

The Cardinality of the Continuum

Manuel Eberl

March 17, 2025

Abstract

This entry presents a short derivation of the cardinality of \mathbb{R} , namely that $|\mathbb{R}| = |2^{\mathbb{N}}| = 2^{\aleph_0}$. This is done by showing the injection $\mathbb{R} \rightarrow 2^{\mathbb{Q}}$, $x \mapsto (-\infty, x) \cap \mathbb{Q}$ (i.e. Dedekind cuts) for one direction and the injection $2^{\mathbb{N}} \rightarrow \mathbb{Q}$, $X \mapsto \sum_{n \in X} 3^{-n}$, i.e. ternary fractions, for the other direction.

Contents

1	Auxiliary material	2
1.1	Miscellaneous facts about cardinalities	2
1.2	The set of finite subsets	4
1.3	The set of functions with finite support	4
2	The Cardinality of the Continuum	5
2.1	$ \mathbb{R} \leq 2^{\mathbb{Q}} $ via Dedekind cuts	5
2.2	$2^{ \mathbb{N} } \leq \mathbb{R} $ via ternary fractions	6
2.3	Equipollence proof	6
2.4	Corollaries for real intervals	7
2.5	Corollaries for vector spaces	7

1 Auxiliary material

```
theory Cardinality_Continuum_Library
  imports "HOL-Library.Equipollence" "HOL-Cardinals.Cardinals"
begin
```

1.1 Miscellaneous facts about cardinalities

```
lemma eqpoll_Pow [intro]:
  assumes "A ≈ B"
  shows   "Pow A ≈ Pow B"
⟨proof⟩
```

```
lemma lepoll_UNIV_nat_iff: "A ≲ (UNIV :: nat set) ⟷ countable A"
⟨proof⟩
```

```
lemma countable_eqpoll:
  assumes "countable A" "A ≈ B"
  shows   "countable B"
⟨proof⟩
```

```
lemma countable_eqpoll_cong: "A ≈ B ⟹ countable A ⟷ countable B"
⟨proof⟩
```

```
lemma eqpoll_UNIV_nat_iff: "A ≈ (UNIV :: nat set) ⟷ countable A ∧
infinite A"
⟨proof⟩
```

```
lemma ordLeq_finite_infinite:
  "finite A ⟹ infinite B ⟹ (card_of A, card_of B) ∈ ordLeq"
⟨proof⟩
```

```
lemma eqpoll_imp_card_of_ordIso: "A ≈ B ⟹ |A| =o |B|"
⟨proof⟩
```

```
lemma card_of_Func: "|Func A B| =o |B| ^c |A|"
⟨proof⟩
```

```
lemma card_of_leq_natLeq_iff_countable:
  "|X| ≤o natLeq ⟷ countable X"
⟨proof⟩
```

```
lemma card_of_Sigma_cong:
  assumes "∧x. x ∈ A ⟹ |B x| =o |B' x|"
  shows   "|SIGMA x:A. B x| =o |SIGMA x:A. B' x|"
⟨proof⟩
```

```
lemma Cfinite_cases:
```

```

    assumes "Cfinite c"
    obtains n :: nat where "(c, natLeq_on n) ∈ ordIso"
  ⟨proof⟩

lemma empty_nat_ordIso_czero: "{ } :: (nat × nat) set) =o czero"
  ⟨proof⟩

lemma card_order_on_empty: "card_order_on { } { }"
  ⟨proof⟩

lemma natLeq_on_plus_ordIso: "natLeq_on (m + n) =o natLeq_on m +c natLeq_on n"
  ⟨proof⟩

lemma natLeq_on_1_ord_iso: "natLeq_on 1 =o BNF_Cardinal_Arithmetic.cone"
  ⟨proof⟩

lemma cexp_infinite_finite_ordLeq:
  assumes "Cinfinite c" "Cfinite c'"
  shows "c  $\hat{~}$  c' ≤o c"
  ⟨proof⟩

lemma cexp_infinite_finite_ordIso:
  assumes "Cinfinite c" "Cfinite c'" "BNF_Cardinal_Arithmetic.cone ≤o c'"
  shows "c  $\hat{~}$  c' =o c"
  ⟨proof⟩

lemma Cfinite_ordLeq_Cinfinite:
  assumes "Cfinite c" "Cinfinite c'"
  shows "c ≤o c'"
  ⟨proof⟩

lemma cfinite_card_of_iff [simp]: "BNF_Cardinal_Arithmetic.cfinite (card_of X)  $\longleftrightarrow$  finite X"
  ⟨proof⟩

lemma cinfinite_card_of_iff [simp]: "BNF_Cardinal_Arithmetic.cinfinite (card_of X)  $\longleftrightarrow$  infinite X"
  ⟨proof⟩

lemma Func_conv_PiE: "Func A B = PiE A (λ_. B)"
  ⟨proof⟩

lemma finite_Func [intro]:
  assumes "finite A" "finite B"
  shows "finite (Func A B)"
  ⟨proof⟩

