

Banach-Steinhaus theorem*

Dominique Unruh José Manuel Rodríguez Caballero

February 6, 2026

Abstract

We formalize in Isabelle/HOL a result [2] due to S. Banach and H. Steinhaus [1] known as Banach-Steinhaus theorem or Uniform boundedness principle: a pointwise-bounded family of continuous linear operators from a Banach space to a normed space is uniformly bounded. Our approach is an adaptation to Isabelle/HOL of a proof due to A. Sokal [3].

Contents

1	Missing results for the proof of Banach-Steinhaus theorem	1
1.1	Results missing for the proof of Banach-Steinhaus theorem . . .	2
2	Banach-Steinhaus theorem	20
2.1	Preliminaries for Sokal’s proof of Banach-Steinhaus theorem . . .	21
2.2	Banach-Steinhaus theorem	25
2.3	A consequence of Banach-Steinhaus theorem	29

1 Missing results for the proof of Banach-Steinhaus theorem

```
theory Banach-Steinhaus-Missing
imports
  HOL-Analysis.Bounded-Linear-Function
  HOL-Analysis.Line-Segment
```

```
begin
```

*Supported by the ERC consolidator grant CerQuS (819317), the PRG team grant Secure Quantum Technology (PRG946) from the Estonian Research Council, and the Estonian Centre of Excellence in IT (EXCITE) funded by ERDF.

1.1 Results missing for the proof of Banach-Steinhaus theorem

The results proved here are preliminaries for the proof of Banach-Steinhaus theorem using Sokal's approach, but they do not explicitly appear in Sokal's paper [3].

Notation for the norm

```
open-bundle norm-syntax begin
notation norm (⟨||-||⟩)
end
```

Notation for apply bilinear function

```
open-bundle blinfun-apply-syntax begin
notation blinfun-apply (infixr ⟨*,v⟩ 70)
end
```

lemma *bdd-above-plus*:

```
fixes f::⟨'a ⇒ real⟩
assumes ⟨bdd-above (f ' S)⟩ and ⟨bdd-above (g ' S)⟩
shows ⟨bdd-above ((λ x. f x + g x) ' S)⟩
```

Explanation: If the images of two real-valued functions f, g are bounded above on a set S , then the image of their sum is bounded on S .

proof –

```
obtain M where ⟨∧ x. x ∈ S ⇒ f x ≤ M⟩
using ⟨bdd-above (f ' S)⟩ unfolding bdd-above-def by blast
obtain N where ⟨∧ x. x ∈ S ⇒ g x ≤ N⟩
using ⟨bdd-above (g ' S)⟩ unfolding bdd-above-def by blast
have ⟨∧ x. x ∈ S ⇒ f x + g x ≤ M + N⟩
using ⟨∧ x. x ∈ S ⇒ f x ≤ M⟩ ⟨∧ x. x ∈ S ⇒ g x ≤ N⟩ by fastforce
thus ?thesis unfolding bdd-above-def by blast
qed
```

The maximum of two functions

```
definition pointwise-max:: (⟨'a ⇒ 'b::ord⟩ ⇒ ⟨'a ⇒ 'b⟩ ⇒ ⟨'a ⇒ 'b⟩) where
⟨pointwise-max f g = (λ x. max (f x) (g x))⟩
```

lemma *max-Sup-absorb-left*:

```
fixes f g::⟨'a ⇒ real⟩
assumes ⟨X ≠ {}⟩ and ⟨bdd-above (f ' X)⟩ and ⟨bdd-above (g ' X)⟩ and ⟨Sup
(f ' X) ≥ Sup (g ' X)⟩
shows ⟨Sup ((pointwise-max f g) ' X) = Sup (f ' X)⟩
```

Explanation: For real-valued functions f and g , if the supremum of f is greater-equal the supremum of g , then the supremum of $\max f g$ equals the supremum of f . (Under some technical conditions.)

proof –

```
have y-Sup: ⟨y ∈ ((λ x. max (f x) (g x)) ' X) ⇒ y ≤ Sup (f ' X)⟩ for y
```

proof–

assume $\langle y \in ((\lambda x. \max (f x) (g x)) \text{ ' } X) \rangle$
then obtain x **where** $\langle y = \max (f x) (g x) \rangle$ **and** $\langle x \in X \rangle$
by *blast*
have $\langle f x \leq \text{Sup} (f \text{ ' } X) \rangle$
by (*simp add: $\langle x \in X \rangle \langle \text{bdd-above} (f \text{ ' } X) \rangle \text{cSUP-upper}$*)
moreover have $\langle g x \leq \text{Sup} (g \text{ ' } X) \rangle$
by (*simp add: $\langle x \in X \rangle \langle \text{bdd-above} (g \text{ ' } X) \rangle \text{cSUP-upper}$*)
ultimately have $\langle \max (f x) (g x) \leq \text{Sup} (f \text{ ' } X) \rangle$
using $\langle \text{Sup} (f \text{ ' } X) \geq \text{Sup} (g \text{ ' } X) \rangle$ **by** *auto*
thus *?thesis* **by** (*simp add: $\langle y = \max (f x) (g x) \rangle$*)

qed

have $y\text{-}f\text{-}X$: $\langle y \in f \text{ ' } X \implies y \leq \text{Sup} ((\lambda x. \max (f x) (g x)) \text{ ' } X) \rangle$ **for** y

proof–

assume $\langle y \in f \text{ ' } X \rangle$
then obtain x **where** $\langle x \in X \rangle$ **and** $\langle y = f x \rangle$
by *blast*
have $\langle \text{bdd-above} ((\lambda \xi. \max (f \xi) (g \xi)) \text{ ' } X) \rangle$
by (*metis (no-types) $\langle \text{bdd-above} (f \text{ ' } X) \rangle \langle \text{bdd-above} (g \text{ ' } X) \rangle \text{bdd-above-image-sup-sup-max}$*)
moreover have $\langle e > 0 \implies \exists k \in (\lambda \xi. \max (f \xi) (g \xi)) \text{ ' } X. y \leq k + e \rangle$
for $e::\text{real}$
using $\langle \text{Sup} (f \text{ ' } X) \geq \text{Sup} (g \text{ ' } X) \rangle$
by (*smt (verit, best) $\langle x \in X \rangle \langle y = f x \rangle \text{imageI}$*)
ultimately show *?thesis*
using $\langle x \in X \rangle \langle y = f x \rangle \text{cSUP-upper}$ **by** *fastforce*

qed

have $\langle \text{Sup} ((\lambda x. \max (f x) (g x)) \text{ ' } X) \leq \text{Sup} (f \text{ ' } X) \rangle$
using $y\text{-}Sup$ **by** (*simp add: $\langle X \neq \{\} \rangle \text{cSup-least}$*)
moreover have $\langle \text{Sup} ((\lambda x. \max (f x) (g x)) \text{ ' } X) \geq \text{Sup} (f \text{ ' } X) \rangle$
using $y\text{-}f\text{-}X$ **by** (*metis (mono-tags) $\text{cSup-least calculation empty-is-image}$*)
ultimately show *?thesis* **unfolding** *pointwise-max-def* **by** *simp*

qed

lemma *max-Sup-absorb-right*:

fixes $f g::\langle 'a \Rightarrow \text{real} \rangle$
assumes $\langle X \neq \{\} \rangle$ **and** $\langle \text{bdd-above} (f \text{ ' } X) \rangle$ **and** $\langle \text{bdd-above} (g \text{ ' } X) \rangle$ **and** $\langle \text{Sup} (f \text{ ' } X) \leq \text{Sup} (g \text{ ' } X) \rangle$
shows $\langle \text{Sup} ((\text{pointwise-max } f g) \text{ ' } X) = \text{Sup} (g \text{ ' } X) \rangle$

Explanation: For real-valued functions f and g and a nonempty set X , such that the f and g are bounded above on X , if the supremum of f on X is lower-equal the supremum of g on X , then the supremum of *pointwise-max* $f g$ on X equals the supremum of g . This is the right analog of *max-Sup-absorb-left*.

proof–

have $\langle \text{Sup} ((\text{pointwise-max } g f) \text{ ' } X) = \text{Sup} (g \text{ ' } X) \rangle$
using *assms* **by** (*simp add: $\text{max-Sup-absorb-left}$*)
moreover have $\langle \text{pointwise-max } g f = \text{pointwise-max } f g \rangle$

unfolding *pointwise-max-def* **by** *auto*
ultimately show *?thesis* **by** *simp*
qed

lemma *max-Sup*:

fixes $f\ g :: \langle 'a \Rightarrow \text{real} \rangle$
assumes $\langle X \neq \{\} \rangle$ **and** $\langle \text{bdd-above } (f \text{ ' } X) \rangle$ **and** $\langle \text{bdd-above } (g \text{ ' } X) \rangle$
shows $\langle \text{Sup } ((\text{pointwise-max } f\ g) \text{ ' } X) = \text{max } (\text{Sup } (f \text{ ' } X)) (\text{Sup } (g \text{ ' } X)) \rangle$

Explanation: Let X be a nonempty set. Two supremum over X of the maximum of two real-value functions is equal to the maximum of their suprema over X , provided that the functions are bounded above on X .

proof(*cases* $\langle \text{Sup } (f \text{ ' } X) \geq \text{Sup } (g \text{ ' } X) \rangle$)

case *True* **thus** *?thesis* **by** (*simp add: assms(1) assms(2) assms(3) max-Sup-absorb-left*)
next

case *False*

have $f1: \neg 0 \leq \text{Sup } (f \text{ ' } X) + - 1 * \text{Sup } (g \text{ ' } X)$

using *False* **by** *linarith*

hence $\text{Sup } (\text{Banach-Steinhaus-Missing.pointwise-max } f\ g \text{ ' } X) = \text{Sup } (g \text{ ' } X)$

by (*simp add: assms(1) assms(2) assms(3) max-Sup-absorb-right*)

thus *?thesis*

using $f1$ **by** *linarith*

qed

lemma *identity-telescopic*:

fixes $x :: \langle - \Rightarrow 'a :: \text{real-normed-vector} \rangle$

assumes $\langle x \longrightarrow l \rangle$

shows $\langle (\lambda N. \text{sum } (\lambda k. x (\text{Suc } k) - x k) \{n..N\}) \longrightarrow l - x n \rangle$

Expression of a limit as a telescopic series. Explanation: If x converges to l then the sum $\sum k = n..N. x (\text{Suc } k) - x k$ converges to $l - x n$ as N goes to infinity.

proof–

have $\langle (\lambda p. x (p + \text{Suc } n)) \longrightarrow l \rangle$

using $\langle x \longrightarrow l \rangle$ **by** (*rule LIMSEQ-ignore-initial-segment*)

hence $\langle (\lambda p. x (\text{Suc } n + p)) \longrightarrow l \rangle$

by (*simp add: add.commute*)

hence $\langle (\lambda p. x (\text{Suc } (n + p))) \longrightarrow l \rangle$

by *simp*

hence $\langle (\lambda t. (- (x n)) + (\lambda p. x (\text{Suc } (n + p))) t) \longrightarrow (- (x n)) + l \rangle$

using *tendsto-add-const-iff* **by** *metis*

hence $f1: \langle (\lambda p. x (\text{Suc } (n + p)) - x n) \longrightarrow l - x n \rangle$

by *simp*

have $\langle \text{sum } (\lambda k. x (\text{Suc } k) - x k) \{n..n+p\} = x (\text{Suc } (n+p)) - x n \rangle$ **for** p

by (*simp add: sum-Suc-diff*)

moreover **have** $\langle (\lambda N. \text{sum } (\lambda k. x (\text{Suc } k) - x k) \{n..N\}) (n + t) \rangle$

$= (\lambda p. \text{sum } (\lambda k. x (\text{Suc } k) - x k) \{n..n+p\}) t$ **for** t

by *blast*

ultimately have $\langle (\lambda p. (\lambda N. \text{sum } (\lambda k. x (\text{Suc } k) - x k) \{n..N\}) (n + p)) \longrightarrow l - x n \rangle$
using *f1 by simp*
hence $\langle (\lambda p. (\lambda N. \text{sum } (\lambda k. x (\text{Suc } k) - x k) \{n..N\}) (p + n)) \longrightarrow l - x n \rangle$
by (*simp add: add.commute*)
hence $\langle (\lambda p. (\lambda N. \text{sum } (\lambda k. x (\text{Suc } k) - x k) \{n..N\}) p) \longrightarrow l - x n \rangle$
using *Topological-Spaces.LIMSEQ-offset* [**where** $f = (\lambda N. \text{sum } (\lambda k. x (\text{Suc } k) - x k) \{n..N\})$]
and $a = l - x n$ **and** $k = n$ **by** *blast*
hence $\langle (\lambda M. (\lambda N. \text{sum } (\lambda k. x (\text{Suc } k) - x k) \{n..N\}) M) \longrightarrow l - x n \rangle$
by *simp*
thus *?thesis* **by** *blast*
qed

lemma *bound-Cauchy-to-lim*:

assumes $\langle y \longrightarrow x \rangle$ **and** $\langle \bigwedge n. \|y (\text{Suc } n) - y n\| \leq c \hat{\ } n \rangle$ **and** $\langle y 0 = 0 \rangle$ **and**
 $\langle c < 1 \rangle$
shows $\langle \|x - y (\text{Suc } n)\| \leq (c / (1 - c)) * c \hat{\ } n \rangle$

Inequality about a sequence of approximations assuming that the sequence of differences is bounded by a geometric progression. Explanation: Let y be a sequence converging to x . If y satisfies the inequality $\|y (\text{Suc } n) - y n\| \leq c \hat{\ } n$ for some $c < 1$ and assuming $y 0 = 0$ then the inequality $\|x - y (\text{Suc } n)\| \leq (c / (1 - c)) * c \hat{\ } n$ holds.

