

Equivalents of the Axiom of Choice

Krzysztof Grąbczewski

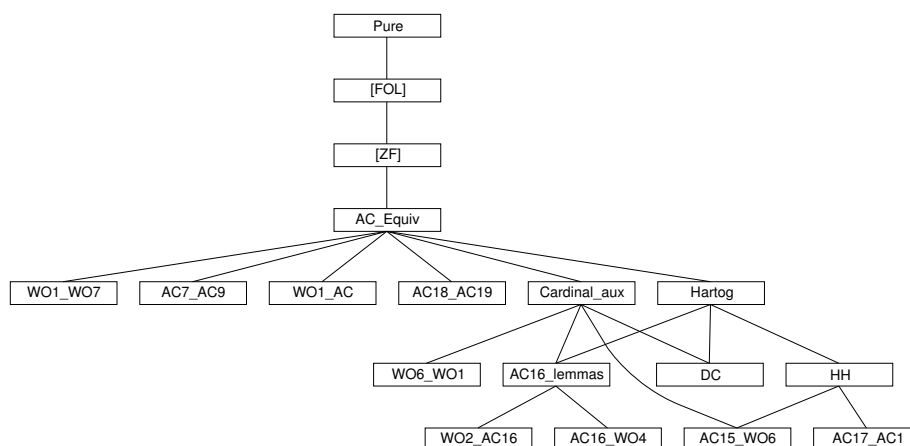
June 21, 2010

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	23
0.2	Lemmas used in the proofs of $AC1 \Rightarrow WO2$ and $AC17$ $\Rightarrow AC1$	24
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 == $\exists m \in \text{nat}. 1 \leq m \ \& \ AC13(m)$ "

definition

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

definition

"AC16(n, k) ==
 $\forall A. \sim \text{Finite}(A) \rightarrow$
 $(\exists T. T \subseteq \{X \in \text{Pow}(A). X \approx_{\text{succ}}(n)\} \ \& \$
 $(\forall X \in \{X \in \text{Pow}(A). X \approx_{\text{succ}}(k)\}. \exists ! Y. Y \in T \ \& \ X \subseteq Y))$ "

definition

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

locale AC18 =

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

definition

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

lemma rvimage_id: " $\text{rvimage}(A, \text{id}(A), r) = r \ \text{Int } A * A$ "
 <proof>

lemma ordertype_Int:

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

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

lemma inj_strengthen_type:

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

lemma *nat_not_Finite*: " $\sim \text{Finite}(\text{nat})$ "

<proof>

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

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

<proof>

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

<proof>

lemma *first_in_B*:

" $[| \text{well_ord}(\text{Union}(A), r); 0 \notin A; B \in A |] \implies (\text{THE } b. \text{first}(b, B, r)) \in B$ "

<proof>

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

<proof>

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

<proof>

lemma *lepoll_m_imp_domain_lepoll_m*:

" $[| m \in \text{nat}; u \lesssim m |] \implies \text{domain}(u) \lesssim m$ "

<proof>

```

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 $A \cup B \text{ lepoll } \text{---}A\text{---} + \text{---}B\text{---} = \max(\text{---}A\text{---}, \text{---}B\text{---}) \text{ lepoll } i$. 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>

schematic_lemma paired_bij: "?f \in bij({y,z}. y \in x), x)"
 <proof>

lemma paired_eqpoll: "{y,z}. y \in x \approx x"
 <proof>

lemma ex_eqpoll_disjoint: " \exists B. B \approx A & B Int C = 0"
 <proof>

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

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

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

lemma UN_subset_split:
 " $(\bigcup x \in X. P(x)) \subseteq (\bigcup x \in X. P(x)-Q(x)) \text{ Un } (\bigcup x \in X. Q(x))$ "
 <proof>

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

lemma UN_fun_lepoll_lemma [rule_format]:
 "[| well_ord(T, R); \sim Finite(a); Ord(a); n \in nat |]
 ==> $\forall f. (\forall b \in a. f'b \lesssim n \ \& \ f'b \subseteq T) \rightarrow (\bigcup b \in a. f'b) \lesssim a$ "
 <proof>

lemma UN_fun_lepoll:
 "[| $\forall b \in a. f'b \lesssim n \ \& \ f'b \subseteq T$; well_ord(T, R);
 \sim Finite(a); Ord(a); n \in nat |] ==> $(\bigcup b \in a. f'b) \lesssim a$ "
 <proof>

lemma UN_lepoll:
 "[| $\forall b \in a. F(b) \lesssim n \ \& \ F(b) \subseteq T$; well_ord(T, R);
 \sim Finite(a); Ord(a); n \in nat |]
 ==> $(\bigcup b \in a. F(b)) \lesssim a$ "
 <proof>

```

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

lemma lepoll_imp_eqpoll_subset:
  "a  $\lesssim$  X ==>  $\exists Y. Y \subseteq X$  & 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) ==  $\lambda b \in a++a.$  if  $b < a$  then vv1(f,m,b) 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) ==
     $\lambda g \in a++a.$  if  $g < a$  then vv2(f,b,g,s) 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"
 $\langle proof \rangle$

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

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

lemma *W04_W05*: "[*m* ∈ *nat*; 1 ≤ *m*; *W04*(*m*) *I*] ==> *W05*"
 $\langle proof \rangle$

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

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

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

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

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

lemma *uu_subset2*: "*uu*(*f*,*b*,*g*,*d*) ⊆ *f*'*d*"

$\langle proof \rangle$

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

lemma *cases*:

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

$\langle proof \rangle$

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

lemma *vv1_subset*: " $vv1(f, m, b) \subseteq f' b$ "
 $\langle proof \rangle$

lemma *UN_gg1_eq*:

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

lemma *domain_gg1*: " $domain(gg1(f, a, m)) = a++a$ "
 $\langle proof \rangle$

lemma *nested_LeastI*:

