

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

Andrei Popescu      Dmitriy Traytel

February 6, 2026

## 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

1

## 1 The Instantiation

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

**lemma**  $Fvars\_tm\_simps[simp]$ :

$Fvars\ Zero = \{\}$   
 $Fvars\ (Var\ a) = \{a\}$   
 $Fvars\ (Eats\ x\ y) = Fvars\ x \cup Fvars\ y$   
*<proof>*

**lemma**  $finite\_Fvars\_tm[simp]$ :

**fixes**  $t :: tm$   
**shows**  $finite\ (Fvars\ t)$   
*<proof>*

**lemma**  $Fvars\_fm\_simps[simp]$ :

$Fvars\ (x\ IN\ y) = Fvars\ x \cup Fvars\ y$   
 $Fvars\ (x\ EQ\ y) = Fvars\ x \cup Fvars\ y$   
 $Fvars\ (A\ OR\ B) = Fvars\ A \cup Fvars\ B$   
 $Fvars\ (A\ AND\ B) = Fvars\ A \cup Fvars\ B$   
 $Fvars\ (A\ IMP\ B) = Fvars\ A \cup Fvars\ B$   
 $Fvars\ Fls = \{\}$   
 $Fvars\ (Neg\ A) = Fvars\ A$   
 $Fvars\ (Ex\ a\ A) = Fvars\ A - \{a\}$   
 $Fvars\ (All\ a\ A) = Fvars\ A - \{a\}$   
*<proof>*

**lemma**  $finite\_Fvars\_fm[simp]$ :

**fixes**  $A :: fm$   
**shows**  $finite\ (Fvars\ A)$

*<proof>*

**lemma** *subst\_tm\_subst\_tm*[simp]:

$x \neq y \implies \text{atom } x \# u \implies \text{subst } y \ u \ (\text{subst } x \ t \ v) = \text{subst } x \ (\text{subst } y \ u \ t) \ (\text{subst } y \ u \ v)$

*<proof>*

**lemma** *subst\_fm\_subst\_fm*[simp]:

$x \neq y \implies \text{atom } x \# u \implies (A(x::=t))(y::=u) = (A(y::=u))(x::=\text{subst } y \ u \ t)$

*<proof>*

**lemma** *Fvars\_ground\_aux*:  $Fvars \ t \subseteq B \implies \text{ground\_aux } t \ (\text{atom } 'B)$

*<proof>*

**lemma** *ground\_Fvars*:  $\text{ground } t \longleftrightarrow Fvars \ t = \{\}$

*<proof>*

**lemma** *Fvars\_ground\_fm\_aux*:  $Fvars \ A \subseteq B \implies \text{ground\_fm\_aux } A \ (\text{atom } 'B)$

*<proof>*

**lemma** *ground\_fm\_Fvars*:  $\text{ground\_fm } A \longleftrightarrow Fvars \ A = \{\}$

*<proof>*

**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* =  $\lambda t \ u \ x. \text{subst } x \ u \ t$

**and** *Fvars* = Fvars

**and** *subst* =  $\lambda A \ u \ x. \text{subst\_fm } A \ x \ u$

*<proof>*

**lemma** *coding\_tm\_Fvars\_empty*[simp]:  $\text{coding\_tm } t \implies Fvars \ t = \{\}$

*<proof>*

**lemma** *Fvars\_empty\_ground*[simp]:  $Fvars \ t = \{\} \implies \text{ground } t$

*<proof>*

**interpretation** *Syntax\_with\_Numerals* **where**

*var* = UNIV :: name set

**and** *trm* = UNIV :: tm set

**and** *fmla* = UNIV :: fm set

**and** *num* =  $\{t. \text{ground } t\}$

**and** *Var* = Var

**and** *FvarsT* = Fvars

**and** *substT* =  $\lambda t \ u \ x. \text{subst } x \ u \ t$

**and** *Fvars* = Fvars

**and** *subst* =  $\lambda A \ u \ x. \text{subst\_fm } A \ x \ u$

*<proof>*

**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* =  $\{t. \text{ground } t\}$

```

and Var = Var
and FvarsT = Fvars
and substT =  $\lambda t u x. \text{subst } x u t$ 
and Fvars = Fvars
and subst =  $\lambda A u x. \text{subst\_fm } A x u$ 
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 =  $\lambda t u x. \text{subst } x u t$ 
and Fvars = Fvars
and subst =  $\lambda A u x. \text{subst\_fm } A x u$ 
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 =  $\lambda t u x. \text{subst } x u t$ 
and Fvars = Fvars
and subst =  $\lambda A u x. \text{subst\_fm } A x u$ 
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 =  $\lambda t u x. \text{subst } x u t$ 
  and Fvars = Fvars
  and subst =  $\lambda A u x. \text{subst\_fm } A x u$ 
  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>
```

**theorem** *Goedel\_II*:  $\neg \{\} \vdash \text{Fls} \implies \neg \{\} \vdash \text{neg (PfP «Fls»)}$   
<proof>

**lemma** *ground\_fm\_PrfP[simp]*:  
 $\text{ground\_fm (PrfP } s k t) \longleftrightarrow \text{ground } s \wedge \text{ground } k \wedge \text{ground } t$   
<proof>

**lemma** *Fvars\_HPPair[simp]*:  $\text{Fvars (HPair } t u) = \text{Fvars } t \cup \text{Fvars } u$   
<proof>

**lemma** *ground\_HPPair[simp]*:  $\text{ground (HPair } t u) \longleftrightarrow \text{ground } t \wedge \text{ground } u$   
<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 =  $\lambda t u x. \text{subst } x u t$ 
  and Fvars = Fvars
  and subst =  $\lambda A u x. \text{subst\_fm } A x u$ 
  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 = ( $\vdash$ ) {}
  <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 =  $\lambda t u x. \text{subst } x u t$ 
  and Fvars = Fvars
  and subst =  $\lambda A u x. \text{subst\_fm } A x u$ 
  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*:  $A \subseteq B \implies \text{ground\_aux } t A \implies \text{ground\_aux } t B$

<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 =  $\lambda t u x. \text{subst } x u t$ 
  and Fvars = Fvars
  and subst =  $\lambda A u x. \text{subst\_fm } A x u$ 
  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*:  $\exists \varphi. \neg \{\} \vdash \varphi \wedge \neg \{\} \vdash \text{Neg } \varphi \wedge \text{eval\_fm } e0 \varphi$

*<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* =  $\lambda t u x. \text{subst } x u t$

**and** *Fvars* = *Fvars*

**and** *subst* =  $\lambda A u x. \text{subst\_fm } A x u$

**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* = *isTrue*

*<proof>*

**theorem** *Goedel\_I'*:  $\exists \varphi. \neg \{ \} \vdash \varphi \wedge \neg \{ \} \vdash \text{Neg } \varphi \wedge \text{isTrue } \varphi$

*<proof>*