intended to be empty, and which is written $\perp$ when it is desired to emphasize that it is a proposition, is defined by

$$\mathsf{void} \equiv (\forall x : \mathsf{Prop})x.$$

It follows from strong normalization that there is no closed term of type void. The *negation* of a proposition A is defined by

$$\neg A \equiv A \supset \perp.$$

The *existential quantifier* $(\forall x : A)B$ where $A, B : \mathsf{Prop}$ and $x \notin \mathrm{FV}(A)$ is defined by

$$(\exists x : A)B \equiv (\forall w : \mathsf{Prop})((\forall x : A)(B \supset w) \supset w).$$

The terms of type $(\exists x : A)B$ are pairs whose first element is in A and whose second element is in B. However, the pairs are defined differently from those of conjunction types, and the different typing means that although there is a first projection, there is no typable second projection; see [11]. This pairing operator and the partial projection function give us the usual properties of the existential quantifier, including

$$\begin{array}{ll} [M/x]B \supset (\exists x : A)B & \text{(for } M : A\text{)}, \\ (\exists x : A)B \supset (\forall w : \mathsf{Prop})((\forall x : A)(B \supset w) \supset w). & \end{array}$$

We can also define *Leibniz equality* over any type: if $M, N : A$, then

$$M =_A N \equiv (\forall z : A \to \mathsf{Prop})(zM \supset zN).$$

Then it is easy to prove the usual properties of equality, including the following (for $M, N : A$):

$$\begin{array}{l} M =_A M, \\ M =_A N \supset (\forall z : (A \to \mathsf{Prop}) \supset (zM \supset zN)). \end{array}$$

It is not hard to see from this that we have all the usual properties of constructive predicate logic with equality.

We can interpret classical logic by assigning to an atomic constant the type

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

We can also represent truth values:

$$\begin{array}{rl} \mathsf{Bool} \equiv & (\forall u : \mathsf{Prop})(u \to u \to u), \\ \mathsf{T} \equiv & \lambda u : \mathsf{Prop} \,.\, \lambda x : u \,.\, \lambda y : u \,.\, x, \\ \mathsf{F} \equiv & \lambda u : \mathsf{Prop} \,.\, \lambda x : u \,.\, \lambda y : u \,.\, y. \end{array}$$

It is easy to prove $\mathsf{Bool} : \mathsf{Prop}$, $\vdash \mathsf{T} : \mathsf{Bool}$, and $\vdash \mathsf{F} : \mathsf{Bool}$.

Berardi [12] assumes extensionality in the following form:

$$(\forall X : A)(\forall Y : A)((\forall x_1 : A_1)(\forall x_2 : A_2)\ldots(\forall x_n : A_n)$$
$$((Xx_1x_2\ldots x_n \leftrightarrow Yx_1x_2\ldots x_n) \supset X =_A Y),$$

where $A \equiv (\forall x_1 : A_1)(\forall x_2 : A_2)\ldots(\forall x_n : A_n)\mathsf{Prop}$. This implies (as Berardi shows) that all inhabited small types are models of untyped λ-calculus, and hence that the successor function on the natural numbers, $\boldsymbol{\sigma} : \mathsf{N} \to \mathsf{N}$, has a fixed-point! For this reason, we will not assume this form of extensionality in this paper.

For more on extensionality, see Appendix A.

3 Representing Logic with Equality, Frege Style

To carry out the Frege style representation consistently, we need to assume

$$\mathsf{boolcon} : \neg(\mathsf{T} =_{\mathsf{Bool}} \mathsf{F}).$$

Representing the logical connectives in the Frege style is easy; see [7, pp. 73–74]. We start with the familiar `if ...then ...else` operator:

$$\mathsf{Cond} \equiv \lambda u\,.\,\mathsf{Prop}\,.\,\lambda v : \mathsf{Bool}\,.\,\lambda x : u\,.\,\lambda y : u\,.\,vuxy.$$

Then if $A : \mathsf{Prop}, M : A, N : A$,

$$\mathsf{Cond}A\mathsf{T}MN =_\beta M$$

and

$$\mathsf{Cond}A\mathsf{F}MN =_\beta N.$$

The connectives can now be defined as follows:

$$\begin{aligned}
\neg_b &\equiv \lambda x : \mathsf{Bool}\,.\,\mathsf{Cond}\,\mathsf{Bool}\,x\,\mathsf{F}\,\mathsf{T},\\
\wedge_b &\equiv \lambda x : \mathsf{Bool}\,.\,\neg_b x\,\mathsf{Bool}\,\mathsf{F},\\
\vee_b &\equiv \lambda x : \mathsf{Bool}\,.\,x\,\mathsf{Bool}\,\mathsf{T},\\
\supset_b &\equiv \lambda x : \mathsf{Bool}\,.\,\lambda y : \mathsf{Bool}\,.\,\neg_b(\wedge_b\,x(\neg_b\,y)),\\
\leftrightarrow_b &\equiv \lambda x : \mathsf{Bool}\,.\,\lambda y : \mathsf{Bool}\,.\,(x \supset_b y) \wedge_b (y \supset_b x).
\end{aligned}$$

Then the usual truth table rules for these connectives hold as conversions.

