Research Article

A “Propositions as Types” Interpretation of Classical Logic

Andrew Powell[1]

1. Imperial College London, United Kingdom

This paper presents a theorem–proof style account of a “propositions as types” interpretation of classical logic. Inhabited and uninhabited types are represented by $\top$ and $\bot$, and a small family of type-level reductions is used to evaluate logical types at the level of truth values. When a type reduces to the canonical inhabited form $\bot \to \bot$, its canonical reduced witness is the identity inhabitant $i_{\bot\to\bot} : \bot \to \bot$. A witness of the original type is then treated as an induced witness, read from this reduced identity form and interpreted schematically, and in general contextually, at the original type. The resulting framework is applied to first-order classical propositional logic and to first-, second-, and higher-order classical predicate logic. The emphasis is on a uniform formal presentation while retaining the motivating idea that classical reasoning may be represented without introducing explicit control operators into the term language.

Corresponding author: Andrew Powell, andrew.powell@imperial.co.uk

1. Introduction

The Curry–Howard correspondence identifies propositions with types and proofs with programs. In its most familiar form, a proposition is true exactly when the corresponding type is inhabited, and a proof of the proposition is represented by a term inhabiting that type. This viewpoint is especially natural in typed systems of the λ-calculus, where terms already function as programs and types constrain how those programs may be formed; see for example [1][2][3][4][5][6][7].

Throughout, “true” means “evaluates to $\top$” in the Boolean inhabitation semantics fixed below, rather than “derivable as a closed judgment in a particular deductive system”.

 doi.org/10.32388/IY3BKH.4

The "propositions as types" view of logic is due initially to [8][9][10]; for historical discussion see also [11]. A systematic modern treatment may be found in [12].

For intuitionistic logic this correspondence is standard. In particular, existential statements require explicit witnesses, and principles such as double negation elimination do not in general hold. This viewpoint is closely associated with intuitionistic type theory in the sense of Martin-Löf [13][14][15], building on the broader intuitionistic tradition discussed, for example, in [16].

Classical logic can also be related to typed computation, but the usual accounts often proceed by extending the term language with control operators or continuation-based mechanisms. The classic starting point is Griffin's continuation-based interpretation of classical logic [17], followed by a substantial literature including [18][19][20][21]. In these systems, classical proofs are interpreted as programs with first-class control operators, whose primary computational content lies in manipulating continuations (implementing, for instance, backtracking, exceptions, and non-local exits) rather than delivering canonical data witnesses for existential statements in the simple Curry–Howard sense.

The aim of this paper is different. We remain within an ordinary typed λ-calculus and instead separate two levels of reasoning. First, we evaluate types at the level of truth values by using a small collection of reduction rules. Second, when a type is thereby shown to be inhabited, we assign to it an induced witness obtained from the canonical reduced witness of the identity form. The point of this approach is not to deny that continuation-based interpretations exist, but to isolate a simpler mechanism within an ordinary typed setting. The resulting account is especially suited to logically valid principles, where the central task is not to compute a specific mathematical object but to show that a type corresponding to a logical truth is inhabited.

1.1. Strategy of the paper

The strategy is as follows.

- We represent inhabited and uninhabited types by $\top$ and $\bot$.
- We identify the truth values of the four basic function types

  $\bot \to \bot, \bot \to \top, \top \to \top, \top \to \bot$.
- We adopt a small family of type-level reduction principles, such as

  $red(A \to \bot) = \bot$,

  when A is inhabited, where red is a truth valuation function for particular formulas.

- When a type reduces to $\bot \to \bot$, we take $i_{\bot\to\bot}$ as the canonical reduced witness of that reduced form.
- We then write $g_T : T$ for the induced witness of the original type expression T, understood schematically and, where required, relative to context.
- We apply this method to produce witnesses for logically true propositions in propositional, first-order, second-order, and higher-order classical logic.

In many cases these witnesses are *contextual*: they depend on externally supplied inhabitants in a surrounding context rather than being closed canonical terms.

The novelty claimed here is modest and specific. The point is not that classical logic admits some propositions-as-types interpretation, but that many classical principles may be handled uniformly by a small collection of truth-value reduction rules together with a canonical reduced witness of identity form, without enriching the term language with dedicated control constructs. These type-level truth-value reductions do the work otherwise assigned to control operators, use context just like control constructions and use hypothetical reasoning (such as "if a type is inhabited then ...", or "if a type is uninhabited then..."), while the passage back to the original type is recorded by the induced witness notation.

2. Comparison with related approaches

The approach taken in this paper stands in contrast to three prominent families of work: classical Curry–Howard interpretations via control operators (Griffin, Parigot), Krivine's classical realizability, and the double-negation translations of Gödel–Gentzen and Friedman. We briefly situate the present framework with respect to each.

Classical Curry–Howard via control operators

A first line of work realises classical logic by extending the term language with control operators. Griffin's classic paper [17] shows that a Scheme-like `call/cc` operator can be given a type corresponding to Peirce's law, thereby providing a formulae-as-types interpretation of classical logic in which proofs are represented by programs with first-class continuations. Parigot's $\lambda\mu$-calculus [19] refines this perspective by introducing explicit continuation variables and μ-abstractions, yielding an extension of the typed λ-calculus that corresponds closely to classical natural deduction; normalization and consistency are then established via CPS and negative-translation techniques.

The present framework deliberately avoids adding such control constructs to the term language. Instead, classical reasoning is internalised at the level of types: Boolean inhabitation semantics and the reduction principles of Definition 4.3 collapse many classical types to the canonical identity form $\bot \to \bot$, and the induced witness $g_T : T$ records inhabitation of the original type schematically. Compared with Griffin's and Parigot's calculi, this trades computational richness for simplicity: witnesses are in general contextual rather than closed control terms, but the underlying term calculus remains an ordinary typed λ-calculus without additional operators.

Krivine's classical realizability

Krivine's classical realizability [20] extends the Curry–Howard correspondence to classical second-order arithmetic by enriching the term language with a control operator and interpreting programs via a process machine (the "λ_c-calculus"). A classical proof is represented as a closed term, and realizability is defined as a relation between terms and *stacks*, with a distinguished pole encoding computational adequacy. One important outcome is that realizers are executable: given a realizer of a Σ^0_1 formula $(\exists x : A)P(x)$, one can run the term against an appropriate stack to extract a concrete witness $t : A$ together with evidence $P(t)$; see, for example, [22] for a detailed account of witness extraction in this framework.

The present paper takes a strictly different path. We work entirely within an ordinary typed λ-calculus and introduce no control operators or process-machine semantics. In place of computational extraction, we use type-level Boolean reductions to establish inhabitation at a coarse truth-value level, and the induced witness $g_T : T$ records the resulting inhabitant schematically. The witnesses produced are in general *contextual* rather than closed: they depend on inhabitants supplied by a surrounding context Γ, which function as Henkin-style witness variables (Definition 3.13). This is an explicit trade-off. Classical realizability provides computationally richer, executable witnesses at the cost of extending the term language; the present framework provides uniform inhabitation certificates within an unextended typed λ-calculus at the cost of the witnesses being schematic.

