

Zermelo-Fraenkel set theory in HOL (part 1)

Roger Bishop Jones

ICL Defence Systems

ABSTRACT

This document consists of a formal axiomatisation in HOL of Zermelo-Fraenkel set theory.

1. INTRODUCTION

This is a minimally modified transcription into HOL of the formalisation of ZF found in *The Logical Foundations of Mathematics* by William S. Hatcher (Pergammon 1982 ISBN 0-08-025800). The fact that this is an axiomatisation in higher order logic rather than first order logic has some consequences, but doesn't cause any very major changes.

I mention here just two points:

- ⇔ In HOL logical equivalence is interchangeable with equality over type bool (i.e. $\vdash \forall(a:\text{bool})(b:\text{bool}). (a=b) \Leftrightarrow (a \Leftrightarrow b)$). Since the rewrite facilities depend upon equality it is easier to dispense with equivalence and use equality instead, which I have done here.
- = Equality is in the basic HOL system, and so we do not need to define it as Hatcher does. However, Hatcher has a curious division of labour between his definition of equality and his axiom of extensionality (ZF1). It looks as though his definition of equality in fact gives half of the necessary facts for equality together with extensionality, and his axiom of extensionality is just the other half of the equality axiom. The fact that equality is built into HOL means that Hatcher's ZF1 is provable in HOL, but his definition of equality isn't!

2. ZERMELO-FRAENKEL SET THEORY

2.1. Formalisation of ZF

First we start a new theory for ZF and declare the new type SET.

```

ML
new_theory 'zf120';;
%
new_parent 'infra';;
loadf 'infra';;
%
new_type 0 'SET';;
loadf '/escher/usr2/projects/infra/pholfiles/TAUT';;

```

2.1.1. Membership and Extensionality

Next we introduce the membership predicate and the axiom of extensionality. This differs from Hatcher's presentation since the equality rules are already built into HOL, whereas Hatcher's definition of equality is in fact the rule of extensionality.

```

ML
new_infix ('∈',":SET → SET → bool");;

let EXT = new_axiom ('EXT',"
  ∀ (x:SET)(y:SET)•(∀ (z:SET)• z ∈ x = z ∈ y) ⇒(x = y)
");;

```

Hatchers theorem 1 is trivially proved in HOL:

```

ML
set_goal([], "
  ∀(x1:SET)(x2:SET)•(x1 = x2) ⇒ ∀(x⊗:SET)• x1 ∈ x⊗ = x2 ∈ x⊗
");;
expand (REPEAT STRIP_TAC THEN ASM_REWRITE_TAC []);;
let ZF_thm1 = save_top_thm 'ZF_thm1';;

```

The following rewrite rule will be useful for proving equality of sets.

```

ML
set_goal([], "
  ∀(x:SET)(y:SET)• (x = y) = (∀(z:SET)• z ∈ x = z ∈ y)
");;
expand (EVERY [REPEAT STRIP_TAC;EQ_TAC;REPEAT STRIP_TAC;ASM_REWRITE_TAC[]]);;
expand (IMP_RES_TAC (SPEC_ALL EXT));;
let ZF_le1 = save_top_thm 'ZF_le1';;

```

Theorem 2 is too trivial to be worth saving:

```

ML
let ZF_thm2 = REFL "x:SET";;

```

Now we define non-membership and inequality.

```

ML
let  $\notin$ _DEF = new_infix_definition(' $\notin$ _DEF',"
   $\notin$  (x:SET)(y:SET) =  $\neg$ (x  $\in$  y)
");;
let NE_DEF = new_infix_definition('NE_DEF',"
   $\neq$  (x:*)(y:*) =  $\neg$ (x = y)
");;

```

2.1.2. Separation

The constant *sep* is introduced for separation.

```

ML
new_constant ('sep',":SET  $\rightarrow$  (SET  $\rightarrow$  bool)  $\rightarrow$  SET");;
let ZF2 = new_axiom('ZF2',"
   $\forall$ (A:SET $\rightarrow$ bool)(z:SET)(x:SET)• x  $\in$  (sep z A) = x  $\in$  z  $\wedge$  A x
");;

```

2.1.3. The Empty Set

```

ML
new_constant (' $\emptyset$ ',":SET");;

let ZF3 = new_axiom ('ZF3',"
  ( $\emptyset$ :SET) = sep  $\emptyset$   $\lambda$  x1:SET• F
");;

```

We now prove that nothing is a member of the empty set.

```

ML
set_goal([], "
   $\forall$ (x1:SET)• x1  $\notin$   $\emptyset$ 
");;
expand (PURE_REWRITE_TAC [ $\notin$ _DEF]);;
expand (PURE_ONCE_REWRITE_TAC [ZF3]);;
expand (REWRITE_TAC [ZF2]);;
let ZF_thm3 = save_top_thm 'ZF_thm3';;

```

The following form is more convenient.

```

ML
let ZF_le2 = save_thm ('ZF_le2', REWRITE_RULE [∈_DEF] ZF_thm3);

```

The empty set is unique.

```

ML
set_goal([], "
  ∀(x1:SET)•(∀(x2:SET)•x2 ∈ x1) = (∅ = x1)
");;
expand (STRIP_TAC THEN EQ_TAC);;
expand (REWRITE_TAC [∈_DEF]);;
expand STRIP_TAC;;
expand (RMP_TAC "(∀z:SET• z ∈ ∅ = z ∈ x1:SET)");;
expand (ACCEPT_TAC (SPECL ["∅";"x1:SET"] EXT));;
expand (ONCE_REWRITE_TAC [ZF3]);;
expand (ASM_REWRITE_TAC [ZF2]);;
expand STRIP_TAC;;
expand (IMP_RES_TAC (SPECL ["∅";"x1:SET"] (INST_TYPE [":SET",":*"] EQ_SYM)));;
expand (ONCE_ASM_REWRITE_TAC []);;
expand (REWRITE_TAC [ZF_thm3]);;
let ZF_thm4 = save_top_thm 'ZF_thm4';;

```

A helpful lemma for proving non-emptiness:

```

ML
set_goal([], "
  ∀(x1:SET)(x2:SET)• x2 ∈ x1 ⇒ x1 ≠ ∅
");;
expand (REWRITE_TAC [NE_DEF]);;
expand (TAUT_REWRITE_TAC "a ⇒ ¬b = b ⇒ ¬a");;
expand STRIP_TAC;;
expand STRIP_TAC;;
expand STRIP_TAC;;
expand (ASM_REWRITE_TAC[ZF_le2]);;
let ZF_le3 = save_top_thm 'ZF_le3';;

```

2.1.4. Power Sets

The power set is defined using the subset relationship:

```

ML
let  $\subseteq\_DEF = new\_infix\_definition('subseteq\_DEF',
    (\subseteq:SET \rightarrow SET \rightarrow bool) a b = \forall c:SET. c \in a \Rightarrow c \in b$ 
");;

```

We prove some obvious properties of \subseteq :

```

ML
set_goal([], "
     $\forall (x^1:SET)(x^2:SET). (x^1 \subseteq x^2 \wedge x^2 \subseteq x^1) \Rightarrow (x^1 = x^2)$ 
");;
expand (REWRITE_TAC [ $\subseteq\_DEF$ ]);;
expand (REPEAT STRIP_TAC);;
expand (TMP_TAC (SPEC ["x^1:SET"; "x^2:SET"] EXT));;
expand (STRIP_TAC THEN EQ_TAC THEN ASM_REWRITE_TAC []);;
let ZF_thm5 = save_top_thm 'ZF_thm5';;

```

```

ML
set_goal([], "
     $\forall (x^1:SET). x^1 \subseteq x^1$ 
");;
expand (REWRITE_TAC [ $\subseteq\_DEF$ ]);;
let ZF_thm6 = save_top_thm 'ZF_thm6';;

```

```

ML
set_goal([], "
     $\forall (x^1:SET)(x^2:SET)(x^\circ:SET). (x^1 \subseteq x^2 \wedge x^2 \subseteq x^\circ) \Rightarrow (x^1 \subseteq x^\circ)$ 
");;
expand (REWRITE_TAC [ $\subseteq\_DEF$ ]);;
expand (REPEAT STRIP_TAC);;
expand (RES_TAC THEN RES_TAC);;
let ZF_thm7 = save_top_thm 'ZF_thm7';;

```

```

ML
set_goal([], "
     $\forall (x^1:SET). \emptyset \subseteq x^1$ 
");;
expand (REWRITE_TAC [ $\subseteq\_DEF$ ]);;
expand (ONCE_REWRITE_TAC [ZF3]);;
expand (REWRITE_TAC [ZF2]);;
let ZF_thm8 = save_top_thm 'ZF_thm8';;

```

```

ML
new_constant('P', ":SET → SET");;
let ZF4 = new_axiom('ZF4', "
  ∀(y:SET)(x:SET)• x ∈ (P y) = x ⊆ y
");;