The quantifiers, however, are another matter. We want

(1) $(\forall_b x : A)B =_{\mathsf{Bool}} \mathsf{T}$ iff $(\forall x : A)(B =_{\mathsf{Bool}} \mathsf{T})$, and
(2) $(\exists_b x : A)B =_{\mathsf{Bool}} \mathsf{T}$ iff $(\exists x : A)(B =_{\mathsf{Bool}} \mathsf{T})$.

(We may need to add arguments to $(\forall_b x : A)B$ and $(\exists_b x : A)B$.) There are two problems in achieving these properties. One is that the logic of the calculus of constructions is not classical. This problem can be easily fixed: postulate

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

This assumption and the assumption that $(\forall n : \mathsf{N})(\neg(\boldsymbol{\sigma} n =_{\mathsf{N}} \mathbf{0}))$ are proved consistent in [7, Theorem 23]. Essentially the same method can be used to prove that the above classical assumption and $\neg(\mathsf{T} =_{\mathsf{Bool}} \mathsf{F})$ are consistent. For the rest of this paper, we will assume both of these.

The other problem is more serious: even if the logic is classical, we cannot assume for $A : \mathsf{Prop}$ and $B : A \to \mathsf{Bool}$

$$(\forall x : A)(Bx =_{\mathsf{Bool}} \mathsf{T}) \vee (\exists x : A)(Bx =_{\mathsf{Bool}} \mathsf{F}).$$

Furthermore, within the logic, there is no way to prove that the only terms of type Bool are T and F.

Nevertheless, we can define $(\forall_b x : A)(Bx)$, where $A : \mathsf{Prop}$ and $B : A \to \mathsf{Bool}$. To do this, we first let

$$E_1 \equiv (\forall x : A)(Bx =_{\mathsf{Bool}} \mathsf{T}),$$
$$E_2 \equiv (\exists x : A)(Bx =_{\mathsf{Bool}} \mathsf{F}).$$

Then we can define $(\forall_b x : A)(Bx)$ to be $\Pi_b AB$, where

$$\begin{aligned}&\Pi_b \equiv \lambda A : \mathsf{Prop} \,.\, \lambda B : A \to \mathsf{Bool} \,.\, \lambda u : E_1 \vee E_2 \,.\\ &\lambda v : (\exists w : E_1)(u =_{E_1 \vee E_2} \mathsf{inl} E_1 E_2 w) \vee (\exists w : E_2)(u =_{E_1 \vee E_2} \mathsf{inr} E_1 E_2 w) \,.\\ &\mathsf{case} E_1 E_2 v \mathsf{Bool}(\lambda w : E_1 \,.\, \mathsf{T})(\lambda w : E_2 \,.\, \mathsf{F}),\end{aligned}$$

where case is defined along with the injections inl and inr for the disjoint sum (disjunction) in [7, Definition 21]. Note that it is shown in [7] that for $A, B, C : \mathsf{Prop}$, $M : A$, $N : B$, $F : A \to C$, and $G : B \to C$,

$$\mathsf{case} AB(\mathsf{inl} ABM)CFG =_\beta FM$$

and

$$\mathsf{case}AB(\mathsf{inr}ABN)CFG =_\beta GN.$$

Note that the type of Π_b is given by

$$\begin{aligned}&\Pi_b : (\forall A : \mathsf{Prop})(\forall B : A \to \mathsf{Bool})(\forall u : E_1 \vee E_2)\\ &(\forall v : (\exists w : E_1)(u =_{E_1 \vee E_2} \mathsf{inl}E_1E_2w) \vee\\ &(\exists w : E_2)(u =_{E_1 \vee E_2} \mathsf{inr}E_1E_2w))\mathsf{Bool}.\end{aligned}$$

It is then easy to show that if $M : E_1$, then $\mathsf{inl}E_1E_2M : E_1 \vee E_2$ and

$$\Pi_b AB(\mathsf{inl}E_1E_2M)V =_\beta \mathsf{T},$$

where V is the obvious proof that

$$\begin{aligned}&(\exists w : E_1)(\mathsf{inl}E_1E_2M =_{E_1 \vee E_2} \mathsf{inl}E_1E_2w)\\ &\qquad\qquad \vee(\exists w : E_2)(\mathsf{inl}E_1E_2M =_{E_1 \vee E_2} \mathsf{inr}E_1E_2w).\end{aligned}$$

On the other hand, if $M : E_2$, then $\mathsf{inr}E_1E_2M : E_1 \vee E_2$ and

$$\Pi_b AB(\mathsf{inr}E_1E_2M)V =_\beta \mathsf{F}.$$

where V is the obvious proof that

$$\begin{aligned}&(\exists w : E_1)(\mathsf{inr}E_1E_2M =_{E_1 \vee E_2} \mathsf{inl}E_1E_2w)\\ &\qquad\qquad \vee(\exists w : E_2)(\mathsf{inr}E_1E_2M =_{E_1 \vee E_2} \mathsf{inr}E_1E_2w).\end{aligned}$$

These are two important properties of Π_b in one direction. To get the other direction, assume that we have a proof U of

$$U : \Pi_b ABMV =_{\mathsf{Bool}} \mathsf{T}.$$

Since $\Pi_b ABMV : \mathsf{Bool}$, it follows that we must have $M : E_1 \vee E_2$ and

$$V : (\exists w : E_1)(M =_{E_1 \vee E_2} \mathsf{inl}E_1E_2w) \vee (\exists w : E_2)(M =_{E_1 \vee E_2} \mathsf{inr}E_1E_2w).$$

Now from the second disjunct, we get, as above,

$$\Pi_b ABMV =_\beta \mathsf{F},$$

which contradicts the conclusion of U. It therefore follows that the first disjunct holds, and so we have E_1, or $(\forall x : A)(Bx =_{\mathsf{Bool}} \mathsf{T})$. Similarly, given a proof U of

$$U : \Pi_b ABMV =_{\mathsf{Bool}} \mathsf{F},$$

we can conclude that $(\exists x : A)(Bx =_{\mathsf{Bool}} \mathsf{F})$. Thus, we have the desired properties of the universal quantifier in both directions.