Double-negation translations

The Gödel–Gentzen negative translation [23][24] embeds classical logic into intuitionistic logic by a systematic replacement of subformulas with their double negations. Concretely, given a classical formula φ, its negative translate φ^N prefixes (in one standard variant) atomic subformulas and disjunctions with

$\neg\neg$, while leaving implication and universal quantification unchanged. The fundamental result is that φ is classically provable if and only if φ^N is intuitionistically provable. Friedman's translation [25] is a further refinement: it replaces $\bot$ throughout by an arbitrary formula R, yielding a parametrised translation $(\cdot)^R$ that is useful in particular for witness extraction and conservativity arguments.

These translations operate at the *formula* level: they produce a new formula which is then given an ordinary intuitionistic proof. The relationship to the present paper is as follows. The definitions of disjunction and existential quantification adopted here,

$$A \vee B := (A \to \bot) \to B, (\exists x : A)P(x) := ((\forall x : A)(P(x) \to \bot)) \to \bot,$$

are exactly the double-negation normal forms that a negative translation would produce from the standard connectives. In this sense, the present framework may be read as working directly inside the negative-translation image of classical logic, taking those classically equivalent definitions as primitive. However, the purpose is different: rather than establishing a conservativity embedding into intuitionistic logic, we remain within a classical Boolean meta-logic and identify a uniform family of type-level reductions that witness the inhabitation of the resulting types. The induced witness g_T records that a type is inhabited in virtue of its reduction to canonical identity form $\bot \to \bot$; no appeal is made to intuitionistic proof rules for the translated formula.

Summary

As observed in [22], in the Σ_1^0 case Krivine's witness-extraction method can be related systematically to Friedman's negative translation via an embedding into intuitionistic second-order arithmetic. Since the definitions of disjunction and existential quantification adopted here coincide with the negative-translation images of those connectives, the present framework sits, in a loose sense, at the intersection of these approaches: it shares with the negative translation the choice of classically equivalent connective definitions, and shares with classical realizability the goal of assigning witnesses to classical principles without passing to an intuitionistic meta-theory. The specific contribution claimed here is the identification of a small family of type-level reduction rules (Definition 4.3) and the induced witness notation as a uniform device for this purpose, within an ordinary typed λ-calculus and without the overhead of control-operator or process-machine semantics.

3. Preliminaries

We work in a typed λ-calculus with kinds $Type$ and $Prop$, and with a Boolean type

$$Bool := \{\top, \bot\}.$$

We use the same symbols $A, B, \ldots$ for types and the corresponding propositions where there is no risk of confusion. The general background is standard; see [1][7][12].

Definition 3.1. *If A is a type and a is a term, then $a : A$ means that a inhabits A. We write informally*

$$A = \top$$

to mean that A is inhabited, and

$$A = \bot$$

to mean that A is uninhabited.

Remark 3.2. *The meta-logic is classical. Thus for each type A, either $A = \top$ or $A = \bot$.*

Remark 3.3 (Bracketing conventions). *We adopt the standard association conventions of the typed λ-calculus. Function types associate to the right, so that*

$$A \to B \to C$$

means

$$A \to (B \to C).$$

Lambda abstraction also associates to the right, so that

$$\lambda x : A \lambda y : B t$$

means

$$\lambda x : A. (\lambda y : B. t).$$

Application associates to the left, so that

$$stu$$

means

$$(st)u.$$

Thus

$$\lambda x : A \lambda y : B f x y$$

is parsed as

$$\lambda x : A.\, (\lambda y : B.\, ((f x) y)).$$

We nevertheless insert brackets explicitly whenever this improves readability.

Definition 3.4 (Primitive connectives). *Implication is represented by the function type:*

$$A \Rightarrow B \quad \text{corresponds to} \quad A \to B.$$

Negation is defined by

$$\neg A := A \to \bot.$$

Definition 3.5 (Derived connectives). *We define*

$$A \vee B := (A \to \bot) \to B$$

and

$$A \wedge B := (A \to (B \to \bot)) \to \bot.$$

Remark 3.6. *The connectives defined here are intended to capture classical truth conditions under the Boolean inhabitation semantics, not necessarily the usual constructive introduction and elimination rules. In particular, some derived connectives admit induced witnesses rather than canonical internal constructors.*

Remark 3.7. *The chosen definition of conjunction is classically equivalent to the usual conjunction, but its introduction and elimination rules below rely on reduction to identity form and induced witnessing rather than on an internal pairing constructor with projections.*

The chosen definition of disjunction is classically equivalent to more familiar formulations built from negation and implication. We use the form

$$A \vee B := (A \to \bot) \to B$$

because, in the present framework, it leads to especially transparent type-level reductions. In particular, excluded middle takes the form

$$A \vee \neg A = ((A \to \bot) \to (A \to \bot)).$$

Consequently, $A \vee B$ does not carry standard left/right tags; it is tailored to make certain classical principles, such as excluded middle, reduce to identity types at the truth-value level.

Definition 3.8 (First-order quantifiers). *Let A be a non-empty type and let $P : A \to Bool$ be a Boolean-valued predicate.*

A dependent type over A is a family of types indexed by elements of A. Thus P may be viewed as assigning to each $x : A$ a truth value $P(x)$, and correspondingly a proposition depending on x.

A dependent function on such a family is a function p such that, for each $x : A$, the value $p(x)$ has the type assigned to x. Accordingly, the universal quantifier is interpreted as the type of dependent functions:

$$(\forall x : A)P(x)$$

is inhabited exactly when there is a function p such that, for every $x : A$, $p(x) : P(x)$.

We define existential quantification by classical negation:

$$(\exists x : A)P(x) := ((\forall x : A)(P(x) \to \bot)) \to \bot.$$

Definition 3.9 (Higher-order predicate types and quantifiers). *Let A be a non-empty type. Define*

$$TR(1)(A) := A \to Bool, TR(n+1)(A) := TR(n)(A) \to Bool$$

for $n \in \mathbb{N}, n > 0$.

An element of $TR(1)(A)$ is a Boolean-valued predicate on A. An element of $TR(2)(A)$ is a Boolean-valued predicate on predicates on A, and so on.

If $P : TR(n)(A)$ and $S(P)$ is a proposition depending on P, then the universal quantifier

$$(\forall P : TR(n)(A))S(P)$$

is interpreted as the type of dependent functions assigning to each $P : TR(n)(A)$ a witness of $S(P)$.

The corresponding existential quantifier is defined by classical negation:

$$(\exists P : TR(n)(A))S(P) := ((\forall P : TR(n)(A))(S(P) \to \bot)) \to \bot.$$

Remark 3.10. *The case $n = 1$ corresponds to second-order predicate logic, since $TR(1)(A) = A \to Bool$. Thus second-order quantification over predicates on A is a special case of the higher-order scheme in Definition 3.9.*

Definition 3.11 (Canonical reduced witness and induced witness). *We assume an identity inhabitant*

$$i_{\bot\to\bot} : \bot \to \bot.$$

Whenever a type expression T reduces at truth-value level to

$$\bot \to \bot,$$

we call $i_{\bot\to\bot}$ the canonical reduced witness of the reduced form of T.

