Formalization of set theory fragments in Isabelle/HOL

Zuzana Haniková and Štěpán Holub

CLG & BL, February 5, 2026

Outline

- Motivation:
  - unified and verified results
  - exploring Alternative Set Theory
- Design principles
  - locales
  - axiom schemes
- Results
  - dependence
  - independence (models)

Motivation

- AST without semisets is just an axiomatization of heridatarily finite sets (`ZFfin`).
- The literature on `ZFfin`/`PA` is rich but often sketchy.
- When dealing with fragments of `ZFfin`, detail matters.

Results scattered in the literature

P. Hájek and P. Vopěnka, *Über die Gültigkeit des Fundierungsaxioms in speziellen Systemen der Mengentheorie*, **Z. Math. Logik Grundlagen Math.** vol. 9 (1963), pp. 235–241.

L. Rieger, *A contribution to Gödel's axiomatic set theory*, **Czechoslovak Math. J.** vol. 7 (1957), pp. 323–357.

P. Hájek and P.Pudlák, **Metamathematics of First-order Arithmetic, Perspectives in Mathematical Logic**, Springer-Verlag, Berlin, 1998.

A. Sochor, *Metamathematics of the alternative set theory. I-III*, **Commentationes Mathematicae Universitatis Carolinae**, (1979, 1982, 1983)

L. Běhounek, *Nezávislost axiomů ve dvou axiomatikách teorie konečných množin*, **Ročníková práce**, 1998

Results scattered in the literature

S. Baratella and R. Ferro, *A theory of sets with the negation of the axiom of infinity*, **Mathematical Logic Quarterly** vol. 39 (1993), pp. 338-352.

R. Kaye and T. Wong, *On interpretations of arithmetic and set theory*, **Notre Dame Journal of Formal Logic**, vol. 48, Number 4 (2007), pp. 497-510.

A. Enayat, James H. Schmerl, and A. Visser, *ω-models of finite set theory* **Set Theory, Arithmetic, and Foundations of Mathematics**, Cambridge 2011

A., Mancini and D., Zambella, *A note on recursive models of set theories*, **Notre Dame Journal of Formal Logic**, vol. 42 (2001), no. 2, pp. 109–115.

Locales

- Machinery for uniform treatment of a set of assumptions
- Dependence through **sublocale**
- Automatic transfer of results (**interpretation**)
- Logically, **locale** is a *predicate* about its *parameters*

Locales

```
locale set_signature =
  fixes membership :: "'a ⇒ 'a ⇒ bool" (infix "ε" 50)
```

```
locale L_setext = set_signature +
  assumes setext: "x = y ⟷ (∀ z. z ε x ⟷ z ε y)"
```

```
locale L_setext_empty_setsuc_setind = L_setext + L_empty + L_setsuc + L_setind
```

```
sublocale L_union
proof (unfold_locales, rule, rule setind_SP[rule_format])
  show "∃z. ∀u. (u ε z) ⟷ (∃v. v ε ∅ ∧ u ε v)"
    by (meson empty_is_empty)
next
  fix x y
  have aux: "(∀u. (u ε z) ⟷ (∃v. (v ε x ∨ v = y) ∧ u ε v)) ⟷ (∀u. (u ε z) ⟷ (∃ v. v
  v) ∨ u ε y)" for z
    by blast
  let ?Q = "λ z. ∀u. (u ε z) = (∃v. (v ε x ∨ v = y) ∧ u ε v)"
  assume "∃z. ∀u. (u ε z) ⟷ (∃v. v ε x ∧ u ε v)"
  thus "∃z. ∀u. (u ε z) ⟷ (∃v. v ε (x∪M{y}M) ∧ u ε v)"
    unfolding setsuc_def' aux using binunion_ex[rule_format, of _ y] by metis
next
  show "SetProperty (λa. ∃z. ∀u. (u ε z) = (∃v. v ε a ∧ u ε v))"
    unfolding SetProperty_def logsimps by rule+
qed
```

Schemes and set formula predicate

```
locale L_setind = set_signature +
  assumes setind: "⋀ P. SetFormulaPredicate P ⟹
    ∀ x. P(Ξ(0:= ∅)) ⟶ (∀ x y. P(Ξ(0:= x)) ⟶ P (Ξ(0:= x∪M{y}M))) 
    ⟶ P(Ξ(0:= x))"
```