proof –

have $\langle c \geq 0 \rangle$
using $\langle \bigwedge n. \|y (\text{Suc } n) - y n\| \leq c \hat{\ } n \rangle$
by (*metis dual-order.trans norm-ge-zero power-one-right*)
have *norm-1*: $\langle \text{norm } (\sum k = \text{Suc } n..N. y (\text{Suc } k) - y k) \leq (c \hat{\ } \text{Suc } n) / (1 - c) \rangle$ **for** N
proof (*cases* $\langle N < \text{Suc } n \rangle$)
case *True*
hence $\langle \|\text{sum } (\lambda k. y (\text{Suc } k) - y k) \{ \text{Suc } n .. N \}\| = 0 \rangle$
by *auto*
thus *?thesis* **using** $\langle c \geq 0 \rangle$ $\langle c < 1 \rangle$ **by** *auto*
next
case *False*
hence $\langle N \geq \text{Suc } n \rangle$
by *auto*
have $\langle c \hat{\ } (\text{Suc } N) \geq 0 \rangle$
using $\langle c \geq 0 \rangle$ **by** *auto*
have $\langle 1 - c > 0 \rangle$
by (*simp add: c < 1*)
hence $\langle (1 - c) / (1 - c) = 1 \rangle$
by *auto*
have $\langle \|\text{sum } (\lambda k. y (\text{Suc } k) - y k) \{ \text{Suc } n .. N \}\| \leq (\text{sum } (\lambda k. \|y (\text{Suc } k) - y k\|) \{ \text{Suc } n .. N \}) \rangle$
by (*simp add: sum-norm-le*)

hence $\langle \| \text{sum } (\lambda k. y (\text{Suc } k) - y k) \{ \text{Suc } n .. N \} \| \leq (\text{sum } (\text{power } c) \{ \text{Suc } n .. N \}) \rangle$
by (*simp add: assms(2) sum-norm-le*)
hence $\langle (1 - c) * \| \text{sum } (\lambda k. y (\text{Suc } k) - y k) \{ \text{Suc } n .. N \} \| \leq (1 - c) * (\text{sum } (\text{power } c) \{ \text{Suc } n .. N \}) \rangle$
using $\langle 0 < 1 - c \rangle$ *mult-le-cancel-left-pos* **by** *blast*
also have $\langle \dots = c^\wedge(\text{Suc } n) - c^\wedge(\text{Suc } N) \rangle$
using *Set-Interval.sum-gp-multiplied* $\langle \text{Suc } n \leq N \rangle$ **by** *blast*
also have $\langle \dots \leq c^\wedge(\text{Suc } n) \rangle$
using $\langle c^\wedge(\text{Suc } N) \geq 0 \rangle$ **by** *auto*
finally have $\langle (1 - c) * \| \sum k = \text{Suc } n..N. y (\text{Suc } k) - y k \| \leq c^\wedge \text{Suc } n \rangle$
by *blast*
hence $\langle ((1 - c) * \| \sum k = \text{Suc } n..N. y (\text{Suc } k) - y k \|) / (1 - c) \leq (c^\wedge \text{Suc } n) / (1 - c) \rangle$
using $\langle 0 < 1 - c \rangle$ *divide-le-cancel* **by** *fastforce*
thus $\langle \| \sum k = \text{Suc } n..N. y (\text{Suc } k) - y k \| \leq (c^\wedge \text{Suc } n) / (1 - c) \rangle$
using $\langle 0 < 1 - c \rangle$ **by** *auto*
qed
have $\langle (\lambda N. (\text{sum } (\lambda k. y (\text{Suc } k) - y k) \{ \text{Suc } n .. N \})) \longrightarrow x - y (\text{Suc } n) \rangle$
by (*metis (no-types) <y >> x identity-telescopic*)
hence $\langle (\lambda N. \| \text{sum } (\lambda k. y (\text{Suc } k) - y k) \{ \text{Suc } n .. N \} \|) \longrightarrow \| x - y (\text{Suc } n) \| \rangle$
using *tendsto-norm* **by** *blast*
hence $\langle \| x - y (\text{Suc } n) \| \leq (c^\wedge \text{Suc } n) / (1 - c) \rangle$
using *norm-1 Lim-bounded* **by** *blast*
hence $\langle \| x - y (\text{Suc } n) \| \leq (c^\wedge \text{Suc } n) / (1 - c) \rangle$
by *auto*
moreover have $\langle (c^\wedge \text{Suc } n) / (1 - c) = (c / (1 - c)) * (c^\wedge n) \rangle$
by (*simp add: divide-inverse-commute*)
ultimately show $\langle \| x - y (\text{Suc } n) \| \leq (c / (1 - c)) * (c^\wedge n) \rangle$ **by** *linarith*
qed

lemma *onorm-open-ball:*

includes *norm-syntax*

shows $\langle \| f \| = \text{Sup } \{ \| f *_v x \| \mid x. \| x \| < 1 \} \rangle$

Explanation: Let f be a bounded linear operator. The operator norm of f is the supremum of $\| f *_v x \|$ for x such that $\| x \| < 1$.

proof(*cases* $\langle (\text{UNIV}::'a \text{ set}) = 0 \rangle$)

case *True*

hence $\langle x = 0 \rangle$ **for** $x::'a$

by *auto*

hence $\langle f *_v x = 0 \rangle$ **for** x

by (*metis (full-types) blinfun.zero-right*)

hence $\langle \| f \| = 0 \rangle$

by (*simp add: blinfun-eqI zero-blinfun.rep-eq*)

have $\langle \{ \| f *_v x \| \mid x. \| x \| < 1 \} = \{ 0 \} \rangle$

by (*smt (verit, ccfv-SIG) Collect-cong* $\langle \bigwedge x. f *_v x = 0 \rangle$ *norm-zero singleton-conv*)

```

hence ⟨Sup { ||f *v x|| | x. ||x|| < 1 } = 0⟩
  by simp
thus ?thesis using ⟨||f|| = 0⟩ by auto
next
case False
hence ⟨(UNIV::'a set) ≠ 0⟩
  by simp
have nonnegative: ⟨||f *v x|| ≥ 0⟩ for x
  by simp
have ⟨∃ x::'a. x ≠ 0⟩
  using ⟨UNIV ≠ 0⟩ by auto
then obtain x::'a where ⟨x ≠ 0⟩
  by blast
hence ⟨||x|| ≠ 0⟩
  by auto
define y where ⟨y = x /R ||x||⟩
have ⟨norm y = || x /R ||x|| ||⟩
  unfolding y-def by auto
also have ⟨... = ||x|| /R ||x||⟩
  by auto
also have ⟨... = 1⟩
  using ⟨||x|| ≠ 0⟩ by auto
finally have ⟨||y|| = 1⟩
  by blast
hence norm-1-non-empty: ⟨{ ||f *v x|| | x. ||x|| = 1 } ≠ {}⟩
  by blast
have norm-1-bounded: ⟨bdd-above { ||f *v x|| | x. ||x|| = 1 }⟩
  unfolding bdd-above-def apply auto
  by (metis norm-blinfun)
have norm-less-1-non-empty: ⟨{ ||f *v x|| | x. ||x|| < 1 } ≠ {}⟩
  by (metis (mono-tags, lifting) Collect-empty-eq-bot bot-empty-eq empty-iff norm-zero
    zero-less-one)
have norm-less-1-bounded: ⟨bdd-above { ||f *v x|| | x. ||x|| < 1 }⟩
proof-
  have ⟨∃ r. ||a r|| < 1 ⟶ ||f *v (a r)|| ≤ r⟩ for a :: real ⇒ 'a
  proof-
    obtain r :: ('a ⇒L 'b) ⇒ real where
      ∧f x. 0 ≤ r f ∧ (bounded-linear f ⟶ ||f *v x|| ≤ ||x|| * r f)
    by (metis mult.commute norm-blinfun norm-ge-zero)
    have ⟨¬ ||f|| < 0⟩
      by simp
    hence ⟨∃ r. ||f|| * ||a r|| ≤ r⟩ ∨ ⟨∃ r. ||a r|| < 1 ⟶ ||f *v a r|| ≤ r⟩
      by (meson less-eq-real-def mult-le-cancel-left2)
    thus ?thesis using dual-order.trans norm-blinfun by blast
  qed
hence ⟨∃ M. ∀ x. ||x|| < 1 ⟶ ||f *v x|| ≤ M⟩
  by metis
thus ?thesis by auto

```

```

qed
have Sup-non-neg:  $\langle \text{Sup } \{\|f *_{\nu} x\| \mid x. \|x\| = 1\} \geq 0 \rangle$ 
by (metis (mono-tags, lifting)  $\langle \|y\| = 1 \rangle$  cSup-upper2 mem-Collect-eq norm-1-bounded
norm-ge-zero)
have  $\langle \{0::\text{real}\} \neq \{\} \rangle$ 
by simp
have  $\langle \text{bdd-above } \{0::\text{real}\} \rangle$ 
by simp
show  $\langle \|f\| = \text{Sup } \{\|f *_{\nu} x\| \mid x. \|x\| < 1\} \rangle$ 
proof (cases  $\langle \forall x. f *_{\nu} x = 0 \rangle$ )
  case True
    have  $\langle \|f *_{\nu} x\| = 0 \rangle$  for  $x$ 
    by (simp add: True)
    hence  $\langle \{\|f *_{\nu} x\| \mid x. \|x\| < 1\} \subseteq \{0\} \rangle$ 
    by blast
    moreover have  $\langle \{\|f *_{\nu} x\| \mid x. \|x\| < 1\} \supseteq \{0\} \rangle$ 
    using calculation norm-less-1-non-empty by fastforce
    ultimately have  $\langle \{\|f *_{\nu} x\| \mid x. \|x\| < 1\} = \{0\} \rangle$ 
    by blast
    hence Sup1:  $\langle \text{Sup } \{\|f *_{\nu} x\| \mid x. \|x\| < 1\} = 0 \rangle$ 
    by simp
    have  $\langle \|f\| = 0 \rangle$ 
    by (simp add: True blinfun-eqI)
    moreover have  $\langle \text{Sup } \{\|f *_{\nu} x\| \mid x. \|x\| < 1\} = 0 \rangle$ 
    using Sup1 by blast
    ultimately show ?thesis by simp
  next
    case False
    have norm-f-eq-leg:  $\langle y \in \{\|f *_{\nu} x\| \mid x. \|x\| = 1\} \implies$ 
       $y \leq \text{Sup } \{\|f *_{\nu} x\| \mid x. \|x\| < 1\} \rangle$  for  $y$ 
    proof–
      assume  $\langle y \in \{\|f *_{\nu} x\| \mid x. \|x\| = 1\} \rangle$ 
      hence  $\langle \exists x. y = \|f *_{\nu} x\| \wedge \|x\| = 1 \rangle$ 
      by blast
      then obtain  $x$  where  $\langle y = \|f *_{\nu} x\| \rangle$  and  $\langle \|x\| = 1 \rangle$ 
      by auto
      define  $y'$  where  $\langle y' n = (1 - (\text{inverse } (\text{real } (\text{Suc } n)))) *_{\mathbb{R}} y \rangle$  for  $n$ 
      have  $\langle y' n \in \{\|f *_{\nu} x\| \mid x. \|x\| < 1\} \rangle$  for  $n$ 
      proof–
        have  $\langle y' n = (1 - (\text{inverse } (\text{real } (\text{Suc } n)))) *_{\mathbb{R}} \|f *_{\nu} x\| \rangle$ 
        using y'-def  $\langle y = \|f *_{\nu} x\| \rangle$  by blast
        also have  $\langle \dots = |(1 - (\text{inverse } (\text{real } (\text{Suc } n))))| *_{\mathbb{R}} \|f *_{\nu} x\| \rangle$ 
        by (metis (mono-tags, opaque-lifting)  $\langle y = \|f *_{\nu} x\| \rangle$  abs-1 abs-le-self-iff
abs-of-nat
abs-of-nonneg add-diff-cancel-left' add-eq-if cancel-comm-monoid-add-class.diff-cancel
diff-ge-0-iff-ge eq-iff-diff-eq-0 inverse-1 inverse-le-iff-le nat.distinct(1))
        of-nat-0
        of-nat-Suc of-nat-le-0-iff zero-less-abs-iff zero-neq-one)
        also have  $\langle \dots = \|f *_{\nu} ((1 - (\text{inverse } (\text{real } (\text{Suc } n)))) *_{\mathbb{R}} x)\| \rangle$ 