A witness of the original type expression T is then written

$$g_T : T$$

and is called the induced witness of T. This notation means that T is read as inhabited in virtue of its reduction to $\bot \to \bot$, with the induced witness interpreted schematically and, in general, relative to a context supplying the relevant inhabitants.

Remark 3.12. *The induced witness $g_T : T$ is not identified literally with*

$$i_{\bot\to\bot} : \bot \to \bot,$$

since these expressions have different types. Rather, $i_{\bot\to\bot}$ is the canonical witness of the reduced identity form of T, while g_T denotes the corresponding witness of the original type expression T under that reduction.

Definition 3.13 (Contextual interpretation of induced witnesses). *Let T be a type expression such that, under assumptions collected in a context Γ, the type T reduces at truth-value level to*

$$\bot \to \bot.$$

Then the induced witness

$$g_T : T$$

is interpreted relative to Γ according to the outer logical form of T.

The contextual data supplied by Γ function as Henkin-style witness variables: they are variables or witnesses fixed in the surrounding context relative to which the induced witness is read at the original type. They need not be given by a closed canonical term extracted from the reduced identity form itself; rather, they provide the contextual parameters that determine how the induced witness is interpreted.

- *If*

  $T = A \to B,$

  then

$g_{A \to B} : A \to B$

denotes the induced witness of the implication type. It is interpreted as a schematic function which, when applied to a contextual witness of A, yields the corresponding contextual witness of B.

- *If*

  $T = A \wedge B$,

  then

  $g_{A \wedge B} : A \wedge B$

  denotes the induced witness of the conjunction type. It is interpreted relative to contextual witness data

  $a : A, b : B$.

- *If*

  $T = A \vee B$,

  then

  $g_{A \vee B} : A \vee B$

  denotes the induced witness of the disjunction type. Since

  $A \vee B := (A \to \bot) \to B$

  by Definition 3.5, its contextual interpretation is governed by the implication clause together with the contextual interpretation of the codomain. In particular, if the context Γ supplies a contextual witness of B, then

  $g_{A \vee B}$

  is read as the corresponding schematic map from a refutation of A to that contextual witness of B. Dually, in classically equivalent situations where the disjunction reduces to identity form directly, the induced witness is read through that reduction without constructive left/right tags.

- *If*

  $T = (A \to \bot) \to \bot$,

  then

  $g_{(A \to \bot) \to \bot} : (A \to \bot) \to \bot$

  denotes the induced witness of the double-negation type. It is interpreted relative to whatever contextual witness of A is supplied by Γ.

- *If*

  $T = (\exists x : A) P(x)$,

  then

  $g_{(\exists x : A) P(x)} : (\exists x : A) P(x)$

denotes the induced witness of the first-order existential type. It is interpreted relative to contextual witness data

$a : A, p(a) : P(a),$

where these contextual data function as Henkin-style witness variables for the existential statement.

- *If*

  $T = (\exists P : TR(n)(A))S(P),$

  then

  $g_{(\exists P:TR(n)(A))S(P)} : (\exists P : TR(n)(A))S(P)$

  denotes the induced witness of the higher-order existential type. It is interpreted relative to contextual witness data

  $Q : TR(n)(A), s(Q) : S(Q),$

  again functioning as Henkin-style witness variables.

4. Truth functions and type-level reductions

We begin with the four basic function types determined by $\top$ and $\bot$.

Proposition 4.1. *The following hold:*

$$\bot \to \bot = \top, \bot \to \top = \top, \top \to \top = \top, \top \to \bot = \bot.$$

Proof. The type $\bot \to \bot$ is inhabited by the identity function $i_{\bot \to \bot}$, hence it is inhabited.

The type $\bot \to \top$ is also inhabited. To see this, let B be any inhabited type and choose some $b : B$. A term of type $\bot \to B$ may be written

$$(\lambda x : \bot)b.$$

This is well-typed because a function from $\bot$ has no actual input to process: there is no inhabitant $x : \bot$, so the requirement that the function assign an output to each input is vacuously satisfied. We can also write the term as $f : \bot \to B$ such that $f() : B$. Hence in both approaches $\bot \to B$ is inhabited whenever B is inhabited, and in particular $\bot \to \top = \top$.

The type $\top \to \top$ is inhabited, for example by the identity on an inhabited type. By contrast, $\top \to \bot$ is uninhabited, since a total function from a non-empty type to the empty type would assign an element of $\bot$ to each inhabitant of $\top$, which is impossible. □

Remark 4.2. *Proposition 4.1 gives the truth table used throughout the paper. At this stage, nothing specifically classical has yet occurred beyond the adoption of Boolean truth values for inhabitation.*

Definition 4.3 (Type-level reduction rules). *Assume A and $P(a)$ are inhabited. At the truth-value level, we adopt the following reduction principles that can be applied recursively to a type formula:*

$$red(A \to \bot) = \bot,$$

$$red(P(a) \to \bot) = \bot$$

for $a : A$,

$$red((\forall x : A)\bot) = \bot,$$

$$red((\forall P : A \to Bool)\bot) = \bot,$$

and, more generally,

$$red((\forall P : TR(n)(A))\bot) = \bot$$

for all $n > 0$.

Remark 4.4. *These are rewrite rules on inhabitation truth values that operate recursively on a type formula, not computational reductions of terms. Their purpose is to collapse negated or universally quantified types to $\bot$ whenever the relevant domain is inhabited and some contradictory instance is forced. Negations of such types then reduce to $\bot \to \bot$, whose canonical reduced witness is identity. If we have for example $(\forall x : A)(P(x) \to \bot)$, then $red((\forall x : A)(P(x) \to \bot)) = (\forall x : A)\bot$ and $red((\forall x : A)\bot) = \bot$. Hence $red((\forall x : A)(P(x) \to \bot)) = \bot$.*

5. Identity form and induced witnessing

The original motivation of the paper is clarified by viewing truth functions as operators on types.

Definition 5.1. *We interpret the four basic function types as follows:*

- *$\bot \to \bot$ as the canonical identity form, inhabited by $i_{\bot\to\bot}$;*
- *$\bot \to \top$ as a vacuous function type, inhabited because the codomain is inhabited;*
- *$\top \to \top$ as the ordinary case of a type inhabited by a function between inhabited types, with identity as the canonical example;*
- *$\top \to \bot$ as uninhabited at term level.*

doi.org/10.32388/IY3BKH.4

Remark 5.2. *The mathematically important point is the inhabitation behaviour described in Proposition 4.1. In the present framework, the privileged role of $\bot \to \bot$ is that it serves as the canonical reduced identity form to which many classical types collapse.*

Theorem 5.3 (Ex falso). *The type $\bot \to A$ is inhabited for every type A.*

Proof. By classical bivalence, either $A = \top$ or $A = \bot$.