"[| $P(a, b); Ord(a); Ord(b);$
 $Least_a = (LEAST a. \exists x. Ord(x) \ \& \ P(a, x))$ |]
 $\implies P(Least_a, LEAST b. P(Least_a, b))$ "
 $\langle proof \rangle$

```

lemmas nested_Least_instance =
  nested_LeastI [of "%g d. domain(uu(f,b,g,d)) ≠ 0 &
                    domain(uu(f,b,g,d)) ≲ 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:
  "[|  $\forall g < a. \forall d < a. \text{domain}(\text{uu}(f, b, g, d)) \neq 0 \rightarrow$   

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

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

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

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

  ==>  $\text{uu}(f, b, g, \text{LEAST } d. \text{uu}(f, b, g, d) \neq 0) \in f' b \rightarrow f' g$ "
<proof>

lemma vv2_subset:
  "[|  $\forall g < a. \forall d < a. \text{domain}(\text{uu}(f, b, g, d)) \neq 0 \rightarrow$   

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

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

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

  ==>  $\text{vv2}(f, b, g, s) \subseteq f' g$ "
<proof>

lemma UN_gg2_eq:
  "[|  $\forall g < a. \forall d < a. \text{domain}(\text{uu}(f, b, g, d)) \neq 0 \rightarrow$   

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

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

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

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

lemma domain_gg2: "domain(gg2(f, a, b, s)) = a ++ a"
<proof>

lemma vv2_lepoll: "[|  $m \in \text{nat}; m \neq 0$  |] ==>  $\text{vv2}(f, b, g, s) \lesssim m$ "
<proof>

lemma ww2_lepoll:
  "[|  $\forall b < a. f' b \lesssim \text{succ}(m); g < a; m \in \text{nat}; \text{vv2}(f, b, g, d) \subseteq f' g$  |]  

  ==>  $\text{ww2}(f, b, g, d) \lesssim m$ "
<proof>

lemma gg2_lepoll_m:
  "[|  $\forall g < a. \forall d < a. \text{domain}(\text{uu}(f, b, g, d)) \neq 0 \rightarrow$   

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

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

```

```

      ( $\bigcup b < a. f(b) = y; \quad b < a; \quad s \in f(b); \quad m \in \text{nat}; \quad m \neq 0; \quad g < a + a \mid$ )
    ==> gg2(f,a,b,s) ' g  $\lesssim$  m"
  <proof>

```

```

lemma lemma_ii: "[| succ(m)  $\in$  NN(y); y*y  $\subseteq$  y; m  $\in$  nat; m  $\neq$  0  $\mid$ ] ==>
m  $\in$  NN(y)"
  <proof>

```

```

lemma z_n_subset_z_succ_n:
  " $\forall n \in \text{nat}. \text{rec}(n, x, \%k \ r. \ r \ \text{Un} \ r*r) \subseteq \text{rec}(\text{succ}(n), x, \%k \ r. \ r \ \text{Un} \ r*r)$ "
  <proof>

```

```

lemma le_subsets:
  "[|  $\forall n \in \text{nat}. f(n) \leq f(\text{succ}(n))$ ; n  $\leq$  m; n  $\in$  nat; m  $\in$  nat  $\mid$ ]
  ==> f(n)  $\leq$  f(m)"
  <proof>

```

```

lemma le_imp_rec_subset:
  "[| n  $\leq$  m; m  $\in$  nat  $\mid$ ]
  ==>  $\text{rec}(n, x, \%k \ r. \ r \ \text{Un} \ r*r) \subseteq \text{rec}(m, x, \%k \ r. \ r \ \text{Un} \ r*r)$ "
  <proof>

```

```

lemma lemma_iv: " $\exists y. x \ \text{Un} \ y*y \subseteq y$ "
  <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; Ord(a) |] ==> ∃ c<a. f' c
 = {x}"
 <proof>

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

lemma NN_imp_ex_inj: "1 ∈ NN(y) ==> ∃ a f. Ord(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 ∈ 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 ==> W01"
⟨proof⟩
```



```

lemma converse_Memrel_not_wf_on:
  "[| Ord(a); ~Finite(a) |] ==> ~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 \notin A; B \in A |] ==> (nat->Union(A)) * B \neq 0"
 <proof>

lemma *inj_lemma*:
 "C \in A ==> (λ g \in (nat->Union(A))*C.
 (λ n \in nat. if(n=0, snd(g), fst(g)^(n #- 1))))
 \in inj((nat->Union(A))*C, (nat->Union(A))) "
 <proof>

lemma *Sigma_fun_space_eqpoll*:
 "[| C \in A; 0 \notin A |] ==> (nat->Union(A)) * C \approx (nat->Union(A))"
 <proof>

lemma *AC6_AC7*: "AC6 ==> AC7"
 <proof>

lemma *lemma1_1*: "y \in (Π B \in A. Y*B) ==> (λ B \in A. snd(y'B)) \in (Π B \in A. B)"
 <proof>

lemma *lemma1_2*:
 "y \in (Π B \in {Y*C. C \in A}. B) ==> (λ B \in A. y'(Y*B)) \in (Π B \in A. Y*B)"
 <proof>

lemma *AC7_AC6_lemma1*:
 "(Π B \in {(nat->Union(A))*C. C \in A}. B) \neq 0 ==> (Π B \in A. B) \neq 0"
 <proof>

lemma *AC7_AC6_lemma2*: "0 \notin A ==> 0 \notin {(nat -> Union(A)) * C. C \in A}"

$\langle proof \rangle$

lemma AC7_AC6: "AC7 ==> AC6"

$\langle proof \rangle$

lemma AC1_AC8_lemma1:

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

$\langle proof \rangle$

lemma AC1_AC8_lemma2:

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

$\langle proof \rangle$

lemma AC1_AC8: "AC1 ==> AC8"

$\langle proof \rangle$

lemma AC8_AC9_lemma:

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

$\langle proof \rangle$

lemma AC8_AC9: "AC8 ==> AC9"

$\langle proof \rangle$

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

$\langle proof \rangle$

