

Equivalents of the Axiom of Choice

Krzysztof Grąbczewski

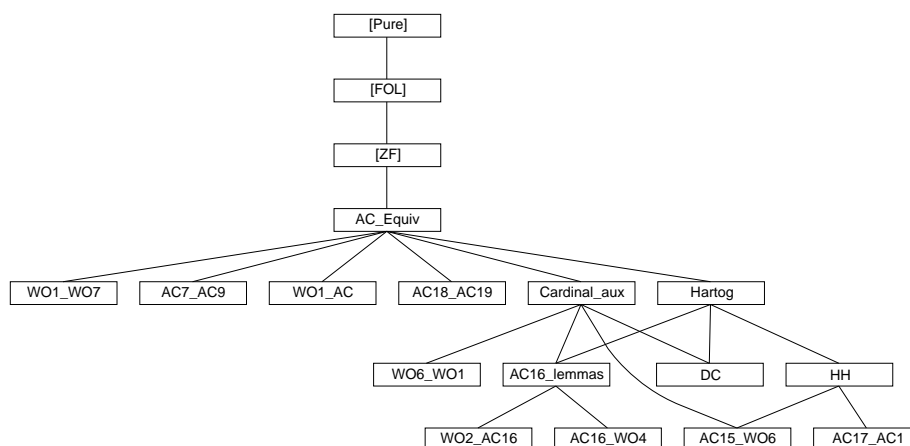
June 8, 2008

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 $AC1 \Rightarrow WO2$ and $AC17 \Rightarrow AC1$	23
0.3	The proof of $AC1 \Rightarrow WO2$	25



```
theory AC_Equiv imports Main begin
```

```
definition
```

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

```
definition
```

```
"W02 ==  $\forall A. \exists a. \text{Ord}(a) \ \& \ A \approx a$ "
```

```
definition
```

```
"W03 ==  $\forall A. \exists a. \text{Ord}(a) \ \& \ (\exists b. b \subseteq a \ \& \ A \approx b)$ "
```

```
definition
```

```
"W04(m) ==  $\forall A. \exists a \ f. \text{Ord}(a) \ \& \ \text{domain}(f)=a \ \& \$   

 $(\bigcup b < a. f' b) = A \ \& \ (\forall b < a. f' b \lesssim m)$ "
```

```
definition
```

```
"W05 ==  $\exists m \in \text{nat}. 1 \leq m \ \& \ W04(m)$ "
```

```
definition
```

```
"W06 ==  $\forall A. \exists m \in \text{nat}. 1 \leq m \ \& \ (\exists a \ f. \text{Ord}(a) \ \& \ \text{domain}(f)=a \ \& \$   

 $(\bigcup b < a. f' b) = A \ \& \ (\forall b < a. f' b \lesssim m))$ "
```

```
definition
```

```
"W07 ==  $\forall A. \text{Finite}(A) \leftrightarrow (\forall R. \text{well\_ord}(A, R) \rightarrow \text{well\_ord}(A, \text{converse}(R)))$ "
```

```
definition
```

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

```
definition
```

```
pairwise_disjoint :: "i => o" where  

"pairwise_disjoint(A) ==  $\forall A1 \in A. \forall A2 \in A. A1 \text{ Int } A2 \neq 0 \rightarrow A1=A2$ "
```

```
definition
```

```
sets_of_size_between :: "[i, i, i] => o" where  

"sets_of_size_between(A, m, n) ==  $\forall B \in A. m \lesssim B \ \& \ B \lesssim n$ "
```

```
definition
```

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

```
definition
```

```
"AC1 ==  $\forall A. 0 \notin A \rightarrow (\exists f. f \in (\prod X \in A. X))$ "
```

definition

"AC2 == $\forall A. 0 \notin A \ \& \ \text{pairwise_disjoint}(A)$
 $\rightarrow (\exists C. \forall B \in A. \exists y. B \text{ Int } C = \{y\})$ "

definition

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

definition

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

definition

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

definition

"AC6 == $\forall A. 0 \notin A \rightarrow (\prod B \in A. B) \neq 0$ "

definition

"AC7 == $\forall A. 0 \notin A \ \& \ (\forall B1 \in A. \forall B2 \in A. B1 \approx B2) \rightarrow (\prod B \in A. B) \neq 0$ "

definition

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

definition

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

definition

"AC10(n) == $\forall A. (\forall B \in A. \sim \text{Finite}(B)) \rightarrow$
 $(\exists f. \forall B \in A. (\text{pairwise_disjoint}(f'B) \ \& \ \text{sets_of_size_between}(f'B, 2, \text{succ}(n)) \ \& \ \text{Union}(f'B) = B))$ "

definition

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

definition

"AC12 == $\forall A. (\forall B \in A. \sim \text{Finite}(B)) \rightarrow$
 $(\exists n \in \text{nat}. 1 \leq n \ \& \ (\exists f. \forall B \in A. (\text{pairwise_disjoint}(f'B)$
 $\ \& \ \text{sets_of_size_between}(f'B, 2, \text{succ}(n)) \ \& \ \text{Union}(f'B) = B)))$ "

definition

"AC13(m) == $\forall A. 0 \notin A \rightarrow (\exists f. \forall B \in A. f'B \neq 0 \ \& \ f'B \subseteq B \ \& \ f'B \lesssim m)$ "

definition

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

definition

```
"AC15 == ∀ A. 0 ∉ A -->
      (∃ m ∈ nat. 1 ≤ m & (∃ f. ∀ B ∈ A. f' B ≠ 0 & f' B ⊆ B &
f' B ≲ m))"
```

definition

```
"AC16(n, k) ==
  ∀ A. ~Finite(A) -->
    (∃ T. T ⊆ {X ∈ Pow(A). X ≈ succ(n)} &
    (∀ X ∈ {X ∈ Pow(A). X ≈ succ(k)}. ∃ ! Y. Y ∈ T & X ⊆ Y))"
```

definition

```
"AC17 == ∀ A. ∀ g ∈ (Pow(A) - {0} -> A) -> Pow(A) - {0}.
      ∃ f ∈ Pow(A) - {0} -> A. f' (g' f) ∈ g' f"
```

locale AC18 =

```
  assumes AC18: "A ≠ 0 & (∀ a ∈ A. B(a) ≠ 0) -->
    ((∩ a ∈ A. ∪ b ∈ B(a). X(a, b)) =
    (∪ f ∈ Π a ∈ A. B(a). ∩ a ∈ A. X(a, f' a)))"
  — AC18 cannot be expressed within the object-logic
```

definition

```
"AC19 == ∀ A. A ≠ 0 & 0 ∉ A --> ((∩ a ∈ A. ∪ b ∈ a. b) =
      (∪ f ∈ (Π B ∈ A. B). ∩ a ∈ A. f' a))"
```

lemma rvimage_id: "rvimage(A, id(A), r) = r Int A*A"
 <proof>

lemma ordertype_Int:
 "well_ord(A, r) ==> ordertype(A, r Int A*A) = ordertype(A, r)"
 <proof>

lemma lam_sing_bij: "(λ x ∈ A. {x}) ∈ bij(A, {{x}. x ∈ A})"
 <proof>

lemma inj_strengthen_type:
 "[| f ∈ inj(A, B); !!a. a ∈ A ==> f'a ∈ C |] ==> f ∈ inj(A, C)"
 <proof>

lemma nat_not_Finite: "~ Finite(nat)"

$\langle proof \rangle$

lemmas *le_imp_lepoll* = *le_imp_subset* [THEN *subset_imp_lepoll*]

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

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

lemma *first_in_B*:
" $[| \text{well_ord}(\text{Union}(A), r); 0 \notin A; B \in A |] \implies (\text{THE } b. \text{first}(b, B, r)) \in B$ "
 $\langle proof \rangle$

lemma *ex_choice_fun*: " $[| \text{well_ord}(\text{Union}(A), R); 0 \notin A |] \implies \exists f. f: (\prod X \in A. X)$ "
 $\langle proof \rangle$

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

lemma *lepoll_m_imp_domain_lepoll_m*:
" $[| m \in \text{nat}; u \lesssim m |] \implies \text{domain}(u) \lesssim m$ "
 $\langle proof \rangle$

lemma *rel_domain_ex1*:

"[| succ(m) \lesssim domain(r); r \lesssim succ(m); m \in nat |] ==> function(r)"
 <proof>

lemma rel_is_fun:
 "[| succ(m) \lesssim domain(r); r \lesssim succ(m); m \in nat;
 r \subseteq A*B; A=domain(r) |] ==> r \in A \rightarrow B"
 <proof>

end

theory Cardinal_aux imports AC_Equiv begin

lemma Diff_lepoll: "[| A \lesssim succ(m); B \subseteq A; B \neq 0 |] ==> A-B \lesssim m"
 <proof>

lemma lepoll_imp_ex_le_eqpoll:
 "[| A \lesssim i; Ord(i) |] ==> $\exists j. j \leq i$ & A \approx j"
 <proof>

lemma lesspoll_imp_ex_lt_eqpoll:
 "[| A \prec i; Ord(i) |] ==> $\exists j. j < i$ & A \approx j"
 <proof>

lemma Inf_Ord_imp_InfCard_cardinal: "[| \sim Finite(i); Ord(i) |] ==> InfCard(|i|)"
 <proof>

An alternative and more general proof goes like this: A and B are both well-ordered (because they are injected into an ordinal), either A lepoll B or B lepoll A. Also both are equipollent to their cardinalities, so (if A and B are infinite) then A Un B lepoll $\text{---}A\text{---} + \text{---}B\text{---} = \max(\text{---}A\text{---}, \text{---}B\text{---})$ lepoll i. In fact, the correctly strengthened version of this theorem appears below.

