

HOL in HOL over a set-theoretic universe: consistency of NK with infinity and choice

Christoph Benzmüller

Daniel Kirchner

October 3, 2026

Abstract

This companion to the entry *A deep embedding of HOL in HOL* (HOL_in_HOL_Deep) [2] certifies the natural-deduction calculus NK of that entry consistent with a genuine Dedekind-style *axiom* of infinity—an injective, non-surjective self-map of the individuals—and, because its standard model has full function spaces, with the relational axiom of *choice* as well. Such an axiom has *no* finite models, so the compactness argument of the main entry—which lifts arbitrarily large *finite* models to an infinity *scheme*—does not apply: a genuinely *infinite* model is required. We construct one over Paulson’s ZFC_in_HOL [8]: the individuals are interpreted as the finite ordinals ω , with full function spaces at every type, yielding a *standard* Σ -model of NK over the set-theoretic universe V . As this meta-theory is stronger and non-conservative, the guarantee is *relative* consistency at ZF strength; the machine-checked formalisation lives in Isabelle/HOL with Paulson’s ZFC_in_HOL (the ZF reading is metatheoretic, made precise in the introduction). Because NK already provides extensionality and typed description (after Andrews), this certifies the consistency of *full* classical higher-order logic in Church’s sense, choice included as a separate relational scheme.

The infinity axiom and the weaker infinity *scheme* of the main entry turn out to be logically independent. That the scheme does not entail the axiom is shown in the main entry; the present entry settles the other direction, since the axiom does not entail any of the scheme’s individual inequalities: its set-theoretic model interprets all individual constants alike, so those inequalities fail while the axiom holds. Neither sentence, then, implies the other.

The plain-HOL results—soundness, Henkin completeness, and finite-model consistency, including the compactness route to the infinity scheme—are proved in the main entry [2] and imported here *unchanged*. What the present entry *adds* is the set-theoretic route to axioms that plain HOL alone cannot model.

This contribution is part of the LogiKEy project (<https://logikey.org>).

Use of generative AI. Parts of this formalisation and its documentation were drafted with a generative AI assistant (Anthropic’s Claude) under continuous human direction. All proofs are machine-checked by Isabelle; the entire development, including all comments and references, was reviewed, revised and simplified by the authors, who take full responsibility for its content.

Contents

1	Introduction	3
2	Full classical HOL, abstractly: the choice scheme, the axioms’ satisfaction, and <i>hol-universe</i>	5

3	Consistency of full classical HOL: infinity and choice	5
4	Minimising the meta-theory: a weaker universe suffices	9
5	Minimising further: a Harrison-style universe axiom	10
6	The set-theoretic instantiation: V as a <i>hol-universe</i>	15

1 Introduction

Two entries, and where each result is proved. This is the second of two Archive of Formal Proofs entries on a deep embedding of classical higher-order logic—Church’s simple theory of types, following Benzmüller, Brown and Kohlhase [1]—inside Isabelle/HOL. The first entry, *A deep embedding of HOL in HOL* (HOL_in_HOL_Deep) [2], stays *entirely within plain Isabelle/HOL, with no set-theoretic universe*. There the natural-deduction calculus NK is proved sound, Henkin-complete and consistent; consistency comes from a concrete finite standard model, and, by compactness over arbitrarily large *finite* models, NK is shown consistent even with an infinity *scheme* $\{c_i \neq c_j \mid i \neq j\}$ —still without ever constructing an infinite model. The present entry adds one result that plain HOL cannot reach: the consistency of NK with a genuine Dedekind-style *axiom* of infinity, and with the axiom of *choice*. It imports the first entry unchanged and supplies the set-theoretic model that this result needs.

We are careful throughout—here, in the theory comments, and in both abstracts—to say which of two tiers a statement belongs to: what holds in plain HOL, and what needs the set-theoretic universe. Within this entry the split is visible in the theory structure itself. The set-theory-free half, theory HOL_Universe, imports only the first entry and never ZFC_in_HOL. It contains the choice scheme with its satisfaction condition (the infinity axiom and its satisfaction are inherited from the first entry), the abstract relative-consistency theorem, the completeness results, and two universe interfaces: the locale `hol_universe`, and the (at most as strong) Harrison-style axiom `harrison_universe`, after Harrison’s self-verification of HOL Light [4], from which `hol_universe` is in turn *constructed*. The set-theoretic universe itself appears only in the second theory, which instantiates both interfaces with Paulson’s set-theoretic universe V (the type of all sets of ZFC_in_HOL, introduced below).

Why the meta-theory must be strengthened. A single Dedekind-style axiom of infinity has *no* finite models, by the pigeonhole principle. So the route of the first entry—compactness over finite models—yields nothing here, and a genuinely *infinite* model is required. We do not try to build one inside plain HOL, for two reasons. First, no type that plain HOL can construct is large enough to carry a universe of full function spaces: by Cantor’s theorem each such type is outgrown by the tower of function spaces built over it, and that no other plain-HOL route succeeds follows from Gödel’s second theorem (discussed below). Second, the term-model construction of the completeness proof would be circular for a single sentence. Instead we borrow a ready-made, stronger meta-theory: Paulson’s ZFC_in_HOL [8], which axiomatises a Zermelo–Fraenkel universe as a type V . This is *not* a conservative extension of HOL— V is postulated by axioms, and ZF is proof-theoretically stronger than plain HOL—so what we obtain is a *relative* consistency result: NK with infinity is consistent *relative to* ZF. (The formalisation rests on the full ZFC_in_HOL, whose set-theoretic choice axiom we never invoke essentially; that the construction in fact needs only ZF strength—indeed less—is made precise in *How little of ZF is used* below. This is why we write ZF, not ZFC.) This pins down exactly where set theory is used, namely only to certify consistency (an axiom without finite models has no finite consistency witness). The model that realises the consistency is, by contrast, a *countable* Henkin model built in plain HOL, by the (ZFC-free) model-existence half of the first entry. For that single axiom the stronger meta-theory supplies the consistency premise and nothing else.

Choice, and full HOL. The same standard model settles more. Because its function spaces are *full*, it validates every instance of the relational axiom of *choice*: the required choice function is simply the set-theoretic λ -abstraction of a meta-level Hilbert choice. Hence NK with the

Dedekind axiom of infinity *and* choice is consistent relative to ZF. Since NK already carries extensionality and typed description, this is the consistency of full classical higher-order logic in the sense of Church. To be precise, by *full classical HOL in the sense of Church* we mean exactly this system: Church’s monomorphic simple theory of types [3]—the calculus NK with Boolean and functional extensionality and typed description, extended by the Dedekind axiom of infinity and the relational axiom of choice. It is *not* the polymorphic Isabelle/HOL of the meta-level, which goes beyond Church’s system with type variables, type definitions and overloading.

Two tiers of consistency, and Gödel’s second theorem. Write *O-HOL* for the embedded object logic (whose calculus is NK) and *M-HOL* for the meta-logic Isabelle/HOL in which it lives; the two must be kept apart. The first entry proves $M-HOL \vdash \text{con}(O-HOL)$: the bare object logic is consistent, and this is shown in plain HOL with *no* set universe. (With a single individual every domain is finite, so M-HOL exhibits a model directly; and O-HOL, having finite models, lacks the arithmetic strength that Gödel’s second theorem presupposes.) The present entry adds the *full* logic: $M-HOL + V \vdash \text{con}(O-HOL + \text{infinity} + \text{choice})$, that is, M-HOL extended by the set universe V proves full classical HOL consistent. This is as close as one can come to a self-standing $\vdash \text{con}(HOL)$, and it is emphatically *not* the object logic proving its own consistency.

That the meta-logic here cannot be plain M-HOL either is a *metatheoretic* observation, outside the machine-checked development. With infinity and choice, O-HOL is itself Church’s simple type theory with infinity and choice, and this interprets the M-HOL core (its individuals becoming an infinite type, its function types the function types). So the two share their proof-theoretic strength: by the classical analysis both are equiconsistent with a Zermelo-style set theory [7]. (Here “M-HOL” means the HOL *core*, whose definitional and overloading apparatus is separately known consistent [6].) Were plain M-HOL to prove $\text{con}(O-HOL + \text{infinity} + \text{choice})$, it would thereby prove its own consistency, which Gödel’s second incompleteness theorem forbids; so the step up to the set universe V is *necessary*, not a convenience. The same obstacle, and the same step to a stronger meta-theory, appears in Harrison’s self-verification of HOL Light [4] and its successors [5]. Throughout it is M-HOL reasoning *about* O-HOL, one level up—a difference of role, not of logical power: the object logic is in no way the weaker system. The certificate is tied to exactly the axioms modelled here. Extending O-HOL with a further axiom stays legitimate, but the machine-checked guarantee does not extend automatically: one must exhibit a model of the new axiom, and if the extension reaches the consistency strength of $M-HOL + V$, certifying it needs a yet stronger meta-theory—again by Gödel’s second theorem.