If we define the predicate

$$\mathcal{BOOL} \equiv \lambda x : \mathsf{Bool} \,.\, x =_{\mathsf{Bool}} \mathsf{T} \vee x =_{\mathsf{Bool}} \mathsf{F},$$

then it follows that

$$\begin{array}{l} A : \mathsf{Prop}, B : A \to \mathsf{Bool}, w : (\forall x : A)(\mathcal{BOOL}(Bx)), u : E_1 \vee E_2, \\ v : (\exists w : E_1)(u =_{E_1} \mathsf{inl} E_1 E_2 w) \vee (\exists w : E_2)(u =_{E_2} \mathsf{inr} E_1 E_2 w) \\ \quad \vdash \ \mathcal{BOOL}(\Pi_b ABuv). \end{array}$$

For the existential quantifier, let

$$\begin{array}{l} E_3 \equiv (\exists x : A)(Bx =_{\mathsf{Bool}} \mathsf{T}), \\ E_4 \equiv (\forall x : A)(Bx =_{\mathsf{Bool}} \mathsf{F}). \end{array}$$

and define

$$\begin{array}{l} \Sigma_b : (\forall A : \mathsf{Prop})(\forall B : A \to \mathsf{Bool})(\forall u : E_3 \vee E_4) \\ (\forall v : (\exists w : E_3)(u =_{E_1 \vee E_2} \mathsf{inl} E_3 E_4 w) \vee \\ (\exists w : E_4)(u =_{E_3 \vee E_4} \mathsf{inr} E_3 E_4 w))\mathsf{Bool}. \end{array}$$

by

$$\begin{array}{l} \Sigma_b \equiv \lambda A : \mathsf{Prop} \,.\, \lambda B : A \to \mathsf{Bool} \,.\, \lambda u : E_3 \vee E_4 \,. \\ \lambda v : (\exists w : E_3)(u =_{E_3 \vee E_4} \mathsf{inl} E_3 E_4 w) \vee (\exists w : E_3)(u =_{E_3 \vee E_4} \mathsf{inr} E_3 E_4 w) \,. \\ \mathsf{case} E_3 E_4 v \mathsf{Bool}(\lambda w : E_3 \,.\, \mathsf{T})(\lambda w : E_4 \,.\, \mathsf{F}). \end{array}$$

The desired properties of the existential quantifier follow in much the same way as do those for the universal quantifier.

This is an example of a general method of defining operators in the Frege style. Suppose we want an operator $O(A_1, A_2, \ldots, A_n) : A$ with parameters $A_1, A_2, \ldots, A_n$ with the property that

$$O(A_1, A_2, \ldots, A_n) =_A \begin{cases} \mathsf{T} \text{ if and only if } C_T(A_1, A_2, \ldots, A_n) \\ \mathsf{F} \text{ if and only if } C_F(A_1, A_2, \ldots, A_n). \end{cases}$$

To do this we need the universal closure of

$$C_T(A_1, A_2, \ldots, A_n) \vee C_F(A_1, A_2, \ldots, A_n).$$

If we have that, we can define $O(A_1, A_2, \ldots, A_n)$, which we may as well write as $O(\vec{A})$, to be $OpA_1A_2 \ldots A_n$, where

$$\begin{aligned} &OpA_1A_2 \ldots A_n \equiv \lambda u : C_T(\vec{A}) \vee C_F(\vec{A}) \ . \\ &\lambda v : (\exists w : C_T(\vec{A})(u =_{C_T(\vec{A})} \mathsf{inl} C_T(\vec{A}) C_F(\vec{A}) w) \vee \\ &(\exists w : C_F(\vec{A}))(u =_{C_F(\vec{A})} \mathsf{inr} C_T(\vec{A}) C_F(\vec{A}) w) \ . \\ &\mathsf{case} C_T(\vec{A}) C_F(\vec{A}) u \mathsf{Bool}(\lambda w : C_T(\vec{A}) \ . \ \mathsf{T})(\lambda w : C_F(\vec{A}) \ . \ \mathsf{F}). \end{aligned}$$

As an example, let us define Boolean Leibniz equality $M =_{bA} N$ over a type $A : \mathsf{Prop}$, where $M, N : A$. Here,

$$\begin{aligned} C_T &\equiv (\forall z : A \to \mathsf{Bool})((\forall x : A)(\mathcal{BOOL}(zx)) \supset ((zM \supset_b zN) =_{\mathsf{Bool}} \mathsf{T})), \\ C_F &\equiv (\exists z : A \to \mathsf{Bool})((\forall x : A)(\mathcal{BOOL}(zx)) \wedge ((zM \supset_b zN) =_{\mathsf{Bool}} \mathsf{F})). \end{aligned}$$

The definition is then

$$\begin{aligned} &Eq_b \equiv \lambda A : \mathsf{Prop} \ . \ \lambda x : A \ . \ \lambda y : A \ . \ \lambda u : C_T \vee C_F \ . \\ &\lambda v \ . \ (\exists w : C_T)(u =_{C_T \vee C_F} \mathsf{inl} C_T C_F w) \vee (\exists w : C_F)(u =_{C_T \vee C_F} \mathsf{inr} C_T C_F w \ . \\ &\mathsf{case} C_T C_F u \mathsf{Bool}(\lambda w : C_T \ . \ \mathsf{T})(\lambda w : C_F \ . \ \mathsf{F}). \end{aligned}$$

We can then take $M =_{bA} N$ as an abbreviation for $Eq_b AMN$.