lemma Un_lepoll_Inf_Ord_weak:
 "[| A \approx i; B \approx i; \neg Finite(i); Ord(i) |] ==> A \cup B \lesssim i"
 <proof>

lemma Un_eqpoll_Inf_Ord:
 "[| A \approx i; B \approx i; \sim Finite(i); Ord(i) |] ==> A Un B \approx i"
 <proof>

lemma paired_bij: "?f ∈ bij({{y,z}. y ∈ x}, x)"
 <proof>

lemma paired_eqpoll: "{{y,z}. y ∈ x} ≈ x"
 <proof>

lemma ex_eqpoll_disjoint: "∃B. B ≈ A & B Int C = 0"
 <proof>

lemma Un_lepoll_Inf_Ord:
 "[| A ≲ i; B ≲ i; ~Finite(i); Ord(i) |] ==> A Un B ≲ i"
 <proof>

lemma Least_in_Ord: "[| P(i); i ∈ j; Ord(j) |] ==> (LEAST i. P(i)) ∈ j"
 <proof>

lemma Diff_first_lepoll:
 "[| well_ord(x,r); y ⊆ x; y ≲ succ(n); n ∈ nat |]
 ==> y - {THE b. first(b,y,r)} ≲ n"
 <proof>

lemma UN_subset_split:
 "(⋃ x ∈ X. P(x)) ⊆ (⋃ x ∈ X. P(x)-Q(x)) Un (⋃ x ∈ X. Q(x))"
 <proof>

lemma UN_sing_lepoll: "Ord(a) ==> (⋃ x ∈ a. {P(x)}) ≲ a"
 <proof>

lemma UN_fun_lepoll_lemma [rule_format]:
 "[| well_ord(T, R); ~Finite(a); Ord(a); n ∈ nat |]
 ==> ∀f. (∀ b ∈ a. f' b ≲ n & f' b ⊆ T) --> (⋃ b ∈ a. f' b) ≲ a"
 <proof>

lemma UN_fun_lepoll:
 "[| ∀ b ∈ a. f' b ≲ n & f' b ⊆ T; well_ord(T, R);
 ~Finite(a); Ord(a); n ∈ nat |] ==> (⋃ b ∈ a. f' b) ≲ a"
 <proof>

lemma UN_lepoll:
 "[| ∀ b ∈ a. F(b) ≲ n & F(b) ⊆ T; well_ord(T, R);
 ~Finite(a); Ord(a); n ∈ nat |]
 ==> (⋃ b ∈ a. F(b)) ≲ a"
 <proof>

lemma UN_eq_UN_Diffs:
 "Ord(a) ==> (⋃ b ∈ a. F(b)) = (⋃ b ∈ a. F(b) - (⋃ c ∈ b. F(c)))"

<proof>

lemma *lepoll_imp_eqpoll_subset*:
"a \lesssim X ==> $\exists Y. Y \subseteq X$ & a \approx Y"
<proof>

lemma *Diff_lesspoll_eqpoll_Card_lemma*:
"[| A \approx a; \sim Finite(a); Card(a); B \prec a; A-B \prec a |] ==> P"
<proof>

lemma *Diff_lesspoll_eqpoll_Card*:
"[| A \approx a; \sim Finite(a); Card(a); B \prec a |] ==> A - B \approx a"
<proof>

end

theory *W06_W01* imports *Cardinal_aux* begin

definition

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

definition

uu :: "[i, i, i, i] => i" where
"uu(f, beta, gamma, delta) == (f'beta * f'gamma) Int f'delta"

definition

vv1 :: "[i, i, i] => i" where
"vv1(f,m,b) ==
let g = LEAST g. ($\exists d. \text{Ord}(d)$ & (domain(uu(f,b,g,d)) \neq 0 &
domain(uu(f,b,g,d)) \lesssim m));
d = LEAST d. domain(uu(f,b,g,d)) \neq 0 &
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] => i" where
"ww1(f,m,b) == f'b - vv1(f,m,b)"

definition

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

definition

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

definition

```
ww2 :: "[i, i, i, i] => i" where
  "ww2(f,b,g,s) == f'g - vv2(f,b,g,s)"
```

definition

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

lemma W02_W03: "W02 ==> W03"

<proof>

lemma W03_W01: "W03 ==> W01"

<proof>

lemma W01_W02: "W01 ==> W02"

<proof>

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

<proof>

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

<proof>

lemma surj_imp_eq: " $[f \in \text{surj}(A,B); \text{Ord}(A)] \implies (\bigcup a < A. \{f'a\}) = B$ "

<proof>

lemma W01_W04: "W01 ==> W04(1)"

$\langle proof \rangle$

lemma *W04_mono*: "[| $m \leq n$; $W04(m)$ |] ==> $W04(n)$ "
 $\langle proof \rangle$

lemma *W04_W05*: "[| $m \in \text{nat}$; $1 \leq m$; $W04(m)$ |] ==> $W05$ "
 $\langle proof \rangle$

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

lemma *lt_oadd_odiff_disj*:
 "[| $k < i++j$; $Ord(i)$; $Ord(j)$ |]
 ==> $k < i$ | ($\sim k < i$ & $k = i ++ (k--i)$ & $(k--i) < j$)"
 $\langle proof \rangle$

lemma *domain_uu_subset*: " $\text{domain}(uu(f,b,g,d)) \subseteq f'b$ "
 $\langle proof \rangle$

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

lemma *uu_subset1*: " $uu(f,b,g,d) \subseteq f'b * f'g$ "
 $\langle proof \rangle$

lemma *uu_subset2*: " $uu(f,b,g,d) \subseteq f'd$ "
 $\langle proof \rangle$

lemma *uu_lepoll_m*: "[| $\forall b < a. f'b \lesssim m$; $d < a$ |] ==> $uu(f,b,g,d) \lesssim m$ "
 $\langle proof \rangle$

lemma cases:

```
" $\forall b < a. \forall g < a. \forall d < a. u(f, b, g, d) \lesssim m$ "
==> ( $\forall b < a. f' b \neq 0 \rightarrow$ 
      ( $\exists g < a. \exists d < a. u(f, b, g, d) \neq 0 \ \& \ u(f, b, g, d) \prec m$ ))
  | ( $\exists b < a. f' b \neq 0 \ \& \ (\forall g < a. \forall d < a. u(f, b, g, d) \neq 0 \rightarrow$ 
       $u(f, b, g, d) \approx m)$ )"
```

$\langle proof \rangle$

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

$\langle proof \rangle$

lemma vv1_subset: " $vv1(f, m, b) \subseteq f' b$ "

$\langle proof \rangle$

lemma UN_gg1_eq:

```
"[| Ord(a); m  $\in$  nat |] ==> ( $\bigcup b < a ++ a. gg1(f, a, m)'b$ ) = ( $\bigcup b < a. f' b$ )"
```

$\langle proof \rangle$

lemma domain_gg1: " $domain(gg1(f, a, m)) = a ++ a$ "

$\langle proof \rangle$

lemma nested_LeastI:

```
"[| P(a, b); Ord(a); Ord(b);
   Least_a = (LEAST a.  $\exists x. Ord(x) \ \& \ P(a, x)$ ) |]
==> P(Least_a, LEAST b. P(Least_a, b))"
```

$\langle proof \rangle$

lemmas nested_Least_instance =

```
nested_LeastI [of "%g d. domain(uu(f, b, g, d))  $\neq$  0 &
                  domain(uu(f, b, g, d))  $\lesssim$  m",
                standard]
```

```

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 |]
  ==> uu(f,b,g,LEAST d. (uu(f,b,g,d) ≠ 0)) ≠ 0"
<proof>

```

```

lemma not_empty_rel_imp_domain: "[| r ⊆ A*B; r ≠ 0 |] ==> domain(r) ≠ 0"
<proof>

```

```

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

```

```

lemma subset_Diff_sing: "[| B ⊆ A; a ∉ B |] ==> B ⊆ A-{a}"
<proof>

```

```

lemma supset_lepoll_imp_eq:
  "[| A ≲ m; m ≲ B; B ⊆ A; m ∈ nat |] ==> A=B"
<proof>

```

```

lemma uu_Least_is_fun:
  "[| ∀ g<a. ∀ d<a. domain(uu(f, b, g, d)) ≠ 0 -->
    domain(uu(f, b, g, d)) ≈ succ(m);
    ∀ b<a. f' b ≲ succ(m); y*y ⊆ y;

```

$$(\bigcup b < a. f' b) = y; \quad b < a; \quad g < a; \quad d < a;$$

$$f' b \neq 0; \quad f' g \neq 0; \quad m \in \text{nat}; \quad s \in f' b \quad |]$$

$$\implies \text{uu}(f, b, g, \text{LEAST } d. \text{uu}(f, b, g, d) \neq 0) \in f' b \rightarrow f' g$$

$$\langle \text{proof} \rangle$$

lemma `vv2_subset`:

$$"[| \forall g < a. \forall d < a. \text{domain}(\text{uu}(f, b, g, d)) \neq 0 \rightarrow$$

$$\text{domain}(\text{uu}(f, b, g, d)) \approx \text{succ}(m);$$

$$\forall b < a. f' b \lesssim \text{succ}(m); \quad y * y \subseteq y;$$

$$(\bigcup b < a. f' b) = y; \quad b < a; \quad g < a; \quad m \in \text{nat}; \quad s \in f' b \quad |]$$

$$\implies \text{vv2}(f, b, g, s) \subseteq f' g$$

$$\langle \text{proof} \rangle$$

lemma `UN_gg2_eq`:

$$"[| \forall g < a. \forall d < a. \text{domain}(\text{uu}(f, b, g, d)) \neq 0 \rightarrow$$

$$\text{domain}(\text{uu}(f, b, g, d)) \approx \text{succ}(m);$$

$$\forall b < a. f' b \lesssim \text{succ}(m); \quad y * y \subseteq y;$$

$$(\bigcup b < a. f' b) = y; \quad 0 \text{rd}(a); \quad m \in \text{nat}; \quad s \in f' b; \quad b < a \quad |]$$

$$\implies (\bigcup g < a ++ a. \text{gg2}(f, a, b, s) \text{ ' } g) = y$$

$$\langle \text{proof} \rangle$$

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

$$\langle \text{proof} \rangle$$

lemma `vv2_lepoll`: $\text{"}[| m \in \text{nat}; \quad m \neq 0 \quad |] \implies \text{vv2}(f, b, g, s) \lesssim m"$

$$\langle \text{proof} \rangle$$

lemma `ww2_lepoll`:

$$"[| \forall b < a. f' b \lesssim \text{succ}(m); \quad g < a; \quad m \in \text{nat}; \quad \text{vv2}(f, b, g, d) \subseteq f' g \quad |]$$

$$\implies \text{ww2}(f, b, g, d) \lesssim m$$

$$\langle \text{proof} \rangle$$

lemma `gg2_lepoll_m`:

$$"[| \forall g < a. \forall d < a. \text{domain}(\text{uu}(f, b, g, d)) \neq 0 \rightarrow$$

$$\text{domain}(\text{uu}(f, b, g, d)) \approx \text{succ}(m);$$

$$\forall b < a. f' b \lesssim \text{succ}(m); \quad y * y \subseteq y;$$

$$(\bigcup b < a. f' b) = y; \quad b < a; \quad s \in f' b; \quad m \in \text{nat}; \quad m \neq 0; \quad g < a ++ a \quad |]$$

$$\implies \text{gg2}(f, a, b, s) \text{ ' } g \lesssim m$$

$$\langle \text{proof} \rangle$$

```

lemma lemma_ii: "[| succ(m) ∈ NN(y); y*y ⊆ y; m ∈ nat; m≠0 |] ==>
m ∈ NN(y)"
⟨proof⟩

```

```

lemma z_n_subset_z_succ_n:
  "∀n ∈ nat. rec(n, x, %k r. r Un r*r) ⊆ rec(succ(n), x, %k r. r
Un r*r)"
⟨proof⟩