How little of ZF is used. A closer look pins down the exact surplus over plain HOL. The object-side argument is carrier-generic and uses no set theory at all: full HOL is consistent as soon as *some* standard Σ -frame has Dedekind-infinite individuals (an injective, non-surjective self-map). Plain HOL has infinity already; what it lacks is a single type gathering the full function spaces over an infinite base (a model formalised in HOL keeps all its domains inside one carrier type)—by Cantor these outgrow every type it can construct—and that is the whole reason for a stronger universe. Choice is *not* part of this surplus: the same full model validates it for free, so the price of full HOL over plain HOL is the axiom of infinity alone. Nor need that universe be full ZF. It is enough to assume full function spaces, booleans, extensionality, infinitely many individuals and a domain-respecting parameter interpretation—no replacement and no ordinals—and from these the logical constants (negation, disjunction, quantification, equality and description) are *constructed* rather than assumed. Every domain then has bounded rank, and this is machine-checked: the domain of individuals sits at rank ω , and each function-space layer adds only finitely

much, so $\text{rank}(D_\sigma) < \omega + \omega$ for every type σ , and every domain is an element of the set $V_{\omega \cdot 2}$. Two things should be kept apart here. What is *machine-checked* is precisely this rank bound together with Isabelle/HOL + ZFC_in_HOL $\vdash \text{con}(\text{O-HOL} + \text{infinity} + \text{choice})$. That the same consistency follows from far less than ZF is a *metatheoretic* reading of that proof, read off from the ranks of the objects it constructs: a single derivation mentions only finitely many types, so its refutation is validated using only sets of bounded rank—comfortably within Zermelo set theory, and without ever forming $V_{\omega \cdot 2}$ as a completed set (which would already call on a trace of replacement). This reading is not itself a formalised relative-consistency theorem. Under it the relative consistency factors as $\text{con}(\text{ZF}) \Rightarrow \text{con}(\text{weak universe}) \Rightarrow \text{con}(\text{full HOL})$; the first implication also absorbs the choice that the meta-theory uses throughout—by Gödel’s constructible universe L at ZF strength, while at Zermelo strength the corresponding elimination of choice without replacement is genuinely delicate (see Mathias [7]). The surplus strength of ZFC_in_HOL—which *exceeds* ZF, as it proves $\text{con}(\text{ZFC})$ —is never used.

2 Full classical HOL, abstractly: the choice scheme, the axioms’ satisfaction, and *hol-universe*

This theory is the *set-theory-free* half of the companion development: it imports only the main entry HOL_in_HOL_Deep and never mentions a set-theoretic universe. The Dedekind axiom of infinity *DInf*, its satisfaction conditions and its separation from the infinity scheme are inherited from the main entry (theory *NK-Infinity*); here the relational choice scheme *ACrel* is added as an object formula with its satisfaction condition in an *arbitrary* Σ -standard model, the abstract relative-consistency theorem is derived — full classical *HOL* is consistent as soon as *some* standard model has Dedekind-infinite individuals — and the universe interfaces are isolated from which the required standard model is *constructed*: the locale *hol-universe*, and beneath it an at most as strong Harrison-style locale *harrison-universe*. Completeness for the extension by *DInf* is likewise established here, in plain HOL. That the assumptions of these locales are satisfiable is a separate question — no *plain-HOL* type answers it: Cantor’s tower outgrows every type plain HOL can construct, and Gödel’s second theorem bars every other plain-HOL route (see the document introduction) — and is deferred to this entry’s final theory, *HOL-in-HOL-Deep-Infinity-ZFC*, which instantiates both locales with Paulson’s axiomatic set-theoretic universe V . The split makes the dependency structure machine-visible: nothing in this theory rests on *ZFC-in-HOL*.

3 Consistency of full classical HOL: infinity and choice

NK already carries extensionality (property *f*) and typed description. Adding the Dedekind axiom of infinity *DInf* and the axiom of *choice* makes the object logic full classical higher-order logic in the sense of Church. The consistency of the whole package is reduced, in one carrier-generic argument, to the existence of a standard frame with Dedekind-infinite individuals.

3.1 The axiom of choice as a relational scheme

Choice is stated, without any new primitive, as the relational scheme $AC_{\sigma, \tau} = \Pi R. (\Pi X. \exists Y. R \cdot X \cdot Y) \supset (\exists F. \Pi X. R \cdot X \cdot (F \cdot X))$ at every pair of types.

definition rR where $rR = (5::nat)$

definition rX where $rX = (6::nat)$

definition rY where $rY = (7::nat)$

definition rF where $rF = (\delta :: nat)$

definition $ACatom1 :: ty \Rightarrow ty \Rightarrow 'p \text{ tm}$ where

$ACatom1 \sigma \tau = (((rR^f \sigma \Rightarrow \tau \Rightarrow o) \cdot (rX^f \sigma)) \cdot (rY^f \tau))$

definition $ACatom2 :: ty \Rightarrow ty \Rightarrow 'p \text{ tm}$ where

$ACatom2 \sigma \tau = (((rR^f \sigma \Rightarrow \tau \Rightarrow o) \cdot (rX^f \sigma)) \cdot ((rF^f \sigma \Rightarrow \tau) \cdot (rX^f \sigma)))$

definition $ACprem :: ty \Rightarrow ty \Rightarrow 'p \text{ tm}$ where

$ACprem \sigma \tau = \prod rX_\sigma. \exists rY_\tau. ACatom1 \sigma \tau$

definition $ACconcl :: ty \Rightarrow ty \Rightarrow 'p \text{ tm}$ where

$ACconcl \sigma \tau = \exists rF_\sigma \Rightarrow \tau. \prod rX_\sigma. ACatom2 \sigma \tau$

definition $ACrel :: ty \Rightarrow ty \Rightarrow 'p \text{ tm}$ where

$ACrel \sigma \tau = \prod rR_\sigma \Rightarrow \tau \Rightarrow o. (ACprem \sigma \tau \supset ACconcl \sigma \tau)$

The scheme in full, as for $DInf$: assembled by definitional unfolding, and in its locally-nameless normal form:

lemma $ACrel$ -unfolded:

$(ACrel \sigma \tau :: 'p \text{ tm})$

$= (\prod rR_\sigma \Rightarrow \tau \Rightarrow o.$

$((\prod rX_\sigma. \exists rY_\tau. (((rR^f \sigma \Rightarrow \tau \Rightarrow o) \cdot (rX^f \sigma)) \cdot (rY^f \tau))))$

$\supset (\exists rF_\sigma \Rightarrow \tau. \prod rX_\sigma.$

$((((rR^f \sigma \Rightarrow \tau \Rightarrow o) \cdot (rX^f \sigma)) \cdot ((rF^f \sigma \Rightarrow \tau) \cdot (rX^f \sigma))))))$

by (simp only: $ACrel$ -def $ACprem$ -def $ACconcl$ -def $ACatom1$ -def $ACatom2$ -def)

lemma $ACrel$ -norm:

$(ACrel \sigma \tau :: 'p \text{ tm})$

$= \prod \sigma \Rightarrow \tau \Rightarrow o$

$((\prod \sigma (\exists \tau ((Bnd (Suc (Suc 0)) \cdot Bnd (Suc 0)) \cdot Bnd 0)))$

$\supset (\exists \sigma \Rightarrow \tau (\prod \sigma ((Bnd (Suc (Suc 0)) \cdot Bnd 0) \cdot (Bnd (Suc 0) \cdot Bnd 0))))$

by (simp add: $ACrel$ -def $ACprem$ -def $ACconcl$ -def $ACatom1$ -def $ACatom2$ -def

ExN -def $AllN$ -def $Forall$ -def $ImpB$ -def $rdefs$)

3.2 Satisfaction of the choice scheme in an arbitrary standard model

Every instance of the scheme holds in *any* Σ -standard model: the full function space supplies the choice function as the λ -abstraction of a meta-level Hilbert choice. (The corresponding satisfaction analysis for $DInf$ — $DInf$ -sat and its relatives — lives in the main entry's *NK-Infinity* and is used below unchanged.)

context *standard-model*

begin

lemma $ACrel$ -sat:

assumes xi : $bkkA.asg \xi$

shows $den (ACrel \sigma \tau) \xi = Tv$

proof —

have $body$: $den (ACprem \sigma \tau \supset ACconcl \sigma \tau) (\xi(rR_\sigma \Rightarrow \tau \Rightarrow o := d)) = Tv$

if d : $Dm (\sigma \Rightarrow \tau \Rightarrow o) d$ **for** d

proof —

define η where $\eta = \xi(rR_\sigma \Rightarrow \tau \Rightarrow o := d)$

have ea : $bkkA.asg \eta$ **unfolding** η -def **using** $xi d$ **by** (rule $bkkA.asg-upd$)

— meaning of the premise

have $prem$: $(den (ACprem \sigma \tau) \eta = Tv)$

$= (\forall a. Dm \sigma a \longrightarrow (\exists b. Dm \tau b \wedge (d @ a) @ b = Tv))$