```

lemma nat_lepoll_lemma:
  "[| 0 ∉ A; B ∈ A |] ==> nat ≲ ((nat → Union(A)) × B) × nat"
⟨proof⟩

lemma AC9_AC1_lemma1:
  "[| 0 ∉ A; A ≠ 0;
    C = {(nat → Union(A)) * B} * nat. B ∈ A} 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 \lesssim X |] ==> $\exists Y. Y \subseteq X \ \& \ (\exists R. R \subseteq X*X \ \& \ \text{ordertype}(Y,R)=a)$ "

<proof>

lemma *Ords_lepoll_set_lemma*:

"($\forall a. \text{Ord}(a) \rightarrow a \lesssim X$) ==>

$\forall a. \text{Ord}(a) \rightarrow$

$a \in \{b. Z \in \text{Pow}(X)*\text{Pow}(X*X), \exists Y R. Z=\langle Y,R \rangle \ \& \ \text{ordertype}(Y,R)=b\}$ "

<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*: "[| $i < \text{Hartog}(A)$; $\sim i \lesssim A$ |] ==> P "

<proof>

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

<proof>

lemma *Card_Hartog*: " $\text{Card}(\text{Hartog}(A))$ "

<proof>

end

theory *HH*

imports *AC_Equiv Hartog*

begin

definition

HH :: " $[i, i, i] \Rightarrow i$ " **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*:

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

<proof>

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

<proof>

lemma *subset_imp_Diff_eq*:

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

<proof>

lemma *Ord_DiffE*: " $[\mid c \in a - b; b < a \mid] \implies c = b \mid b < c \ \& \ c < a$ "

<proof>

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

<proof>

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

<proof>

lemma *HH_is_x_gt_too*: " $[\mid HH(f, x, b) = \{x\}; b < a \mid] \implies HH(f, x, a) = \{x\}$ "

<proof>

lemma *HH_subset_x_lt_too*:

" $[\mid HH(f, x, a) \in \text{Pow}(x) - \{0\}; b < a \mid] \implies HH(f, x, b) \in \text{Pow}(x) - \{0\}$ "

<proof>

lemma *HH_subset_x_imp_subset_Diff_UN*:

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

<proof>

lemma *HH_eq_arg_lt*:

" $[\mid HH(f, x, v) = HH(f, x, w); HH(f, x, v) \in \text{Pow}(x) - \{0\}; v \in w \mid] \implies P$ "

<proof>

lemma *HH_eq_imp_arg_eq*:

" $[\mid HH(f, x, v) = HH(f, x, w); HH(f, x, w) \in \text{Pow}(x) - \{0\}; \text{Ord}(v); \text{Ord}(w) \mid] \implies v = w$ "

<proof>

lemma *HH_subset_x_imp_lepoll*:

" $[\mid HH(f, x, i) \in \text{Pow}(x) - \{0\}; \text{Ord}(i) \mid] \implies i \text{ lepoll } \text{Pow}(x) - \{0\}$ "

<proof>

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

$\langle proof \rangle$

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

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

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

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

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

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

 $\implies (\lambda a \in A. F(a)) \in \text{surj}(A, \{\{y\}. y \in x\})$ "
 $\langle proof \rangle$