```

```

lemma le_subsets:
  "[| ∀n ∈ nat. f(n)≤f(succ(n)); n≤m; n ∈ nat; m ∈ nat |]
==> f(n)≤f(m)"
⟨proof⟩

```

```

lemma le_imp_rec_subset:
  "[| n≤m; m ∈ nat |]
==> rec(n, x, %k r. r Un r*r) ⊆ rec(m, x, %k r. r Un r*r)"
⟨proof⟩

```

```

lemma lemma_iv: "∃y. x Un y*y ⊆ y"
⟨proof⟩

```

lemma W06_imp_NN_not_empty: "W06 ==> NN(y) ≠ 0"
 <proof>

lemma lemma1:
 "[| (⋃ b<a. f' b)=y; x ∈ y; ∀ b<a. f' b ≲ 1; 0rd(a) |] ==> ∃ c<a. f' c
 = {x}"
 <proof>

lemma lemma2:
 "[| (⋃ b<a. f' b)=y; x ∈ y; ∀ b<a. f' b ≲ 1; 0rd(a) |]
 ==> f' (LEAST i. f' i = {x}) = {x}"
 <proof>

lemma NN_imp_ex_inj: "1 ∈ NN(y) ==> ∃ a f. 0rd(a) & f ∈ inj(y, a)"
 <proof>

lemma y_well_ord: "[| y*y ⊆ y; 1 ∈ NN(y) |] ==> ∃ r. well_ord(y, r)"
 <proof>

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*: " $0 \in NN(y) \implies y=0$ "

<proof>

lemma *W06_imp_W01*: " $W06 \implies W01$ "

<proof>

end

theory *W01_W07* imports *AC_Equiv* begin

definition

"LEMMA ==

$\forall X. \sim Finite(X) \longrightarrow (\exists R. well_ord(X,R) \ \& \ \sim well_ord(X, converse(R)))$ "

lemma *W07_iff_LEMMA*: " $W07 \longleftrightarrow LEMMA$ "

<proof>

lemma *LEMMA_imp_W01*: " $LEMMA \implies W01$ "

<proof>

lemma *converse_Memrel_not_wf_on*:

"[| Ord(a); $\sim Finite(a)$ |] $\implies \sim wf[a](converse(Memrel(a)))$ "

<proof>

lemma *converse_Memrel_not_well_ord*:


```

    "[| Ord(a); ~Finite(a) |] ==> ~well_ord(a, converse(Memrel(a)))"
  <proof>

lemma well_ord_rvimage_ordertype:
  "well_ord(A,r) ==>
    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)) |]
    ==> well_ord(ordertype(A,r), converse(Memrel(ordertype(A,r))))"

  <proof>

lemma W01_imp_LEMMA: "W01 ==> LEMMA"
  <proof>

lemma W01_iff_W07: "W01 <-> W07"
  <proof>


lemma W01_W08: "W01 ==> W08"
  <proof>


lemma W08_W01: "W08 ==> W01"
  <proof>

end


theory AC7_AC9 imports AC_Equiv begin


lemma Sigma_fun_space_not0: "[| 0 ∉ A; B ∈ A |] ==> (nat->Union(A)) *
  B ≠ 0"

```

$\langle proof \rangle$

lemma *inj_lemma*:

" $C \in A \implies (\lambda g \in (\text{nat} \rightarrow \text{Union}(A)) * C.$
 $(\lambda n \in \text{nat}. \text{if}(n=0, \text{snd}(g), \text{fst}(g)'(n \#- 1))))$
 $\in \text{inj}((\text{nat} \rightarrow \text{Union}(A)) * C, (\text{nat} \rightarrow \text{Union}(A)))$ "

$\langle proof \rangle$

lemma *Sigma_fun_space_eqpoll*:

" $[| C \in A; 0 \notin A |] \implies (\text{nat} \rightarrow \text{Union}(A)) * C \approx (\text{nat} \rightarrow \text{Union}(A))$ "

$\langle proof \rangle$

lemma *AC6_AC7*: " $AC6 \implies AC7$ "

$\langle proof \rangle$

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*:

" $(\prod B \in \{(\text{nat} \rightarrow \text{Union}(A)) * C. C \in A\}. B) \neq 0 \implies (\prod B \in A. B) \neq 0$ "

$\langle proof \rangle$

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

$\langle proof \rangle$

lemma *AC7_AC6*: " $AC7 \implies AC6$ "

$\langle proof \rangle$

lemma AC1_AC8_lemma1:

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

$\langle \text{proof} \rangle$

lemma AC1_AC8_lemma2:

" $[\mid f \in (\prod X \in \text{RepFun}(A, p). X); D \in A \mid] \implies (\lambda x \in A. f'p(x))'D$
 $\in p(D)$ "

$\langle \text{proof} \rangle$

lemma AC1_AC8: "AC1 \implies AC8"

$\langle \text{proof} \rangle$

lemma AC8_AC9_lemma:

" $\forall B1 \in A. \forall B2 \in A. B1 \approx B2$
 $\implies \forall B \in A * A. \exists B1\ B2. B = \langle B1, B2 \rangle \ \& \ B1 \approx B2$ "

$\langle \text{proof} \rangle$

lemma AC8_AC9: "AC8 \implies AC9"

$\langle \text{proof} \rangle$

lemma snd_lepoll_SigmaI: " $b \in B \implies X \lesssim B \times X$ "

$\langle \text{proof} \rangle$

lemma nat_lepoll_lemma:

" $[\mid 0 \notin A; B \in A \mid] \implies \text{nat} \lesssim ((\text{nat} \rightarrow \text{Union}(A)) \times B) \times \text{nat}$ "

$\langle \text{proof} \rangle$

lemma AC9_AC1_lemma1:

" $[\mid 0 \notin A; A \neq 0;$
 $C = \{ (\text{nat} \rightarrow \text{Union}(A)) * B * \text{nat}. B \in A \} \quad \text{Un}$ "

```

      {cons(0,((nat->Union(A))*B)*nat). B ∈ A};
      B1 ∈ C; B2 ∈ C |]
    ==> B1 ≈ B2"
  <proof>

lemma AC9_AC1_lemma2:
  "∀ B1 ∈ {(F*B)*N. B ∈ A} Un {cons(0,(F*B)*N). B ∈ A}.
   ∀ B2 ∈ {(F*B)*N. B ∈ A} Un {cons(0,(F*B)*N). B ∈ A}.
   f'<B1,B2> ∈ bij(B1, B2)
   ==> (λB ∈ A. snd(fst((f'<cons(0,(F*B)*N),(F*B)*N>'0))) ∈ (Π X
∈ A. X))"
  <proof>

lemma AC9_AC1: "AC9 ==> AC1"
  <proof>

end

theory W01_AC imports AC_Equiv begin

theorem W01_AC1: "W01 ==> AC1"
  <proof>

lemma lemma1: "[| W01; ∀ B ∈ A. ∃ C ∈ D(B). P(C,B) |] ==> ∃ f. ∀ B ∈
A. P(f'B,B)"
  <proof>

lemma lemma2_1: "[| ~Finite(B); W01 |] ==> |B| + |B| ≈ B"
  <proof>

lemma lemma2_2:
  "f ∈ bij(D+D, B) ==> {{f'Inl(i), f'Inr(i)}. i ∈ D} ∈ Pow(Pow(B))"
  <proof>

lemma lemma2_3:
  "f ∈ bij(D+D, B) ==> pairwise_disjoint({{f'Inl(i), f'Inr(i)}.
i ∈ D})"
  <proof>

```

```

lemma lemma2_4:
  "[| f ∈ bij(D+D, B); 1 ≤ n |]
  ==> sets_of_size_between({f'Inl(i), f'Inr(i)}. i ∈ D}, 2, succ(n))"
⟨proof⟩

lemma lemma2_5:
  "f ∈ bij(D+D, B) ==> Union({f'Inl(i), f'Inr(i)}. i ∈ D}=B"
⟨proof⟩

lemma lemma2:
  "[| W01; ~Finite(B); 1 ≤ n |]
  ==> ∃ C ∈ Pow(Pow(B)). pairwise_disjoint(C) &
    sets_of_size_between(C, 2, succ(n)) &
    Union(C)=B"
⟨proof⟩

theorem W01_AC10: "[| W01; 1 ≤ n |] ==> AC10(n)"
⟨proof⟩

end

theory Hartog imports AC_Equiv begin

definition
  Hartog :: "i => i" where
    "Hartog(X) == LEAST i. ~ i ≲ X"

lemma Ords_in_set: "∀ a. Ord(a) --> a ∈ X ==> P"
⟨proof⟩

lemma Ord_lepoll_imp_ex_well_ord:
  "[| Ord(a); a ≲ X |]
  ==> ∃ Y. Y ⊆ X & (∃ R. well_ord(Y,R) & ordertype(Y,R)=a)"
⟨proof⟩

lemma Ord_lepoll_imp_eq_ordertype:
  "[| Ord(a); a ≲ X |] ==> ∃ Y. Y ⊆ X & (∃ R. R ⊆ X*X & ordertype(Y,R)=a)"
⟨proof⟩

lemma Ords_lepoll_set_lemma:
  "(∀ a. Ord(a) --> a ≲ X) ==>
  ∀ a. Ord(a) -->
  a ∈ {b. Z ∈ Pow(X)*Pow(X*X), ∃ Y R. Z=<Y,R> & ordertype(Y,R)=b}"
⟨proof⟩

```

```

lemma Ords_lepoll_set: " $\forall a. \text{Ord}(a) \rightarrow a \lesssim X \Rightarrow P$ "
<proof>

lemma ex_Ord_not_lepoll: " $\exists a. \text{Ord}(a) \ \& \ \sim a \lesssim X$ "
<proof>

lemma not_Hartog_lepoll_self: " $\sim \text{Hartog}(A) \lesssim A$ "
<proof>

lemmas Hartog_lepoll_selfE = not_Hartog_lepoll_self [THEN notE, standard]

lemma Ord_Hartog: " $\text{Ord}(\text{Hartog}(A))$ "
<proof>

lemma less_HartogE1: " $[| i < \text{Hartog}(A); \sim i \lesssim A |] \Rightarrow P$ "
<proof>

lemma less_HartogE: " $[| i < \text{Hartog}(A); i \approx \text{Hartog}(A) |] \Rightarrow P$ "
<proof>

lemma Card_Hartog: " $\text{Card}(\text{Hartog}(A))$ "
<proof>

end

```

theory HH imports AC_Equiv Hartog begin

definition

```

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

```

0.1 Lemmas useful in each of the three proofs

```

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

```

```

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

```

```

lemma subset_imp_Diff_eq:
  " $B \subseteq A \Rightarrow X - (\bigcup a \in A. P(a)) = X - (\bigcup a \in A - B. P(a)) - (\bigcup b \in B. P(b))$ "
<proof>