```

by (simp add: blinfun.scaleR-right)
 finally have $y'-1: \langle y' n = \|f *_v ((1 - (\text{inverse} (\text{real} (\text{Suc } n)))) *_R x) \| \rangle$
 by blast
 have $\langle \| (1 - (\text{inverse} (\text{Suc } n))) *_R x \| = (1 - (\text{inverse} (\text{real} (\text{Suc } n)))) * \|x\| \rangle$
 by (simp add: linordered-field-class.inverse-le-1-iff)
 hence $\langle \| (1 - (\text{inverse} (\text{Suc } n))) *_R x \| < 1 \rangle$
 by (simp add: $\langle \|x\| = 1 \rangle$)
 thus ?thesis using $y'-1$ by blast
 qed
 have $\langle (\lambda n. (1 - (\text{inverse} (\text{real} (\text{Suc } n))))) \longrightarrow 1 \rangle$
 using Limits.LIMSEQ-inverse-real-of-nat-add-minus by simp
 hence $\langle (\lambda n. (1 - (\text{inverse} (\text{real} (\text{Suc } n)))) *_R y \longrightarrow 1 *_R y \rangle$
 using Limits.tendsto-scaleR by blast
 hence $\langle (\lambda n. (1 - (\text{inverse} (\text{real} (\text{Suc } n)))) *_R y \longrightarrow y \rangle$
 by simp
 hence $\langle (\lambda n. y' n) \longrightarrow y \rangle$
 using y' -def by simp
 hence $\langle y' \longrightarrow y \rangle$
 by simp
 have $\langle y' n \leq \text{Sup} \{ \|f *_v x\| \mid x. \|x\| < 1 \} \rangle$ for n
 using cSup-upper $\langle \bigwedge n. y' n \in \{ \|f *_v x\| \mid x. \|x\| < 1 \} \rangle$ norm-less-1-bounded
 by blast
 hence $\langle y \leq \text{Sup} \{ \|f *_v x\| \mid x. \|x\| < 1 \} \rangle$
 using $\langle y' \longrightarrow y \rangle$ Topological-Spaces.Sup-lim by (meson LIMSEQ-le-const2)
 thus ?thesis by blast
 qed
 hence $\langle \text{Sup} \{ \|f *_v x\| \mid x. \|x\| = 1 \} \leq \text{Sup} \{ \|f *_v x\| \mid x. \|x\| < 1 \} \rangle$
 by (metis (lifting) cSup-least norm-1-non-empty)
 have $\langle y \in \{ \|f *_v x\| \mid x. \|x\| < 1 \} \implies y \leq \text{Sup} \{ \|f *_v x\| \mid x. \|x\| = 1 \} \rangle$ for y
 proof (cases $\langle y = 0 \rangle$)
 case True thus ?thesis by (simp add: Sup-non-neg)
 next
 case False
 hence $\langle y \neq 0 \rangle$ by blast
 assume $\langle y \in \{ \|f *_v x\| \mid x. \|x\| < 1 \} \rangle$
 hence $\langle \exists x. y = \|f *_v x\| \wedge \|x\| < 1 \rangle$
 by blast
 then obtain x where $\langle y = \|f *_v x\| \rangle$ and $\langle \|x\| < 1 \rangle$
 by blast
 have $\langle (1/\|x\|) * y = (1/\|x\|) * \|f *_v x\| \rangle$
 by (simp add: $\langle y = \|f *_v x\| \rangle$)
 also have $\langle \dots = |1/\|x\|| * \|f *_v x\| \rangle$
 by simp
 also have $\langle \dots = \|(1/\|x\|) *_R (f *_v x)\| \rangle$
 by simp
 also have $\langle \dots = \|f *_v ((1/\|x\|) *_R x)\| \rangle$
 by (simp add: blinfun.scaleR-right)
 finally have $\langle (1/\|x\|) * y = \|f *_v ((1/\|x\|) *_R x)\| \rangle$

```

    by blast
  have ⟨x ≠ 0⟩
    using ⟨y ≠ 0⟩ ⟨y = ‖f *v x‖⟩ blinfun.zero-right by auto
  have ⟨‖(1/‖x‖) *R x‖ = |(1/‖x‖)| * ‖x‖⟩
    by simp
  also have ⟨... = (1/‖x‖) * ‖x‖⟩
    by simp
  finally have ⟨‖(1/‖x‖) *R x‖ = 1⟩
    using ⟨x ≠ 0⟩ by simp
  hence ⟨(1/‖x‖) * y ∈ {‖f *v x‖ | x. ‖x‖ = 1}⟩
    using ⟨1 / ‖x‖ * y = ‖f *v (1 / ‖x‖) *R x‖⟩ by blast
  hence ⟨(1/‖x‖) * y ≤ Sup {‖f *v x‖ | x. ‖x‖ = 1}⟩
    by (simp add: cSup-upper norm-1-bounded)
  moreover have ⟨y ≤ (1/‖x‖) * y⟩
    by (metis ⟨‖x‖ < 1⟩ ⟨y = ‖f *v x‖⟩ mult-le-cancel-right1 norm-not-less-zero)

    order.strict-implies-order ⟨x ≠ 0⟩ less-divide-eq-1-pos zero-less-norm-iff)
  ultimately show ?thesis by linarith
qed
  hence ⟨Sup {‖f *v x‖ | x. ‖x‖ < 1} ≤ Sup {‖f *v x‖ | x. ‖x‖ = 1}⟩
    by (smt (verit, del-insts) less-cSupD norm-less-1-non-empty)
  hence ⟨Sup {‖f *v x‖ | x. ‖x‖ = 1} = Sup {‖f *v x‖ | x. ‖x‖ < 1}⟩
    using ⟨Sup {‖f *v x‖ | x. norm x = 1} ≤ Sup {‖f *v x‖ | x. ‖x‖ < 1}⟩ by
linarith
  have f1: ⟨(SUP x. ‖f *v x‖ / ‖x‖) = Sup {‖f *v x‖ / ‖x‖ | x. True}⟩
    by (simp add: full-SetCompr-eq)
  have ⟨y ∈ {‖f *v x‖ / ‖x‖ | x. True} ⟹ y ∈ {‖f *v x‖ | x. ‖x‖ = 1} ∪ {0}⟩
    for y
  proof-
    assume ⟨y ∈ {‖f *v x‖ / ‖x‖ | x. True}⟩ show ?thesis
  proof(cases ⟨y = 0⟩)
    case True thus ?thesis by simp
  next
    case False
    have ⟨∃ x. y = ‖f *v x‖ / ‖x‖⟩
      using ⟨y ∈ {‖f *v x‖ / ‖x‖ | x. True}⟩ by auto
    then obtain x where ⟨y = ‖f *v x‖ / ‖x‖⟩
      by blast
    hence ⟨y = |(1/‖x‖)| * ‖f *v x‖⟩
      by simp
    hence ⟨y = ‖(1/‖x‖) *R (f *v x)‖⟩
      by simp
    hence ⟨y = ‖f ((1/‖x‖) *R x)‖⟩
      by (simp add: blinfun.scaleR-right)
    moreover have ⟨‖(1/‖x‖) *R x‖ = 1⟩
      using False ⟨y = ‖f *v x‖ / ‖x‖⟩ by auto
    ultimately have ⟨y ∈ {‖f *v x‖ | x. ‖x‖ = 1}⟩
      by blast
    thus ?thesis by blast
  end

```

```

    qed
  qed
  moreover have  $\langle y \in \{\|f x\ | x. \|x\| = 1\} \cup \{0\} \implies y \in \{\|f *_v x\ | \|x\| | x. True\} \rangle$ 
  for  $y$ 
  proof(cases  $\langle y = 0 \rangle$ )
    case True thus ?thesis by auto
  next
    case False
    hence  $\langle y \notin \{0\} \rangle$ 
    by simp
    moreover assume  $\langle y \in \{\|f *_v x\ | x. \|x\| = 1\} \cup \{0\} \rangle$ 
    ultimately have  $\langle y \in \{\|f *_v x\ | x. \|x\| = 1\} \rangle$ 
    by simp
    then obtain  $x$  where  $\langle \|x\| = 1 \rangle$  and  $\langle y = \|f *_v x\| \rangle$ 
    by auto
    have  $\langle y = \|f *_v x\ | \|x\| \rangle$  using  $\langle \|x\| = 1 \rangle$   $\langle y = \|f *_v x\| \rangle$ 
    by simp
    thus ?thesis by auto
  qed
  ultimately have  $\langle \{\|f *_v x\ | \|x\| | x. True\} = \{\|f *_v x\ | x. \|x\| = 1\} \cup \{0\} \rangle$ 
  by blast
  hence  $\langle Sup \{\|f *_v x\ | \|x\| | x. True\} = Sup (\{\|f *_v x\ | x. \|x\| = 1\} \cup \{0\}) \rangle$ 
  by simp
  have  $\bigwedge r s. \neg (r::real) \leq s \vee sup r s = s$ 
  by (metis (lifting) sup.absorb-iff1 sup-commute)
  hence  $\langle Sup (\{\|f *_v x\ | x. \|x\| = 1\} \cup \{(0::real)\}) = max (Sup \{\|f *_v x\ | x. \|x\| = 1\}) (Sup \{0::real\}) \rangle$ 
  using  $\langle 0 \leq Sup \{\|f *_v x\ | x. \|x\| = 1\} \rangle$   $\langle bdd-above \{0\} \rangle$   $\langle \{0\} \neq \{\} \rangle$ 
  cSup-singleton
  cSup-union-distrib max.absorb-iff1 sup-commute norm-1-bounded norm-1-non-empty
  by (metis (no-types, lifting) )
  moreover have  $\langle Sup \{(0::real)\} = (0::real) \rangle$ 
  by simp
  ultimately have  $\langle Sup (\{\|f *_v x\ | x. \|x\| = 1\} \cup \{0\}) = Sup \{\|f *_v x\ | x. \|x\| = 1\} \rangle$ 
  using Sup-non-neg by linarith
  moreover have  $\langle Sup (\{\|f *_v x\ | x. \|x\| = 1\} \cup \{0\}) = max (Sup \{\|f *_v x\ | x. \|x\| = 1\}) (Sup \{0\}) \rangle$ 
  using Sup-non-neg  $\langle Sup (\{\|f *_v x\ | x. \|x\| = 1\} \cup \{0\}) = max (Sup \{\|f *_v x\ | x. \|x\| = 1\}) (Sup \{0\}) \rangle$ 
  by auto
  ultimately have f2:  $\langle Sup \{\|f *_v x\ | \|x\| | x. True\} = Sup \{\|f *_v x\ | x. \|x\| = 1\} \rangle$ 
  using  $\langle Sup \{\|f *_v x\ | \|x\| | x. True\} = Sup (\{\|f *_v x\ | x. \|x\| = 1\} \cup \{0\}) \rangle$ 
  by linarith
  have  $\langle (SUP x. \|f *_v x\ | \|x\|) = Sup \{\|f *_v x\ | x. \|x\| = 1\} \rangle$ 
  using f1 f2 by linarith
  hence  $\langle (SUP x. \|f *_v x\ | \|x\|) = Sup \{\|f *_v x\ | x. \|x\| < 1\} \rangle$ 

```

by (simp add: $\langle \text{Sup } \{\|f *_{\nu} x\| \mid x. \|x\| = 1\} = \text{Sup } \{\|f *_{\nu} x\| \mid x. \|x\| < 1\} \rangle$)

thus *?thesis* apply transfer by (simp add: onorm-def)

qed

qed

lemma onorm-r:

includes norm-syntax

assumes $\langle r > 0 \rangle$

shows $\langle \|f\| = \text{Sup } ((\lambda x. \|f *_{\nu} x\|) \text{ ` } (ball\ 0\ r)) / r \rangle$

Explanation: The norm of f is $1 / r$ of the supremum of the norm of $f *_{\nu} x$ for x in the ball of radius r centered at the origin.

proof–

have $\langle \|f\| = \text{Sup } \{\|f *_{\nu} x\| \mid x. \|x\| < 1\} \rangle$

using onorm-open-ball by blast

moreover have *: $\langle \{\|f *_{\nu} x\| \mid x. \|x\| < 1\} = (\lambda x. \|f *_{\nu} x\|) \text{ ` } (ball\ 0\ 1) \rangle$

unfolding ball-def by auto

ultimately have onorm-f: $\langle \|f\| = \text{Sup } ((\lambda x. \|f *_{\nu} x\|) \text{ ` } (ball\ 0\ 1)) \rangle$

by simp

have s2: $\langle x \in (\lambda t. r *_{\mathbb{R}} \|f *_{\nu} t\|) \text{ ` } ball\ 0\ 1 \implies x \leq r * \text{Sup } ((\lambda t. \|f *_{\nu} t\|) \text{ ` } ball\ 0\ 1) \rangle$ for x

proof–

assume $\langle x \in (\lambda t. r *_{\mathbb{R}} \|f *_{\nu} t\|) \text{ ` } ball\ 0\ 1 \rangle$

hence $\langle \exists t. x = r *_{\mathbb{R}} \|f *_{\nu} t\| \wedge \|t\| < 1 \rangle$

by auto

then obtain t where t : $\langle x = r *_{\mathbb{R}} \|f *_{\nu} t\| \wedge \|t\| < 1 \rangle$

by blast

define y where $\langle y = x /_{\mathbb{R}} r \rangle$

have $\langle x = r * (\text{inverse } r * x) \rangle$

using $\langle x = r *_{\mathbb{R}} \text{norm } (f\ t) \rangle$ by auto

hence $\langle x - (r * (\text{inverse } r * x)) \leq 0 \rangle$

by linarith

hence $\langle x \leq r * (x /_{\mathbb{R}} r) \rangle$

by auto

have $\langle y \in (\lambda k. \|f *_{\nu} k\|) \text{ ` } ball\ 0\ 1 \rangle$

unfolding y-def using assms t * by fastforce

moreover have $\langle x \leq r * y \rangle$

using $\langle x \leq r * (x /_{\mathbb{R}} r) \rangle$ y-def by blast

ultimately have y-norm-f: $\langle y \in (\lambda t. \|f *_{\nu} t\|) \text{ ` } ball\ 0\ 1 \wedge x \leq r * y \rangle$

by blast

have $\langle (\lambda t. \|f *_{\nu} t\|) \text{ ` } ball\ 0\ 1 \neq \{\} \rangle$

by simp

moreover have $\langle \text{bdd-above } ((\lambda t. \|f *_{\nu} t\|) \text{ ` } ball\ 0\ 1) \rangle$

by (simp add: bounded-linear-image blinfun.bounded-linear-right bounded-imp-bdd-above

bounded-norm-comp)

moreover have $\langle \exists y. y \in (\lambda t. \|f *_{\nu} t\|) \text{ ` } ball\ 0\ 1 \wedge x \leq r * y \rangle$

using y-norm-f by blast

ultimately show *?thesis*
 by (*meson assms cSup-upper dual-order.trans mult-le-cancel-left-pos*)
qed
have $s3: \langle \bigwedge x. x \in (\lambda t. r * \|f *_{\nu} t\|) \text{ 'ball } 0 \ 1 \implies x \leq y \rangle \implies$
 $r * \text{Sup } ((\lambda t. \|f *_{\nu} t\|) \text{ 'ball } 0 \ 1) \leq y \text{ for } y$
proof–
assume $\langle \bigwedge x. x \in (\lambda t. r * \|f *_{\nu} t\|) \text{ 'ball } 0 \ 1 \implies x \leq y \rangle$
have $x\text{-leq}: \langle x \in (\lambda t. \|f *_{\nu} t\|) \text{ 'ball } 0 \ 1 \implies x \leq y / r \rangle$ **for** x
proof–
assume $\langle x \in (\lambda t. \|f *_{\nu} t\|) \text{ 'ball } 0 \ 1 \rangle$
then obtain t **where** $\langle t \in \text{ball } (0::'a) \ 1 \rangle$ **and** $\langle x = \|f *_{\nu} t\| \rangle$
by *auto*
define x' **where** $\langle x' = r *_{\mathbb{R}} x \rangle$
have $\langle x' = r * \|f *_{\nu} t\| \rangle$
by (*simp add: x = \|f *_{\nu} t\| x'-def*)
hence $\langle x' \in (\lambda t. r * \|f *_{\nu} t\|) \text{ 'ball } 0 \ 1 \rangle$
using $\langle t \in \text{ball } (0::'a) \ 1 \rangle$ **by** *auto*
hence $\langle x' \leq y \rangle$
using $\langle \bigwedge x. x \in (\lambda t. r * \|f *_{\nu} t\|) \text{ 'ball } 0 \ 1 \implies x \leq y \rangle$ **by** *blast*
thus $\langle x \leq y / r \rangle$
unfolding $x'\text{-def}$ **using** $\langle r > 0 \rangle$ **by** (*simp add: mult.commute pos-le-divide-eq*)