This gives us classical logic with equality. The special case of extensionality

$$(\forall_b x : \mathsf{Bool})(\forall_b y : \mathsf{Bool})((x \leftrightarrow_b y) \supset_b (x =_{b\mathsf{Bool}} y)) =_{\mathsf{Bool}} \mathsf{T}$$

now follows by definition.

4 Interpreting HOL

Since the HOL theorem prover has a mechanism to convert some[2] predicates into types, it seems natural to interpret HOL types as calculus of constructions predicates. This gives us a class of interpretations.

The types of HOL have the following syntax:

$$\sigma \longrightarrow \alpha | c | c(\sigma_1, \sigma_2, \ldots, \sigma_n)^{\,3} \, | \sigma_1 \to \sigma_2,$$

where α represents a type variable and c represents a type constant. These types can be interpreted in the calculus of constructions as follows:

(1) Interpret α as variable $\alpha^* : A \to \mathsf{Prop}$ for some type A.
(2) Interpret c with arity n as a constant

$$c^* : (A_1 \to \mathsf{Prop}) \to (A_2 \to \mathsf{Prop}) \to \ \ldots \ \to (A_n \to \mathsf{Prop}) \to (A \to \mathsf{Prop})$$

for some types $A_1, A_2, \ldots, A_n, A$ (n may be 0).
(3) Given HOL types σ_1 and σ_2 with calculus of constructions interpretations $\sigma_1^* : A_1 \to \mathsf{Prop}$ and $\sigma_2^* : A_2 \to \mathsf{Prop}$,

$$(\sigma_1 \to \sigma_2)^* \equiv \lambda u : A_1 \to A_2 \,.\, (\forall x : A_1)(\sigma_1^* x \supset \sigma_2^*(ux)).$$

HOL terms have the syntax:

$$t_\sigma \longrightarrow x_\sigma | c_\sigma | (t_{\sigma'} \to_\sigma t_{\sigma'})_\sigma | (\lambda x_{\sigma_1} \,.\, t_{\sigma_2})_{\sigma_1 \to \sigma_2}$$

The HOL terms are interpreted in the calculus of constructions as follows:

(1) $x_\sigma^* \equiv x : A$ where $\sigma^* : A \to \mathsf{Prop}$ and $\vdash \sigma^* x$ (i.e., $\sigma^* x$ is provable).
(2) $c_\sigma^* \equiv c : A$ where $\sigma^* : A \to \mathsf{Prop}$ and $\vdash \sigma^* c$.
(3) if $\sigma^* : A \to \mathsf{Prop}$ and $\sigma'^* : A' \to \mathsf{Prop}$, then

$$(t_{\sigma'} \to_\sigma t'_{\sigma'})^* \equiv t^*_{\sigma'} \to_\sigma (t'_{\sigma'})^*$$

From $\vdash (\sigma' \to \sigma)^* t^*$ and $\vdash \sigma'^* t'$, it follows that $\vdash \sigma^*(tt')^*$.
(4) if $\sigma_1 : A_1 \to \mathsf{Prop}$ and $\sigma_2 : A_2 \to \mathsf{Prop}$, then if $x^*_{\sigma_1} \equiv x : A_1$ and $t^*_{\sigma_2} \equiv t : A_2$, then

$$(\lambda x_{\sigma_1} \,.\, t_{\sigma_2})^* \equiv \lambda x : A_1 \,.\, t : A_1 \to A_2$$

and $\vdash (\sigma_1 \to \sigma_2)^*(\lambda x : A_1 \,.\, t)$.

[2] In particular, the predicates converted to types must be satisfied by some closed terms.

An HOL type structure is *standard* if it contains types bool of truth values and ind of individuals.

We can interpret a standard type structure if we have the corresponding predicates. We get them as follows:

(1) $(\mathsf{bool})^* \equiv \mathcal{BOOL} : \mathsf{Bool} \to \mathsf{Prop}$, and
(2) $(\mathsf{ind})^* \equiv \lambda x : \mathsf{Ind} \,.\, x =_{\mathsf{Bool}} x : \mathsf{Ind} \to \mathsf{Prop}$, where $\mathsf{Ind} : \mathsf{Prop}$ is a new type constant.

An HOL signature is *standard* if it includes

(1) $\Rightarrow_{\mathsf{bool} \to \mathsf{bool} \to \mathsf{bool}}$,
(2) $=_{\alpha \to \alpha \to \mathsf{bool}}$, and
(3) $\epsilon_{\alpha \to (\alpha \to \mathsf{bool}) \to \alpha}$.