```

lemma *Ord_DiffE*: " $[| c \in a-b; b < a \ |] \implies c=b \ | \ b < c \ \& \ c < a$ "
 $\langle proof \rangle$

lemma *Diff_UN_eq_self*: " $(!!y. y \in A \implies P(y) = \{x\}) \implies x - (\bigcup y \in A. P(y)) = x$ "
 $\langle proof \rangle$

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*: " $[| HH(f, x, b) = \{x\}; b < a \ |] \implies HH(f, x, a) = \{x\}$ "
 $\langle proof \rangle$

lemma *HH_subset_x_lt_too*:
 $"[| HH(f, x, a) \in Pow(x) - \{0\}; b < a \ |] \implies HH(f, x, b) \in Pow(x) - \{0\}"$
 $\langle proof \rangle$

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

lemma *HH_eq_arg_lt*:
 $"[| HH(f, x, v) = HH(f, x, w); HH(f, x, v) \in Pow(x) - \{0\}; v \in w \ |] \implies P"$
 $\langle proof \rangle$

lemma *HH_eq_imp_arg_eq*:
 $"[| HH(f, x, v) = HH(f, x, w); HH(f, x, w) \in Pow(x) - \{0\}; Ord(v); Ord(w) \ |] \implies v=w"$
 $\langle proof \rangle$

lemma *HH_subset_x_imp_lepoll*:
 $"[| HH(f, x, i) \in Pow(x) - \{0\}; Ord(i) \ |] \implies i \text{ lepoll } Pow(x) - \{0\}"$
 $\langle proof \rangle$

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

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

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

0.2 Lemmas used in the proofs of AC1 \implies WO2 and AC17 \implies AC1

lemma *lam_Least_HH_inj_Pow*:

```

      "(\lambda a \in (LEAST i. HH(f,x,i)={x}). HH(f,x,a))
      \in inj(LEAST i. HH(f,x,i)={x}, Pow(x)-{0})"
    <proof>

lemma lam_Least_HH_inj:
  "\forall a \in (LEAST i. HH(f,x,i)={x}). \exists z \in x. HH(f,x,a) = {z}
  ==> (\lambda a \in (LEAST i. HH(f,x,i)={x}). HH(f,x,a))
  \in inj(LEAST i. HH(f,x,i)={x}, {\{y\}. y \in x})"
  <proof>

lemma lam_surj_sing:
  "[| x - (\bigcup a \in A. F(a)) = 0; \ \forall a \in A. \exists z \in x. F(a) = {z} |]

  ==> (\lambda a \in A. F(a)) \in surj(A, {\{y\}. y \in x})"
  <proof>

lemma not_emptyI2: "y \in Pow(x)-{0} ==> x \neq 0"
  <proof>

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}

  ==> HH(f, x, i) \in Pow(x) - {0}"
  <proof>

lemma f_subsets_imp_UN_HH_eq_x:
  "\forall z \in Pow(x)-{0}. f'z \in Pow(z)-{0}
  ==> x - (\bigcup j \in (LEAST i. HH(f,x,i)={x}). HH(f,x,j)) = 0"
  <proof>

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

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

lemma f_sing_imp_HH_sing:
  "[| f \in (Pow(x)-{0}) -> {\{z\}. z \in x};
  a \in (LEAST i. HH(f,x,i)={x}) |] ==> \exists z \in x. HH(f,x,a) = {z}"
  <proof>

lemma f_sing_lam_bij:
  "[| x - (\bigcup j \in (LEAST i. HH(f,x,i)={x}). HH(f,x,j)) = 0;
  f \in (Pow(x)-{0}) -> {\{z\}. z \in x} |]
  ==> (\lambda a \in (LEAST i. HH(f,x,i)={x}). HH(f,x,a))
  \in bij(LEAST i. HH(f,x,i)={x}, {\{y\}. y \in x})"
  <proof>

```



```

lemma lam_singI:
  "f ∈ (Π X ∈ Pow(x)-{0}. F(X))
  ==> (λX ∈ Pow(x)-{0}. {f'X}) ∈ (Π X ∈ Pow(x)-{0}. {{z}. z ∈ 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], standard]

```

0.3 The proof of AC1 ==_i WO2

```

lemma bijection:
  "f ∈ (Π X ∈ Pow(x) - {0}. X)
  ==> ∃g. g ∈ bij(x, LEAST i. HH(λX ∈ Pow(x)-{0}. {f'X}, x, i) =
{x})"
⟨proof⟩

```

```

lemma AC1_WO2: "AC1 ==> WO2"
⟨proof⟩

```

end

```

theory AC15_WO6 imports HH Cardinal_aux begin

```

```

lemma lepoll_Sigma: "A ≠ 0 ==> B ≲ A*B"
⟨proof⟩

```

```

lemma cons_times_nat_not_Finite:
  "0 ∉ A ==> ∀B ∈ {cons(0,x*nat). x ∈ A}. ~Finite(B)"
⟨proof⟩

```

```

lemma lemma1: "[| Union(C)=A; a ∈ A |] ==> ∃B ∈ C. a ∈ B & B ⊆ A"
⟨proof⟩

```

```

lemma lemma2:
  "[| pairwise_disjoint(A); B ∈ A; C ∈ A; a ∈ B; a ∈ C |] ==>
B=C"

```

$\langle proof \rangle$

lemma lemma3:

" $\forall B \in \{\text{cons}(0, x*\text{nat}). x \in A\}. \text{pairwise_disjoint}(f'B) \ \& \ \text{sets_of_size_between}(f'B, 2, n) \ \& \ \text{Union}(f'B)=B$
 $\implies \forall B \in A. \exists! u. u \in f'\text{cons}(0, B*\text{nat}) \ \& \ u \subseteq \text{cons}(0, B*\text{nat}) \ \&$

$0 \in u \ \& \ 2 \lesssim u \ \& \ u \lesssim n$ "

$\langle proof \rangle$

lemma lemma4: " $[| A \lesssim i; \text{Ord}(i) |] \implies \{P(a). a \in A\} \lesssim i$ "

$\langle proof \rangle$

lemma lemma5_1:

" $[| B \in A; 2 \lesssim u(B) |] \implies (\lambda x \in A. \{fst(x). x \in u(x)-\{0\}\})'B \neq 0$ "

$\langle proof \rangle$

lemma lemma5_2:

" $[| B \in A; u(B) \subseteq \text{cons}(0, B*\text{nat}) |]$
 $\implies (\lambda x \in A. \{fst(x). x \in u(x)-\{0\}\})'B \subseteq B$ "

$\langle proof \rangle$

lemma lemma5_3:

" $[| n \in \text{nat}; B \in A; 0 \in u(B); u(B) \lesssim \text{succ}(n) |]$
 $\implies (\lambda x \in A. \{fst(x). x \in u(x)-\{0\}\})'B \lesssim n$ "

$\langle proof \rangle$

lemma ex_fun_AC13_AC15:

" $[| \forall B \in \{\text{cons}(0, x*\text{nat}). x \in A\}. \text{pairwise_disjoint}(f'B) \ \& \ \text{sets_of_size_between}(f'B, 2, \text{succ}(n)) \ \& \ \text{Union}(f'B)=B;$

$n \in \text{nat} |]$

$\implies \exists f. \forall B \in A. f'B \neq 0 \ \& \ f'B \subseteq B \ \& \ f'B \lesssim n$ "

$\langle proof \rangle$

theorem AC10_AC11: " $[| n \in \text{nat}; 1 \leq n; \text{AC10}(n) |] \implies \text{AC11}$ "

$\langle proof \rangle$

theorem AC11_AC12: "AC11 ==> AC12"
 <proof>

theorem AC12_AC15: "AC12 ==> AC15"
 <proof>

lemma OUN_eq_UN: "Ord(x) ==> ($\bigcup a < x. F(a)$) = ($\bigcup a \in x. F(a)$)"
 <proof>

lemma AC15_W06_aux1:
 " $\forall x \in \text{Pow}(A) - \{0\}. f'x \neq 0 \ \& \ f'x \subseteq x \ \& \ f'x \lesssim m$
 ==> ($\bigcup i < \text{LEAST } x. \text{HH}(f, A, x) = \{A\}. \text{HH}(f, A, i)$) = A"
 <proof>

lemma AC15_W06_aux2:
 " $\forall x \in \text{Pow}(A) - \{0\}. f'x \neq 0 \ \& \ f'x \subseteq x \ \& \ f'x \lesssim m$
 ==> $\forall x < (\text{LEAST } x. \text{HH}(f, A, x) = \{A\}). \text{HH}(f, A, x) \lesssim m$ "
 <proof>

theorem AC15_W06: "AC15 ==> W06"
 <proof>

theorem AC10_AC13: "[| n ∈ nat; 1 ≤ n; AC10(n) |] ==> AC13(n)"
 <proof>

lemma *AC1_AC13*: " $AC1 \implies AC13(1)$ "
 $\langle proof \rangle$

lemma *AC13_mono*: " $[\mid m \leq n; AC13(m) \mid] \implies AC13(n)$ "
 $\langle proof \rangle$

theorem *AC13_AC14*: " $[\mid n \in nat; 1 \leq n; AC13(n) \mid] \implies AC14$ "
 $\langle proof \rangle$

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

lemma *lemma_aux*: " $[\mid A \neq 0; A \lesssim 1 \mid] \implies \exists a. A = \{a\}$ "
 $\langle proof \rangle$

lemma *AC13_AC1_lemma*:

$$\begin{aligned} & \forall B \in A. f(B) \neq 0 \ \& \ f(B) \leq B \ \& \ f(B) \lesssim 1 \\ & \implies (\lambda x \in A. \text{THE } y. f(x) = \{y\}) \in (\prod X \in A. X) \end{aligned}$$
 $\langle proof \rangle$

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

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

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

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

lemma InfCard_Least_in:

$$[| \text{InfCard}(x); y \subseteq x; y \approx \text{succ}(z) |] \implies (\text{LEAST } i. i \in y) \in y$$
 $\langle proof \rangle$

lemma subsets_lepoll_lemma1:

$$[| \text{InfCard}(x); n \in \text{nat} |]$$

$$\implies \{y \in \text{Pow}(x). y \approx \text{succ}(\text{succ}(n))\} \lesssim x * \{y \in \text{Pow}(x). y \approx \text{succ}(n)\}$$

$\langle proof \rangle$

lemma *set_of_Ord_succ_Union*: " $(\forall y \in z. \text{Ord}(y)) \implies z \subseteq \text{succ}(\text{Union}(z))$ "
 $\langle proof \rangle$