Formula is represented as a truth function of variable evaluations $\Xi : (i \mapsto a_i)$.

```
inductive SetFormulaPredicate :: "((nat ⇒ 'a) ⇒ bool) ⇒ bool"
  where
  SFP_mem[simp]: "⋀ m n. SetFormulaPredicate (λ Ξ. (Ξ m) ε (Ξ n))"
  — ‹formula ‹xm ε xn››
| SFP_eq[simp]: "⋀ m n. SetFormulaPredicate (λ Ξ. (Ξ m) = (Ξ n))"
— ‹formula ‹xm = xn››
| SFP_neg[simp]: "SetFormulaPredicate P ⟹ SetFormulaPredicate (λ
Ξ. ¬ P Ξ)" — ‹formula ‹¬ φ››
| SFP_disj[simp]: "SetFormulaPredicate P ⟹ SetFormulaPredicate Q
        ⟹ SetFormulaPredicate (λ Ξ. P Ξ ∨ Q Ξ)"  — ‹formula ‹φ ∨
       ψ››
| SFP_all[simp]: "⋀ n. SetFormulaPredicate P ⟹
SetFormulaPredicate (λ Ξ. ∀ a. P (Ξ(n:=a)))"
    — ‹formula ‹∀ xn. φ››
```

Dependences and equivalences

Finiteness axioms:

- Inductive set existence negated
- Tarski finiteness (maximal element w.r.t inclusion)
- Dedekind finiteness (no bijection on a proper subset)
- AST-finite (set-successor and scheme of induction)
- Cardinality-finite (bijection on a natural number)

Dependences and equivalences

```
theorem (in L_setext_empty_setsuc)
  shows setind_implies_tarski: "L_setind (ε) ⟹ L_tarski (ε)" and
        setind_implies_fin_by_setsuc: "L_setind (ε) ⟹ L_fin (ε)"
```

```
theorem (in L_setext_empty_power_union_repl)
   dedekind_implies_fin: "L_dedekind (ε) ⟹ L_fin (ε)"
```

```
theorem (in L_setext_empty_power_union_repl_reg)
        fin_implies_tarski: "L_fin (ε) ⟹ L_tarski (ε)" and
        tarski_implies_fin: "L_tarski (ε) ⟹ L_fin (ε)" and
        fin_implies_setind: "L_fin (ε) ⟹ L_setind (ε)" and
        setind_implies_fin: "L_setind (ε) ⟹ L_fin (ε)" and
        fin_implies_dedekind: "L_fin (ε) ⟹ L_dedekind (ε)"
```

Independence and models

Hereditarily finite sets are implemented in Isabelle through Ackermann encoding of `nat`, denoted as $\in$.

```
interpretation zffin: ZFfin "(∈)"
  rewrites "zffin.empty_setM = 0" and
           "zffin.singletonM y = {|y|}" and
           "zffin.binunionM x (zffin.singletonM y) = x ◁ y"
```

Modification: $0 \in 0$, $1 = \emptyset$, the rest as by Ackermann

```
definition hmem' :: "hf ⇒ hf ⇒ bool"  (infixl ‹∈ʳ› 50)
  where "hmem' a b ⟷ (b ≠ 1 ∧ a ∈ b) ∨ (a = 0 ∧ b = 0)"
```

```
theorem "L_setext_empty (∈ʳ)" and "L_setsuc (∈ʳ)" and "L_setind (∈ʳ)"
and "¬ L_reg (∈ʳ)"
  by unfold_locales  (metis L_reg.reg zero_hmem'_iff)
```

Independence and models

```
theorem not_reg_implies_regsch:
  assumes "L_setext_empty_power_union_repl_reg_fin (m :: 'a ⇒ 'a ⇒ bool)"
  shows "¬ (∀ (mem :: 'a ⇒ 'a ⇒ bool). L_setext_empty_power_union_repl mem
  ∧ L_reg mem ⟶ L_regsch mem)"
```

- Mancini-Zambella example, using permutation
- They call it Fraenkel-Mostowski permutation method
- Enyat et. al remark: “the method was invented by Bernays and fine-tuned by Rieger”

Thank you