We interpret these as follows:

(1) $(\Rightarrow_{\mathsf{bool} \to \mathsf{bool} \to \mathsf{bool}})^* \equiv \supset_b : \mathsf{Bool} \to \mathsf{Bool} \to \mathsf{Bool}$, where it is provable that

$$\vdash \ (\forall x : \mathsf{Bool})(\mathcal{BOOL}x \to (\forall y : \mathsf{Bool})(\mathcal{BOOL}y \to \mathcal{BOOL}(\supset_b xy))).$$

(2) $(=_{\alpha \to \alpha \to \mathsf{bool}})^* \equiv Eq_b A$, where $\alpha^* : A \to \mathsf{Prop}$.
(3) $(\epsilon_{(\alpha \to \mathsf{bool}) \to \alpha})^* \equiv \epsilon_b A$, where ϵ_b is defined by

$$\epsilon_b \equiv \lambda A : \mathsf{Prop} \,.\, \lambda x : A \,.\, \lambda f : A \to \mathsf{Bool} \,.\, \lambda u : (fx =_{\mathsf{Bool}} \mathsf{T}) \,.\, x.$$

It is easy to prove that

$$\epsilon_b : (\forall A : \mathsf{Prop})(\forall x : A)(\forall f : A \to \mathsf{Bool})(\forall u : (fx =_{\mathsf{Bool}} \mathsf{T}))A.$$

Remark 2 *This definition of ϵ_b does not fully capture all the semantics of a choice operator, since it specifies a choice based on one of the additional arguments, in particular the term in A for which there is a proof that $fx =_{\mathsf{Bool}} \mathsf{T}$. This means that it would be possible to make HOL assumptions that would be incompatible with this interpretation. Nevertheless, this definition does satisfy the postulates for the choice operator in HOL.*

(In some ways, ϵ_b is more like the independent choice operator δ of [13], which is a choice operator for which the choices are not fixed, but different choices on the same set may be different. However, it differs from δ in that the value of a choice is determined by all the arguments of ϵ_b.)

If we wanted to have a definition of ϵ_b fully compatible with the semantics for a general choice operator, we would need to postulate a new constant

$$\epsilon_b : (\forall A : \mathsf{Prop})(A \to (A \to \mathsf{Bool}) \to A)$$

and then postulate for it

$$(\forall A : \mathsf{Prop})(\forall P : A \to \mathsf{Prop})((\forall x : A)(\mathcal{BOOL}(Px)) \supset$$
$$((\exists_b x : A)(Px) =_{\mathsf{Bool}} \mathsf{T}) \supset (\forall a : A)(P(\epsilon_b AaP) =_{\mathsf{Bool}} \mathsf{T})) \wedge$$
$$((\forall_b x : A)(Px) =_{\mathsf{Bool}} \mathsf{F}) \supset (\forall a : A)(P(\epsilon_b AaP) =_{\mathsf{Bool}} \mathsf{F})).$$

We would need the extra argument of type A in this case to take care of the case, which cannot arise in HOL but can in the calculus of constructions, in which the type A is not inhabited. In this case, we expect the value of $\epsilon_b AaF$ to be a. The problem with this approach is that it requires a new unproved assumption, and this would either require a consistency proof or else undermine our consistency claims.

Remark 3 *Note that if we were dealing with the definition description operator, we could use this technique and define*

$$\iota_b \equiv \lambda A : \mathsf{Prop} \,.\, \lambda a : A \,.\, \lambda f : A \to \mathsf{Bool} \,.\, \lambda x : A \,.$$
$$\lambda u : (fx =_{\mathsf{Bool}} \mathsf{T}) \wedge (\forall y : A)(fy =_{\mathsf{Bool}} \mathsf{T} \supset y =_A x)) \,.\, x.$$

This makes the value of $\iota_b Aaf$ the *object which can be proved to satisfy the Boolean predicate function f. For the description operator, the problem of choice does not arise.*

Deductive systems in HOL are defined in terms of sequents. An HOL sequent is a pair (Γ, t), where Γ is a finite set of terms of type bool (the HOL type) and t is a term of that type. The interpretation t^* of an HOL term t of type bool has type Bool. If $\Gamma \equiv t_1, t_2, \ldots, t_n$, then the interpretation of Γ is $\Gamma^* \equiv t_1^*, t_2^*, \ldots, t_n^*$. In HOL, $\Gamma \vdash t$ means that the sequent (Γ, t) follows from the inference rules. Here, $(\Gamma \vdash t)^*$ will be

$$\Gamma^* =_{b\mathsf{Bool}} \mathsf{T} \supset t^* =_{b\mathsf{Bool}} \mathsf{T},$$

where if $\Gamma \equiv t_1, t_2, \ldots t_n$, then

$$(\Gamma^* =_{b\mathsf{Bool}} \mathsf{T}) \text{ means } t_1^* =_{b\mathsf{Bool}} \mathsf{T} \wedge t_2^* =_{b\mathsf{Bool}} \mathsf{T} \wedge \ldots \wedge t_n^* =_{b\mathsf{Bool}} \mathsf{T}.$$

The deductive system of HOL is defined by eight *rules of inference*, each of which has the form

$$\frac{\Gamma_1 \vdash t_1 \quad \Gamma_2 \vdash t_2 \quad \ldots \quad \Gamma_n \vdash t_n}{\Gamma \vdash t}$$

For each of these eight rules of inference, we would like to prove in the calculus

of constructions

$$(\Gamma_1 \ \vdash \ t_1)^* \supset (\Gamma_2 \ \vdash \ t_2)^* \supset \ldots \supset (\Gamma_n \ \vdash \ t_n)^* \supset (\Gamma \ \vdash \ t)^*.$$

This turns out to be easy for the following seven rules:

(1) **Assumption introduction**

$$\frac{}{t \ \vdash \ t}.$$

(2) **Reflexivity**

$$\frac{}{\ \vdash \ t = t}.$$

(3) **Beta-conversion**

$$\frac{}{\ \vdash \ (\lambda x \,.\, t_1)t_2 = [t_2/x]t_1}.$$

where $[t_2/x]t_1$ is substitution with bound variables changed automatically to avoid clashes.