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

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

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

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

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

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


```

lemma f_sing_imp_HH_sing:
  "[| f ∈ (Pow(x)-{0}) -> {{z}. z ∈ x};
    a ∈ (LEAST i. HH(f,x,i)={x}) |] ==> ∃ z ∈ x. HH(f,x,a) = {z}"
⟨proof⟩

lemma f_sing_lam_bij:
  "[| x - (⋃ j ∈ (LEAST i. HH(f,x,i)={x}). HH(f,x,j)) = 0;
    f ∈ (Pow(x)-{0}) -> {{z}. z ∈ x} |]
  ==> (λ a ∈ (LEAST i. HH(f,x,i)={x}). HH(f,x,a))
    ∈ bij(LEAST i. HH(f,x,i)={x}, {{y}. y ∈ x})"
⟨proof⟩

lemma lam_singI:
  "f ∈ (Π X ∈ Pow(x)-{0}. F(X))
  ==> (λ X ∈ 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(λ X ∈ 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"
⟨proof⟩

lemma lemma3:
  "∀ B ∈ {cons(0, x*nat). x ∈ A}. pairwise_disjoint(f'B) &
    sets_of_size_between(f'B, 2, n) & Union(f'B)=B
  ==> ∀ B ∈ A. ∃ ! u. u ∈ f'cons(0, B*nat) & u ⊆ cons(0, B*nat) &

    0 ∈ u & 2 ≲ u & u ≲ n"
⟨proof⟩

lemma lemma4: "[| A ≲ i; Ord(i) |] ==> {P(a). a ∈ A} ≲ i"
⟨proof⟩

lemma lemma5_1:
  "[| B ∈ A; 2 ≲ u(B) |] ==> (λx ∈ A. {fst(x). x ∈ u(x)-{0}})'B ≠
0"
⟨proof⟩

lemma lemma5_2:
  "[| B ∈ A; u(B) ⊆ cons(0, B*nat) |]
  ==> (λx ∈ A. {fst(x). x ∈ u(x)-{0}})'B ⊆ B"
⟨proof⟩

lemma lemma5_3:
  "[| n ∈ nat; B ∈ A; 0 ∈ u(B); u(B) ≲ succ(n) |]
  ==> (λx ∈ A. {fst(x). x ∈ u(x)-{0}})'B ≲ n"
⟨proof⟩

lemma ex_fun_AC13_AC15:
  "[| ∀ B ∈ {cons(0, x*nat). x ∈ A}.
    pairwise_disjoint(f'B) &
    sets_of_size_between(f'B, 2, succ(n)) & Union(f'B)=B;

    n ∈ 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: "[| $n \in nat$; $1 \leq n$; $AC10(n)$ |] $\implies AC11$ "
 $\langle proof \rangle$

theorem AC11_AC12: " $AC11 \implies AC12$ "
 $\langle proof \rangle$

theorem AC12_AC15: " $AC12 \implies AC15$ "
 $\langle proof \rangle$

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

lemma AC15_W06_aux1:

$$\begin{aligned} & " \forall x \in Pow(A) - \{0\}. f'x \neq 0 \ \& \ f'x \subseteq x \ \& \ f'x \lesssim m \\ \implies & (\bigcup i < LEAST \ x. HH(f, A, x) = \{A\}. HH(f, A, i)) = A " \end{aligned}$$

 $\langle proof \rangle$

lemma AC15_W06_aux2:

$$\begin{aligned} & " \forall x \in Pow(A) - \{0\}. f'x \neq 0 \ \& \ f'x \subseteq x \ \& \ f'x \lesssim m \\ \implies & \forall x < (LEAST \ x. HH(f, A, x) = \{A\}). HH(f, A, x) \lesssim m " \end{aligned}$$

 $\langle proof \rangle$

theorem AC15_W06: " $AC15 \implies W06$ "
 $\langle proof \rangle$

theorem *AC10_AC13*: "[| $n \in \text{nat}; 1 \leq n; \text{AC10}(n)$ |] ==> $\text{AC13}(n)$ "
<proof>

lemma *AC1_AC13*: " $\text{AC1} ==> \text{AC13}(1)$ "
<proof>

lemma *AC13_mono*: "[| $m \leq n; \text{AC13}(m)$ |] ==> $\text{AC13}(n)$ "
<proof>

theorem *AC13_AC14*: "[| $n \in \text{nat}; 1 \leq n; \text{AC13}(n)$ |] ==> AC14 "
<proof>

theorem AC14_AC15: "AC14 ==> AC15"
 <proof>

lemma lemma_aux: "[| A ≠ 0; A ≲ 1 |] ==> ∃ a. A = {a}"
 <proof>

lemma AC13_AC1_lemma:
 "∀ B ∈ A. f(B) ≠ 0 & f(B) ≤ B & f(B) ≲ 1
 ==> (λ x ∈ A. THE y. f(x) = {y}) ∈ (Π X ∈ A. X)"
 <proof>

theorem AC13_AC1: "AC13(1) ==> AC1"
 <proof>

theorem AC11_AC14: "AC11 ==> AC14"
 <proof>

end

theory AC16_lemmas
imports AC_Equiv Hartog Cardinal_aux
begin

lemma cons_Diff_eq: "a ∉ A ==> cons(a, A) - {a} = A"
 <proof>

lemma nat_1_lepoll_iff: "1 ≲ X <-> (∃ x. x ∈ X)"
 <proof>

lemma eqpoll_1_iff_singleton: "X ≈ 1 <-> (∃ x. X = {x})"
 <proof>

lemma cons_eqpoll_succ: "[| x ≈ n; y ∉ x |] ==> cons(y, x) ≈ succ(n)"
 <proof>

lemma subsets_eqpoll_1_eq: "{Y ∈ Pow(X). Y ≈ 1} = {{x}. x ∈ X}"
 <proof>

lemma eqpoll_RepFun_sing: "X ≈ {{x}. x ∈ X}"
 <proof>

lemma subsets_eqpoll_1_eqpoll: "{Y ∈ Pow(X). Y ≈ 1} ≈ X"
 <proof>

lemma InfCard_Least_in:
 "[| InfCard(x); y ⊆ x; y ≈ succ(z) |] ==> (LEAST i. i ∈ y) ∈ y"
 <proof>

lemma subsets_lepoll_lemma1:
 "[| InfCard(x); n ∈ nat |]
 ==> {y ∈ Pow(x). y ≈ succ(succ(n))} ≲ x * {y ∈ Pow(x). y ≈ succ(n)}"
 <proof>

lemma set_of_Ord_succ_Union: "(∀ y ∈ z. Ord(y)) ==> z ⊆ succ(Union(z))"
 <proof>

lemma subset_not_mem: "j ⊆ i ==> i ∉ j"
 <proof>

lemma succ_Union_not_mem:
 "(!!y. y ∈ z ==> Ord(y)) ==> succ(Union(z)) ∉ z"
 <proof>

lemma Union_cons_eq_succ_Union:
 "Union(cons(succ(Union(z)), z)) = succ(Union(z))"
 <proof>

lemma Un_Ord_disj: "[| Ord(i); Ord(j) |] ==> i Un j = i | i Un j = j"
 <proof>

lemma Union_eq_Un: "x ∈ X ==> Union(X) = x Un Union(X - {x})"
 <proof>

lemma Union_in_lemma [rule_format]:
 "n ∈ nat ==> ∀ z. (∀ y ∈ z. Ord(y)) & z ≈ n & z ≠ 0 --> Union(z) ∈ z"
 <proof>

lemma Union_in: "[| ∀ x ∈ z. Ord(x); z ≈ n; z ≠ 0; n ∈ nat |] ==> Union(z) ∈ z"
 <proof>

lemma succ_Union_in_x:

```

    "[| InfCard(x); z ∈ Pow(x); z≈n; n ∈ nat |] ==> succ(Union(z))
∈ x"
⟨proof⟩

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,
        %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*:

" $i \leq j \implies \text{recfunAC16}(f, g, i, a) \subseteq \text{recfunAC16}(f, g, j, a)$ "
 <proof>

lemma *lemma3_1*:

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

lemma *lemma3*:

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

lemma *lemma4*:

