

Equivalents of the Axiom of Choice

Krzysztof Grąbczewski

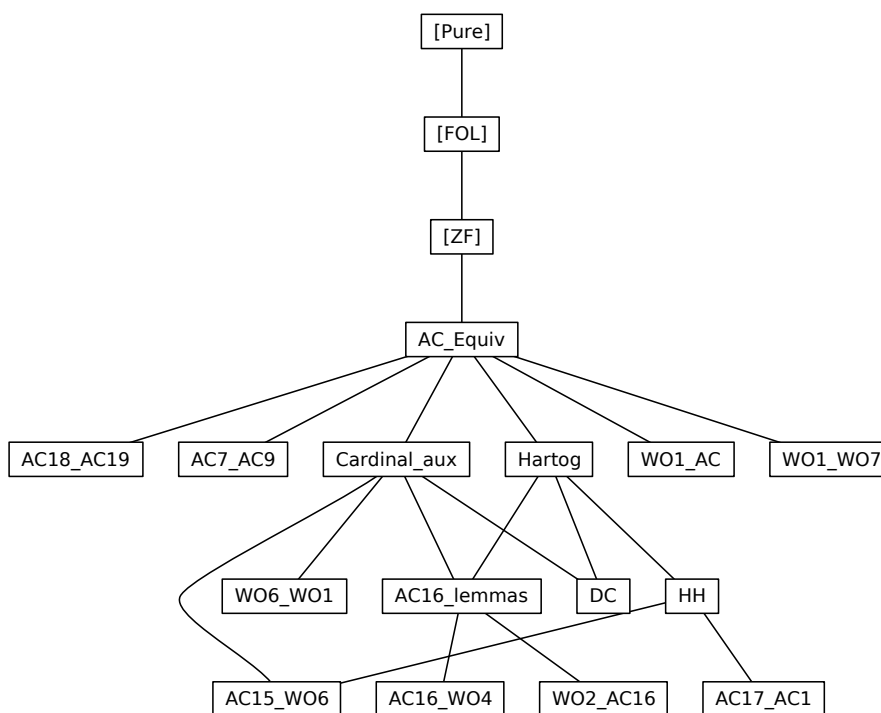
January 18, 2026

Abstract

This development [1] proves the equivalence of seven formulations of the well-ordering theorem and twenty formulations of the axiom of choice. It formalizes the first two chapters of the monograph *Equivalents of the Axiom of Choice* by Rubin and Rubin [2]. Some of this material involves extremely complex techniques.

Contents

0.1	Lemmas useful in each of the three proofs	22
0.2	Lemmas used in the proofs of $AC1 \implies WO2$ and $AC17 \implies AC1$	23
0.3	The proof of $AC1 \implies WO2$	24



```

theory AC_Equiv
imports ZF
begin

```

definition

```

"W01  $\equiv \forall A. \exists R. \text{well\_ord}(A,R)$ "

```

definition

```

"W02  $\equiv \forall A. \exists a. \text{Ord}(a) \wedge A \approx a$ "

```

definition

```

"W03  $\equiv \forall A. \exists a. \text{Ord}(a) \wedge (\exists b. b \subseteq a \wedge A \approx b)$ "

```

definition

```

"W04(m)  $\equiv \forall A. \exists a f. \text{Ord}(a) \wedge \text{domain}(f)=a \wedge$ 
 $(\bigcup b < a. f' b) = A \wedge (\forall b < a. f' b \lesssim m)$ "

```

definition

```

"W05  $\equiv \exists m \in \text{nat}. 1 \leq m \wedge W04(m)$ "

```

definition

```

"W06  $\equiv \forall A. \exists m \in \text{nat}. 1 \leq m \wedge (\exists a f. \text{Ord}(a) \wedge \text{domain}(f)=a$ 
 $\wedge (\bigcup b < a. f' b) = A \wedge (\forall b < a. f' b \lesssim m))$ "

```

definition

```

"W07  $\equiv \forall A. \text{Finite}(A) \longleftrightarrow (\forall R. \text{well\_ord}(A,R) \longrightarrow \text{well\_ord}(A,\text{converse}(R)))$ "

```

definition

```

"W08  $\equiv \forall A. (\exists f. f \in (\prod X \in A. X)) \longrightarrow (\exists R. \text{well\_ord}(A,R))$ "

```

definition

```

pairwise_disjoint :: "i  $\Rightarrow$  o" where
"pairwise_disjoint(A)  $\equiv \forall A1 \in A. \forall A2 \in A. A1 \cap A2 \neq 0 \longrightarrow A1=A2$ "

```

definition

```

sets_of_size_between :: "[i, i, i]  $\Rightarrow$  o" where
"sets_of_size_between(A,m,n)  $\equiv \forall B \in A. m \lesssim B \wedge B \lesssim n$ "

```

definition

```

"AC0  $\equiv \forall A. \exists f. f \in (\prod X \in \text{Pow}(A)-\{0\}. X)$ "

```

definition

$$"AC1 \equiv \forall A. 0 \notin A \longrightarrow (\exists f. f \in (\prod X \in A. X))"$$

definition

$$"AC2 \equiv \forall A. 0 \notin A \wedge \text{pairwise_disjoint}(A) \longrightarrow (\exists C. \forall B \in A. \exists y. B \cap C = \{y\})"$$

definition

$$"AC3 \equiv \forall A B. \forall f \in A \rightarrow B. \exists g. g \in (\prod x \in \{a \in A. f'a \neq 0\}. f'x)"$$

definition

$$"AC4 \equiv \forall R A B. (R \subseteq A * B \longrightarrow (\exists f. f \in (\prod x \in \text{domain}(R). R'\{x\})))"$$

definition

$$"AC5 \equiv \forall A B. \forall f \in A \rightarrow B. \exists g \in \text{range}(f) \rightarrow A. \forall x \in \text{domain}(g). f'(g'x) = x"$$

definition

$$"AC6 \equiv \forall A. 0 \notin A \longrightarrow (\prod B \in A. B) \neq 0"$$

definition

$$"AC7 \equiv \forall A. 0 \notin A \wedge (\forall B1 \in A. \forall B2 \in A. B1 \approx B2) \longrightarrow (\prod B \in A. B) \neq 0"$$

definition

$$"AC8 \equiv \forall A. (\forall B \in A. \exists B1 B2. B = \langle B1, B2 \rangle \wedge B1 \approx B2) \longrightarrow (\exists f. \forall B \in A. f'B \in \text{bij}(\text{fst}(B), \text{snd}(B)))"$$

definition

$$"AC9 \equiv \forall A. (\forall B1 \in A. \forall B2 \in A. B1 \approx B2) \longrightarrow (\exists f. \forall B1 \in A. \forall B2 \in A. f'\langle B1, B2 \rangle \in \text{bij}(B1, B2))"$$

definition

$$"AC10(n) \equiv \forall A. (\forall B \in A. \neg \text{Finite}(B)) \longrightarrow (\exists f. \forall B \in A. (\text{pairwise_disjoint}(f'B) \wedge \text{sets_of_size_between}(f'B, 2, \text{succ}(n)) \wedge \bigcup (f'B) = B))"$$

definition

$$"AC11 \equiv \exists n \in \text{nat}. 1 \leq n \wedge AC10(n)"$$

definition

$$"AC12 \equiv \forall A. (\forall B \in A. \neg \text{Finite}(B)) \longrightarrow (\exists n \in \text{nat}. 1 \leq n \wedge (\exists f. \forall B \in A. (\text{pairwise_disjoint}(f'B) \wedge \text{sets_of_size_between}(f'B, 2, \text{succ}(n)) \wedge \bigcup (f'B) = B)))"$$

definition

$$"AC13(m) \equiv \forall A. 0 \notin A \longrightarrow (\exists f. \forall B \in A. f'B \neq 0 \wedge f'B \subseteq B \wedge f'B \lesssim m)"$$

definition

"AC14 $\equiv \exists m \in \text{nat}. 1 \leq m \wedge \text{AC13}(m)$ "

definition

"AC15 $\equiv \forall A. 0 \notin A \longrightarrow$
 $(\exists m \in \text{nat}. 1 \leq m \wedge (\exists f. \forall B \in A. f'B \neq 0 \wedge f'B \subseteq B \wedge$
 $f'B \lesssim m))$ "

definition

"AC16(n, k) \equiv
 $\forall A. \neg \text{Finite}(A) \longrightarrow$
 $(\exists T. T \subseteq \{X \in \text{Pow}(A). X \approx \text{succ}(n)\} \wedge$
 $(\forall X \in \{X \in \text{Pow}(A). X \approx \text{succ}(k)\}. \exists ! Y. Y \in T \wedge X \subseteq Y))$ "

definition

"AC17 $\equiv \forall A. \forall g \in (\text{Pow}(A) - \{0\} \rightarrow A) \rightarrow \text{Pow}(A) - \{0\}.$
 $\exists f \in \text{Pow}(A) - \{0\} \rightarrow A. f'(g'f) \in g'f$ "

locale AC18 =

assumes AC18: " $A \neq 0 \wedge (\forall a \in A. B(a) \neq 0) \longrightarrow$
 $((\bigcap a \in A. \bigcup b \in B(a). X(a, b)) =$
 $(\bigcup f \in \prod a \in A. B(a). \bigcap a \in A. X(a, f'a)))$ "
 — AC18 cannot be expressed within the object-logic

definition

"AC19 $\equiv \forall A. A \neq 0 \wedge 0 \notin A \longrightarrow ((\bigcap a \in A. \bigcup b \in a. b) =$
 $(\bigcup f \in (\prod B \in A. B). \bigcap a \in A. f'a))$ "

lemma rvimage_id: " $\text{rvimage}(A, \text{id}(A), r) = r \cap A * A$ "

$\langle \text{proof} \rangle$

lemma ordertype_Int:

" $\text{well_ord}(A, r) \implies \text{ordertype}(A, r \cap A * A) = \text{ordertype}(A, r)$ "
 $\langle \text{proof} \rangle$

lemma lam_sing_bij: " $(\lambda x \in A. \{x\}) \in \text{bij}(A, \{\{x\}. x \in A\})$ "

$\langle \text{proof} \rangle$

lemma inj_strengthen_type:

" $\llbracket f \in \text{inj}(A, B); \bigwedge a. a \in A \implies f'a \in C \rrbracket \implies f \in \text{inj}(A, C)$ "
 $\langle \text{proof} \rangle$

lemma *ex1_two_eq*: " $\llbracket \exists ! x. P(x); P(x); P(y) \rrbracket \implies x=y$ "
 $\langle proof \rangle$

lemma *first_in_B*:
 $\llbracket well_ord(\bigcup(A), r); 0 \notin A; B \in A \rrbracket \implies (THE\ b.\ first(b, B, r)) \in B$ "
 $\langle proof \rangle$

lemma *ex_choice_fun*: " $\llbracket well_ord(\bigcup(A), R); 0 \notin A \rrbracket \implies \exists f. f \in (\prod X$
 $\in A. X)$ "
 $\langle proof \rangle$

lemma *ex_choice_fun_Pow*: " $well_ord(A, R) \implies \exists f. f \in (\prod X \in Pow(A) - \{0\}.$
 $X)$ "
 $\langle proof \rangle$

lemma *lepoll_m_imp_domain_lepoll_m*:
 $\llbracket m \in nat; u \lesssim m \rrbracket \implies domain(u) \lesssim m$ "
 $\langle proof \rangle$

lemma *rel_domain_ex1*:
 $\llbracket succ(m) \lesssim domain(r); r \lesssim succ(m); m \in nat \rrbracket \implies function(r)$ "
 $\langle proof \rangle$

lemma *rel_is_fun*:
 $\llbracket succ(m) \lesssim domain(r); r \lesssim succ(m); m \in nat;$
 $r \subseteq A*B; A=domain(r) \rrbracket \implies r \in A \rightarrow B$ "
 $\langle proof \rangle$

end

theory *Cardinal_aux* **imports** *AC_Equiv* **begin**

lemma *Diff_lepoll*: " $\llbracket A \lesssim \text{succ}(m); B \subseteq A; B \neq 0 \rrbracket \implies A - B \lesssim m$ "
 $\langle \text{proof} \rangle$

lemma *lepoll_imp_ex_le_eqpoll*:
 $\llbracket A \lesssim i; \text{Ord}(i) \rrbracket \implies \exists j. j \leq i \wedge A \approx j$
 $\langle \text{proof} \rangle$

lemma *lesspoll_imp_ex_lt_eqpoll*:
 $\llbracket A < i; \text{Ord}(i) \rrbracket \implies \exists j. j < i \wedge A \approx j$
 $\langle \text{proof} \rangle$

lemma *Un_eqpoll_Inf_Ord*:
 assumes A : " $A \approx i$ " and B : " $B \approx i$ " and NFI : " $\neg \text{Finite}(i)$ " and i :
 $\text{"Ord}(i)"$
 shows " $A \cup B \approx i$ "
 $\langle \text{proof} \rangle$

schematic_goal *paired_bij*: " $?f \in \text{bij}(\{y, z\}. y \in x, x)$ "
 $\langle \text{proof} \rangle$

lemma *paired_eqpoll*: " $\{y, z\}. y \in x \approx x$ "
 $\langle \text{proof} \rangle$

lemma *ex_eqpoll_disjoint*: " $\exists B. B \approx A \wedge B \cap C = 0$ "
 $\langle \text{proof} \rangle$

lemma *Un_lepoll_Inf_Ord*:
 $\llbracket A \lesssim i; B \lesssim i; \neg \text{Finite}(i); \text{Ord}(i) \rrbracket \implies A \cup B \lesssim i$
 $\langle \text{proof} \rangle$

lemma *Least_in_Ord*: " $\llbracket P(i); i \in j; \text{Ord}(j) \rrbracket \implies (\mu i. P(i)) \in j$ "
 $\langle \text{proof} \rangle$

lemma *Diff_first_lepoll*:
 $\llbracket \text{well_ord}(x, r); y \subseteq x; y \lesssim \text{succ}(n); n \in \text{nat} \rrbracket$
 $\implies y - \{\text{THE } b. \text{first}(b, y, r)\} \lesssim n$
 $\langle \text{proof} \rangle$

```

lemma UN_subset_split:
  " $(\bigcup x \in X. P(x)) \subseteq (\bigcup x \in X. P(x) \neg Q(x)) \cup (\bigcup x \in X. Q(x))$ "
  <proof>

lemma UN_sing_lepoll: " $Ord(a) \implies (\bigcup x \in a. \{P(x)\}) \lesssim a$ "
  <proof>

lemma UN_fun_lepoll_lemma [rule_format]:
  " $\llbracket well\_ord(T, R); \neg Finite(a); Ord(a); n \in nat \rrbracket$ 
 $\implies \forall f. (\forall b \in a. f' b \lesssim n \wedge f' b \subseteq T) \longrightarrow (\bigcup b \in a. f' b) \lesssim a$ "
  <proof>

lemma UN_fun_lepoll:
  " $\llbracket \forall b \in a. f' b \lesssim n \wedge f' b \subseteq T; well\_ord(T, R);$ 
 $\neg Finite(a); Ord(a); n \in nat \rrbracket \implies (\bigcup b \in a. f' b) \lesssim a$ "
  <proof>

lemma UN_lepoll:
  " $\llbracket \forall b \in a. F(b) \lesssim n \wedge F(b) \subseteq T; well\_ord(T, R);$ 
 $\neg Finite(a); Ord(a); n \in nat \rrbracket$ 
 $\implies (\bigcup b \in a. F(b)) \lesssim a$ "
  <proof>

lemma UN_eq_UN_Diffs:
  " $Ord(a) \implies (\bigcup b \in a. F(b)) = (\bigcup b \in a. F(b) - (\bigcup c \in b. F(c)))$ "
  <proof>

lemma lepoll_imp_eqpoll_subset:
  " $a \lesssim X \implies \exists Y. Y \subseteq X \wedge a \approx Y$ "
  <proof>

lemma Diff_lesspoll_eqpoll_Card_lemma:
  " $\llbracket A \approx a; \neg Finite(a); Card(a); B \prec a; A - B \prec a \rrbracket \implies P$ "
  <proof>

lemma Diff_lesspoll_eqpoll_Card:
  " $\llbracket A \approx a; \neg Finite(a); Card(a); B \prec a \rrbracket \implies A - B \approx a$ "
  <proof>

end

theory W06_W01
imports Cardinal_aux
begin