(4) **Substitution**

$$\frac{\Gamma_1 \ \vdash \ t_1 = t_1' \quad \Gamma_2 \ \vdash \ t_2 = t_2' \quad \ldots \quad \Gamma_n \ \vdash \ t_n = t_n' \quad \Gamma \ \vdash \ t[t_1, t_2, \ldots, t_n]}{\Gamma_1 \cup \Gamma_2 \cup \ldots \cup \Gamma_n \cup \Gamma \ \vdash \ t[t_1', t_2', \ldots, t_n']},$$

where $t[t_1, t_2, \ldots, t_n]$ denotes a term t with some free occurrences of subterms $t_1, t_2, \ldots, t_n$ singled out and $t[t_1', t_2', \ldots, t_n']$ denotes the result of replacing each selected occurrence of t_i by t_i' for $1 \leq i \leq n$, with suitable renaming of bound variables to prevent any free variables from becoming bound in the replacement.

(5) **Type instantiation**

$$\frac{\Gamma \ \vdash \ t}{\Gamma \ \vdash \ [\sigma_1/\alpha_1, \sigma_2/\alpha_2, \ldots, \sigma_n/\alpha_n]t},$$

where none of the type variables $\alpha_1, \alpha_2, \ldots, \alpha_n$ occur free in Γ and no distinct variables in t become identified after the substitution of the σ_i for the α_i.

(6) **Discharging an assumption**

$$\frac{\Gamma \cup \{t_1\} \ \vdash \ t_2}{\Gamma \ \vdash \ t_1 \Rightarrow t_2}.$$

(7) **Modus Ponens**

$$\frac{\Gamma_1 \ \vdash \ t_1 \Rightarrow t_2 \quad \Gamma_2 \ \vdash \ t_1}{\Gamma_1 \cup \Gamma_2 \ \vdash \ t_2}.$$

However, there is a problem with the eighth rule:

Abstraction

$$\frac{\Gamma \ \vdash \ t_1 = t_2}{\Gamma \ \vdash \ (\lambda x \,.\, t_1) = (\lambda x \,.\, t_2),}$$

where x does not occur free in Γ.

The problem is that it is not possible to prove in HOL that if $\Gamma \ \vdash \ t_1 = t_2$ then t_1 and t_2 are convertible. Proving the interpretation of this rule would require postulating the following version of *Boolean extensionality*:

$$\begin{array}{l}(\forall A : \mathsf{Prop})(\forall B : \mathsf{Prop})(\forall x : A)(\forall y : A) \\ \ ([(\forall_b u : B \to A)(xu =_{bA} yu) \supset_b (x =_{b(B \to A)} y)] =_{\mathsf{Bool}} \mathsf{T}),\end{array}$$

which does not seem to be provable.

5 Conclusion

This interpretation depends on three unproved assumptions:

(1) $\mathsf{boolcon} : \neg(\mathsf{T} =_{\mathsf{Bool}} \mathsf{F})$,
(2) $\mathsf{cl} : (\forall u : \mathsf{Prop})(\neg\neg u \supset u)$,
(3) $ext : (\forall A : \mathsf{Prop})(\forall B : \mathsf{Prop})(\forall x : A)(\forall y : A)$
$([(\forall_b u : B \to A)(xu =_{bA} yu) \supset_b (x =_{b(B \to A)} y)] =_{\mathsf{Bool}} \mathsf{T})$.

The first two of these are consistent, by an easy modification of the proof of [7, Theorem 23], as noted above. The third seems reasonable, since it is known that η-reduction satisfies the Church-Rosser property. I conjecture that the consistency of all three of these umproved assumptions can be proved consistent by using a term model and so modelling HOL that two HOL terms are Boolean equal if and only if they are convertible, but I have not checked the details.

It is natural to ask whether this interpretation of the HOL theorem prover in the calculus of constructions would be consistent if the choice function were defined as in Remark 2 to satisfy the general semantics of a choice operator, with its extra unproved assumption. There might be cause for nervousness about this unproved assumption because we would have both extensionality and the choice operator. However, the argument of [2] fails in the Frege-style

representation of logic being used here. Barbanera and Berardi's argument depends on defining a type

$$U \equiv \forall X : \mathsf{Prop} \,.\, X \to \mathsf{Bool}$$

with $U : \mathsf{Prop}$. For this type U, $U \to \mathsf{Prop}$ is a retract of it. In our setting, the corresponding type

$$V \equiv \forall X : \mathsf{Bool} \,.\, X \to \mathsf{Bool}$$

has type Prop but *not* type Bool, so the argument is blocked. Furthermore, the interpretation actually used in the paper does not require this extra unproved assumption.