```

2.1.5. Pairs

```

ML
new_constant('pair', ":SET → SET → SET");;
let ZF5 = new_axiom('ZF5', "
  ∀(y:SET)(z:SET)(x:SET)• x ∈ (pair y z) = (x=y) ∨ (x=z)
");;

```

```

ML
set_goal([], "
  ∀(x:SET)(y:SET)• x ∈ (pair x y) ∧ y ∈ (pair x y)
");;
expand (REPEAT STRIP_TAC THEN REWRITE_TAC [ZF5]);;
let ZF_le4 = save_top_thm 'ZF_le4';;

```

```

ML
let unit_DEF = new_definition('unit_DEF', "
  (unit:SET → SET) x = pair x x
");;

```

```

ML
set_goal([], "
  ∀x1:SET• x1 ∈ unit x2 = (x1 = x2)
");;
expand (STRIP_TAC THEN REWRITE_TAC [unit_DEF;ZF5;ZF_le4]);;
let ZF_thm9 = save_top_thm 'ZF_thm9';;

```

```

ML
set_goal([], "
   $\forall(x:\text{SET})(y:\text{SET}) \bullet (\text{unit } x = \text{unit } y) = (x = y)$ 
");;
expand (REPEAT STRIP_TAC THEN EQ_TAC);;
expand (REWRITE_TAC [SPECL["unit x";"unit y"] ZF_1e1]);;
expand (REWRITE_TAC [ZF_thm9]);;
expand STRIP_TAC;;
expand (ASSUME_TAC (REFL "x:SET"));;
expand (DEF_RES_TAC (ASSUMP " $\forall z:\text{SET} \bullet (z = x) = (z = y)$ "));;
expand (STRIP_TAC THEN ASM_REWRITE_TAC[]);;
let ZF_1e13 = save_top_thm 'ZF_1e13';;

```

```

ML
let  $\mapsto$ _DEF = new_infix_definition (' $\mapsto$ _DEF', "
   $(\mapsto:\text{SET} \rightarrow \text{SET} \rightarrow \text{SET})$  a b = pair (unit a) (pair a b)
");;

```

```

ML
set_goal([], "
   $\forall(x:\text{SET})(y:\text{SET}) \bullet (\text{unit } x) \in (x \mapsto y) \wedge (\text{pair } x y) \in (x \mapsto y)$ 
");;
expand (REPEAT STRIP_TAC THEN REWRITE_TAC [ $\mapsto$ _DEF;ZF_1e4]);;
let ZF_1e5 = save_top_thm 'ZF_1e5';;

```

2.1.6. Union and Intersection

```

ML
new_constant('U',":SET  $\rightarrow$  SET");;
let ZF6 = new_axiom('ZF6', "
   $\forall(y:\text{SET})(x:\text{SET}) \bullet x \in (\cup y) = \exists(z:\text{SET}) \bullet z \in y \wedge x \in z$ 
");;

```

```

ML
let  $\cap\_DEF = new\_definition(' \cap\_DEF', "$ 
     $(\cap : SET \rightarrow SET) x = sep (\cup x) \lambda y : SET. \forall (z : SET). z \in x \Rightarrow y \in z$ 
 $"$ );;

let  $\cup\_DEF = new\_infix\_definition(' \cup\_DEF', "$ 
     $\cup (x : SET)(y : SET) = \cup (pair\ x\ y)$ 
 $"$ );;

let  $\cap\_DEF = new\_infix\_definition(' \cap\_DEF', "$ 
     $\cap (x : SET)(y : SET) = \cap (pair\ x\ y)$ 
 $"$ );;

```

```

ML
set_goal([], " $\forall (x^1 : SET)(x^2 : SET)(x^\circledast : SET). x^1 \in (x^2 \cup x^\circledast) = x^1 \in x^2 \vee x^1 \in x^\circledast$ 
 $"$ );;
expand(REWRITE_TAC [ $\cup\_DEF$ ;ZF6;ZF5]);;
expand (REPEAT STRIP_TAC);;
expand EQ_TAC;;
expand (EVERY [
    STRIP_TAC;
    UNDISCH_TAC " $x^1 \in z$ ";
    ASM_REWRITE_TAC [];
    STRIP_TAC;
    ASM_REWRITE_TAC []]);;
expand STRIP_TAC;;
expand (EXISTS_TAC " $x^2 : SET$ ");;
expand (ASM_REWRITE_TAC []);;
expand (EXISTS_TAC " $x^\circledast : SET$ ");;
expand (ASM_REWRITE_TAC []);;
let ZF_thm10 = save_top_thm 'ZF_thm10';;

```



```

ML
set_goal([], "
   $\forall(x^1:\text{SET})(x^2:\text{SET})(x^\circ:\text{SET}) \bullet x^1 \in (x^2 \cap x^\circ) = x^1 \in x^2 \wedge x^1 \in x^\circ$ 
");;
expand(REWRITE_TAC [ $\cap$ _DEF; $\cap$ _DEF;ZF2;ZF5;ZF6]);;
expand BETA_TAC;;
expand (REPEAT STRIP_TAC);;
expand EQ_TAC;;
let rr t = REWRITE_RULE []
  (SPEC t (ASSUME " $\forall z:\text{SET} \bullet (z = x^2) \vee (z = x^\circ) \Rightarrow x^1 \in z$ "))
in expand (REPEAT STRIP_TAC THEN REWRITE_TAC (map rr ["x^2:SET";"x^\circ:SET"]));;
expand (REPEAT STRIP_TAC THEN ASM_REWRITE_TAC[]);;
expand (EXISTS_TAC "x^2:SET" THEN ASM_REWRITE_TAC[]);;
let ZF_thm11 = save_top_thm 'ZF_thm11';;

```

2.1.7. Natural Numbers

```

ML
let suc_DEF = new_definition('suc_DEF', "
   $\text{suc } (x:\text{SET}) = x \cup (\text{unit } x)$ 
");;

let Nat_DEF = new_prim_rec_definition('Nat', "
   $(\text{Nat } 0 = \emptyset) \wedge$ 
   $(\text{Nat } (\text{SUC } n) = \text{suc } (\text{Nat } n))$ 
");;

```

2.1.8. The Axiom of Regularity

The following axiom Hatcher calls the axiom of regularity, sometimes it is called the axiom of well foundedness.

```

ML
let ZF7 = new_axiom ('ZF7', "
   $\forall(x:\text{SET}) \bullet x \neq \emptyset \Rightarrow \exists(y:\text{SET}) \bullet y \in x \wedge (y \cap x = \emptyset)$ 
");;

```

We now prove some consequences of well foundedness.

```

ML
set_goal([], "
   $\forall x^1: \text{SET} \bullet x^1 \notin x^1$ 
");;
expand (STRIP_TAC THEN REWRITE_TAC [notin_DEF]);;
expand STRIP_TAC;;
expand (MP_TAC ZF7);;
expand (RMP_TAC " $\exists x \bullet x \neq \emptyset \wedge \neg(\exists y \bullet y \in x \wedge (y \cap x = \emptyset))$ ");;
expand (REWRITE_TAC[NOT_FORALL_CONV " $\neg(\forall x \bullet x \neq \emptyset \Rightarrow (\exists y \bullet y \in x \wedge (y \cap x = \emptyset)))$ "]);;
expand (RMP_TAC "
   $\forall (x: \text{SET}) \bullet x \neq \emptyset \wedge \neg(\exists y \bullet y \in x \wedge (y \cap x = \emptyset))$ 
  =  $\neg(x \neq \emptyset \Rightarrow (\exists y \bullet y \in x \wedge (y \cap x = \emptyset)))$ ");;
expand STRIP_TAC;;
expand (ASM_REWRITE_TAC[]);;
expand (STRIP_TAC THEN TAUT_TAC);;
expand (EXISTS_TAC "unit x1:SET");;
expand (REWRITE_TAC [NE_DEF;ZF_thm9;NOT_EXISTS_CONV " $\neg(\exists y \bullet (y = x^1) \wedge (y \cap (\text{unit } x^1) = \emptyset))$ "]);;

```

```

ML
expand (STRIP_TAC);;
expand (RMP_TAC " $x^1 \in \text{unit } x^1$ ");;
expand (ASM_REWRITE_TAC[REWRITE_RULE [notin_DEF] ZF_thm3]);;
expand (REWRITE_TAC [ZF_thm9;ZF_le1]);;
expand (REWRITE_TAC
  [NOT_FORALL_CONV " $\neg(\forall z: \text{SET} \bullet (\forall z': \text{SET} \bullet z' \in z = z' \in x^1) = z \in \emptyset)$ "]);;
expand (EXISTS_TAC " $x^1: \text{SET}$ ");;
expand (REWRITE_TAC[REWRITE_RULE [notin_DEF]ZF_thm3]);;
expand (REWRITE_TAC [ZF_thm9]);;