```

definition

```

NN  :: "i  $\Rightarrow$  i"  where
  "NN(y)  $\equiv$  {m  $\in$  nat.  $\exists$  a.  $\exists$  f. Ord(a)  $\wedge$  domain(f)=a  $\wedge$ 
    ( $\bigcup$  b<a. f' b) = y  $\wedge$  ( $\forall$  b<a. f' b  $\lesssim$  m)}"
```

definition

```

uu  :: "[i, i, i, i]  $\Rightarrow$  i"  where
  "uu(f, beta, gamma, delta)  $\equiv$  (f' beta * f' gamma)  $\cap$  f' delta"
```

definition

```

vv1 :: "[i, i, i]  $\Rightarrow$  i"  where
  "vv1(f,m,b)  $\equiv$ 
    let g =  $\mu$  g. ( $\exists$  d. Ord(d)  $\wedge$  (domain(uu(f,b,g,d))  $\neq$  0  $\wedge$ 
      domain(uu(f,b,g,d))  $\lesssim$  m));
    d =  $\mu$  d. domain(uu(f,b,g,d))  $\neq$  0  $\wedge$ 
      domain(uu(f,b,g,d))  $\lesssim$  m
    in if f' b  $\neq$  0 then domain(uu(f,b,g,d)) else 0"
```

definition

```

ww1 :: "[i, i, i]  $\Rightarrow$  i"  where
  "ww1(f,m,b)  $\equiv$  f' b - vv1(f,m,b)"
```

definition

```

gg1 :: "[i, i, i]  $\Rightarrow$  i"  where
  "gg1(f,a,m)  $\equiv$   $\lambda$  b  $\in$  a++a. if b<a then vv1(f,m,b) else ww1(f,m,b--a)"
```

definition

```

vv2 :: "[i, i, i, i]  $\Rightarrow$  i"  where
  "vv2(f,b,g,s)  $\equiv$ 
    if f' g  $\neq$  0 then {uu(f, b, g,  $\mu$  d. uu(f,b,g,d)  $\neq$  0) 's} else
0"
```

definition

```

ww2 :: "[i, i, i, i]  $\Rightarrow$  i"  where
  "ww2(f,b,g,s)  $\equiv$  f' g - vv2(f,b,g,s)"
```

definition

```

gg2 :: "[i, i, i, i]  $\Rightarrow$  i"  where
  "gg2(f,a,b,s)  $\equiv$ 
     $\lambda$  g  $\in$  a++a. if g<a then vv2(f,b,g,s) else ww2(f,b,g--a,s)"
```

lemma W02_W03: "W02 \implies W03"

$\langle proof \rangle$

lemma *W03_W01*: " $W03 \implies W01$ "
 $\langle proof \rangle$

lemma *W01_W02*: " $W01 \implies W02$ "
 $\langle proof \rangle$

lemma *lam_sets*: " $f \in A \rightarrow B \implies (\lambda x \in A. \{f'x\}): A \rightarrow \{\{b\}. b \in B\}$ "
 $\langle proof \rangle$

lemma *surj_imp_eq'*: " $f \in \text{surj}(A, B) \implies (\bigcup a \in A. \{f'a\}) = B$ "
 $\langle proof \rangle$

lemma *surj_imp_eq*: " $\llbracket f \in \text{surj}(A, B); \text{Ord}(A) \rrbracket \implies (\bigcup a \in A. \{f'a\}) = B$ "
 $\langle proof \rangle$

lemma *W01_W04*: " $W01 \implies W04(1)$ "
 $\langle proof \rangle$

lemma *W04_mono*: " $\llbracket m \leq n; W04(m) \rrbracket \implies W04(n)$ "
 $\langle proof \rangle$

lemma *W04_W05*: " $\llbracket m \in \text{nat}; 1 \leq m; W04(m) \rrbracket \implies W05$ "
 $\langle proof \rangle$

lemma *W05_W06*: " $W05 \implies W06$ "
 $\langle proof \rangle$

lemma *lt_oadd_odiff_disj*:
" $\llbracket k < i++j; \text{Ord}(i); \text{Ord}(j) \rrbracket$
 $\implies k < i \mid (\neg k < i \wedge k = i ++ (k--i) \wedge (k--i) < j)$ "
 $\langle proof \rangle$

lemma domain_uu_subset: "domain(uu(f,b,g,d)) \subseteq f' b"
 <proof>

lemma quant_domain_uu_lepoll_m:
 " $\forall b < a. f' b \lesssim m \implies \forall b < a. \forall g < a. \forall d < a. \text{domain}(uu(f,b,g,d)) \lesssim m$ "
 <proof>

lemma uu_subset1: "uu(f,b,g,d) \subseteq f' b * f' g"
 <proof>

lemma uu_subset2: "uu(f,b,g,d) \subseteq f' d"
 <proof>

lemma uu_lepoll_m: " $\llbracket \forall b < a. f' b \lesssim m; d < a \rrbracket \implies uu(f,b,g,d) \lesssim m$ "
 <proof>

lemma cases:
 " $\forall b < a. \forall g < a. \forall d < a. u(f,b,g,d) \lesssim m$
 $\implies (\forall b < a. f' b \neq 0 \longrightarrow$
 $(\exists g < a. \exists d < a. u(f,b,g,d) \neq 0 \wedge u(f,b,g,d) \prec m))$
 | $(\exists b < a. f' b \neq 0 \wedge (\forall g < a. \forall d < a. u(f,b,g,d) \neq 0 \longrightarrow$
 $u(f,b,g,d) \approx m))$ "
 <proof>

lemma UN_oadd: "Ord(a) $\implies (\bigcup b < a ++ a. C(b)) = (\bigcup b < a. C(b) \cup C(a ++ b))$ "
 <proof>

lemma vv1_subset: "vv1(f,m,b) \subseteq f' b"
 <proof>

lemma *UN_gg1_eq*:
 "[[Ord(a); m ∈ nat]] ⇒ (⋃ b<a++a. gg1(f,a,m) 'b) = (⋃ b<a. f 'b)"
 <proof>

lemma *domain_gg1*: "domain(gg1(f,a,m)) = a++a"
 <proof>

lemma *nested_LeastI*:
 "[[P(a, b); Ord(a); Ord(b);
 Least_a = (μ a. ∃ x. Ord(x) ∧ P(a, x))]]
 ⇒ P(Least_a, μ b. P(Least_a, b))]"
 <proof>

lemmas *nested_Least_instance* =
 nested_LeastI [of "λg d. domain(uu(f,b,g,d)) ≠ 0 ∧
 domain(uu(f,b,g,d)) ≲ m"] for f b m

lemma *gg1_lepoll_m*:
 "[[Ord(a); m ∈ nat;
 ∀ b<a. f 'b ≠ 0 →
 (∃ g<a. ∃ d<a. domain(uu(f,b,g,d)) ≠ 0 ∧
 domain(uu(f,b,g,d)) ≲ m);
 ∀ b<a. f 'b ≲ succ(m); b<a++a]]
 ⇒ gg1(f,a,m) 'b ≲ m]"
 <proof>

lemma *ex_d_uu_not_empty*:
 "[[b<a; g<a; f 'b ≠ 0; f 'g ≠ 0;
 y*y ⊆ y; (⋃ b<a. f 'b)=y]]
 ⇒ ∃ d<a. uu(f,b,g,d) ≠ 0]"
 <proof>

lemma *uu_not_empty*:
 "[[b<a; g<a; f 'b ≠ 0; f 'g ≠ 0; y*y ⊆ y; (⋃ b<a. f 'b)=y]]

$\implies \text{uu}(f, b, g, \mu d. (\text{uu}(f, b, g, d) \neq 0)) \neq 0$
 $\langle \text{proof} \rangle$

lemma *not_empty_rel_imp_domain*: " $\llbracket r \subseteq A*B; r \neq 0 \rrbracket \implies \text{domain}(r) \neq 0$ "
 $\langle \text{proof} \rangle$

lemma *Least_uu_not_empty_lt_a*:
 $\llbracket b < a; g < a; f' b \neq 0; f' g \neq 0; y*y \subseteq y; (\bigcup b < a. f' b) = y \rrbracket$
 $\implies (\mu d. \text{uu}(f, b, g, d) \neq 0) < a$
 $\langle \text{proof} \rangle$

lemma *subset_Diff_sing*: " $\llbracket B \subseteq A; a \notin B \rrbracket \implies B \subseteq A - \{a\}$ "
 $\langle \text{proof} \rangle$

lemma *supset_lepoll_imp_eq*:
 $\llbracket A \lesssim m; m \lesssim B; B \subseteq A; m \in \text{nat} \rrbracket \implies A = B$
 $\langle \text{proof} \rangle$

lemma *uu_Least_is_fun*:
 $\llbracket \forall g < a. \forall d < a. \text{domain}(\text{uu}(f, b, g, d)) \neq 0 \longrightarrow$
 $\text{domain}(\text{uu}(f, b, g, d)) \approx \text{succ}(m);$
 $\forall b < a. f' b \lesssim \text{succ}(m); y*y \subseteq y;$
 $(\bigcup b < a. f' b) = y; b < a; g < a; d < a;$
 $f' b \neq 0; f' g \neq 0; m \in \text{nat}; s \in f' b \rrbracket$
 $\implies \text{uu}(f, b, g, \mu d. \text{uu}(f, b, g, d) \neq 0) \in f' b \rightarrow f' g$
 $\langle \text{proof} \rangle$

lemma *vv2_subset*:
 $\llbracket \forall g < a. \forall d < a. \text{domain}(\text{uu}(f, b, g, d)) \neq 0 \longrightarrow$
 $\text{domain}(\text{uu}(f, b, g, d)) \approx \text{succ}(m);$
 $\forall b < a. f' b \lesssim \text{succ}(m); y*y \subseteq y;$
 $(\bigcup b < a. f' b) = y; b < a; g < a; m \in \text{nat}; s \in f' b \rrbracket$
 $\implies \text{vv2}(f, b, g, s) \subseteq f' g$
 $\langle \text{proof} \rangle$

lemma *UN_gg2_eq*:
 $\llbracket \forall g < a. \forall d < a. \text{domain}(\text{uu}(f, b, g, d)) \neq 0 \longrightarrow$
 $\text{domain}(\text{uu}(f, b, g, d)) \approx \text{succ}(m);$
 $\forall b < a. f' b \lesssim \text{succ}(m); y*y \subseteq y;$
 $(\bigcup b < a. f' b) = y; \text{Ord}(a); m \in \text{nat}; s \in f' b; b < a \rrbracket$
 $\implies (\bigcup g < a++a. \text{gg2}(f, a, b, s) \text{ ' } g) = y$
 $\langle \text{proof} \rangle$

lemma *domain_gg2*: " $\text{domain}(\text{gg2}(f, a, b, s)) = a++a$ "
 $\langle \text{proof} \rangle$

lemma *vv2_lepoll*: " $\llbracket m \in \text{nat}; m \neq 0 \rrbracket \implies \text{vv2}(f, b, g, s) \lesssim m$ "
 $\langle \text{proof} \rangle$

lemma *ww2_lepoll*:
" $\llbracket \forall b < a. f' b \lesssim \text{succ}(m); g < a; m \in \text{nat}; \text{vv2}(f, b, g, d) \subseteq f' g \rrbracket$
 $\implies \text{ww2}(f, b, g, d) \lesssim m$ "
 $\langle \text{proof} \rangle$

lemma *gg2_lepoll_m*:
" $\llbracket \forall g < a. \forall d < a. \text{domain}(\text{uu}(f, b, g, d)) \neq 0 \longrightarrow$
 $\text{domain}(\text{uu}(f, b, g, d)) \approx \text{succ}(m);$
 $\forall b < a. f' b \lesssim \text{succ}(m); y * y \subseteq y;$
 $(\bigcup b < a. f' b) = y; b < a; s \in f' b; m \in \text{nat}; m \neq 0; g < a + a \rrbracket$
 $\implies \text{gg2}(f, a, b, s) \text{ ' } g \lesssim m$ "
 $\langle \text{proof} \rangle$

lemma *lemma_ii*: " $\llbracket \text{succ}(m) \in \text{NN}(y); y * y \subseteq y; m \in \text{nat}; m \neq 0 \rrbracket \implies m \in \text{NN}(y)$ "
 $\langle \text{proof} \rangle$

lemma *z_n_subset_z_succ_n*:
" $\forall n \in \text{nat}. \text{rec}(n, x, \lambda k r. r \cup r * r) \subseteq \text{rec}(\text{succ}(n), x, \lambda k r. r \cup r * r)$ "
 $\langle \text{proof} \rangle$

lemma *le_subsets*:
" $\llbracket \forall n \in \text{nat}. f(n) \leq f(\text{succ}(n)); n \leq m; n \in \text{nat}; m \in \text{nat} \rrbracket$
 $\implies f(n) \leq f(m)$ "
 $\langle \text{proof} \rangle$

lemma *le_imp_rec_subset*:

$$\llbracket n \leq m; m \in \text{nat} \rrbracket$$

$$\implies \text{rec}(n, x, \lambda k r. r \cup r * r) \subseteq \text{rec}(m, x, \lambda k r. r \cup r * r)$$
 $\langle \text{proof} \rangle$

lemma lemma_iv: " $\exists y. x \cup y * y \subseteq y$ "
 $\langle \text{proof} \rangle$

lemma W06_imp_NN_not_empty: " $W06 \implies NN(y) \neq 0$ "
 $\langle \text{proof} \rangle$

lemma lemma1:

$$\llbracket (\bigcup b < a. f' b) = y; x \in y; \forall b < a. f' b \lesssim 1; \text{Ord}(a) \rrbracket \implies \exists c < a. f' c = \{x\}$$
 $\langle \text{proof} \rangle$

lemma lemma2:

$$\llbracket (\bigcup b < a. f' b) = y; x \in y; \forall b < a. f' b \lesssim 1; \text{Ord}(a) \rrbracket$$

$$\implies f' (\mu i. f' i = \{x\}) = \{x\}$$
 $\langle \text{proof} \rangle$

lemma NN_imp_ex_inj: " $1 \in NN(y) \implies \exists a f. \text{Ord}(a) \wedge f \in \text{inj}(y, a)$ "
 $\langle \text{proof} \rangle$

lemma y_well_ord: " $\llbracket y * y \subseteq y; 1 \in NN(y) \rrbracket \implies \exists r. \text{well_ord}(y, r)$ "
 $\langle \text{proof} \rangle$

```

lemma rev_induct_lemma [rule_format]:
  "⟦n ∈ nat; ∧m. ⟦m ∈ nat; m≠0; P(succ(m))⟧⟧ ⇒ P(m)⟧
  ⇒ n≠0 → P(n) → P(1)"
⟨proof⟩

