The Hahn-Banach Theorem
for Real Vector Spaces

Gertrud Bauer
http://www.in.tum.de/~bauerg/

May 25, 2015

Abstract
The Hahn-Banach Theorem is one of the most fundamental results in
functional analysis. We present a fully formal proof of two versions of the
theorem, one for general linear spaces and another for normed spaces. This
development is based on simply-typed classical set-theory, as provided by
Isabelle/HOL.

Contents
1 Preface 3
I Basic Notions 4
  2 Bounds 4
  3 Vector spaces 5
    3.1 Signature .......................... 5
    3.2 Vector space laws .................. 5
  4 Subspaces 11
    4.1 Definition .......................... 12
    4.2 Linear closure ........................ 14
    4.3 Sum of two vectorspaces .......... 15
    4.4 Direct sums ........................ 17
  5 Normed vector spaces 20
    5.1 Quasinorms .......................... 20
    5.2 Norms ............................... 21
    5.3 Normed vector spaces ............. 21
  6 Linearforms 22
CONTENTS

7 An order on functions 23
   7.1 The graph of a function 23
   7.2 Functions ordered by domain extension 23
   7.3 Domain and function of a graph 24
   7.4 Norm-preserving extensions of a function 24

8 The norm of a function 25
   8.1 Continuous linear forms 25
   8.2 The norm of a linear form 26

9 Zorn’s Lemma 30

II Lemmas for the Proof 32

10 The supremum w.r.t. the function order 32

11 Extending non-maximal functions 39

III The Main Proof 45

12 The Hahn-Banach Theorem 45
   12.1 The Hahn-Banach Theorem for vector spaces 45
   12.2 Alternative formulation 50
   12.3 The Hahn-Banach Theorem for normed spaces 50
1 Preface

This is a fully formal proof of the Hahn-Banach Theorem. It closely follows the informal presentation given in Heuser’s textbook [1, § 36]. Another formal proof of the same theorem has been done in Mizar [3]. A general overview of the relevance and history of the Hahn-Banach Theorem is given by Narici and Beckenstein [2].

The document is structured as follows. The first part contains definitions of basic notions of linear algebra: vector spaces, subspaces, normed spaces, continuous linear-forms, norm of functions and an order on functions by domain extension. The second part contains some lemmas about the supremum (w.r.t. the function order) and extension of non-maximal functions. With these preliminaries, the main proof of the theorem (in its two versions) is conducted in the third part. The dependencies of individual theories are as follows.
Part I
Basic Notions

2 Bounds

theory Bounds
imports Main ../../src/HOL/Library/ContNotDenum
begin

locale lub =
fixes A and x
assumes least [intro?]: (∀a. a ∈ A ⇒ a ≤ b) ⇒ x ≤ b
and upper [intro?]: a ∈ A ⇒ a ≤ x

lemmas [elim?] = lub.least lub.upper

definition the-lub :: 'a::order set ⇒ 'a (⨆ A) where the-lub A = The (lub A)

lemma the-lub-equality [elim?):
assumes lub A x
shows ∨ A = (∀a::'a::order)
proof –
interpret lub A x by fact
show ?thesis
proof (unfold the-lub-def)
  from lub A x show The (lub A) = x
  proof
    fix x' assume lub': lub A x'
    show x' = x
  proof (rule order-antisym)
    from lub' show x' ≤ x
    proof
      fix a assume a ∈ A
      then show a ≤ x ..
    qed
    show x ≤ x'
    proof
      fix a assume a ∈ A
      with lub' show a ≤ x' ..
    qed
  qed
  qed
  qed

lemma the-lubI-ex:
assumes ex: ∃x. lub A x
shows lub A (∨ A)
proof –
from ex obtain x where x: lub A x ..
also from x have [symmetric]: ∨ A = x ..
finally show \(?thesis\).

qed

lemma real-complete: \(\exists a::real. \ a \in A \implies \exists y. \ \forall a \in A. \ a \leq y \implies \exists x. \ lub A x\)

by (intro exI[of - Sup A]) (auto intro: cSup-upper cSup-least simp: lub-def)

end

3 Vector spaces

theory Vector-Space

imports Complex-Main Bounds

begin

3.1 Signature

For the definition of real vector spaces a type \(\textquote{'}a\) of the sort \(\{\text{plus, minus, zero}\}\) is considered, on which a real scalar multiplication \(\cdot\) is declared.

consts

\(\text{prod} :: real \Rightarrow 'a :: \{\text{plus, minus, zero}\} \Rightarrow 'a\) (infixr \(\cdot\) 70)

3.2 Vector space laws

A vector space is a non-empty set \(V\) of elements from \(\textquote{'}a\) with the following vector space laws: The set \(V\) is closed under addition and scalar multiplication, addition is associative and commutative; \(-(x)\) is the inverse of \(x\) w. r. t. addition and \(0\) is the neutral element of addition. Addition and multiplication are distributive; scalar multiplication is associative and the real number \(1\) is the neutral element of scalar multiplication.

locale vectorspace =

fixes \(V\)

assumes non-empty [iff, intro?]: \(V \neq \{\}\)

and add-closed [iff]: \(x \in V \implies y \in V \implies x + y \in V\)

and mult-closed [iff]: \(x \in V \implies a \cdot x \in V\)

and add-assoc: \(x \in V \implies y \in V \implies z \in V \implies (x + y) + z = x + (y + z)\)

and add-commute: \(x \in V \implies y \in V \implies x + y = y + x\)

and diff-self [simp]: \(x \in V \implies x - x = 0\)

and add-zero-left [simp]: \(x \in V \implies 0 + x = x\)

and add-mult-distrib1: \(x \in V \implies y \in V \implies a \cdot (x + y) = a \cdot x + a \cdot y\)

and add-mult-distrib2: \(x \in V \implies (a + b) \cdot x = a \cdot x + b \cdot x\)

and mult-assoc: \(x \in V \implies (a \cdot b) \cdot x = a \cdot (b \cdot x)\)

and mult-1 [simp]: \(x \in V \implies 1 \cdot x = x\)

and negate-eq1: \(x \in V \implies - x = (- 1) \cdot x\)

and diff-eq1: \(x \in V \implies y \in V \implies x - y = x + -y\)

begin

lemma negate-eq2: \(x \in V \implies (- 1) \cdot x = -x\)

by (rule negate-eq1 [symmetric])

lemma negate-eq2a: \(x \in V \implies -1 \cdot x = -x\)

by (simp add: negate-eq1)
**lemma** diff-eq2: \( x \in V \implies y \in V \implies x + y = x - y \)
by (rule diff-eq1 [symmetric])

**lemma** diff-closed [iff]: \( x \in V \implies y \in V \implies x - y \in V \)
by (simp add: diff-eq1 negate-eq1)

**lemma** neg-closed [iff]: \( x \in V \implies -x \in V \)
by (simp add: negate-eq1)

**lemma** add-left-commute: \( x \in V \implies y \in V \implies z \in V \implies x + (y + z) = y + (x + z) \)
proof
assumption xyz: \( x \in V \ y \in V \ z \in V \)
then have \( x + (y + z) = (x + y) + z \)
by (simp only: add-assoc)
also from xyz have \( \ldots = (y + x) + z \)
by (simp only: add-commute)
also from xyz have \( \ldots = y + (x + z) \)
by (simp only: add-assoc)
finally show ?thesis .
qed

**theorems** add-ac = add-assoc add-commute add-left-commute

The existence of the zero element of a vector space follows from the non-emptiness of carrier set.

