# From Abstract to Concrete Gödel's Incompleteness Theorems—Part I

Andrei Popescu    Dmitriy Traytel

March 17, 2025

**Abstract**

We validate an abstract formulation of Gödel's First and Second Incompleteness Theorems from a separate AFP entry by instantiating them to the case of *finite sound extensions of the Hereditarily Finite (HF) Set theory*, i.e., FOL theories extending the HF Set theory with a finite set of axioms that are sound in the standard model. The concrete results had been previously formalised in an AFP entry by Larry Paulson; our instantiation reuses the infrastructure developed in that entry.

## Contents



## 1 The Instantiation

**definition** $\mathit{Fvars}\ t = \{a :: \mathit{name}.\ \neg\ \mathit{atom}\ a\ \sharp\ t\}$

**lemma** *Fvars_tm_simps*[*simp*]:
 $\mathit{Fvars}\ \mathit{Zero} = \{\}$
 $\mathit{Fvars}\ (\mathit{Var}\ a) = \{a\}$
 $\mathit{Fvars}\ (\mathit{Eats}\ x\ y) = \mathit{Fvars}\ x \cup \mathit{Fvars}\ y$
 ⟨*proof*⟩

**lemma** *finite_Fvars_tm*[*simp*]:
 **fixes** $t :: \mathit{tm}$
 **shows** $\mathit{finite}\ (\mathit{Fvars}\ t)$
 ⟨*proof*⟩

**lemma** *Fvars_fm_simps*[*simp*]:
 $\mathit{Fvars}\ (x\ \mathit{IN}\ y) = \mathit{Fvars}\ x \cup \mathit{Fvars}\ y$
 $\mathit{Fvars}\ (x\ \mathit{EQ}\ y) = \mathit{Fvars}\ x \cup \mathit{Fvars}\ y$
 $\mathit{Fvars}\ (A\ \mathit{OR}\ B) = \mathit{Fvars}\ A \cup \mathit{Fvars}\ B$
 $\mathit{Fvars}\ (A\ \mathit{AND}\ B) = \mathit{Fvars}\ A \cup \mathit{Fvars}\ B$
 $\mathit{Fvars}\ (A\ \mathit{IMP}\ B) = \mathit{Fvars}\ A \cup \mathit{Fvars}\ B$
 $\mathit{Fvars}\ \mathit{Fls} = \{\}$
 $\mathit{Fvars}\ (\mathit{Neg}\ A) = \mathit{Fvars}\ A$
 $\mathit{Fvars}\ (\mathit{Ex}\ a\ A) = \mathit{Fvars}\ A - \{a\}$
 $\mathit{Fvars}\ (\mathit{All}\ a\ A) = \mathit{Fvars}\ A - \{a\}$
 ⟨*proof*⟩

**lemma** *finite_Fvars_fm*[*simp*]:
 **fixes** $A :: \mathit{fm}$
 **shows** $\mathit{finite}\ (\mathit{Fvars}\ A)$

$\langle proof \rangle$

**lemma** *subst_tm_subst_tm*[*simp*]:
$x \neq y \Longrightarrow atom\ x \sharp u \Longrightarrow subst\ y\ u\ (subst\ x\ t\ v) = subst\ x\ (subst\ y\ u\ t)\ (subst\ y\ u\ v)$
$\langle proof \rangle$

**lemma** *subst_fm_subst_fm*[*simp*]:
$x \neq y \Longrightarrow atom\ x \sharp u \Longrightarrow (A(x{::=}t))(y{::=}u) = (A(y{::=}u))(x{::=}subst\ y\ u\ t)$
$\langle proof \rangle$

**lemma** *Fvars_ground_aux*: $Fvars\ t \subseteq B \Longrightarrow ground\_aux\ t\ (atom\ `\ B)$
$\langle proof \rangle$

**lemma** *ground_Fvars*: $ground\ t \longleftrightarrow Fvars\ t = \{\}$
$\langle proof \rangle$

**lemma** *Fvars_ground_fm_aux*: $Fvars\ A \subseteq B \Longrightarrow ground\_fm\_aux\ A\ (atom\ `\ B)$
$\langle proof \rangle$