```

```

lemma rev_induct:
  "⟦n ∈ nat; P(n); n≠0;
  ∧m. ⟦m ∈ nat; m≠0; P(succ(m))⟧⟧ ⇒ P(m)⟧
  ⇒ P(1)"
⟨proof⟩

```

```

lemma NN_into_nat: "n ∈ NN(y) ⇒ n ∈ nat"
⟨proof⟩

```

```

lemma lemma3: "⟦n ∈ NN(y); y*y ⊆ y; n≠0⟧ ⇒ 1 ∈ NN(y)"
⟨proof⟩

```

```

lemma NN_y_0: "0 ∈ NN(y) ⇒ y=0"
⟨proof⟩

```

```

lemma W06_imp_W01: "W06 ⇒ W01"
⟨proof⟩

```

end

```

theory W01_W07
imports AC_Equiv
begin

```

```

definition
  "LEMMA ≡
  ∀X. ¬Finite(X) → (∃R. well_ord(X,R) ∧ ¬well_ord(X,converse(R)))"

```

```

lemma W07_iff_LEMMA: "W07 ↔ LEMMA"
⟨proof⟩

```

```

lemma LEMMA_imp_W01: "LEMMA  $\implies$  W01"
  <proof>

```

```

lemma converse_Memrel_not_wf_on:
  "[[Ord(a);  $\neg$ Finite(a)]]  $\implies$   $\neg$ wf[a](converse(Memrel(a)))"
  <proof>

```

```

lemma converse_Memrel_not_well_ord:
  "[[Ord(a);  $\neg$ Finite(a)]]  $\implies$   $\neg$ well_ord(a, converse(Memrel(a)))"
  <proof>

```

```

lemma well_ord_rvimage_ordertype:
  "well_ord(A,r)  $\implies$ 
    rvimage (ordertype(A,r), converse(ordermap(A,r)),r) =
    Memrel(ordertype(A,r))"
  <proof>

```

```

lemma well_ord_converse_Memrel:
  "[[well_ord(A,r); well_ord(A, converse(r))]]
   $\implies$  well_ord(ordertype(A,r), converse(Memrel(ordertype(A,r))))"
  <proof>

```

```

lemma W01_imp_LEMMA: "W01  $\implies$  LEMMA"
  <proof>

```

```

lemma W01_iff_W07: "W01  $\longleftrightarrow$  W07"
  <proof>

```



```
lemma W01_W08: "W01  $\implies$  W08"
<proof>
```

```
lemma W08_W01: "W08  $\implies$  W01"
<proof>
```

```
end
```

```
theory AC7_AC9
imports AC_Equiv
begin
```

```
lemma Sigma_fun_space_not0: "[ $0 \notin A$ ;  $B \in A$ ]  $\implies$   $(\text{nat} \rightarrow \bigcup (A)) * B \neq 0$ "
<proof>
```

```
lemma inj_lemma:
  "C  $\in A \implies (\lambda g \in (\text{nat} \rightarrow \bigcup (A)) * C.
    (\lambda n \in \text{nat}. \text{if}(n=0, \text{snd}(g), \text{fst}(g) ' (n \#- 1))))
    \in \text{inj}((\text{nat} \rightarrow \bigcup (A)) * C, (\text{nat} \rightarrow \bigcup (A))) "$ "
  <proof>
```

```
lemma Sigma_fun_space_eqpoll:
  "[ $C \in A$ ;  $0 \notin A$ ]  $\implies (\text{nat} \rightarrow \bigcup (A)) * C \approx (\text{nat} \rightarrow \bigcup (A))$ "
  <proof>
```

```
lemma AC6_AC7: "AC6  $\implies$  AC7"
<proof>
```

```
lemma lemma1_1: " $y \in (\prod B \in A. Y * B) \implies (\lambda B \in A. \text{snd}(y ' B)) \in (\prod B \in$ 
```

$A. B)$ "
 $\langle proof \rangle$

lemma lemma1_2:
 $"y \in (\prod B \in \{Y * C. C \in A\}. B) \implies (\lambda B \in A. y'(Y * B)) \in (\prod B \in A. Y * B)"$
 $\langle proof \rangle$

lemma AC7_AC6_lemma1:
 $"(\prod B \in \{(\text{nat} \rightarrow \bigcup (A)) * C. C \in A\}. B) \neq 0 \implies (\prod B \in A. B) \neq 0"$
 $\langle proof \rangle$

lemma AC7_AC6_lemma2: $"0 \notin A \implies 0 \notin \{(\text{nat} \rightarrow \bigcup (A)) * C. C \in A\}"$
 $\langle proof \rangle$

lemma AC7_AC6: $"AC7 \implies AC6"$
 $\langle proof \rangle$

lemma AC1_AC8_lemma1:
 $"\forall B \in A. \exists B1 B2. B = \langle B1, B2 \rangle \wedge B1 \approx B2$
 $\implies 0 \notin \{ \text{bij}(\text{fst}(B), \text{snd}(B)). B \in A \}"$
 $\langle proof \rangle$

lemma AC1_AC8_lemma2:
 $"\llbracket f \in (\prod X \in \text{RepFun}(A, p). X); D \in A \rrbracket \implies (\lambda x \in A. f'p(x))'D \in p(D)"$
 $\langle proof \rangle$

lemma AC1_AC8: $"AC1 \implies AC8"$
 $\langle proof \rangle$

lemma AC8_AC9_lemma:
 $"\forall B1 \in A. \forall B2 \in A. B1 \approx B2$
 $\implies \forall B \in A * A. \exists B1 B2. B = \langle B1, B2 \rangle \wedge B1 \approx B2"$
 $\langle proof \rangle$

lemma AC8_AC9: $"AC8 \implies AC9"$

$\langle proof \rangle$

lemma *snd_lepoll_SigmaI*: " $b \in B \implies X \lesssim B \times X$ "
 $\langle proof \rangle$

lemma *nat_lepoll_lemma*:
" $\llbracket 0 \notin A; B \in A \rrbracket \implies \text{nat} \lesssim ((\text{nat} \rightarrow \bigcup (A)) \times B) \times \text{nat}$ "
 $\langle proof \rangle$

lemma *AC9_AC1_lemma1*:
" $\llbracket 0 \notin A; A \neq 0; \\ C = \{((\text{nat} \rightarrow \bigcup (A)) * B) * \text{nat}. B \in A\} \cup \\ \{\text{cons}(0, ((\text{nat} \rightarrow \bigcup (A)) * B) * \text{nat}). B \in A\}; \\ B1 \in C; B2 \in C \rrbracket \\ \implies B1 \approx B2$ "
 $\langle proof \rangle$

lemma *AC9_AC1_lemma2*:
" $\forall B1 \in \{(F*B)*N. B \in A\} \cup \{\text{cons}(0, (F*B)*N). B \in A\}. \\ \forall B2 \in \{(F*B)*N. B \in A\} \cup \{\text{cons}(0, (F*B)*N). B \in A\}. \\ f' \langle B1, B2 \rangle \in \text{bij}(B1, B2) \\ \implies (\lambda B \in A. \text{snd}(\text{fst}((f' \langle \text{cons}(0, (F*B)*N), (F*B)*N \rangle) '0))) \in (\prod X \\ \in A. X)$ "
 $\langle proof \rangle$

lemma *AC9_AC1*: "*AC9* \implies *AC1*"
 $\langle proof \rangle$

end

theory *W01_AC*
imports *AC_Equiv*
begin

theorem *W01_AC1*: "*W01* \implies *AC1*"

$\langle proof \rangle$

lemma lemma1: " $\llbracket W01; \forall B \in A. \exists C \in D(B). P(C,B) \rrbracket \implies \exists f. \forall B \in A. P(f'B,B)$ "
 $\langle proof \rangle$

lemma lemma2_1: " $\llbracket \neg Finite(B); W01 \rrbracket \implies |B| + |B| \approx B$ "
 $\langle proof \rangle$

lemma lemma2_2:
" $f \in bij(D+D, B) \implies \{\{f'Inl(i), f'Inr(i)\}. i \in D\} \in Pow(Pow(B))$ "
 $\langle proof \rangle$

lemma lemma2_3:
" $f \in bij(D+D, B) \implies pairwise_disjoint(\{\{f'Inl(i), f'Inr(i)\}. i \in D\})$ "
 $\langle proof \rangle$

lemma lemma2_4:
" $f \in bij(D+D, B); 1 \leq n$
 $\implies sets_of_size_between(\{\{f'Inl(i), f'Inr(i)\}. i \in D\}, 2, succ(n))$ "
 $\langle proof \rangle$

lemma lemma2_5:
" $f \in bij(D+D, B) \implies \bigcup (\{\{f'Inl(i), f'Inr(i)\}. i \in D\}) = B$ "
 $\langle proof \rangle$

lemma lemma2:
" $\llbracket W01; \neg Finite(B); 1 \leq n \rrbracket$
 $\implies \exists C \in Pow(Pow(B)). pairwise_disjoint(C) \wedge$
 $sets_of_size_between(C, 2, succ(n)) \wedge$
 $\bigcup (C) = B$ "
 $\langle proof \rangle$

theorem W01_AC10: " $\llbracket W01; 1 \leq n \rrbracket \implies AC10(n)$ "
 $\langle proof \rangle$

end

theory Hartog
imports AC_Equiv
begin

definition

```

Hartog :: "i  $\Rightarrow$  i" where
  "Hartog(X)  $\equiv \mu$  i.  $\neg$  i  $\lesssim$  X"

lemma Ords_in_set: " $\forall$  a. Ord(a)  $\longrightarrow$  a  $\in$  X  $\implies$  P"
  <proof>

lemma Ord_lepoll_imp_ex_well_ord:
  "[[Ord(a); a  $\lesssim$  X]]
 $\implies \exists$  Y. Y  $\subseteq$  X  $\wedge (\exists$  R. well_ord(Y,R)  $\wedge$  ordertype(Y,R)=a)"
  <proof>

lemma Ord_lepoll_imp_eq_ordertype:
  "[[Ord(a); a  $\lesssim$  X]]  $\implies \exists$  Y. Y  $\subseteq$  X  $\wedge (\exists$  R. R  $\subseteq$  X*X  $\wedge$  ordertype(Y,R)=a)"
  <proof>

lemma Ords_lepoll_set_lemma:
  " $(\forall$  a. Ord(a)  $\longrightarrow$  a  $\lesssim$  X)  $\implies$ 
 $\forall$  a. Ord(a)  $\longrightarrow$ 
  a  $\in$  {b. Z  $\in$  Pow(X)*Pow(X*X),  $\exists$  Y R. Z= $\langle$ Y,R $\rangle \wedge$  ordertype(Y,R)=b}"
  <proof>

lemma Ords_lepoll_set: " $\forall$  a. Ord(a)  $\longrightarrow$  a  $\lesssim$  X  $\implies$  P"
  <proof>

lemma ex_Ord_not_lepoll: " $\exists$  a. Ord(a)  $\wedge \neg$  a  $\lesssim$  X"
  <proof>

lemma not_Hartog_lepoll_self: " $\neg$  Hartog(A)  $\lesssim$  A"
  <proof>

lemmas Hartog_lepoll_selfE = not_Hartog_lepoll_self [THEN notE]

lemma Ord_Hartog: "Ord(Hartog(A))"
  <proof>

lemma less_HartogE1: "[[i < Hartog(A);  $\neg$  i  $\lesssim$  A]]  $\implies$  P"
  <proof>

lemma less_HartogE: "[[i < Hartog(A); i  $\approx$  Hartog(A)]]  $\implies$  P"
  <proof>

lemma Card_Hartog: "Card(Hartog(A))"
  <proof>

end

theory HH
imports AC_Equiv Hartog

```

begin

definition

$HH :: "[i, i, i] \Rightarrow i"$ where
 $HH(f, x, a) \equiv \text{transrec}(a, \lambda b \ r. \text{let } z = x - (\bigcup c \in b. r'c)$
 $\text{in if } f'z \in \text{Pow}(z) - \{0\} \text{ then } f'z \text{ else }$
 $\{x\})"$

0.1 Lemmas useful in each of the three proofs

lemma $HH_def_satisfies_eq$:

$HH(f, x, a) = (\text{let } z = x - (\bigcup b \in a. HH(f, x, b))$
 $\text{in if } f'z \in \text{Pow}(z) - \{0\} \text{ then } f'z \text{ else } \{x\})"$
 $\langle proof \rangle$

lemma HH_values : $HH(f, x, a) \in \text{Pow}(x) - \{0\} \mid HH(f, x, a) = \{x\}"$