proof –
have *inner*: $(\text{den } (\text{ACatom1 } \sigma \tau) ((\eta(rX_\sigma := a))(rY_\tau := b)) = Tv) = ((d @ a) @ b = Tv)$
if $a: Dm \sigma a$ **and** $b: Dm \tau b$ **for** $a b$
proof –
define ϱ **where** $\varrho = (\eta(rX_\sigma := a))(rY_\tau := b)$
have $\text{den } (\text{ACatom1 } \sigma \tau) \varrho = (d @ a) @ b$
unfolding *ACatom1-def* **by** (*simp add: den.simps(8) \varrho-def \eta-def upd-def rdefs*)
thus *?thesis* **by** (*simp add: \varrho-def*)
qed
have *step*: $(\text{den } (\exists rY_\tau. \text{ACatom1 } \sigma \tau) (\eta(rX_\sigma := a)) = Tv)$
 $= (\exists b. Dm \tau b \wedge (d @ a) @ b = Tv)$ **if** $a: Dm \sigma a$ **for** a
proof –
have $(\text{den } (\exists rY_\tau. \text{ACatom1 } \sigma \tau) (\eta(rX_\sigma := a)) = Tv)$
 $= (\exists b. Dm \tau b \wedge \text{den } (\text{ACatom1 } \sigma \tau) ((\eta(rX_\sigma := a))(rY_\tau := b)) = Tv)$
by (*rule sat-ExN[OF wff-ACatom1 bkkA.asg-upd[OF ea a]]*)
thus *?thesis* **using** *inner[OF a]* **by** (*simp cong: conj-cong*)
qed
have $(\text{den } (\text{ACprem } \sigma \tau) \eta = Tv)$
 $= (\forall a. Dm \sigma a \longrightarrow \text{den } (\exists rY_\tau. \text{ACatom1 } \sigma \tau) (\eta(rX_\sigma := a)) = Tv)$
unfolding *ACprem-def* **by** (*rule sat-AllN[OF wff-ExN[OF wff-ACatom1] ea]*)
thus *?thesis* **using** *step* **by** *simp*
qed
— the conclusion, with the choice function $Lm \sigma (\lambda x. \text{SOME } y. \dots)$ as witness; it lands in the full function space by $(\bigwedge d. \mathcal{D}_{? \sigma} d \implies \mathcal{D}_{? \tau} (?h d)) \implies \mathcal{D}_{? \sigma} \Rightarrow_{? \tau} (Lm ?\sigma ?h)$
have *concl*: $\text{den } (\text{ACconcl } \sigma \tau) \eta = Tv$
if $P: \forall a. Dm \sigma a \longrightarrow (\exists b. Dm \tau b \wedge (d @ a) @ b = Tv)$
proof –
define g **where** $g = Lm \sigma (\lambda x. \text{SOME } y. Dm \tau y \wedge (d @ x) @ y = Tv)$
have *resp*: $Dm \tau (\text{SOME } y. Dm \tau y \wedge (d @ x) @ y = Tv)$ **if** $x: Dm \sigma x$ **for** x
proof –
from $P x$ **have** $\exists y. Dm \tau y \wedge (d @ x) @ y = Tv$ **by** *blast*
from *someI-ex[OF this]* **show** *?thesis* **by** *simp*
qed
have *gdom*: $Dm (\sigma \Rightarrow \tau) g$
unfolding *g-def* **by** (*rule Lm-dom*) (*rule resp*)
have *gapp*: $(d @ a) @ (g @ a) = Tv$ **if** $a: Dm \sigma a$ **for** a
proof –
have $e: g @ a = (\text{SOME } y. Dm \tau y \wedge (d @ a) @ y = Tv)$
unfolding *g-def* **by** (*rule beta-Lm[OF resp a]*)
from $P a$ **have** $ex: \exists y. Dm \tau y \wedge (d @ a) @ y = Tv$ **by** *blast*
from *someI-ex[OF ex]* **show** *?thesis* **by** (*simp add: e*)
qed
have *inner2*: $(\text{den } (\text{ACatom2 } \sigma \tau) ((\eta(rF_\sigma \Rightarrow \tau := g))(rX_\sigma := a)) = Tv)$
 $= ((d @ a) @ (g @ a) = Tv)$ **if** $a: Dm \sigma a$ **for** a
proof –
define ϱ **where** $\varrho = (\eta(rF_\sigma \Rightarrow \tau := g))(rX_\sigma := a)$
have $\text{den } (\text{ACatom2 } \sigma \tau) \varrho = (d @ a) @ (g @ a)$
unfolding *ACatom2-def* **by** (*simp add: den.simps(8) \varrho-def \eta-def upd-def rdefs*)
thus *?thesis* **by** (*simp add: \varrho-def*)
qed
have *step2*: $(\text{den } (\Pi rX_\sigma. \text{ACatom2 } \sigma \tau) (\eta(rF_\sigma \Rightarrow \tau := g)) = Tv)$
 $= (\forall a. Dm \sigma a \longrightarrow (d @ a) @ (g @ a) = Tv)$
proof –
have $(\text{den } (\Pi rX_\sigma. \text{ACatom2 } \sigma \tau) (\eta(rF_\sigma \Rightarrow \tau := g)) = Tv)$

```

    = (∀ a. Dm σ a → den (ACatom2 σ τ) ((η(rFσ ⇒ τ := g))(rXσ := a)) = Tv)
  by (rule sat-AllN[OF wff-ACatom2 bkkA.asg-upd[OF ea gdom]])
  thus ?thesis using inner2 by simp
qed
have (den (ACconcl σ τ) η = Tv)
  = (∃ k. Dm (σ ⇒ τ) k ∧ den (ΠrXσ. ACatom2 σ τ) (η(rFσ ⇒ τ := k)) = Tv)
  unfolding ACconcl-def by (rule sat-ExN[OF wff-AllN[OF wff-ACatom2] ea])
  thus ?thesis using gdom step2 gapp by auto
qed
have imp: (den (ACprem σ τ ⊃ ACconcl σ τ) η = Tv)
  = ((den (ACprem σ τ) η = Tv) → (den (ACconcl σ τ) η = Tv))
  by (rule sat-ImpBB[OF wff-ACprem wff-ACconcl ea])
have den (ACprem σ τ ⊃ ACconcl σ τ) η = Tv
  unfolding imp using prem concl by blast
  thus ?thesis by (simp add: η-def)
qed
have (den (ACrel σ τ) ξ = Tv)
  = (∀ d. Dm (σ ⇒ τ ⇒ o) d
    → den (ACprem σ τ ⊃ ACconcl σ τ) (ξ(rRσ ⇒ τ ⇒ o := d)) = Tv)
  unfolding ACrel-def by (rule sat-AllN[OF wff-ImpB[OF wff-ACprem wff-ACconcl] xi])
  thus den (ACrel σ τ) ξ = Tv using body by simp
qed
end

```

The relative-consistency theorem, abstractly. Suppose *some* standard Σ -model has an injective but not surjective self-map of its individuals — that is, Dedekind-infinitely many individuals. Then full classical *HOL* is consistent. Consistency is the predicate *con* of the main entry, defined by $con \Phi \longleftrightarrow \neg(\Phi \vdash \perp)$: the conclusion below states that *NK* cannot derive falsity from *DInf* together with all instances of *ACrel*. The eigen-parameter starvation that makes *con* a weak reading for impure infinite contexts (main entry, *NK-Infinity*) cannot arise here: the axioms use no parameters at all, so the context is pure, and over an infinite signature $con \Phi$ and $\neg(\Phi \Vdash \perp)$ provably coincide (*fprov-eq-bprov*). The proof works for an *arbitrary* such model, whatever type its elements come from, and uses no set theory. The one thing plain *HOL* cannot supply is the premise — a model with infinitely many individuals; producing one is the central task of the set-theoretic theory that concludes this entry.¹

theorem *con-full-HOL-rel*:

fixes $Dm :: ty \Rightarrow 'u \Rightarrow bool$ **and** $Ap :: 'u \Rightarrow 'u \Rightarrow 'u$

¹Why not? Plain *HOL* has infinite *types*, and every single level nat , $nat \Rightarrow bool$, ... of the full tower exists as a type. What it lacks is *one carrier* for the whole tower at once. In BKK's sense "standard" means only that the function domains are *full*; but a model formalised inside plain *HOL* necessarily has one further feature: the object types are values of the datatype *ty*, so any formalised model — the locale *standard-model* included — interprets all domains inside a single carrier type *'u* (simple types cannot assign each object type its own meta-type). Over infinite individuals with full function spaces that one carrier must outgrow every level of the tower. No plain-*HOL* type reaches that far: every type plain *HOL* can construct is bounded in cardinality by some finite tower of function spaces over its one primitive infinite type *ind* (of which *nat* is the familiar copy). One might hope to escape with a *Henkin* model, whose function domains need not be full and which therefore fits into a small carrier. But a sparse function domain need not provide what the axiom asks for. *DInf* says: *there exists* an injective, non-surjective self-map of the individuals. In a Henkin model that map has to be an element of the (now sparse) function domain $D_\iota \Rightarrow \iota$ — and it may simply not be there, even when the individuals are infinite. This is not a hypothetical: the main entry's *henkin-scheme-refutes-DInf* constructs a Henkin model with infinitely many individuals in which *DInf* is nevertheless *false*. And the canonical way to build a Henkin model that does contain the witness — the term-model construction — takes the consistency of *DInf* as its *premise*, and so cannot be the way to prove it (that *every* other route fails as well is Gödel's second theorem, see the introduction).

```

and  $Lm :: ty \Rightarrow ('u \Rightarrow 'u) \Rightarrow 'u$  and  $Tv Fv Ngv Dsv :: 'u$ 
and  $Iv Ev Piv :: ty \Rightarrow 'u$  and  $Jv :: 'p \Rightarrow ty \Rightarrow 'u$ 
assumes  $M$ : standard-model  $Dm Ap Lm Tv Fv Ngv Dsv Iv Ev Piv Jv$ 
and  $inf :: \exists d. Dm (\iota \Rightarrow \iota) d$ 
 $\wedge (\forall a. Dm \iota a \longrightarrow (\forall b. Dm \iota b \longrightarrow (Ap d a = Ap d b \longrightarrow a = b)))$ 
 $\wedge (\exists c. Dm \iota c \wedge (\forall z. Dm \iota z \longrightarrow Ap d z \neq c))$ 
shows  $con (insert (DInf :: 'p tm) (range (\lambda(\sigma,\tau). ACrel \sigma \tau)))$ 
proof –
interpret standard-model  $Dm Ap Lm Tv Fv Ngv Dsv Iv Ev Piv Jv$  by (rule  $M$ )
— the constants inhabit their domains, so a domain-respecting assignment exists
define  $\xi :: nat \Rightarrow ty \Rightarrow 'u$  where  $\xi = (\lambda n \varrho. Jv \text{ undefined } \varrho)$ 
have  $xi$ :  $bkkA.asg \xi$  by (simp add:  $\xi$ -def bkkA.asg-def Jv-dom)
from  $inf$  obtain  $d$  where  $d: Dm (\iota \Rightarrow \iota) d$ 
and  $dinj$ :  $\forall a. Dm \iota a \longrightarrow (\forall b. Dm \iota b \longrightarrow (Ap d a = Ap d b \longrightarrow a = b))$ 
and  $dns$ :  $\exists c. Dm \iota c \wedge (\forall z. Dm \iota z \longrightarrow Ap d z \neq c)$  by blast
have  $satD$ :  $den (DInf :: 'p tm) \xi = Tv$  by (rule  $DInf\text{-sat}[OF \ xi \ d \ dinj \ dns]$ )
have  $satAC$ :  $den (ACrel \sigma \tau :: 'p tm) \xi = Tv$  for  $\sigma \tau$  by (rule  $ACrel\text{-sat}[OF \ xi]$ )
show ?thesis
by (rule  $model\text{-con}[OF \ xi]$ ) (use  $wff\text{-}DInf \ wff\text{-}ACrel \ satD \ satAC$  in auto)
qed