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

lemma *succ_Union_not_mem*:
" $(\neg \exists y. y \in z \implies \text{Ord}(y)) \implies \text{succ}(\text{Union}(z)) \notin z$ "
 $\langle proof \rangle$

lemma *Union_cons_eq_succ_Union*:
" $\text{Union}(\text{cons}(\text{succ}(\text{Union}(z)), z)) = \text{succ}(\text{Union}(z))$ "
 $\langle proof \rangle$

lemma *Un_Ord_disj*: " $[| \text{Ord}(i); \text{Ord}(j) |] \implies i \cup j = i \mid i \cup j = j$ "
 $\langle proof \rangle$

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

lemma *Union_in_lemma* [rule_format]:
" $n \in \text{nat} \implies \forall z. (\forall y \in z. \text{Ord}(y)) \ \& \ z \approx n \ \& \ z \neq 0 \implies \text{Union}(z) \in z$ "
 $\langle proof \rangle$

lemma *Union_in*: " $[| \forall x \in z. \text{Ord}(x); z \approx n; z \neq 0; n \in \text{nat} |] \implies \text{Union}(z) \in z$ "
 $\langle proof \rangle$

lemma *succ_Union_in_x*:
" $[| \text{InfCard}(x); z \in \text{Pow}(x); z \approx n; n \in \text{nat} |] \implies \text{succ}(\text{Union}(z)) \in x$ "
 $\langle proof \rangle$

lemma *succ_lepoll_succ_succ*:
" $[| \text{InfCard}(x); n \in \text{nat} |] \implies \{y \in \text{Pow}(x). y \approx \text{succ}(n)\} \lesssim \{y \in \text{Pow}(x). y \approx \text{succ}(\text{succ}(n))\}$ "
 $\langle proof \rangle$

lemma *subsets_eqpoll_X*:
" $[| \text{InfCard}(X); n \in \text{nat} |] \implies \{Y \in \text{Pow}(X). Y \approx \text{succ}(n)\} \approx X$ "
 $\langle proof \rangle$

lemma *image_vimage_eq*:
" $[| f \in \text{surj}(A, B); y \subseteq B |] \implies f^{-1}(\text{converse}(f)^{-1}y) = y$ "
 $\langle proof \rangle$

```

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,
        %g r. if (∃ y ∈ r. h‘g ⊆ y) then r
          else r Un {f‘(LEAST 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) Un
        {f ' (LEAST 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:
  "[| ∀ y<x. ∀ z<a. z<y | (∃ Y ∈ F(y). f(z)<=Y) --> (∃ ! Y. Y ∈ F(y)
  & f(z)<=Y);
  ∀ i j. i≤j --> F(i) ⊆ F(j); j≤i; i<x; z<a;
  V ∈ F(i); f(z)<=V; W ∈ F(j); f(z)<=W |]"

```


==> V = W"
 <proof>

lemma lemma3:
 "[| $\forall y < x. \forall z < a. z < y \mid (\exists Y \in F(y). f(z) \leq Y) \rightarrow (\exists ! Y. Y \in F(y) \& f(z) \leq Y)$;
 $\forall i j. i \leq j \rightarrow F(i) \subseteq F(j); i < x; j < x; z < a;$
 $V \in F(i); f(z) \leq V; W \in F(j); f(z) \leq W$ |]
 ==> V = W"
 <proof>

lemma lemma4:
 "[| $\forall y < x. F(y) \subseteq X \&$
 $(\forall x < a. x < y \mid (\exists Y \in F(y). h(x) \subseteq Y) \rightarrow$
 $(\exists ! Y. Y \in F(y) \& h(x) \subseteq Y))$;
 $x < a$ |]
 ==> $\forall y < x. \forall z < a. z < y \mid (\exists Y \in F(y). h(z) \subseteq Y) \rightarrow$
 $(\exists ! Y. Y \in F(y) \& h(z) \subseteq Y)$ "
 <proof>

lemma lemma5:
 "[| $\forall y < x. F(y) \subseteq X \&$
 $(\forall x < a. x < y \mid (\exists Y \in F(y). h(x) \subseteq Y) \rightarrow$
 $(\exists ! Y. Y \in F(y) \& h(x) \subseteq Y))$;
 $x < a; \text{Limit}(x); \forall i j. i \leq j \rightarrow F(i) \subseteq F(j)$ |]
 ==> $(\bigcup_{x < x} F(x)) \subseteq X \&$
 $(\forall xa < a. xa < x \mid (\exists x \in \bigcup_{x < x} F(x). h(xa) \subseteq x)$
 $\rightarrow (\exists ! Y. Y \in (\bigcup_{x < x} F(x)) \& h(xa) \subseteq Y))$ "
 <proof>

lemma dbl_Diff_eqpoll_Card:
 "[| $A \approx a; \text{Card}(a); \sim \text{Finite}(a); B < a; C < a$ |] ==> $A - B - C \approx a$ "
 <proof>

```

lemma Finite_lesspoll_infinite_Ord:
  "[| Finite(X); ~Finite(a); Ord(a) |] ==> X<a"
  <proof>

lemma Union_lesspoll:
  "[|  $\forall x \in X. x \text{ lepoll } n \ \& \ x \subseteq T$ ; well_ord(T, R); X lepoll b;
    b<a; ~Finite(a); Card(a); n  $\in$  nat |]
    ==> Union(X) <a"
  <proof>

lemma Un_sing_eq_cons: "A Un {a} = cons(a, A)"
  <proof>

lemma Un_lepoll_succ: "A lepoll B ==> A Un {a} lepoll succ(B)"
  <proof>

lemma Diff_UN_succ_empty: "Ord(a) ==> F(a) - ( $\bigcup b<\text{succ}(a). F(b)$ ) = 0"
  <proof>

lemma Diff_UN_succ_subset: "Ord(a) ==> F(a) Un X - ( $\bigcup b<\text{succ}(a). F(b)$ )
   $\subseteq$  X"
  <proof>

lemma recfunAC16_Diff_lepoll_1:
  "Ord(x)
    ==> recfunAC16(f, g, x, a) - ( $\bigcup i<x. \text{recfunAC16}(f, g, i, a)$ ) lepoll
    1"
  <proof>

lemma in_Least_Diff:
  "[| z  $\in$  F(x); Ord(x) |]
    ==> z  $\in$  F(LEAST i. z  $\in$  F(i)) - ( $\bigcup j<(\text{LEAST } i. z \in F(i)). F(j)$ )"
  <proof>

lemma Least_eq_imp_ex:
  "[| (LEAST i. w  $\in$  F(i)) = (LEAST i. z  $\in$  F(i));
    w  $\in$  ( $\bigcup i<a. F(i)$ ); z  $\in$  ( $\bigcup i<a. F(i)$ ) |]
    ==>  $\exists b<a. w \in (F(b) - (\bigcup c<b. F(c))) \ \& \ z \in (F(b) - (\bigcup c<b. F(c)))$ "
  <proof>

lemma two_in_lepoll_1: "[| A lepoll 1; a  $\in$  A; b  $\in$  A |] ==> a=b"
  <proof>

```

```

lemma UN_lepoll_index:
  "[|  $\forall i < a. F(i) \rightarrow (\bigcup_{j < i} F(j)) \text{ lepoll } 1$ ; Limit(a) |]
  ==>  $(\bigcup_{x < a} F(x)) \text{ lepoll } a$ "
<proof>

lemma recfunAC16_lepoll_index: "Ord(y) ==> recfunAC16(f, h, y, a) lepoll
y"
<proof>

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

lemma dbl_Diff_eqpoll:
  "[| recfunAC16(f, h, y, a)  $\subseteq \{X \in \text{Pow}(A) . X \approx \text{succ}(k \ \#\ m)\}$ ;
  Card(a);  $\sim \text{Finite}(a)$ ; A  $\approx$  a;
  k  $\in \text{nat}$ ; y < a;
  h  $\in \text{bij}(a, \{Y \in \text{Pow}(A). Y \approx \text{succ}(k)\})$  |]
  ==> A - Union(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,
    standard]

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  $\in \text{nat}$  |]
  ==> h ' i Un x  $\in \{x \in \text{Pow}(A) . x \approx k \ \#\ m\}$ "
<proof>

lemma lemma6:
  "[|  $\forall y < \text{succ}(j). F(y) \leq X \ \& \ (\forall x < a. x < y \mid P(x,y) \rightarrow Q(x,y))$ ; succ(j) < a
  |]
  ==> F(j)  $\leq$  X  $\ \& \ (\forall x < a. x < j \mid P(x,j) \rightarrow Q(x,j))$ "

```

$\langle proof \rangle$

lemma lemma7:

"[| $\forall x < a. x < j \mid P(x, j) \rightarrow Q(x, j); \text{succ}(j) < a$ |]
==> $P(j, j) \rightarrow (\forall x < a. x \leq j \mid P(x, j) \rightarrow Q(x, j))$ "

$\langle proof \rangle$

lemma ex_subset_eqpoll:

"[| $A \approx a; \sim \text{Finite}(a); \text{Ord}(a); m \in \text{nat}$ |] ==> $\exists X \in \text{Pow}(A). X \approx_m$ "

$\langle proof \rangle$

lemma subset_Un_disjoint: "[| $A \subseteq B \cup C; A \cap C = \emptyset$ |] ==> $A \subseteq B$ "

$\langle proof \rangle$

lemma Int_empty:

"[| $X \in \text{Pow}(A - \text{Union}(B) - C); T \in B; F \subseteq T$ |] ==> $F \cap X = \emptyset$ "

$\langle proof \rangle$

lemma subset_imp_eq_lemma:

" $m \in \text{nat} \Rightarrow \forall A B. A \subseteq B \ \& \ m \text{ lepoll } A \ \& \ B \text{ lepoll } m \rightarrow A=B$ "

$\langle proof \rangle$

lemma subset_imp_eq: "[| $A \subseteq B; m \text{ lepoll } A; B \text{ lepoll } m; m \in \text{nat}$ |] ==> $A=B$ "

$\langle proof \rangle$

lemma bij_imp_arg_eq:

"[| $f \in \text{bij}(a, \{Y \in X. Y \approx \text{succ}(k)\}); k \in \text{nat}; f' b \subseteq f' y; b < a; y < a$
|]
==> $b=y$ "

$\langle proof \rangle$

lemma ex_next_set:

```

"[/| recfunAC16(f, h, y, a)  $\subseteq$  {X  $\in$  Pow(A) . X $\approx$ succ(k #+ m)};
  Card(a); ~ Finite(a); A $\approx$ a;
  k  $\in$  nat; m  $\in$  nat; y<a;
  h  $\in$  bij(a, {Y  $\in$  Pow(A). Y $\approx$ succ(k)});
  ~ ( $\exists$  Y  $\in$  recfunAC16(f, h, y, a). h'y  $\subseteq$  Y) [/]
==>  $\exists$  X  $\in$  {Y  $\in$  Pow(A). Y $\approx$ succ(k #+ m)}. h'y  $\subseteq$  X &
  ( $\forall$  b<a. h'b  $\subseteq$  X -->
    ( $\forall$  T  $\in$  recfunAC16(f, h, y, a). ~ h'b  $\subseteq$  T))"
<proof>