If $A = \top$, then $\bot \to A$ has truth value

$$\bot \to \top = \top$$

by Proposition 4.1, so it is inhabited in the Boolean inhabitation semantics. Concretely, for any chosen $a : A$, the term

$$\lambda x : \bot . a : \bot \to A$$

is well-typed and vacuously total.

If $A = \bot$, then $\bot \to A$ is $\bot \to \bot$, which has canonical inhabitant

$$i_{\bot \to \bot} : \bot \to \bot$$

by Proposition 4.1. Hence $\bot \to A$ is inhabited in this case as well.

Thus in all cases $\bot \to A$ is inhabited. □

Theorem 5.4 (Double negation at type level). *If A is inhabited, then*

$$(A \to \bot) \to \bot$$

reduces at type level to

$$\bot \to \bot.$$

If A is uninhabited, then

$$(A \to \bot) \to \bot = \bot.$$

Proof. Assume first that A is inhabited. By Definition 4.3,

$$red(A \to \bot) = \bot.$$

Hence

$$(A \to \bot) \to \bot$$

reduces to

$$\bot \to \bot.$$

By Proposition 4.1, this type is inhabited.

Now assume that A is uninhabited. Then $A = \bot$, so $A \to \bot$ is inhabited and has truth value $\top$. Therefore

$$(A \to \bot) \to \bot$$

has truth value

$$\top \to \bot = \bot$$

by Proposition 4.1. □

Corollary 5.5. *A type A is inhabited if and only if $\neg\neg A$ is inhabited.*

Proof. Immediate from Theorem 5.4. □

Proposition 5.6 (Induced witness extraction). *Let T be a type expression occurring in a derivation, and suppose that, under assumptions collected in a context Γ, the type T reduces at truth-value level to*

$$\bot \to \bot.$$

Then T is inhabited at the truth-value level, with canonical reduced witness

$$i_{\bot\to\bot} : \bot \to \bot.$$

If, in addition, the context Γ supplies the data needed to interpret the original type expression T, we write

$$g_T : T$$

for the induced witness of T relative to Γ.

Proof. By hypothesis, T reduces at truth-value level to

$$\bot \to \bot.$$

Hence T is inhabited in the coarse Boolean sense used throughout the paper. The reduced form has canonical witness

$$i_{\bot\to\bot} : \bot \to \bot.$$

Accordingly, the original type expression T is assigned an induced witness

$$g_T : T,$$

obtained by reading T through its reduction to the canonical identity form. When the construction represented by T depends on inhabitants supplied by a context Γ, this induced witness is interpreted relative to Γ. □

Remark 5.7. *The truth-value computation alone shows only that a type is inhabited in a coarse Boolean sense. A witness of the original type is recorded by the induced witness notation g_T, and in general this witness is schematic or contextual rather than a closed canonical term of the underlying term calculus.*

6. First-order classical propositional logic

The presentation here follows the general spirit of classical logical systems in the tradition of [26], while also using introduction and elimination rules familiar from natural deduction; see [24] and [27].

Theorem 6.1 (Implication introduction). *The type*

$$A \to (B \to A)$$

is inhabited.

Proof. A witness is the standard lambda term

$$\lambda x : A.\, \lambda y : B.\, x.$$

□

Theorem 6.2 (Modus ponens). *The type*

$$A \to ((A \to B) \to B)$$

is inhabited.

Proof. A witness is the standard lambda term

$$\lambda x : A.\, \lambda y : A \to B.\, y(x).$$

□

Theorem 6.3 (Falsum introduction). *The type*

$$A \to ((A \to \bot) \to \bot)$$

is inhabited.

Proof. Suppose first that A is inhabited. Then by Definition 4.3,

$$A \to \bot$$

reduces to $\bot$, so

$$(A \to \bot) \to \bot$$

reduces to

$$\bot \to \bot.$$

Hence $(A \to \bot) \to \bot$ is inhabited at truth-value level, with canonical reduced witness

$$i_{\bot \to \bot} : \bot \to \bot.$$

An induced witness for

$$(A \to \bot) \to \bot$$

may therefore be written as

$$g_{(A \to \bot) \to \bot}.$$

By Definition 3.13, this induced witness is interpreted relative to whatever contextual witness of A is supplied when the implication type is used. Accordingly, a witness for

$$A \to ((A \to \bot) \to \bot)$$

may be written schematically as

$$\lambda x : A.\, g_{(A \to \bot) \to \bot}.$$

We note that

$$(\lambda x : A.\, g_{(A \to \bot) \to \bot})(a) = g_{(A \to \bot) \to \bot}$$

for $a : A$.

If A is uninhabited, then the domain of the outer implication is empty, so the type is vacuously inhabited (by Ex falso, Theorem 5.3). □

Theorem 6.4 (Contextual double negation elimination). *Let Γ be a context containing an inhabitant $a : A$. Then, relative to Γ, the type*

$$((A \to \bot) \to \bot) \to A$$

is inhabited.

Proof. Assume $\Gamma \vdash a : A$. Then A is inhabited in the context Γ. By Definition 4.3,

$$red(A \to \bot) = \bot.$$

Hence, at truth-value level,

$$(A \to \bot) \to \bot$$

reduces to

$$\bot \to \bot.$$

By Proposition 4.1, the reduced form is inhabited, with canonical reduced witness

$$i_{\bot \to \bot} : \bot \to \bot.$$

Accordingly, by Proposition 5.6, the original type admits the induced witness

$$g_{(A \to \bot) \to \bot} : (A \to \bot) \to \bot.$$

By Definition 3.13, this induced witness is interpreted relative to the contextual witness

$$a : A,$$

which here functions as a Henkin-style witness variable fixed by Γ. A witness for

$$((A \to \bot) \to \bot) \to A$$

may therefore be written as

$$\lambda k : (A \to \bot) \to \bot.\, a.$$

We note that

$$(\lambda k : (A \to \bot) \to \bot.\, a)(g_{(A \to \bot) \to \bot}) = a.$$

□

doi.org/10.32388/IY3BKH.4

Remark 6.5. *Theorem 6.4 does not assert that double negation elimination is witnessed by a closed term constructed independently of assumptions. Rather, once a context already fixes an inhabitant of A, double negation elimination reduces to a function from an induced witness of the double negation type to the inhabitant fixed by the context. This is similar in spirit to Griffin's construction [17], where continuation-based control allows a program to ignore intermediate contexts and return a pre-supplied witness.*

Theorem 6.6 (Excluded middle). *The type*

$$A \vee \neg A$$

is inhabited.