```

```

ML
expand STRIP_TAC;;
expand (RMP_TAC " $(y = x^1) \Rightarrow \neg (y \cap (\text{unit } x^1) = \emptyset)$ ");;
expand TAUT_TAC;;
expand STRIP_TAC;;
expand (ASM_REWRITE_TAC[ZF_le1;ZF_thm11]);;
expand (REWRITE_TAC [REWRITE_RULE [notin_DEF]ZF_thm3]);;
expand (ASM_REWRITE_TAC[NOT_FORALL_CONV " $\neg(\forall z \bullet \neg(z \in x^1 \wedge z \in (\text{unit } x^1)))$ "]);;
expand (EXISTS_TAC " $x^1: \text{SET}$ ");;
expand (ASM_REWRITE_TAC [ZF_thm9]);;
let ZF_thm12 = save_top_thm 'ZF_thm12';;

```

```

ML
set_goal([], "¬ $∀ (pf:*→bool) = $∃ (λ x:*• ¬ pf x)");;
expand (RMP_TAC "(pf:*→bool) = λ x:*• pf x");;
expand (STRIP_TAC THEN ONCE_ASM_REWRITE_TAC[]);;
expand BETA_TAC;;
expand (REWRITE_TAC [NOT_FORALL_CONV "¬(∀x• pf x)"]);;
expand (REWRITE_TAC [ETA_AX]);;

```

```

ML
let NOT_FORALL = save_top_thm 'NOT_FORALL';;

let NOT_FORALL_TAC = REWRITE_TAC[NOT_FORALL] THEN BETA_TAC;;
let PURE_NOT_FORALL_TAC = PURE_REWRITE_TAC[NOT_FORALL] THEN BETA_TAC;;

```

```

ML
set_goal([], "¬ $∃ (pf:*→bool) = $∀ (λ x:*• ¬ pf x)");;
expand (RMP_TAC "(pf:*→bool) = λ x:*• pf x");;
expand (STRIP_TAC THEN ONCE_ASM_REWRITE_TAC[]);;
expand BETA_TAC;;
expand (REWRITE_TAC [NOT_EXISTS_CONV "¬(∃x• pf x)"]);;
expand (REWRITE_TAC [ETA_AX]);;

```

```

ML
let NOT_EXISTS = save_top_thm 'NOT_EXISTS';;

let NOT_EXISTS_TAC = REWRITE_TAC[NOT_EXISTS] THEN BETA_TAC;;
let PURE_NOT_EXISTS_TAC = PURE_REWRITE_TAC[NOT_EXISTS] THEN BETA_TAC;;

```

```

ML
set_goal([], "
   $\forall(x^1:\text{SET})(x^2:\text{SET}) \bullet x^1 \in x^2 \Rightarrow x^2 \notin x^1$ 
");;
expand (REWRITE_TAC[ $\notin$ _DEF; TAUT_RULE "a  $\Rightarrow$   $\neg$ b =  $\neg$ (a  $\wedge$  b)"]);;
expand (REPEAT STRIP_TAC);;
expand (MP_TAC ZF7 THEN TAUT_REWRITE_TAC "a  $\Rightarrow$  F =  $\neg$ a");;
expand NOT_FORALL_TAC;;
expand (EXISTS_TAC "pair (x1:SET) (x2:SET)");;
expand (REWRITE_TAC [NE_DEF;ZF_1e1]);;
expand (REWRITE_TAC [ZF_thm11]);;
expand (REWRITE_TAC [REWRITE_RULE[ $\notin$ _DEF]ZF_thm3]);;
expand (TAUT_REWRITE_TAC " $\neg$  ( $\neg$ a  $\Rightarrow$  b) =  $\neg$ a  $\wedge$   $\neg$ b");;
expand (STRIP_TAC THEN NOT_FORALL_TAC);;
expand (EXISTS_TAC "x1:SET");;
expand (ASM_REWRITE_TAC [ZF5]);;

```

```

ML
expand (ASM_REWRITE_TAC [ZF5]);;
expand NOT_EXISTS_TAC;;
expand (TAUT_REWRITE_TAC " $\neg$ (a  $\wedge$  b) = a  $\Rightarrow$   $\neg$ b");;
expand (EVERY [STRIP_TAC; STRIP_TAC; NOT_FORALL_TAC; ASM_REWRITE_TAC[]]);;
expand (EXISTS_TAC "x2:SET" THEN ASM_REWRITE_TAC []);;
expand (EXISTS_TAC "x1:SET" THEN ASM_REWRITE_TAC []);;
let ZF_thm13 = save_top_thm 'ZF_thm13';;

```

The following function *set* takes a list of sets and returns the set containing just those elements:

```

ML
let set = new_list_rec_definition('set',"
  (set [] =  $\emptyset$ )  $\wedge$ 
  (set (CONS h t) = (unit h)  $\cup$  (set t))
");;

```

One more consequence of well-foundedness:

```

ML
set_goal([], "
   $\forall(x^1:\text{SET})(x^2:\text{SET})(x^\circ:\text{SET}) \bullet x^1 \in x^2 \wedge x^2 \in x^\circ \Rightarrow x^\circ \notin x^1 \wedge x^\circ \neq x^1$ 
");;
expand (EVERY[
  REWRITE_TAC[ $\notin$ _DEF; NE_DEF];
  REPEAT STRIP_TAC;
  MP_TAC ZF7;
  TAUT_REWRITE_TAC "a  $\Rightarrow$  F =  $\neg$ a";
  NOT_FORALL_TAC;
  EXISTS_TAC "set[(x1:SET);(x2:SET);(x∘:SET)]";
  REWRITE_TAC [set;NE_DEF;ZF_le1;ZF_thm11;
    ZF_thm10;REWRITE_RULE[ $\notin$ _DEF]ZF_thm3;unit_DEF];
  TAUT_REWRITE_TAC " $\neg$  ( $\neg$ a  $\Rightarrow$  b) =  $\neg$ a  $\wedge$   $\neg$ b";
  STRIP_TAC ]);;

```

Giving four subgoals.

The first:

```

ML
expand (EVERY[
  NOT_FORALL_TAC;
  EXISTS_TAC "x1:SET";
  ASM_REWRITE_TAC [ZF5]]);;

```

The second:

```

ML
expand (ASM_REWRITE_TAC [ZF5]);;
expand (NOT_EXISTS_TAC);;
expand (TAUT_REWRITE_TAC " $\neg$ (a  $\wedge$  b) = a  $\Rightarrow$   $\neg$ b");;
expand (EVERY [STRIP_TAC; STRIP_TAC; NOT_FORALL_TAC; ASM_REWRITE_TAC[]]);;
expand (EXISTS_TAC "x∘:SET" THEN ASM_REWRITE_TAC []);;
expand (EXISTS_TAC "x1:SET" THEN ASM_REWRITE_TAC []);;
expand (EXISTS_TAC "x2:SET" THEN ASM_REWRITE_TAC []);;

```

The third:

```

ML
expand NOT_FORALL_TAC;;
expand (EXISTS_TAC "x1:SET");;
expand (ASM_REWRITE_TAC [ZF5]);;

```

And the fourth:

```

ML
expand (EVERY [
  RMP_TAC "F";
  REWRITE_TAC[];
  MP_TAC ZF_thm13;
  REWRITE_TAC [≠_DEF];
  NOT_FORALL_TAC;
  NOT_FORALL_TAC;
  EXISTS_TAC "x1:SET";
  EXISTS_TAC "x2:SET";
  ASM_REWRITE_TAC [];
  ACCEPT_TAC (REWRITE_RULE [ASSUME "x@ = x1"] (ASSUME "x2 ∈ x@")));;

let ZF_thm14 = save_top_thm 'ZF_thm14';;

```

2.1.9. Ordinals

Transitive sets:

```

ML
let Trans_DEF = new_definition('Trans_DEF',
  (Trans:SET → bool) x = ∀(y:SET)• y ∈ x ⇒ y ⊆ x
  ");;

```

Connected sets:

```

ML
let Con_DEF = new_definition('Con_DEF',
  (Con:SET → bool) x =
  ∀(y:SET)(z:SET)• (y ∈ x ∧ z ∈ x ∧ z ≠ y) ⇒ (z ∈ y ∨ y ∈ z)
  ");;

```

```

ML
let ZF_le6 = save_thm('ZF_le6',
  TAUT_REWRITE_RULE "a ∧ b ∧ ¬ c ⇒ d = a ∧ b ⇒ c ∨ d"
  (REWRITE_RULE [NE_DEF] Con_DEF));;

```

```

ML
let ZF_le7 = save_thm('ZF_le7',
  TAUT_REWRITE_RULE "a ∧ b ∧ ¬ c ⇒ d = a ⇒ b ⇒ c ∨ d"
  (REWRITE_RULE [NE_DEF] Con_DEF));;

```

```

ML
set_goal([], "
  Trans(x1:SET) =  $\forall(x2:SET)(x3:SET) \bullet (x2 \in x1 \wedge x3 \in x2 \Rightarrow x3 \in x1)$ 
");;
expand (REWRITE_TAC [Trans_DEF;  $\subseteq$ _DEF]);;
expand (EQ_TAC THEN REPEAT STRIP_TAC THEN RES_TAC);;
let ZF_le8 = save_top_thm 'ZF_le8';;

