Simplistic ZFC Formalization

Sebastian Ullrich

April 19, 2014

Contents

1 Zermelo-Fraenkel Axiom System **2**

2 Lemmas on Unions and Intersections **4**

3 Ordered Pairs **4**

theory *ZFC*
imports *HOL*
begin

typedecl *set*

axiomatization
  $member :: set \Rightarrow set \Rightarrow bool$

notation
  member $(op :)$ **and**
  member $((\text{-}/ : \text{-}) [51, 51] 50)$

abbreviation *not-member* **where**
  $not\text{-}member\ x\ A \equiv {}^{\sim}\ (x : A)$ — non-membership

notation
  not-member $(op\ {}^{\sim}:)$ **and**
  not-member $((\text{-}/ {}^{\sim}: \text{-}) [51, 51] 50)$

notation (*xsymbols*)
  member $(op \in)$ **and**
  member $((\text{-}/ \in \text{-}) [51, 51] 50)$ **and**
  not-member $(op \notin)$ **and**
  not-member $((\text{-}/ \notin \text{-}) [51, 51] 50)$

1 Zermelo-Fraenkel Axiom System

axiomatization where
  extensionality: $\forall z. (z \in x \longleftrightarrow z \in y) \Longrightarrow x = y$ **and**
  foundation: $\exists y. y \in x \Longrightarrow \exists y. y \in x \wedge (\forall z. \neg(z \in x \wedge z \in y))$ **and**
  subset-set: $\exists y. \forall z. z \in y \longleftrightarrow z \in x \wedge P z$ **and**
  empty-set: $\exists y. \forall x. x \notin y$ **and**
  pair-set: $\exists y. \forall x. x \in y \longleftrightarrow x = z_1 \vee x = z_2$ **and**
  power-set: $\exists y. \forall z. z \in y \longleftrightarrow (\forall u. u \in z \longrightarrow u \in x)$ **and**
  sum-set: $\exists y. \forall z. z \in y \longleftrightarrow (\exists u. z \in u \wedge u \in x)$

definition *empty* :: *set* $(\{\})$ **where**
  $empty \equiv THE\ y. \forall x. x \notin y$

axiomatization where
  infinity: $\exists w. \{\} \in w \wedge (\forall x. x \in w \longrightarrow (\exists z. z \in w \wedge (\forall u. u \in z \longleftrightarrow u \in x \vee u = x)))$ **and**
  replacement: $P\ x\ y \wedge P\ x\ z \longrightarrow y = z \Longrightarrow \exists u. (\forall w_1. w_1 \in u \longleftrightarrow (\exists w_2. w_2 \in a \wedge P\ w_2\ w_1))$

lemma *empty*[*simp*]: $x \notin empty$

proof−
  have $\exists!y.\ \forall x.\ x \notin y$
  proof (*rule ex-ex1I*)
    fix $y\ y'$
    assume $\forall x.\ x \notin y\ \forall x.\ x \notin y'$
    thus $y = y'$ **by** −(*rule extensionality*, *simp*)
  qed (*rule empty-set*)
  hence $\forall x.\ x \notin \{\}$
    unfolding *empty-def*
    by (*rule theI'*)
  thus *?thesis* **..**
qed

Let's try to generalize that for the other introduction axioms.

lemma *exAxiomD1*:
  assumes $\exists y.\ \forall x.\ x \in y \longleftrightarrow P\ x$
  shows $\exists!y.\ \forall x.\ x \in y \longleftrightarrow P\ x$
using *assms*
by (*auto intro*:*extensionality*)

lemma *exAxiomD2*:
  assumes $\exists y.\ \forall x.\ x \in y \longleftrightarrow P\ x$
  shows $x \in (THE\ y.\ \forall x.\ x \in y \longleftrightarrow P\ x) \longleftrightarrow P\ x$
apply (*rule spec*[*of - x*])
by (*rule theI'*[*OF assms*[*THEN exAxiomD1*]])

lemma *exAxiomD3*:
  assumes $\exists y.\ \forall x.\ x \in y \longleftrightarrow P\ x\ \ x \in (THE\ y.\ \forall x.\ x \in y \longleftrightarrow P\ x)$
  shows $P\ x$
using *assms exAxiomD2*
by *auto*

lemma *empty'*: $x \notin empty$
apply (*rule exAxiomD2*[*of* λ-. *False*, *simplified*, *folded empty-def*])
by (*rule empty-set*)

definition *pair* :: *set* ⇒ *set* ⇒ *set* ({-, -}) **where**
  $pair\ z_1\ z_2 \equiv THE\ y.\ \forall x.\ x \in y \longleftrightarrow x = z_1 \lor x = z_2$

definition *singleton* :: *set* ⇒ *set* ({-}) **where**
  $singleton\ x \equiv \{x,\ x\}$

definition *sum* :: *set* ⇒ *set* **where**
  $sum\ x \equiv THE\ y.\ \forall z.\ z \in y \longleftrightarrow (\exists u.\ z \in u \land u \in x)$

definition *subset* :: (*set* ⇒ *bool*) ⇒ *set* ⇒ *set* **where**
  $subset\ P\ x \equiv THE\ y.\ \forall z.\ z \in y \longleftrightarrow z \in x \land P\ z$

syntax
  subset :: *pttrn* => *set* ⇒ *bool* => *set* ((*1*{- ∈ -./ -}))
translations
  {z ∈ x. P} == *subset* (%z. P) x