Proof. By definition,

$$A \vee \neg A = ((A \to \bot) \to (A \to \bot)).$$

This is literally an identity type, hence inhabited by

$$i_{(A\to\bot)\to(A\to\bot)}.$$

As noted in Definition 3.13, the present disjunction does not carry constructive left/right tags; in this case it is read directly through its identity form. □

Remark 6.7. *With the present definition of disjunction, excluded middle is represented by an explicit identity type. More generally, disjunction is treated here as a classically behaved derived connective whose type-level behaviour is tailored to the reduction rules used throughout the paper.*

Theorem 6.8 (Conjunction introduction). *If A and B are inhabited, then*

$$A \wedge B$$

is inhabited.

Proof. By definition,

$$A \wedge B = (A \to (B \to \bot)) \to \bot.$$

If A and B are inhabited, then $B \to \bot$ reduces to $\bot$, hence

$$A \to (B \to \bot)$$

reduces to $A \to \bot$, which in turn reduces to $\bot$. Therefore

$$(A \to (B \to \bot)) \to \bot$$

reduces to

$$\bot \to \bot.$$

Hence $A \wedge B$ is inhabited at truth-value level, with canonical reduced witness

$$i_{\bot \to \bot} : \bot \to \bot.$$

We therefore write

$$g_{A \wedge B} : A \wedge B$$

for the induced witness of the conjunction type. By Definition 3.13, this induced witness is interpreted relative to contextual witness data

$$a : A, b : B,$$

which function here as Henkin-style witness variables for the conjunction. □

Theorem 6.9 (Conjunction elimination, contextual form). *If A and B are inhabited relative to the context supplying A and B, then both*

$$A \wedge B \to A \qquad \text{and} \qquad A \wedge B \to B$$

are inhabited.

Proof. Assume that the context supplies inhabitants $a : A$ and $b : B$. As in the proof of conjunction introduction, the type

$$A \wedge B$$

reduces to

$$\bot \to \bot.$$

Hence $A \wedge B$ is inhabited at truth-value level, with canonical reduced witness $i_{\bot \to \bot}$ and induced witness $g_{A \wedge B} : A \wedge B$. By Definition 3.13, this induced witness is interpreted relative to the contextual witness data

$$a : A, b : B,$$

doi.org/10.32388/IY3BKH.4

which function here as Henkin-style witness variables for the conjunction.

Relative to the supplied inhabitants, the elimination types admit induced witnesses

$$\lambda z : A \wedge B.\, a \qquad \text{and} \qquad \lambda z : A \wedge B.\, b.$$

We note that $((\lambda z : A \wedge B)a)g_{A\wedge B} = a$ and $((\lambda z : A \wedge B)b)g_{A\wedge B} = b$.

Thus both

$$A \wedge B \to A \qquad \text{and} \qquad A \wedge B \to B$$

are inhabited relative to context. □

Remark 6.10. *Unlike in constructive treatments, we do not define projections $\pi_1, \pi_2 : A \wedge B \to A, B$ that extract components from a canonical pair. Instead, elimination proceeds by reducing $A \wedge B$ to canonical identity form at the truth-value level and then assigning induced witnesses relative to contextually supplied inhabitants.*

7. First-order predicate logic

We now pass to first-order quantification. The main novelty here is not the witness terms for universal quantification, which are standard, but the treatment of existential quantification by reduction to double negation. General background on such type-theoretic treatments may be found in [12].

Theorem 7.1 (Universal introduction). *Let $P : A \to Bool$. If, in context Γ under the assumption variable $\Gamma \vdash a : A$ not free in P, one has a witness*

$$\Gamma \vdash p(a) : P(a),$$

then

$$\Gamma \vdash (\forall x : A)P(x)$$

is inhabited.

Proof.

If

$$\Gamma \vdash p(a) : P(a)$$

for variable $a : A$ then

$$\Gamma \vdash (\lambda a : A)p(a) : (\forall a : A)P(a)$$

if variable a is not free in P by definition of universal quantification. The result follows by relabelling the bound variable to x. □

Theorem 7.2 (Universal elimination). *If*

$$\Gamma \vdash p : (\forall x : A) P(x)$$

and term $t : A$*, then*

$$\Gamma \vdash p(t) : P(t).$$

Equivalently, the type

$$(\forall x : A) P(x) \to P(t)$$

is inhabited if $\Gamma \vdash p(t) : P(t)$*.*

Proof.

Since a witness of

$$(\forall x : A) P(x)$$

is, by definition, a dependent function assigning to each $x : A$ a witness of $P(x)$, application at t yields

$$p(t) : P(t).$$

□

Theorem 7.3 (Existential introduction). *If* $\Gamma \vdash t : A$ *and* $\Gamma \vdash p(t) : P(t)$*, then, relative to* Γ*, the type*

$$(\exists x : A) P(x)$$

is inhabited.

Proof. By definition,

$$(\exists x : A) P(x) = ((\forall x : A)(P(x) \to \bot)) \to \bot.$$

Assume $\Gamma \vdash t : A$ and $\Gamma \vdash p(t) : P(t)$. Then the type

$$P(t) \to \bot$$

is uninhabited in the context Γ by Definition 4.3, since any inhabitant of it would send $p(t) : P(t)$ to an inhabitant of $\bot$.

Therefore the universally quantified type

$$(\forall x : A)(P(x) \to \bot)$$

is uninhabited in context Γ by Definition 4.3, because any inhabitant of it would in particular yield an inhabitant of $P(t) \to \bot$ when evaluated at t.

Hence the existential type

$$(\exists x : A)P(x)$$

reduces to

$$\bot \to \bot.$$

Therefore it is inhabited at truth-value level, with canonical reduced witness

$$i_{\bot \to \bot} : \bot \to \bot.$$

Accordingly, the original existential type admits the induced witness

$$g_{(\exists x:A)P(x)} : (\exists x : A)P(x).$$

By Definition 3.13, this induced witness is interpreted relative to the contextual witness data

$$t : A, p(t) : P(t),$$

which function as Henkin-style witness variables for the existential statement. □

Theorem 7.4 (Existential elimination, contextual form). *Let A be a non-empty type and let $P : A \to Bool$. Assume that*

$$\Gamma \vdash e : (\exists x : A)P(x),$$

and suppose that for a variable $a : A$

$$\Gamma, a : A, p(a) : P(a) \vdash q : Q,$$

where a is not free in Q. Then Q is inhabited relative to Γ.

