Heriditarily Finite Sets in Constructive Type Theory

Gert Smolka

Saarland University

Based on joint work with Kathrin Stark
Paper at ITP 2016 (Springer LNCS 9807)

HF Sets in Naive Set Theory

- An HF set is a finite set of HF sets
- Inductive definition
- Pure sets

- An HF set is a set whose transitive closure is finite
- Transitive closure: least superset closed under elements of elements

- We consider only wellfounded HF sets (e.g., $x \notin x$)
- All sets are well-founded in ZF set theory

Epsilon Induction

- A property p holds for all sets if $\forall x.\ (\forall z \in x.\ pz) \to px$
- Epsilon induction is valid iff all sets are well-founded

Adjunction

$$x.y \;:=\; \{x\} \cup y$$

- Similar to cons for lists
- Can express membership: $x \in y \;\leftrightarrow\; x.y = y$

HF Sets as Numbers (Ackermann 1937)

- $m \in n$ iff position m in binary representation of n is 1
- Example: $21 \rightsquigarrow 10101 \rightsquigarrow \{4, 2, 0\}$
- Yields model of ZF without infinity

HF Sets Simplify Gödel's Incompleteness Proof

- Świerczkowski 2003
- Paulson 2015 (formalisation in Isabelle/HOL)

- Useful data structure for state sets of automata in HOL (Paulson 2015)

Peano Axiomatisation of Numbers

- N : Type, $0 : N$, $S : N \to N$
- $\forall p.\ p0 \to (\forall n.pn \to p(Sn)) \to \forall n.pn$
- $0 \neq Sn$
- $Sm = Sn \to m = n$

- Unique model (up to isomorphism)
- Computationally complete if $p : N \to$ Type
- Can define primitive recursion operator

Axiomatisation of Binary Trees

- T : Type, $\emptyset : T$, $. : T \to T \to T$
- $\forall p.\ p\emptyset \to (\forall xy.px \to py \to p(x.y)) \to \forall x.px$
- $\emptyset \neq x.y$
- $x.y = x'.y' \to x = x' \land y = y'$

- Unique model, computationally complete

- Axiomatisation of lists is similar

Axiomatisations of HF Sets

- Different from ZF
- Givant and Tarski 1977, Takahashi 1977 (classical)
  - $\emptyset$, $x.y$, $x \in y$
  - induction principle based on $\emptyset$ and $x.y$
  - extensionality axiom
- Previale 1994 (intuitionistic)
  - $\emptyset$, $x.y$, $x \in y$, $x \in^* y$, $x \setminus \{y\}$
  - extensionality axiom
- Kirby 2009 (classical)
  - $\emptyset$, $x.y$
  - membership defined
  - no extensionality axiom

Our Axiomatisation of HF Sets Agrees with Kirby's

- X : Type, $\emptyset : X$, $. : X \to X \to X$
- $\forall p.\ p\emptyset \to (\forall xy. px \to py \to p(x.y)) \to \forall x. px$
- $\emptyset \neq x.y$

- $x.x.y = x.y$ *cancel*
- $x.y.z = y.x.z$ *swap*
- $x \in y.z \to x = y \lor x \in z$ *membership*

where

- $x \in y := (x.y = y)$
- $p : X \to$ Type

Main Contributations

- Minimal constructive axiomatization
- Constructive proofs of extensionality and decidability
- Construction of operations for transitive closure and cardinality
- Unique model property (categoricity)
- Everything in constructive type theory
- Formalisation in Coq

Extensionality Shown Together with Decidability

1. $x \subseteq y$ and $y \subseteq x$ are decidable
2. $x \in y$ and $y \in x$ are decidable
3. $x \subseteq y \to y \subseteq x \to x = y$
4. $x = y$ is decidable

Proof by nested HF induction on x and y using several lemmas:

1. $\emptyset \subseteq x$ and $x \subseteq \emptyset$ and $x \in \emptyset$ and $x = \emptyset$ are decidable
2. If $x = a$ and $x \in y$ are decidable, $x \in a.y$ is decidable
3. If $a \in y$ and $x \subseteq y$ are decidable, $a.x \subseteq y$ is decidable
4. $\emptyset \in x$ is decidable
5. $a \in x \to \Sigma u.\ x = a.u \land a \notin u$
   provided $a \in z$ and $a = z$ are decidable for all z

Lemmas 4 and 5 follow by HF induction on x.

Partition Operator

$$\forall x.\ x = \emptyset + \Sigma a y.\ x = a.y \land a \notin y$$

Can be constructed with HF induction on x
using decidability of membership and equality

Construction of Union $x \cup y$

- Recursive specification

$$\emptyset \cup y = y$$
$$(a.x) \cup y = a.(x.y)$$

- Extensional specification

$$z \in x \cup y \leftrightarrow z \in x \vee z \in y$$

- Both have unique solution
- Recall: Axiomatisation doesn't provide recursor
- Both are satisfied by unique function of type

$$\forall xy\ \Sigma u\ \forall z.\ z \in u \leftrightarrow z \in x \vee z \in y$$

obtainable with HF induction on x following recursive specification

Naive Recursor Dosn't Exist

$$f\emptyset := \emptyset$$
$$f(a.x) := a$$

If f exists, all sets are equal: $a = f(a.b.\emptyset) = f(b.a.\emptyset) = b$

Other Set Operations

- big union
- power set
- separation
- replacement
- transitive closure

can be constructed similar to binary union

Cardinality

- Ordinals

$$\frac{}{\mathscr{O}\,\emptyset} \qquad\qquad \frac{\mathscr{O}x}{\mathscr{O}(x.x)}$$

- Equipotence

$$\frac{}{\emptyset \sim \emptyset} \qquad\qquad \frac{a \notin x \qquad b \notin y \qquad x \sim y}{a.x \sim b.y}$$

- Cardinality relation

$$\frac{}{C\emptyset\emptyset} \qquad\qquad \frac{a \notin x \qquad Cx\alpha}{C(a.x)(\alpha.\alpha)}$$

- Cardinality function can be obtained from cardinality relation
- Subtype of ordinals yields model of Peano axioms

Categoricity

Let X and Y be HF structures.
Construct an isomorphism between X and Y as follows:

- Define inductive predicate $R : X \to Y \to \mathrm{Prop}$

$$\frac{}{R\emptyset\emptyset} \qquad\qquad \frac{Rab \qquad Rxy}{R(a.x)(b.y)}$$

- R is total
- R is functional
  - follows with $\in$-induction, extensionality, and $Rxy \to a \in x \to \exists b.\ b \in y \wedge Rab$
- R is symmetric
- R yields isomorphism between X and Y

Two Model Constructions

1. HF sets as numbers (Ackermann's encoding)
2. Quotient of binary tree type
   - $s, t, u ::= \emptyset \mid s.t$
   - $s.s.t \approx s.t$ *cancel*
   - $s.t.u \approx t.s.u$ *swap*
   - Quotient obtained as subtype of lexically sorted trees

$$\frac{}{\emptyset < s.t} \qquad \frac{s < s'}{s.t < s'.t'} \qquad \frac{t < t'}{s.t < s.t'}$$

   - Insertion sort provides normalizer for $s \approx t$

Formalisation in Coq

- 2000 lines of Coq
- Tactic-based automation is essential for simple facts about sets
- Coq proofs agree with mathematical proofs
- Impredicative Prop (probably not essential)
- Inductive types only needed for model construction

Future Work

- Dependently typed recursor
- HF as least fixed point of finite sets: $\mathsf{HF} := \mathsf{finset}\,(\mathsf{HF})$
- Non-wellfounded sets