$\langle proof \rangle$

lemma $subset_imp_Diff_eq$:

$B \subseteq A \implies X - (\bigcup a \in A. P(a)) = X - (\bigcup a \in A - B. P(a)) - (\bigcup b \in B. P(b))"$
 $\langle proof \rangle$

lemma Ord_DiffE : $\llbracket c \in a - b; b < a \rrbracket \implies c = b \mid b < c \wedge c < a"$

$\langle proof \rangle$

lemma $Diff_UN_eq_self$: $(\bigwedge y. y \in A \implies P(y) = \{x\}) \implies x - (\bigcup y \in A. P(y)) = x"$

$\langle proof \rangle$

lemma HH_eq : $x - (\bigcup b \in a. HH(f, x, b)) = x - (\bigcup b \in a1. HH(f, x, b))$
 $\implies HH(f, x, a) = HH(f, x, a1)"$

$\langle proof \rangle$

lemma $HH_is_x_gt_too$: $\llbracket HH(f, x, b) = \{x\}; b < a \rrbracket \implies HH(f, x, a) = \{x\}"$

$\langle proof \rangle$

lemma $HH_subset_x_lt_too$:

$\llbracket HH(f, x, a) \in \text{Pow}(x) - \{0\}; b < a \rrbracket \implies HH(f, x, b) \in \text{Pow}(x) - \{0\}"$
 $\langle proof \rangle$

lemma $HH_subset_x_imp_subset_Diff_UN$:

$HH(f, x, a) \in \text{Pow}(x) - \{0\} \implies HH(f, x, a) \in \text{Pow}(x - (\bigcup b \in a. HH(f, x, b))) - \{0\}"$
 $\langle proof \rangle$

lemma $HH_eq_arg_lt$:

$\llbracket HH(f, x, v) = HH(f, x, w); HH(f, x, v) \in \text{Pow}(x) - \{0\}; v \in w \rrbracket \implies P"$
 $\langle proof \rangle$

lemma $HH_eq_imp_arg_eq$:

" $\llbracket HH(f, x, v) = HH(f, x, w); HH(f, x, w) \in Pow(x) - \{0\}; Ord(v); Ord(w) \rrbracket \implies v = w$ "
 $\langle proof \rangle$

lemma *HH_subset_x_imp_lepoll*:
 " $\llbracket HH(f, x, i) \in Pow(x) - \{0\}; Ord(i) \rrbracket \implies i \lesssim Pow(x) - \{0\}$ "
 $\langle proof \rangle$

lemma *HH_Hartog_is_x*: " $HH(f, x, Hartog(Pow(x) - \{0\})) = \{x\}$ "
 $\langle proof \rangle$

lemma *HH_Least_eq_x*: " $HH(f, x, \mu i. HH(f, x, i) = \{x\}) = \{x\}$ "
 $\langle proof \rangle$

lemma *less_Least_subset_x*:
 " $a \in (\mu i. HH(f, x, i) = \{x\}) \implies HH(f, x, a) \in Pow(x) - \{0\}$ "
 $\langle proof \rangle$

0.2 Lemmas used in the proofs of $AC1 \implies W02$ and $AC17 \implies AC1$

lemma *lam_Least_HH_inj_Pow*:
 " $(\lambda a \in (\mu i. HH(f, x, i) = \{x\}). HH(f, x, a))$
 $\in inj(\mu i. HH(f, x, i) = \{x\}, Pow(x) - \{0\})$ "
 $\langle proof \rangle$

lemma *lam_Least_HH_inj*:
 " $\forall a \in (\mu i. HH(f, x, i) = \{x\}). \exists z \in x. HH(f, x, a) = \{z\}$
 $\implies (\lambda a \in (\mu i. HH(f, x, i) = \{x\}). HH(f, x, a))$
 $\in inj(\mu i. HH(f, x, i) = \{x\}, \{\{y\}. y \in x\})$ "
 $\langle proof \rangle$

lemma *lam_surj_sing*:
 " $\llbracket x - (\bigcup a \in A. F(a)) = 0; \forall a \in A. \exists z \in x. F(a) = \{z\} \rrbracket$
 $\implies (\lambda a \in A. F(a)) \in surj(A, \{\{y\}. y \in x\})$ "
 $\langle proof \rangle$

lemma *not_emptyI2*: " $y \in Pow(x) - \{0\} \implies x \neq 0$ "
 $\langle proof \rangle$

lemma *f_subset_imp_HH_subset*:
 " $f'(x - (\bigcup j \in i. HH(f, x, j))) \in Pow(x - (\bigcup j \in i. HH(f, x, j))) - \{0\}$
 $\implies HH(f, x, i) \in Pow(x) - \{0\}$ "
 $\langle proof \rangle$

lemma *f_subsets_imp_UN_HH_eq_x*:
 " $\forall z \in Pow(x) - \{0\}. f'z \in Pow(z) - \{0\}$
 $\implies x - (\bigcup j \in (\mu i. HH(f, x, i) = \{x\}). HH(f, x, j)) = 0$ "
 $\langle proof \rangle$

lemma *HH_values2*: " $HH(f,x,i) = f'(x - (\bigcup j \in i. HH(f,x,j))) \mid HH(f,x,i)=\{x\}$ "
 <proof>

lemma *HH_subset_imp_eq*:
 " $HH(f,x,i): Pow(x)-\{0\} \implies HH(f,x,i)=f'(x - (\bigcup j \in i. HH(f,x,j)))$ "
 <proof>

lemma *f_sing_imp_HH_sing*:
 " $\llbracket f \in (Pow(x)-\{0\}) \rightarrow \{\{z\}. z \in x\};$
 $a \in (\mu i. HH(f,x,i)=\{x\}) \rrbracket \implies \exists z \in x. HH(f,x,a) = \{z\}$ "
 <proof>

lemma *f_sing_lam_bij*:
 " $\llbracket x - (\bigcup j \in (\mu i. HH(f,x,i)=\{x\}). HH(f,x,j)) = 0;$
 $f \in (Pow(x)-\{0\}) \rightarrow \{\{z\}. z \in x\} \rrbracket$
 $\implies (\lambda a \in (\mu i. HH(f,x,i)=\{x\}). HH(f,x,a))$
 $\in \text{bij}(\mu i. HH(f,x,i)=\{x\}, \{\{y\}. y \in x\})$ "
 <proof>

lemma *lam_singI*:
 " $f \in (\prod X \in Pow(x)-\{0\}. F(X))$
 $\implies (\lambda X \in Pow(x)-\{0\}. \{f'X\}) \in (\prod X \in Pow(x)-\{0\}. \{\{z\}. z \in F(X)\})$ "
 <proof>

lemmas *bij_Least_HH_x* =
 comp_bij [OF f_sing_lam_bij [OF _ lam_singI]
 lam_sing_bij [THEN bij_converse_bij]]

0.3 The proof of $AC1 \implies W02$

lemma *bijection*:
 " $f \in (\prod X \in Pow(x) - \{0\}. X)$
 $\implies \exists g. g \in \text{bij}(x, \mu i. HH(\lambda X \in Pow(x)-\{0\}. \{f'X\}, x, i) = \{x\})$ "
 <proof>

lemma *AC1_W02*: " $AC1 \implies W02$ "
 <proof>

end

theory *AC15_W06*
imports *HH Cardinal_aux*
begin

lemma *lepoll_Sigma*: " $A \neq 0 \implies B \lesssim A * B$ "
 <proof>

lemma *cons_times_nat_not_Finite*:
 " $0 \notin A \implies \forall B \in \{\text{cons}(0, x * \text{nat}). x \in A\}. \neg \text{Finite}(B)$ "
 <proof>

lemma *lemma1*: " $\llbracket \bigcup (C) = A; a \in A \rrbracket \implies \exists B \in C. a \in B \wedge B \subseteq A$ "
 <proof>

lemma *lemma2*:
 " $\llbracket \text{pairwise_disjoint}(A); B \in A; C \in A; a \in B; a \in C \rrbracket \implies B = C$ "
 <proof>

lemma *lemma3*:
 " $\forall B \in \{\text{cons}(0, x * \text{nat}). x \in A\}. \text{pairwise_disjoint}(f' B) \wedge$
 $\text{sets_of_size_between}(f' B, 2, n) \wedge \bigcup (f' B) = B$
 $\implies \forall B \in A. \exists ! u. u \in f' \text{cons}(0, B * \text{nat}) \wedge u \subseteq \text{cons}(0, B * \text{nat}) \wedge$
 $0 \in u \wedge 2 \lesssim u \wedge u \lesssim n$ "
 <proof>

lemma *lemma4*: " $\llbracket A \lesssim i; \text{Ord}(i) \rrbracket \implies \{P(a). a \in A\} \lesssim i$ "
 <proof>

lemma *lemma5_1*:
 " $\llbracket B \in A; 2 \lesssim u(B) \rrbracket \implies (\lambda x \in A. \{fst(x). x \in u(x) - \{0\}\})' B \neq 0$ "
 <proof>

lemma *lemma5_2*:
 " $\llbracket B \in A; u(B) \subseteq \text{cons}(0, B * \text{nat}) \rrbracket$
 $\implies (\lambda x \in A. \{fst(x). x \in u(x) - \{0\}\})' B \subseteq B$ "
 <proof>

lemma *lemma5_3*:
 " $\llbracket n \in \text{nat}; B \in A; 0 \in u(B); u(B) \lesssim \text{succ}(n) \rrbracket$
 $\implies (\lambda x \in A. \{fst(x). x \in u(x) - \{0\}\})' B \lesssim n$ "
 <proof>

lemma *ex_fun_AC13_AC15*:
 " $\llbracket \forall B \in \{\text{cons}(0, x * \text{nat}). x \in A\}. \text{pairwise_disjoint}(f' B) \wedge$
 $\text{sets_of_size_between}(f' B, 2, \text{succ}(n)) \wedge \bigcup (f' B) = B;$
 $n \in \text{nat} \rrbracket$

$\implies \exists f. \forall B \in A. f'B \neq 0 \wedge f'B \subseteq B \wedge f'B \lesssim n$
 $\langle proof \rangle$

theorem *AC10_AC11*: " $\llbracket n \in \text{nat}; 1 \leq n; \text{AC10}(n) \rrbracket \implies \text{AC11}$ "
 $\langle proof \rangle$

theorem *AC11_AC12*: " $\text{AC11} \implies \text{AC12}$ "
 $\langle proof \rangle$

theorem *AC12_AC15*: " $\text{AC12} \implies \text{AC15}$ "
 $\langle proof \rangle$

lemma *OUN_eq_UN*: " $\text{Ord}(x) \implies (\bigcup a < x. F(a)) = (\bigcup a \in x. F(a))$ "
 $\langle proof \rangle$

lemma *AC15_W06_aux1*:
 $"\forall x \in \text{Pow}(A) - \{0\}. f'x \neq 0 \wedge f'x \subseteq x \wedge f'x \lesssim m$
 $\implies (\bigcup i < \mu x. \text{HH}(f, A, x) = \{A\}. \text{HH}(f, A, i)) = A"$
 $\langle proof \rangle$

lemma *AC15_W06_aux2*:
 $"\forall x \in \text{Pow}(A) - \{0\}. f'x \neq 0 \wedge f'x \subseteq x \wedge f'x \lesssim m$
 $\implies \forall x < (\mu x. \text{HH}(f, A, x) = \{A\}). \text{HH}(f, A, x) \lesssim m"$
 $\langle proof \rangle$

theorem *AC15_W06*: " $\text{AC15} \implies \text{W06}$ "
 $\langle proof \rangle$

theorem *AC10_AC13*: " $\llbracket n \in \text{nat}; 1 \leq n; AC10(n) \rrbracket \implies AC13(n)$ "
<proof>

lemma *AC1_AC13*: " $AC1 \implies AC13(1)$ "
<proof>

lemma *AC13_mono*: " $\llbracket m \leq n; AC13(m) \rrbracket \implies AC13(n)$ "
<proof>

theorem *AC13_AC14*: " $\llbracket n \in \text{nat}; 1 \leq n; AC13(n) \rrbracket \implies AC14$ "
<proof>

theorem AC14_AC15: " $AC14 \implies AC15$ "
 $\langle proof \rangle$

lemma lemma_aux: " $\llbracket A \neq 0; A \lesssim 1 \rrbracket \implies \exists a. A = \{a\}$ "
 $\langle proof \rangle$

lemma AC13_AC1_lemma:
 $\forall B \in A. f(B) \neq 0 \wedge f(B) \leq B \wedge f(B) \lesssim 1$
 $\implies (\lambda x \in A. \text{THE } y. f(x) = \{y\}) \in (\prod X \in A. X)$ "
 $\langle proof \rangle$

theorem AC13_AC1: " $AC13(1) \implies AC1$ "
 $\langle proof \rangle$

theorem AC11_AC14: " $AC11 \implies AC14$ "
 $\langle proof \rangle$

end

theory AC16_lemmas
imports AC_Equiv Hartog Cardinal_aux
begin

lemma cons_Diff_eq: " $a \notin A \implies \text{cons}(a, A) - \{a\} = A$ "
 $\langle proof \rangle$

lemma nat_1_lepoll_iff: " $1 \lesssim X \longleftrightarrow (\exists x. x \in X)$ "
 $\langle proof \rangle$

lemma eqpoll_1_iff_singleton: " $X \approx 1 \longleftrightarrow (\exists x. X = \{x\})$ "
 $\langle proof \rangle$

lemma cons_eqpoll_succ: " $\llbracket x \approx n; y \notin x \rrbracket \implies \text{cons}(y, x) \approx \text{succ}(n)$ "
 $\langle proof \rangle$

lemma subsets_eqpoll_1_eq: " $\{Y \in \text{Pow}(X). Y \approx 1\} = \{\{x\}. x \in X\}$ "