qed
have $\langle (\lambda t. \|f *_{\nu} t\|) \text{ 'ball } 0 \ 1 \neq \{\} \rangle$
by *simp*
moreover have $\langle \text{bdd-above } ((\lambda t. \|f *_{\nu} t\|) \text{ 'ball } 0 \ 1) \rangle$
by (*simp add: bounded-linear-image blinfun.bounded-linear-right bounded-imp-bdd-above*
bounded-norm-comp)
ultimately have $\langle \text{Sup } ((\lambda t. \|f *_{\nu} t\|) \text{ 'ball } 0 \ 1) \leq y / r \rangle$
using $x\text{-leq}$ **by** (*simp add: bdd-above ((\lambda t. \|f *_{\nu} t\|) \text{ 'ball } 0 \ 1) cSup-least*)
thus *?thesis* **using** $\langle r > 0 \rangle$
by (*simp add: mult.commute pos-le-divide-eq*)
qed
have $\text{norm-scaleR}: \langle \text{norm} \circ ((*_R) \ r) = ((*_R) \ |r|) \circ (\text{norm}::'a \Rightarrow \text{real}) \rangle$
by *auto*
have $f\text{-x1}: \langle f (r *_{\mathbb{R}} x) = r *_{\mathbb{R}} f x \rangle$ **for** x
by (*simp add: blinfun.scaleR-right*)
have $\langle \text{ball } (0::'a) \ r = ((*_R) \ r) \text{ 'ball } 0 \ 1 \rangle$
by (*smt (verit) assms ball-scale nonzero-mult-div-cancel-left right-inverse-eq*
scale-zero-right)
hence $\langle \text{Sup } ((\lambda t. \|f *_{\nu} t\|) \text{ 'ball } 0 \ r) = \text{Sup } ((\lambda t. \|f *_{\nu} t\|) \text{ 'ball } 0 \ 1) \text{ 'ball } 0 \ 1 \rangle$
by *simp*
also have $\langle \dots = \text{Sup } (((\lambda t. \|f *_{\nu} t\|) \circ ((*_R) \ r)) \text{ 'ball } 0 \ 1) \rangle$
using *Sup.SUP-image* **by** *auto*
also have $\langle \dots = \text{Sup } ((\lambda t. \|f *_{\nu} (r *_{\mathbb{R}} t)\|) \text{ 'ball } 0 \ 1) \rangle$
using $f\text{-x1}$ **by** (*simp add: comp-assoc*)
also have $\langle \dots = \text{Sup } ((\lambda t. |r| *_{\mathbb{R}} \|f *_{\nu} t\|) \text{ 'ball } 0 \ 1) \rangle$

```

    using norm-scaleR f-x1 by auto
  also have ⟨... = Sup ((λt. r *R ||f *v t||) ‘(ball 0 1)⟩
    using ⟨r > 0⟩ by auto
  also have ⟨... = r * Sup ((λt. ||f *v t||) ‘(ball 0 1)⟩
    apply (rule cSup-eq-non-empty) apply simp using s2 apply auto using s3
  by auto
  also have ⟨... = r * ||f||⟩
    using onorm-f by auto
  finally have ⟨Sup ((λt. ||f *v t||) ‘ball 0 r) = r * ||f||⟩
    by blast
  thus ⟨||f|| = Sup ((λx. ||f *v x||) ‘(ball 0 r)) / r⟩ using ⟨r > 0⟩ by simp
  qed

```

Pointwise convergence

```

definition pointwise-convergent-to ::
  ⟨(nat ⇒ ('a ⇒ 'b::topological-space)) ⇒ ('a ⇒ 'b) ⇒ bool⟩
  ⟨((-) / -pointwise→ (-)⟩ [60, 60] 60 where
  ⟨pointwise-convergent-to x l = (∀ t::'a. (λ n. (x n) t) → l t)⟩

```

```

lemma linear-limit-linear:
  fixes f :: ⟨- ⇒ ('a::real-vector ⇒ 'b::real-normed-vector)⟩
  assumes ⟨∧ n. linear (f n)⟩ and ⟨f -pointwise→ F⟩
  shows ⟨linear F⟩

```

Explanation: If a family of linear operators converges pointwise, then the limit is also a linear operator.

```

proof
  show F (x + y) = F x + F y for x y
  proof -
    have ∀ a. F a = lim (λn. f n a)
      using ⟨f -pointwise→ F⟩ unfolding pointwise-convergent-to-def by (metis
      (full-types) limI)
    moreover have ∀ f b c g. (lim (λn. g n + f n) = (b::'b) + c ∨ ¬ f → c)
      ∨ ¬ g → b
      by (metis (no-types) limI tendsto-add)
    moreover have ∧ a. (λn. f n a) → F a
      using assms(2) pointwise-convergent-to-def by force
    ultimately have
      lim-sum: ⟨lim (λ n. (f n) x + (f n) y) = lim (λ n. (f n) x) + lim (λ n. (f n)
      y)⟩
      by metis
    have ⟨(f n) (x + y) = (f n) x + (f n) y⟩ for n
      using ⟨∧ n. linear (f n)⟩ unfolding linear-def using Real-Vector-Spaces.linear-iff
      assms(1)
      by auto
    hence ⟨lim (λ n. (f n) (x + y)) = lim (λ n. (f n) x + (f n) y)⟩
      by simp
    hence ⟨lim (λ n. (f n) (x + y)) = lim (λ n. (f n) x) + lim (λ n. (f n) y)⟩
      using lim-sum by simp
  qed

```

moreover have $\langle (\lambda n. (f n) (x + y)) \longrightarrow F (x + y) \rangle$
using $\langle f \text{ -pointwise} \rightarrow F \rangle$ **unfolding** *pointwise-convergent-to-def* **by** *blast*
moreover have $\langle (\lambda n. (f n) x) \longrightarrow F x \rangle$
using $\langle f \text{ -pointwise} \rightarrow F \rangle$ **unfolding** *pointwise-convergent-to-def* **by** *blast*
moreover have $\langle (\lambda n. (f n) y) \longrightarrow F y \rangle$
using $\langle f \text{ -pointwise} \rightarrow F \rangle$ **unfolding** *pointwise-convergent-to-def* **by** *blast*
ultimately show *?thesis*
by *(metis limI)*

qed

show $F (r *_R x) = r *_R F x$ **for** r **and** x

proof –

have $\langle (f n) (r *_R x) = r *_R (f n) x \rangle$ **for** n
using $\langle \bigwedge n. \text{linear} (f n) \rangle$
by *(simp add: Real-Vector-Spaces.linear-def real-vector.linear-scale)*
hence $\langle \lim (\lambda n. (f n) (r *_R x)) = \lim (\lambda n. r *_R (f n) x) \rangle$
by *simp*
have $\langle \text{convergent} (\lambda n. (f n) x) \rangle$
by *(metis assms(2) convergentI pointwise-convergent-to-def)*
moreover have $\langle \text{isCont} (\lambda t::'b. r *_R t) tt \rangle$ **for** tt
by *(simp add: bounded-linear-scaleR-right)*
ultimately have $\langle \lim (\lambda n. r *_R ((f n) x)) = r *_R \lim (\lambda n. (f n) x) \rangle$
using $\langle f \text{ -pointwise} \rightarrow F \rangle$ **unfolding** *pointwise-convergent-to-def*
by *(metis (mono-tags) isCont-tendsto-compose limI)*
hence $\langle \lim (\lambda n. (f n) (r *_R x)) = r *_R \lim (\lambda n. (f n) x) \rangle$
using $\langle \lim (\lambda n. (f n) (r *_R x)) = \lim (\lambda n. r *_R (f n) x) \rangle$ **by** *simp*
moreover have $\langle (\lambda n. (f n) x) \longrightarrow F x \rangle$
using $\langle f \text{ -pointwise} \rightarrow F \rangle$ **unfolding** *pointwise-convergent-to-def* **by** *blast*
moreover have $\langle (\lambda n. (f n) (r *_R x)) \longrightarrow F (r *_R x) \rangle$
using $\langle f \text{ -pointwise} \rightarrow F \rangle$ **unfolding** *pointwise-convergent-to-def* **by** *blast*
ultimately show *?thesis*
by *(metis limI)*

qed

qed

lemma *non-Cauchy-unbounded*:

fixes $a :: \text{real} \Rightarrow \text{real}$
assumes $\langle \bigwedge n. a n \geq 0 \rangle$ **and** $\langle e > 0 \rangle$
and $\langle \forall M. \exists m. \exists n. m \geq M \wedge n \geq M \wedge m > n \wedge \text{sum } a \{ \text{Suc } n..m \} \geq e \rangle$
shows $\langle (\lambda n. (\text{sum } a \{ 0..n \})) \longrightarrow \infty \rangle$

Explanation: If the sequence of partial sums of nonnegative terms is not Cauchy, then it converges to infinite.

proof –

define $S :: \text{ereal set}$ **where** $\langle S = \text{range} (\lambda n. \text{sum } a \{ 0..n \}) \rangle$
have $\langle \exists s \in S. k * e \leq s \rangle$ **for** $k :: \text{nat}$
proof *(induction k)*
case 0
from $\langle \forall M. \exists m. \exists n. m \geq M \wedge n \geq M \wedge m > n \wedge \text{sum } a \{ \text{Suc } n..m \} \geq e \rangle$

obtain $m\ n$ **where** $\langle m \geq 0 \rangle$ **and** $\langle n \geq 0 \rangle$ **and** $\langle m > n \rangle$ **and** $\langle \text{sum } a \{ \text{Suc } n..m \} \geq e \rangle$ **by** *blast*
have $\langle n < \text{Suc } n \rangle$
by *simp*
hence $\langle \{0..n\} \cup \{\text{Suc } n..m\} = \{0..m\} \rangle$
using *Set-Interval.ivl-disj-un(7)* $\langle n < m \rangle$ **by** *auto*
moreover have $\langle \text{finite } \{0..n\} \rangle$
by *simp*
moreover have $\langle \text{finite } \{\text{Suc } n..m\} \rangle$
by *simp*
moreover have $\langle \{0..n\} \cap \{\text{Suc } n..m\} = \{\} \rangle$
by *simp*
ultimately have $\langle \text{sum } a \{0..n\} + \text{sum } a \{\text{Suc } n..m\} = \text{sum } a \{0..m\} \rangle$
by *(metis sum.union-disjoint)*
moreover have $\langle \text{sum } a \{\text{Suc } n..m\} > 0 \rangle$
using $\langle e > 0 \rangle$ $\langle \text{sum } a \{\text{Suc } n..m\} \geq e \rangle$ **by** *linarith*
moreover have $\langle \text{sum } a \{0..n\} \geq 0 \rangle$
by *(simp add: assms(1) sum-nonneg)*
ultimately have $\langle \text{sum } a \{0..m\} > 0 \rangle$
by *linarith*
moreover have $\langle \text{sum } a \{0..m\} \in S \rangle$
unfolding *S-def* **by** *blast*
ultimately have $\langle \exists s \in S. 0 \leq s \rangle$
using *ereal-less-eq(5)* **by** *fastforce*
thus *?case*
by *(simp add: zero-ereal-def)*
next
case *(Suc k)*
assume $\langle \exists s \in S. k * e \leq s \rangle$
then obtain s **where** $\langle s \in S \rangle$ **and** $\langle \text{ereal } (k * e) \leq s \rangle$
by *blast*
have $\langle \exists N. s = \text{sum } a \{0..N\} \rangle$
using $\langle s \in S \rangle$ **unfolding** *S-def* **by** *blast*
then obtain N **where** $\langle s = \text{sum } a \{0..N\} \rangle$
by *blast*
from $\langle \forall M. \exists m. \exists n. m \geq M \wedge n \geq M \wedge m > n \wedge \text{sum } a \{\text{Suc } n..m\} \geq e \rangle$
obtain $m\ n$ **where** $\langle m \geq \text{Suc } N \rangle$ **and** $\langle n \geq \text{Suc } N \rangle$ **and** $\langle m > n \rangle$ **and** $\langle \text{sum } a \{\text{Suc } n..m\} \geq e \rangle$
by *blast*
have $\langle \text{finite } \{\text{Suc } N..n\} \rangle$
by *simp*
moreover have $\langle \text{finite } \{\text{Suc } n..m\} \rangle$
by *simp*
moreover have $\langle \{\text{Suc } N..n\} \cup \{\text{Suc } n..m\} = \{\text{Suc } N..m\} \rangle$
using *Set-Interval.ivl-disj-un*
by *(metis Suc N ≤ n < n < m atLeastSucAtMost-greaterThanAtMost order-less-imp-le)*
moreover have $\langle \{\} = \{\text{Suc } N..n\} \cap \{\text{Suc } n..m\} \rangle$
by *simp*