```

Completeness carries over to the axiomatic extension, in its strongest form: $\{DInf\}$ is a finite context, so *completeness-hyps-open-finite* applies — free variables are allowed in the conclusion, the carrier and the signature are any infinite types, and purity is automatic. Any formula true in every infinite-carrier model of $DInf$ is thus already derivable from $DInf$ in NK , and with soundness the two coincide — established by the plain-HOL completeness of the main entry, with no set universe.

theorem *NK-completeness-DInf*:

```

fixes  $A :: 'p::infinite \ tm$ 
assumes  $wff_o(A)$  and  $\{DInf\} \models ('u::infinite) \ A$ 
shows  $\{DInf\} \vdash A$ 
proof (rule  $completeness\text{-hyps-open-finite}[OF \ - \ assms(1) \ assms(2)]$ )
show  $finite \ \{DInf :: 'p \ tm\}$  by simp
next
fix  $B :: 'p \ tm$  assume  $B \in \{DInf\}$ 
thus  $wff_o(B)$  by (simp add: wff-DInf)
qed

```

theorem *sound-and-complete-DInf*:

```

fixes  $A :: 'p::infinite \ tm$ 
assumes  $wff_o(A)$ 
shows  $\{DInf\} \vdash A \longleftrightarrow \{DInf\} \models ('u::infinite) \ A$ 
using  $assms \ NK\text{-completeness-DInf} \ soundness\text{-sat}$  by blast

```

4 Minimising the meta-theory: a weaker universe suffices

We weaken the premise of *con-full-HOL-rel* once more. Instead of *assuming* the logical constants (as *standard-model* does), the locale *hol-universe* adds Dedekind-infinite individuals to the constants-construction interface *lambda-universe* of the main entry: a universe with full function spaces, booleans, extensionality and a domain-respecting parameter interpretation, over which the constants are *constructed*. Any model of these weaker assumptions — the final theory provides one over V — thus yields the relative consistency of full *HOL*.

```

locale hol-universe = lambda-universe Dm Ap Lm Tv Fv Jv
  for Dm :: ty  $\Rightarrow$  'u  $\Rightarrow$  bool and Ap :: 'u  $\Rightarrow$  'u  $\Rightarrow$  'u
    and Lm :: ty  $\Rightarrow$  ('u  $\Rightarrow$  'u)  $\Rightarrow$  'u and Tv Fv :: 'u and Jv :: 'p  $\Rightarrow$  ty  $\Rightarrow$  'u +
  assumes ind-inf:  $\exists d. Dm \iota \Rightarrow \iota$  d
     $\wedge (\forall a. Dm \iota a \longrightarrow (\forall b. Dm \iota b \longrightarrow (Ap d a = Ap d b \longrightarrow a = b)))$ 
     $\wedge (\exists c. Dm \iota c \wedge (\forall z. Dm \iota z \longrightarrow Ap d z \neq c))$ 

begin

theorem con-full-HOL-universe:
  con (insert (DInf :: 'p tm) (range ( $\lambda(\sigma, \tau). ACrel \sigma \tau$ )))
  by (rule con-full-HOL-rel[OF is-standard-model ind-inf])

end

```

5 Minimising further: a Harrison-style universe axiom

Harrison's self-verification of HOL Light [4] proves the soundness of HOL-with-infinity in a meta-logic strengthened by a single *universe axiom*: an infinite type closed under the set-forming operations that full function spaces require. The locale *harrison-universe* is our rendering of that axiom, one level below *hol-universe*: it mentions neither types, domains, application nor abstraction. Fixed are an injective pairing *pr*, an injective coding *enc* of the *small* collections *Sm*, and a small Dedekind-infinite collection *D0* of individuals; smallness is closed downwards, under products and — the strong-limit core of Harrison's axiom — under *coded power collections* (*Sm-pow*).² From these data alone the whole *hol-universe* structure is *constructed*: domains as collections of coded functional graphs, abstraction by coding, application by decoding.

```

locale harrison-universe =
  fixes pr :: 'u  $\Rightarrow$  'u  $\Rightarrow$  'u and enc :: ('u  $\Rightarrow$  bool)  $\Rightarrow$  'u
    and Sm :: ('u  $\Rightarrow$  bool)  $\Rightarrow$  bool and D0 :: 'u  $\Rightarrow$  bool
  assumes pr-inj: pr a b = pr c d  $\implies$  a = c  $\wedge$  b = d
    and enc-inj:  $\llbracket Sm X; Sm Y; enc X = enc Y \rrbracket \implies X = Y$ 
    and Sm-sub:  $\llbracket Sm X; Y \leq X \rrbracket \implies Sm Y$ 
    and Sm-prod:  $\llbracket Sm X; Sm Y \rrbracket \implies Sm (\lambda z. \exists a b. X a \wedge Y b \wedge z = pr a b)$ 
    and Sm-pow: Sm X  $\implies Sm (\lambda z. \exists Y. Y \leq X \wedge Sm Y \wedge z = enc Y)$ 
    and Sm-D0: Sm D0
    and D0-inf:  $\exists f c. (\forall a. D0 a \longrightarrow D0 (f a))$ 
     $\wedge (\forall a b. D0 a \longrightarrow D0 b \longrightarrow f a = f b \longrightarrow a = b)$ 
     $\wedge D0 c \wedge (\forall a. D0 a \longrightarrow f a \neq c)$ 

begin

```

Smallness of the empty collection need not be assumed: it is a sub-collection of the small *D0*.

```

lemma Sm-empty: Sm ( $\lambda-. False$ )
  by (rule Sm-sub[OF Sm-D0]) (simp add: le-fun-def)

```

5.1 Truth values, domains, abstraction, application

The truth values are the first stages of the cumulative hierarchy in miniature: *HTv* codes the empty collection, *HFv* its singleton. Both collections are small — the empty one by *Sm-empty*,

²In effect this axiom postulates a fresh base type *v* that gathers the domains of the whole type hierarchy over ι and $\mathbf{o}$, so that every finite-type domain embeds into it. The subtlety is that *v* is closed only under the operations those finite types require — pairing and *small* (coded) power collections — and *not* under its own full function space $v \Rightarrow \mathbf{o}$, which by Cantor would be strictly larger. This restricted closure, rather than a full powerset, is exactly what keeps the axiom consistent and of merely strong-limit strength.

its singleton as the coded power collection of the empty one (*Sm-pow*) — and *enc-inj* keeps the two codes apart. A function domain consists of the codes of the functional graphs over the argument domain; application decodes, abstraction codes.

definition $HTv :: 'u \text{ where } HTv = \text{enc } (\lambda-. \text{False})$

definition $HFv :: 'u \text{ where } HFv = \text{enc } (\lambda z. z = HTv)$

primrec $HD :: ty \Rightarrow 'u \Rightarrow \text{bool} \text{ where}$

$HD \iota = D0$

| $HD \circ = (\lambda x. x = HTv \vee x = HFv)$

| $HD (\sigma \Rightarrow \tau) = (\lambda x. \exists g. (\forall a. HD \sigma a \longrightarrow HD \tau (g a)) \wedge x = \text{enc } (\lambda z. \exists a. HD \sigma a \wedge z = \text{pr } a (g a)))$

definition $Hgr :: ty \Rightarrow ('u \Rightarrow 'u) \Rightarrow 'u \Rightarrow \text{bool} \text{ where}$

$Hgr \sigma g = (\lambda z. \exists a. HD \sigma a \wedge z = \text{pr } a (g a))$

definition $HLm :: ty \Rightarrow ('u \Rightarrow 'u) \Rightarrow 'u \text{ where } HLm \sigma g = \text{enc } (Hgr \sigma g)$

The *THE* below is Isabelle's *definite description* at the meta-level — unique choice, not a choice principle; decoding is well-defined by *enc-inj*. (The *SOME* in *HJv* is meta-level Hilbert choice of Isabelle/HOL. Neither adds an axiom to the embedded object logic.)

definition $Hdec :: 'u \Rightarrow 'u \Rightarrow \text{bool} \text{ where } Hdec x = (\text{THE } X. \text{Sm } X \wedge \text{enc } X = x)$

definition $HAp :: 'u \Rightarrow 'u \Rightarrow 'u \text{ where } HAp f a = (\text{THE } b. Hdec f (\text{pr } a b))$

definition $HJv :: 'p \Rightarrow ty \Rightarrow 'u \text{ where } HJv p \sigma = (\text{SOME } d. HD \sigma d)$

lemma *HD-Fun*: $HD (\sigma \Rightarrow \tau) x \longleftrightarrow (\exists g. (\forall a. HD \sigma a \longrightarrow HD \tau (g a)) \wedge x = HLm \sigma g)$
by (*simp add: HLm-def Hgr-def*)