$\langle proof \rangle$

lemma eqpoll_RepFun_sing: " $X \approx \{\{x\}. x \in X\}$ "
 $\langle proof \rangle$

lemma subsets_eqpoll_1_eqpoll: " $\{Y \in Pow(X). Y \approx 1\} \approx X$ "
 $\langle proof \rangle$

lemma InfCard_Least_in:
" $\llbracket InfCard(x); y \subseteq x; y \approx succ(z) \rrbracket \implies (\mu i. i \in y) \in y$ "
 $\langle proof \rangle$

lemma subsets_lepoll_lemma1:
" $\llbracket InfCard(x); n \in nat \rrbracket$
 $\implies \{y \in Pow(x). y \approx succ(succ(n))\} \lesssim x * \{y \in Pow(x). y \approx succ(n)\}$ "
 $\langle proof \rangle$

lemma set_of_Ord_succ_Union: " $(\forall y \in z. Ord(y)) \implies z \subseteq succ(\bigcup(z))$ "
 $\langle proof \rangle$

lemma subset_not_mem: " $j \subseteq i \implies i \notin j$ "
 $\langle proof \rangle$

lemma succ_Union_not_mem:
" $(\bigwedge y. y \in z \implies Ord(y)) \implies succ(\bigcup(z)) \notin z$ "
 $\langle proof \rangle$

lemma Union_cons_eq_succ_Union:
" $\bigcup(cons(succ(\bigcup(z)), z)) = succ(\bigcup(z))$ "
 $\langle proof \rangle$

lemma Un_Ord_disj: " $\llbracket Ord(i); Ord(j) \rrbracket \implies i \cup j = i \mid i \cup j = j$ "
 $\langle proof \rangle$

lemma Union_eq_Un: " $x \in X \implies \bigcup(X) = x \cup \bigcup(X - \{x\})$ "
 $\langle proof \rangle$

lemma Union_in_lemma [rule_format]:
" $n \in nat \implies \forall z. (\forall y \in z. Ord(y)) \wedge z \approx n \wedge z \neq 0 \longrightarrow \bigcup(z) \in z$ "
 $\langle proof \rangle$

lemma Union_in: " $\llbracket \forall x \in z. Ord(x); z \approx n; z \neq 0; n \in nat \rrbracket \implies \bigcup(z) \in z$ "
 $\langle proof \rangle$

lemma succ_Union_in_x:
" $\llbracket InfCard(x); z \in Pow(x); z \approx n; n \in nat \rrbracket \implies succ(\bigcup(z)) \in x$ "
 $\langle proof \rangle$

lemma succ_lepoll_succ_succ:

```

      "[InfCard(x); n ∈ nat]
      ⇒ {y ∈ Pow(x). y≈succ(n)} ≲ {y ∈ Pow(x). y≈succ(succ(n))}"
    <proof>

lemma subsets_eqpoll_X:
  "[InfCard(X); n ∈ nat] ⇒ {Y ∈ Pow(X). Y≈succ(n)} ≈ X"
  <proof>

lemma image_vimage_eq:
  "[f ∈ surj(A,B); y ⊆ B] ⇒ f``(converse(f)``y) = y"
  <proof>

lemma vimage_image_eq: "[f ∈ inj(A,B); y ⊆ A] ⇒ converse(f)``(f``y)
= y"
  <proof>

lemma subsets_eqpoll:
  "A≈B ⇒ {Y ∈ Pow(A). Y≈n}≈{Y ∈ Pow(B). Y≈n}"
  <proof>

lemma W02_imp_ex_Card: "W02 ⇒ ∃ a. Card(a) ∧ X≈a"
  <proof>

lemma lepoll_infinite: "[X ≲ Y; ¬Finite(X)] ⇒ ¬Finite(Y)"
  <proof>

lemma infinite_Card_is_InfCard: "[¬Finite(X); Card(X)] ⇒ InfCard(X)"
  <proof>

lemma W02_infinite_subsets_eqpoll_X: "[W02; n ∈ nat; ¬Finite(X)]
  ⇒ {Y ∈ Pow(X). Y≈succ(n)}≈X"
  <proof>

lemma well_ord_imp_ex_Card: "well_ord(X,R) ⇒ ∃ a. Card(a) ∧ X≈a"
  <proof>

lemma well_ord_infinite_subsets_eqpoll_X:
  "[well_ord(X,R); n ∈ nat; ¬Finite(X)] ⇒ {Y ∈ Pow(X). Y≈succ(n)}≈X"
  <proof>

end

theory W02_AC16 imports AC_Equiv AC16_lemmas Cardinal_aux begin

definition
  recfunAC16 :: "[i,i,i,i] ⇒ i" where

```

```

"recfunAC16(f,h,i,a) =
  transrec2(i, 0,
    λg r. if (∃y ∈ r. h'g ⊆ y) then r
    else r ∪ {f'(μ i. h'g ⊆ f'i ∧
      (∀b<a. (h'b ⊆ f'i → (∀t ∈ r. ¬ h'b ⊆ t))))})"

```

lemma *recfunAC16_0*: "recfunAC16(f,h,0,a) = 0"
 <proof>

lemma *recfunAC16_succ*:
 "recfunAC16(f,h,succ(i),a) =
 (if (∃y ∈ recfunAC16(f,h,i,a). h' i ⊆ y) then recfunAC16(f,h,i,a)

 else recfunAC16(f,h,i,a) ∪
 {f' (μ j. h' i ⊆ f' j ∧
 (∀b<a. (h'b ⊆ f'j
 → (∀t ∈ recfunAC16(f,h,i,a). ¬ h'b ⊆ t))))})"
 <proof>

lemma *recfunAC16_Limit*: "Limit(i)
 ⇒ recfunAC16(f,h,i,a) = (⋃ j<i. recfunAC16(f,h,j,a))"
 <proof>

lemma *transrec2_mono_lemma* [rule_format]:
 "⟦∧g r. r ⊆ B(g,r); Ord(i)⟧
 ⇒ j<i → transrec2(j, 0, B) ⊆ transrec2(i, 0, B)"
 <proof>

lemma *transrec2_mono*:
 "⟦∧g r. r ⊆ B(g,r); j≤i⟧
 ⇒ transrec2(j, 0, B) ⊆ transrec2(i, 0, B)"
 <proof>

lemma *recfunAC16_mono*:
 "i≤j ⇒ recfunAC16(f, g, i, a) ⊆ recfunAC16(f, g, j, a)"
 <proof>

lemma lemma3_1:

" $\llbracket \forall y < x. \forall z < a. z < y \mid (\exists Y \in F(y). f(z) \leq Y) \longrightarrow (\exists ! Y. Y \in F(y) \wedge f(z) \leq Y) \rrbracket$;

$\forall i \ j. i \leq j \longrightarrow F(i) \subseteq F(j); j \leq i; i < x; z < a;$

$V \in F(i); f(z) \leq V; W \in F(j); f(z) \leq W \rrbracket$

$\implies V = W$ "

$\langle proof \rangle$

lemma lemma3:

" $\llbracket \forall y < x. \forall z < a. z < y \mid (\exists Y \in F(y). f(z) \leq Y) \longrightarrow (\exists ! Y. Y \in F(y) \wedge f(z) \leq Y) \rrbracket$;

$\forall i \ j. i \leq j \longrightarrow F(i) \subseteq F(j); i < x; j < x; z < a;$

$V \in F(i); f(z) \leq V; W \in F(j); f(z) \leq W \rrbracket$

$\implies V = W$ "

$\langle proof \rangle$

lemma lemma4:

" $\llbracket \forall y < x. F(y) \subseteq X \wedge$

$(\forall x < a. x < y \mid (\exists Y \in F(y). h(x) \subseteq Y) \longrightarrow$
 $(\exists ! Y. Y \in F(y) \wedge h(x) \subseteq Y));$

$x < a \rrbracket$

$\implies \forall y < x. \forall z < a. z < y \mid (\exists Y \in F(y). h(z) \subseteq Y) \longrightarrow$
 $(\exists ! Y. Y \in F(y) \wedge h(z) \subseteq Y)"$

$\langle proof \rangle$

lemma lemma5:

" $\llbracket \forall y < x. F(y) \subseteq X \wedge$

$(\forall x < a. x < y \mid (\exists Y \in F(y). h(x) \subseteq Y) \longrightarrow$
 $(\exists ! Y. Y \in F(y) \wedge h(x) \subseteq Y));$

$x < a; Limit(x); \forall i \ j. i \leq j \longrightarrow F(i) \subseteq F(j) \rrbracket$

$\implies (\bigcup_{x < x} F(x)) \subseteq X \wedge$

$(\forall xa < a. xa < x \mid (\exists x \in \bigcup_{x < x} F(x). h(xa) \subseteq x)$

$\longrightarrow (\exists ! Y. Y \in (\bigcup_{x < x} F(x)) \wedge h(xa) \subseteq Y))"$

$\langle proof \rangle$

lemma *dbl_Diff_eqpoll_Card*:
 " $\llbracket A \approx a; \text{Card}(a); \neg \text{Finite}(a); B \prec a; C \prec a \rrbracket \implies A - B - C \approx a$ "
 $\langle \text{proof} \rangle$

lemma *Finite_lespoll_infinite_Ord*:
 " $\llbracket \text{Finite}(X); \neg \text{Finite}(a); \text{Ord}(a) \rrbracket \implies X \prec a$ "
 $\langle \text{proof} \rangle$

lemma *Union_lespoll*:
 " $\llbracket \forall x \in X. x \lesssim n \wedge x \subseteq T; \text{well_ord}(T, R); X \lesssim b; b \prec a; \neg \text{Finite}(a); \text{Card}(a); n \in \text{nat} \rrbracket$
 $\implies \bigcup (X) \prec a$ "
 $\langle \text{proof} \rangle$

lemma *Un_sing_eq_cons*: " $A \cup \{a\} = \text{cons}(a, A)$ "
 $\langle \text{proof} \rangle$

lemma *Un_lepoll_succ*: " $A \lesssim B \implies A \cup \{a\} \lesssim \text{succ}(B)$ "
 $\langle \text{proof} \rangle$

lemma *Diff_UN_succ_empty*: " $\text{Ord}(a) \implies F(a) - (\bigcup b \prec \text{succ}(a). F(b)) = 0$ "
 $\langle \text{proof} \rangle$

lemma *Diff_UN_succ_subset*: " $\text{Ord}(a) \implies F(a) \cup X - (\bigcup b \prec \text{succ}(a). F(b)) \subseteq X$ "
 $\langle \text{proof} \rangle$

lemma *recfunAC16_Diff_lepoll_1*:
 " $\text{Ord}(x)$
 $\implies \text{recfunAC16}(f, g, x, a) - (\bigcup i \prec x. \text{recfunAC16}(f, g, i, a)) \lesssim 1$ "
 $\langle \text{proof} \rangle$

lemma *in_Least_Diff*:
 " $\llbracket z \in F(x); \text{Ord}(x) \rrbracket$
 $\implies z \in F(\mu i. z \in F(i)) - (\bigcup j \prec (\mu i. z \in F(i)). F(j))$ "
 $\langle \text{proof} \rangle$

lemma *Least_eq_imp_ex*:
 "[($\mu i. w \in F(i)$) = ($\mu i. z \in F(i)$);
 $w \in (\bigcup i < a. F(i))$; $z \in (\bigcup i < a. F(i))$]
 $\implies \exists b < a. w \in (F(b) - (\bigcup c < b. F(c))) \wedge z \in (F(b) - (\bigcup c < b. F(c)))$ "
 <proof>

lemma *two_in_lepoll_1*: "[$A \lesssim 1$; $a \in A$; $b \in A$] $\implies a=b$ "
 <proof>

lemma *UN_lepoll_index*:
 "[$\forall i < a. F(i) - (\bigcup j < i. F(j)) \lesssim 1$; $\text{Limit}(a)$]
 $\implies (\bigcup x < a. F(x)) \lesssim a$ "
 <proof>

lemma *recfunAC16_lepoll_index*: " $\text{Ord}(y) \implies \text{recfunAC16}(f, h, y, a) \lesssim y$ "
 <proof>

lemma *Union_recfunAC16_lesspoll*:
 "[$\text{recfunAC16}(f, g, y, a) \subseteq \{X \in \text{Pow}(A). X \approx n\}$;
 $A \approx a$; $y < a$; $\neg \text{Finite}(a)$; $\text{Card}(a)$; $n \in \text{nat}$]
 $\implies \bigcup (\text{recfunAC16}(f, g, y, a)) \prec a$ "
 <proof>

lemma *dbl_Diff_eqpoll*:
 "[$\text{recfunAC16}(f, h, y, a) \subseteq \{X \in \text{Pow}(A) . X \approx \text{succ}(k \# m)\}$;
 $\text{Card}(a)$; $\neg \text{Finite}(a)$; $A \approx a$;
 $k \in \text{nat}$; $y < a$;
 $h \in \text{bij}(a, \{Y \in \text{Pow}(A). Y \approx \text{succ}(k)\})$]
 $\implies A - \bigcup (\text{recfunAC16}(f, h, y, a)) - h' y \approx a$ "
 <proof>

lemmas *disj_Un_eqpoll_nat_sum* =
 eqpoll_trans [THEN eqpoll_trans,
 OF disj_Un_eqpoll_sum sum_eqpoll_cong nat_sum_eqpoll_sum]

lemma *Un_in_Collect*: "[$x \in \text{Pow}(A - B - h'i)$; $x \approx m$;
 $h \in \text{bij}(a, \{x \in \text{Pow}(A) . x \approx k\})$; $i < a$; $k \in \text{nat}$; $m \in \text{nat}$]
 $\implies h' i \cup x \in \{x \in \text{Pow}(A) . x \approx k \# m\}$ "
 <proof>

lemma lemma6:

$$\llbracket \forall y < \text{succ}(j). F(y) \leq X \wedge (\forall x < a. x < y \mid P(x, y) \longrightarrow Q(x, y)); \text{succ}(j) < a \rrbracket$$

$$\implies F(j) \leq X \wedge (\forall x < a. x < j \mid P(x, j) \longrightarrow Q(x, j))$$
 $\langle \text{proof} \rangle$

lemma lemma7:

$$\llbracket \forall x < a. x < j \mid P(x, j) \longrightarrow Q(x, j); \text{succ}(j) < a \rrbracket$$

$$\implies P(j, j) \longrightarrow (\forall x < a. x \leq j \mid P(x, j) \longrightarrow Q(x, j))$$
 $\langle \text{proof} \rangle$

lemma ex_subset_eqpoll:

$$\llbracket A \approx a; \neg \text{Finite}(a); \text{Ord}(a); m \in \text{nat} \rrbracket \implies \exists X \in \text{Pow}(A). X \approx_m$$
 $\langle \text{proof} \rangle$

lemma subset_Un_disjoint: $\llbracket A \subseteq B \cup C; A \cap C = 0 \rrbracket \implies A \subseteq B$
 $\langle \text{proof} \rangle$

lemma Int_empty:

$$\llbracket X \in \text{Pow}(A - \bigcup (B) - C); T \in B; F \subseteq T \rrbracket \implies F \cap X = 0$$
 $\langle \text{proof} \rangle$

lemma subset_imp_eq_lemma:

$$m \in \text{nat} \implies \forall A B. A \subseteq B \wedge m \lesssim A \wedge B \lesssim m \longrightarrow A=B$$
 $\langle \text{proof} \rangle$

lemma subset_imp_eq: $\llbracket A \subseteq B; m \lesssim A; B \lesssim m; m \in \text{nat} \rrbracket \implies A=B$
 $\langle \text{proof} \rangle$

lemma bij_imp_arg_eq:

$$\llbracket f \in \text{bij}(a, \{Y \in X. Y \approx \text{succ}(k)\}); k \in \text{nat}; f' b \subseteq f' y; b < a; y < a \rrbracket$$