**lemma** zero [iff]: 0 \( \in V \)
proof
assumption obtain x where x: \( x \in V \)
then have \( 0 = x - x \)
by (rule diff-self [symmetric])
also from x have \( \ldots = (y + x) + z \)
by (simp only: add-commute)
also from x have \( \ldots = y + (x + z) \)
by (simp only: add-assoc)
finally show ?thesis .
qed

**lemma** add-zero-right [simp]: \( x \in V \implies x + 0 = x \)
proof
assumption x: \( x \in V \)
from this and zero have \( x + 0 = 0 + x \)
by (rule add-commute)
also from x have \( \ldots = x \)
by (rule add-zero-left)
finally show ?thesis .
qed

**lemma** mult-assoc2: \( a \cdot b \cdot x = (a \ast b) \cdot x \)
by (simp only: mult-assoc)

**lemma** diff-mult-distrib1: \( x \in V \implies y \in V \implies a \cdot (x - y) = a \cdot x - a \cdot y \)
by (simp add: diff-eq1 negate-eq1 add-mult-distrib1 mult-assoc2)

**lemma** diff-mult-distrib2: \( x \in V \implies (a - b) \cdot x = a \cdot x - (b \cdot x) \)
proof
assumption x: \( x \in V \)
have \( (a - b) \cdot x = (a + - b) \cdot x \)
by simp
also from x have \( \ldots = a \cdot x + (- b) \cdot x \)
by (rule add-mult-distrib2)
also from $x$ have \ldots $a \cdot x + - (b \cdot x)$
by (simp add: negate-eq1 mult-assoc2)
also from $x$ have \ldots $a \cdot x - (b \cdot x)$
by (simp add: diff-eq1)
finally show \textit{?thesis}.
qed

lemmas distrib =
add-mult-distrib1 add-mult-distrib2
diff-mult-distrib1 diff-mult-distrib2

Further derived laws:

\textbf{lemma} mult-zero-left \textit{[simp]}: $x \in V \implies 0 \cdot x = 0$
\textbf{proof} –
\textbf{assume} $x$: $x \in V$
\textbf{have} $0 \cdot x = (1 - 1) \cdot x$ \textbf{by simp}
\textbf{also have} \ldots $= (1 + - 1) \cdot x$ \textbf{by simp}
\textbf{also from} $x$ \textbf{have} \ldots $= 1 \cdot x + (- 1) \cdot x$
\textbf{by (rule add-mult-distrib2)}
\textbf{also from} $x$ \textbf{have} \ldots $= x + (- 1) \cdot x$ \textbf{by simp}
\textbf{also from} $x$ \textbf{have} \ldots $= x + - x$ \textbf{by (simp add: negate-eq2a)}
\textbf{also from} $x$ \textbf{have} \ldots $= x - x$ \textbf{by (simp add: diff-eq2)}
\textbf{also from} $x$ \textbf{have} \ldots $= 0$ \textbf{by simp}
\textbf{finally show} \textit{?thesis}.
qed

\textbf{lemma} mult-zero-right \textit{[simp]}: $a \cdot 0 = (0::a)$
\textbf{proof} –
\textbf{have} $a \cdot 0 = a \cdot (0 - (0::a))$ \textbf{by simp}
\textbf{also have} \ldots $= a \cdot 0 - a \cdot 0$
\textbf{by (rule diff-mult-distrib1) simp-all}
\textbf{also have} \ldots $= 0$ \textbf{by simp}
\textbf{finally show} \textit{?thesis}.
qed

\textbf{lemma} minus-mult-cancel \textit{[simp]}: $x \in V \implies (- a) \cdot - x = a \cdot x$
\textbf{by (simp add: negate-eq1 mult-assoc2)}

\textbf{lemma} add-minus-left-eq-diff: $x \in V \implies y \in V \implies - x + y = y - x$
\textbf{proof} –
\textbf{assume} $xy$: $x \in V \ y \in V$
\textbf{then have} $- x + y = y - x$ \textbf{by (simp add: add-commute)}
\textbf{also from} $xy$ \textbf{have} \ldots $= y - x$ \textbf{by (simp add: diff-eq1)}
\textbf{finally show} \textit{?thesis}.
qed

\textbf{lemma} add-minus \textit{[simp]}: $x \in V \implies x + - x = 0$
\textbf{by (simp add: diff-eq2)}

\textbf{lemma} add-minus-left \textit{[simp]}: $x \in V \implies - x + x = 0$
\textbf{by (simp add: diff-eq2 add-commute)}

\textbf{lemma} minus-minus \textit{[simp]}: $x \in V \implies - (- x) = x$
\textbf{by (simp add: negate-eq1 mult-assoc2)}
lemma minus-zero [simp]: $-(0::'a) = 0$
  by (simp add: negate-eq1)

lemma minus-zero-iff [simp]:
  assumes $x: x \in V$
  shows $(-x = 0) = (x = 0)$
proof
  from $x$ have $x = -(-x)$ by simp
  also assume $-x = 0$
  also have $\ldots = 0$ by (rule minus-zero)
  finally show $x = 0$.
next
  assume $x = 0$
  then show $-x = 0$ by simp
qed

lemma add-minus-cancel [simp]: $x \in V \Longrightarrow y \in V \Longrightarrow x + (-x + y) = y$
  by (simp add: addassoc [symmetric])

lemma minus-add-cancel [simp]: $x \in V \Longrightarrow y \in V \Longrightarrow -x + (x + y) = y$
  by (simp add: addassoc [symmetric])

lemma minus-add-distrib [simp]: $x \in V \Longrightarrow y \in V \Longrightarrow -(x+y) = -x - y$
  by (simp add: negate-eq1 add-mult-distrib1)

lemma diff-zero [simp]: $x \in V \Longrightarrow x - 0 = x$
  by (simp add: diff-eq1)

lemma diff-zero-right [simp]: $x \in V \Longrightarrow 0 - x = -x$
  by (simp add: diff-eq1)

lemma add-left-cancel:
  assumes $x: x \in V$ and $y: y \in V$ and $z: z \in V$
  shows $(x + y = x + z) = (y = z)$
proof
  from $y$ have $y = 0 + y$ by simp
  also from $x$ have $\ldots = (-x + x) + y$ by simp
  also from $x$ have $\ldots = -x + (x + y)$ by (simp add: addassoc)
  also assume $x + y = x + z$
  also from $z$ have $-x + (x + z) = -x + x + z$ by (simp add: addassoc)
  also from $z$ have $\ldots = z$ by simp
  finally show $y = z$.
next
  assume $y = z$
  then show $x + y = x + z$ by (simp only:)
qed

lemma add-right-cancel: $x \in V \Longrightarrow y \in V \Longrightarrow z \in V \Longrightarrow (y + x = z + x) = (y = z)$
  by (simp only: add-commute add-left-cancel)

lemma add-assoc-cong:
  $x \in V \Longrightarrow y \in V \Longrightarrow x' \in V \Longrightarrow y' \in V \Longrightarrow z \in V$
3.2 Vector space laws

\[ x + y = x' + y' \Rightarrow x + (y + z) = x' + (y' + z) \]
by (simp only: add-assoc [symmetric])