"[$\forall y < x. F(y) \subseteq X \&$
 $(\forall x < a. x < y \mid (\exists Y \in F(y). h(x) \subseteq Y) \implies$
 $(\exists ! Y. Y \in F(y) \& h(x) \subseteq Y));$
 $x < a \mid]$
 $\implies \forall y < x. \forall z < a. z < y \mid (\exists Y \in F(y). h(z) \subseteq Y) \implies$
 $(\exists ! Y. Y \in F(y) \& h(z) \subseteq Y)"$
 <proof>

lemma *lemma5*:

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

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

lemma *Finite_lesspoll_infinite_Ord*:
 "[| $\text{Finite}(X)$; $\sim \text{Finite}(a)$; $\text{Ord}(a)$ |] $\implies X \prec a$ "
 <proof>

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

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

lemma *Un_lepoll_succ*: " $A \text{ lepoll } B \implies A \text{ Un } \{a\} \text{ lepoll } \text{succ}(B)$ "
 <proof>

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

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

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

1"
 <proof>

lemma in_Least_Diff:
 "[| z ∈ F(x); Ord(x) |]
 ==> z ∈ F(LEAST i. z ∈ F(i)) - (⋃ j<(LEAST i. z ∈ F(i)). F(j))"
 <proof>

lemma Least_eq_imp_ex:
 "[| (LEAST i. w ∈ F(i)) = (LEAST i. z ∈ F(i));
 w ∈ (⋃ i<a. F(i)); z ∈ (⋃ i<a. F(i)) |]
 ==> ∃ b<a. w ∈ (F(b) - (⋃ c<b. F(c))) & z ∈ (F(b) - (⋃ c<b. F(c)))"
 <proof>

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

lemma UN_lepoll_index:
 "[| ∀ i<a. F(i) - (⋃ j<i. F(j)) lepoll 1; Limit(a) |]
 ==> (⋃ x<a. F(x)) 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) ⊆ {X ∈ Pow(A). X≈n};
 A≈a; y<a; ~Finite(a); Card(a); n ∈ nat |]
 ==> Union(recfunAC16(f,g,y,a))<a"
 <proof>

lemma dbl_Diff_eqpoll:
 "[| recfunAC16(f, h, y, a) ⊆ {X ∈ Pow(A) . X≈succ(k #+ m)};
 Card(a); ~Finite(a); A≈a;
 k ∈ nat; y<a;
 h ∈ bij(a, {Y ∈ Pow(A). Y≈succ(k)}) |]
 ==> A - Union(recfunAC16(f, h, y, a)) - h'y≈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 ∈ Pow(A - B - h'i); x≈m;
 h ∈ bij(a, {x ∈ Pow(A) . x≈k}); i<a; k ∈ nat; m ∈ nat |]
 ==> h ' i Un x ∈ {x ∈ Pow(A) . x≈k #+ m}"
 <proof>

lemma *lemma6*:
 "[| ∀y<succ(j). F(y)<=X & (∀x<a. x<y | P(x,y) --> Q(x,y)); succ(j)<a
 |]
 ==> F(j)<=X & (∀x<a. x<j | P(x,j) --> Q(x,j))"
 <proof>

lemma *lemma7*:
 "[| ∀x<a. x<j | P(x,j) --> Q(x,j); succ(j)<a |]
 ==> P(j,j) --> (∀x<a. x≤j | P(x,j) --> Q(x,j))"
 <proof>

lemma *ex_subset_eqpoll*:
 "[| A≈a; ~ Finite(a); Ord(a); m ∈ nat |] ==> ∃X ∈ Pow(A). X≈m"
 <proof>

lemma *subset_Un_disjoint*: "[| A ⊆ B Un C; A Int C = 0 |] ==> A ⊆ B"
 <proof>

lemma *Int_empty*:
 "[| X ∈ Pow(A - Union(B) -C); T ∈ B; F ⊆ T |] ==> F Int X = 0"
 <proof>

lemma *subset_imp_eq_lemma*:

"m ∈ nat ==> ∀ A B. A ⊆ B & m lepoll A & B lepoll m --> A=B"
 <proof>

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

lemma bij_imp_arg_eq:
 "[| f ∈ bij(a, {Y ∈ X. Y ≈ succ(k)}); k ∈ nat; f' b ⊆ f' y; b < a; y < a
 |]
 ==> b=y"
 <proof>

lemma ex_next_set:
 "[| recfunAC16(f, h, y, a) ⊆ {X ∈ Pow(A) . X ≈ succ(k #+ m)};
 Card(a); ~ Finite(a); A ≈ a;
 k ∈ nat; m ∈ nat; y < a;
 h ∈ bij(a, {Y ∈ Pow(A). Y ≈ succ(k)});
 ~ (∃ Y ∈ recfunAC16(f, h, y, a). h' y ⊆ Y) |]
 ==> ∃ X ∈ {Y ∈ Pow(A). Y ≈ succ(k #+ m)}. h' y ⊆ X &
 (∀ b < a. h' b ⊆ X -->
 (∀ T ∈ recfunAC16(f, h, y, a). ~ h' b ⊆ T))"
 <proof>

lemma ex_next_Ord:
 "[| recfunAC16(f, h, y, a) ⊆ {X ∈ Pow(A) . X ≈ succ(k #+ m)};
 Card(a); ~ Finite(a); A ≈ a;
 k ∈ nat; m ∈ nat; y < a;
 h ∈ bij(a, {Y ∈ Pow(A). Y ≈ succ(k)});
 f ∈ bij(a, {Y ∈ Pow(A). Y ≈ succ(k #+ m)});
 ~ (∃ Y ∈ recfunAC16(f, h, y, a). h' y ⊆ Y) |]
 ==> ∃ c < a. h' y ⊆ f' c &
 (∀ b < a. h' b ⊆ f' c -->
 (∀ T ∈ recfunAC16(f, h, y, a). ~ h' b ⊆ T))"
 <proof>

```

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