ultimately have $\langle \text{sum } a \{ \text{Suc } N..m \} = \text{sum } a \{ \text{Suc } N..n \} + \text{sum } a \{ \text{Suc } n..m \} \rangle$
by *(metis sum.union-disjoint)*
moreover have $\langle \text{sum } a \{ \text{Suc } N..n \} \geq 0 \rangle$
using $\langle \bigwedge n. a \ n \geq 0 \rangle$ **by** *(simp add: sum-nonneg)*
ultimately have $\langle \text{sum } a \{ \text{Suc } N..m \} \geq e \rangle$
using $\langle e \leq \text{sum } a \{ \text{Suc } n..m \} \rangle$ **by** *linarith*
have $\langle \text{finite } \{ 0..N \} \rangle$
by *simp*
have $\langle \text{finite } \{ \text{Suc } N..m \} \rangle$
by *simp*
moreover have $\langle \{ 0..N \} \cup \{ \text{Suc } N..m \} = \{ 0..m \} \rangle$
using *Set-Interval.int-disj-un*(γ) $\langle \text{Suc } N \leq m \rangle$ **by** *auto*
moreover have $\langle \{ 0..N \} \cap \{ \text{Suc } N..m \} = \{ \} \rangle$
by *simp*
ultimately have $\langle \text{sum } a \{ 0..N \} + \text{sum } a \{ \text{Suc } N..m \} = \text{sum } a \{ 0..m \} \rangle$
by *(metis finite {0..N} sum.union-disjoint)*
hence $\langle e + k * e \leq \text{sum } a \{ 0..m \} \rangle$
using $\langle \text{ereal } (\text{real } k * e) \leq s \rangle \langle s = \text{ereal } (\text{sum } a \{ 0..N \}) \rangle \langle e \leq \text{sum } a \{ \text{Suc } N..m \} \rangle$ **by** *auto*
moreover have $\langle e + k * e = (\text{Suc } k) * e \rangle$
by *(simp add: semiring-normalization-rules(3))*
ultimately have $\langle (\text{Suc } k) * e \leq \text{sum } a \{ 0..m \} \rangle$
by *linarith*
hence $\langle \text{ereal } ((\text{Suc } k) * e) \leq \text{sum } a \{ 0..m \} \rangle$
by *auto*
moreover have $\langle \text{sum } a \{ 0..m \} \in S \rangle$
unfolding *S-def* **by** *blast*
ultimately show *?case* **by** *blast*
qed
hence $\langle \exists s \in S. (\text{real } n) \leq s \rangle$ **for** *n*
by *(meson assms(2) ereal-le-le ex-less-of-nat-mult less-le-not-le)*
hence $\langle \text{Sup } S = \infty \rangle$
using *Sup-le-iff Sup-subset-mono dual-order.strict-trans1 leD less-PInf-Ex-of-nat subsetI*
by *metis*
hence *Sup*: $\langle \text{Sup } ((\text{range } (\lambda n. (\text{sum } a \{ 0..n \})))::\text{ereal set}) = \infty \rangle$ **using** *S-def*
by *blast*
have $\langle \text{incseq } (\lambda n. (\text{sum } a \{ ..<n \})) \rangle$
using $\langle \bigwedge n. a \ n \geq 0 \rangle$ **using** *Extended-Real.incseq-sumI* **by** *auto*
hence $\langle \text{incseq } (\lambda n. (\text{sum } a \{ ..< \text{Suc } n \})) \rangle$
by *(meson incseq-Suc-iff)*
hence $\langle \text{incseq } (\lambda n. (\text{sum } a \{ 0..n \})::\text{ereal}) \rangle$
using *incseq-ereal* **by** *(simp add: atLeast0AtMost lessThan-Suc-atMost)*
hence $\langle (\lambda n. \text{sum } a \{ 0..n \}) \longrightarrow \text{Sup } (\text{range } (\lambda n. (\text{sum } a \{ 0..n \})::\text{ereal})) \rangle$
using *LIMSEQ-SUP* **by** *auto*
thus *?thesis* **using** *Sup PInfy-neq-ereal* **by** *auto*
qed

lemma *sum-Cauchy-positive*:

fixes $a :: \langle - \Rightarrow \text{real} \rangle$

assumes $\langle \bigwedge n. a\ n \geq 0 \rangle$ **and** $\langle \exists K. \forall n. (\text{sum } a\ \{0..n\}) \leq K \rangle$

shows $\langle \text{Cauchy } (\lambda n. \text{sum } a\ \{0..n\}) \rangle$

Explanation: If a series of nonnegative reals is bounded, then the series is Cauchy.

proof (*unfold Cauchy-altdef2, rule, rule*)

fix $e :: \text{real}$

assume $\langle e > 0 \rangle$

have $\langle \exists M. \forall m \geq M. \forall n \geq M. m > n \longrightarrow \text{sum } a\ \{\text{Suc } n..m\} < e \rangle$

proof (*rule classical*)

assume $\langle \neg (\exists M. \forall m \geq M. \forall n \geq M. m > n \longrightarrow \text{sum } a\ \{\text{Suc } n..m\} < e) \rangle$

hence $\langle \forall M. \exists m. \exists n. m \geq M \wedge n \geq M \wedge m > n \wedge \neg (\text{sum } a\ \{\text{Suc } n..m\} < e) \rangle$

by *blast*

hence $\langle \forall M. \exists m. \exists n. m \geq M \wedge n \geq M \wedge m > n \wedge \text{sum } a\ \{\text{Suc } n..m\} \geq e \rangle$

by *fastforce*

hence $\langle (\lambda n. (\text{sum } a\ \{0..n\})) \longrightarrow \infty \rangle$

using *non-Cauchy-unbounded* $\langle 0 < e \rangle$ **assms**(1) **by** *blast*

from $\langle \exists K. \forall n. \text{sum } a\ \{0..n\} \leq K \rangle$

obtain K **where** $\langle \forall n. \text{sum } a\ \{0..n\} \leq K \rangle$

by *blast*

from $\langle (\lambda n. \text{sum } a\ \{0..n\}) \longrightarrow \infty \rangle$

have $\langle \forall B. \exists N. \forall n \geq N. (\lambda n. (\text{sum } a\ \{0..n\}))\ n \geq B \rangle$

using *Lim-PIfty* **by** *simp*

hence $\langle \exists n. (\text{sum } a\ \{0..n\}) \geq K+1 \rangle$

using *ereal-less-eq(3)* **by** *blast*

thus *?thesis* **using** $\langle \forall n. (\text{sum } a\ \{0..n\}) \leq K \rangle$ **by** (*smt (verit, best)*)

qed

have $\langle \text{sum } a\ \{\text{Suc } n..m\} = \text{sum } a\ \{0..m\} - \text{sum } a\ \{0..n\} \rangle$

if $m > n$ **for** $m\ n$

by (*metis add-diff-cancel-left' atLeast0AtMost less-imp-add-positive sum-up-index-split that*)

hence $\langle \exists M. \forall m \geq M. \forall n \geq M. m > n \longrightarrow \text{sum } a\ \{0..m\} - \text{sum } a\ \{0..n\} < e \rangle$

using $\langle \exists M. \forall m \geq M. \forall n \geq M. m > n \longrightarrow \text{sum } a\ \{\text{Suc } n..m\} < e \rangle$ **by** *presburger*

then obtain M **where** $\langle \forall m \geq M. \forall n \geq M. m > n \longrightarrow \text{sum } a\ \{0..m\} - \text{sum } a\ \{0..n\} < e \rangle$

by *blast*

moreover have $\langle m > n \implies \text{sum } a\ \{0..m\} \geq \text{sum } a\ \{0..n\} \rangle$ **for** $m\ n$

using $\langle \bigwedge n. a\ n \geq 0 \rangle$ **by** (*simp add: sum-mono2*)

ultimately have $\langle \exists M. \forall m \geq M. \forall n \geq M. m > n \longrightarrow |\text{sum } a\ \{0..m\} - \text{sum } a\ \{0..n\}| < e \rangle$

by *auto*

hence $\langle \exists M. \forall m \geq M. \forall n \geq M. m \geq n \longrightarrow |\text{sum } a\ \{0..m\} - \text{sum } a\ \{0..n\}| < e \rangle$

by (*metis* $\langle 0 < e \rangle$ *abs-zero cancel-comm-monoid-add-class.diff-cancel diff-is-0-eq'*

less-irrefl-nat linorder-neqE-nat zero-less-diff)

hence $\langle \exists M. \forall m \geq M. \forall n \geq M. |\text{sum } a\ \{0..m\} - \text{sum } a\ \{0..n\}| < e \rangle$

by *(metis abs-minus-commute nat-le-linear)*
hence $\langle \exists M. \forall m \geq M. \forall n \geq M. \text{dist} (\text{sum } a \{0..m\}) (\text{sum } a \{0..n\}) < e \rangle$
 by *(simp add: dist-real-def)*
hence $\langle \exists M. \forall m \geq M. \forall n \geq M. \text{dist} (\text{sum } a \{0..m\}) (\text{sum } a \{0..n\}) < e \rangle$ **by blast**
thus $\langle \exists N. \forall n \geq N. \text{dist} (\text{sum } a \{0..n\}) (\text{sum } a \{0..N\}) < e \rangle$ **by auto**
qed

lemma *convergent-series-Cauchy*:

fixes $a::\langle \text{nat} \Rightarrow \text{real} \rangle$ **and** $\varphi::\langle \text{nat} \Rightarrow 'a::\text{metric-space} \rangle$
assumes $\langle \exists M. \forall n. \text{sum } a \{0..n\} \leq M \rangle$ **and** $\langle \bigwedge n. \text{dist} (\varphi (\text{Suc } n)) (\varphi n) \leq a \ n \rangle$
shows $\langle \text{Cauchy } \varphi \rangle$

Explanation: Let a be a real-valued sequence and let φ be sequence in a metric space. If the partial sums of a are uniformly bounded and the distance between consecutive terms of φ are bounded by the sequence a , then φ is Cauchy.

proof *(unfold Cauchy-altdef2, rule, rule)*

fix $e::\text{real}$
assume $\langle e > 0 \rangle$
have $\langle \bigwedge k. a \ k \geq 0 \rangle$
using $\langle \bigwedge n. \text{dist} (\varphi (\text{Suc } n)) (\varphi n) \leq a \ n \rangle$ *dual-order.trans zero-le-dist* **by blast**
hence $\langle \text{Cauchy} (\lambda k. \text{sum } a \{0..k\}) \rangle$
using $\langle \exists M. \forall n. \text{sum } a \{0..n\} \leq M \rangle$ *sum-Cauchy-positive* **by blast**
hence $\langle \exists M. \forall m \geq M. \forall n \geq M. \text{dist} (\text{sum } a \{0..m\}) (\text{sum } a \{0..n\}) < e \rangle$
unfolding *Cauchy-def* **using** $\langle e > 0 \rangle$ **by blast**
hence $\langle \exists M. \forall m \geq M. \forall n \geq M. m > n \longrightarrow \text{dist} (\text{sum } a \{0..m\}) (\text{sum } a \{0..n\}) < e \rangle$
by blast
have $\langle \text{dist} (\text{sum } a \{0..m\}) (\text{sum } a \{0..n\}) = \text{sum } a \{ \text{Suc } n..m \} \rangle$ **if** $\langle n < m \rangle$ **for**
 $m \ n$
proof –
have $\langle n < \text{Suc } n \rangle$
by simp
have $\langle \text{finite } \{0..n\} \rangle$
by simp
moreover have $\langle \text{finite } \{ \text{Suc } n..m \} \rangle$
by simp
moreover have $\langle \{0..n\} \cup \{ \text{Suc } n..m \} = \{0..m\} \rangle$
using $\langle n < \text{Suc } n \rangle \langle n < m \rangle$ **by auto**
moreover have $\langle \{0..n\} \cap \{ \text{Suc } n..m \} = \{ \} \rangle$
by simp
ultimately have *sum-plus*: $\langle (\text{sum } a \{0..n\}) + \text{sum } a \{ \text{Suc } n..m \} = (\text{sum } a \{0..m\}) \rangle$
by (metis sum.union-disjoint)
have $\langle \text{dist} (\text{sum } a \{0..m\}) (\text{sum } a \{0..n\}) = |(\text{sum } a \{0..m\}) - (\text{sum } a \{0..n\})| \rangle$
using *dist-real-def* **by blast**
moreover have $\langle (\text{sum } a \{0..m\}) - (\text{sum } a \{0..n\}) = \text{sum } a \{ \text{Suc } n..m \} \rangle$
using *sum-plus* **by linarith**

```

ultimately show ?thesis
  by (simp add: ‹ $\bigwedge k. 0 \leq a k$ › sum-nonneg)
qed
hence sum-a: ‹ $\exists M. \forall m \geq M. \forall n \geq M. m > n \longrightarrow \text{sum } a \{ \text{Suc } n..m \} < e$ ›
  by (metis ‹ $\exists M. \forall m \geq M. \forall n \geq M. \text{dist } (\text{sum } a \{ 0..m \}) (\text{sum } a \{ 0..n \}) < e$ ›)
obtain M where ‹ $\forall m \geq M. \forall n \geq M. m > n \longrightarrow \text{sum } a \{ \text{Suc } n..m \} < e$ ›
  using sum-a ‹ $e > 0$ › by blast
hence ‹ $\forall m. \forall n. \text{Suc } m \geq \text{Suc } M \wedge \text{Suc } n \geq \text{Suc } M \wedge \text{Suc } m > \text{Suc } n \longrightarrow \text{sum } a \{ \text{Suc } n.. \text{Suc } m - 1 \} < e$ ›
  by simp
hence ‹ $\forall m \geq 1. \forall n \geq 1. m \geq \text{Suc } M \wedge n \geq \text{Suc } M \wedge m > n \longrightarrow \text{sum } a \{ n..m - 1 \} < e$ ›
  by (metis Suc-le-D)
hence sum-a2: ‹ $\exists M. \forall m \geq M. \forall n \geq M. m > n \longrightarrow \text{sum } a \{ n..m-1 \} < e$ ›
  by (meson add-leE)
have ‹ $\text{dist } (\varphi (n+p+1)) (\varphi n) \leq \text{sum } a \{ n..n+p \}$ › for p n :: nat
  proof (induction p)
    case 0 thus ?case by (simp add: assms(2))
  next
    case (Suc p) thus ?case
      by (smt (verit, ccfv-SIG) Suc-eq-plus1 add-Suc-right add-less-same-cancel1
        assms(2) dist-self dist-triangle2
        gr-implies-not0 sum.cl-ivl-Suc)
  qed
hence ‹ $m > n \implies \text{dist } (\varphi m) (\varphi n) \leq \text{sum } a \{ n..m-1 \}$ › for m n :: nat
  by (metis Suc-eq-plus1 Suc-le-D diff-Suc-1 gr0-implies-Suc less-eq-Suc-le less-imp-Suc-add
    zero-less-Suc)
hence ‹ $\exists M. \forall m \geq M. \forall n \geq M. m > n \longrightarrow \text{dist } (\varphi m) (\varphi n) < e$ ›
  using sum-a2 ‹ $e > 0$ › by (smt (verit))
thus ‹ $\exists N. \forall n \geq N. \text{dist } (\varphi n) (\varphi N) < e$ ›
  using ‹ $0 < e$ › by fastforce
qed

unbundle blinfun-apply-syntax

unbundle no norm-syntax

end

```

2 Banach-Steinhaus theorem

```

theory Banach-Steinhaus
  imports Banach-Steinhaus-Missing
begin

```

We formalize Banach-Steinhaus theorem as theorem *banach-steinhaus*. This theorem was originally proved in Banach-Steinhaus's paper [1]. For the proof, we follow Sokal's approach [3]. Furthermore, we prove as a corollary

a result about pointwise convergent sequences of bounded operators whose domain is a Banach space.