**Lemma:** mult-left-commute: \( x \in V \implies a \cdot b \cdot x = b \cdot a \cdot x \)
by (simp add: mult.commute mult-assoc2)

**Lemma:** mult-zero-uniq:
assumes \( x: \ x \in V \ x \neq 0 \) and \( ax: \ a \cdot x = 0 \)
shows \( a = 0 \)
proof (rule classical)
assume \( a: \ a \neq 0 \)
from \( x \ a \ have \ x = (inverse a * a) \cdot x \ by \ simp \)
also from \( \langle x \in V \rangle \ have \ldots = inverse a \cdot (a \cdot x) \ by \ (rule \ mult-assoc) \)
also from \( ax \ have \ldots = 0 \ by \ simp \)
also have \( \ldots = 0 \ by \ simp \)
finally have \( x = 0 \).
with \( \langle x \neq 0 \rangle \) show \( a = 0 \) by contradiction
qed

**Lemma:** mult-left-cancel:
assumes \( x: \ x \in V \) and \( y: \ y \in V \) and \( a: \ a \neq 0 \)
shows \( (a \cdot x = a \cdot y) = (x = y) \)
proof
from \( x \ have \ x = 1 \cdot x \ by \ simp \)
also from \( a \ have \ldots = (inverse a * a) \cdot x \ by \ simp \)
also from \( x \ have \ldots = inverse a \cdot (a \cdot x) \)
by (simp only: mult-assoc)
also assume \( a \cdot x = a \cdot y \)
also from \( a y \ have \ inverse a \cdot \ldots = y \)
by (simp add: mult-assoc2)
finally show \( x = y \).
next
assume \( x = y \)
then show \( a \cdot x = a \cdot y \) by (simp only:)
qed

**Lemma:** mult-right-cancel:
assumes \( x: \ x \in V \) and \( neq: \ x \neq 0 \)
shows \( (a \cdot x = b \cdot x) = (a = b) \)
proof
from \( x \ have \ (a - b) \cdot x = a \cdot x - b \cdot x \)
by (simp add: diff-mult-distrib2)
also assume \( a \cdot x = b \cdot x \)
with \( x \ have \ a \cdot x - b \cdot x = 0 \ by \ simp \)
finally have \( (a - b) \cdot x = 0 \).
with \( x \ neq \ have \ a - b = 0 \ by \ (rule \ mult-zero-uniq) \)
then show \( a = b \) by simp
next
assume \( a = b \)
then show \( a \cdot x = b \cdot x \) by (simp only:)
qed

**Lemma:** eq-diff-eq:
assumes \( x: \ x \in V \) and \( y: \ y \in V \) and \( z: \ z \in V \)
shows \((x = z - y) = (x + y = z)\)

proof
assume \(x = z - y\)
then have \(x + y = z - y + y\) by simp
also from \(y\) have \(\ldots = z - y + y\)
  by (simp add: diff-eq1)
also have \(\ldots = z + (-y + y)\)
  by (rule add-assoc) (simp-all add: \(y\)
also from \(y\) have \(\ldots = z + 0\)
  by (simp only: add-minus-left)
also from \(z\) have \(\ldots = z\)
  by (simp only: add-zero-right)
finally show \(x + y = z\).

next
assume \(x + y = z\)
then have \(z - y = (x + y) - y\) by simp
also from \(x\) have \(\ldots = x + y + y\)
  by (simp add: diff-eq1)
also have \(\ldots = x + (y + y)\)
  by (rule add-assoc) (simp-all add: \(x\)
also from \(x\) have \(\ldots = x\) by simp
finally show \(x - y\).

qed

lemma add-minus-eq-minus:
assumes \(x: x \in V\) and \(y: y \in V\) and \(xy: x + y = 0\)
shows \(x = -y\)
proof
  from \(x\) have \(eq: x + (-y + y) + x\) by simp
  also from \(x\) have \(\ldots = -y + (x + y)\)
    by (simp add: add-ac)
  also note \(xy\)
  also from \(y\) have \(-y + 0 = -y\) by simp
  finally show \(x = -y\).

qed

lemma add-minus-eq:
assumes \(x: x \in V\) and \(y: y \in V\) and \(xy: x - y = 0\)
shows \(x = y\)
proof
  from \(x\) and \(y\) have \(eq: x + (-y) = 0\)
    by (simp add: diff-eq1)
  with \(-y\) have \(x = (-y)\)
    by (rule add-minus-eq-minus) (simp-all add: \(x\)
  with \(x\) show \(x = y\) by simp

qed

lemma add-diff-swap:
assumes \(vs: a \in V\) \(b \in V\) \(c \in V\) \(d \in V\)
  and eq: \(a + b = c + d\)
shows \(a - c = d - b\)
proof
  from \(assms\) have \(-c + (a + b) = -c + (c + d)\)
    by (simp add: add-left-cancel)
  also have \(\ldots = d\)
    using \(c \in V\) \(d \in V\) by (rule minus-add-cancel)
  finally have \(eq\)
    by (rule minus-add-cancel)

from vs have $a - c = (-c + (a + b)) + -b$
  by (simp add: add-ac diff-eq1)
also from vs eq have $\ldots = d + -b$
  by (simp add: add-right-cancel)
also from vs have $\ldots = d - b$ by (simp add: diff-eq2)
finally show $a - c = d - b$.
qed

lemma vs-add-cancel-21:
  assumes vs: $x \in V \ y \in V \ z \in V \ u \in V$
  shows $(x + (y + z)) = y + (u + x)$
proof
  from vs have $x + z = -y + y + (x + z)$ by simp
  also have $\ldots = -y + (y + (x + z))$
    by (rule add-assoc) (simp-all add: vs)
  also from vs have $y + (x + z) = x + (y + z)$
    by (simp add: add-ac)
  also assume $x + (y + z) = y + u$
  also from vs have $u - y + (y + u) = -z$
    by simp
  finally show $x + z = u$.
next
  assume $x + z = u$
  with vs show $x + (y + z) = y + u$
    by (simp only: add-left-commute [of z])
qed

lemma add-cancel-end:
  assumes vs: $x \in V \ y \in V \ z \in V$
  shows $(x + (y + z)) = (x = -z)$
proof
  assume $x + (y + z) = y$
  with vs have $(x + z) + y = 0 + y$ by (simp add: add-ac)
  with vs have $x + z = 0$
    by (simp only: add-right-cancel add-closed zero)
  with vs show $x = -z$
    by (simp add: add-minus-eq-minus)
next
  assume eq: $x = -z$
  then have $x + (y + z) = -z + (y + z)$ by simp
  also have $\ldots = y + (-z + z)$
    by (rule add-left-commute) (simp-all add: vs)
  also from vs have $\ldots = y$
    by simp
  finally show $x + (y + z) = y$.
qed
end

4 Subspaces

theory Subspace
  imports Vector-Space ~~/src/HOL/Library/Set-Algebras
begin
4.1 Definition

A non-empty subset $U$ of a vector space $V$ is a subspace of $V$, iff $U$ is closed under addition and scalar multiplication.