```

```

lemma main_induct:
  "[|  $b < a; f \in \text{bij}(a, \{Y \in \text{Pow}(A) . Y \approx \text{succ}(k \ \# \ m)\});$ 
     $h \in \text{bij}(a, \{Y \in \text{Pow}(A) . Y \approx \text{succ}(k)\});$ 
     $\sim \text{Finite}(a); \text{Card}(a); A \approx a; k \in \text{nat}; m \in \text{nat} \mid]$ 
  ==>  $\text{recfunAC16}(f, h, b, a) \subseteq \{X \in \text{Pow}(A) . X \approx \text{succ}(k \ \# \ m)\} \ \&$ 
     $(\forall x < a. x < b \mid (\exists Y \in \text{recfunAC16}(f, h, b, a). h \restriction x \subseteq Y) \rightarrow$ 
     $(\exists! Y. Y \in \text{recfunAC16}(f, h, b, a) \ \& \ h \restriction x \subseteq Y))$ "
<proof>

```

```

lemma lemma_simp_induct:
  "[|  $\forall b. b < a \rightarrow F(b) \subseteq S \ \& \ (\forall x < a. (x < b \mid (\exists Y \in F(b). f \restriction x \subseteq Y))$ 
     $\rightarrow (\exists! Y. Y \in F(b) \ \& \ f \restriction x \subseteq Y));$ 
     $f \in a \rightarrow f''(a); \text{Limit}(a);$ 
     $\forall i \ j. i \leq j \rightarrow F(i) \subseteq F(j) \mid]$ 
  ==>  $(\bigcup j < a. F(j)) \subseteq S \ \&$ 
     $(\forall x \in f''a. \exists! Y. Y \in (\bigcup j < a. F(j)) \ \& \ x \subseteq Y)$ "
<proof>

```

```

theorem W02_AC16: "[| W02;  $0 < m; k \in \text{nat}; m \in \text{nat} \mid]$  ==>  $\text{AC16}(k \ \# \ m, k)$ "
<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 ⊆ B; ~ A ⊆ C |] ==> ~ B ⊆ C"
<proof>

```

```

lemmas lepoll_paired = paired_eqpoll [THEN eqpoll_sym, THEN eqpoll_imp_lepoll]

```

```

lemma lemma2: "∃ y R. well_ord(y,R) & x Int y = 0 & ~y ⊆ z & ~Finite(y)"
<proof>

```

```

lemma infinite_Un: "~Finite(B) ==> ~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: "[|  $\sim A \approx B$ ;  $A \lesssim B$  |] ==>  $\text{succ}(A) \lesssim B$ "
  <proof>

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

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

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

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

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

lemma eq_imp_Int_eq: " $A = B$  ==>  $A \text{ Int } C = B \text{ Int } C$ "
  <proof>

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

  ==> A-B  $\approx$  m"
<proof>

```

```

lemma subsets_lepoll_0_eq_unit: "{x  $\in$  Pow(X). x  $\lesssim$  0} = {0}"
<proof>

```

```

lemma subsets_lepoll_succ:
  "n  $\in$  nat ==> {z  $\in$  Pow(y). z  $\lesssim$  succ(n)} =
    {z  $\in$  Pow(y). z  $\lesssim$  n} Un {z  $\in$  Pow(y). z  $\approx$  succ(n)}"
<proof>

```

```

lemma Int_empty:
  "n  $\in$  nat ==> {z  $\in$  Pow(y). z  $\lesssim$  n} Int {z  $\in$  Pow(y). z  $\approx$  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(1)"
    and MM_def:      "MM == {v  $\in$  t_n. succ(k)  $\lesssim$  v Int y}"
    and LL_def:      "LL == {v 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  $\ \& \ k \lesssim v$  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 Int y = 0"
and "includes": "t_n  $\subseteq$  {v  $\in$  Pow(x Un y). v  $\approx$  succ(k #+ m)}"
and WO_R[iff]: "well_ord(y,R)"
and lnat[iff]: "1  $\in$  nat"
and mnat[iff]: "m  $\in$  nat"
and mpos[iff]: "0 < m"
and Infinite[iff]: "~ Finite(y)"
and noLepoll: "~ y  $\lesssim$  {v  $\in$  Pow(x). v  $\approx$  m}"

lemma (in AC16) knat [iff]: "k  $\in$  nat"
<proof>

lemma (in AC16) Diff_Finite_eqpoll: "[| 1  $\approx$  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; 1  $\approx$  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:
  "[| 1  $\approx$  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:
  "[|  $\forall v \in s(u). \text{succ}(1) \approx v$  Int y;
    1  $\approx$  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:

```

```

    "[|  $\forall v \in s(u). \text{succ}(l) \approx v \text{ Int } y;$   

        $l \approx a; a \subseteq y; u \in x$  |]"  

    ==>  $y \lesssim \{v \in s(u). a \subseteq v\}$ "
  <proof>

```

```

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

```

```

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

```

```

lemma (in AC16) w_Int_eqpoll_m:
  "[|  $w \in \{v \in s(u). a \subseteq v\};$   

      $l \approx a; u \in x;$   

      $\forall v \in s(u). \text{succ}(l) \approx v \text{ Int } y$  |]"  

  ==>  $w \text{ Int } (x - \{u\}) \approx m$ "
  <proof>

```

```

lemma (in AC16) eqpoll_m_not_empty: " $a \approx m \implies a \neq 0$ "
  <proof>

```

```

lemma (in AC16) cons_cons_in:
  "[|  $z \in xa \text{ Int } (x - \{u\}); l \approx a; a \subseteq y; u \in x$  |]"  

  ==>  $\exists ! w. w \in t_n \ \& \ \text{cons}(z, \text{cons}(u, a)) \subseteq w$ "
  <proof>