```

Ordinal numbers:

```

ML
let On_DEF = new_definition('On_DEF', "
  (On:SET  $\rightarrow$  bool) x = Trans x  $\wedge$  Con x
");;

```

Hatcher's exercise to prove that an ordinal is transitive under \in .

```

ML
set_goal([], "
   $\forall(x^1:SET)(x^2:SET)(x^\circ:SET)(x^\pm:SET) \bullet$ 
     $((\text{On}(x^1) \wedge x^2 \in x^1 \wedge x^\circ \in x^1 \wedge x^\pm \in x^1 \wedge x^2 \in x^\circ \wedge x^\circ \in x^\pm)$ 
       $\Rightarrow x^2 \in x^\pm)$ 
");;
expand (REWRITE_TAC [On_DEF; Con_DEF; ZF_le8; NE_DEF;  $\subseteq$ _DEF] THEN REPEAT STRIP_TAC);;

expand (LEMMA " $\neg(x^\pm \in x^2) \wedge \neg(x^\pm = x^2)$ ");;
expand RES_TAC;;
expand RES_TAC;;
expand (LEMMA " $(x^2 \in x^\circ) \wedge (x^\circ \in x^\pm)$ ");;
expand (IMP_RES_TAC (SPEC [ "x2", "x∘", "x±" ] (REWRITE_RULE [NE_DEF;  $\notin$ _DEF] ZF_thm14)));;
expand (ASM_REWRITE_TAC []);;
expand (ASM_REWRITE_TAC []);;
let ZF_le9 = save_top_thm 'ZF_le9';;

```

2.1.10. Ordinals

```

ML
set_goal([], "
 $\forall(p:SET \rightarrow bool)(q:SET \rightarrow bool) \bullet (\forall(x:SET) \bullet p\ x \Rightarrow q\ x) \Rightarrow ((\forall(x:SET) \bullet p\ x) \Rightarrow \forall(x:SET) \bullet q\ x)$ 
");;
expand (STRIP_TAC THEN STRIP_TAC THEN STRIP_TAC);;
expand FORALL_OUT_TAC;;
expand (ASM_REWRITE_TAC[]);;
let ZF_le10 = save_top_thm 'ZF_le10';;

```

The ordinals are totally ordered by \in .

```

ML
set_goal([], "
 $\forall(x^1:SET)(x^2:SET) \bullet x^1 \subseteq x^2 \wedge x^1 \neq \emptyset \wedge (\text{On } x^2)$ 
 $\Rightarrow \exists(x^\circ:SET) \bullet x^\circ \in x^1$ 
 $\wedge \forall(x^\pm:SET) \bullet x^\pm \in x^1 \Rightarrow (x^\pm = x^\circ) \vee x^\circ \in x^\pm$ 
");;
expand (REWRITE_TAC [subseteq_DEF; NE_DEF; On_DEF; ZF_le7]);;
expand (REPEAT STRIP_TAC);;
expand (MP_TAC (SPEC "x^1:SET" ZF7));;
expand (ASM_REWRITE_TAC[NE_DEF]);;
expand (TAUT_REWRITE_TAC "a  $\Rightarrow$  b =  $\neg$ b  $\Rightarrow$   $\neg$ a");;
expand NOT_EXISTS_TAC;;
expand FORALL_OUT_TAC;;
expand (TAUT_REWRITE_TAC " $\neg(a \wedge b) \Rightarrow \neg(a \wedge c) = (a \wedge c) \Rightarrow b$ ");;
expand (REWRITE_TAC [SPEC "x  $\cap$  x^1" ZF_le1]);;
expand (REWRITE_TAC [ZF_thm11; REWRITE_RULE [notin_DEF] ZF_thm3]);;
expand (TAUT_REWRITE_TAC " $\neg(a \wedge b) = b \Rightarrow \neg a$ ");;
expand (REPEAT STRIP_TAC);;
expand RES_TAC;;
expand (MP_TAC (SPECL ["x:SET"; "x^\pm:SET"] (ASSUME " $\forall y\ z \bullet$ 
 $y \in x^2 \Rightarrow$ 
 $z \in x^2 \Rightarrow$ 
 $(z = y) \vee z \in y \vee y \in z$ "))));;
expand (RMP_TAC "(x  $\in$  x^2  $\wedge$  x^\pm  $\in$  x^2  $\wedge$   $\neg(x^\pm \in x)$ ");;
expand TAUT_TAC;;
expand (ASM_REWRITE_TAC []);;

let ZF_thm15 = save_top_thm 'ZF_thm15';;

```

The empty set is an ordinal.


```

ML
let ZF_thm16 = prove_thm ('ZF_thm16',
  On  $\emptyset$ 
  ",
  REWRITE_TAC [On_DEF;Trans_DEF;Con_DEF;ZF_1e2]);

```

A subset of a connected set is connected.

```

ML
set_goal([],
   $\forall(x^1:SET)(x^2:SET) \bullet \text{Con } x^1 \wedge x^2 \subseteq x^1 \Rightarrow \text{Con } x^2$ 
  "));
expand(REWRITE_TAC [Con_DEF;NE_DEF; $\subseteq$ _DEF]);;
expand(TAUT_REWRITE_TAC "a  $\wedge$  b  $\Rightarrow$  c = b  $\Rightarrow$  a  $\Rightarrow$  c");;
expand (STRIP_TAC THEN STRIP_TAC THEN STRIP_TAC);;
expand (FORALL_OUT_TAC THEN STRIP_TAC THEN FORALL_OUT_TAC);;
expand (TAUT_REWRITE_TAC "a  $\Rightarrow$  b  $\Rightarrow$  c = b  $\Rightarrow$  a  $\Rightarrow$  c");;
expand (STRIP_TAC THEN STRIP_TAC);;
expand (RES_TAC THEN ASM_REWRITE_TAC []);;
let ZF_thm17 = save_top_thm 'ZF_thm17';;

```

Every member of an ordinal is an ordinal.

```

ML
set_goal([],
   $\forall(x^1:SET)(x^2:SET) \bullet \text{On } x^1 \wedge x^2 \in x^1 \Rightarrow \text{On } x^2$ 
  "));
expand (REWRITE_TAC [On_DEF]);;
expand (TAUT_REWRITE_TAC "a  $\Rightarrow$  b  $\wedge$  c = (a  $\Rightarrow$  b)  $\wedge$  (a  $\Rightarrow$  c)");;
expand (STRIP_TAC THEN STRIP_TAC THEN STRIP_TAC);;

```

subgoal 1

```

ML
expand (REWRITE_TAC[Trans_DEF; $\subseteq$ _DEF]);;
expand (REPEAT STRIP_TAC);;
expand RES_TAC;;
expand RES_TAC;;
expand (IMP_RES_TAC (SPECL ["c:SET";"y:SET"]
  (REWRITE_RULE [NE_DEF; $\notin$ _DEF] ZF_thm14)));;
expand (IMP_RES_TAC (REWRITE_RULE [Con_DEF;NE_DEF] (ASSUME "Con (x^1:SET)")));;
expand (IMP_RES_TAC (TAUT_RULE "x^2  $\in$  c  $\Rightarrow$   $\neg$ x^2  $\in$  c  $\Rightarrow$  c  $\in$  x^2"));;

```

subgoal 2

```

ML
expand (REWRITE_TAC [Trans_DEF]);;
expand (REPEAT STRIP_TAC);;
expand (RES_TAC THEN IMP_RES_TAC ZF_thm17);;
let ZF_thm18 = save_top_thm 'ZF_thm18';;

```

Successor:

```

ML
let Sc_DEF = new_definition('Sc_DEF',"
  (Sc:SET→bool) x = ∃(y:SET)• On y ∧ (suc y = x)
");;

```

Limit ordinals:

```

ML
let Lim_DEF = new_definition('Lim_DEF',"
  (Lim:SET→bool) x = On x ∧ x ≠ ∅ ∧ ¬(Sc x)
");;

```

Natural numbers:

```

ML
let N_DEF = new_definition('N_DEF',"
  (N:SET→bool) x = On x ∧ ((x = ∅) ∨ (Sc x)) ∧ ∀(y:SET)•(y ∈ x ⇒ (y = ∅) ∨ (Sc y))
");;

```

The empty set is a natural number.

```

ML
let ZF_thm19 = prove_thm('ZF_thm19',"
  N ∅
",
  REWRITE_TAC [N_DEF;ZF_thm16;ZF_le2]);;

```

Every non-zero Natural number is a successor.

```

ML
set_goal([], "
   $\forall(x:\text{SET}) \bullet N\ x \wedge x \neq \emptyset \Rightarrow \text{Sc}\ x$ 
");;
expand (REPEAT STRIP_TAC);;
expand (DEF_RES_TAC N_DEF);;
expand (DEF_RES_TAC NE_DEF);;
let ZF_le11 = save_top_thm 'ZF_le11';;