5.2 Every domain is small and inhabited

lemma *Sm-singleton-HTv*: $\text{Sm } (\lambda z. z = HTv)$

proof —

have $(\lambda z. \exists Y. Y \leq (\lambda-. \text{False}) \wedge \text{Sm } Y \wedge z = \text{enc } Y) = (\lambda z. z = HTv)$

proof (*rule ext*)

fix z **show** $(\exists Y. Y \leq (\lambda-. \text{False}) \wedge \text{Sm } Y \wedge z = \text{enc } Y) = (z = HTv)$

proof

assume $\exists Y. Y \leq (\lambda-. \text{False}) \wedge \text{Sm } Y \wedge z = \text{enc } Y$

then obtain Y **where** $Y \leq (\lambda-. \text{False})$ **and** $z = \text{enc } Y$ **by** *blast*

from *this(1)* **have** $Y = (\lambda-. \text{False})$ **by** (*auto simp: le-fun-def fun-eq-iff*)

with z **show** $z = HTv$ **by** (*simp add: HTv-def*)

next

assume $z = HTv$

hence $(\lambda-. \text{False}) \leq (\lambda-. \text{False}) \wedge \text{Sm } (\lambda-. \text{False}) \wedge z = \text{enc } (\lambda-. \text{False})$

using *Sm-empty* **by** (*simp add: HTv-def*)

thus $\exists Y. Y \leq (\lambda-. \text{False}) \wedge \text{Sm } Y \wedge z = \text{enc } Y$ **by** *blast*

qed

qed

with *Sm-pow[OF Sm-empty]* **show** *?thesis* **by** *simp*

qed

lemma *HTv-neq-HFv*: $HTv \neq HFv$

proof

assume $HTv = HFv$

hence $(\lambda-. \text{False}) = (\lambda z. z = HTv)$

using *enc-inj[OF Sm-empty Sm-singleton-HTv]* *HTv-def HFv-def* **by** *metis*

```

from fun-cong[OF this, of HTv] show False by simp
qed

lemma Sm-bool: Sm (HD o)
proof –
  have sub:  $Y = (\lambda -. \text{False}) \vee Y = (\lambda z. z = \text{HTv})$  if le:  $Y \leq (\lambda z. z = \text{HTv})$  for Y
  proof –
    have imp:  $Y \ w \implies w = \text{HTv}$  for w using le-funD[OF le, of w] by simp
    show ?thesis
    proof (cases Y HTv)
      case True
        have  $Y = (\lambda z. z = \text{HTv})$  by (rule ext) (use imp True in blast)
        thus ?thesis ..
      next
        case False
        have  $Y = (\lambda -. \text{False})$  by (rule ext) (use imp False in blast)
        thus ?thesis ..
    qed
  qed
have  $(\lambda z. \exists Y. Y \leq (\lambda z. z = \text{HTv}) \wedge \text{Sm } Y \wedge z = \text{enc } Y) = \text{HD } o$ 
proof (rule ext)
  fix z show  $(\exists Y. Y \leq (\lambda z. z = \text{HTv}) \wedge \text{Sm } Y \wedge z = \text{enc } Y) = \text{HD } o \ z$ 
  proof
    assume  $\exists Y. Y \leq (\lambda z. z = \text{HTv}) \wedge \text{Sm } Y \wedge z = \text{enc } Y$ 
    then obtain Y where  $Y \leq (\lambda z. z = \text{HTv})$  and  $z = \text{enc } Y$  by blast
    from sub[OF this(1)] z show  $\text{HD } o \ z$  by (auto simp: HTv-def HFv-def)
  next
    assume  $\text{HD } o \ z$ 
    then consider  $z = \text{HTv} \mid z = \text{HFv}$  by auto
    thus  $\exists Y. Y \leq (\lambda z. z = \text{HTv}) \wedge \text{Sm } Y \wedge z = \text{enc } Y$ 
    proof cases
      case 1
        hence  $(\lambda -. \text{False}) \leq (\lambda z. z = \text{HTv}) \wedge \text{Sm } (\lambda -. \text{False}) \wedge z = \text{enc } (\lambda -. \text{False})$ 
        using Sm-empty by (simp add: le-fun-def HTv-def)
        thus ?thesis by blast
      next
        case 2
        hence  $(\lambda z. z = \text{HTv}) \leq (\lambda z. z = \text{HTv}) \wedge \text{Sm } (\lambda z. z = \text{HTv}) \wedge z = \text{enc } (\lambda z. z = \text{HTv})$ 
        using Sm-singleton-HTv by (simp add: HFv-def)
        thus ?thesis by blast
    qed
  qed
qed
with Sm-pow[OF Sm-singleton-HTv] show ?thesis by simp
qed

```

The function-space case is the Harrison core: every coded graph is a small subcollection of the product of the two domains, so the domain of codes sits inside the coded power collection of that product.

```

lemma Sm-HD: Sm (HD σ)
proof (induction σ)
  case Ind show ?case by (simp add: Sm-D0)
next
  case Bool show ?case by (rule Sm-bool)

```

next
case (*Fun* σ τ)
let $?P = \lambda z. \exists a b. HD \sigma a \wedge HD \tau b \wedge z = pr a b$
have $HD (\sigma \Rightarrow \tau) \leq (\lambda z. \exists Y. Y \leq ?P \wedge Sm Y \wedge z = enc Y)$
proof (*intro le-funI le-boolI*)
fix x **assume** $HD (\sigma \Rightarrow \tau) x$
hence $\exists g. (\forall a. HD \sigma a \longrightarrow HD \tau (g a)) \wedge x = HLM \sigma g$
by (*simp add: HLM-def Hgr-def*)
then obtain g **where** $g: \forall a. HD \sigma a \longrightarrow HD \tau (g a)$ **and** $x: x = HLM \sigma g$ **by** *blast*
have $le: Hgr \sigma g \leq ?P$ **using** g **by** (*auto simp: Hgr-def le-fun-def*)
have $Sm (Hgr \sigma g)$ **by** (*rule Sm-sub[OF Sm-prod[OF Fun.IH] le]*)
with le x **show** $\exists Y. Y \leq ?P \wedge Sm Y \wedge x = enc Y$ **by** (*auto simp: HLM-def*)
qed
from *Sm-sub[OF Sm-pow[OF Sm-prod[OF Fun.IH]] this]* **show** $?case$.
qed

lemma *HLM-dom*: **assumes** $\bigwedge d. HD \sigma d \Longrightarrow HD \tau (h d)$ **shows** $HD (\sigma \Rightarrow \tau) (HLM \sigma h)$
unfolding *HD-Fun* **using** *assms* **by** *blast*

lemma *HD-ne*: $\exists x. HD \sigma x$
proof (*induction* σ)
case *Ind* **show** $?case$ **using** *D0-inf* **by** *auto*
next
case *Bool* **show** $?case$ **by** *auto*
next
case (*Fun* σ τ)
from *Fun.IH*(2) **obtain** w **where** $w: HD \tau w$ **by** *blast*
have $HD (\sigma \Rightarrow \tau) (HLM \sigma (\lambda-. w))$ **by** (*rule HLM-dom*) (*rule* w)
thus $?case$ **by** *blast*
qed

5.3 Decoding is inverse to coding, and the frame conditions follow

lemma *Hdec-enc*: **assumes** $Sm X$ **shows** $Hdec (enc X) = X$
unfolding *Hdec-def* **by** (*rule the-equality*) (*use assms enc-inj in <blast+>*)

lemma *Hgr-pr*: **assumes** $HD \sigma a$ **shows** $Hgr \sigma h (pr a b) \longleftrightarrow b = h a$
using *assms* **by** (*auto simp: Hgr-def dest: pr-inj*)

lemma *Sm-Hgr*: **assumes** $\bigwedge d. HD \sigma d \Longrightarrow HD \tau (h d)$ **shows** $Sm (Hgr \sigma h)$
proof (*rule Sm-sub[OF Sm-prod[OF Sm-HD Sm-HD]]*)
show $Hgr \sigma h \leq (\lambda z. \exists a b. HD \sigma a \wedge HD \tau b \wedge z = pr a b)$
using *assms* **by** (*auto simp: Hgr-def le-fun-def*)
qed

lemma *HAp-HLM*: **assumes** $sm: Sm (Hgr \sigma h)$ **and** $a: HD \sigma a$ **shows** $HAp (HLM \sigma h) a = h a$
unfolding *HAp-def HLM-def Hdec-enc[OF sm]*
by (*rule the-equality*) (*simp-all add: Hgr-pr[OF a]*)

lemma *HD-FunE*:
assumes $f: HD (\sigma \Rightarrow \tau) f$
shows $\exists g. (\forall a. HD \sigma a \longrightarrow HD \tau (g a)) \wedge f = HLM \sigma g \wedge (\forall a. HD \sigma a \longrightarrow HAp f a = g a)$
proof –
from f **have** $\exists g. (\forall a. HD \sigma a \longrightarrow HD \tau (g a)) \wedge f = HLM \sigma g$

by (simp add: HLM-def Hgr-def)
 then obtain g where $g: \forall a. HD \sigma a \longrightarrow HD \tau (g a)$ and $fe: f = HLM \sigma g$ by blast
 have $sm: Sm (Hgr \sigma g)$ by (rule Sm-Hgr) (use g in blast)
 have $\forall a. HD \sigma a \longrightarrow HAp f a = g a$ unfolding fe using $HAp-HLM[OF sm]$ by blast
 with $g fe$ show ?thesis by blast
 qed

lemma *HAp-dom*: assumes $f: HD (\sigma \Rightarrow \tau) f$ and $a: HD \sigma a$ shows $HD \tau (HAp f a)$
 proof –
 from $HD-FunE[OF f]$ obtain g where $g: \forall a. HD \sigma a \longrightarrow HD \tau (g a)$
 and $app: \forall a. HD \sigma a \longrightarrow HAp f a = g a$ by blast
 from $g app a$ show ?thesis by auto
 qed

lemma *HAp-ext*:
 assumes $g: HD (\sigma \Rightarrow \tau) g$ and $k: HD (\sigma \Rightarrow \tau) k$
 and $e: \bigwedge a. HD \sigma a \Longrightarrow HAp g a = HAp k a$
 shows $g = k$
 proof –
 from $HD-FunE[OF g]$ obtain $g0$ where $g0: g = HLM \sigma g0$
 and $appg: \forall a. HD \sigma a \longrightarrow HAp g a = g0 a$ by blast
 from $HD-FunE[OF k]$ obtain $k0$ where $k0: k = HLM \sigma k0$
 and $appk: \forall a. HD \sigma a \longrightarrow HAp k a = k0 a$ by blast
 have $gk: g0 a = k0 a$ if $HD \sigma a$ for a
 using $appg appk e[OF that]$ that by simp
 have $Hgr \sigma g0 = Hgr \sigma k0$ by (rule ext) (auto simp: Hgr-def gk)
 thus ?thesis by (simp add: $g0 k0 HLM-def$)
 qed

lemma *HJv-dom*: $HD \sigma (HJv p \sigma)$
 unfolding *HJv-def* by (rule someI-ex[OF HD-ne])