```

lemma *ordLeq_antisym*: " $(c, c') \in \text{ordLeq} \implies (c', c) \in \text{ordLeq} \implies (c, c') \in \text{ordIso}$ "
 ⟨*proof*⟩

lemma *cmax_cong*:
 assumes " $(c1, c1') \in \text{ordIso}$ " " $(c2, c2') \in \text{ordIso}$ " "*Card_order* *c1*" "*Card_order* *c2*"
 shows "*cmax* *c1* *c2* =o *cmax* *c1'* *c2'*"
 ⟨*proof*⟩

1.2 The set of finite subsets

We define an operator $\text{FinPow}(X)$ that, given a set X , returns the set of all finite subsets of that set. For finite X , this is boring since it is obviously just the power set. For infinite X , it is however a useful concept to have.

We will show that if X is infinite then the cardinality of $\text{FinPow}(X)$ is exactly the same as that of X .

definition *FinPow* :: "'a set \Rightarrow 'a set set" **where**
 "*FinPow* *X* = {*Y*. $Y \subseteq X \wedge \text{finite } Y$ }"

lemma *finite_FinPow* [*intro*]: "*finite* *A* $\implies \text{finite } (\text{FinPow } A)$ "
 ⟨*proof*⟩

lemma *in_FinPow_iff*: " $Y \in \text{FinPow } X \iff Y \subseteq X \wedge \text{finite } Y$ "
 ⟨*proof*⟩

lemma *FinPow_subseteq_Pow*: "*FinPow* *X* $\subseteq \text{Pow } X$ "
 ⟨*proof*⟩

lemma *FinPow_eq_Pow*: "*finite* *X* $\implies \text{FinPow } X = \text{Pow } X$ "
 ⟨*proof*⟩

theorem *card_of_FinPow_infinite*:
 assumes "*infinite* *A*"
 shows " $|\text{FinPow } A| =o |A|$ "
 ⟨*proof*⟩

1.3 The set of functions with finite support

Next, we define an operator $\text{Func_finsupp}_z(A, B)$ that, given sets A and B and an element $z \in B$, returns the set of functions $f : A \rightarrow B$ that have *finite support*, i.e. that map all but a finite subset of A to z .

definition *Func_finsupp* :: "'b \Rightarrow 'a set \Rightarrow 'b set \Rightarrow ('a \Rightarrow 'b) set" **where**
 "*Func_finsupp* *z* *A* *B* = { $f \in A \rightarrow B. (\forall x. x \notin A \longrightarrow f\ x = z) \wedge \text{finite } \{x. f\ x \neq z\}$ }"

lemma *bij_betw_Func_finsup_Func_finite*:

```

assumes "finite A"
shows "bij_betw ( $\lambda f. \text{restrict } f \ A$ ) (Func_finsupp z A B) (Func A B)"
<proof>

```

```

lemma eqpoll_Func_finsupp_Func_finite: "finite A  $\implies$  Func_finsupp z A B
 $\approx$  Func A B"
<proof>

```

```

lemma card_of_Func_finsupp_finite: "finite A  $\implies$  |Func_finsupp z A B|
 $=_o$  |B|  $\wedge_c$  |A|"
<proof>

```

The cases where A and B are both finite or $B = \{0\}$ or $A = \emptyset$ are of course trivial.