```

Every natural number is transitive and connected.

```

ML
set_goal([], "
   $\forall(x:\text{SET}) \bullet N\ x \Rightarrow \text{Trans}\ x \wedge \text{Con}\ x$ 
");;
expand (REWRITE_TAC[N_DEF; On_DEF]);;
expand (REPEAT STRIP_TAC THEN ASM_REWRITE_TAC []);;
let ZF_le12 = save_top_thm 'ZF_le12';;

```

Zero is not the successor of any set.

```

ML
set_goal([], "
   $\forall(x^1:\text{SET}) \bullet \emptyset \neq \text{suc}\ x^1$ 
");;
expand (REWRITE_TAC [NE_DEF; suc_DEF; ZF_le1; ZF_le2]);;
expand (STRIP_TAC THEN NOT_FORALL_TAC);;
expand (EXISTS_TAC "x1:SET");;
expand (REWRITE_TAC [ZF_thm10; ZF_thm9]);;
let ZF_thm20 = save_top_thm 'ZF_thm20';;

```

The successor of an ordinal is an ordinal.

```

ML
set_goal([], "
   $\forall(x^1:\text{SET}) \bullet \text{On } x^1 \Rightarrow \text{On } (\text{suc } x^1)$ 
");;
expand (STRIP_TAC THEN STRIP_TAC);;
expand (REWRITE_TAC [On_DEF; suc_DEF]);;
expand (DEF_RES_TAC On_DEF);;
expand (DEF_RES_TAC (REWRITE_RULE [ $\subseteq$ _DEF] Trans_DEF));;
expand STRIP_TAC;;
expand (REWRITE_TAC [Trans_DEF; ZF_thm9; ZF_thm10]);;
expand (REPEAT STRIP_TAC);;
expand (DEF_RES_TAC Trans_DEF);;
expand (IMP_RES_TAC
  (((SPECL ["y:SET"; "x^1:SET"; "(x^1:SET)  $\cup$  (unit x^1)"] o
    (TAUT_REWRITE_RULE "a  $\wedge$  b  $\Rightarrow$  c = a  $\Rightarrow$  b  $\Rightarrow$  c")) ZF_thm7));;
expand (RMP_TAC "(x^1:SET)  $\subseteq$  (x^1  $\cup$  (unit x^1))");;
expand (ASM_REWRITE_TAC[]);;
expand (PURE_REWRITE_TAC [ $\subseteq$ _DEF; ZF_thm10]);;
expand (TAUT_SIMP_TAC THEN REWRITE_TAC[]);;
expand (ASM_REWRITE_TAC [ $\subseteq$ _DEF; ZF_thm10]
  THEN TAUT_SIMP_TAC THEN REWRITE_TAC[]);;

```

```

ML
expand(PURE_REWRITE_TAC[ZF_1e7; NE_DEF; ZF_thm9; ZF_thm10]);;
expand (DEF_RES_TAC ZF_1e7);;
expand (REPEAT STRIP_TAC);;

```

Leaving four sugoals

first

```

ML
expand (ACCEPT_TAC (MP (MP (SPEC_ALL (ASSUME " $\forall y z \bullet$ 
   $y \in x^1 \Rightarrow z \in x^1 \Rightarrow (z = y) \vee z \in y \vee y \in z$ ")) (ASSUME "y  $\in x^1$ ")) (ASSUME "z  $\in x^1$ "))));;

```

second

```

ML
expand (REWRITE_TAC [REWRITE_RULE [SYM (ASSUME "z = x^1")] (ASSUME "y  $\in x^1$ )]);;

```

third

```

ML
expand (REWRITE_TAC [REWRITE_RULE [SYM (ASSUME "y = x^1")] (ASSUME "z  $\in x^1$ )]);;

```

fourth

```

ML
expand (REWRITE_TAC [REWRITE_RULE [SYM (ASSUME "y = x'")] (ASSUME "z = x'")]);;
let ZF_thm21 = save_top_thm 'ZF_thm21';;

```

The successor of a natural number is a natural number.

```

ML
set_goal([], "
   $\forall(x^1:SET) \bullet N\ x^1 \Rightarrow N\ (suc\ x^1)$ 
");;
expand (PURE_REWRITE_TAC [N_DEF]);;
expand (STRIP_TAC THEN STRIP_TAC THEN IMP_RES_TAC ZF_thm21
  THEN ASM_REWRITE_TAC[]);;

```

```

ML
expand STRIP_TAC;;
expand DISJ2_TAC;;
expand (PURE_REWRITE_TAC [Sc_DEF]);;
expand (EXISTS_TAC "Ø");;
expand (REWRITE_TAC[ZF_thm16]);;

```

```

ML
expand STRIP_TAC;;
expand (REWRITE_TAC [suc_DEF;Sc_DEF;ZF_thm10;ZF_1e2;ZF_thm9]);;
expand TAUT_TAC;;

```

```

ML
expand STRIP_TAC;;
expand DISJ2_TAC;;
expand (PURE_REWRITE_TAC [suc_DEF; Sc_DEF]);;
expand (EXISTS_TAC "x^1:SET");;
expand (ASM_REWRITE_TAC[]);;

```

```

ML
expand (PURE_REWRITE_TAC [suc_DEF; ZF_thm9; ZF_thm10]);;
expand (REPEAT STRIP_TAC);;
expand (RES_TAC THEN ASM_REWRITE_TAC []);;
expand DISJ2_TAC;;
expand (ACCEPT_TAC (REWRITE_RULE [SYM (ASSUME "(y:SET) = x1") (ASSUME "Sc x1"))));;

let ZF_thm22 = save_top_thm 'ZF_thm22';;

```

The successor function is injective.

```

ML
set_goal([], "
   $\forall(x^1:\text{SET})(x^2:\text{SET}) \bullet (\text{suc } x^1 = \text{suc } x^2) \Rightarrow (x^1 = x^2)$ 
");;
expand (TAUT_REWRITE_TAC "a  $\Rightarrow$  b =  $\neg$ b  $\Rightarrow$   $\neg$ a");;
expand (STRIP_TAC THEN STRIP_TAC THEN STRIP_TAC);;
expand (PURE_REWRITE_TAC [suc_DEF; ZF_le1; ZF_thm10]);;
expand (PURE_REWRITE_TAC [ZF_thm9]);;
expand (TAUT_REWRITE_TAC "(a = b) = (a  $\Rightarrow$  b)  $\wedge$  (b  $\Rightarrow$  a)");;
expand STRIP_TAC;;
expand (ASSUME_TAC (REFL "x1"));;
expand (ASSUME_TAC (REFL "x2"));;
expand RES_TAC;;

```

```

ML
expand (MP_TAC ZF_thm13);;
expand (TAUT_REWRITE_TAC "a  $\Rightarrow$  F =  $\neg$ a");;
expand (NOT_FORALL_TAC THEN NOT_FORALL_TAC);;
expand (EXISTS_TAC "x1:SET");;
expand (EXISTS_TAC "x2:SET");;
expand (ASM_REWRITE_TAC [ $\notin$ _DEF]);;

```

```

ML
expand (MP_TAC (ASSUME " $\neg(x^1 = x^2)$ ") THEN REWRITE_TAC [ASSUME "x2 = x1"]);;

let ZF_thm23 = save_top_thm 'ZF_thm23';;

```

Every member of a natural number is a natural number.

```

ML
set_goal([], "
   $\forall(x^1:\text{SET})(x^2:\text{SET}) \bullet N x^1 \wedge x^2 \in x^1 \Rightarrow N x^2$ 
");;
expand (PURE_REWRITE_TAC [N_DEF]);;
expand (STRIP_TAC THEN STRIP_TAC);;
expand (TAUT_REWRITE_TAC "a  $\wedge$  b  $\Rightarrow$  c = b  $\Rightarrow$  a  $\Rightarrow$  c");;
expand STRIP_TAC;;
expand (LEMMA "x2  $\in$  x1  $\Rightarrow$   $\neg$ (x1 =  $\emptyset$ )");;
expand RES_TAC;;
expand (ASM_REWRITE_TAC []);;
expand STRIP_TAC;;
expand (IMP_RES_TAC ZF_thm18);;
expand (ASM_REWRITE_TAC []);;
expand STRIP_TAC;;
expand (IMP_RES_TAC (SPEC "x2:SET"
  (ASSUME " $\forall y \bullet y \in x^1 \Rightarrow (y = \emptyset) \vee \exists c y$ ")
  THEN ASM_REWRITE_TAC []));;

```

```

ML
expand (DEF_RES_TAC On_DEF);;
expand (DEF_RES_TAC Trans_DEF);;

expand STRIP_TAC;;
expand STRIP_TAC;;
expand RES_TAC;;
expand (MP_TAC (REWRITE_RULE [ASSUME "x2 =  $\emptyset$ "] (ASSUME "y  $\in$  x2")));;
expand (REWRITE_TAC [SPEC "y:SET" ZF_1e2]);;