```

```

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

  ==>  $\{v \in s(u). a \subseteq v\} \lesssim \{v \in \text{Pow}(x). v \approx m\}$ "
  <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 \lesssim ordertype(y, R)"
<proof>

lemma (in AC16) ex_subset_eqpoll_n: "n \in nat $\implies \exists z. z \subseteq y \ \& \ n \approx z$ "
<proof>

lemma (in AC16) exists_proper_in_s: "u \in x $\implies \exists v \in s(u). \text{succ}(k) \lesssim v$ Int y"
<proof>

lemma (in AC16) exists_in_MM: "u \in x $\implies \exists w \in MM. u \in w$ "
<proof>

lemma (in AC16) exists_in_LL: "u \in x $\implies \exists w \in LL. u \in GG'w$ "
<proof>

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

lemma (in AC16) in_MM_eqpoll_n: "w \in MM $\implies w \approx \text{succ}(k \ \# \ m)$ "
<proof>

lemma (in AC16) in_LL_eqpoll_n: "w \in LL $\implies \text{succ}(k) \lesssim w$ "
<proof>

lemma (in AC16) in_LL: "w \in LL $\implies w \subseteq (THE x. x \in MM \ \& \ w \subseteq x)$ "
<proof>

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

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

term AC16

theorem AC16_W04:

"[| AC_Equiv.AC16(k #+ m, k); 0 < k; 0 < m; k ∈ nat; m ∈ nat |]
 ==> W04(m) "
 ⟨proof⟩

end

theory AC17_AC1

imports HH

begin

lemma AC0_AC1_lemma: "[| f: (Π X ∈ A. X); D ⊆ A |] ==> ∃ g. g: (Π X ∈
 D. X) "
 ⟨proof⟩

lemma AC0_AC1: "AC0 ==> AC1 "
 ⟨proof⟩

lemma AC1_AC0: "AC1 ==> AC0 "
 ⟨proof⟩

lemma AC1_AC17_lemma: "f ∈ (Π X ∈ Pow(A) - {0}. X) ==> f ∈ (Pow(A)
 - {0} -> A) "
 ⟨proof⟩

lemma AC1_AC17: "AC1 ==> AC17 "
 ⟨proof⟩

lemma UN_eq_imp_well_ord:

"[| x - (⋃ j ∈ LEAST i. HH(λX ∈ Pow(x)-{0}. {f'X}, x, i) = {x}).

```

HH( $\lambda X \in \text{Pow}(x) - \{0\}. \{f'X\}, x, j)) = 0;$ 
 $f \in \text{Pow}(x) - \{0\} \rightarrow x \mid]$ 
 $\Rightarrow \exists r. \text{well\_ord}(x, r)"$ 
<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:
  "[ $\mid \forall f \in \text{Pow}(x) - \{0\} \rightarrow x. \exists u \in \text{Pow}(x) - \{0\}. f'u \notin u;$ 
     $\exists f \in \text{Pow}(x) - \{0\} \rightarrow x.$ 
     $x - (\bigcup a \in (\text{LEAST } i. \text{HH}(\lambda X \in \text{Pow}(x) - \{0\}. \{f'X\}, x, i) = \{x\})).$ 

     $\text{HH}(\lambda X \in \text{Pow}(x) - \{0\}. \{f'X\}, x, a)) = 0 \mid]$ 
     $\Rightarrow P"$ 
  <proof>

lemma AC17_AC1_aux2:
  "~ ( $\exists f \in \text{Pow}(x) - \{0\} \rightarrow x. x - F(f) = 0$ )
  ==> ( $\lambda f \in \text{Pow}(x) - \{0\} \rightarrow x. x - F(f)$ )
     $\in (\text{Pow}(x) - \{0\} \rightarrow x) \rightarrow \text{Pow}(x) - \{0\}"$ 
  <proof>

lemma AC17_AC1_aux3:
  "[ $\mid f'Z \in Z; Z \in \text{Pow}(x) - \{0\} \mid]$ 
  ==> ( $\lambda X \in \text{Pow}(x) - \{0\}. \{f'X\}'Z \in \text{Pow}(Z) - \{0\}"$ 
  <proof>

lemma AC17_AC1_aux4:
  " $\exists f \in F. f'((\lambda f \in F. Q(f))'f) \in (\lambda f \in F. Q(f))'f$ 
  ==>  $\exists f \in F. f'Q(f) \in Q(f)"$ 
  <proof>

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

```

lemma AC1_AC2_aux1:
 "[/ f:($\prod X \in A. X$); $B \in A$; $0 \notin A$ /] ==> {f'B} \subseteq B Int {f'C. C
 $\in A$ }"
 <proof>

lemma AC1_AC2_aux2:
 "[/ pairwise_disjoint(A); $B \in A$; $C \in A$; $D \in B$; $D \in C$ /] ==>
 f'B = f'C"
 <proof>

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

lemma AC2_AC1_aux1: " $0 \notin A$ ==> $0 \notin \{B*\{B\}. B \in A\}$ "
 <proof>

lemma AC2_AC1_aux2: "[/ $X*\{X\}$ Int C = {y}; $X \in A$ /]
 ==> (THE y. $X*\{X\}$ Int C = {y}): $X*A$ "
 <proof>

lemma AC2_AC1_aux3:
 " $\forall D \in \{E*\{E\}. E \in A\}. \exists y. D$ Int C = {y}"
 ==> ($\lambda x \in A. \text{fst}(\text{THE } z. (x*\{x\} \text{ Int } C = \{z\}))$) $\in (\prod X \in A. X)$ "
 <proof>

lemma AC2_AC1: "AC2 ==> AC1"
 <proof>

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

lemma AC1_AC4: "AC1 ==> AC4"
 <proof>

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

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

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

lemma AC4_AC5: " $AC4 \implies AC5$ "
 $\langle \text{proof} \rangle$

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

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

lemma AC5_AC4_aux3: " $[| \exists f \in A \rightarrow C. P(f, \text{domain}(f)); A=B |] \implies \exists f \in B \rightarrow C. P(f, B)$ "
 $\langle \text{proof} \rangle$

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