$\implies b=y$ "
 $\langle proof \rangle$

lemma ex_next_set:

$$\begin{aligned} & \llbracket \text{recfunAC16}(f, h, y, a) \subseteq \{X \in \text{Pow}(A) \mid X \approx \text{succ}(k \# m)\}; \\ & \quad \text{Card}(a); \neg \text{Finite}(a); A \approx a; \\ & \quad k \in \text{nat}; m \in \text{nat}; y < a; \\ & \quad h \in \text{bij}(a, \{Y \in \text{Pow}(A) \mid Y \approx \text{succ}(k)\}); \\ & \quad \neg (\exists Y \in \text{recfunAC16}(f, h, y, a). h'y \subseteq Y) \rrbracket \\ \implies & \exists X \in \{Y \in \text{Pow}(A) \mid Y \approx \text{succ}(k \# m)\}. h'y \subseteq X \wedge \\ & \quad (\forall b < a. h'b \subseteq X \longrightarrow \\ & \quad (\forall T \in \text{recfunAC16}(f, h, y, a). \neg h'b \subseteq T)) \rrbracket \end{aligned}$$

 $\langle proof \rangle$

lemma ex_next_Ord:

$$\begin{aligned} & \llbracket \text{recfunAC16}(f, h, y, a) \subseteq \{X \in \text{Pow}(A) \mid X \approx \text{succ}(k \# m)\}; \\ & \quad \text{Card}(a); \neg \text{Finite}(a); A \approx a; \\ & \quad k \in \text{nat}; m \in \text{nat}; y < a; \\ & \quad h \in \text{bij}(a, \{Y \in \text{Pow}(A) \mid Y \approx \text{succ}(k)\}); \\ & \quad f \in \text{bij}(a, \{Y \in \text{Pow}(A) \mid Y \approx \text{succ}(k \# m)\}); \\ & \quad \neg (\exists Y \in \text{recfunAC16}(f, h, y, a). h'y \subseteq Y) \rrbracket \\ \implies & \exists c < a. h'y \subseteq f'c \wedge \\ & \quad (\forall b < a. h'b \subseteq f'c \longrightarrow \\ & \quad (\forall T \in \text{recfunAC16}(f, h, y, a). \neg h'b \subseteq T)) \rrbracket \end{aligned}$$

 $\langle proof \rangle$

lemma lemma8:

$$\begin{aligned} & \llbracket \forall x < a. x < j \mid (\exists xa \in F(j). P(x, xa)) \\ & \quad \longrightarrow (\exists ! Y. Y \in F(j) \wedge P(x, Y)); F(j) \subseteq X; \\ & \quad L \in X; P(j, L) \wedge (\forall x < a. P(x, L) \longrightarrow (\forall xa \in F(j). \neg P(x, xa))) \rrbracket \\ \implies & F(j) \cup \{L\} \subseteq X \wedge \\ & \quad (\forall x < a. x \leq j \mid (\exists xa \in (F(j) \cup \{L\}). P(x, xa)) \longrightarrow \\ & \quad (\exists ! Y. Y \in (F(j) \cup \{L\}) \wedge P(x, Y))) \rrbracket \end{aligned}$$

 $\langle proof \rangle$

```

lemma main_induct:
  "[[b < a; f ∈ bij(a, {Y ∈ Pow(A) . Y ≈ succ(k #+ m)}});
   h ∈ bij(a, {Y ∈ Pow(A) . Y ≈ succ(k)}});
   ¬Finite(a); Card(a); A ≈ a; k ∈ nat; m ∈ nat]]
  ⇒ recfunAC16(f, h, b, a) ⊆ {X ∈ Pow(A) . X ≈ succ(k #+ m)} ∧
    (∀ x < a. x < b | (∃ Y ∈ recfunAC16(f, h, b, a). h ' x ⊆ Y) →

    (∃ ! Y. Y ∈ recfunAC16(f, h, b, a) ∧ h ' x ⊆ Y))"
<proof>

```

```

lemma lemma_simp_induct:
  "[[∀ b. b < a → F(b) ⊆ S ∧ (∀ x < a. (x < b | (∃ Y ∈ F(b). f ' x ⊆ Y))
    → (∃ ! Y. Y ∈ F(b) ∧ f ' x ⊆ Y));
   f ∈ a → f ' (a); Limit(a);
   ∀ i j. i ≤ j → F(i) ⊆ F(j)]]
  ⇒ (⋃ j < a. F(j)) ⊆ S ∧
    (∀ x ∈ f ' a. ∃ ! Y. Y ∈ (⋃ j < a. F(j)) ∧ x ⊆ Y)"
<proof>

```

```

theorem W02_AC16: "[[W02; 0 < m; k ∈ nat; m ∈ nat]] ⇒ AC16(k #+ m, k)"
<proof>

```

end

```

theory AC16_W04
imports AC16_lemmas
begin

```

```

lemma lemma1:
  "[[Finite(A); 0 < m; m ∈ nat]]
  ⇒ ∃ a f. Ord(a) ∧ domain(f) = a ∧

```

$\langle proof \rangle$ $(\bigcup b < a. f' b) = A \wedge (\forall b < a. f' b \lesssim m)$

lemmas *well_ord_paired* = *paired_bij* [THEN *bij_is_inj*, THEN *well_ord_rvimage*]

lemma *lepoll_trans1*: " $\llbracket A \lesssim B; \neg A \lesssim C \rrbracket \implies \neg B \lesssim C$ "
 $\langle proof \rangle$

lemmas *lepoll_paired* = *paired_eqpoll* [THEN *eqpoll_sym*, THEN *eqpoll_imp_lepoll*]

lemma *lemma2*: " $\exists y R. \text{well_ord}(y, R) \wedge x \cap y = 0 \wedge \neg y \lesssim z \wedge \neg \text{Finite}(y)$ "
 $\langle proof \rangle$

lemma *infinite_Un*: " $\neg \text{Finite}(B) \implies \neg \text{Finite}(A \cup B)$ "
 $\langle proof \rangle$

lemma *succ_not_lepoll_lemma*:
 $\llbracket \neg (\exists x \in A. f' x = y); f \in \text{inj}(A, B); y \in B \rrbracket$
 $\implies (\lambda a \in \text{succ}(A). \text{if}(a=A, y, f' a)) \in \text{inj}(\text{succ}(A), B)$
 $\langle proof \rangle$

lemma *succ_not_lepoll_imp_eqpoll*: " $\llbracket \neg A \approx B; A \lesssim B \rrbracket \implies \text{succ}(A) \lesssim B$ "
 $\langle proof \rangle$

```

lemmas ordertype_eqpoll =
  ordermap_bij [THEN exI [THEN eqpoll_def [THEN def_imp_iff, THEN
iffD2]]]

```

```

lemma cons_cons_subset:
  "[a ⊆ y; b ∈ y-a; u ∈ x] ⇒ cons(b, cons(u, a)) ∈ Pow(x ∪ y)"
<proof>

```

```

lemma cons_cons_eqpoll:
  "[a ≈ k; a ⊆ y; b ∈ y-a; u ∈ x; x ∩ y = 0]
  ⇒ cons(b, cons(u, a)) ≈ succ(succ(k))"
<proof>

```

```

lemma set_eq_cons:
  "[succ(k) ≈ A; k ≈ B; B ⊆ A; a ∈ A-B; k ∈ nat] ⇒ A = cons(a,
B)"
<proof>

```

```

lemma cons_eqE: "[cons(x,a) = cons(y,a); x ∉ a] ⇒ x = y "
<proof>

```

```

lemma eq_imp_Int_eq: "A = B ⇒ A ∩ C = B ∩ C"
<proof>

```

```

lemma eqpoll_sum_imp_Diff_lepoll_lemma [rule_format]:
  "[k ∈ nat; m ∈ nat]
  ⇒ ∀ A B. A ≈ k #+ m ∧ k ≲ B ∧ B ⊆ A → A-B ≲ m"
<proof>

```

```

lemma eqpoll_sum_imp_Diff_lepoll:
  "[A ≈ succ(k #+ m); B ⊆ A; succ(k) ≲ B; k ∈ nat; m ∈ nat]
  ⇒ A-B ≲ m"
<proof>

```

```

lemma eqpoll_sum_imp_Diff_eqpoll_lemma [rule_format]:
  "[k ∈ nat; m ∈ nat]
  ⇒ ∀ A B. A ≈ k #+ m ∧ k ≈ B ∧ B ⊆ A → A-B ≈ m"
<proof>

```

```

lemma eqpoll_sum_imp_Diff_eqpoll:
  "[[A ≈ succ(k #+ m); B ⊆ A; succ(k) ≈ B; k ∈ nat; m ∈ nat]]
  ⇒ A-B ≈ m"
⟨proof⟩

lemma subsets_lepoll_0_eq_unit: "{x ∈ Pow(X). x ≲ 0} = {0}"
⟨proof⟩

lemma subsets_lepoll_succ:
  "n ∈ nat ⇒ {z ∈ Pow(y). z ≲ succ(n)} =
    {z ∈ Pow(y). z ≲ n} ∪ {z ∈ Pow(y). z ≈ succ(n)}"
⟨proof⟩

lemma Int_empty:
  "n ∈ nat ⇒ {z ∈ Pow(y). z ≲ n} ∩ {z ∈ Pow(y). z ≈ succ(n)} =
  0"
⟨proof⟩

locale AC16 =
  fixes x and y and k and l and m and t_n and R and MM and LL and
  GG and s
  defines k_def:      "k ≡ succ(l)"
    and MM_def:      "MM ≡ {v ∈ t_n. succ(k) ≲ v ∩ y}"
    and LL_def:      "LL ≡ {v ∩ y. v ∈ MM}"
    and GG_def:      "GG ≡ λv ∈ LL. (THE w. w ∈ MM ∧ v ⊆ w) - v"
    and s_def:       "s(u) ≡ {v ∈ t_n. u ∈ v ∧ k ≲ v ∩ y}"
  assumes all_ex:     "∀z ∈ {z ∈ Pow(x ∪ y) . z ≈ succ(k)}.
    ∃ ! w. w ∈ t_n ∧ z ⊆ w"
    and disjoint[iff]: "x ∩ y = 0"
    and "includes":   "t_n ⊆ {v ∈ Pow(x ∪ y). v ≈ succ(k #+ m)}"
    and WO_R[iff]:    "well_ord(y,R)"
    and lnat[iff]:     "l ∈ nat"
    and mnat[iff]:     "m ∈ nat"
    and mpos[iff]:     "0 < m"
    and Infinite[iff]: "¬ Finite(y)"
    and noLepoll:     "¬ y ≲ {v ∈ Pow(x). v ≈ m}"
begin

lemma knat [iff]: "k ∈ nat"
⟨proof⟩

```


lemma *Diff_Finite_eqpoll*: " $\llbracket l \approx a; a \subseteq y \rrbracket \implies y - a \approx y$ "
 $\langle proof \rangle$

lemma *s_subset*: " $s(u) \subseteq t_n$ "
 $\langle proof \rangle$

lemma *sI*:
 $\llbracket w \in t_n; cons(b, cons(u, a)) \subseteq w; a \subseteq y; b \in y - a; l \approx a \rrbracket$
 $\implies w \in s(u)$
 $\langle proof \rangle$

lemma *in_s_imp_u_in*: " $v \in s(u) \implies u \in v$ "
 $\langle proof \rangle$

lemma *ex1_superset_a*:
 $\llbracket l \approx a; a \subseteq y; b \in y - a; u \in x \rrbracket$
 $\implies \exists ! c. c \in s(u) \wedge a \subseteq c \wedge b \in c$
 $\langle proof \rangle$

lemma *the_eq_cons*:
 $\llbracket \forall v \in s(u). succ(l) \approx v \cap y;$
 $l \approx a; a \subseteq y; b \in y - a; u \in x \rrbracket$
 $\implies (THE c. c \in s(u) \wedge a \subseteq c \wedge b \in c) \cap y = cons(b, a)$
 $\langle proof \rangle$

lemma *y_lepoll_subset_s*:
 $\llbracket \forall v \in s(u). succ(l) \approx v \cap y;$
 $l \approx a; a \subseteq y; u \in x \rrbracket$
 $\implies y \lesssim \{v \in s(u). a \subseteq v\}$
 $\langle proof \rangle$

lemma *x_imp_not_y* [*dest*]: " $a \in x \implies a \notin y$ "
 $\langle proof \rangle$

lemma *w_Int_eq_w_Diff*:
 $w \subseteq x \cup y \implies w \cap (x - \{u\}) = w - cons(u, w \cap y)$
 $\langle proof \rangle$

lemma *w_Int_eqpoll_m*:
 "⟦ $w \in \{v \in s(u). a \subseteq v\};$
 $l \approx a; u \in x;$
 $\forall v \in s(u). \text{succ}(l) \approx v \cap y$ ⟧
 $\implies w \cap (x - \{u\}) \approx m$ "
 <proof>

lemma *eqpoll_m_not_empty*: " $a \approx m \implies a \neq 0$ "
 <proof>

lemma *cons_cons_in*:
 "⟦ $z \in xa \cap (x - \{u\}); l \approx a; a \subseteq y; u \in x$ ⟧
 $\implies \exists ! w. w \in t_n \wedge \text{cons}(z, \text{cons}(u, a)) \subseteq w$ "
 <proof>

lemma *subset_s_lepoll_w*:
 "⟦ $\forall v \in s(u). \text{succ}(l) \approx v \cap y; a \subseteq y; l \approx a; u \in x$ ⟧
 $\implies \{v \in s(u). a \subseteq v\} \lesssim \{v \in \text{Pow}(x). v \approx m\}$ "
 <proof>