```

```

ML
expand (DEF_RES_TAC  $\subseteq$ _DEF);;
expand (RES_TAC THEN ASM_REWRITE_TAC []);;

expand (TAUT_REWRITE_TAC "a  $\Rightarrow$   $\neg$ b = b  $\Rightarrow$   $\neg$ a");;
expand (STRIP_TAC THEN REWRITE_TAC[ASSUME "x1 =  $\emptyset$ ";ZF_1e2]);;

let ZF_thm24 = save_top_thm 'ZF_thm24';;

```

The principle of induction over the natural numbers.

```

ML
set_goal([], "
   $\forall(A:SET \rightarrow \text{bool}) \bullet A \emptyset \wedge (\forall(x:SET) \bullet N x \wedge A x \Rightarrow A (\text{succ } x))$ 
   $\Rightarrow \forall(x:SET) \bullet N x \Rightarrow A x$ 
");;
expand (STRIP_TAC THEN STRIP_TAC);;
expand (TAUT_REWRITE_TAC "a =  $\neg\neg a$ ");;
expand PURE_NOT_FORALL_TAC;;
expand (TAUT_REWRITE_TAC " $\neg(a \Rightarrow b) = a \wedge \neg b$ ");;
expand STRIP_TAC;;

```

```

ML
lemma " $\exists L:SET \bullet (L = \text{sep } (\text{succ } x) \lambda x:SET \bullet \neg (A x))$ ";;

expand (EXISTS_TAC " $\text{sep } (\text{succ } x) \lambda x:SET \bullet \neg (A x)$ ");;
expand (ASM_REWRITE_TAC[]);;

```

```

ML
lemma "x  $\in$  L";;

expand (ASM_REWRITE_TAC[ZF2;succ_DEF;ZF_thm9;ZF_thm10]);;
expand (BETA_TAC THEN ASM_REWRITE_TAC[]);;

```

```

ML
lemma " $\forall(y:SET) \bullet y \in L \Rightarrow \neg ((A:SET \rightarrow \text{bool}) y)$ ";;

expand (REPEAT STRIP_TAC);;
expand (MP_TAC (ASSUME " $L = \text{sep}(\text{succ } x)(\lambda x \bullet \neg A x)$ "));;
expand (PURE_REWRITE_TAC [ZF_1e1; ZF2]);;
expand BETA_TAC;;
expand (TAUT_REWRITE_TAC " $a \Rightarrow F = \neg a$ ");;
expand NOT_FORALL_TAC;;
expand (EXISTS_TAC "y:SET");;
expand (ASM_REWRITE_TAC[]);;

```



```

ML
lemma "L ≠ ∅";

expand (PURE_REWRITE_TAC[NE_DEF;ZF_1e1]);
expand NOT_FORALL_TAC;;
expand (EXISTS_TAC "x:SET");
expand (ASM_REWRITE_TAC[ZF_1e2]);

```

```

ML
lemma "∃b:SET• b ∈ L ∧ (b ∩ L = ∅)";

expand (IMP_RES_TAC ZF7);

```

```

ML
lemma "L ⊆ (suc x)";

expand (REWRITE_TAC[⊆_DEF]);
expand (ASM_REWRITE_TAC[ZF2]);
expand TAUT_SIMP_TAC;;

```

```

ML
lemma "b ∈ (suc x)";

expand (DEF_RES_TAC ⊆_DEF);

```

```

ML
lemma "N (suc x)";

expand (IMP_RES_TAC ZF_thm22);

```

```

ML
lemma "N b";

expand (IMP_RES_TAC ZF_thm24);

```

```

ML
lemma "b ≠ ∅";

expand (PURE_REWRITE_TAC[NE_DEF]);;
expand STRIP_TAC;;
expand (IMP_RES_TAC (SPEC "b" (ASSUME "∀y• y ∈ L ⇒ ¬A y")));;
expand (MP_TAC (REWRITE_RULE [SYM (ASSUME "b = ∅")] (ASSUME "(A:SET→bool) ∅")));;
expand (ASM_REWRITE_TAC[]);;

```

```

ML
lemma "Sc b";

expand (IMP_RES_TAC ZF_le11);;

```

```

ML
lemma "∃c:SET• (b = suc c) ∧ On c";

expand (MP_TAC (REWRITE_RULE [Sc_DEF] (ASSUME "Sc b")));;
expand (TAUT_REWRITE_TAC "a ⇒ b = ¬b ⇒ ¬a");;
expand NOT_EXISTS_TAC;;
expand FORALL_OUT_TAC;;
expand (TAUT_REWRITE_TAC "¬b ⇒ ¬a = a ⇒ b");;
expand (REPEAT STRIP_TAC THEN ASM_REWRITE_TAC[]);;

```

```

ML
lemma "N c";

lemma "N (suc c)";

expand (REWRITE_TAC[SYM (ASSUME "b = suc c");ASSUME "N b"]);;

lemma "c ∈ (suc c)";

expand (REWRITE_TAC[suc_DEF; ZF_thm9; ZF_thm10]);;

expand (IMP_RES_TAC(SPECL["suc (c:SET)";"c:SET"]ZF_thm24));;

```

```

ML
lemma "c ∈ b";;

expand (ASM_REWRITE_TAC[]);;
expand (REWRITE_TAC[suc_DEF; ZF_thm9; ZF_thm10]);;

```

```

ML
lemma "¬(c ∈ L)";;

expand STRIP_TAC;;
expand (MP_TAC (ASSUME "b ∩ L = ∅"));;
expand (PURE_REWRITE_TAC[ZF_1e1;ZF_thm9;ZF_thm10;ZF_thm11;ZF_1e2]);;
expand (TAUT_REWRITE_TAC "a ⇒ F = ¬a");;
expand NOT_FORALL_TAC;;
expand (EXISTS_TAC "c:SET");;
expand (ASM_REWRITE_TAC[]);;

```

```

ML
lemma "Trans (suc x)";;

expand (REWRITE_TAC [MP (SPEC "suc x" ZF_1e12) (ASSUME "N(suc x)")]);;

```

```

ML
lemma "c ∈ (suc x)";;
expand (DEF_RES_TAC Trans_DEF);;
expand (DEF_RES_TAC ⊆_DEF);;

```

```

ML
lemma "(A:SET→bool) c";;
expand (MP_TAC (CONJ (ASSUME "c ∈ (suc x)")(ASSUME "¬c ∈ L")));;
expand(PURE_REWRITE_TAC[ASSUME "L = sep(suc x)(λx• ¬(A:SET→bool) x)";ZF2]);;
expand BETA_TAC;;
expand TAUT_TAC;;

```

```

ML
lemma "(A:SET→bool) b";;
expand (ASM_REWRITE_TAC[]);;
expand RES_TAC;;

```

```

ML
expand RES_TAC;;
let ZF_thm25 = save_top_thm 'ZF_thm25';;

```

No the example of proof by induction.

```

ML
%
set_goal([], "
   $\forall(x^1:\text{SET})(x^2:\text{SET}) \bullet (N x^1 \wedge x^2 \in x^1) \Rightarrow (\text{suc } x^2) \in (\text{suc } x^1)$ 
");;
expand (REPEAT STRIP_TAC);;

lemma " $\forall(x^1:\text{SET}) \bullet N x^1 \Rightarrow \forall(x^2:\text{SET}) \bullet x^2 \in x^1 \Rightarrow (\text{suc } x^2) \in (\text{suc } x^1)$ ";;

lemma " $\exists A:\text{SET} \rightarrow \text{bool} \bullet A = [S1]:\text{SET} \bullet \forall(x^2:\text{SET}) \bullet x^2 \in x^1 \Rightarrow (\text{suc } x^2) \in (\text{suc } x^1)$ ";;

expand (EXISTS_TAC "[S1]:SET)  $\bullet \forall(x^2:\text{SET}) \bullet x^2 \in x^1 \Rightarrow (\text{suc } x^2) \in (\text{suc } x^1)$ ");;
expand (ASM_REWRITE_TAC[]);;

lemma "(A:SET  $\rightarrow$  bool)  $\emptyset \wedge \forall(x:\text{SET}) \bullet N x \wedge A x \Rightarrow A (\text{suc } x)$ ";;
expand (REPEAT STRIP_TAC);;

expand (ASM_REWRITE_TAC[]);;
expand BETA_TAC;;
expand (REWRITE_TAC [ZF_1e2]);;

expand (ASM_REWRITE_TAC[] THEN BETA_TAC);;
expand (REPEAT STRIP_TAC);;
expand (REWRITE_TAC [SPEC "suc x" (GEN_ALL suc_DEF);ZF_thm9;ZF_thm10]);;
expand DISJ2_TAC;;

expand (IMP_RES_TAC ZF_thm22);;

expand (MP_TAC (SPEC_ALL ZF_thm25));;
expand (ASM_REWRITE_TAC[] THEN BETA_TAC);;
expand (TAUT_REWRITE_TAC " $((a \wedge b) \Rightarrow c) = a \Rightarrow b \Rightarrow c$ ");;
expand (REPEAT STRIP_TAC);;
%