Appendix

A More on Extensionality

One might think that the fact that the form of extensionality assumed by Berardi [12] implies that the successor function on the natural numbers has a fixed point is enough reason not to assume this form of extensionality. But it might seem that there is another reason: it raises possible problems with the axiom of choice (AC).

Barbanera and Berardi [2], following Coquand [3] and Pottinger [4], prove that adding both AC and excluded middle (EM) implies the principle of *proof irrelevance* (PI), which is that all terms of any type $A : \mathsf{Prop}$ are Leibniz equal. PI therefore implies

$$\mathsf{T} =_{\mathsf{Bool}} \mathsf{F},$$

which is clearly undesirable in any useful proof assistant. Now, by a theorem of Diaconescu, which was called to my attention by John Bell (see his [14]), and which states that AC implies EM in a topos, it would seem that extensionality and AC would imply PI.

However, it turns out that the form of AC needed for the theorem of Diaconescu does not follow from the forms usually assumed for type theories.

First, let us see the Theorem of Diaconescu and its proof.

Theorem 4 *If there is a choice function*

$$f : (\mathsf{N} \to \mathsf{Prop}) \to \mathsf{N}$$

which satisfies

$$(\forall u : \mathsf{N} \to \mathsf{Prop})((\exists x : \mathsf{N})(ux) \to u(fu)),$$

and if extensionality holds, then both EM and PI hold.

PROOF. Since AC and EM imply PI, it is sufficient to prove that the existence of f implies EM. Let $\alpha : \mathsf{Prop}$, and define $U, V : \mathsf{N} \to \mathsf{Prop}$ by

$$U \equiv \lambda x : \mathsf{N} \,.\, x =_{\mathsf{N}} \mathbf{0} \vee \alpha,$$
$$V \equiv \lambda x : \mathsf{N} \,.\, x =_{\mathsf{N}} \mathbf{1} \vee \alpha.$$

Then $fU, fV : \mathsf{N}$, $\vdash U\mathbf{0}$, and $\vdash V\mathbf{1}$. Let $a \equiv fU$ and $b \equiv fV$. Then $a, b : \mathsf{N}$ and, by the hypothesis on the properties of f, we have

$$\vdash Ua, \qquad\qquad \vdash Vb.$$

Thus,

$$\vdash (a =_{\mathsf{N}} \mathbf{0} \vee \alpha) \wedge (b =_{\mathsf{N}} \mathbf{1} \vee \alpha).$$

By a constructively valid part of distribution,

$$\vdash (a =_{\mathsf{N}} \mathbf{0} \wedge b =_{\mathsf{N}} \mathbf{1}) \vee \alpha. \tag{A.1}$$

Now

$$y : \alpha \vdash (\forall x : \mathsf{N})(Ux \leftrightarrow Vx).$$

From this and extensionality, we get

$$y : \alpha \vdash U =_{\mathsf{N} \to \mathsf{Prop}} V, \tag{A.2}$$

Hence,

$$y : \alpha \vdash fU =_{\mathsf{N}} fV,$$

or

$$y : \alpha \ \vdash \ a =_{\mathsf{N}} b. \tag{A.3}$$

From this follows

$$y : \alpha \ \vdash \ \neg(a =_{\mathsf{N}} \mathbf{0} \wedge b =_{\mathsf{N}} \mathbf{1}),$$

from which we can get

$$z : (a =_{\mathsf{N}} \mathbf{0} \wedge b =_{\mathsf{N}} \mathbf{1}) \ \vdash \ \neg\alpha.$$

Hence, by (A.1),

$$\vdash \ \neg\alpha \vee \alpha. \qquad \square$$

To apply this theorem to type theories, we would have to have the kind of choice function given in the theorem. Now, the usual form of the axiom of choice in type theory is

$$\begin{aligned}&(\forall A : \mathsf{Prop})(\forall B : \mathsf{Prop})(\forall P : A \to B \to \mathsf{Prop})\\ &((\forall x : A)(\exists y : B)Pxy \supset (\exists f : A \to B)(\forall x : A)Px(fx)).\end{aligned}$$

In order for the function f required for Theorem 4 to be definable from this assumption, we would need $A \equiv \mathsf{N} \to \mathsf{Prop}$, $B \equiv \mathsf{N}$, and

$$P \equiv \lambda u : \mathsf{N} \to \mathsf{Prop} \,.\, \lambda x : \mathsf{N} \,.\, ux.$$

But then we do not have $A : \mathsf{Prop}$, as required by the axiom. And even if we allow $A : \mathsf{Type}$ in AC, which would work for $A \equiv \mathsf{N} \to \mathsf{Prop}$, we still do not have the right properties for f: the hypothesis of this form of the axiom of choice would require a proof that $(\forall u : \mathsf{N} \to \mathsf{Prop})(\exists x : \mathsf{N})ux$, which is not the hypothesis of Theorem 4 and is false: take $U : \mathsf{N} \to \mathsf{Prop}$ to be $U \equiv \lambda x : \mathsf{N} \,.\, \neg(x =_{\mathsf{N}} x)$.