```

```

lemma ex_next_Ord:
"[/| recfunAC16(f, h, y, a)  $\subseteq$  {X  $\in$  Pow(A) . X $\approx$ succ(k #+ m)};
  Card(a); ~ Finite(a); A $\approx$ a;
  k  $\in$  nat; m  $\in$  nat; y<a;
  h  $\in$  bij(a, {Y  $\in$  Pow(A). Y $\approx$ succ(k)});
  f  $\in$  bij(a, {Y  $\in$  Pow(A). Y $\approx$ succ(k #+ m)});
  ~ ( $\exists$  Y  $\in$  recfunAC16(f, h, y, a). h'y  $\subseteq$  Y) [/]
==>  $\exists$  c<a. h'y  $\subseteq$  f'c &
  ( $\forall$  b<a. h'b  $\subseteq$  f'c -->
    ( $\forall$  T  $\in$  recfunAC16(f, h, y, a). ~ h'b  $\subseteq$  T))"
<proof>

```

```

lemma lemma8:
"[/|  $\forall$  x<a. x<j | ( $\exists$  xa  $\in$  F(j). P(x, xa))
  --> ( $\exists$ ! Y. Y  $\in$  F(j) & P(x, Y)); F(j)  $\subseteq$  X;
  L  $\in$  X; P(j, L) & ( $\forall$  x<a. P(x, L) --> ( $\forall$  xa  $\in$  F(j). ~P(x, xa)))
[/]
==> F(j) Un {L}  $\subseteq$  X &
  ( $\forall$  x<a. x $\leq$ j | ( $\exists$  xa  $\in$  (F(j) Un {L}). P(x, xa)) -->
    ( $\exists$ ! Y. Y  $\in$  (F(j) Un {L}) & P(x, Y)))"
<proof>

```

```

lemma main_induct:
"[/| b < a; f  $\in$  bij(a, {Y  $\in$  Pow(A) . Y $\approx$ 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 &
      (⋃ b < a. f ' b) = A & (∀ b < a. f ' b ≲ m)"
⟨proof⟩

```

lemmas well_ord_paired = paired_bij [THEN bij_is_inj, THEN well_ord_rvimage]

lemma lepoll_trans1: "[| A \lesssim B; \sim A \lesssim C |] ==> \sim B \lesssim C"
 <proof>

lemmas lepoll_paired = paired_eqpoll [THEN eqpoll_sym, THEN eqpoll_imp_lepoll]

lemma lemma2: " $\exists y$ R. well_ord(y,R) & x Int y = 0 & $\sim y \lesssim z$ & \sim Finite(y)"
 <proof>

lemma infinite_Un: " \sim Finite(B) ==> \sim Finite(A Un B)"
 <proof>

lemma succ_not_lepoll_lemma:
 "[| $\sim(\exists x \in A. f'x=y)$; $f \in \text{inj}(A, B)$; $y \in B$ |]
 ==> ($\lambda a \in \text{succ}(A). \text{if}(a=A, y, f'a) \in \text{inj}(\text{succ}(A), B)$)"
 <proof>

lemma succ_not_lepoll_imp_eqpoll: "[| $\sim A \approx B$; A \lesssim B |] ==> succ(A)
 \lesssim B"
 <proof>

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

```

lemma cons_cons_subset:
  "[| a  $\subseteq$  y; b  $\in$  y-a; u  $\in$  x |] ==> cons(b, cons(u, a))  $\in$  Pow(x Un
y)"
<proof>

lemma cons_cons_eqpoll:
  "[| a  $\approx$  k; a  $\subseteq$  y; b  $\in$  y-a; u  $\in$  x; x Int y = 0 |]
  ==> cons(b, cons(u, a))  $\approx$  succ(succ(k))"
<proof>

lemma set_eq_cons:
  "[| succ(k)  $\approx$  A; k  $\approx$  B; B  $\subseteq$  A; a  $\in$  A-B; k  $\in$  nat |] ==> A = cons(a,
B)"
<proof>

lemma cons_eqE: "[| cons(x,a) = cons(y,a); x  $\notin$  a |] ==> x = y "
<proof>

lemma eq_imp_Int_eq: "A = B ==> A Int C = B Int C"
<proof>


lemma eqpoll_sum_imp_Diff_lepoll_lemma [rule_format]:
  "[| k  $\in$  nat; m  $\in$  nat |]
  ==>  $\forall A B. A \approx k \# m \ \& \ k \lesssim B \ \& \ B \subseteq A \ \rightarrow A-B \lesssim m$ "
<proof>

lemma eqpoll_sum_imp_Diff_lepoll:
  "[| A  $\approx$  succ(k # m); B  $\subseteq$  A; succ(k)  $\lesssim$  B; k  $\in$  nat; m  $\in$  nat |]
  ==> A-B  $\lesssim$  m"
<proof>


lemma eqpoll_sum_imp_Diff_eqpoll_lemma [rule_format]:
  "[| k  $\in$  nat; m  $\in$  nat |]
  ==>  $\forall A B. A \approx k \# m \ \& \ k \approx B \ \& \ B \subseteq A \ \rightarrow A-B \approx m$ "
<proof>

lemma eqpoll_sum_imp_Diff_eqpoll:
  "[| A  $\approx$  succ(k # m); B  $\subseteq$  A; succ(k)  $\approx$  B; k  $\in$  nat; m  $\in$  nat |]

```


$\Rightarrow A-B \approx m$
 $\langle proof \rangle$

lemma *subsets_lepoll_0_eq_unit*: " $\{x \in Pow(X). x \lesssim 0\} = \{0\}$ "
 $\langle proof \rangle$

lemma *subsets_lepoll_succ*:
 $"n \in nat \Rightarrow \{z \in Pow(y). z \lesssim succ(n)\} =$
 $\{z \in Pow(y). z \lesssim n\} \cup \{z \in Pow(y). z \approx succ(n)\}"$
 $\langle proof \rangle$

lemma *Int_empty*:
 $"n \in nat \Rightarrow \{z \in Pow(y). z \lesssim n\} \cap \{z \in Pow(y). z \approx succ(n)\}$
 $= \emptyset"$
 $\langle proof \rangle$

locale (open) *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 : " $MM == \{v \in t_n. succ(k) \lesssim v \text{ Int } y\}$ "
 and LL_def : " $LL == \{v \text{ Int } y. v \in MM\}$ "
 and GG_def : " $GG == \lambda v \in LL. (THE w. w \in MM \ \& \ v \subseteq w) - v$ "
 and s_def : " $s(u) == \{v \in t_n. u \in v \ \& \ k \lesssim v \text{ Int } y\}$ "
 assumes all_ex : " $\forall z \in \{z \in Pow(x \cup y) . z \approx succ(k)\}.$
 $\exists ! w. w \in t_n \ \& \ z \subseteq w$ "
 and $disjoint[iff]$: " $x \text{ Int } y = 0$ "
 and $"includes"$: " $t_n \subseteq \{v \in Pow(x \cup y). v \approx succ(k \#+ m)\}$ "
 and $WO_R[iff]$: " $well_ord(y, R)$ "
 and $lnat[iff]$: " $l \in nat$ "
 and $mnat[iff]$: " $m \in nat$ "
 and $mpos[iff]$: " $0 < m$ "
 and $Infinite[iff]$: " $\sim Finite(y)$ "
 and $noLepoll$: " $\sim y \lesssim \{v \in Pow(x). v \approx m\}$ "

lemma (in *AC16*) *knat* [iff]: " $k \in nat$ "
 $\langle proof \rangle$

lemma (in AC16) Diff_Finite_eqpoll: "[| l \approx a; a \subseteq y |] ==> y - a \approx y"
 <proof>

lemma (in AC16) s_subset: "s(u) \subseteq t_n"
 <proof>

lemma (in AC16) sI:
 "[| w \in t_n; cons(b, cons(u, a)) \subseteq w; a \subseteq y; b \in y-a; l \approx a |]
 ==> w \in s(u)"
 <proof>

lemma (in AC16) in_s_imp_u_in: "v \in s(u) ==> u \in v"
 <proof>

lemma (in AC16) ex1_superset_a:
 "[| l \approx a; a \subseteq y; b \in y - a; u \in x |]
 ==> $\exists!$ c. c \in s(u) & a \subseteq c & b \in c"
 <proof>

lemma (in AC16) the_eq_cons:
 "[| $\forall v \in s(u). \text{succ}(l) \approx v$ Int y;
 l \approx a; a \subseteq y; b \in y - a; u \in x |]
 ==> (THE c. c \in s(u) & a \subseteq c & b \in c) Int y = cons(b, a)"
 <proof>

lemma (in AC16) y_lepoll_subset_s:
 "[| $\forall v \in s(u). \text{succ}(l) \approx v$ Int y;
 l \approx a; a \subseteq y; u \in x |]
 ==> y \lesssim {v \in s(u). a \subseteq v}"
 <proof>

lemma (in AC16) x_imp_not_y [dest]: "a \in x ==> a \notin y"
 <proof>

lemma (in AC16) w_Int_eq_w_Diff:
 "w \subseteq x Un y ==> w Int (x - {u}) = w - cons(u, w Int y)"
 <proof>

```

lemma (in AC16) w_Int_eqpoll_m:
  "[| w ∈ {v ∈ s(u). a ⊆ v};
    l ≈ a; u ∈ x;
    ∀ v ∈ s(u). succ(l) ≈ v Int y |]
  ==> w Int (x - {u}) ≈ m"
⟨proof⟩

```

```

lemma (in AC16) eqpoll_m_not_empty: "a ≈ m ==> a ≠ 0"
⟨proof⟩

```

```

lemma (in AC16) cons_cons_in:
  "[| z ∈ xa Int (x - {u}); l ≈ a; a ⊆ y; u ∈ x |]
  ==> ∃! w. w ∈ t_n & cons(z, cons(u, a)) ⊆ w"
⟨proof⟩

```

```

lemma (in AC16) subset_s_lepoll_w:
  "[| ∀ v ∈ s(u). succ(l) ≈ v Int y; a ⊆ y; l ≈ a; u ∈ x |]
  ==> {v ∈ s(u). a ⊆ v} ≲ {v ∈ Pow(x). v ≈ m}"
⟨proof⟩

```

```

lemma (in AC16) well_ord_subsets_eqpoll_n:
  "n ∈ nat ==> ∃ S. well_ord({z ∈ Pow(y) . z ≈ succ(n)}, S)"
⟨proof⟩

```

```

lemma (in AC16) well_ord_subsets_lepoll_n:
  "n ∈ nat ==> ∃ R. well_ord({z ∈ Pow(y). z ≲ n}, R)"
⟨proof⟩