locale subspace =  
  fixes $U :: a :: \{\text{minus}, \text{plus}, \text{zero}, \text{uminus}\}$ set and $V$  
  assumes non-empty [iff, intro]: $U \not= \{\}$  
  and subset [iff]: $U \subseteq V$  
  and add-closed [iff]: $x \in U \Longrightarrow y \in U \Longrightarrow x + y \in U$  
  and mult-closed [iff]: $x \in U \Longrightarrow a \cdot x \in U$

notation (symbols)  
  subspace (infix $\subseteq$ 50)

declare vectorspace.intro [intro?] subspace.intro [intro?]

lemma subspace-subset [elim]: $U \subseteq V \Longrightarrow U \subseteq V$  
  by (rule subspace.subset)

lemma (in subspace) subsetD [iff]: $x \in U \Longrightarrow x \in V$  
  using subset by blast

lemma subspaceD [elim]: $U \subseteq V \Longrightarrow x \in U \Longrightarrow x \in V$  
  by (rule subspace.subsetD)

lemma rev-subspaceD [elim?]: $x \in U \Longrightarrow U \subseteq V \Longrightarrow x \in V$  
  by (rule subspace.subsetD)

lemma (in subspace) diff-closed [iff]:  
  assumes vectorspace $V$  
  assumes $x \in U$ and $y \in U$  
  shows $x - y \in U$  
  proof –  
  interpret vectorspace $V$ by fact  
  from $x \ y$ show ?thesis by (simp add: diff-eq1 negate-eq1)  
  qed

Similar as for linear spaces, the existence of the zero element in every subspace follows from the non-emptiness of the carrier set and by vector space laws.

lemma (in subspace) zero [intro]:  
  assumes vectorspace $V$  
  shows $0 \in U$  
  proof –  
  interpret $V$: vectorspace $V$ by fact  
  have $U \not= \{}$ by (rule non-empty)  
  then obtain $x$ where $x \in U$ by blast  
  then have $x \in V$ .. then have $0 = x - x$ by simp  
  also from vectorspace $V \ x \ x$ have \ldots $\in U$ by (rule diff-closed)  
  finally show ?thesis .  
  qed

lemma (in subspace) neg-closed [iff]:  
  assumes vectorspace $V$
4.1 Definition

assumes $x \in U$
shows $- x \in U$
proof
  interpret $vectorspace \ V$ by fact
  from $x$ show $?thesis$ by (simp add: negate-eq1)
qed

Further derived laws: every subspace is a vector space.

lemma (in $vectorspace$) $subspace$-refl [intro]:
assumes $vectorspace \ V$
shows $vectorspace \ U$
proof
  interpret $vectorspace \ V$ by fact
  show $?thesis$
  proof
    show $U \neq \{\}$ ..
    fix $x \ y \ z$ assume $x: x \in U$ and $y: y \in U$ and $z: z \in U$
    fix $a \ b :: real$
    from $x \ y$ show $x + y \in U$ by simp
    from $x \ show \ a \cdot x \in U$ by simp
    from $x \ y \ z$ show $(x + y) + z = x + (y + z)$ by (simp add: add-ac)
    from $x \ y$ show $x + y = y + x$ by (simp add: add-ac)
    from $x \ show \ x - x = 0$ by simp
    from $x \ show \ 0 + x = x$ by simp
    from $x \ y$ show $a \cdot (x + y) = a \cdot x + a \cdot y$ by (simp add: distrib)
    from $x \ show \ (a + b) \cdot x = a \cdot x + b \cdot x$ by (simp add: distrib)
    from $x \ show \ (a \cdot b) \cdot x = a \cdot b \cdot x$ by (simp add: mult-assoc)
    from $x \ show \ 1 \cdot x = x$ by simp
    from $x \ show \ - x = -1 \cdot x$ by (simp add: negate-eq1)
    from $x \ y$ show $x - y = x + (-y)$ by (simp add: diff-eq1)
  qed
qed

The subspace relation is reflexive.

lemma (in $vectorspace$) $subspace$-trans [trans]:
$U \subseteq V \Longrightarrow V \subseteq W \Longrightarrow U \subseteq W$
proof
  assume $uv: U \subseteq V$ and $vw: V \subseteq W$
  from $uv$ show $U \neq \{\}$ by (rule $subspace$.non-empty)
  show $U \subseteq W$
  proof –
from $uv$ have $U \subseteq V$ by (rule subspace.subset)
also from $uv$ have $V \subseteq W$ by (rule subspace.subset)
finally show \( \text{thesis} \).
qed

fix $x, y$ assume $x : x \in U$ and $y : y \in U$
from $uv$ and $x y$ show $x + y \in U$ by (rule subspace.add-closed)
from $uv$ and $x$ show $\bigwedge a. \ a \cdot x \in U$ by (rule subspace.mult-closed)
qed

4.2 Linear closure

The linear closure of a vector $x$ is the set of all scalar multiples of $x$.

definition lin :: (′a::{minus, plus, zero}) ⇒ ′a set
  where lin $x$ = \{ $a \cdot x | a. \ True \}

lemma linI [intro]: $y = a \cdot x \Longrightarrow y \in lin x$
  unfolding lin-def by blast

lemma linI′ [iff]: $a \cdot x \in lin x$
  unfolding lin-def by blast

lemma linE [elim]: $x \in lin v \Longrightarrow (\bigwedge a::real. \ x = a \cdot v \Longrightarrow C) \Longrightarrow C$
  unfolding lin-def by blast

Every vector is contained in its linear closure.

lemma (in vectorspace) x-lin-x [iff]: $x \in V \Longrightarrow x \in lin x$
proof –
  assume $x \in V$
  then have $x = 1 \cdot x$ by simp
  also have \ldots \in lin $x$ ..
  finally show \( \text{thesis} \).
qed

lemma (in vectorspace) 0-lin-x [iff]: $x \in V \Longrightarrow 0 \in lin x$
proof
  assume $x \in V$
  then show $0 = 0 \cdot x$ by simp
qed

Any linear closure is a subspace.

lemma (in vectorspace) lin-subspace [intro]:
  assumes $x: x \in V$
  shows $lin x \subseteq V$
proof
  from $x$ show $lin x \neq \{ \}$ by auto
next
  show $lin x \subseteq V$
proof
    fix $x'$ assume $x' \in lin x$
    then obtain $a$ where $x' = a \cdot x$ ..
    with $x$ show $x' \in V$ by simp
  qed
next
4.3 Sum of two vectorspaces

fix $x'$ $x''$ assume $x': x' \in \text{lin } x$ and $x'': x'' \in \text{lin } x$
show $x' + x'' \in \text{lin } x$
proof
  from $x'$ obtain $a'$ where $x' = a' \cdot x$.
moreover from $x''$ obtain $a''$ where $x'' = a'' \cdot x$.
ultimately have $x' + x'' = (a' + a'') \cdot x$
using $x$ by (simp add: distrib)
also have $\ldots \in \text{lin } x$.
finally show ?thesis.
qed

fix $a :: \text{real}$
show $a \cdot x' \in \text{lin } x$
proof
  from $x'$ obtain $a'$ where $x' = a' \cdot x$.
  with $x$ have $a \cdot x' = (a * a') \cdot x$ by (simp add: mult-assoc)
also have $\ldots \in \text{lin } x$.
finally show ?thesis.
qed

Any linear closure is a vector space.

lemma (in vectorspace) lin-vectorspace [intro]:
assumes $x \in V$
shows vectorspace (lin $x$)
proof
  from $(x \in V)$ have subspace (lin $x$) $V$
  by (rule lin-subspace)
  from this and vectorspace-axioms show ?thesis
  by (rule subspace.vectorspace)
qed

