Formalization of finite sets in Lean

Johannes Tantow

July 28, 2023

What is a set ?

Zermelo-Fraenkel set theory:

1. Axiom of Extensionality: $\forall x, y.(\forall z.(z \in x \leftrightarrow z \in y) \to x = y)$
2. Axiom of Regularity $\forall x.(x \neq \emptyset \to \exists y. y \in x \wedge y \cap x = \emptyset)$
3. Axiom of Empty Set: $\exists y.\forall x : \neg x \in y$
4. ...

What is a set ?

Zermelo-Fraenkel set theory:

1. Axiom of Extensionality: $\forall x, y.(\forall z.(z \in x \leftrightarrow z \in y) \rightarrow x = y)$
2. Axiom of Regularity $\forall x.(x \neq \emptyset \rightarrow \exists y. y \in x \wedge y \cap x = \emptyset)$
3. Axiom of Empty Set: $\exists y. \forall x : \neg x \in y$
4. ...

Lean: $(s : \text{set } A) = A \rightarrow \text{Prop}$

What is a finite set ?

1. there is a list containing all elements
2. there is a tree containing all elements
3. there exists some $N \in \mathbb{N}$ such that every list of length at least N contains duplicates
4. there is no surjection from this set to the natural numbers
5. ...

Current state

Finite by construction

Axioms and Functions

```
axiom union_comm (X Y: KSet A): X ∪ Y = Y ∪ X

noncomputable def first{A:Type} [nonempty A]:
KSet A -> A
 | KSet.empty := classical.choice  _inst_1
 | {a} := a
 | (X ∪ Y) := first (X)
```

Axioms don't care about preservation by functions

One million dollar

```
    theorem p_eq_np (l: language) (l_np: l ∈ NP):
l ∈ P :=
    begin
        exfalso,
        let X:= {1} ∪ {2},
        have first1: first (X) = 1,
        dunfold first,
        refl,
        have first2: ¬  first(X) = 1,
        have X2: X = {2} ∪ {1} by union_comm,
        rw X2,
        simp[first],
        exact absurd first1 first2,
    end
```

Quotient

Definition

Let A : Type and $R : A \times A \to$ Prop be an equivalence relation. Then A/R, the quotient of A by R, is a type.

Tree equality

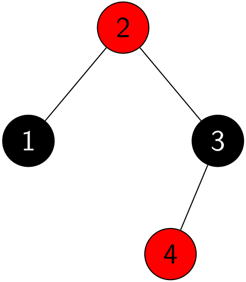

Figure: flatten T = [1,2,3,4]

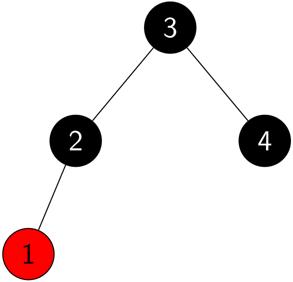

Figure: flatten T = [1,2,3,4]

Use a quotient based on flatten.

Properties of flatten

Lemma(Extensionality)

$flatten(T1) = flatten(T2) \leftrightarrow \forall x, x \in T1 \leftrightarrow x \in T2$

Proof

Idea:

- two list are equal iff they are both sorted and permutations of each other
- Ordered trees are sorted
- Ordered trees are duplicate free: permutations

Lemma

size(T) = len(flatten(T))

Don't do it twice

Mathlib data.set

```
theorem mem_inter_iff (x : α) (a b : set α) :
    x ∈ a ∩ b ↔ (x ∈ a ∧ x ∈ b)
```

Kuratowski sets

```
    lemma in_intersection_iff_in_both {A:\text{{
Type u}}} [decidable_eq A] (X Y: Kuratowski A) (a
:A): kuratowski_member_prop a (X ∩ Y) =
    (kuratowski_member_prop a X
    ∧ kuratowski_member_prop a Y)
```

Finite by proof II

Definition

A finite set S_f is a pair of a set S and a proof of its finiteness.

Finite by proof II

Definition

A finite set S_f is a pair of a set S and a proof of its finiteness.

Examples

1. Bijection finite:
   $\exists (n : \mathbb{N}), (f : \mathbb{N} \to A).\ \text{set.bij\_on}\ f\ \{x \in \mathbb{N} \mid x < n\}\ S$
2. Surjection finite:
   $\exists (n : \mathbb{N}), (f : \mathbb{N} \to A).\ \text{set.surj\_on}\ f\ \{x \in \mathbb{N} \mid x < n\}\ S$
3. Dedekind finite: $\forall (S' \subsetneq S), (f : A \to A).\ \neg \text{set.bij\_on}\ f\ S'\ S$

Size

```
def is_finite {A: Type} (S: set A): Prop :=
    ∃ (n:ℕ) (f: ℕ → A),
set.bij_on f (set_of (λ (a:ℕ), a < n)) S

noncomputable def size (s: set A) (fin:
is_finite s): ℕ
   := classical.some fin
```

Noncomputable size

Let M be a Turing-machine