Perhaps not completely obviously, it turns out that in all other cases, the cardinality of $\text{Func_finsupp}_z(A, B)$ is exactly $\max(|A|, |B|)$.

```

theorem card_of_Func_finsupp_infinite:
  assumes "z  $\in$  B" and "B - {z}  $\neq$  {}" and "A  $\neq$  {}"
  assumes "infinite A  $\vee$  infinite B"
  shows "|Func_finsupp z A B|  $=_o$  cmax |A| |B|"
<proof>

```

end

2 The Cardinality of the Continuum

```

theory Cardinality_Continuum
  imports Complex_Main Cardinality_Continuum_Library
begin

```

2.1 $|\mathbb{R}| \leq |2^{\mathbb{Q}}|$ via Dedekind cuts

```

lemma le_cSup_iff:
  fixes A :: "'a :: conditionally_complete_linorder set"
  assumes "A  $\neq$  {}" "bdd_above A"
  shows "Sup A  $\geq$  c  $\longleftrightarrow$  ( $\forall x < c. \exists y \in A. y > x$ )"
<proof>

```

We show that the function mapping a real number to all the rational numbers below it is an injective map from the reals to $2^{\mathbb{Q}}$. This is the same idea that is used in the Dedekind cut definition of the reals.

```

lemma inj_Dedekind_cut:
  fixes f :: "real  $\Rightarrow$  rat set"
  defines "f  $\equiv$  ( $\lambda x :: \text{real}. \{r :: \text{rat}. \text{of\_rat } r < x\}$ )"
  shows "inj f"
<proof>

```

2.2 $2^{\mathbb{N}} \leq |\mathbb{R}|$ via ternary fractions

For the other direction, we construct an injective function that maps a set of natural numbers A to a real number by constructing a ternary decimal number of the form $d_0.d_1d_2d_3\dots$ where d_m is 1 if $m \in A$ and 0 otherwise.

We will first show a few more general results about such n -ary fraction expansions.

```
lemma geometric_sums':
  fixes c :: "'a :: real_normed_field"
  assumes "norm c < 1"
  shows "(λn. c ^ (n + m)) sums (c ^ m / (1 - c))"
⟨proof⟩
```

```
lemma summable_nary_fraction:
  fixes d :: real and f :: "nat ⇒ real"
  assumes "∧n. norm (f n) ≤ c" "d > 1"
  shows "summable (λn. f n / d ^ n)"
⟨proof⟩
```

Consider two n -ary fraction expansions $u = u_1.u_2u_3\dots$ and $v = v_1.v_2v_3\dots$ with $n \geq 2$. Suppose that all the u_i and v_i are between 0 and $n - 2$ (i.e. the highest digit does not occur). Then u and v are equal if and only if all $u_i = v_i$ for all i .

Note that without the additional restriction the result does not hold, as e.g. the decimal numbers 0.2 and $0.1\bar{9}$ are equal.

The reasoning boils down to showing that if m is the smallest index where the two sequences differ, then $|u - v| \geq \frac{1}{d-1} > 0$.

```
lemma nary_fraction_unique:
  fixes u v :: "nat ⇒ nat"
  assumes f_eq: "(∑n. real (u n) / real d ^ n) = (∑n. real (v n) /
real d ^ n)"
  assumes uv: "∧n. u n ≤ d - 2" "∧n. v n ≤ d - 2" and d: "d ≥ 2"
  shows "u = v"
⟨proof⟩
```

It now follows straightforwardly that mapping sets of natural numbers to ternary fraction expansions is indeed injective. For binary fractions, this would not work due to the aforementioned issue.

```
lemma inj_nat_set_to_ternary:
  fixes f :: "nat set ⇒ real"
  defines "f ≡ (λA. ∑n. (if n ∈ A then 1 else 0) / 3 ^ n)"
  shows "inj f"
⟨proof⟩
```