```
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 ≲ X"
  <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) ==  $\forall X R. R \subseteq \text{Pow}(X) * X \ \&$ 
 $(\forall Y \in \text{Pow}(X). Y \prec a \rightarrow (\exists x \in X. \langle Y, x \rangle \in R))$ 
 $\rightarrow (\exists f \in a \rightarrow X. \forall b \prec a. \langle f' b, f' b \rangle \in R)"$ 

```

definition

```

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

```

definition

```

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

```

locale DC0_imp =

fixes XX and RR and X and R

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

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

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

```

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

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

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

theorem DCO_imp_DC_nat: "DC0 ==> DC(nat)"
<proof>

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

<proof>

```

locale imp_DC0 =
  fixes XX and RR and x and R and f and allRR
  defines XX_def: "XX ==  $(\bigcup n \in \text{nat}.$ 
```

```

    {f ∈ succ(n) → domain(R). ∀ k ∈ n. <f'k, f'succ(k)>
∈ R})"
  and RR_def:
    "RR == {<z1,z2>:Fin(XX)*XX.
      (domain(z2)=succ(⋃ f ∈ z1. domain(f))
      & (∀ f ∈ z1. restrict(z2, domain(f)) = f))
      | (¬ (∃ g ∈ XX. domain(g)=succ(⋃ f ∈ z1. domain(f))
      & (∀ f ∈ z1. restrict(g, domain(f)) = f)) & z2={<0,x>})}"
  and allRR_def:
    "allRR == ∀ b<nat.
      <f''b, f'b> ∈
      {<z1,z2>∈Fin(XX)*XX. (domain(z2)=succ(⋃ f ∈ z1. domain(f))
      & (⋃ f ∈ z1. domain(f)) = b
      & (∀ f ∈ z1. restrict(z2,domain(f))
= f))}"

lemma (in imp_DC0) lemma4:
  "[| range(R) ⊆ domain(R); x ∈ domain(R) |]
  ==> RR ⊆ Pow(XX)*XX &
  (∀ Y ∈ Pow(XX). Y ≺ nat --> (∃ x ∈ XX. <Y,x>:RR))"
  <proof>

lemma (in imp_DC0) UN_image_succ_eq:
  "[| f ∈ nat → X; n ∈ nat |]
  ==> (⋃ x ∈ f''succ(n). P(x)) = P(f'n) Un (⋃ x ∈ f''n. P(x))"
  <proof>

lemma (in imp_DC0) UN_image_succ_eq_succ:
  "[| (⋃ x ∈ f''n. P(x)) = y; P(f'n) = succ(y);
  f ∈ nat → X; n ∈ nat |] ==> (⋃ x ∈ f''succ(n). P(x)) = succ(y)"
  <proof>

lemma (in imp_DC0) apply_domain_type:
  "[| h ∈ succ(n) → D; n ∈ nat; domain(h)=succ(y) |] ==> h'y ∈ D"
  <proof>

lemma (in imp_DC0) image_fun_succ:
  "[| h ∈ nat → X; n ∈ nat |] ==> h''succ(n) = cons(h'n, h''n)"
  <proof>

lemma (in imp_DC0) f_n_type:
  "[| domain(f'n) = succ(k); f ∈ nat → XX; n ∈ nat |]
  ==> f'n ∈ succ(k) → domain(R)"
  <proof>

lemma (in imp_DC0) f_n_pairs_in_R [rule_format]:
  "[| h ∈ nat → XX; domain(h'n) = succ(k); n ∈ nat |]
  ==> ∀ i ∈ k. <h'n'i, h'n'succ(i)> ∈ R"

```

<proof>

```
lemma (in imp_DC0) restrict_cons_eq_restrict:
  "[| restrict(h, domain(u))=u; h ∈ n->X; domain(u) ⊆ n |]
  ==> restrict(cons(<n, y>, h), domain(u)) = u"
<proof>
```

```
lemma (in imp_DC0) all_in_image_restrict_eq:
  "[| ∀ x ∈ f'`n. restrict(f'n, domain(x))=x;
    f ∈ nat -> XX;
    n ∈ nat; domain(f'n) = succ(n);
    (⋃ x ∈ f'`n. domain(x)) ⊆ n |]
  ==> ∀ x ∈ f'`succ(n). restrict(cons(<succ(n),y>, f'n), domain(x))
  = x"
<proof>
```

```
lemma (in imp_DC0) simplify_recursion:
  "[| ∀ b<nat. <f'`b, f'b> ∈ RR;
    f ∈ nat -> XX; range(R) ⊆ domain(R); x ∈ domain(R) |]
  ==> allRR"
<proof>
```

```
lemma (in imp_DC0) lemma2:
  "[| allRR; f ∈ nat->XX; range(R) ⊆ domain(R); x ∈ domain(R); n
  ∈ nat |]
  ==> f'n ∈ succ(n) -> domain(R) & (∀ i ∈ n. <f'n'i, f'n'succ(i)>: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"
 <proof>

lemma W01_DC_lemma:
 "[| Card(K); well_ord(X, Q);
 ∀ Y ∈ Pow(X). Y < K --> (∃ x ∈ X. <Y, x> ∈ R); b ∈ K |]
 ==> ff(b, X, Q, R) ∈ {x ∈ X. <(λc ∈ b. ff(c, X, Q, R))'`b, x>
 ∈ R}"
 <proof>

theorem W01_DC_Card: "W01 ==> ∀ K. Card(K) --> DC(K)"
 <proof>

end

References

- [1] Lawrence C. Paulson and Krzysztof Grąbczewski. 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.