**lemma** *ground_fm_Fvars*: $ground\_fm\ A \longleftrightarrow Fvars\ A = \{\}$
$\langle proof \rangle$

**interpretation** *Generic_Syntax* **where**
$var = UNIV :: name\ set$
**and** $trm = UNIV :: tm\ set$
**and** $fmla = UNIV :: fm\ set$
**and** $Var = Var$
**and** $FvarsT = Fvars$
**and** $substT = \lambda t\ u\ x.\ subst\ x\ u\ t$
**and** $Fvars = Fvars$
**and** $subst = \lambda A\ u\ x.\ subst\_fm\ A\ x\ u$
$\langle proof \rangle$

**lemma** *coding_tm_Fvars_empty*[*simp*]: $coding\_tm\ t \Longrightarrow Fvars\ t = \{\}$
$\langle proof \rangle$

**lemma** *Fvars_empty_ground*[*simp*]: $Fvars\ t = \{\} \Longrightarrow ground\ t$
$\langle proof \rangle$

**interpretation** *Syntax_with_Numerals* **where**
$var = UNIV :: name\ set$
**and** $trm = UNIV :: tm\ set$
**and** $fmla = UNIV :: fm\ set$
**and** $num = \{t.\ ground\ t\}$
**and** $Var = Var$
**and** $FvarsT = Fvars$
**and** $substT = \lambda t\ u\ x.\ subst\ x\ u\ t$
**and** $Fvars = Fvars$
**and** $subst = \lambda A\ u\ x.\ subst\_fm\ A\ x\ u$
$\langle proof \rangle$

**declare** *FvarsT_num*[*simp del*]

**interpretation** *Deduct2_with_False* **where**
$var = UNIV :: name\ set$
**and** $trm = UNIV :: tm\ set$
**and** $fmla = UNIV :: fm\ set$
**and** $num = \{t.\ ground\ t\}$

**and** *Var* = *Var*
**and** *FvarsT* = *Fvars*
**and** *substT* = $\lambda$*t u x. subst x u t*
**and** *Fvars* = *Fvars*
**and** *subst* = $\lambda$*A u x. subst_fm A x u*
**and** *eql* = (*EQ*)
**and** *cnj* = (*AND*)
**and** *imp* = (*IMP*)
**and** *all* = *All*
**and** *exi* = *Ex*
**and** *fls* = *Fls*
**and** *prv* = ($\vdash$) {}
**and** *bprv* = ($\vdash$) {}
⟨*proof*⟩

**interpretation** *HBL1* **where**
*var* = *UNIV* :: *name set*
**and** *trm* = *UNIV* :: *tm set*
**and** *fmla* = *UNIV* :: *fm set*
**and** *num* = {*t. ground t*}
**and** *Var* = *Var*
**and** *FvarsT* = *Fvars*
**and** *substT* = $\lambda$*t u x. subst x u t*
**and** *Fvars* = *Fvars*
**and** *subst* = $\lambda$*A u x. subst_fm A x u*
**and** *eql* = (*EQ*)
**and** *cnj* = (*AND*)
**and** *imp* = (*IMP*)
**and** *all* = *All*
**and** *exi* = *Ex*
**and** *prv* = ($\vdash$) {}
**and** *bprv* = ($\vdash$) {}
**and** *enc* = *quot*
**and** *P* = *PfP* (*Var xx*)
⟨*proof*⟩

**interpretation** *Goedel_Form* **where**
*var* = *UNIV* :: *name set*
**and** *trm* = *UNIV* :: *tm set*
**and** *fmla* = *UNIV* :: *fm set*
**and** *num* = {*t. ground t*}
**and** *Var* = *Var*
**and** *FvarsT* = *Fvars*
**and** *substT* = $\lambda$*t u x. subst x u t*
**and** *Fvars* = *Fvars*
**and** *subst* = $\lambda$*A u x. subst_fm A x u*
**and** *eql* = (*EQ*)
**and** *cnj* = (*AND*)
**and** *imp* = (*IMP*)
**and** *all* = *All*
**and** *exi* = *Ex*
**and** *fls* = *Fls*
**and** *prv* = ($\vdash$) {}
**and** *bprv* = ($\vdash$) {}
**and** *enc* = *quot*
**and** *S* = *KRP* (*quot* (*Var xx*)) (*Var xx*) (*Var yy*)
**and** *P* = *PfP* (*Var xx*)
⟨*proof*⟩