```

```

lemma (in AC16) LL_subset: "LL ⊆ {z ∈ Pow(y). z ≲ succ(k #+ m)}"
⟨proof⟩

```

```

lemma (in AC16) well_ord_LL: "∃ S. well_ord(LL, S)"
⟨proof⟩

```

```

lemma (in AC16) unique_superset_in_MM:
  "v ∈ LL ==> ∃! w. w ∈ MM & v ⊆ w"
⟨proof⟩

```

```

lemma (in AC16) Int_in_LL: "w ∈ MM ==> w Int y ∈ LL"
⟨proof⟩

```

```

lemma (in AC16) in_LL_eq_Int:
  "v ∈ LL ==> v = (THE x. x ∈ MM & v ⊆ x) Int y"
⟨proof⟩

```

```

lemma (in AC16) unique_superset1: "a ∈ LL ==> (THE x. x ∈ MM ∧ a ⊆
x) ∈ MM"
⟨proof⟩

```

```

lemma (in AC16) the_in_MM_subset:
  "v ∈ LL ==> (THE x. x ∈ MM & v ⊆ x) ⊆ x Un y"
⟨proof⟩

```

```

lemma (in AC16) GG_subset: "v ∈ LL ==> GG ' v ⊆ x"
⟨proof⟩

```

```

lemma (in AC16) nat_lepoll_ordertype: "nat ≲ ordertype(y, R)"
⟨proof⟩

```

```

lemma (in AC16) ex_subset_eqpoll_n: "n ∈ nat ==> ∃z. z ⊆ y & n ≈ z"
⟨proof⟩

```

```

lemma (in AC16) exists_proper_in_s: "u ∈ x ==> ∃v ∈ s(u). succ(k) ≲
v Int y"
⟨proof⟩

```

```

lemma (in AC16) exists_in_MM: "u ∈ x ==> ∃w ∈ MM. u ∈ w"
⟨proof⟩

```

```

lemma (in AC16) exists_in_LL: "u ∈ x ==> ∃w ∈ LL. u ∈ GG'w"
⟨proof⟩

```

```

lemma (in AC16) OUN_eq_x: "well_ord(LL,S) ==>
  ( $\bigcup b < \text{ordertype}(LL,S). \text{GG } '(\text{converse}(\text{ordermap}(LL,S)) \text{ ' } b) = x$ "
  <proof>

```

```

lemma (in AC16) in_MM_eqpoll_n: "w ∈ MM ==> w ≈ succ(k #+ m)"
  <proof>

```

```

lemma (in AC16) in_LL_eqpoll_n: "w ∈ LL ==> succ(k) ≲ w"
  <proof>

```

```

lemma (in AC16) in_LL: "w ∈ LL ==> w ⊆ (THE x. x ∈ MM ∧ w ⊆ x)"
  <proof>

```

```

lemma (in AC16) all_in_lepoll_m:
  "well_ord(LL,S) ==>
     $\forall b < \text{ordertype}(LL,S). \text{GG } '(\text{converse}(\text{ordermap}(LL,S)) \text{ ' } b) \lesssim m$ "
  <proof>

```

```

lemma (in AC16) conclusion:
  " $\exists a f. \text{Ord}(a) \ \& \ \text{domain}(f) = a \ \& \ (\bigcup b < a. f \text{ ' } b) = x \ \& \ (\forall b < a. f \text{ ' } b \lesssim m)$ "
  <proof>

```

```

theorem AC16_W04:
  " $[| \text{AC16}(k \text{ \#+ } m, k); 0 < k; 0 < m; k \in \text{nat}; m \in \text{nat} \ |] ==> \text{W04}(m)$ "
  <proof>

```

```

end

```

```

theory AC17_AC1 imports HH begin

```

```

lemma AC0_AC1_lemma: " $[| f: (\Pi X \in A. X); D \subseteq A \ |] ==> \exists g. g: (\Pi X \in D. X)$ "
  <proof>

```

lemma *AC0_AC1*: "*AC0* ==> *AC1*"

<proof>

lemma *AC1_AC0*: "*AC1* ==> *AC0*"

<proof>

lemma *AC1_AC17_lemma*: "*f* ∈ ($\prod X \in \text{Pow}(A) - \{0\}. X$) ==> *f* ∈ ($\text{Pow}(A) - \{0\} \rightarrow A$)"

<proof>

lemma *AC1_AC17*: "*AC1* ==> *AC17*"

<proof>

lemma *UN_eq_imp_well_ord*:

"[| *x* - ($\bigcup j \in \text{LEAST } i. \text{HH}(\lambda X \in \text{Pow}(x) - \{0\}. \{f'X\}, x, i) = \{x\}.$

$\text{HH}(\lambda X \in \text{Pow}(x) - \{0\}. \{f'X\}, x, j)) = 0;$

$f \in \text{Pow}(x) - \{0\} \rightarrow x$ |]

==> $\exists r. \text{well_ord}(x, r)$ "

<proof>

lemma *not_AC1_imp_ex*:

"~*AC1* ==> $\exists A. \forall f \in \text{Pow}(A) - \{0\} \rightarrow A. \exists u \in \text{Pow}(A) - \{0\}. f'u \notin u$ "

<proof>

lemma *AC17_AC1_aux1*:

"[| $\forall f \in \text{Pow}(x) - \{0\} \rightarrow x. \exists u \in \text{Pow}(x) - \{0\}. f'u \notin u;$

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

$x - (\bigcup a \in (\text{LEAST } i. \text{HH}(\lambda X \in \text{Pow}(x) - \{0\}. \{f'X\}, x, i) = \{x\}).$

$\text{HH}(\lambda X \in \text{Pow}(x) - \{0\}. \{f'X\}, x, a)) = 0$ |]

==> *P*"

<proof>

```

lemma AC17_AC1_aux2:
  "~ (∃ f ∈ Pow(x) - {0} -> x. x - F(f) = 0)
  ==> (λ f ∈ Pow(x) - {0} -> x . x - F(f))
      ∈ (Pow(x) - {0} -> x) -> Pow(x) - {0}"
<proof>

lemma AC17_AC1_aux3:
  "[| f'Z ∈ Z; Z ∈ Pow(x) - {0} |]"
  ==> (λ X ∈ Pow(x) - {0}. {f'X})'Z ∈ Pow(Z) - {0}"
<proof>

lemma AC17_AC1_aux4:
  "∃ f ∈ F. f'((λ f ∈ F. Q(f))'f) ∈ (λ f ∈ F. Q(f))'f
  ==> ∃ f ∈ F. f'Q(f) ∈ Q(f)"
<proof>

lemma AC17_AC1: "AC17 ==> AC1"
<proof>

```

```

lemma AC1_AC2_aux1:
  "[| f: (Π X ∈ A. X); B ∈ A; 0 ∉ A |]" ==> {f'B} ⊆ B Int {f'C. C
  ∈ A}"
<proof>

lemma AC1_AC2_aux2:
  "[| pairwise_disjoint(A); B ∈ A; C ∈ A; D ∈ B; D ∈ C |]" ==>
  f'B = f'C"
<proof>

lemma AC1_AC2: "AC1 ==> AC2"
<proof>

```

```

lemma AC2_AC1_aux1: "0 ∉ A ==> 0 ∉ {B*{B}. B ∈ A}"
<proof>

lemma AC2_AC1_aux2: "[| X*{X} Int C = {y}; X ∈ A |]"
  ==> (THE y. X*{X} Int C = {y}): X*A"

```

$\langle proof \rangle$

lemma AC2_AC1_aux3:

" $\forall D \in \{E * \{E\}. E \in A\}. \exists y. D \text{ Int } C = \{y\}$
 $\implies (\lambda x \in A. \text{fst}(\text{THE } z. (x * \{x\} \text{ Int } C = \{z\}))) \in (\Pi X \in A. X)$ "

$\langle proof \rangle$

lemma AC2_AC1: "AC2 \implies AC1"

$\langle proof \rangle$

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

$\langle proof \rangle$

lemma AC1_AC4: "AC1 \implies AC4"

$\langle proof \rangle$

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

$\langle proof \rangle$

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

$\langle proof \rangle$

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

$\langle proof \rangle$

lemma AC4_AC3: "AC4 \implies AC3"

$\langle proof \rangle$

lemma AC3_AC1_lemma:

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

$\langle proof \rangle$

lemma AC3_AC1: "AC3 \implies AC1"

$\langle proof \rangle$


```
lemma AC4_AC5: "AC4 ==> AC5"
<proof>
```

```
lemma AC5_AC4_aux1: "R ⊆ A*B ==> (λx ∈ R. fst(x)) ∈ R -> A"
<proof>
```

```
lemma AC5_AC4_aux2: "R ⊆ A*B ==> range(λx ∈ R. fst(x)) = domain(R)"
<proof>
```

```
lemma AC5_AC4_aux3: "[| ∃ f ∈ A->C. P(f,domain(f)); A=B |] ==> ∃ f ∈
B->C. P(f,B)"
<proof>
```

```
lemma AC5_AC4_aux4: "[| R ⊆ A*B; g ∈ C->R; ∀ x ∈ C. (λz ∈ R. fst(z))'
(g'x) = x |]
==> (λx ∈ C. snd(g'x)): (Π x ∈ C. R' '{x})"
<proof>
```

```
lemma AC5_AC4: "AC5 ==> AC4"
<proof>
```

```
lemma AC1_iff_AC6: "AC1 <-> AC6"
<proof>
```

```
end
```

```
theory AC18_AC19 imports AC_Equiv begin
```

```
definition
```

```
uu      :: "i => i" where
"uu(a) == {c Un {0}. c ∈ a}"
```

lemma *PROD_subsets*:
 "[| f ∈ (Π b ∈ {P(a). a ∈ A}. b); ∀ a ∈ A. P(a) ≤ Q(a) |]
 ==> (λ a ∈ A. f'P(a)) ∈ (Π a ∈ A. Q(a))"
 <proof>

lemma *lemma_AC18*:
 "[| ∀ A. 0 ∉ A --> (∃ f. f ∈ (Π X ∈ A. X)); A ≠ 0 |]
 ==> (∩ a ∈ A. ∪ b ∈ B(a). X(a, b)) ⊆
 (∪ f ∈ Π a ∈ A. B(a). ∩ a ∈ A. X(a, f'a))"
 <proof>

lemma *AC1_AC18*: "AC1 ==> PROP AC18"
 <proof>

theorem (in *AC18*) *AC19*
 <proof>

lemma *RepRep_conj*:
 "[| A ≠ 0; 0 ∉ A |] ==> {uu(a). a ∈ A} ≠ 0 & 0 ∉ {uu(a). a
 ∈ A}"
 <proof>