4.3 Sum of two vectorspaces

The sum of two vectorspaces $U$ and $V$ is the set of all sums of elements from $U$ and $V$.

lemma sum-def: $U + V = \{ u + v \mid u, v. u \in U \land v \in V \}$
unfolding set-plus-def by auto

lemma sumE [elim]:
$x \in U + V \Rightarrow (\bigwedge u v. x = u + v \Rightarrow u \in U \Rightarrow v \in V \Rightarrow C) \Rightarrow C$
unfolding sum-def by blast

lemma sumI [intro]:
$u \in U \Rightarrow v \in V \Rightarrow x = u + v \Rightarrow x \in U + V$
unfolding sum-def by blast

lemma sumI' [intro]:
$u \in U \Rightarrow v \in V \Rightarrow u + v \in U + V$
unfolding sum-def by blast

$U$ is a subspace of $U + V$.

lemma subspace-sum1 [iff]:
assumes vectorspace $U$ vectorspace $V$
shows $U \subseteq U + V$

proof –
interpret vectorspace $U$ by fact
interpret vectorspace $V$ by fact
show \[\text{thesis}\]
proof
show $U \neq \emptyset$ ..
show $U \subseteq U + V$
proof
fix $x$ assume $x: x \in U$
moreover have $0 \in V$ ..
ultimately have $x + 0 \in U + V$ ..
with $x$ show $x \in U + V$ by simp
qed
fix $x\ y$ assume $x: x \in U$ and $y \in U$
then show $x + y \in U$ by simp
from $x$ show $\forall a.\ a \cdot x \in U$ by simp
qed
qed

The sum of two subspaces is again a subspace.

lemma sum-subspace [intro?]:
assumes \(\text{subspace } U \ E\ \text{vectorspace } E\ \text{subspace } V \ E\)
shows $U + V \subseteq E$
proof –
interpret subspace $U \ E$ by fact
interpret vectorspace $E$ by fact
interpret subspace $V \ E$ by fact
show \[\text{thesis}\]
proof
have $0 \in U + V$
proof
show $0 \in U$ using (vectorspace $E$) ..
show $0 \in V$ using (vectorspace $E$) ..
show $(0,\:'a) = 0 + 0$ by simp
qed
then show $U + V \neq \emptyset$ by blast
show $U + V \subseteq E$
proof
fix $x$ assume $x \in U + V$
then obtain $u\ v$ where $x = u + v$ and
$u \in U$ and $v \in V$ ..
then show $x \in E$ by simp
qed
next
fix $x\ y$ assume $x: x \in U + V$ and $y: y \in U + V$
show $x + y \in U + V$
proof –
from $x$ obtain $ux\ vx$ where $x = ux + vx$ and $ux \in U$ and $vx \in V$ ..
moreover
from $y$ obtain $uy\ vy$ where $y = uy + vy$ and $uy \in U$ and $vy \in V$ ..
ultimately
have $ux + uy \in U$
and $vx + vy \in V$
and \( x + y = (ux + uy) + (vx + vy) \)

using \( x y \) by \((\text{simp-all add: add-ac})\)

then show \(?\text{thesis}..\)

qed

fix \( a \) show \( a \cdot x \in U + V \)

proof

from \( x \) obtain \( u v \) where \( x = u + v \) and \( u \in U \) and \( v \in V \) ..

then have \( a \cdot u \in U \) and \( a \cdot v \in V \)

and \( a \cdot x = (a \cdot u) + (a \cdot v) \) by \((\text{simp-all add: distrib})\)

then show \(?\text{thesis}..\)

qed

The sum of two subspaces is a vectorspace.

**4.4 Direct sums**

The sum of \( U \) and \( V \) is called direct, iff the zero element is the only common element of \( U \) and \( V \). For every element \( x \) of the direct sum of \( U \) and \( V \) the decomposition in \( x = u + v \) with \( u \in U \) and \( v \in V \) is unique.

**lemma** **decomp:**

assumes vectorspace \( E \) subspace \( U \) \( E \) subspace \( V \) \( E \)

assumes direct: \( U \cap V = \{0\} \)

and \( u1: u1 \in U \) and \( u2: u2 \in U \)

and \( v1: v1 \in V \) and \( v2: v2 \in V \)

and sum: \( u1 + v1 = u2 + v2 \)

shows \( u1 = u2 \land v1 = v2 \)

**proof** –

interpret vectorspace \( E \) by fact

interpret subspace \( U \) \( E \) by fact

interpret subspace \( V \) \( E \) by fact

show \(?\text{thesis}\)

proof

have \( U: \text{vectorspace } U \)

using \((\text{subspace } U \ E \) \( \langle \text{vectorspace } E \rangle \) by \((\text{rule subspace.vectorspace})\)

have \( V: \text{vectorspace } V \)

using \((\text{subspace } V \ E \) \( \langle \text{vectorspace } E \rangle \) by \((\text{rule subspace.vectorspace})\)

from \( u1 \ u2 \ v1 \ v2 \) and sum have eq: \( u1 - u2 = v2 - v1 \)

by \((\text{simp add: add-diff-swap})\)

from \( u1 \ u2 \) have: \( u1 - u2 \in U \)

by \((\text{rule vectorspace.diff-closed \[OF } U]\))

with eq have: \( v2 - v1 \in U \) by \((\text{simp only})\)

from \( v2 \ v1 \) have: \( v2 - v1 \in V \)

by \((\text{rule vectorspace.diff-closed \[OF } V]\))

with eq have: \( u1 - u2 \in V \) by \((\text{simp only})\)

show \( u1 = u2 \)

proof \((\text{rule add-minus-eq})\)

from \( u1 \) show \( u1 \in E \) ..
from $u_2$ show $u_2 \in E$ ..  
from $u u'$ and direct show $u_1 - u_2 = 0$ by blast  
qed  
show $v_1 = v_2$  
proof (rule add-minus-eq [symmetric])  
from $v_1$ show $v_1 \in E$ ..  
from $v_2$ show $v_2 \in E$ ..  
from $v v'$ and direct show $v_2 - v_1 = 0$ by blast  
qed  
qed  

An application of the previous lemma will be used in the proof of the Hahn-
Banach Theorem (see page 41): for any element $y + a \cdot x_0$ of the direct sum of  
a vectorspace $H$ and the linear closure of $x_0$ the components $y \in H$ and $a$ are  
uniquely determined.