**interpretation** *g2*: *Goedel_Second_Assumptions* **where**
*var* = *UNIV* :: *name set*
**and** *trm* = *UNIV* :: *tm set*
**and** *fmla* = *UNIV* :: *fm set*
**and** *num* = {*t*. *ground t*}
**and** *Var* = *Var*
**and** *FvarsT* = *Fvars*
**and** *substT* = λ*t u x*. *subst x u t*
**and** *Fvars* = *Fvars*
**and** *subst* = λ*A u x*. *subst_fm A x u*
**and** *eql* = (*EQ*)
**and** *cnj* = (*AND*)
**and** *imp* = (*IMP*)
**and** *all* = *All*
**and** *exi* = *Ex*
**and** *fls* = *Fls*
**and** *prv* = (⊢) {}
**and** *bprv* = (⊢) {}
**and** *enc* = *quot*
**and** *S* = *KRP* (*quot* (*Var xx*)) (*Var xx*) (*Var yy*)
**and** *P* = *PfP* (*Var xx*)
⟨*proof*⟩

**theorem** *Goedel_II*: ¬ {} ⊢ *Fls* ⟹ ¬ {} ⊢ *neg* (*PfP* «*Fls*»)
⟨*proof*⟩

**lemma** *ground_fm_PrfP*[*simp*]:
*ground_fm* (*PrfP s k t*) ⟷ *ground s* ∧ *ground k* ∧ *ground t*
⟨*proof*⟩

**lemma** *Fvars_HPair*[*simp*]: *Fvars* (*HPair t u*) = *Fvars t* ∪ *Fvars u*
⟨*proof*⟩

**lemma** *ground_HPair*[*simp*]: *ground* (*HPair t u*) ⟷ *ground t* ∧ *ground u*
⟨*proof*⟩

**interpretation** *dwfd*: *Deduct2_with_False_Disj* **where**
*var* = *UNIV* :: *name set*
**and** *trm* = *UNIV* :: *tm set*
**and** *fmla* = *UNIV* :: *fm set*
**and** *num* = {*t*. *ground t*}
**and** *Var* = *Var*
**and** *FvarsT* = *Fvars*
**and** *substT* = λ*t u x*. *subst x u t*
**and** *Fvars* = *Fvars*
**and** *subst* = λ*A u x*. *subst_fm A x u*
**and** *eql* = (*EQ*)
**and** *cnj* = (*AND*)
**and** *dsj* = (*OR*)
**and** *imp* = (*IMP*)
**and** *all* = *All*
**and** *exi* = *Ex*
**and** *fls* = *Fls*
**and** *prv* = (⊢) {}
**and** *bprv* = (⊢) {}
⟨*proof*⟩

**interpretation** *Minimal_Truth_Soundness* **where**
$var = UNIV :: name\ set$
**and** $trm = UNIV :: tm\ set$
**and** $fmla = UNIV :: fm\ set$
**and** $num = \{t.\ ground\ t\}$
**and** $Var = Var$
**and** $FvarsT = Fvars$
**and** $substT = \lambda t\ u\ x.\ subst\ x\ u\ t$
**and** $Fvars = Fvars$
**and** $subst = \lambda A\ u\ x.\ subst\_fm\ A\ x\ u$
**and** $eql = (EQ)$
**and** $cnj = (AND)$
**and** $dsj = (OR)$
**and** $imp = (IMP)$
**and** $all = All$
**and** $exi = Ex$
**and** $fls = Fls$
**and** $prv = (\vdash)\ \{\}$
**and** $isTrue = eval\_fm\ e0$
⟨*proof*⟩