```

2.2. The completing axioms

2.2.1. The axiom of infinity

```

ML
new_constant ('N', ":SET");
let ZF8 = new_axiom ('ZF8', "
   $\forall(x^1:SET) \bullet x^1 \in \mathbb{N} = N \ x^1$ 
");

```

2.2.2. Replacement

```

ML
let ZF9 = new_axiom ('ZF9', "
   $\forall(f:SET \rightarrow SET \rightarrow bool)(r:SET) \bullet (\forall(x:SET)(y:SET)(z:SET) \bullet$ 
   $(f \ x \ y \wedge f \ x \ z \Rightarrow (z = y))) \Rightarrow$ 
   $\exists(w:SET) \bullet \forall(y:SET) \bullet y \in w = \exists(x:SET) \bullet x \in r \wedge f \ x \ y$ 
");

```

2.3. Relations, functions, and simple recursion

Relations:

```

ML
let relation_DEF = new_definition ('relation', "
  (relation:SET  $\rightarrow$  bool) x =  $\forall(y:SET) \bullet y \in x \Rightarrow \exists(w:SET)(z:SET) \bullet y = (w \mapsto z)$ 
");

```

Functions:

```

ML
let function_DEF = new_definition ('function', "
  (function:SET  $\rightarrow$  bool) x =
    relation x  $\wedge$ 
     $\forall(y:SET)(w:SET)(z:SET) \bullet (y \mapsto z) \in x \wedge (y \mapsto w) \in x \Rightarrow (z = w)$ 
");

```

```

ML
let domain_DEF = new_definition ('domain_DEF',
  (domain:SET→SET) x = sep (∪ (∪ x)) λ y:SET• ∃ z:SET• (y ↦ z) ∈ x
  ));

let image_DEF = new_definition ('image_DEF',
  (image:SET → SET) x = sep (∪ (∪ x)) λ y:SET• ∃ z:SET• (z ↦ y) ∈ x
  ));

```

```

ML
let ©_DEF = new_infix_definition ('©_DEF',
  (©:SET → SET → SET) x z = ∪ (sep (image x) λ y:SET• (z ↦ y) ∈ x)
  ));

```

```

ML
let a_DEF = new_infix_definition ('a_DEF',
  (a:SET → SET → SET) t r =
    sep (ℙ(ℙ(t ∪ r))) λ(y:SET)•∃(u:SET)(v:SET)• (y = (u ↦ v)) ∧ u ∈ t ∧ v ∈ r
  ));

```

2.4. The Axiom of Choice

The axiom of choice is here introduced as an axiom. This isn't strictly necessary since there is a choice function in HOL already, and the choice function in set theory could be defined using the HOL choice function. However the presentation follows closer to Hatcher by just introducing Hatcher's choice axiom.

```

ML
new_constant ('μ',":SET → SET");;
let ZF10 = new_axiom ('ZF10',
  ∀ x1:SET • x1 ≠ ∅ ⇒ (μ x1) ∈ x1
  ));

```

3. REFORMULATION of REPLACEMENT

```

ML
set_goal([], "
   $\forall(f:SET \rightarrow SET)(d:SET) \bullet \exists(s:SET) \bullet \forall(x:SET) \bullet$ 
     $x \in s = \exists(y:SET) \bullet y \in d \wedge (x = f y)$ 
");;
expand (REPEAT GEN_TAC);;
lemma_proof " $\exists(\text{rel}:SET \rightarrow SET \rightarrow \text{bool}) \bullet \forall(x:SET)(y:SET) \bullet \text{rel } x y = (y = f x)$ "
  [EXISTS_TAC " $\lambda(x:SET)(y:SET) \bullet y = f x$ "; REPEAT GEN_TAC;
   BETA_TAC; REFL_TAC];;
lemma_proof " $\forall(x:SET)(y:SET)(z:SET) \bullet \text{rel } x y \wedge \text{rel } x z \Rightarrow (z = y)$ "
  [ASM_REWRITE_TAC[] THEN REPEAT STRIP_TAC THEN ASM_REWRITE_TAC[]];;
lemma_proof " $\exists w \bullet \forall y \bullet y \in w = (\exists x \bullet x \in d \wedge \text{rel } x y)$ "
  [IMP_RES_TAC (SPECL ["rel:SET $\rightarrow$ SET $\rightarrow$ bool"; "d:SET"] ZF9)];;
expand (EXISTS_TAC "w:SET" THEN ASM_REWRITE_TAC[]);;
let ZF_thm26 = save_top_thm 'ZF_thm26';;

```

4. THE END

```

ML
close_theory 'zf120';;
print_theory 'zf120';;

```

5. THE THEORY ZF

The Theory ZF

Parents -- HOL

Types -- ":SET"

Constants --

```
sep ":SET → ((SET → bool) → SET)"  ∅ ":SET"
IP ":SET → SET"    pair ":SET → (SET → SET)"
unit ":SET → SET"  ∪ ":SET → SET"    ∩ ":SET → SET"
suc ":SET → SET"   Nat ":N → SET"
set ":(SET)list → SET"   Trans ":SET → bool"
Con ":SET → bool"   On ":SET → bool"   Sc ":SET → bool"
Lim ":SET → bool"   N ":SET → bool"   ℕ ":SET"
relation ":SET → bool"   function ":SET → bool"
domain ":SET → SET"   image ":SET → SET"
μ ":SET → SET"
```

Curried Infixes --

```
∈ ":SET → (SET → bool)"  ∉ ":SET → (SET → bool)"
≠ ":* → (* → bool)"    ⊆ ":SET → (SET → bool)"
↳ ":SET → (SET → SET)"  ∪ ":SET → (SET → SET)"
∩ ":SET → (SET → SET)"  ⊙ ":SET → (SET → SET)"
a ":SET → (SET → SET)"
```

Axioms --

```
EXT  ⊢ ∀x y • (∀z • z ∈ x = z ∈ y) ⇒ (x = y)
ZF2  ⊢ ∀A z x • x ∈ (sep z A) = x ∈ z ∧ A x
ZF3  ⊢ ∅ = sep ∅ (λx1 • F)
ZF4  ⊢ ∀y x • x ∈ (IP y) = x ⊆ y
ZF5  ⊢ ∀y z x • x ∈ (pair y z) = (x = y) ∨ (x = z)
ZF6  ⊢ ∀y x • x ∈ (∪ y) = (∃z • z ∈ y ∧ x ∈ z)
ZF7  ⊢ ∀x • x ≠ ∅ ⇒ (∃y • y ∈ x ∧ (y ∩ x = ∅))
ZF8  ⊢ ∀x1 • x1 ∈ ℕ = N x1
ZF9
  ⊢ ∀f r •
    (∀x y z • f x y ∧ f x z ⇒ (z = y)) ⇒
    (∃w • ∀y • y ∈ w = (∃x • x ∈ r ∧ f x y))
ZF10 ⊢ ∀x1 • x1 ≠ ∅ ⇒ (μ x1) ∈ x1
```

Definitions --

```
∉_DEF ⊢ x ∉ y = ¬x ∈ y
NE_DEF ⊢ x ≠ y = ¬(x = y)
⊆_DEF ⊢ a ⊆ b = (∀c • c ∈ a ⇒ c ∈ b)
unit_DEF ⊢ unit x = pair x x
↳_DEF ⊢ a ↳ b = pair (unit a) (pair a b)
∩_DEF ⊢ ∩ x = sep (∪ x) (λy • ∀z • z ∈ x ⇒ y ∈ z)
∪_DEF ⊢ x ∪ y = ∪ (pair x y)
∩_DEF ⊢ x ∩ y = ∩ (pair x y)
suc_DEF ⊢ suc x = x ∪ (unit x)
```