lemma decomp-$H'$:  
assumes vectorspace $E$ subspace $H$ $E$  
assumes $y_1$: $y_1 \in H$ and $y_2$: $y_2 \in H$  
and $x'$: $x' \notin H$ $x' \in E$ $x' \neq 0$  
and eq: $y_1 + a_1 \cdot x' = y_2 + a_2 \cdot x'$  
shows $y_1 = y_2 \land a_1 = a_2$  
proof  
interpret vectorspace $E$ by fact  
interpret subspace $H$ $E$ by fact  
show \phantom{thesis}  
proof  
have $c$: $y_1 = y_2 \land a_1 \cdot x' = a_2 \cdot x'$  
proof (rule decomp)  
show $a_1 \cdot x' \in \text{lin } x'$ ..  
show $a_2 \cdot x' \in \text{lin } x'$ ..  
show $H \cap \text{lin } x' = \{0\}$  
proof  
show $H \cap \text{lin } x' \subseteq \{0\}$  
proof  
fix $x$ assume $x$: $x \in H \cap \text{lin } x'$  
thен obtain $a$ where $xx'$: $x = a \cdot x'$  
by blast  
have $x = 0$  
proof cases  
assume $a = 0$  
with $xx'$ and $x'$ show \phantom{thesis} by simp  
next  
assume $a$: $a \neq 0$  
from $x$ have $x \in H$ ..  
with $xx'$ have inverse $a \cdot a \cdot x' \in H$ by simp  
with $a$ and $x'$ have $x' \in H$ by (simp add: mult-assoc2)  
with ($x' \notin H$) show \phantom{thesis} by contradiction  
qed  
then show $x \in \{0\}$ ..  
qed  
show $\{0\} \subseteq H \cap \text{lin } x'$  
proof  
have $0 \in H$ using (vectorspace $E$) ..
4.4 Direct sums

moreover have \(0 \in \text{lin } x'\) using \((x' \in E)\) ..
ultimately show \(\text{thesis by blast}\)
qed
qed
show \(\text{lin } x' \subseteq E\) using \((x' \in E)\) ..
qed (rule \((\text{vectorspace } E)\), rule \((\text{subspace } H E)\), rule y1, rule y2, rule eq)
then show \(y1 = y2\) ..
from \(c\) have \(a1 \cdot x' = a2 \cdot x'\) ..
with \(x'\) show \(a1 = a2\) by (simp add: mult-right-cancel)
qed
qed

Since for any element \(y + a \cdot x'\) of the direct sum of a vectorspace \(H\) and the linear closure of \(x'\) the components \(y \in H\) and \(a\) are unique, it follows from \(y \in H\) that \(a = 0\).

lemma \texttt{decomp-H'-H'}:
assumes \(\text{vectorspace } E \text{ subspace } H E\)
assumes \(t \in H\)
and \(x': x' \notin H \iff x' \in E \wedge x' \neq 0\)
shows \((\text{SOME } (y, a). t = y + a \cdot x' \wedge y \in H) = (t, 0)\)
proof –
interpret \texttt{vectorspace } \(E\) by \texttt{fact}
interpret \texttt{subspace } \(H E\) by \texttt{fact}
show \(\text{thesis}\)
proof (rule, simp-all only; split-paired-all split-conv)
from \(t x'\) show \(t = t + 0 \cdot x' \wedge t \in H\) by simp
fix \(y\) and \(a\) assume \(ya\): \(t = y + a \cdot x' \wedge y \in H\)
have \(y = t \wedge a = 0\)
proof (rule \texttt{decomp-H'})
from \(ya\) \(x'\) show \(y + a \cdot x' = t + 0 \cdot x'\) by simp
from \(ya\) show \(y \in H\) ..
qed (rule \((\text{vectorspace } E)\), rule \((\text{subspace } H E)\), rule \(t\), (rule \(x'\))+)
with \(t x'\) show \((y, a) = (y + a \cdot x', 0)\) by simp
qed
qed

The components \(y \in H\) and \(a\) in \(y + a \cdot x'\) are unique, so the function \(h'\) defined by \(h'(y + a \cdot x') = h(y + a \cdot x')\) is definite.

lemma \texttt{h'-definite}:
fixes \(H\)
assumes \(\text{h'-def}\):
\(h' \equiv \lambda x.\)
\(\text{let } (y, a) = \text{SOME } (y, a). (x = y + a \cdot x' \wedge y \in H)\)
in \((h y) + a \cdot x_i\)
and \(x: x = y + a \cdot x'\)
assumes \(\text{vectorspace } E \text{ subspace } H E\)
assumes \(y \in H\)
and \(x': x' \notin H \iff x' \in E \wedge x' \neq 0\)
shows \(h' x = h y + a \cdot x_i\)
proof –
interpret \texttt{vectorspace } \(E\) by \texttt{fact}
interpret \texttt{subspace } \(H E\) by \texttt{fact}
from \(x y x'\) have \(x \in H + \text{lin } x'\) by \texttt{auto}
have \( \exists! p. (\lambda(y, a). x = y + a \cdot x' \land y \in H) \) \( p \) (is \( \exists! p. ?P p \))
proof (rule ex-ex1I)
from \( x \ y \) show \( \exists! p. ?P p \) by blast
fix \( p \ q \) assume \( p : ?P p \) and \( q : ?P q \)
show \( p = q \)
proof
from \( p \) have \( xp : x = \text{fst } p + \text{snd } p \cdot x' \land \text{fst } p \in H \)
by (cases \( p \)) simp
from \( q \) have \( xq : x = \text{fst } q + \text{snd } q \cdot x' \land \text{fst } q \in H \)
by (cases \( q \)) simp
have \( \text{fst } p = \text{fst } q \land \text{snd } p = \text{snd } q \)
proof (rule decomp-H')
from \( xp \) show \( \text{fst } p \in H \)
from \( xq \) show \( \text{fst } q \in H \)
from \( xp \) and \( xq \) show \( \text{fst } p + \text{snd } p \cdot x' = \text{fst } q + \text{snd } q \cdot x' \)
by simp
qed (rule (vectorspace \( E \)), rule (subspace \( H E \)), (rule \( x' \)) +)
then show \( ?\text{thesis} \) by (cases \( p \), cases \( q \)) simp
qed

5 Normed vector spaces

theory Normed-Space
imports Subspace
begin

5.1 Quasinorms

A seminorm \( \| \cdot \| \) is a function on a real vector space into the reals that has the following properties: it is positive definite, absolute homogeneous and subadditive.

locale seminorm =
fixes \( V : 'a::{\text{minus, plus, zero, uminus}} \) set
fixes norm :: 'a \Rightarrow real (\( \| \cdot \| \))
assumes ge-zero [iff?]: \( x \in V \Rightarrow 0 \leq \| x \| \)
and abs-homogenous [iff?]: \( x \in V \Rightarrow \| a \cdot x \| = |a| \cdot \| x \| \)
and subadditive [iff?]: \( x \in V \Rightarrow y \in V \Rightarrow \| x + y \| \leq \| x \| + \| y \| \)

declare seminorm.intro [intro?]

lemma (in seminorm) diff-subadditive:
assumes vectorspace \( V \)
shows \( x \in V \Rightarrow y \in V \Rightarrow \| x - y \| \leq \| x \| + \| y \| \)
proof
interpret vectorspace \( V \) by fact
assume \( x : x \in V \) and \( y : y \in V \)
5.2 Norms

A norm \( \| \cdot \| \) is a seminorm that maps only the 0 vector to 0.

locale norm = seminorm +
assumes zero-iff [iff]: \( x \in V \implies (\| x \| = 0) = (x = 0) \)

5.3 Normed vector spaces

A vector space together with a norm is called a normed space.

locale normed-vectorspace = vectorspace + norm
declare normed-vectorspace.intro [intro?]

lemma (in normed-vectorspace) gt-zero [intro?):
assumes x: \( x \in V \) and neq: \( x \neq 0 \)
shows \( \theta < \| x \| \)
proof –
from x have \( \theta \leq \| x \| \)
also have \( \theta \neq \| x \| \)
proof
assume \( \theta = \| x \| \)
with x have \( x = 0 \) by simp
with neq show False by contradiction
qed
finally show \( ?thesis \).
qed