lemma *lemma1_1*: "[| c ∈ a; x = c Un {0}; x ∉ a |] ==> x - {0} ∈ a"
 <proof>

lemma *lemma1_2*:
 "[| f'(uu(a)) ∉ a; f ∈ (Π B ∈ {uu(a). a ∈ A}. B); a ∈ A |]
 ==> f'(uu(a)) - {0} ∈ a"
 <proof>

lemma *lemma1*: "∃ f. f ∈ (Π B ∈ {uu(a). a ∈ A}. B) ==> ∃ f. f ∈ (Π
 B ∈ A. B)"
 <proof>

lemma *lemma2_1*: "a ≠ 0 ==> 0 ∈ (∪ b ∈ uu(a). b)"
 <proof>

lemma lemma2: "[| A ≠ 0; 0 ∉ A |] ==> (⋂ x ∈ {uu(a). a ∈ A}. ⋃ b ∈ x.
b) ≠ 0"
⟨proof⟩

lemma AC19_AC1: "AC19 ==> AC1"
⟨proof⟩

end

theory DC imports AC_Equiv Hartog Cardinal_aux **begin**

lemma RepFun_lepoll: "Ord(a) ==> {P(b). b ∈ a} ≲ a"
⟨proof⟩

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

lemma image_Ord_lepoll: "[| f ∈ X->Y; Ord(X) |] ==> f'X ≲ Y"
⟨proof⟩

lemma range_subset_domain:
" [| R ⊆ X*X; !!g. g ∈ X ==> ∃u. <g,u> ∈ R |]
==> range(R) ⊆ domain(R)"
⟨proof⟩

lemma cons_fun_type: "g ∈ n->X ==> cons(<n,x>, g) ∈ succ(n) -> cons(x,
X)"
⟨proof⟩

lemma cons_fun_type2:
" [| g ∈ n->X; x ∈ X |] ==> cons(<n,x>, g) ∈ succ(n) -> X"
⟨proof⟩

lemma cons_image_n: "n ∈ nat ==> cons(<n,x>, g)'n = g'n"
⟨proof⟩

lemma cons_val_n: "g ∈ n->X ==> cons(<n,x>, g)'n = x"
⟨proof⟩

lemma cons_image_k: "k ∈ n ==> cons(<n,x>, g)'k = g'k"
⟨proof⟩

lemma cons_val_k: "[| k ∈ n; g ∈ n->X |] ==> cons(<n,x>, g)'k = g'k"
⟨proof⟩

lemma domain_cons_eq_succ: "domain(f)=x ==> domain(cons(<x,y>, f)) =
succ(x)"
⟨proof⟩

```

lemma restrict_cons_eq: "g ∈ n→X ==> restrict(cons(<n,x>, g), n) =
g"
⟨proof⟩

lemma succ_in_succ: "[| Ord(k); i ∈ k |] ==> succ(i) ∈ succ(k)"
⟨proof⟩

lemma restrict_eq_imp_val_eq:
  "[| restrict(f, domain(g)) = g; x ∈ domain(g) |]
  ==> f'x = g'x"
⟨proof⟩

lemma domain_eq_imp_fun_type: "[| domain(f)=A; f ∈ B→C |] ==> f ∈ A→C"
⟨proof⟩

lemma ex_in_domain: "[| R ⊆ A * B; R ≠ 0 |] ==> ∃x. x ∈ domain(R)"
⟨proof⟩

definition
  DC :: "i => o" where
    "DC(a) == ∀X R. R ⊆ Pow(X)*X &
      (∀Y ∈ Pow(X). Y < a --> (∃x ∈ X. <Y,x> ∈ R))
      --> (∃f ∈ a→X. ∀b<a. <f' b, f' b> ∈ R)"

definition
  DCO :: o where
    "DCO == ∀A B R. R ⊆ A*B & R≠0 & range(R) ⊆ domain(R)
      --> (∃f ∈ nat→domain(R). ∀n ∈ nat. <f' n, f' succ(n)>:R)"

definition
  ff :: "[i, i, i, i] => i" where
    "ff(b, X, Q, R) ==
      transrec(b, %c r. THE x. first(x, {x ∈ X. <r' c, x> ∈ R},
Q))"

locale (open) DCO_imp =
  fixes XX and RR and X and R

  assumes all_ex: "∀Y ∈ Pow(X). Y < nat --> (∃x ∈ X. <Y, x> ∈ R)"

  defines XX_def: "XX == (⋃n ∈ nat. {f ∈ n→X. ∀k ∈ n. <f' k, f' k>
∈ R})"
  and RR_def: "RR == {<z1,z2>:XX*XX. domain(z2)=succ(domain(z1))
& restrict(z2, domain(z1)) = z1}"

```

```
lemma (in DCO_imp) lemma1_1: "RR  $\subseteq$  XX*XX"
<proof>
```

```
lemma (in DCO_imp) lemma1_2: "RR  $\neq$  0"
<proof>
```

```
lemma (in DCO_imp) lemma1_3: "range(RR)  $\subseteq$  domain(RR)"
<proof>
```

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

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

```
lemma (in DCO_imp) lemma3:
```

```

"[/  $\forall n \in \text{nat}. \langle f'n, f'succ(n) \rangle \in RR; f \in \text{nat} \rightarrow XX; m \in \text{nat} \mid]$ 

 $\Rightarrow (\lambda x \in \text{nat}. f'succ(x)'x) \text{ `` } m = f'succ(m) \text{ `` } m$ 
<proof>

```

```

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

```

```

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

```

```

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

```

```

lemma (in imp_DC0) lemma4:
  "[/ range(R)  $\subseteq$  domain(R); x  $\in$  domain(R)  $\mid$ ]
 $\Rightarrow RR \subseteq \text{Pow}(XX)*XX$  &
  ( $\forall Y \in \text{Pow}(XX). Y \prec \text{nat} \rightarrow (\exists x \in XX. \langle Y,x \rangle : RR)$ )"
<proof>

```

```

lemma (in imp_DC0) UN_image_succ_eq:
  "[/ f  $\in \text{nat} \rightarrow X; n \in \text{nat} \mid]$ 
 $\Rightarrow (\bigcup x \in f'succ(n). P(x)) = P(f'n) \text{ Un } (\bigcup x \in f'n. P(x))$ "
<proof>

```

```

lemma (in imp_DC0) UN_image_succ_eq_succ:
  "[| ( $\bigcup x \in f'`n. P(x)$ ) = y;  $P(f'n) = \text{succ}(y)$ ;
     $f \in \text{nat} \rightarrow X$ ;  $n \in \text{nat}$  |] ==> ( $\bigcup x \in f'`\text{succ}(n). P(x)$ ) =  $\text{succ}(y)$ "
<proof>

```

```

lemma (in imp_DC0) apply_domain_type:
  "[|  $h \in \text{succ}(n) \rightarrow D$ ;  $n \in \text{nat}$ ;  $\text{domain}(h) = \text{succ}(y)$  |] ==>  $h'y \in D$ "
<proof>

```

```

lemma (in imp_DC0) image_fun_succ:
  "[|  $h \in \text{nat} \rightarrow X$ ;  $n \in \text{nat}$  |] ==>  $h'`\text{succ}(n) = \text{cons}(h'n, h'`n)$ "
<proof>

```

```

lemma (in imp_DC0) f_n_type:
  "[|  $\text{domain}(f'n) = \text{succ}(k)$ ;  $f \in \text{nat} \rightarrow XX$ ;  $n \in \text{nat}$  |]
  ==>  $f'n \in \text{succ}(k) \rightarrow \text{domain}(R)$ "
<proof>

```

```

lemma (in imp_DC0) f_n_pairs_in_R [rule_format]:
  "[|  $h \in \text{nat} \rightarrow XX$ ;  $\text{domain}(h'n) = \text{succ}(k)$ ;  $n \in \text{nat}$  |]
  ==>  $\forall i \in k. \langle h'n'i, h'n'\text{succ}(i) \rangle \in R$ "
<proof>

```

```

lemma (in imp_DC0) restrict_cons_eq_restrict:
  "[|  $\text{restrict}(h, \text{domain}(u)) = u$ ;  $h \in \text{nat} \rightarrow X$ ;  $\text{domain}(u) \subseteq n$  |]
  ==>  $\text{restrict}(\text{cons}(\langle n, y \rangle, h), \text{domain}(u)) = u$ "
<proof>

```

```

lemma (in imp_DC0) all_in_image_restrict_eq:
  "[|  $\forall x \in f'`n. \text{restrict}(f'n, \text{domain}(x)) = x$ ;
     $f \in \text{nat} \rightarrow XX$ ;
     $n \in \text{nat}$ ;  $\text{domain}(f'n) = \text{succ}(n)$ ;
    ( $\bigcup x \in f'`n. \text{domain}(x)$ )  $\subseteq n$  |]
  ==>  $\forall x \in f'`\text{succ}(n). \text{restrict}(\text{cons}(\langle \text{succ}(n), y \rangle, f'n), \text{domain}(x))$ 
  =  $x$ "
<proof>

```

```

lemma (in imp_DC0) simplify_recursion:
  "[|  $\forall b \in \text{nat}. \langle f'`b, f'`b \rangle \in R$ ;
     $f \in \text{nat} \rightarrow XX$ ;  $\text{range}(R) \subseteq \text{domain}(R)$ ;  $x \in \text{domain}(R)$  |]
  ==>  $\text{all}R$ "
<proof>

```

```

lemma (in imp_DC0) lemma2:
  "[|  $\text{all}R$ ;  $f \in \text{nat} \rightarrow XX$ ;  $\text{range}(R) \subseteq \text{domain}(R)$ ;  $x \in \text{domain}(R)$ ;  $n$ 
   $\in \text{nat}$  |]
  ==>  $f'n \in \text{succ}(n) \rightarrow \text{domain}(R) \ \& \ (\forall i \in n. \langle f'n'i, f'n'\text{succ}(i) \rangle \in R)$ "

```

<proof>

lemma (in imp_DC0) lemma3:

"[| allRR; f ∈ nat->XX; n ∈ nat; range(R) ⊆ domain(R); x ∈ domain(R)

|]

==> f'n'n = f'succ(n)'n"

<proof>

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>

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 |]

==> {x ∈ X. <Z, x> ∈ R} ≠ 0"

$\langle proof \rangle$

lemma *W01_DC_lemma*:

```
"[| Card(K); well_ord(X,Q);  
   $\forall Y \in Pow(X). Y \prec K \rightarrow (\exists x \in X. \langle Y, x \rangle \in R); b \in K$  |]  
  ==> ff(b, X, Q, R)  $\in \{x \in X. <(\lambda c \in b. ff(c, X, Q, R))\text{'b, x}'$   
   $\in R\}$ "  
 $\langle proof \rangle$ 
```

theorem *W01_DC_Card*: "*W01* ==> $\forall K. Card(K) \rightarrow DC(K)$ "
 $\langle proof \rangle$

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.