Proof. By definition,

$$(\exists x : A)P(x) = ((\forall x : A)(P(x) \to \bot)) \to \bot.$$

Thus the assumption

$$\Gamma \vdash e : (\exists x : A)P(x)$$

means that the type

$$((\forall x : A)(P(x) \to \bot)) \to \bot$$

is inhabited in the context Γ. Let

$$X := (\forall x : A)(P(x) \to \bot).$$

Since $X \to \bot$ is inhabited, we have $X \to \bot = \top$. By classical bivalence, either $X = \top$ or $X = \bot$. If $X = \top$, then Proposition 4.1 gives

$$X \to \bot = \top \to \bot = \bot,$$

contradiction. Hence $X = \bot$. Therefore

$$(\forall x : A)(P(x) \to \bot) = \bot,$$

and so

$$(\exists x : A)P(x) = ((\forall x : A)(P(x) \to \bot)) \to \bot$$

reduces to

$$\bot \to \bot.$$

So $(\exists x : A)P(x)$ has canonical reduced witness

$$i_{\bot \to \bot} : \bot \to \bot$$

and induced witness

$$g_{(\exists x:A)P(x)} : (\exists x : A)P(x).$$

By Definition 3.13, this induced witness is interpreted relative to contextual witness data

$$a : A, p(a) : P(a),$$

which function as Henkin-style witness variables for the existential statement. By hypothesis, from such a witnessing instance one may derive a witness of Q. Hence Q is inhabited relative to Γ. □

Remark 7.5. *Again, the conclusion that Q is inhabited is relative to Γ and to the witnessing instance(s) used in the derivation of $q : Q$; the framework does not provide a closed eliminator term for $(\exists x : A)P(x)$. The point is*

that a witnessing instance forces the existential type to reduce to canonical identity form, from which an induced witness of the original existential type is recorded.

8. Higher-order predicate logic

The higher-order case is a finite iteration of the second-order pattern. Formally, second-order logic corresponds to the case $n = 1$ in the hierarchy $TR(n)(A)$ from Definition 3.9; we therefore treat all orders uniformly in what follows.

Theorem 8.1 (Higher-order universal introduction). *If, under the assumption $P : TR(n)(A)$, one has a witness of $S(P)$, then*

$$(\forall P : TR(n)(A))S(P)$$

is inhabited.

Proof. Exactly as before, by abstraction on the higher-order predicate variable. A witness is of the form

$$\lambda P : TR(n)(A).\, s(P).$$

□

Theorem 8.2 (Higher-order universal elimination). *If*

$$u : (\forall P : TR(n)(A))S(P)$$

and $Q : TR(n)(A)$, then

$$u(Q) : S(Q).$$

Equivalently, the type

$$(\forall P : TR(n)(A))S(P) \to S(Q)$$

is inhabited.

Proof. A witness is

$$\lambda u : (\forall P : TR(n)(A))S(P).\, u(Q).$$

□

Theorem 8.3 (Higher-order existential introduction). *Let* $\Gamma \vdash Q : TR(n)(A)$ *and suppose* $\Gamma \vdash s : S(Q)$. *Then, relative to* Γ,

$$(\exists P : TR(n)(A))S(P)$$

is inhabited.

Proof. By definition,

$$(\exists P : TR(n)(A))S(P) = ((\forall P : TR(n)(A))(S(P) \to \bot)) \to \bot.$$

Since $\Gamma \vdash s : S(Q)$, the type

$$S(Q) \to \bot$$

is uninhabited. Therefore

$$(\forall P : TR(n)(A))(S(P) \to \bot)$$

is uninhabited, because any inhabitant of it would yield an inhabitant of $S(Q) \to \bot$ when evaluated at Q. Therefore

$$((\forall P : TR(n)(A))(S(P) \to \bot)) \to \bot$$

reduces to

$$\bot \to \bot,$$

which is inhabited at the truth-value level, with canonical reduced witness $i_{\bot \to \bot}$. The original existential type therefore admits the induced witness

$$g_{(\exists P : TR(n)(A))S(P)} : (\exists P : TR(n)(A))S(P).$$

By Definition 3.13, this induced witness is interpreted relative to the contextual witness data

$$Q : TR(n)(A), s(Q) : S(Q),$$

which function as Henkin-style witness variables for the higher-order existential statement. □

Theorem 8.4 (Higher-order existential elimination, contextual form). *Let* A *be a non-empty type and let* $n > 0$. *Assume that*

$$\Gamma \vdash e : (\exists P : TR(n)(A))S(P),$$

and suppose that for an arbitrary $Q : TR(n)(A)$ one has

$$\Gamma, Q : TR(n)(A), s(Q) : S(Q) \vdash r : R,$$

where Q is not free in R. Then R is inhabited relative to Γ.

Proof. By Definition 3.9,

$$(\exists P : TR(n)(A))S(P) = ((\forall P : TR(n)(A))(S(P) \to \bot)) \to \bot.$$

Thus the assumption

$$\Gamma \vdash e : (\exists P : TR(n)(A))S(P)$$

means that the type

$$((\forall P : TR(n)(A))(S(P) \to \bot)) \to \bot$$

is inhabited in the context Γ. Let

$$X := (\forall P : TR(n)(A))(S(P) \to \bot).$$

Since $X \to \bot$ is inhabited, we have $X \to \bot = \top$. By classical bivalence, either $X = \top$ or $X = \bot$. If $X = \top$, then Proposition 4.1 gives

$$X \to \bot = \top \to \bot = \bot,$$

contradiction. Hence $X = \bot$. Therefore

$$(\forall P : TR(n)(A))(S(P) \to \bot) = \bot,$$

and so

$$(\exists P : TR(n)(A))S(P) = ((\forall P : TR(n)(A))(S(P) \to \bot)) \to \bot$$

reduces to

$$\bot \to \bot.$$

So $(\exists P : TR(n)(A))S(P)$ has canonical reduced witness

$$i_{\bot \to \bot} : \bot \to \bot$$

and induced witness

$$g_{(\exists P:TR(n)(A))S(P)} : (\exists P : TR(n)(A))S(P).$$

By Definition 3.13, this induced witness is interpreted relative to contextual witness data

$$Q : TR(n)(A), s(Q) : S(Q),$$

which function as Henkin-style witness variables for the higher-order existential statement. By the elimination hypothesis

$$\Gamma, Q : TR(n)(A), s(Q) : S(Q) \vdash r : R,$$

such a witnessing instance suffices to derive a witness of R. Hence R is inhabited relative to Γ. □

Remark 8.5. *At higher order the witness pattern is unchanged: a singular witness $Q : TR(n)(A)$ and an instantiated witness $s(Q) : S(Q)$ force the existential type to reduce to canonical identity form. As in the first-order case, the resulting inhabitant is contextual and is recorded by an induced witness of the original existential type.*

9. Discussion

The formal development above relies on a distinction between truth-value inhabitation and induced witnessing. The ambient reasoning is carried out in a classical meta-logic, and all uses of “true”, “logical truth”, and “inhabited” are to be read in this meta-theoretic Boolean sense. A type-level reduction to

$$\bot \to \bot$$

shows that a target type is inhabited in the coarse Boolean sense relevant to classical logic. The canonical reduced witness of this identity form is

$$i_{\bot\to\bot} : \bot \to \bot.$$

A witness of the original type, however, is represented by an induced witness

$$g_T : T,$$

obtained by reading the original type through its reduction to the canonical identity form, and in general interpreted relative to context.

This explains why the present account is best suited to logical truths. For many such principles, the essential point is that a type is inhabited uniformly in virtue of its logical form. The witness may then be

represented by its canonical reduced witness together with the corresponding induced witness notation at the original type. The method is not intended to replace more informative constructive proofs in settings where the goal is to compute a specific mathematical object.