Dedekind infinity transfers from the meta-level self-map f of $D0$ to the object level: the witness in $HD (\iota \Rightarrow \iota)$ is its coded graph $HLM \iota f$.

lemma *HD-inf*:
 $\exists d. HD (\iota \Rightarrow \iota) d$
 $\wedge (\forall a. HD \iota a \longrightarrow (\forall b. HD \iota b \longrightarrow (HAp d a = HAp d b \longrightarrow a = b)))$
 $\wedge (\exists c. HD \iota c \wedge (\forall z. HD \iota z \longrightarrow HAp d z \neq c))$
 proof –
 from *D0-inf* obtain $f c$ where $f: \forall a. D0 a \longrightarrow D0 (f a)$
 and $inj: \forall a b. D0 a \longrightarrow D0 b \longrightarrow f a = f b \longrightarrow a = b$
 and $c: D0 c$ and $ns: \forall a. D0 a \longrightarrow f a \neq c$ by blast
 have $sm: Sm (Hgr \iota f)$ by (rule Sm-Hgr[where $\tau = \iota$]) (use f in simp)
 have $dty: HD (\iota \Rightarrow \iota) (HLM \iota f)$ by (rule HLM-dom) (use f in simp)
 have $dapp: \bigwedge a. HD \iota a \Longrightarrow HAp (HLM \iota f) a = f a$ by (rule HAp-HLM[OF sm])
 have $\forall a. HD \iota a \longrightarrow (\forall b. HD \iota b \longrightarrow (HAp (HLM \iota f) a = HAp (HLM \iota f) b \longrightarrow a = b))$
 using inj by (simp add: $dapp$)
 moreover have $\exists c'. HD \iota c' \wedge (\forall z. HD \iota z \longrightarrow HAp (HLM \iota f) z \neq c')$
 using $c ns$ by (auto simp: $dapp intro!: exI[of - c]$)
 ultimately show ?thesis using dty by blast
 qed

5.4 Every Harrison universe is a *hol-universe*

theorem *harrison-hol-universe*: *hol-universe* *HD HAp HLm HTv HFv (HJv :: 'p $\Rightarrow$ ty $\Rightarrow$ 'u)*

proof *unfold-locales*

show $\bigwedge \sigma \tau g k. HD (\sigma \Rightarrow \tau) g \Rightarrow HD (\sigma \Rightarrow \tau) k \Rightarrow$
 $(\bigwedge a. HD \sigma a \Rightarrow HAp g a = HAp k a) \Rightarrow g = k$ **by** (*rule HAp-ext*)

show $\bigwedge a. HD \circ a \longleftrightarrow a = HTv \vee a = HFv$ **by** *simp*

qed(*safe intro!*: *HD-inf HJv-dom HTv-neq-HFv HAp-dom HLm-dom HAp-HLm*[*OF Sm-Hgr*])

lemmas *harrison-standard-model* =

lambda-universe.is-standard-model[*OF hol-universe.axioms*(1)][*OF harrison-hol-universe*]

theorem *con-full-HOL-harrison*:

con (insert (DInf :: 'p tm) (range ($\lambda(\sigma, \tau). ACrel \sigma \tau$)))

by (*rule hol-universe.con-full-HOL-universe*[*OF harrison-hol-universe*])

end

6 The set-theoretic instantiation: *V* as a *hol-universe*

This theory supplies the one ingredient the abstract half *HOL-Universe* leaves open: a type satisfying the *hol-universe* assumptions. Over Paulson's *ZFC_in_HOL* the individuals are interpreted as the finite ordinals ω and every function space is full, so the successor map witnesses Dedekind infinity and all assumptions of the locale are discharged. The consistency of full classical *HOL* and of *NK* with *DInf* then fall out of the abstract theorems; *V* also validates the at most as strong Harrison-style interface *harrison-universe* outright — and that most abstract interface carries the exported consistency theorem; a rank bound closes the development, showing every domain of the witnessing model lives inside the set *Vset* ($\omega + \omega$). Only this theory imports *ZFC-in-HOL*.

6.1 Domains, application, abstraction

primrec *D* :: *ty $\Rightarrow$ V* **where**

D $\iota = \omega$

| *D* $\circ = set \{0, 1\}$

| *D* ($\sigma \Rightarrow \tau$) = *VPi* (*D* σ) ($\lambda\cdot. D \tau$)

definition *VD* :: *ty $\Rightarrow$ V $\Rightarrow$ bool* **where** *VD* $\sigma v \equiv v \in elts (D \sigma)$

definition *VLm* :: *ty $\Rightarrow$ (V $\Rightarrow$ V) $\Rightarrow$ V* **where** *VLm* $\sigma h \equiv VLambda (D \sigma) h$

lemma *VD-bool*: *VD* $\circ v \longleftrightarrow v = 0 \vee v = 1$ **by** (*simp add: VD-def*)

lemma *VD-nonempty*: $\exists v. VD \sigma v$

proof (*induction* σ)

case (*Fun* $\sigma \tau$)

then obtain *w* **where** *w*: $w \in elts (D \tau)$ **by** (*auto simp: VD-def*)

have *VLambda* (*D* σ) ($\lambda\cdot. w$) $\in elts (D (\sigma \Rightarrow \tau))$ **by** (*simp add: VPi-I w*)

thus ?*case* **by** (*auto simp: VD-def*)

qed (*auto simp: VD-def*)

lemma *D-nonempty-elts*: $\exists v. v \in elts (D \sigma)$

using *VD-nonempty*[of σ] **by** (*simp add: VD-def*)

lemma *VLm-beta*: $VD \sigma a \implies app (VLm \sigma h) a = h a$ **by** (*simp add: VLm-def VD-def*)

lemma *VLm-dom*: $(\bigwedge d. VD \sigma d \implies VD \tau (h d)) \implies VD (\sigma \Rightarrow \tau) (VLm \sigma h)$

by (*simp add: VLm-def VD-def VPi-I*)

lemma *VAp-dom*: $VD (\sigma \Rightarrow \tau) f \implies VD \sigma a \implies VD \tau (app f a)$

by (*metis D.simps(3) VD-def VPi-D*)

lemma *VAp-ext*:

$VD (\sigma \Rightarrow \tau) f \implies VD (\sigma \Rightarrow \tau) g \implies (\bigwedge a. VD \sigma a \implies app f a = app g a) \implies f = g$

by (*metis D.simps(3) VD-def fun-ext*)

definition *VJv* :: $'p \Rightarrow ty \Rightarrow V$ **where** $VJv p \sigma = (SOME v. v \in elts (D \sigma))$

lemma *VJv-dom*: $VD \sigma (VJv p \sigma)$

unfolding *VJv-def VD-def* **using** *D-nonempty-elts*[of σ] **by** (*rule someI-ex*)

lemma *VJv-param-independent*: $VJv p \sigma = VJv q \sigma$

by (*simp add: VJv-def*)

The V -frame discharges the Dedekind-infinity assumption *ind-inf* of *hol-universe* (verbatim the frame premise of *con-full-HOL-rel*): the successor on ω is an injective, non-surjective self-map of the individuals.

lemma *V-inf*:

$\exists d. VD (\iota \Rightarrow \iota) d$

$\wedge (\forall a. VD \iota a \longrightarrow (\forall b. VD \iota b \longrightarrow (app d a = app d b \longrightarrow a = b)))$

$\wedge (\exists c. VD \iota c \wedge (\forall z. VD \iota z \longrightarrow app d z \neq c))$

proof –

have *beta*: $app (VLm \iota succ) a = succ a$ **if** $VD \iota a$ **for** a **by** (*rule VLm-beta[OF that]*)

have *dom*: $VD (\iota \Rightarrow \iota) (VLm \iota succ)$ **by** (*rule VLm-dom*) (*simp add: VD-def*)

have *invj*: $\forall a. VD \iota a \longrightarrow (\forall b. VD \iota b \longrightarrow (app (VLm \iota succ) a = app (VLm \iota succ) b \longrightarrow a = b))$

by (*auto simp: beta*)

have *nsv*: $\exists c. VD \iota c \wedge (\forall z. VD \iota z \longrightarrow app (VLm \iota succ) z \neq c)$

by (*rule exI[of - 0]*) (*auto simp: beta VD-def*)

show *?thesis* **using** *dom invj nsv* **by** *blast*

qed

V is such a universe: the *hol-universe* assumptions are satisfiable (relative to ZF) — yet they demand far less than full ZF .

theorem *V-hol-universe*:

hol-universe $VD app VLm (1::V) 0 (VJv :: 'p \Rightarrow ty \Rightarrow V)$

proof *unfold-locales*

show $\bigwedge \sigma \tau g k. VD (\sigma \Rightarrow \tau) g \implies VD (\sigma \Rightarrow \tau) k \implies$

$(\bigwedge a. VD \sigma a \implies app g a = app k a) \implies g = k$ **by** (*rule VAp-ext*)

qed(*auto intro!: V-inf VJv-dom VAp-dom VLm-dom VLm-beta simp: VD-bool*)

6.2 The standard model, by instantiation

The constructed constants of the locale turn the V -frame into a standard model outright — no hand-built logical constants over V are needed. This natural model carries the Dedekind witness and the rank bound below; the consistency *theorem* itself is exported once, from the Harrison-style interface in the next subsection.

lemmas *V-standard-model* =

lambda-universe.is-standard-model[*OF hol-universe.axioms(1)*][*OF V-hol-universe*]