Nat_DEF $\vdash \text{Nat} = \text{PRIM_REC } \emptyset (\lambda g00012 \ n \bullet \text{suc } g00012)$
 set_DEF
 $\vdash \text{set} =$
 $(\lambda g00013 \bullet$
 $\text{LIST_PRIM_REC}(\lambda g00014 \ h \ t \bullet (\text{unit } h) \cup g00014) \emptyset g00013)$
 Trans_DEF $\vdash \text{Trans } x = (\forall y \bullet y \in x \Rightarrow y \subseteq x)$
 Con_DEF
 $\vdash \text{Con } x =$
 $(\forall y \ z \bullet y \in x \wedge z \in x \wedge z \neq y \Rightarrow z \in y \vee y \in z)$
 On_DEF $\vdash \text{On } x = \text{Trans } x \wedge \text{Con } x$
 Sc_DEF $\vdash \text{Sc } x = (\exists y \bullet \text{On } y \wedge (\text{suc } y = x))$
 Lim_DEF $\vdash \text{Lim } x = \text{On } x \wedge x \neq \emptyset \wedge \neg \text{Sc } x$
 N_DEF
 $\vdash \text{N } x =$
 $\text{On } x \wedge$
 $((x = \emptyset) \vee \text{Sc } x) \wedge$
 $(\forall y \bullet y \in x \Rightarrow (y = \emptyset) \vee \text{Sc } y)$
 relation $\vdash \text{relation } x = (\forall y \bullet y \in x \Rightarrow (\exists w \ z \bullet y = w \mapsto z))$
 function
 $\vdash \text{function } x =$
 $\text{relation } x \wedge$
 $(\forall y \ w \ z \bullet (y \mapsto z) \in x \wedge (y \mapsto w) \in x \Rightarrow (z = w))$
 domain_DEF $\vdash \text{domain } x = \text{sep}(\cup(\cup x))(\lambda y \bullet \exists z \bullet (y \mapsto z) \in x)$
 image_DEF $\vdash \text{image } x = \text{sep}(\cup(\cup x))(\lambda y \bullet \exists z \bullet (z \mapsto y) \in x)$
 $\odot_DEF \vdash x \odot z = \cup(\text{sep}(\text{image } x)(\lambda y \bullet (z \mapsto y) \in x))$
 a_DEF
 $\vdash t \ ^a r =$
 sep
 $(\mathbb{P}(\mathbb{P}(t \cup r)))$
 $(\lambda y \bullet \exists u \ v \bullet (y = u \mapsto v) \wedge u \in t \wedge v \in r)$

Theorems --

ZF_thm1
 $\vdash \forall x^1 \ x^2 \bullet$
 $(x^1 = x^2) \Rightarrow$
 $(\forall x \mathbb{R} \bullet x^1 \in x \mathbb{R} = x^2 \in x \mathbb{R})$
 ZF_le1 $\vdash \forall x \ y \bullet (x = y) = (\forall z \bullet z \in x = z \in y)$
 ZF_thm3 $\vdash \forall x^1 \bullet x^1 \notin \emptyset$
 ZF_le2 $\vdash \forall x^1 \bullet \neg x^1 \in \emptyset$
 ZF_thm4 $\vdash \forall x^1 \bullet (\forall x^2 \bullet x^2 \notin x^1) = (\emptyset = x^1)$
 ZF_le3 $\vdash \forall x^1 \ x^2 \bullet x^2 \in x^1 \Rightarrow x^1 \neq \emptyset$
 ZF_thm5
 $\vdash \forall x^1 \ x^2 \bullet$
 $x^1 \subseteq x^2 \wedge x^2 \subseteq x^1 \Rightarrow (x^1 = x^2)$
 ZF_thm6 $\vdash \forall x^1 \bullet x^1 \subseteq x^1$
 ZF_thm7
 $\vdash \forall x^1 \ x^2 \ x \mathbb{R} \bullet$

$$x^1 \subseteq x^2 \wedge x^2 \subseteq x^{\textcircled{R}} \Rightarrow x^1 \subseteq x^{\textcircled{R}}$$

ZF_thm8 $\vdash \forall x^1 \bullet \emptyset \subseteq x^1$

ZF_le4 $\vdash \forall x y \bullet x \in (\text{pair } x y) \wedge y \in (\text{pair } x y)$

ZF_thm9 $\vdash \forall x^1 \bullet x^1 \in (\text{unit } x^2) = (x^1 = x^2)$

ZF_le13 $\vdash \forall x y \bullet (\text{unit } x = \text{unit } y) = (x = y)$

ZF_le5
 $\vdash \forall x y \bullet (\text{unit } x) \in (x \mapsto y) \wedge (\text{pair } x y) \in (x \mapsto y)$

ZF_thm10
 $\vdash \forall x^1 x^2 x^{\textcircled{R}} \bullet$
 $x^1 \in (x^2 \cup x^{\textcircled{R}}) =$
 $x^1 \in x^2 \vee x^1 \in x^{\textcircled{R}}$

ZF_thm11
 $\vdash \forall x^1 x^2 x^{\textcircled{R}} \bullet$
 $x^1 \in (x^2 \cap x^{\textcircled{R}}) =$
 $x^1 \in x^2 \wedge x^1 \in x^{\textcircled{R}}$

Nat $\vdash (\text{Nat } 0 = \emptyset) \wedge (\text{Nat}(\text{SUC } n) = \text{suc}(\text{Nat } n))$

ZF_thm12 $\vdash \forall x^1 \bullet x^1 \notin x^1$

NOT_FORALL $\vdash \neg \forall \text{ pf} = (\exists x \bullet \neg \text{pf } x)$

NOT_EXISTS $\vdash \neg \exists \text{ pf} = (\forall x \bullet \neg \text{pf } x)$

ZF_thm13 $\vdash \forall x^1 x^2 \bullet x^1 \in x^2 \Rightarrow x^2 \notin x^1$

set $\vdash (\text{set}[] = \emptyset) \wedge (\text{set}(\text{CONS } h t) = (\text{unit } h) \cup (\text{set } t))$

ZF_thm14
 $\vdash \forall x^1 x^2 x^{\textcircled{R}} \bullet$
 $x^1 \in x^2 \wedge x^2 \in x^{\textcircled{R}} \Rightarrow$
 $x^{\textcircled{R}} \notin x^1 \wedge x^{\textcircled{R}} \neq x^1$

ZF_le6
 $\vdash \text{Con } x =$
 $(\forall y z \bullet y \in x \wedge z \in x \Rightarrow (z = y) \vee z \in y \vee y \in z)$

ZF_le7
 $\vdash \text{Con } x =$
 $(\forall y z \bullet y \in x \Rightarrow z \in x \Rightarrow (z = y) \vee z \in y \vee y \in z)$

ZF_le8
 $\vdash \text{Trans } x1 = (\forall x2 x3 \bullet x2 \in x1 \wedge x3 \in x2 \Rightarrow x3 \in x1)$

ZF_le9
 $\vdash \forall x^1 x^2 x^{\textcircled{R}} x^{\pm} \bullet$
 On $x^1 \wedge$
 $x^2 \in x^1 \wedge$
 $x^{\textcircled{R}} \in x^1 \wedge$
 $x^{\pm} \in x^1 \wedge$
 $x^2 \in x^{\textcircled{R}} \wedge$
 $x^{\textcircled{R}} \in x^{\pm} \Rightarrow$
 $x^2 \in x^{\pm}$

ZF_le10 $\vdash \forall p q \bullet (\forall x \bullet p x \Rightarrow q x) \Rightarrow (\forall x \bullet p x) \Rightarrow (\forall x \bullet q x)$

ZF_thm15
 $\vdash \forall x^1 x^2 \bullet$
 $x^1 \subseteq x^2 \wedge x^1 \neq \emptyset \wedge \text{On } x^2 \Rightarrow$
 $(\exists x^{\textcircled{R}} \bullet$

$$x^{\circledast} \in x^1 \wedge$$

$$(\forall x^{\pm} \bullet$$

$$x^{\pm} \in x^1 \Rightarrow (x^{\pm} = x^{\circledast}) \vee x^{\circledast} \in x^{\pm}))$$

ZF_thm16 $\vdash \text{On } \emptyset$

ZF_thm17 $\vdash \forall x^1 x^2 \bullet \text{Con } x^1 \wedge x^2 \subseteq x^1 \Rightarrow \text{Con } x^2$

ZF_thm18 $\vdash \forall x^1 x^2 \bullet \text{On } x^1 \wedge x^2 \in x^1 \Rightarrow \text{On } x^2$

ZF_thm19 $\vdash \text{N } \emptyset$

ZF_le11 $\vdash \forall x \bullet \text{N } x \wedge x \neq \emptyset \Rightarrow \text{Sc } x$

ZF_le12 $\vdash \forall x \bullet \text{N } x \Rightarrow \text{Trans } x \wedge \text{Con } x$

ZF_thm20 $\vdash \forall x^1 \bullet \emptyset \neq (\text{suc } x^1)$

ZF_thm21 $\vdash \forall x^1 \bullet \text{On } x^1 \Rightarrow \text{On}(\text{suc } x^1)$

ZF_thm22 $\vdash \forall x^1 \bullet \text{N } x^1 \Rightarrow \text{N}(\text{suc } x^1)$

ZF_thm23 $\vdash \forall x^1 x^2 \bullet (\text{suc } x^1 = \text{suc } x^2) \Rightarrow (x^1 = x^2)$

ZF_thm24 $\vdash \forall x^1 x^2 \bullet \text{N } x^1 \wedge x^2 \in x^1 \Rightarrow \text{N } x^2$

ZF_thm25

$$\vdash \forall A \bullet A \emptyset \wedge (\forall x \bullet \text{N } x \wedge A x \Rightarrow A(\text{suc } x)) \Rightarrow (\forall x \bullet \text{N } x \Rightarrow A x)$$

ZF_thm26 $\vdash \forall f d \bullet \exists s \bullet \forall x \bullet x \in s = (\exists y \bullet y \in d \wedge (x = f y))$