The contextual interpretation also clarifies the role of the externally supplied witness data. In existential and related cases, the relevant contextual variables function in the manner of Henkin-style witness variables: they fix the parameters relative to which the induced witness of the original type is read, without requiring a closed witness term to be extracted directly from the reduced identity form.

The relation with continuation-based interpretations is also clearer in this form. Those approaches enrich the term language to encode classical reasoning operationally [17][18][19][21]. The present account instead keeps the term language ordinary and shifts the classical component into type-level reductions and classical meta-logic. Whether one regards this as a semantic alternative or chiefly as an expository reorganisation, it isolates the main mechanism in a simple and uniform way.

10. Conclusion

We have presented a theorem–proof style account of a propositions-as-types interpretation of classical logic based on Boolean inhabitation semantics, the truth values of the four basic function types, and a small family of type-level reduction rules.

With these ingredients in place, many classical principles can be treated uniformly. Propositional principles such as excluded middle and contextual double negation elimination, first-order existential reasoning, and their second- and higher-order analogues all reduce to the same core pattern: a type is shown to reduce to

$$\bot \to \bot,$$

whose canonical reduced witness is the identity inhabitant

$$i_{\bot\to\bot} : \bot \to \bot.$$

A witness of the original type is then treated as an induced witness, read from this reduced identity form and interpreted schematically or contextually as required.

Appendix A. Further examples of witness construction

This appendix records several additional examples illustrating the methods developed in the main text.

Proposition A.1 (Curry's rule). *The type*

$$(A \to (B \to C)) \to ((A \to B) \to (A \to C))$$

is inhabited.

Proof. A witness is

$$\lambda f : A \to (B \to C).\, \lambda g : A \to B.\, \lambda a : A.\, (f(a))(g(a)).$$

□

Proposition A.2 (Peirce's law with contextual witness). *The proposition*

$$((A \to B) \to A) \to A$$

is logically true. Here "logically true" means that for every assignment of $\top/\bot$ to A and B consistent with the Boolean inhabitation semantics, the resulting type evaluates to $\top$. Moreover, if Γ is a context containing an inhabitant

$$a : A,$$

then the type

$$((A \to B) \to A) \to A$$

is inhabited relative to Γ.

Proof. First we verify logical truth. If $A = \top$, then the codomain is inhabited, so the whole implication is inhabited. If $A = \bot$, then

$$A \to B = \top,$$

and therefore

$$(A \to B) \to A = \top \to \bot = \bot.$$

Hence

$$((A \to B) \to A) \to A = \bot \to \bot = \top.$$

Thus the proposition is logically true in all cases.

Now assume

$$\Gamma \vdash a : A.$$

Then the target type is inhabited relative to Γ by the induced witness

$$\lambda h : (A \to B) \to A.\, a.$$

This is the contextual analogue of an induced witness at the original type. □

Proposition A.3 (A contextual existential–universal example). *Let Γ be a context containing*

$$a : A, p(a) : P(a).$$

Suppose further that there is a function

$$u : (\forall y : A)(P(a) \to P(y)).$$

Then, relative to Γ, the type

$$P(a) \to (\exists x : A)(\forall y : A)(P(x) \to P(y))$$

is inhabited.

Proof. Assume

$$\Gamma \vdash a : A, \Gamma \vdash p(a) : P(a), \Gamma \vdash u : (\forall y : A)(P(a) \to P(y)).$$

For any fixed $w : A$, Theorem 7.2 yields

$$u(w) : P(a) \to P(w).$$

In particular, applying $u(w)$ to the contextual witness $p(a) : P(a)$ gives

$$u(w)(p(a)) : P(w).$$

(Thus, relative to $p(a)$, each $P(w)$ holds, although we will not need the explicit term $\lambda y : A.\, u(y)(p(a))$ below.)

To apply Theorem 7.3, take the singular witness $t := a : A$ and let

$$Q(x) := (\forall y : A)(P(x) \to P(y)).$$

At $x = a$ we have, by assumption,

$$u : (\forall y : A)(P(a) \to P(y)) = Q(a).$$

Hence the hypotheses of Theorem 7.3 are satisfied in the context Γ, and we obtain an induced witness of

$$(\exists x : A)(\forall y : A)(P(x) \to P(y))$$

relative to Γ.

By Definition 3.13, this induced witness is interpreted relative to the contextual witness data

$$a : A, p(a) : P(a), u : (\forall y : A)(P(a) \to P(y)),$$

which function here as Henkin-style witness variables for the existential statement. Therefore the type

$$P(a) \to (\exists x : A)(\forall y : A)(P(x) \to P(y))$$

is inhabited relative to Γ. $\square$

Proposition A.4 (A purely classical existential–universal example). *Let A be a non-empty type and let $P : A \to Bool$. Assume*

$$(\forall x : A)\neg P(x).$$

Then the type

$$(\exists x : A)(\forall y : A)(P(x) \to P(y))$$

is inhabited at the truth-value level.

Proof. Since A is non-empty, fix some $a : A$. By assumption $(\forall x : A)\neg P(x)$, we have $\neg P(a)$, so $P(a) = \bot$. For any $y : A$, the type $P(a) \to P(y)$ then has truth value $\top$, since its domain is $\bot$ and Proposition 4.1 gives $\bot \to B = \top$ whenever B is inhabited. Thus

$$(\forall y : A)(P(a) \to P(y))$$

is inhabited at the truth-value level.

Now consider the existential type

$$(\exists x : A)(\forall y : A)(P(x) \to P(y)) = ((\forall x : A)(\forall y : A)(P(x) \to P(y) \to \bot)) \to \bot,$$

as in Definition 3.8. Since we have just seen that the matrix $(\forall y : A)(P(a) \to P(y))$ holds at the truth-value level, any witness of

$$(\forall x : A)(\forall y : A)(P(x) \to P(y) \to \bot)$$

would in particular yield a contradiction at $x = a$. Hence this universal type reduces to $\bot$ by Definition 4.3, and therefore its double negation reduces to $\bot \to \bot$, which is inhabited by $i_{\bot\to\bot}$. It follows that

$$(\exists x : A)(\forall y : A)(P(x) \to P(y))$$

is inhabited at the truth-value level. □

Remark A.5. *Proposition A.4 illustrates the purely classical character of the Boolean inhabitation semantics adopted here. Under the assumption $(\forall x : A)\neg P(x)$, no instance $P(x)$ is inhabited, so there is no constructive information about any particular witness of P. Nevertheless, the formula*

$$(\exists x : A)(\forall y : A)(P(x) \to P(y))$$

is validated at the truth-value level: once $P(a) = \bot$ for some $a : A$ in a non-empty type A, the implications $P(a) \to P(y)$ are trivially true for all y, and the existential quantifier merely records the existence of such an a without fixing it as a canonical term. This is a typical situation in classical reasoning: the existential is true because every potential witness vacuously satisfies the matrix, even though no positive instance of P is available. In the present framework this is reflected by the fact that $(\exists x : A)(\forall y : A)(P(x) \to P(y))$ reduces to the canonical identity form $\bot \to \bot$ at the truth-value level, while the induced witness notation does not attempt to select a distinguished closed inhabitant of A.