2.1 Preliminaries for Sokal's proof of Banach-Steinhaus theorem

lemma *linear-plus-norm*:

includes *norm-syntax*

assumes $\langle \text{linear } f \rangle$

shows $\langle \|f \xi\| \leq \max \|f (x + \xi)\| \|f (x - \xi)\| \rangle$

Explanation: For arbitrary x and a linear operator f , $\|f \xi\|$ is upper bounded by the maximum of the norms of the shifts of f (i.e., $f (x + \xi)$ and $f (x - \xi)$).

proof –

have $\langle \text{norm } (f \xi) = \text{norm } ((\text{inverse } (\text{of-nat } 2)) *_{\mathbb{R}} (f (x + \xi) - f (x - \xi))) \rangle$

by (*metis* (*no-types*, *opaque-lifting*) *add commute assms diff-diff-eq2 group-cancel.sub1 linear-cmul linear-diff of-nat-numeral real-vector-affinity-eq scaleR-2 scaleR-right-diff-distrib zero-neq-numeral*)

also have $\langle \dots = \text{inverse } (\text{of-nat } 2) * \text{norm } (f (x + \xi) - f (x - \xi)) \rangle$

using *Real-Vector-Spaces.real-normed-vector-class.norm-scaleR* **by** *simp*

also have $\langle \dots \leq \text{inverse } (\text{of-nat } 2) * (\text{norm } (f (x + \xi)) + \text{norm } (f (x - \xi))) \rangle$

by (*simp add: norm-triangle-ineq4*)

also have $\langle \dots \leq \max (\text{norm } (f (x + \xi))) (\text{norm } (f (x - \xi))) \rangle$

by *auto*

finally show *?thesis by blast*

qed

lemma *onorm-Sup-on-ball*:

includes *norm-syntax*

assumes $\langle r > 0 \rangle$

shows $\|f\| \leq \text{Sup } ((\lambda x. \|f *_v x\|) \text{ ` } (\text{ball } x \ r) \text{ `}) / r$

Explanation: Let f be a bounded operator and let x be a point. For any $0 < r$, the operator norm of f is bounded above by the supremum of f applied to the open ball of radius r around x , divided by r .

proof –

have *bdd-above-3*: $\langle \text{bdd-above } ((\lambda x. \|f *_v x\|) \text{ ` } (\text{ball } 0 \ r) \text{ `}) \rangle$

proof –

obtain M **where** $\langle \bigwedge \xi. \|f *_v \xi\| \leq M * \text{norm } \xi \rangle$ **and** $\langle M \geq 0 \rangle$

using *norm-blinfun norm-ge-zero* **by** *blast*

hence $\langle \bigwedge \xi. \xi \in \text{ball } 0 \ r \implies \|f *_v \xi\| \leq M * r \rangle$

using $\langle r > 0 \rangle$ **by** (*smt* (*verit*) *mem-ball-0 mult-left-mono*)

thus *?thesis by* (*meson bdd-aboveI2*)

qed

have *bdd-above-2*: $\langle \text{bdd-above } ((\lambda \xi. \|f *_v (x + \xi)\|) \text{ ` } (\text{ball } 0 \ r) \text{ `}) \rangle$

proof –

have $\langle \text{bdd-above } ((\lambda \xi. \|f *_v x\|) \text{ ` } (\text{ball } 0 \ r) \text{ `}) \rangle$

by *auto*
 moreover have $\langle \text{bdd-above } ((\lambda \xi. \|f *_{\nu} \xi\|) \text{ ' (ball 0 r)}) \rangle$
 using *bdd-above-3* by *blast*
 ultimately have $\langle \text{bdd-above } ((\lambda \xi. \|f *_{\nu} x\| + \|f *_{\nu} \xi\|) \text{ ' (ball 0 r)}) \rangle$
 by (*rule bdd-above-plus*)
 then obtain M where $\langle \bigwedge \xi. \xi \in \text{ball } 0 \text{ r} \implies \|f *_{\nu} x\| + \|f *_{\nu} \xi\| \leq M \rangle$
 unfolding *bdd-above-def* by (*meson image-eqI*)
 moreover have $\langle \|f *_{\nu} (x + \xi)\| \leq \|f *_{\nu} x\| + \|f *_{\nu} \xi\| \text{ for } \xi \rangle$
 by (*simp add: blinfun.add-right norm-triangle-ineq*)
 ultimately have $\langle \bigwedge \xi. \xi \in \text{ball } 0 \text{ r} \implies \|f *_{\nu} (x + \xi)\| \leq M \rangle$
 by (*simp add: blinfun.add-right norm-triangle-le*)
 thus *?thesis* by (*meson bdd-aboveI2*)
 qed
 have *bdd-above-4*: $\langle \text{bdd-above } ((\lambda \xi. \|f *_{\nu} (x - \xi)\|) \text{ ' (ball 0 r)}) \rangle$
 proof –
 obtain K where *K-def*: $\langle \bigwedge \xi. \xi \in \text{ball } 0 \text{ r} \implies \|f *_{\nu} (x + \xi)\| \leq K \rangle$
 using $\langle \text{bdd-above } ((\lambda \xi. \text{norm } (f (x + \xi))) \text{ ' (ball 0 r)}) \rangle$ unfolding
bdd-above-def
 by (*meson image-eqI*)
 have $\langle \xi \in \text{ball } (0::'a) \text{ r} \implies -\xi \in \text{ball } 0 \text{ r} \rangle$ for ξ
 by *auto*
 thus *?thesis* by (*metis K-def ab-group-add-class.ab-diff-conv-add-uminus bdd-aboveI2*)
 qed
 have *bdd-above-1*: $\langle \text{bdd-above } ((\lambda \xi. \max \|f *_{\nu} (x + \xi)\| \|f *_{\nu} (x - \xi)\|) \text{ ' (ball } 0 \text{ r)}) \rangle$
 proof –
 have $\langle \text{bdd-above } ((\lambda \xi. \|f *_{\nu} (x + \xi)\|) \text{ ' (ball 0 r)}) \rangle$
 using *bdd-above-2* by *blast*
 moreover have $\langle \text{bdd-above } ((\lambda \xi. \|f *_{\nu} (x - \xi)\|) \text{ ' (ball 0 r)}) \rangle$
 using *bdd-above-4* by *blast*
 ultimately show *?thesis*
 unfolding *max-def* apply *auto* apply (*meson bdd-above-Int1 bdd-above-mono image-Int-subset*)
 by (*meson bdd-above-Int1 bdd-above-mono image-Int-subset*)
 qed
 have *bdd-above-6*: $\langle \text{bdd-above } ((\lambda t. \|f *_{\nu} t\|) \text{ ' ball } x \text{ r}) \rangle$
 proof –
 have $\langle \text{bounded } (\text{ball } x \text{ r}) \rangle$
 by *simp*
 hence $\langle \text{bounded } ((\lambda t. \|f *_{\nu} t\|) \text{ ' ball } x \text{ r}) \rangle$
 by (*metis (no-types) add.left-neutral bdd-above-2 bdd-above-norm bounded-norm-comp*)

image-add-ball image-image
 thus *?thesis*
 by (*simp add: bounded-imp-bdd-above*)
 qed
 have *norm-1*: $\langle (\lambda \xi. \|f *_{\nu} (x + \xi)\|) \text{ ' ball } 0 \text{ r} = (\lambda t. \|f *_{\nu} t\|) \text{ ' ball } x \text{ r} \rangle$
 by (*metis add.right-neutral ball-translation image-image*)
 have *bdd-above-5*: $\langle \text{bdd-above } ((\lambda \xi. \text{norm } (f (x + \xi))) \text{ ' ball } 0 \text{ r}) \rangle$

by (*simp add: bdd-above-2*)
 have *norm-2*: $\langle \|\xi\| < r \implies \|f *_{\nu} (x - \xi)\| \in (\lambda\xi. \|f *_{\nu} (x + \xi)\|) \text{ 'ball } 0 r \rangle$
 for ξ
 proof –
 assume $\langle \|\xi\| < r \rangle$
 hence $\langle \xi \in \text{ball } (0::'a) r \rangle$
 by *auto*
 hence $\langle -\xi \in \text{ball } (0::'a) r \rangle$
 by *auto*
 thus *?thesis*
 by (*metis (no-types, lifting) ab-group-add-class.ab-diff-conv-add-uminus image-iff*)
 qed
 have *norm-2'*: $\langle \|\xi\| < r \implies \|f *_{\nu} (x + \xi)\| \in (\lambda\xi. \|f *_{\nu} (x - \xi)\|) \text{ 'ball } 0 r \rangle$
 for ξ
 proof –
 assume $\langle \text{norm } \xi < r \rangle$
 hence $\langle \xi \in \text{ball } (0::'a) r \rangle$
 by *auto*
 hence $\langle -\xi \in \text{ball } (0::'a) r \rangle$
 by *auto*
 thus *?thesis*
 by (*metis (no-types, lifting) diff-minus-eq-add image-iff*)
 qed
 have *bdd-above-6*: $\langle \text{bdd-above } ((\lambda\xi. \|f *_{\nu} (x - \xi)\|) \text{ 'ball } 0 r) \rangle$
 by (*simp add: bdd-above-4*)
 have *Sup-2*: $\langle (\text{SUP } \xi \in \text{ball } 0 r. \max \|f *_{\nu} (x + \xi)\| \|f *_{\nu} (x - \xi)\|) = \max (\text{SUP } \xi \in \text{ball } 0 r. \|f *_{\nu} (x + \xi)\|) (\text{SUP } \xi \in \text{ball } 0 r. \|f *_{\nu} (x - \xi)\|) \rangle$
 for ξ
 proof –
 have $\langle \text{ball } (0::'a) r \neq \{\} \rangle$
 using $\langle r > 0 \rangle$ by *auto*
 moreover have $\langle \text{bdd-above } ((\lambda\xi. \|f *_{\nu} (x + \xi)\|) \text{ 'ball } 0 r) \rangle$
 using *bdd-above-5* by *blast*
 moreover have $\langle \text{bdd-above } ((\lambda\xi. \|f *_{\nu} (x - \xi)\|) \text{ 'ball } 0 r) \rangle$
 using *bdd-above-6* by *blast*
 ultimately show *?thesis*
 using *max-Sup*
 by (*metis (mono-tags, lifting) Banach-Steinhaus-Missing.pointwise-max-def image-cong*)
 qed
 have *Sup-3'*: $\langle \|\xi\| < r \implies \|f *_{\nu} (x + \xi)\| \in (\lambda\xi. \|f *_{\nu} (x - \xi)\|) \text{ 'ball } 0 r \rangle$ for $\xi::'a$
 by (*simp add: norm-2'*)
 have *Sup-3''*: $\langle \|\xi\| < r \implies \|f *_{\nu} (x - \xi)\| \in (\lambda\xi. \|f *_{\nu} (x + \xi)\|) \text{ 'ball } 0 r \rangle$ for $\xi::'a$
 by (*simp add: norm-2*)
 have *Sup-3*: $\langle \max (\text{SUP } \xi \in \text{ball } 0 r. \|f *_{\nu} (x + \xi)\|) (\text{SUP } \xi \in \text{ball } 0 r. \|f *_{\nu} (x - \xi)\|) =$