lemma *pair*[*simp*]: $x \in \{z_1, z_2\} \longleftrightarrow x = z_1 \vee x = z_2$
by (*rule exAxiomD2*[*of* $\lambda x.\ x = z_1 \vee x = z_2$, *simplified*, *folded pair-def*]) (*rule pair-set*)

lemma *singleton*[*simp*]: $x \in \{y\} \longleftrightarrow x = y$
by −(*unfold singleton-def*, *simp*)

lemma *sum*[*simp*]: $z \in$ *sum* $x \longleftrightarrow (\exists u.\ z \in u \wedge u \in x)$
by (*rule exAxiomD2*[*of* $\lambda z.\ \exists u.\ z \in u \wedge u \in x$, *simplified*, *folded sum-def*]) (*rule sum-set*)

lemma *subset*[*simp*]: $z \in \{z \in x.\ P\ z\} \longleftrightarrow z \in x \wedge P\ z$
by (*rule exAxiomD2*[*of* $\lambda z.\ z \in x \wedge P\ z$, *simplified*, *folded subset-def*]) (*rule subset-set*)

2 Lemmas on Unions and Intersections

definition *union* :: *set* ⇒ *set* ⇒ *set* (**infixl** ∪ *65*) **where**
  union x y ≡ *sum* (*pair* x y)

lemma *union*[*simp*]: $z \in a \cup b \longleftrightarrow z \in a \vee z \in b$
by (*auto simp*:*union-def*)

lemma *union-script*: $\exists y.\ \forall z.\ z \in y \longleftrightarrow z \in a \vee z \in b$
by (*rule exI*[*of* - $a \cup b$]) *simp*

definition *intersect* :: *set* ⇒ *set* ⇒ *set* (**infixl** ∩ *70*) **where**
  $a \cap b \equiv \{x \in a.\ x \in b\}$

lemma *intersect*[*simp*]: $z \in a \cap b \longleftrightarrow z \in a \wedge z \in b$
by (*simp add*:*intersect-def*)

lemma *intersect-script*: $\exists y.\ \forall z.\ z \in y \longleftrightarrow z \in a \wedge z \in b$

by (*rule subset-set*)

3 Ordered Pairs

definition *ordered-pair* :: *set* ⇒ *set* ⇒ *set* (⟨-,-⟩) **where**
  $\langle a,b\rangle \equiv \{\{a\}, \{a,b\}\}$

lemma *intersect-singleton*[*simp*]: $x \cap \{y\} =$ (*if* $y \in x$ *then* $\{y\}$ *else* $\{\}$)
by (*auto intro*:*extensionality*)

lemma *empty-singleton-neq*[*simp*]: $\{x\} \neq \{\}$
proof
  assume *assm*: $\{x\} = \{\}$
  have $x \notin \{\}$ **by** *simp*
  with *assm* **have** $x \notin \{x\}$ **by** *simp*
  thus *False* **by** *simp*
qed

lemma *singleton-eqD*[*dest*!]: $\{x\} = \{y\} \Longrightarrow x = y$
by (*drule arg-cong*[*of - -*$\lambda z.\ y \in z$]) *simp*

lemma *singleton-pair-eqD*[*dest*!]:
  assumes $\{x\} = \{y,\ z\}$
  shows $x = y \wedge y = z$
proof−
  from *assms* **have** $y \in \{x\} \longleftrightarrow y \in \{y,\ z\}$ **by** *simp*
  hence $x = y$ **by** *simp*
  from *assms* **have** $z \in \{x\} \longleftrightarrow z \in \{y,\ z\}$ **by** *simp*
  hence $x = z$ **by** *simp*
  with ‹$x = y$› **show** *?thesis* **by** *simp*
qed

lemma *singleton-pair-eqD'*[*dest*!]:
  assumes $\{y,\ z\} = \{x\}$
  shows $x = y \wedge y = z$
using *assms*[*symmetric*] **by** (*rule singleton-pair-eqD*)

lemma *pair-singleton*[*simp*]: $\{x,\ x\} = \{x\}$
unfolding *singleton-def* **..**

lemma *pair-eq-fstD*[*dest*!]:
  assumes $\{x,y\} = \{x,z\}$
  shows $y = z$
using *assms*
proof (*cases* $x = y$)
  case *False*
  from *assms* **have** $y \in \{x,y\} \longleftrightarrow y \in \{x,z\}$ **by** *simp*
  with *False* **show** *?thesis* **by** *simp*
qed *auto*

lemma *ordered-pair-eq*[*simp*]: $\langle x_1,x_2\rangle = \langle y_1,y_2\rangle \longleftrightarrow x_1 = y_1 \wedge x_2 = y_2$
proof
  assume *assm*: $\langle x_1,x_2\rangle = \langle y_1,y_2\rangle$
  hence $\{x_1\} \in \langle x_1,x_2\rangle \longleftrightarrow \{x_1\} \in \langle y_1,y_2\rangle$ **by** *simp*
  hence[*simp*]: $x_1 = y_1$ **by** (*auto simp*:*ordered-pair-def*)
  show $x_1 = y_1 \wedge x_2 = y_2$ **using** *assm*
  proof (*cases* $x_2 = x_1$)
    case *False*

from *assm* **have** $\{x_1,x_2\} \in \langle x_1,x_2\rangle \longleftrightarrow \{x_1,x_2\} \in \langle y_1,y_2\rangle$ **by** *simp*
with *False* **show** *?thesis* **by** (*auto simp*:*ordered-pair-def*)
qed (*auto simp*:*ordered-pair-def*)
qed *simp*

end