Theorem A.6 (Choice functional for a total relation). *Let A and B be non-empty types and let $R : A \to B \to Bool$. Consider the formula*

$$T := ((\forall x : A)(\exists y : B)R(x, y)) \to ((\exists f : A \to B)(\forall x : A)R(x, f(x)))\,.$$

Then T is classically valid under the Boolean inhabitation semantics adopted in this paper, and therefore admits an induced witness $g_T : T$.

Proof. Let

$$P := (\forall x : A)(\exists y : B)R(x, y).$$

We argue semantically at the level of truth values.

Case 1: $P = \bot$. Then the implication T has the form

$$P \to (\exists f : A \to B)(\forall x : A)R(x, f(x))$$

with domain $\bot$. By Theorem 5.3, any type of the form $\bot \to A$ is inhabited. Thus T evaluates to $\top$ and is inhabited at the truth-value level in this case.

Case 2: $P = \top$. Assume a context Γ with

$$\Gamma \vdash r : (\forall x : A)(\exists y : B)R(x, y).$$

For each $x : A$, Theorem 7.2 yields

$$\Gamma \vdash r(x) : (\exists y : B)R(x, y).$$

By the contextual form of existential elimination (Theorem 7.4), each such $r(x)$ may be read relative to a context

$$\Gamma_x := \Gamma, b_x : B, r(x, b_x) : R(x, b_x),$$

where b_x is a Henkin-style witness variable and $r(x, b_x)$ is its contextual inhabitant of $R(x, b_x)$, in the sense of Definition 3.13. Intuitively, for each $x : A$ we fix a contextual witness $b_x : B$ with $R(x, b_x)$ inhabited.

Collecting these contextual data for all $x : A$, we may define a schematic choice function

$$f : A \to B, f(x) := b_x,$$

read relative to the combined context containing all b_x and $r(x, b_x)$. For each $x : A$ we then have

$$r(x, b_x) : R(x, b_x) = R(x, f(x)),$$

so the formula

$$(\forall x : A)R(x, f(x))$$

is satisfied at the truth-value level relative to this context. In particular, the conclusion

$$(\exists f : A \to B)(\forall x : A)R(x, f(x))$$

is true in the classical sense that there exists some (possibly non-constructively specified) $f : A \to B$ satisfying the matrix. Thus T again evaluates to $\top$.

In both cases, T evaluates to $\top$ under the Boolean inhabitation semantics. Accordingly, T is inhabited at the truth-value level. □

Remark A.7. *Formally, the higher-order quantifiers in this paper range over predicates $P : TR(n)(A)$ rather than over function types $A \to B$ themselves. A fully internal treatment of the choice functional would encode a function $f : A \to B$ by its graph as a binary relation on $A \times B$, with an appropriate uniqueness condition, and then quantify over such graph predicates. Here we work at a slightly more informal level: we quantify directly over $f : A \to B$ and read f as inducing a binary relation together with a distinguished choice of $y = f(x)$. Theorem A.6 is therefore best understood as a semantic illustration of how the present truth-value reduction and induced-witness machinery accommodates a classical choice-functional principle, rather than as a completely formal derivation within a fixed higher-order syntax.*

References

1. ^a, ^b *Church A (1940). "A Formulation of the Simple Theory of Types." J Symbol Log.* **5***(2):56–68.*
2. ^ *Girard J-Y (1972). "Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supéri eur" [Functional Interpretation and Cut Elimination of Higher-Order Arithmetic].*
3. ^ *Girard J-Y (1973). Quelques résultats sur les interpretations fonctionnelles [Some Results on Functional Inte rpretations]. Springer.*
4. ^ *Girard J-Y (1989). Proofs and Types. Cambridge University Press.*
5. ^ *Girard J-Y (1986). "The System F of Variable Types, Fifteen Years Later." Theor Comput Sci.* **45***:159–192.*
6. ^ *Coquand T, Huet G (1988). "The Calculus of Constructions." Inf Comput.* **76***:95–120.*
7. ^a, ^b *Barendregt H (1991). "Introduction to Generalized Type Systems." J Funct Program.* **1***(2):125–154.*
8. ^ *Curry HB (1934). "Functionality in Combinatory Logic." Proc Natl Acad Sci U S A.* **20***(11):584–90.*
9. ^ *Curry HB, Feys R (1958). Combinatory Logic. North-Holland.*
10. ^ *Howard W (1980). "The Formulae-as-Types Notion of Construction." Academic Press.*
11. ^ *Wadler P (2015). "Propositions As Types." Commun ACM.* **58***(12):75–84.*
12. ^a, ^b, ^c *Geuvers H, Nederpelt R (2014). Type Theory and Formal Proof. Cambridge University Press.*
13. ^ *Martin-Löf P (1982). "Constructive Mathematics and Computer Programming." North-Holland.*
14. ^ *Martin-Löf P (1984). Intuitionistic Type Theory. Bibliopolis.*
15. ^ *Dybjer P, Palmgren E (2024). "Intuitionistic Type Theory." Metaphysics Research Lab, Stanford University.*
16. ^ *Troelstra AS (2011). "History of Constructivism in the 20th Century." Cambridge University Press.*
17. ^a, ^b, ^c, ^d *Griffin TG (1990). "A Formulae-as-Type Notion of Control."*
18. ^a, ^b *Murthy C (1991). "An Evaluation Semantics for Classical Proofs." IEEE Press.*

19. a, b, cParigot M (1992). "$\lambda\mu$-Calculus: An Algorithmic Interpretation of Classical Natural Deduction."

20. a, bKrivine J-L (2009). "Realizability in Classical Logic." Panor Synth. **27**.

21. a, bvan Bakel S (2023). "Adding Negation to Lambda Mu." Log Methods Comput Sci. **19**(2):12:1–12:41.

22. a, bMiquel A (2011). "Existential Witness Extraction in Classical Realizability and Via a Negative Translation." Log Methods Comput Sci. **7**(2):1–57. Available from: https://lmcs.episciences.org/1068.

23. ^Gödel K (1986). Über eine bisher noch nicht benutzte Erweiterung des finiten Standpunktes [On a Hitherto Unused Extension of the Finitary Standpoint]. Oxford University Press. Originally written in 1933.

24. a, bGentzen G (1969). "New Version of the Consistency Proof for Elementary Number Theory." In M. E. Szabo (ed.), The Collected Papers of Gerhard Gentzen:252–286.

25. ^Friedman H (1978). "Classically and Intuitionistically Provably Recursive Functions." Lect Notes Math. **689**:21–27.

26. ^Hilbert D, Ackermann W (1950). Principles of Mathematical Logic. AMS Chelsea.

27. ^Prawitz D (2006). Natural Deduction: A Proof-Theoretical Study. Dover.

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.