2.3 Equipollence proof

```
theorem eqpoll_UNIV_real: "(UNIV :: real set) ≈ (UNIV :: nat set set)"
```

<proof>

We can also write the language in the language of cardinal numbers as $|\mathbb{R}| = 2^{\aleph_0}$ using Isabelle's cardinal number library:

```
corollary card_of_UNIV_real: "|UNIV :: real set| =o ctwo ^c natLeq"  
<proof>
```

2.4 Corollaries for real intervals

It is easy to show that any real interval (whether open, closed, or infinite) is equipollent to the full set of real numbers.

```
lemma eqpoll_Ioo_real:  
  fixes a b :: real  
  assumes "a < b"  
  shows   "{a<..} ≈ (UNIV :: real set)"  
<proof>
```

```
lemma eqpoll_real:  
  assumes "{a::real<..} ⊆ X" "a < b"  
  shows   "X ≈ (UNIV :: real set)"  
<proof>
```

```
lemma eqpoll_Icc_real: "(a::real) < b ⇒ {a..b} ≈ (UNIV :: real set)"  
  and eqpoll_Ioc_real: "(a::real) < b ⇒ {a<..b} ≈ (UNIV :: real set)"  
  and eqpoll_Ico_real: "(a::real) < b ⇒ {a..} ≈ (UNIV :: real set)"  
<proof>
```

```
lemma eqpoll_Ici_real: "{a::real..}" ≈ (UNIV :: real set)"  
  and eqpoll_Ioi_real: "{a::real<..}" ≈ (UNIV :: real set)"  
<proof>
```

```
lemma eqpoll_Iic_real: "{..a::real}" ≈ (UNIV :: real set)"  
  and eqpoll_Iio_real: "{..a::real}" ≈ (UNIV :: real set)"  
<proof>
```

```
lemmas eqpoll_real_ivl =  
  eqpoll_Ioo_real eqpoll_Ioc_real eqpoll_Ico_real eqpoll_Icc_real  
  eqpoll_Iio_real eqpoll_Iic_real eqpoll_Ici_real eqpoll_Ioi_real
```

```
lemmas card_of_ivl_real =  
  eqpoll_real_ivl[THEN eqpoll_imp_card_of_ordIso, THEN ordIso_transitive[OF  
  _ card_of_UNIV_real]]
```

2.5 Corollaries for vector spaces

We will now also show some results about the cardinality of vector spaces. To do this, we use the obvious isomorphism between a vector space V with a basis B and the set of finite-support functions $B \rightarrow V$.

```

lemma (in vector_space) card_of_span:
  assumes "independent B"
  shows "|span B| =o |Func_finsupp 0 B (UNIV :: 'a set)|"
  <proof>

```

We can now easily show the following: Let K be an infinite field and V a non-trivial finite-dimensional K -vector space. Then $|V| = |K|$.

```

lemma (in vector_space) card_of_span_finite_dim_infinite_field:
  assumes "independent B" and "finite B" and "B ≠ {}" and "infinite
  (UNIV :: 'a set)"
  shows "|span B| =o |UNIV :: 'a set|"
  <proof>

```

Similarly, we can show the following: Let V be an infinite-dimensional vector space V over some (not necessarily infinite) field K . Then $|V| = \max(\dim_K(V), |K|)$.

```

lemma (in vector_space) card_of_span_infinite_dim_infinite_field:
  assumes "independent B" "infinite B"
  shows "|span B| =o cmax |B| |UNIV :: 'a set|"
  <proof>

```

end

```

theory Cardinality_Euclidean_Space
  imports "HOL-Analysis.Analysis" Cardinality_Continuum
begin

```

With these results, it is now easy to see that any Euclidean space (i.e. finite-dimensional real vector space) has the same cardinality as \mathbb{R} :

```

corollary card_of_UNIV_euclidean_space:
  "|UNIV :: 'a :: euclidean_space set| =o ctwo ^c natLeq"
  <proof>

```

In particular, this applies to \mathbb{C} and \mathbb{R}^n :

```

corollary card_of_complex: "|UNIV :: complex set| =o ctwo ^c natLeq"
  <proof>

```

```

corollary card_of_real_vec: "|UNIV :: (real ^ 'n :: finite) set| =o ctwo
^c natLeq"
  <proof>

```

end