But this is not the form of the axiom of choice used to prove that the AC + EM implies PI. That form is the postulation of two constants:

$$\begin{aligned}\epsilon &: (\forall A : \mathsf{Prop})(\forall P : A \to \mathsf{Prop})((\exists x : A)(Px) \to A),\\ \epsilon_{\mathrm{in}} &: (\forall A : \mathsf{Prop})(\forall P : A \to \mathsf{Prop})(\forall h : (\exists x : A)(Px))(P(\epsilon APh)).\end{aligned}$$

In [2], these constants are called $\mathcal{AC}_{\mathcal{F}}$ and $\mathcal{AC}_{\mathcal{A}}$ respectively, and the authors say that assuming these two constants is stronger than the set-theoretic form of the axiom of choice.

To use these postulates in the proof of the Theorem of Diaconescu, we need to use

$$\lambda u : \mathsf{N} \to \mathsf{Prop} \,.\, \lambda h : (\exists x : \mathsf{N})(ux) \,.\, \epsilon \mathsf{N} u h$$

for f. But this function has an extra argument, and so a and b now have to be defined by

$$a \equiv \epsilon \mathsf{N} U h_U \qquad\qquad b \equiv \epsilon \mathsf{N} V h_V,$$

where h_U and h_V are the obvious proofs of $(\exists x : \mathsf{N})Ux$ and $(\exists x : \mathsf{N})Vx$ respectively. This means that from (A.2), we do not get (A.3), but only

$$y : \alpha \;\vdash\; \epsilon \mathsf{N} U h_U =_{\mathsf{N}} \epsilon \mathsf{N} V h_U.$$

We could try defining b as $\epsilon \mathsf{N} V h_U$, but then instead of $\vdash Vb$, we would need to use the equality of U and V, and since this requires the equivalence of U and V, instead of $\vdash Vb$, we would get

$$y : \alpha \;\vdash\; Vb.$$

Instead of (A.1), we would have

$$y : \alpha \;\vdash\; (a =_{\mathsf{N}} 0 \wedge b =_{\mathsf{N}} 1) \vee \alpha,$$

and so the conclusion of the proof would be

$$y : \alpha \;\vdash\; \neg\alpha \vee \alpha,$$

a triviality.

It would be possible to get around this by assuming an axiom of the form

$$(\forall A : \mathsf{Prop})(\forall P_1 : A \to \mathsf{Prop})(\forall P_2 : A \to \mathsf{Prop})(\forall h_1 : (\exists x : A)P_1 x)$$
$$(\forall h_2 : (\exists x : A)P_2 x)(\forall x : A)(P_1 x \leftrightarrow P_2 x)(\epsilon A P_1 h_1 =_A \epsilon A P_2 h_2).$$

This axiom is similar to an axiom assumed by Maehara for the ϵ-symbol in [15]. However, with this axiom, extensionality would no longer be needed for the

proof of the Theorem of Diaconescu, since the only inference in its proof that used extensionality would be justified by the above axiom. So this axiom plus AC implies PI. This makes assuming this axiom dubious.

It appears that the claim in [2] that the postulation of the two constants ϵ and ϵ_{in} is stronger than the set-theoretic form of AC is not justified.

References

[1] M. J. C. Gordon, T. F. Melham, Introduction to HOL: A Theorem Proving Environment for Higher Order Logic, Cambridge University Press, 1993.

[2] F. Barbanera, S. Berardi, Proof-irrelevance out of excluded-middle and choice in the calculus of constructions, Journal of Functional Programming 6 (3) (1996) 519–525.

[3] T. Coquand, Metamathematical investigations of a claculus of constructions, in: P. Odifreddi (Ed.), Logic and Computer Science, Academic Press, 1990, pp. 91–122.

[4] G. Pottinger, Definite descriptions and excluded middle in the theory of constructions, circulated electronically to TYPES mailing list (November 1989).

[5] J. 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, actually published in 1988. (November 1987).

[6] J. P. Seldin, Coquand's calculus of constructions: a mathematical foundation for a proof development system, Formal Aspects of Computing 4 (1992) 425–441.

[7] J. P. Seldin, On the proof theory of Coquand's calculus of constructions, Annals of Pure and Applied Logic 83 (1997) 23–101.

[8] J. P. Seldin, A Gentzen-style sequent calculus of constructions with expansion rules, Theoretical Computer Science 243 (2000) 199–215.

[9] J. P. Seldin, On lists and other abstract data types in the calculus of constructions, Mathematical Structures in Computer Science 10 (2000) 261–276, special issue in honor of J. Lambek.

[10] J. P. Seldin, Extensional set equality in the calculus of constructions, Journal of Logic and Computation 11 (3) (2001) 483–493, presented at Festival Workshop in Foundations and Computations held at Heriot-Watt University, Edinburgh, 16-18 July, 2000.

[11] L. Cardelli, A polymorphic λ-calculus with Type : Type, Tech. Rep. 10, Digital Equipment Corporation Systems Research Center, Palo Alto, California, available from author's home page, `http://www.luca.demon.co.uk/`, under Publications. (May 1986).

[12] S. Berardi, Towards a mathematical analysis of the Coquand-Huet calculus of constructions and the other systems of the Barendregt cube, Tech. rep., Department of Computer Science, Carnegie-Mellon University and Dipartimento di Matematica, Universita di Torino (1988).

[13] A. Blass, Y. Gurevich, The logic of choice, The Journal of Symbolic Logic 65 (3) (2000) 1264–1310.

[14] J. Bell, Toposes and Local Set Theories, Oxford University Press, 1988.

[15] S. Maehara, Equality axiom on Hilbert's ϵ-symbol, Journal of the Faculty of Science, University of Tokyo, Section I 7 (1957) 419–435.