$(SUP \xi \in ball\ 0\ r. \|f *_{\nu} (x + \xi)\|)$
proof–
have $\langle (\lambda \xi. \|f *_{\nu} (x + \xi)\|) ' (ball\ 0\ r) = (\lambda \xi. \|f *_{\nu} (x - \xi)\|) ' (ball\ 0\ r) \rangle$
apply *auto* **using** *Sup-3'* **apply** *auto* **using** *Sup-3''* **by** *blast*
hence $\langle Sup\ ((\lambda \xi. \|f *_{\nu} (x + \xi)\|) ' (ball\ 0\ r)) = Sup\ ((\lambda \xi. \|f *_{\nu} (x - \xi)\|) ' (ball\ 0\ r)) \rangle$
by *simp*
thus *?thesis* **by** *simp*
qed
have *Sup-1*: $\langle Sup\ ((\lambda t. \|f *_{\nu} t\|) ' (ball\ 0\ r)) \leq Sup\ ((\lambda \xi. \|f *_{\nu} \xi\|) ' (ball\ x\ r)) \rangle$
proof–
have $\langle (\lambda t. \|f *_{\nu} t\|) \xi \leq max\ \|f *_{\nu} (x + \xi)\| \|f *_{\nu} (x - \xi)\| \rangle$ **for** ξ
apply (*rule linear-plus-norm*) **apply** (*rule bounded-linear.linear*)
by (*simp add: blinfun.bounded-linear-right*)
moreover **have** $\langle bdd-above\ ((\lambda \xi. max\ \|f *_{\nu} (x + \xi)\| \|f *_{\nu} (x - \xi)\|) ' (ball\ 0\ r)) \rangle$
using *bdd-above-1* **by** *blast*
moreover **have** $\langle ball\ (0::'a)\ r \neq \{\} \rangle$
using $\langle r > 0 \rangle$ **by** *auto*
ultimately **have** $\langle Sup\ ((\lambda t. \|f *_{\nu} t\|) ' (ball\ 0\ r)) \leq Sup\ ((\lambda \xi. max\ \|f *_{\nu} (x + \xi)\| \|f *_{\nu} (x - \xi)\|) ' (ball\ 0\ r)) \rangle$
using *cSUP-mono* **by** (*smt (verit)*)
also **have** $\langle \dots = max\ (Sup\ ((\lambda \xi. \|f *_{\nu} (x + \xi)\|) ' (ball\ 0\ r))) (Sup\ ((\lambda \xi. \|f *_{\nu} (x - \xi)\|) ' (ball\ 0\ r))) \rangle$
using *Sup-2* **by** *blast*
also **have** $\langle \dots = Sup\ ((\lambda \xi. \|f *_{\nu} (x + \xi)\|) ' (ball\ 0\ r)) \rangle$
using *Sup-3* **by** *blast*
also **have** $\langle \dots = Sup\ ((\lambda \xi. \|f *_{\nu} \xi\|) ' (ball\ x\ r)) \rangle$
by (*metis add.right-neutral ball-translation image-image*)
finally **show** *?thesis* **by** *blast*
qed
have $\langle \|f\| = (SUP\ x \in ball\ 0\ r. \|f *_{\nu} x\|) / r \rangle$
using $\langle 0 < r \rangle$ *onorm-r* **by** *blast*
moreover **have** $\langle Sup\ ((\lambda t. \|f *_{\nu} t\|) ' (ball\ 0\ r)) / r \leq Sup\ ((\lambda \xi. \|f *_{\nu} \xi\|) ' (ball\ x\ r)) / r \rangle$
using *Sup-1* $\langle 0 < r \rangle$ *divide-right-mono* **by** *fastforce*
ultimately **have** $\langle \|f\| \leq Sup\ ((\lambda t. \|f *_{\nu} t\|) ' ball\ x\ r) / r \rangle$
by *simp*
thus *?thesis* **by** *simp*
qed

lemma *onorm-Sup-on-ball'*:

includes *norm-syntax*

assumes $\langle r > 0 \rangle$ **and** $\langle \tau < 1 \rangle$

shows $\langle \exists \xi \in ball\ x\ r. \tau * r * \|f\| \leq \|f *_{\nu} \xi\| \rangle$

In the proof of Banach-Steinhaus theorem, we will use this variation of the lemma *onorm-Sup-on-ball*.

Explanation: Let f be a bounded operator, let x be a point and let r be

a positive real number. For any real number $\tau < 1$, there is a point ξ in the open ball of radius r around x such that $\tau * r * \|f\| \leq \|f *_{\nu} \xi\|$.

proof(*cases* $\langle f = 0 \rangle$)
 case *True*
 thus *?thesis* **by** (*metis* *assms*(1) *centre-in-ball* *mult-zero-right* *norm-zero-order-refl* *zero-blinfun.rep-eq*)
next
 case *False*
 have *bdd-above-1*: $\langle \text{bdd-above } ((\lambda t. \|(*_{\nu}) f t\|) \text{ 'ball } x r) \rangle$ **for** $f::\langle 'a \Rightarrow_L 'b \rangle$
 using *assms*(1) *bounded-linear-image* **by** (*simp* *add*: *bounded-linear-image* *blinfun.bounded-linear-right* *bounded-imp-bdd-above* *bounded-norm-comp*)
 have $\langle \text{norm } f > 0 \rangle$
 using $\langle f \neq 0 \rangle$ **by** *auto*
 have $\langle \text{norm } f \leq \text{Sup } ((\lambda \xi. \|(*_{\nu}) f \xi\|) \text{ ' (ball } x r)) / r \rangle$
 using $\langle r > 0 \rangle$ **by** (*simp* *add*: *onorm-Sup-on-ball*)
 hence $\langle r * \text{norm } f \leq \text{Sup } ((\lambda \xi. \|(*_{\nu}) f \xi\|) \text{ ' (ball } x r)) \rangle$
 using $\langle 0 < r \rangle$ **by** (*smt* (*verit*) *divide-strict-right-mono* *nonzero-mult-div-cancel-left*)

 moreover **have** $\langle \tau * r * \text{norm } f < r * \text{norm } f \rangle$
 using $\langle \tau < 1 \rangle$ **using** $\langle 0 < \text{norm } f \rangle$ $\langle 0 < r \rangle$ **by** *auto*
 ultimately **have** $\langle \tau * r * \text{norm } f < \text{Sup } ((\text{norm } \circ ((*__{\nu}) f)) \text{ ' (ball } x r)) \rangle$
 by *simp*
 moreover **have** $\langle (\text{norm } \circ ((*__{\nu}) f)) \text{ ' (ball } x r) \neq \{\} \rangle$
 using $\langle 0 < r \rangle$ **by** *auto*
 moreover **have** $\langle \text{bdd-above } ((\text{norm } \circ ((*__{\nu}) f)) \text{ ' (ball } x r)) \rangle$
 using *bdd-above-1* **apply** *transfer* **by** *simp*
 ultimately **have** $\langle \exists t \in (\text{norm } \circ ((*__{\nu}) f)) \text{ ' (ball } x r). \tau * r * \text{norm } f < t \rangle$
 by (*simp* *add*: *less-cSup-iff*)
 thus *?thesis* **by** (*smt* (*verit*) *comp-def image-iff*)
qed

2.2 Banach-Steinhaus theorem

theorem *banach-steinhaus*:
 fixes $f::\langle 'c \Rightarrow ('a::\text{banach} \Rightarrow_L 'b::\text{real-normed-vector}) \rangle$
 assumes $\langle \bigwedge x. \text{bounded } (\text{range } (\lambda n. (f n) *__{\nu} x)) \rangle$
 shows $\langle \text{bounded } (\text{range } f) \rangle$

This is Banach-Steinhaus Theorem.

Explanation: If a family of bounded operators on a Banach space is pointwise bounded, then it is uniformly bounded.

proof(*rule classical*)
 assume $\langle \neg(\text{bounded } (\text{range } f)) \rangle$
 have *sum-1*: $\langle \exists K. \forall n. \text{sum } (\lambda k. \text{inverse } (\text{real-of-nat } 3^{\wedge} k)) \{0..n\} \leq K \rangle$
 proof–
 have $\langle \text{summable } (\lambda n. \text{inverse } ((3::\text{real})^{\wedge} n)) \rangle$
 by (*simp* *flip*: *power-inverse*)
 hence $\langle \text{bounded } (\text{range } (\lambda n. \text{sum } (\lambda k. \text{inverse } (\text{real } 3^{\wedge} k)) \{0..<n\})) \rangle$

using *summable-imp-sums-bounded* [where $f = (\lambda n. \text{inverse } (\text{real-of-nat } 3^{\wedge} n))$]
lessThan-atLeast0 **by** *auto*
hence $\langle \exists M. \forall h \in (\text{range } (\lambda n. \text{sum } (\lambda k. \text{inverse } (\text{real } 3^{\wedge} k)) \{0..<n\})). \text{norm } h \leq M \rangle$
using *bounded-iff* **by** *blast*
then obtain M **where** $\langle h \in \text{range } (\lambda n. \text{sum } (\lambda k. \text{inverse } (\text{real } 3^{\wedge} k)) \{0..<n\}) \Rightarrow \text{norm } h \leq M \rangle$
for h
by *blast*
have *sum-2*: $\langle \text{sum } (\lambda k. \text{inverse } (\text{real-of-nat } 3^{\wedge} k)) \{0..n\} \leq M \rangle$ **for** n
proof–
have $\langle \text{norm } (\text{sum } (\lambda k. \text{inverse } (\text{real } 3^{\wedge} k)) \{0..< \text{Suc } n\}) \leq M \rangle$
using $\langle \wedge h. h \in (\text{range } (\lambda n. \text{sum } (\lambda k. \text{inverse } (\text{real } 3^{\wedge} k)) \{0..<n\})) \Rightarrow \text{norm } h \leq M \rangle$
by *blast*
hence $\langle \text{norm } (\text{sum } (\lambda k. \text{inverse } (\text{real } 3^{\wedge} k)) \{0..n\}) \leq M \rangle$
by (*simp add: atLeastLessThanSuc-atLeastAtMost*)
hence $\langle \text{sum } (\lambda k. \text{inverse } (\text{real } 3^{\wedge} k)) \{0..n\} \leq M \rangle$
by *auto*
thus *?thesis* **by** *blast*
qed
have $\langle \text{sum } (\lambda k. \text{inverse } (\text{real-of-nat } 3^{\wedge} k)) \{0..n\} \leq M \rangle$ **for** n
using *sum-2* **by** *blast*
thus *?thesis* **by** *blast*
qed
have $\langle \text{of-rat } 2/3 < (1::\text{real}) \rangle$
by *auto*
hence $\langle \forall g::'a \Rightarrow_L 'b. \forall x. \forall r. \exists \xi. g \neq 0 \wedge r > 0 \rightarrow (\xi \in \text{ball } x r \wedge (\text{of-rat } 2/3) * r * \text{norm } g \leq \text{norm } ((*_v) g \xi)) \rangle$
using *onorm-Sup-on-ball'* **by** *blast*
hence $\langle \exists \xi. \forall g::'a \Rightarrow_L 'b. \forall x. \forall r. g \neq 0 \wedge r > 0 \rightarrow ((\xi g x r) \in \text{ball } x r \wedge (\text{of-rat } 2/3) * r * \text{norm } g \leq \text{norm } ((*_v) g (\xi g x r))) \rangle$
by *metis*
then obtain ξ **where** *f1*: $\langle \llbracket g \neq 0; r > 0 \rrbracket \Rightarrow \xi g x r \in \text{ball } x r \wedge (\text{of-rat } 2/3) * r * \text{norm } g \leq \text{norm } ((*_v) g (\xi g x r)) \rangle$
for $g::'a \Rightarrow_L 'b$ **and** x **and** r
by *blast*
have $\langle \forall n. \exists k. \text{norm } (f k) \geq 4^{\wedge} n \rangle$
using $\langle \neg(\text{bounded } (\text{range } f)) \rangle$ **by** (*metis (mono-tags, opaque-lifting) boundedI image-iff linear*)
hence $\langle \exists k. \forall n. \text{norm } (f (k n)) \geq 4^{\wedge} n \rangle$
by *metis*
hence $\langle \exists k. \forall n. \text{norm } ((f \circ k) n) \geq 4^{\wedge} n \rangle$
by *simp*
then obtain k **where** $\langle \text{norm } ((f \circ k) n) \geq 4^{\wedge} n \rangle$ **for** n
by *blast*
define T **where** $\langle T = f \circ k \rangle$
have $\langle T n \in \text{range } f \rangle$ **for** n

unfolding T -def **by** *simp*
have $\langle \text{norm } (T\ n) \geq \text{of-nat } (4^{\wedge}n) \rangle$ **for** n
unfolding T -def **using** $\langle \bigwedge n. \text{norm } ((f \circ k)\ n) \geq 4^{\wedge}n \rangle$ **by** *auto*
hence $\langle T\ n \neq 0 \rangle$ **for** n
by (*smt (verit) T-def* $\langle \bigwedge n. 4^{\wedge}n \leq \text{norm } ((f \circ k)\ n) \rangle$ *norm-zero power-not-zero zero-le-power*)
have $\langle \text{inverse } (\text{of-nat } 3^{\wedge}n) > (0::\text{real}) \rangle$ **for** n
by *auto*
define $y::\langle \text{nat} \Rightarrow 'a \rangle$ **where** $\langle y = \text{rec-nat } 0\ (\lambda n\ x.\ \xi\ (T\ n)\ x\ (\text{inverse } (\text{of-nat } 3^{\wedge}n))) \rangle$
have $\langle y\ (\text{Suc } n) \in \text{ball } (y\ n)\ (\text{inverse } (\text{of-nat } 3^{\wedge}n)) \rangle$ **for** n
using *f1* $\langle \bigwedge n. T\ n \neq 0 \rangle$ $\langle \bigwedge n. \text{inverse } (\text{of-nat } 3^{\wedge}n) > 0 \rangle$ **unfolding** y -def **by** *auto*
hence $\langle \text{norm } (y\ (\text{Suc } n) - y\ n) \leq \text{inverse } (\text{of-nat } 3^{\wedge}n) \rangle$ **for** n
unfolding *ball-def* **apply** *auto* **using** *dist-norm* **by** (*smt (verit) norm-minus-commute*)

moreover **have** $\langle \exists K. \forall n. \text{sum } (\lambda k. \text{inverse } (\text{real-of-nat } 3^{\wedge}k))\ \{0..n\} \leq K \rangle$
using *sum-1* **by** *blast*
moreover **have** $\langle \text{Cauchy } y \rangle$
using *convergent-series-Cauchy* [**where** $a = \lambda n. \text{inverse } (\text{of-nat } 3^{\wedge}n)$ **and** $\varphi = y$] *dist-norm*
by (*metis calculation(1) calculation(2)*)
hence $\langle \exists x. y \longrightarrow x \rangle$
by (*simp add: convergent-eq-Cauchy*)
then obtain x **where** $\langle y \longrightarrow x \rangle$
by *blast*
have *norm-2*: $\langle \text{norm } (x - y\ (\text{Suc } n)) \leq (\text{inverse } (\text{of-nat } 2)) * (\text{inverse } (\text{of-nat } 3^{\wedge}n)) \rangle$ **for** n
proof –
have $\langle \text{inverse } (\text{real-of-nat } 3) < 1 \rangle$
by *simp*
moreover **have** $\langle y\ 0 = 0 \rangle$
using y -def **by** *auto*
ultimately **have** $\langle \text{norm } (x - y\ (\text{Suc } n)) \leq (\text{inverse } (\text{of-nat } 3)) * \text{inverse } (1 - (\text{inverse } (\text{of-nat } 3))) * ((\text{inverse } (\text{of-nat } 3))^{\wedge}n) \rangle$
using *bound-Cauchy-to-lim* [**where** $c = \text{inverse } (\text{of-nat } 3)$ **and** $y = y$ **and** $x = x$]
power-inverse semiring-norm(77) $\langle y \longrightarrow x \rangle$
 $\langle \bigwedge n. \text{norm } (y\ (\text{Suc } n) - y\ n) \leq \text{inverse } (\text{of-nat } 3^{\wedge}n) \rangle$ **by** (*metis divide-inverse*)
moreover **have** $\langle \text{inverse } (\text{real-of-nat } 3) * \text{inverse } (1 - (\text{inverse } (\text{of-nat } 3))) = \text{inverse } (\text{of-nat } 2) \rangle$
by *auto*
ultimately **show** *?thesis*
by (*metis power-inverse*)
qed
have $\langle \text{norm } (x - y\ (\text{Suc } n)) \leq (\text{inverse } (\text{of-nat } 2)) * (\text{inverse } (\text{of-nat } 3^{\wedge}n)) \rangle$ **for** n