6.3 V validates the Harrison-style axiom, and consistency follows

The at most as strong interface *harrison-universe* — our rendering of Harrison’s universe axiom — is likewise validated by V directly: pairing is *vpair*, a collection is small in the sense of the locale iff it is *small*, its code is Paulson’s *set*, and the individuals are again the finite ordinals. The coded-power closure is exactly *VPow* seen through the coding. The locale rebuilds domains, application and abstraction from these four constants alone, so the headline consistency theorem rests on nothing beyond them — one spine, through the most abstract interface.

lemma *V-enc-inj*:

assumes *small* $\{x. X x\}$ **and** *small* $\{x. Y x\}$ **and** $\text{set } \{x. X x\} = \text{set } \{x. Y x\}$
shows $X = Y$

proof –

have $\{x. X x\} = \{x. Y x\}$ **using** *arg-cong*[*OF* *assms*(3), *of elts*] *assms*(1,2) **by** *simp*
thus *?thesis* **unfolding** *fun-eq-iff* **by** (*metis mem-Collect-eq*)

qed

lemma *V-small-sub*: **assumes** *small* $\{x. X x\}$ **and** $Y \leq X$ **shows** *small* $\{x. Y x\}$

proof (*rule smaller-than-small*[*OF* *assms*(1)])

show $\{x. Y x\} \subseteq \{x. X x\}$ **using** *assms*(2) **by** (*auto simp: le-fun-def*)

qed

lemma *V-small-prod*:

assumes *small* $\{x. X x\}$ **and** *small* $\{x. Y x\}$
shows *small* $\{z. \exists a b. X a \wedge Y b \wedge z = \text{vpair } a b\}$

proof –

have $\{z. \exists a b. X a \wedge Y b \wedge z = \text{vpair } a b\} = (\lambda(a, b). \text{vpair } a b) \text{ ‘ } (\{x. X x\} \times \{x. Y x\})$
by *auto*
thus *?thesis* **using** *assms* **by** *simp*

qed

lemma *V-small-pow*:

assumes *sX*: *small* $\{x. X x\}$
shows *small* $\{z. \exists Y. Y \leq X \wedge \text{small } \{x. Y x\} \wedge z = \text{set } \{x. Y x\}\}$

proof (*rule smaller-than-small*[*OF* *small-elts*])

show $\{z. \exists Y. Y \leq X \wedge \text{small } \{x. Y x\} \wedge z = \text{set } \{x. Y x\}\} \subseteq \text{elts } (\text{VPow } (\text{set } \{x. X x\}))$

proof

fix z **assume** $z \in \{z. \exists Y. Y \leq X \wedge \text{small } \{x. Y x\} \wedge z = \text{set } \{x. Y x\}\}$
then obtain Y **where** $Y: Y \leq X$ **and** *sY*: *small* $\{x. Y x\}$ **and** $z: z = \text{set } \{x. Y x\}$

by *blast*

have $\text{elts } z \subseteq \text{elts } (\text{set } \{x. X x\})$ **using** *Y sX sY* **by** (*auto simp: z le-fun-def*)

thus $z \in \text{elts } (\text{VPow } (\text{set } \{x. X x\}))$ **by** (*simp add: less-eq-V-def*)

qed

qed

lemma *V-omega-inf*:

$\exists f c. (\forall a. a \in \text{elts } \omega \longrightarrow f a \in \text{elts } \omega)$
 $\wedge (\forall a b. a \in \text{elts } \omega \longrightarrow b \in \text{elts } \omega \longrightarrow f a = f b \longrightarrow a = b)$
 $\wedge c \in \text{elts } \omega \wedge (\forall a. a \in \text{elts } \omega \longrightarrow f a \neq c)$
by (*intro exI*[*of* - *succ*] *exI*[*of* - *0*]) (*auto simp: succ-neq-zero*)

theorem *V-harrison-universe*:

harrison-universe *vpair* $(\lambda X. \text{set } \{x. X x\})$ $(\lambda X. \text{small } \{x. X x\})$ $(\lambda x. x \in \text{elts } \omega)$

proof *unfold-locales*

```

show  $\bigwedge X Y. \text{small } \{x. X x\} \implies \text{small } \{x. Y x\} \implies \text{set } \{x. X x\} = \text{set } \{x. Y x\} \implies X = Y$ 
by (rule V-enc-inj)
show  $\bigwedge X Y. \text{small } \{x. X x\} \implies Y \leq X \implies \text{small } \{x. Y x\}$  by (rule V-small-sub)
qed(safe intro!: V-omega-inf V-small-pow V-small-prod | simp)+

```

theorem *con-full-HOL*:

```

con (insert (DInf :: 'p tm) (range ( $\lambda(\sigma, \tau). \text{ACrel } \sigma \tau$ )))
by (rule harrison-universe.con-full-HOL-harrison[OF V-harrison-universe])

```

6.4 The Dedekind axiom of infinity and a countable Henkin witness

The infinity-only consistency *con* $\{DInf\}$ is the same *V*-model argument, now feeding the successor witness on ω to the carrier-generic *DInf-sat*; it holds for *every* parameter type '*p*'. The *countable* model that witnesses it is the ZFC-free term model of *countable-henkin-sat*, so a set-theoretic universe enters *only* through the consistency premise — the model itself has a countable total domain, carved out of the carrier '*p* tm set, and never mentions *V*.

theorem *con-DInf*: *con* ($\{DInf\} :: 'p \text{ tm set}$)

proof –

```

interpret U: hol-universe VD app VLm 1::V 0::V VJv :: 'p  $\Rightarrow$  ty  $\Rightarrow$  V
by (rule V-hol-universe)

```

```

interpret standard-model VD app VLm 1::V 0::V
  U.Ngv U.Dsv U.Iv U.Ev U.Piv VJv :: 'p  $\Rightarrow$  ty  $\Rightarrow$  V
by (rule U.is-standard-model)

```

```

define  $\xi :: \text{nat} \Rightarrow \text{ty} \Rightarrow V$  where  $\xi = (\lambda n \tau. \text{SOME } v. \text{VD } \tau v)$ 

```

```

have xi: bkkA.asg  $\xi$ 

```

```

  unfolding bkkA.asg-def  $\xi$ -def using VD-nonempty by (metis someI-ex)

```

— the successor on ω is an injective, non-surjective self-map of the individuals

```

define f0 :: V where f0 = VLm  $\iota$  succ

```

```

have f0dom: VD ( $\iota \Rightarrow \iota$ ) f0 unfolding f0-def by (rule VLm-dom) (simp add: VD-def)

```

```

have f0app: app f0 a = succ a if VD  $\iota$  a for a

```

```

  unfolding f0-def by (rule VLm-beta[OF that])

```

```

have satD: den (DInf :: 'p tm)  $\xi = 1$ 

```

```

proof (rule DInf-sat[OF xi f0dom])

```

```

  show  $\forall a. \text{VD } \iota a \longrightarrow (\forall b. \text{VD } \iota b \longrightarrow (\text{app } f0 a = \text{app } f0 b \longrightarrow a = b))$ 

```

```

    by (auto simp: f0app)

```

```

  show  $\exists c. \text{VD } \iota c \wedge (\forall z. \text{VD } \iota z \longrightarrow \text{app } f0 z \neq c)$ 

```

```

    by (rule exI[of - 0]) (auto simp: f0app VD-def)

```

qed

```

show ?thesis by (rule model-con[OF xi]) (use wff-DInf satD in auto)

```

qed

theorem *countable-henkin-model-DInf*:

```

obtains Dm Ap

```

```

  and Ee :: (nat  $\Rightarrow$  ty  $\Rightarrow$  'p::{countable,infinite} tm set)  $\Rightarrow$  'p tm  $\Rightarrow$  'p tm set

```

```

  and vl  $\xi$ 

```

```

where bkk-model Dm Ap Ee vl countable  $\{x. \exists \tau. \text{Dm } \tau x\}$ 

```

```

  app-struct.asg Dm  $\xi vl$  (Ee  $\xi$  (DInf :: 'p tm))

```

```

by (rule countable-henkin-sat[OF cwff-DInf con-DInf])

```

The converse half of the separation of axiom and scheme. The main entry proves that the infinity *scheme* does not yield the axiom (*Diag-not-derives-DInf*, *henkin-scheme-refutes-DInf* in its *NK-Infinity*). The *V*-model settles the other direction: its parameter interpretation *VJv* is constant at each type (*VJv-param-independent*), so all diagram constants denote the same

individual and every inequation $dneq f i j$ is refuted, while $DInf$ holds. By soundness the axiom derives no diagram inequation — as *sentences*, axiom and scheme are incomparable.

theorem $DInf\text{-not-derives-dneq}$:

fixes $f :: nat \Rightarrow 'p::infinite$
shows $\neg (\{DInf\} \vdash dneq f i j)$

proof

assume $d: \{DInf\} \vdash dneq f i j$

interpret $U: hol\text{-universe } VD \text{ app } V Lm \ 1::V \ 0::V \ VJv :: 'p \Rightarrow ty \Rightarrow V$
by (rule $V\text{-hol-universe}$)

interpret $standard\text{-model } VD \text{ app } V Lm \ 1::V \ 0::V$
 $U.Ngv \ U.Dsv \ U.Iv \ U.Ev \ U.Piv \ VJv :: 'p \Rightarrow ty \Rightarrow V$
by (rule $U\text{-is-standard-model}$)

define $\xi :: nat \Rightarrow ty \Rightarrow V$ **where** $\xi = (\lambda n \ \tau. SOME \ v. \ VD \ \tau \ v)$

have $xi: bkkA.asg \ \xi$

unfolding $bkkA.asg\text{-def}$ $\xi\text{-def}$ **using** $VD\text{-nonempty}$ **by** (metis $someI\text{-ex}$)