Any subspace of a normed vector space is again a normed vectorspace.

lemma subspace-normed-us [intro?]:
fixes F E norm
assumes subspace $F \subseteq E$ normed-vectorspace $E$ norm
shows normed-vectorspace $F$ norm
proof –
interpret subspace $F \subseteq E$ by fact
interpret normed-vectorspace $E$ norm by fact
show ?thesis
proof
show vectorspace $F$ by (rule vectorspace) unfold-locales
next
have Normed-Space $E$ norm ..
with subset show Normed-Space $F$ norm
by (simp add: norm-def seminorm-def norm-axioms-def)
qed
qed
end

6 Linearforms

theory Linearform
imports Vector-Space
begin

A linear form is a function on a vector space into the reals that is additive and multiplicative.

locale linearform =
  fixes $V$ :: ′a::{minus, plus, zero, aminus} set and $f$
  assumes add [iff]: $x \in V \implies y \in V \implies f (x + y) = f x + f y$
and mult [iff]: $x \in V \implies f (a \cdot x) = a \cdot f x$

declare linearform.intro [intro?]

lemma (in linearform) neg [iff]:
  assumes vectorspace $V$
  shows $x \in V \implies f (\neg x) = - f x$
proof –
interpret vectorspace $V$ by fact
assume $x : x \in V$
then have $f (\neg x) = f ((\neg 1) \cdot x)$ by (simp add: negate-eq1)
also from $x$ have \ldots = $(- 1) \cdot (f x)$ by (rule mult)
also from $x$ have \ldots = $- (f x)$ by simp
finally show ?thesis .
qed

lemma (in linearform) diff [iff]:
  assumes vectorspace $V$
  shows $x \in V \implies y \in V \implies f (x - y) = f x - f y$
proof –
interpret vectorspace $V$ by fact
assume $x : x \in V$ and $y : y \in V$
then have $x - y = x + (- y)$ by (rule diff-eq1)
also have $f \ldots = f x + f (- y)$ by (rule add) (simp-all add: $x y$)
also have $f (- y) = - f y$ using (vectorspace $V$) $y$ by (rule neg)
finally show thesis by simp

qed

Every linear form yields 0 for the 0 vector.

lemma (in linearform) zero [iff]:
  assumes vectorspace V
  shows f 0 = 0
proof –
  interpret vectorspace V by fact
  have f 0 = f (0 − 0) by simp
  also have ... = f 0 − f 0 using (vectorspace V) by (rule diff) simp-all
  also have ... = 0 by simp
  finally show thesis .

qed

end

7 An order on functions

theory Function-Order
imports Subspace Linearform
begin

7.1 The graph of a function

We define the graph of a (real) function \( f \) with domain \( F \) as the set

\[
\{(x, f x) \mid x \in F\}
\]

So we are modeling partial functions by specifying the domain and the mapping function. We use the term "function" also for its graph.

type-synonym 'a graph = ('a × real) set

definition graph :: 'a set ⇒ ('a ⇒ real) ⇒ 'a graph
  where graph F f = \{(x, f x) \mid x \in F\}

lemma graphI [intro]: x ∈ F ⇒ (x, f x) ∈ graph F f
  unfolding graph-def by blast

lemma graphI2 [intro?]: x ∈ F ⇒ ∃ t ∈ graph F f. t = (x, f x)
  unfolding graph-def by blast

lemma graphE [elim?]:
  assumes (x, y) ∈ graph F f
  obtains x ∈ F and y = f x
  using assms unfolding graph-def by blast

7.2 Functions ordered by domain extension

A function \( h' \) is an extension of \( h \), iff the graph of \( h \) is a subset of the graph of \( h' \).

lemma graph-extI:
\[(\forall x. x \in H \implies h x = h' x) \implies H \subseteq H'\]

unfolding graph-def by blast

lemma graph-extD1 [dest?]: graph H h \subseteq graph H' h' \implies x \in H \implies h x = h' x

unfolding graph-def by blast

lemma graph-extD2 [dest?]: graph H h \subseteq graph H' h' \implies H \subseteq H'

unfolding graph-def by blast

7.3 Domain and function of a graph

The inverse functions to graph are domain and funct.

definition domain :: 'a graph \Rightarrow 'a set
  where domain g = \{x. \exists y. (x, y) \in g\}

definition funct :: 'a graph \Rightarrow ('a \Rightarrow real)
  where funct g = (\lambda x. (SOME y. (x, y) \in g))

The following lemma states that g is the graph of a function if the relation induced by g is unique.

lemma graph-domain-funct:
  assumes uniq: \(\forall x y z. (x, y) \in g \implies (x, z) \in g \implies z = y\)
  shows graph (domain g) (funct g) = g

unfolding domain-def funct-def graph-def

proof auto
  fix a b assume g: (a, b) \in g
  from g show (a, SOME y. (a, y) \in g) \in g by (rule someI2)
  from g show \exists y. (a, y) \in g ..
  from g show b = (SOME y. (a, y) \in g)
  proof (rule some-equality [symmetric])
    fix y assume (a, y) \in g
    with g show y = b by (rule uniq)
  qed
  qed

7.4 Norm-preserving extensions of a function

Given a linear form f on the space F and a seminorm p on E. The set of all linear extensions of f, to superspaces H of F, which are bounded by p, is defined as follows.

definition norm-pres-extensions ::
  'a::{plus, minus, uminus, zero} set \Rightarrow ('a \Rightarrow real) \Rightarrow 'a set \Rightarrow ('a \Rightarrow real)
  \Rightarrow 'a graph set

where

norm-pres-extensions E p F f
= \{g. \exists H h. g = graph H h
  \wedge linearform H h
  \wedge H \subseteq E
  \wedge F \subseteq H
  \wedge graph F f \subseteq graph H h\}
\( \wedge (\forall x \in H. \ h x \leq p x) \)  

**Lemma norm-pres-extensionE [elim]:**

**Assumes** \( g \in \text{norm-pres-extensions } E \ p \ f \)

**Obtains** \( H h \)

**Where** \( g = \text{graph } H h \)

and \( \text{linearform } H h \)

and \( H \subseteq E \)

and \( F \subseteq H \)

and \( \text{graph } F f \subseteq \text{graph } H h \)

and \( \forall x \in H. h x \leq p x \)

using assms unfolding norm-pres-extensions-def by blast

**Lemma norm-pres-extensionI2 [intro]:**

\( \text{linearform } H h \implies H \subseteq E \implies F \subseteq H \)

\( \implies \text{graph } F f \subseteq \text{graph } H h \implies \forall x \in H. h x \leq p x \)

unfolding norm-pres-extensions-def by blast

**Lemma norm-pres-extensionI**:  
\( \exists H h. g = \text{graph } H h \)

\( \wedge \text{linearform } H h \)

\( \wedge F \subseteq H \)

\( \wedge \text{graph } F f \subseteq \text{graph } H h \)

\( \wedge (\forall x \in H. h x \leq p x) \implies g \in \text{norm-pres-extensions } E \ p \ f \)

unfolding norm-pres-extensions-def by blast

end

## 8 The norm of a function

**Theory** Function-Norm  
**Imports** Normed-Space Function-Order

begin

### 8.1 Continuous linear forms

A linear form \( f \) on a normed vector space \( (V, \|\cdot\|) \) is **continuous**, iff it is bounded, i.e.

\[ \exists c \in R. \forall x \in V. |f x| \leq c \cdot \|x\| \]

In our application no other functions than linear forms are considered, so we can define continuous linear forms as bounded linear forms:

**Locale** continuous = linearform +  
**Fixes** \( \text{norm} :: - \Rightarrow \text{real} \quad (\|\cdot\|) \)

**Assumes** bounded: \( \exists c. \forall x \in V. |f x| \leq c \cdot \|x\| \)

declare continuous.intro [intro?] continuous-axioms.intro [intro?]