using *norm-2* **by** *blast*
have $\langle \exists M. \forall n. \text{norm } ((*_v) (T n) x) \leq M \rangle$
unfolding *T-def* **apply** *auto*
by (*metis* $\langle \bigwedge x. \text{bounded } (\text{range } (\lambda n. (*_v) (f n) x)) \rangle \text{ bounded-iff rangeI}$)
then obtain *M* **where** $\langle \text{norm } ((*_v) (T n) x) \leq M \rangle$ **for** *n*
by *blast*
have *norm-1*: $\langle \text{norm } (T n) * \text{norm } (y (Suc n) - x) + \text{norm } ((*_v) (T n) x)$
 $\leq \text{inverse } (\text{real } 2) * \text{inverse } (\text{real } 3 \wedge n) * \text{norm } (T n) + \text{norm } ((*_v) (T n) x) \rangle$ **for** *n*
proof–
have $\langle \text{norm } (y (Suc n) - x) \leq (\text{inverse } (\text{of-nat } 2)) * (\text{inverse } (\text{of-nat } 3 \wedge n)) \rangle$
using $\langle \text{norm } (x - y (Suc n)) \leq (\text{inverse } (\text{of-nat } 2)) * (\text{inverse } (\text{of-nat } 3 \wedge n)) \rangle$
by (*simp add: norm-minus-commute*)
moreover have $\langle \text{norm } (T n) \geq 0 \rangle$
by *auto*
ultimately have $\langle \text{norm } (T n) * \text{norm } (y (Suc n) - x)$
 $\leq (\text{inverse } (\text{of-nat } 2)) * (\text{inverse } (\text{of-nat } 3 \wedge n)) * \text{norm } (T n) \rangle$
by (*simp add:* $\langle \bigwedge n. T n \neq 0 \rangle$)
thus *?thesis* **by** *simp*
qed
have *inverse-2*: $\langle (\text{inverse } (\text{of-nat } 6)) * \text{inverse } (\text{real } 3 \wedge n) * \text{norm } (T n)$
 $\leq \text{norm } ((*_v) (T n) x) \rangle$ **for** *n*
proof–
have $\langle (\text{of-rat } 2/3) * (\text{inverse } (\text{of-nat } 3 \wedge n)) * \text{norm } (T n) \leq \text{norm } ((*_v) (T n) (y$
 $(Suc n))) \rangle$
using *f1* $\langle \bigwedge n. T n \neq 0 \rangle \langle \bigwedge n. \text{inverse } (\text{of-nat } 3 \wedge n) > 0 \rangle$ **unfolding** *y-def*
by *auto*
also have $\langle \dots = \text{norm } ((*_v) (T n) ((y (Suc n) - x) + x)) \rangle$
by *auto*
also have $\langle \dots = \text{norm } ((*_v) (T n) (y (Suc n) - x) + (*_v) (T n) x) \rangle$
apply *transfer* **apply** *auto* **by** (*metis* *diff-add-cancel linear-simps(1)*)
also have $\langle \dots \leq \text{norm } ((*_v) (T n) (y (Suc n) - x)) + \text{norm } ((*_v) (T n) x) \rangle$
by (*simp add: norm-triangle-ineq*)
also have $\langle \dots \leq \text{norm } (T n) * \text{norm } (y (Suc n) - x) + \text{norm } ((*_v) (T n) x) \rangle$
apply *transfer* **apply** *auto* **using** *onorm* **by** *auto*
also have $\langle \dots \leq (\text{inverse } (\text{of-nat } 2)) * (\text{inverse } (\text{of-nat } 3 \wedge n)) * \text{norm } (T n)$
 $+ \text{norm } ((*_v) (T n) x) \rangle$
using *norm-1* **by** *blast*
finally have $\langle (\text{of-rat } 2/3) * \text{inverse } (\text{real } 3 \wedge n) * \text{norm } (T n)$
 $\leq \text{inverse } (\text{real } 2) * \text{inverse } (\text{real } 3 \wedge n) * \text{norm } (T n)$
 $+ \text{norm } ((*_v) (T n) x) \rangle$
by *blast*
hence $\langle (\text{of-rat } 2/3) * \text{inverse } (\text{real } 3 \wedge n) * \text{norm } (T n)$
 $- \text{inverse } (\text{real } 2) * \text{inverse } (\text{real } 3 \wedge n) * \text{norm } (T n) \leq \text{norm } ((*_v) (T$
 $n) x) \rangle$
by *linarith*
moreover have $\langle (\text{of-rat } 2/3) * \text{inverse } (\text{real } 3 \wedge n) * \text{norm } (T n)$
 $- \text{inverse } (\text{real } 2) * \text{inverse } (\text{real } 3 \wedge n) * \text{norm } (T n)$
 $= (\text{inverse } (\text{of-nat } 6)) * \text{inverse } (\text{real } 3 \wedge n) * \text{norm } (T n) \rangle$

```

    by fastforce
    ultimately show ⟨(inverse (of-nat 6)) * inverse (real 3 ^ n) * norm (T n) ≤
norm ((*v) (T n) x)⟩
    by linarith
  qed
  have inverse-3: ⟨(inverse (of-nat 6)) * (of-rat (4/3) ^ n)
≤ (inverse (of-nat 6)) * inverse (real 3 ^ n) * norm (T n)⟩ for n
  proof-
    have ⟨of-rat (4/3) ^ n = inverse (real 3 ^ n) * (of-nat 4 ^ n)⟩
    apply auto by (metis divide-inverse-commute of-rat-divide power-divide
of-rat-numeral-eq)
    also have ⟨... ≤ inverse (real 3 ^ n) * norm (T n)⟩
    using ⟨∧ n. norm (T n) ≥ of-nat (4 ^ n)⟩ by simp
    finally have ⟨of-rat (4/3) ^ n ≤ inverse (real 3 ^ n) * norm (T n)⟩
    by blast
    moreover have ⟨inverse (of-nat 6) > (0::real)⟩
    by auto
    ultimately show ?thesis by auto
  qed
  have inverse-1: ⟨(inverse (of-nat 6)) * (of-rat (4/3) ^ n) ≤ M⟩ for n
  proof-
    have ⟨(inverse (of-nat 6)) * (of-rat (4/3) ^ n)
≤ (inverse (of-nat 6)) * inverse (real 3 ^ n) * norm (T n)⟩
    using inverse-3 by blast
    also have ⟨... ≤ norm ((*v) (T n) x)⟩
    using inverse-2 by blast
    finally have ⟨(inverse (of-nat 6)) * (of-rat (4/3) ^ n) ≤ norm ((*v) (T n) x)⟩
    by auto
    thus ?thesis using ⟨∧ n. norm ((*v) (T n) x) ≤ M⟩ by (smt (verit))
  qed
  have ⟨∃ n. M < (inverse (of-nat 6)) * (of-rat (4/3) ^ n)⟩
    using Real.real-arch-pow by auto
  moreover have ⟨(inverse (of-nat 6)) * (of-rat (4/3) ^ n) ≤ M⟩ for n
    using inverse-1 by blast
  ultimately show ?thesis by (smt (verit))
qed

```

2.3 A consequence of Banach-Steinhaus theorem

corollary *bounded-linear-limit-bounded-linear*:

```

  fixes f::⟨nat ⇒ ('a::banach ⇒L 'b::real-normed-vector)⟩
  assumes ⟨∧ x. convergent (λ n. (f n) *v x)⟩
  shows ⟨∃ g. (λ n. (*v) (f n)) -pointwise→ (*v) g⟩

```

Explanation: If a sequence of bounded operators on a Banach space converges pointwise, then the limit is also a bounded operator.

proof-

```

  have ⟨∃ l. (λ n. (*v) (f n) x) → l⟩ for x
    by (simp add: ⟨∧ x. convergent (λ n. (*v) (f n) x)⟩ convergentD)

```

hence $\langle \exists F. (\lambda n. (*_v) (f n)) -pointwise \rightarrow F \rangle$
unfolding *pointwise-convergent-to-def* **by** *metis*
obtain F **where** $\langle (\lambda n. (*_v) (f n)) -pointwise \rightarrow F \rangle$
using $\langle \exists F. (\lambda n. (*_v) (f n)) -pointwise \rightarrow F \rangle$ **by** *auto*
have $\langle \bigwedge x. (\lambda n. (*_v) (f n) x) \longrightarrow F x \rangle$
using $\langle (\lambda n. (*_v) (f n)) -pointwise \rightarrow F \rangle$ **apply** *transfer*
by (*simp add: pointwise-convergent-to-def*)
have $\langle bounded (range f) \rangle$
using $\langle \bigwedge x. convergent (\lambda n. (*_v) (f n) x) \rangle$ *banach-steinhaus*
 $\langle \bigwedge x. \exists l. (\lambda n. (*_v) (f n) x) \longrightarrow l \rangle$ *convergent-imp-bounded* **by** *blast*
have *norm-f-n*: $\langle \exists M. \forall n. norm (f n) \leq M \rangle$
unfolding *bounded-def*
by (*meson UNIV-I* $\langle bounded (range f) \rangle$ *bounded-iff image-eqI*)
have $\langle isCont (\lambda t::'b. norm t) y \rangle$ **for** $y::'b$
using *Limits.isCont-norm* **by** *simp*
hence $\langle (\lambda n. norm ((*_v) (f n) x)) \longrightarrow (norm (F x)) \rangle$ **for** x
using $\langle \bigwedge x::'a. (\lambda n. (*_v) (f n) x) \longrightarrow F x \rangle$ **by** (*simp add: tendsto-norm*)
hence *norm-f-n-x*: $\langle \exists M. \forall n. norm ((*_v) (f n) x) \leq M \rangle$ **for** x
using *Elementary-Metric-Spaces.convergent-imp-bounded*
by (*metis UNIV-I* $\langle \bigwedge x::'a. (\lambda n. (*_v) (f n) x) \longrightarrow F x \rangle$ *bounded-iff image-eqI*)
have *norm-f*: $\langle \exists K. \forall n. \forall x. norm ((*_v) (f n) x) \leq norm x * K \rangle$
proof–
have $\langle \exists M. \forall n. norm ((*_v) (f n) x) \leq M \rangle$ **for** x
using *norm-f-n-x* $\langle \bigwedge x. (\lambda n. (*_v) (f n) x) \longrightarrow F x \rangle$ **by** *blast*
hence $\langle \exists M. \forall n. norm (f n) \leq M \rangle$
using *norm-f-n* **by** *simp*
then obtain $M::real$ **where** $\langle \exists M. \forall n. norm (f n) \leq M \rangle$
by *blast*
have $\langle \forall n. \forall x. norm ((*_v) (f n) x) \leq norm x * norm (f n) \rangle$
apply *transfer* **apply** *auto* **by** (*metis mult.commute onorm*)
thus *?thesis* **using** $\langle \exists M. \forall n. norm (f n) \leq M \rangle$
by (*metis (no-types, opaque-lifting) dual-order.trans norm-eq-zero order-refl*
mult-le-cancel-left-pos vector-space-over-itself.scale-zero-left zero-less-norm-iff)
qed
have *norm-F-x*: $\langle \exists K. \forall x. norm (F x) \leq norm x * K \rangle$
proof–
have $\langle \exists K. \forall n. \forall x. norm ((*_v) (f n) x) \leq norm x * K \rangle$
using *norm-f* $\langle \bigwedge x. (\lambda n. (*_v) (f n) x) \longrightarrow F x \rangle$ **by** *auto*
thus *?thesis*
using $\langle \bigwedge x::'a. (\lambda n. (*_v) (f n) x) \longrightarrow F x \rangle$ **apply** *transfer*
by (*metis Lim-bounded tendsto-norm*)
qed
have $\langle linear F \rangle$
proof(*rule linear-limit-linear*)
show $\langle linear ((*_v) (f n)) \rangle$ **for** n
apply *transfer* **apply** *auto* **by** (*simp add: bounded-linear.linear*)
show $\langle f -pointwise \rightarrow F \rangle$
using $\langle (\lambda n. (*_v) (f n)) -pointwise \rightarrow F \rangle$ **by** *auto*

qed
moreover have $\langle \text{bounded-linear-axioms } F \rangle$
using $\text{norm-}F\text{-}x$ **by** $(\text{simp add: } \langle \bigwedge x. (\lambda n. (*_v) (f n) x) \longrightarrow F x \rangle \text{ bounded-linear-axioms-def})$

ultimately have $\langle \text{bounded-linear } F \rangle$
unfolding $\text{bounded-linear-def}$ **by** blast
hence $\langle \exists g. (*_v) g = F \rangle$
using $\text{bounded-linear-Blinfun-apply}$ **by** auto
thus $?thesis$ **using** $\langle (\lambda n. (*_v) (f n)) \text{--pointwise} \rightarrow F \rangle$ **apply transfer by auto**
qed

end

References

- [1] S. Banach and H. Steinhaus. Sur le principe de la condensation de singularités. *Fundamenta Mathematicae*, 1(9):50–61, 1927.
- [2] M. S. Moslehian and E. W. Weisstein. Uniform boundedness principle. *From MathWorld—A Wolfram Web Resource*. <http://mathworld.wolfram.com/UniformBoundednessPrinciple.html>.
- [3] A. D. Sokal. A really simple elementary proof of the uniform boundedness theorem. *The American Mathematical Monthly*, 118(5):450–452, 2011.