— $DInf$ holds, by the successor witness of $con\text{-}DInf$

define $f0 :: V$ **where** $f0 = V Lm \ \iota \ succ$

have $f0dom: VD \ (\iota \Rightarrow \iota) \ f0$ **unfolding** $f0\text{-def}$ **by** (rule $V Lm\text{-dom}$) (simp add: $VD\text{-def}$)

have $f0app: app \ f0 \ a = succ \ a$ **if** $VD \ \iota \ a$ **for** a

unfolding $f0\text{-def}$ **by** (rule $V Lm\text{-beta}[OF \ that]$)

have $satD: den \ (DInf :: 'p \ tm) \ \xi = 1$

proof (rule $DInf\text{-sat}[OF \ xi \ f0dom]$)

show $\forall a. \ VD \ \iota \ a \longrightarrow (\forall b. \ VD \ \iota \ b \longrightarrow (app \ f0 \ a = app \ f0 \ b \longrightarrow a = b))$

by (auto simp: $f0app$)

show $\exists c. \ VD \ \iota \ c \wedge (\forall z. \ VD \ \iota \ z \longrightarrow app \ f0 \ z \neq c)$

by (rule $exI[of \ - \ 0]$) (auto simp: $f0app \ VD\text{-def}$)

qed

— the inequation fails: both parameters denote the same individual

have $peq: den \ ((f \ i)^p_\iota =_\iota (f \ j)^p_\iota) \ \xi = 1$

using $sat\text{-}PEqB[OF \ wff\text{-}Par \ wff\text{-}Par \ xi]$ $VJv\text{-param-independent}[of \ f \ i \ \iota \ f \ j]$ **by** simp

have $nsat: den \ (dneq \ f \ i \ j) \ \xi \neq 1$

unfolding $dneq\text{-def}$

using $sat\text{-}NegB[OF \ wff\text{-}PEq[OF \ wff\text{-}Par \ wff\text{-}Par] \ xi]$ peq **by** simp

have $den \ (dneq \ f \ i \ j) \ \xi = 1$

by (rule $soundness\text{-}bkk[OF \ d \ bkk\text{-}model\text{-}pred \ xi]$) (use $wff\text{-}DInf \ satD$ **in** auto)

with $nsat$ **show** $False$ **by** simp

qed

6.5 The universe has bounded rank: no replacement, no proper class

The rank bound is now made precise: *every* domain $D \ \sigma$ has rank below $\omega + \omega$ (that is, $\omega \cdot 2$), so every domain is an element of the set $Vset \ (\omega + \omega)$. The domain of individuals sits at rank ω , and each function-space layer $D \ (\sigma \Rightarrow \tau) = VPi \ (D \ \sigma) \ (\lambda\cdot. \ D \ \tau)$ — contained in $VPow \ (VPow \ (VPow \ (D \ \sigma \sqcup D \ \tau)))$ — raises the rank by only three, which the limit ordinal $\omega + \omega$ absorbs.

lemma $rank\text{-}D\text{-iota}: rank \ (D \ \iota) = \omega$ **by** (simp add: $rank\text{-of-Ord}$)

lemma $rank\text{-}VPow\text{-le}: rank \ (VPow \ x) \leq succ \ (rank \ x)$

proof —

have $small: small \ ((\lambda y. \ succ \ (rank \ y)) \text{ ‘ } elts \ (VPow \ x))$ **by** simp

have $rank \ (VPow \ x) = \bigsqcup ((\lambda y. \ succ \ (rank \ y)) \text{ ‘ } elts \ (VPow \ x))$ **by** (rule $rank\text{-}Sup$)

also have $\dots \leq succ \ (rank \ x)$

unfolding $SUP\text{-le-iff}[OF \ small]$

proof

```

    fix y assume y ∈ elts (VPow x)
    hence y ≤ x by simp
    hence rank y ≤ rank x by (rule rank-mono)
    thus succ (rank y) ≤ succ (rank x) by (simp add: le-succ-iff)
  qed
  finally show ?thesis .
qed

lemma sup-Sup-pair: (a::V) ⊔ b = ⊔ {a, b}
proof (rule order-antisym)
  have a ≤ ⊔ {a, b} by (intro Sup-upper) auto
  moreover have b ≤ ⊔ {a, b} by (intro Sup-upper) auto
  ultimately show a ⊔ b ≤ ⊔ {a, b} by simp
  show ⊔ {a, b} ≤ a ⊔ b by (rule Sup-least) auto
qed

lemma rank-sup-eq: rank (a ⊔ b) = rank a ⊔ rank b
  by (simp add: sup-Sup-pair rank-Union)

lemma VPi-le-VPow: VPi A B ≤ VPow (VSigma A B)
  by (auto simp: VPi-def less-eq-V-def)

lemma VSigma-const-le: VSigma A (λ-. C) ≤ VPow (VPow (A ⊔ C))
proof (subst less-eq-V-def, rule subsetI)
  fix p assume p ∈ elts (VSigma A (λ-. C))
  then obtain a b where p: p = ⟨a, b⟩ and a: a ∈ elts A and b: b ∈ elts C by auto
  have ⟨a, b⟩ ≤ VPow (A ⊔ C)
    unfolding vpair-def' using a b by (auto simp: less-eq-V-def)
  thus p ∈ elts (VPow (VPow (A ⊔ C))) using p by auto
qed

lemma Limit-ww: Limit (ω + ω) by simp

lemma succ-lt-ww:
  assumes Ord β and β < ω + ω shows succ β < ω + ω
proof -
  have O: Ord (ω + ω) using Limit-ww Limit-is-Ord by blast
  from asms O have β ∈ elts (ω + ω) by (simp add: Ord-mem-iff-lt)
  with Limit-ww have succ β ∈ elts (ω + ω) by (simp add: Limit-def)
  with O show ?thesis by (blast intro: OrdmemD)
qed

lemma omega-lt-ww: ω < ω + ω
proof -
  have (0::V) < ω by (metis OrdmemD Ord-ω zero-in-omega)
  hence ω + 0 < ω + ω by simp
  thus ?thesis by simp
qed

lemma rank-VPi-const-lt:
  assumes rank A < ω + ω and rank C < ω + ω
  shows rank (VPi A (λ-. C)) < ω + ω
proof -
  have s1: rank (VPi A (λ-. C)) ≤ succ (rank (VSigma A (λ-. C)))

```

```

  by (rule order-trans[OF rank-mono[OF VPi-le-VPow] rank-VPow-le])
have rank (VSigma A (λ-. C)) ≤ succ (rank (VPow (A ⊔ C)))
  by (rule order-trans[OF rank-mono[OF VSigma-const-le] rank-VPow-le])
also have ... ≤ succ (succ (rank (A ⊔ C)))
  by (simp add: le-succ-iff Ord-succ rank-VPow-le)
finally have s2: rank (VSigma A (λ-. C)) ≤ succ (succ (rank (A ⊔ C))) .
have succ (rank (VSigma A (λ-. C))) ≤ succ (succ (succ (rank (A ⊔ C))))
  using s2 by (simp add: le-succ-iff Ord-succ)
with s1 have s3: rank (VPi A (λ-. C)) ≤ succ (succ (succ (rank (A ⊔ C))))
  by (rule order-trans)
have r: rank (A ⊔ C) < ω + ω
  using assms by (simp add: rank-sup-eq)
have succ (rank (A ⊔ C)) < ω + ω using r by (simp add: succ-lt-ww)
hence succ (succ (rank (A ⊔ C))) < ω + ω by (simp add: succ-lt-ww Ord-succ)
hence succ (succ (succ (rank (A ⊔ C)))) < ω + ω by (simp add: succ-lt-ww Ord-succ)
with s3 show ?thesis by (rule order-le-less-trans)
qed

```

theorem *rank-D-lt*: $\text{rank } (D \sigma) < \omega + \omega$

proof (*induction* σ)

case *Ind* thus ?case by (simp add: rank-of-Ord omega-lt-ww)

next

case *Bool*

have $D \circ \leq \omega$ by (auto simp: less-eq-V-def one-V-def)

hence $\text{rank } (D \circ) \leq \omega$ by (metis Ord- ω rank-mono rank-of-Ord)

thus ?case using omega-lt-ww by (rule order-le-less-trans)

next

case (*Fun* $\sigma \tau$)

have $\text{rank } (D (\sigma \Rightarrow \tau)) = \text{rank } (VPi (D \sigma) (\lambda-. D \tau))$ by *simp*

thus ?case using *Fun.IH* by (simp add: rank-VPi-const-lt)

qed

Hence $D \sigma \in \text{elts } (Vset (\omega + \omega))$ for every type: each domain of the witnessing model is an element of the set $Vset (\omega + \omega)$.

lemma *D-in-Vset-ww*: $D \sigma \in \text{elts } (Vset (\omega + \omega))$

by (rule Ord-VsetI) (simp-all add: rank-D-lt)

References

- [1] C. Benzmüller, C. E. Brown, and M. Kohlhase. Higher-order semantics and extensionality. *Journal of Symbolic Logic*, 69(4):1027–1088, 2004.
- [2] C. Benzmüller and D. Kirchner. A deep embedding of HOL in HOL: Soundness, completeness, consistency. *Archive of Formal Proofs*, 2026. https://isa-afp.org/entries/HOL_in_HOL_Deep.html, Formal proof development.
- [3] A. Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5(2):56–68, 1940.
- [4] J. Harrison. Towards self-verification of HOL Light. In *Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lecture Notes in Computer Science*, pages 177–191. Springer, 2006.

- [5] R. Kumar, R. Arthan, M. O. Myreen, and S. Owens. Self-formalisation of higher-order logic: Semantics, soundness, and a verified implementation. *Journal of Automated Reasoning*, 56(3):221–259, 2016.
- [6] O. Kunčar and A. Popescu. A consistent foundation for Isabelle/HOL. *Journal of Automated Reasoning*, 62(4):531–555, 2019.
- [7] A. R. D. Mathias. The strength of Mac Lane set theory. *Annals of Pure and Applied Logic*, 110(1–3):107–234, 2001.
- [8] L. C. Paulson. Zermelo fraenkel set theory in higher-order logic. *Archive of Formal Proofs*, 2019. https://isa-afp.org/entries/ZFC_in_HOL.html, Formal proof development.