lemma *well_ord_subsets_eqpoll_n*:
 " $n \in \text{nat} \implies \exists S. \text{well_ord}(\{z \in \text{Pow}(y) . z \approx \text{succ}(n)\}, S)$ "
 <proof>

lemma *well_ord_subsets_lepoll_n*:
 " $n \in \text{nat} \implies \exists R. \text{well_ord}(\{z \in \text{Pow}(y). z \lesssim n\}, R)$ "
 <proof>

lemma *LL_subset*: " $LL \subseteq \{z \in \text{Pow}(y). z \lesssim \text{succ}(k \# + m)\}$ "
 <proof>

lemma *well_ord_LL*: " $\exists S. \text{well_ord}(LL, S)$ "
 <proof>

lemma unique_superset_in_MM:
 $"v \in LL \implies \exists ! w. w \in MM \wedge v \subseteq w"$
 $\langle proof \rangle$

lemma Int_in_LL: $"w \in MM \implies w \cap y \in LL"$
 $\langle proof \rangle$

lemma in_LL_eq_Int:
 $"v \in LL \implies v = (THE x. x \in MM \wedge v \subseteq x) \cap y"$
 $\langle proof \rangle$

lemma unique_superset1: $"a \in LL \implies (THE x. x \in MM \wedge a \subseteq x) \in MM"$
 $\langle proof \rangle$

lemma the_in_MM_subset:
 $"v \in LL \implies (THE x. x \in MM \wedge v \subseteq x) \subseteq x \cup y"$
 $\langle proof \rangle$

lemma GG_subset: $"v \in LL \implies GG \text{ ` } v \subseteq x"$
 $\langle proof \rangle$

lemma nat_lepoll_ordertype: $"nat \lesssim ordertype(y, R)"$
 $\langle proof \rangle$

lemma ex_subset_eqpoll_n: $"n \in nat \implies \exists z. z \subseteq y \wedge n \approx z"$
 $\langle proof \rangle$

lemma exists_proper_in_s: $"u \in x \implies \exists v \in s(u). succ(k) \lesssim v \cap y"$
 $\langle proof \rangle$

lemma exists_in_MM: $"u \in x \implies \exists w \in MM. u \in w"$
 $\langle proof \rangle$

lemma exists_in_LL: $"u \in x \implies \exists w \in LL. u \in GG \text{ ` } w"$
 $\langle proof \rangle$

lemma OUN_eq_x: $"well_ord(LL, S) \implies$

$\langle proof \rangle$ $(\bigcup b < \text{ordertype}(LL, S). GG \text{ ' } (\text{converse}(\text{ordermap}(LL, S)) \text{ ' } b)) = x$

lemma `in_MM_eqpoll_n`: " $w \in MM \implies w \approx \text{succ}(k \#+ m)$ "
 $\langle proof \rangle$

lemma `in_LL_eqpoll_n`: " $w \in LL \implies \text{succ}(k) \lesssim w$ "
 $\langle proof \rangle$

lemma `in_LL`: " $w \in LL \implies w \subseteq (\text{THE } x. x \in MM \wedge w \subseteq x)$ "
 $\langle proof \rangle$

lemma `all_in_lepoll_m`:
 $\text{"well_ord}(LL, S) \implies$
 $\forall b < \text{ordertype}(LL, S). GG \text{ ' } (\text{converse}(\text{ordermap}(LL, S)) \text{ ' } b) \lesssim m$
 $\langle proof \rangle$

lemma `"conclusion"`:
 $\text{"}\exists a f. \text{Ord}(a) \wedge \text{domain}(f) = a \wedge (\bigcup b < a. f \text{ ' } b) = x \wedge (\forall b < a. f \text{ ' } b \lesssim m)\text{"}$
 $\langle proof \rangle$

end

theorem `AC16_W04`:
 $\text{"}\llbracket AC_Equiv.AC16(k \#+ m, k); 0 < k; 0 < m; k \in \text{nat}; m \in \text{nat} \rrbracket \implies$
 $W04(m)\text{"}$
 $\langle proof \rangle$

end

theory `AC17_AC1`
imports `HH`
begin

lemma `AC0_AC1_lemma`: " $\llbracket f: (\prod X \in A. X); D \subseteq A \rrbracket \implies \exists g. g: (\prod X \in D.$

$X)$ "
 $\langle proof \rangle$

lemma $AC0_AC1$: " $AC0 \implies AC1$ "
 $\langle proof \rangle$

lemma $AC1_AC0$: " $AC1 \implies AC0$ "
 $\langle proof \rangle$

lemma $AC1_AC17_lemma$: " $f \in (\prod X \in Pow(A) - \{0\}. X) \implies f \in (Pow(A) - \{0\} \rightarrow A)$ "
 $\langle proof \rangle$

lemma $AC1_AC17$: " $AC1 \implies AC17$ "
 $\langle proof \rangle$

lemma $UN_eq_imp_well_ord$:
 $\llbracket x - (\bigcup j \in \mu i. HH(\lambda X \in Pow(x) - \{0\}. \{f'X\}, x, i) = \{x\}. HH(\lambda X \in Pow(x) - \{0\}. \{f'X\}, x, j)) = 0;$
 $f \in Pow(x) - \{0\} \rightarrow x \rrbracket$
 $\implies \exists r. well_ord(x, r)$ "
 $\langle proof \rangle$

lemma $not_AC1_imp_ex$:
 $\llbracket \neg AC1 \implies \exists A. \forall f \in Pow(A) - \{0\} \rightarrow A. \exists u \in Pow(A) - \{0\}. f'u \notin u \rrbracket$
 $\langle proof \rangle$

lemma $AC17_AC1_aux1$:
 $\llbracket \forall f \in Pow(x) - \{0\} \rightarrow x. \exists u \in Pow(x) - \{0\}. f'u \notin u;$
 $\exists f \in Pow(x) - \{0\} \rightarrow x.$
 $x - (\bigcup a \in (\mu i. HH(\lambda X \in Pow(x) - \{0\}. \{f'X\}, x, i) = \{x\}). HH(\lambda X \in Pow(x) - \{0\}. \{f'X\}, x, a)) = 0 \rrbracket$
 $\implies P$ "
 $\langle proof \rangle$

lemma AC17_AC1_aux2:

$$\neg (\exists f \in \text{Pow}(x) - \{0\} \rightarrow x. x - F(f) = 0)$$

$$\implies (\lambda f \in \text{Pow}(x) - \{0\} \rightarrow x. x - F(f))$$

$$\in (\text{Pow}(x) - \{0\} \rightarrow x) \rightarrow \text{Pow}(x) - \{0\}"$$
 $\langle \text{proof} \rangle$

lemma AC17_AC1_aux3:

$$"f'Z \in Z; Z \in \text{Pow}(x) - \{0\}"$$

$$\implies (\lambda X \in \text{Pow}(x) - \{0\}. \{f'X\}'Z \in \text{Pow}(Z) - \{0\}"$$
 $\langle \text{proof} \rangle$

lemma AC17_AC1_aux4:

$$"\exists f \in F. f'((\lambda f \in F. Q(f))'f) \in (\lambda f \in F. Q(f))'f"$$

$$\implies \exists f \in F. f'Q(f) \in Q(f)"$$
 $\langle \text{proof} \rangle$

lemma AC17_AC1: "AC17 \implies AC1"
 $\langle \text{proof} \rangle$

lemma AC1_AC2_aux1:

$$"f: (\prod X \in A. X); B \in A; 0 \notin A \implies \{f'B\} \subseteq B \cap \{f'C. C \in A}"$$
 $\langle \text{proof} \rangle$

lemma AC1_AC2_aux2:

$$"pairwise_disjoint(A); B \in A; C \in A; D \in B; D \in C \implies f'B$$

$$= f'C"$$
 $\langle \text{proof} \rangle$

lemma AC1_AC2: "AC1 \implies AC2"
 $\langle \text{proof} \rangle$

lemma AC2_AC1_aux1: " $0 \notin A \implies 0 \notin \{B * \{B\}. B \in A\}"$
 $\langle \text{proof} \rangle$

lemma AC2_AC1_aux2: " $X * \{X\} \cap C = \{y\}; X \in A$
 $\implies (\text{THE } y. X * \{X\} \cap C = \{y\}): X * A"$
 $\langle \text{proof} \rangle$

lemma *AC2_AC1_aux3*:

$$\begin{aligned} & \text{"}\forall D \in \{E * \{E\}. E \in A\}. \exists y. D \cap C = \{y\} \\ & \implies (\lambda x \in A. \text{fst}(\text{THE } z. (x * \{x\} \cap C = \{z\}))) \in (\prod X \in A. X) \text{"} \\ & \langle \text{proof} \rangle \end{aligned}$$

lemma *AC2_AC1*: "AC2 \implies AC1"
 $\langle \text{proof} \rangle$

lemma *empty_notin_images*: "0 $\notin \{R' \{x\}. x \in \text{domain}(R)\}"$
 $\langle \text{proof} \rangle$

lemma *AC1_AC4*: "AC1 \implies AC4"
 $\langle \text{proof} \rangle$

lemma *AC4_AC3_aux1*: " $f \in A \rightarrow B \implies (\bigcup z \in A. \{z\} * f' z) \subseteq A * \bigcup (B)$ "
 $\langle \text{proof} \rangle$

lemma *AC4_AC3_aux2*: " $\text{domain}(\bigcup z \in A. \{z\} * f(z)) = \{a \in A. f(a) \neq 0\}$ "
 $\langle \text{proof} \rangle$

lemma *AC4_AC3_aux3*: " $x \in A \implies (\bigcup z \in A. \{z\} * f(z))' \{x\} = f(x)$ "
 $\langle \text{proof} \rangle$

lemma *AC4_AC3*: "AC4 \implies AC3"
 $\langle \text{proof} \rangle$

lemma *AC3_AC1_lemma*:

$$\text{"}b \notin A \implies (\prod x \in \{a \in A. \text{id}(A)'a \neq b\}. \text{id}(A)'x) = (\prod x \in A. x) \text{"}$$
 $\langle \text{proof} \rangle$

lemma *AC3_AC1*: "AC3 \implies AC1"
 $\langle \text{proof} \rangle$

lemma *AC4_AC5*: " $AC4 \implies AC5$ "
 $\langle proof \rangle$

lemma *AC5_AC4_aux1*: " $R \subseteq A*B \implies (\lambda x \in R. fst(x)) \in R \rightarrow A$ "
 $\langle proof \rangle$

lemma *AC5_AC4_aux2*: " $R \subseteq A*B \implies range(\lambda x \in R. fst(x)) = domain(R)$ "
 $\langle proof \rangle$

lemma *AC5_AC4_aux3*: " $\llbracket \exists f \in A \rightarrow C. P(f, domain(f)); A=B \rrbracket \implies \exists f \in B \rightarrow C. P(f, B)$ "
 $\langle proof \rangle$

lemma *AC5_AC4_aux4*: " $\llbracket R \subseteq A*B; g \in C \rightarrow R; \forall x \in C. (\lambda z \in R. fst(z)) (g'x) = x \rrbracket \implies (\lambda x \in C. snd(g'x)) : (\prod x \in C. R' \{x\})$ "
 $\langle proof \rangle$

lemma *AC5_AC4*: " $AC5 \implies AC4$ "
 $\langle proof \rangle$

lemma *AC1_iff_AC6*: " $AC1 \longleftrightarrow AC6$ "
 $\langle proof \rangle$

end

theory *AC18_AC19*
imports *AC_Equiv*
begin

definition
 $uu :: "i \Rightarrow i"$ **where**
 $uu(a) \equiv \{c \cup \{0\}. c \in a\}$

lemma *PROD_subsets*:

" $\llbracket f \in (\prod b \in \{P(a). a \in A\}. b); \forall a \in A. P(a) \leq Q(a) \rrbracket$
 $\implies (\lambda a \in A. f'P(a)) \in (\prod a \in A. Q(a))$ "

$\langle proof \rangle$

lemma *lemma_AC18*:

" $\llbracket \forall A. 0 \notin A \longrightarrow (\exists f. f \in (\prod X \in A. X)); A \neq 0 \rrbracket$
 $\implies (\bigcap a \in A. \bigcup b \in B(a). X(a, b)) \subseteq$
 $(\bigcup f \in \prod a \in A. B(a). \bigcap a \in A. X(a, f'a))$ "

$\langle proof \rangle$

lemma *AC1_AC18*: "*AC1* \implies *PROP AC18*"

$\langle proof \rangle$

theorem (*in AC18*) *AC19*

$\langle proof \rangle$

lemma *RepRep_conj*:

" $\llbracket A \neq 0; 0 \notin A \rrbracket \implies \{uu(a). a \in A\} \neq 0 \wedge 0 \notin \{uu(a). a \in A\}$ "

$\langle proof \rangle$

lemma *lemma1_1*: " $\llbracket c \in a; x = c \cup \{0\}; x \notin a \rrbracket \implies x - \{0\} \in a$ "

$\langle proof \rangle$

lemma *lemma1_2*:

" $\llbracket f'(uu(a)) \notin a; f \in (\prod B \in \{uu(a). a \in A\}. B); a \in A \rrbracket$
 $\implies f'(uu(a)) - \{0\} \in a$ "

$\langle proof \rangle$

lemma *lemma1*: " $\exists f. f \in (\prod B \in \{uu(a). a \in A\}. B) \implies \exists f. f \in (\prod B \in A. B)$ "

$\langle proof \rangle$

lemma *lemma2_1*: " $a \neq 0 \implies 0 \in (\bigcup b \in uu(a). b)$ "