**lemma** *neg_Neg*:
$\{\} \vdash neg\ \varphi\ IFF\ Neg\ \varphi$
⟨*proof*⟩

**lemma** *ground_aux_mono*: $A \subseteq B \Longrightarrow ground\_aux\ t\ A \Longrightarrow ground\_aux\ t\ B$
⟨*proof*⟩

**interpretation** *g1*: *Goedel_Form_Minimal_Truth_Soundness_HBL1iff_prv_Compl_Pf_Classical* **where**
$var = UNIV :: name\ set$
**and** $trm = UNIV :: tm\ set$
**and** $fmla = UNIV :: fm\ set$
**and** $num = \{t.\ ground\ t\}$
**and** $Var = Var$
**and** $FvarsT = Fvars$
**and** $substT = \lambda t\ u\ x.\ subst\ x\ u\ t$
**and** $Fvars = Fvars$
**and** $subst = \lambda A\ u\ x.\ subst\_fm\ A\ x\ u$
**and** $eql = (EQ)$
**and** $cnj = (AND)$
**and** $dsj = (OR)$
**and** $imp = (IMP)$
**and** $all = All$
**and** $exi = Ex$
**and** $fls = Fls$
**and** $prv = (\vdash)\ \{\}$
**and** $bprv = (\vdash)\ \{\}$
**and** $enc = quot$
**and** $S = KRP\ (quot\ (Var\ xx))\ (Var\ xx)\ (Var\ yy)$
**and** $P = PfP\ (Var\ xx)$
**and** $isTrue = eval\_fm\ e0$
**and** $Pf = Ex\ xx'\ (Ex\ yy'\ (Var\ yy\ EQ\ HPair\ (Var\ xx')\ (Var\ yy')\ AND\ PrfP\ (Var\ xx')\ (Var\ yy')\ (Var\ xx)))$
⟨*proof*⟩

**theorem** *Goedel_I*: $\exists\varphi.\ \neg\ \{\} \vdash \varphi \wedge \neg\ \{\} \vdash Neg\ \varphi \wedge eval\_fm\ e0\ \varphi$

⟨*proof*⟩

The following interpretation is redundant, because *Goedel_Form_Minimal_Truth_Soundness_HBL1iff_prv_Compl_*
(interpreted above) is a sublocale of *Goedel_Form_Classical_HBL1_rev_prv_Minimal_Truth_Soundness_TIP*. However, the latter requires less infrastructure (no Pf formula).

The definition of isTrue prevents Isabelle from noticing that the locale has already been interpreted via the above g1 interpretation of *Goedel_Form_Minimal_Truth_Soundness_HBL1iff_prv_Compl_Pf_Classical*.

**definition** *isTrue* **where**
  *isTrue* = *eval_fm e0*

**interpretation** *g1'*: *Goedel_Form_Classical_HBL1_rev_prv_Minimal_Truth_Soundness_TIP* **where**
    *var* = *UNIV* :: *name set*
  **and** *trm* = *UNIV* :: *tm set*
  **and** *fmla* = *UNIV* :: *fm set*
  **and** *num* = {*t*. *ground t*}
  **and** *Var* = *Var*
  **and** *FvarsT* = *Fvars*
  **and** *substT* = λ*t u x*. *subst x u t*
  **and** *Fvars* = *Fvars*
  **and** *subst* = λ*A u x*. *subst_fm A x u*
  **and** *eql* = (*EQ*)
  **and** *cnj* = (*AND*)
  **and** *dsj* = (*OR*)
  **and** *imp* = (*IMP*)
  **and** *all* = *All*
  **and** *exi* = *Ex*
  **and** *fls* = *Fls*
  **and** *prv* = (⊢) {}
  **and** *bprv* = (⊢) {}
  **and** *enc* = *quot*
  **and** *S* = *KRP* (*quot* (*Var xx*)) (*Var xx*) (*Var yy*)
  **and** *P* = *PfP* (*Var xx*)
  **and** *isTrue* = *isTrue*
  ⟨*proof*⟩

**theorem** *Goedel_I'*: ∃φ. ¬ {} ⊢ φ ∧ ¬ {} ⊢ *Neg* φ ∧ *isTrue* φ
  ⟨*proof*⟩