$\langle proof \rangle$

lemma *lemma2*: " $\llbracket A \neq 0; 0 \notin A \rrbracket \implies (\bigcap x \in \{uu(a). a \in A\}. \bigcup b \in x. b) \neq$

```

0"
⟨proof⟩

lemma AC19_AC1: "AC19  $\implies$  AC1"
⟨proof⟩

end

theory DC
imports AC_Equiv Hartog Cardinal_aux
begin

lemma RepFun_lepoll: " $\text{Ord}(a) \implies \{P(b). b \in a\} \lesssim a$ "
⟨proof⟩

Trivial in the presence of AC, but here we need a wellordering of X

lemma image_Ord_lepoll: " $\llbracket f \in X \rightarrow Y; \text{Ord}(X) \rrbracket \implies f'X \lesssim X$ "
⟨proof⟩

lemma range_subset_domain:
  " $\llbracket R \subseteq X * X; \bigwedge g. g \in X \implies \exists u. \langle g, u \rangle \in R \rrbracket$ 
 $\implies \text{range}(R) \subseteq \text{domain}(R)$ "
⟨proof⟩

lemma cons_fun_type: " $g \in n \rightarrow X \implies \text{cons}(\langle n, x \rangle, g) \in \text{succ}(n) \rightarrow \text{cons}(x, X)$ "
⟨proof⟩

lemma cons_fun_type2:
  " $\llbracket g \in n \rightarrow X; x \in X \rrbracket \implies \text{cons}(\langle n, x \rangle, g) \in \text{succ}(n) \rightarrow X$ "
⟨proof⟩

lemma cons_image_n: " $n \in \text{nat} \implies \text{cons}(\langle n, x \rangle, g)'n = g'n$ "
⟨proof⟩

lemma cons_val_n: " $g \in n \rightarrow X \implies \text{cons}(\langle n, x \rangle, g)'n = x$ "
⟨proof⟩

lemma cons_image_k: " $k \in n \implies \text{cons}(\langle n, x \rangle, g)'k = g'k$ "
⟨proof⟩

lemma cons_val_k: " $\llbracket k \in n; g \in n \rightarrow X \rrbracket \implies \text{cons}(\langle n, x \rangle, g)'k = g'k$ "
⟨proof⟩

lemma domain_cons_eq_succ: " $\text{domain}(f) = x \implies \text{domain}(\text{cons}(\langle x, y \rangle, f)) = \text{succ}(x)$ "
⟨proof⟩

```

lemma *restrict_cons_eq*: " $g \in n \rightarrow X \implies \text{restrict}(\text{cons}(\langle n, x \rangle, g), n) = g$ "
 <proof>

lemma *succ_in_succ*: " $\llbracket \text{Ord}(k); i \in k \rrbracket \implies \text{succ}(i) \in \text{succ}(k)$ "
 <proof>

lemma *restrict_eq_imp_val_eq*:
 " $\llbracket \text{restrict}(f, \text{domain}(g)) = g; x \in \text{domain}(g) \rrbracket$
 $\implies f'x = g'x$ "
 <proof>

lemma *domain_eq_imp_fun_type*: " $\llbracket \text{domain}(f) = A; f \in B \rightarrow C \rrbracket \implies f \in A \rightarrow C$ "
 <proof>

lemma *ex_in_domain*: " $\llbracket R \subseteq A * B; R \neq 0 \rrbracket \implies \exists x. x \in \text{domain}(R)$ "
 <proof>

definition

DC :: " $i \Rightarrow o$ " **where**
 " $\text{DC}(a) \equiv \forall X R. R \subseteq \text{Pow}(X) * X \wedge$
 $(\forall Y \in \text{Pow}(X). Y \prec a \longrightarrow (\exists x \in X. \langle Y, x \rangle \in R))$
 $\longrightarrow (\exists f \in a \rightarrow X. \forall b \prec a. \langle f' b, f' b \rangle \in R)$ "

definition

DC0 :: o **where**
 " $\text{DC0} \equiv \forall A B R. R \subseteq A * B \wedge R \neq 0 \wedge \text{range}(R) \subseteq \text{domain}(R)$
 $\longrightarrow (\exists f \in \text{nat} \rightarrow \text{domain}(R). \forall n \in \text{nat}. \langle f' n, f' \text{succ}(n) \rangle \in R)$ "

definition

ff :: " $[i, i, i, i] \Rightarrow i$ " **where**
 " $\text{ff}(b, X, Q, R) \equiv$
 $\text{transrec}(b, \lambda c r. \text{THE } x. \text{first}(x, \{x \in X. \langle r' c, x \rangle \in R\},$
 $Q))$ "

locale *DC0_imp* =

fixes *XX* **and** *RR* **and** *X* **and** *R*

assumes *all_ex*: " $\forall Y \in \text{Pow}(X). Y \prec \text{nat} \longrightarrow (\exists x \in X. \langle Y, x \rangle \in R)$ "

defines *XX_def*: " $XX \equiv (\bigcup n \in \text{nat}. \{f \in n \rightarrow X. \forall k \in n. \langle f' k, f' k \rangle \in R\})$ "

and *RR_def*: " $RR \equiv \{\langle z1, z2 \rangle : XX * XX. \text{domain}(z2) = \text{succ}(\text{domain}(z1))$
 $\wedge \text{restrict}(z2, \text{domain}(z1)) = z1\}$ "

begin

lemma lemma1_1: "RR \subseteq XX*XX"
 <proof>

lemma lemma1_2: "RR \neq 0"
 <proof>

lemma lemma1_3: "range(RR) \subseteq domain(RR)"
 <proof>

lemma lemma2:
 "[$\forall n \in \text{nat}. \langle f'n, f'succ(n) \rangle \in \text{RR}; f \in \text{nat} \rightarrow \text{XX}; n \in \text{nat}$]
 $\implies \exists k \in \text{nat}. f'succ(n) \in k \rightarrow X \wedge n \in k$
 $\wedge \langle f'succ(n)'n, f'succ(n)'n \rangle \in R$ "
 <proof>

lemma lemma3_1:
 "[$\forall n \in \text{nat}. \langle f'n, f'succ(n) \rangle \in \text{RR}; f \in \text{nat} \rightarrow \text{XX}; m \in \text{nat}$]
 $\implies \{f'succ(x)'x. x \in m\} = \{f'succ(m)'x. x \in m\}$ "
 <proof>

lemma lemma3:
 "[$\forall n \in \text{nat}. \langle f'n, f'succ(n) \rangle \in \text{RR}; f \in \text{nat} \rightarrow \text{XX}; m \in \text{nat}$]
 $\implies (\lambda x \in \text{nat}. f'succ(x)'x) \text{ `` } m = f'succ(m)'m$ "
 <proof>

end

theorem *DC0_imp_DC_nat*: " $DC0 \implies DC(nat)$ "
 <proof>

lemma *singleton_in_funs*:
 " $x \in X \implies \{ \langle 0, x \rangle \} \in$
 $(\bigcup n \in nat. \{ f \in succ(n) \rightarrow X. \forall k \in n. \langle f'k, f'succ(k) \rangle \in$
 $R \})$ "
 <proof>

locale *imp_DC0* =
 fixes *XX* and *RR* and *x* and *R* and *f* and *allRR*
 defines *XX_def*: " $XX \equiv (\bigcup n \in nat.$
 $\{ f \in succ(n) \rightarrow domain(R). \forall k \in n. \langle f'k, f'succ(k) \rangle$
 $\in R \})$ "
 and *RR_def*:
 $RR \equiv \{ \langle z1, z2 \rangle : Fin(XX) * XX.$
 $(domain(z2) = succ(\bigcup f \in z1. domain(f))$
 $\wedge (\forall f \in z1. restrict(z2, domain(f)) = f))$
 $\mid (\neg (\exists g \in XX. domain(g) = succ(\bigcup f \in z1. domain(f))$
 $\wedge (\forall f \in z1. restrict(g, domain(f)) = f)) \wedge z2 = \{ \langle 0, x \rangle \} \}$ "
 and *allRR_def*:
 $allRR \equiv \forall b < nat.$
 $\langle f' 'b, f' 'b \rangle \in$
 $\{ \langle z1, z2 \rangle \in Fin(XX) * XX. (domain(z2) = succ(\bigcup f \in z1. domain(f))$
 $\wedge (\bigcup f \in z1. domain(f)) = b$
 $\wedge (\forall f \in z1. restrict(z2, domain(f))$
 $= f)) \}$ "
begin

lemma *lemma4*:
 $\llbracket range(R) \subseteq domain(R); x \in domain(R) \rrbracket$
 $\implies RR \subseteq Pow(XX) * XX \wedge$
 $(\forall Y \in Pow(XX). Y \prec nat \longrightarrow (\exists x \in XX. \langle Y, x \rangle : RR))$ "
 <proof>

lemma *UN_image_succ_eq*:
 $\llbracket f \in nat \rightarrow X; n \in nat \rrbracket$
 $\implies (\bigcup x \in f' 'succ(n). P(x)) = P(f'n) \cup (\bigcup x \in f' 'n. P(x))$ "
 <proof>

lemma *UN_image_succ_eq_succ*:

$$\llbracket (\bigcup x \in f' 'n. P(x)) = y; P(f'n) = \text{succ}(y); f \in \text{nat} \rightarrow X; n \in \text{nat} \rrbracket \implies (\bigcup x \in f' ' \text{succ}(n). P(x)) = \text{succ}(y)''$$
 $\langle \text{proof} \rangle$

lemma *apply_domain_type*:

$$\llbracket h \in \text{succ}(n) \rightarrow D; n \in \text{nat}; \text{domain}(h) = \text{succ}(y) \rrbracket \implies h'y \in D''$$
 $\langle \text{proof} \rangle$

lemma *image_fun_succ*:

$$\llbracket h \in \text{nat} \rightarrow X; n \in \text{nat} \rrbracket \implies h' ' \text{succ}(n) = \text{cons}(h'n, h' 'n)''$$
 $\langle \text{proof} \rangle$

lemma *f_n_type*:

$$\llbracket \text{domain}(f'n) = \text{succ}(k); f \in \text{nat} \rightarrow XX; n \in \text{nat} \rrbracket \implies f'n \in \text{succ}(k) \rightarrow \text{domain}(R)''$$
 $\langle \text{proof} \rangle$

lemma *f_n_pairs_in_R* [rule_format]:

$$\llbracket h \in \text{nat} \rightarrow XX; \text{domain}(h'n) = \text{succ}(k); n \in \text{nat} \rrbracket \implies \forall i \in k. \langle h'n'i, h'n' \text{succ}(i) \rangle \in R''$$
 $\langle \text{proof} \rangle$

lemma *restrict_cons_eq_restrict*:

$$\llbracket \text{restrict}(h, \text{domain}(u)) = u; h \in n \rightarrow X; \text{domain}(u) \subseteq n \rrbracket \implies \text{restrict}(\text{cons}(\langle n, y \rangle, h), \text{domain}(u)) = u''$$
 $\langle \text{proof} \rangle$

lemma *all_in_image_restrict_eq*:

$$\begin{aligned} & \llbracket \forall x \in f' 'n. \text{restrict}(f'n, \text{domain}(x)) = x; f \in \text{nat} \rightarrow XX; n \in \text{nat}; \text{domain}(f'n) = \text{succ}(n); (\bigcup x \in f' 'n. \text{domain}(x)) \subseteq n \rrbracket \\ & \implies \forall x \in f' ' \text{succ}(n). \text{restrict}(\text{cons}(\langle \text{succ}(n), y \rangle, f'n), \text{domain}(x)) = x'' \end{aligned}$$
 $\langle \text{proof} \rangle$

lemma *simplify_recursion*:

$$\llbracket \forall b < \text{nat}. \langle f' 'b, f'b \rangle \in RR; f \in \text{nat} \rightarrow XX; \text{range}(R) \subseteq \text{domain}(R); x \in \text{domain}(R) \rrbracket \implies \text{all}RR''$$
 $\langle \text{proof} \rangle$

lemma *lemma2*:

$$\llbracket \text{all}RR; f \in \text{nat} \rightarrow XX; \text{range}(R) \subseteq \text{domain}(R); x \in \text{domain}(R); n \in \text{nat} \rrbracket \implies f'n \in \text{succ}(n) \rightarrow \text{domain}(R) \wedge (\forall i \in n. \langle f'n'i, f'n' \text{succ}(i) \rangle : R)''$$
 $\langle \text{proof} \rangle$

```

lemma lemma3:
  "⟦allRR; f ∈ nat->XX; n∈nat; range(R) ⊆ domain(R); x ∈ domain(R)⟧
    ⇒ f'n'n = f'succ(n)'n"
  ⟨proof⟩

```

end

```

theorem DC_nat_imp_DC0: "DC(nat) ⇒ DC0"
  ⟨proof⟩

```

```

lemma fun_Ord_inj:
  "⟦f ∈ a->X; Ord(a);
    ∧ b c. ⟦b<c; c ∈ a⟧ ⇒ f'b≠f'c⟧
    ⇒ f ∈ inj(a, X)"
  ⟨proof⟩

```

```

lemma value_in_image: "⟦f ∈ X->Y; A ⊆ X; a ∈ A⟧ ⇒ f'a ∈ f' 'A"
  ⟨proof⟩

```

```

lemma lesspoll_lemma: "⟦¬ A < B; C < B⟧ ⇒ A - C ≠ 0"
  ⟨proof⟩

```

```

theorem DC_W03: "(∀ K. Card(K) → DC(K)) ⇒ W03"
  ⟨proof⟩

```

```

lemma images_eq:
  "⟦∀ x ∈ A. f'x=g'x; f ∈ Df->Cf; g ∈ Dg->Cg; A ⊆ Df; A ⊆ Dg⟧
    ⇒ f' 'A = g' 'A"
  ⟨proof⟩

```

```

lemma lam_images_eq:
  "⟦Ord(a); b ∈ a⟧ ⇒ (λx ∈ a. h(x))' 'b = (λx ∈ b. h(x))' 'b"
  ⟨proof⟩

```

```

lemma lam_type_RepFun: "(λb ∈ a. h(b)) ∈ a -> {h(b). b ∈ a}"
  ⟨proof⟩

```

```

lemma lemmaX:
  "⟦∀ Y ∈ Pow(X). Y < K → (∃ x ∈ X. ⟨Y, x⟩ ∈ R);
    b ∈ K; Z ∈ Pow(X); Z < K⟧

```

$\implies \{x \in X. \langle Z, x \rangle \in R\} \neq 0$ "
 $\langle proof \rangle$

lemma *W01_DC_lemma*:

" $\llbracket \text{Card}(K); \text{well_ord}(X, Q);$
 $\forall Y \in \text{Pow}(X). Y \prec K \longrightarrow (\exists x \in X. \langle Y, x \rangle \in R); b \in K \rrbracket$
 $\implies \text{ff}(b, X, Q, R) \in \{x \in X. <(\lambda c \in b. \text{ff}(c, X, Q, R))\}^{\langle b, x \rangle \in R}$ "
 $\langle proof \rangle$

theorem *W01_DC_Card*: " $W01 \implies \forall K. \text{Card}(K) \longrightarrow DC(K)$ "
 $\langle proof \rangle$

end

References

- [1] Lawrence C. Paulson and Krzysztof Gŗabczewski. Mechanizing set theory: Cardinal arithmetic and the axiom of choice. *Journal of Automated Reasoning*, 17(3):291–323, December 1996.
- [2] Herman Rubin and Jean E. Rubin. *Equivalents of the Axiom of Choice, II*. North-Holland, 1985.