**Lemma** continuousI [intro]:

**Fixes** \( \text{norm} :: - \Rightarrow \text{real} \quad (\|\cdot\|) \)
assumes linearform $Vf$
assumes $r$: $\forall x. x \in V \Rightarrow |f x| \leq c \cdot \|x\|
shows continuous $Vf$ norm
proof
show linearform $Vf$ by fact
from $r$ have $\exists c. \forall x \in V. |f x| \leq c \cdot \|x\|$ by blast
then show continuous-axioms $Vf$ norm ..
qed

8.2 The norm of a linear form

The least real number $c$ for which holds

$$\forall x \in V. |f x| \leq c \cdot \|x\|$$

is called the norm of $f$.

For non-trivial vector spaces $V \neq \{0\}$ the norm can be defined as

$$\|f\| = \sup x \neq 0. |f x| / \|x\|$$

For the case $V = \{0\}$ the supremum would be taken from an empty set. Since $\mathbb{R}$ is unbounded, there would be no supremum. To avoid this situation it must be guaranteed that there is an element in this set. This element must be $\{\} \geq 0$ so that fn-norm has the norm properties. Furthermore it does not have to change the norm in all other cases, so it must be $0$, as all other elements are $\{\} \geq 0$.

Thus we define the set $B$ where the supremum is taken from as follows:

$$\{0\} \cup \{|f x| / \|x\|. x \neq 0 \land x \in F\}$$

fn-norm is equal to the supremum of $B$, if the supremum exists (otherwise it is undefined).

locale fn-norm =
fixes norm :: $\Rightarrow$ real ($\|\cdot\|$)
fixes $B$ defines $B V f \equiv \{0\} \cup \{|f x| / \|x\|. x \neq 0 \land x \in V\}$
fixes fn-norm ($\|\cdot\| V f$)
defines $\|f\| V f \equiv \bigvee (B V f)$

locale normed-vectorspace-with-fn-norm = normed-vectorspace + fn-norm

lemma (in fn-norm) B-not-empty [intro]: $0 \in B V f$
by (simp add: B-def)

The following lemma states that every continuous linear form on a normed space $(V, \|\cdot\|)$ has a function norm.

lemma (in normed-vectorspace-with-fn-norm) fn-norm-works:
assumes continuous $Vf$ norm
shows lub (B $V f$) ($\|f\| V f$)
proof −
interpret continuous $Vf$ norm by fact

The existence of the supremum is shown using the completeness of the reals. Completeness means, that every non-empty bounded set of reals has a supremum.
8.2 The norm of a linear form

have \(\exists a. \, \text{lub} (B V f) \ a\)
proof (rule real-complete)

First we have to show that \(B\) is non-empty:

have \(0 \in B V f\)
then show \(\exists x. \, x \in B V f\)

Then we have to show that \(B\) is bounded:

show \(\exists c. \, \forall y \in B V f. \, y \leq c\)
proof –

We know that \(f\) is bounded by some value \(c\).

from bounded obtain \(c\) where \(c: \forall x \in V. \, |f x| \leq c * \|x\|\)

To prove the thesis, we have to show that there is some \(b\), such that \(y \leq b\) for all \(y \in B\). Due to the definition of \(B\) there are two cases.

def \(b \equiv \max c 0\)
have \(\forall y \in B V f. \, y \leq b\)
proof

fix \(y\)
assume \(y: y \in B V f\)
show \(y \leq b\)
proof cases

assume \(y = 0\)
then show \(?\text{thesis}\)
unfolding \(b\)-def by arith

next

The second case is \(y = |f x| / \|x\|\) for some \(x \in V\) with \(x \neq 0\).

assume \(y \neq 0\)
with \(y\) obtain \(x\) where \(y\)-rep: \(y = |f x| * \text{inverse} \|x\|\)
and \(x: x \in V\) and \(\text{neq}: x \neq 0\)
by (auto simp add: \(B\)-def divide-inverse)
from \(\text{neq}\) have \(gt: 0 < \|x\|\)

The thesis follows by a short calculation using the fact that \(f\) is bounded.

note \(y\)-rep
also have \(|f x| * \text{inverse} \|x\| \leq (c * \|x\|) * \text{inverse} \|x\|\)
proof (rule mult-right-mono)
from \(c \ x\) show \(|f x| \leq c * \|x\|\)
from \(\text{gt}\) have \(0 < \text{inverse} \|x\|\)
by (rule positive-imp-inverse-positive)
then show \(0 \leq \text{inverse} \|x\|\)
by (rule order-less-imp-le)
qed
also have \(\ldots = c * (\|x\| * \text{inverse} \|x\|)\)
by (rule Groups.mult.assoc)
also
from \(\text{gt}\) have \(\|x\| \neq 0\) by simp
then have \(\|x\| * \text{inverse} \|x\| = 1\) by simp
also have \(c * 1 \leq b\)
by (simp add: \(b\)-def)
finally show \(y \leq b\)
qed

then show \(?\text{thesis}\)
qed
The norm of a continuous function is always ≥ 0.

The fundamental property of function norms is:

\[ |f(x)| \leq \|f\| \cdot \|x\| \]
8.2 The norm of a linear form

proof cases
  assume \( x = 0 \)
  then have \(|f x| = |f 0|\) by simp
  also have \( f 0 = 0 \) by rule unfold-locales
  also have \(|\ldots| = 0\) by simp
  also have \( a: 0 \leq \|f\| - V\)
    using (continuous V f norm) by (rule fn-norm-ge-zero)
  from \( x \) have \( 0 \leq \|f\| - V \cdot \|x\|\)
    by simp
  finally show \(|f x| \leq \|f\| - V \cdot \|x\|\).
next
  assume \( x \neq 0 \)
  with \( x \) have neq: \( \|x\| \neq 0 \) by simp
  then have \(|f x| = (|f x| \cdot \text{inverse} \ |\|x\||) \cdot \|x\|\) by simp
  also have \( \ldots \leq \|f\| - V \cdot \|x\|\)
    proof (rule fn-right-mono)
      from \( x \) show \( 0 \leq \|x\|\)
      then have \(|f x| \cdot \text{inverse} \ |\|x\|| \in B V f\)
        by (auto simp add: B-def divide-inverse)
      with (continuous V f norm) show \(|f x| \cdot \text{inverse} \ |\|x\|| \leq \|f\| - V\)
        by (rule fn-norm-ub)
    qed
  finally show \(?thesis\).
qed
qed

The function norm is the least positive real number for which the following
inequation holds:

\(|f x| \leq c \cdot \|x\|\)

lemma (in normed-vectorspace-with-fn-norm) fn-norm-least [intro?]:
  assumes continuous V f norm
  assumes neq: \( \forall x. \ x \in V \Rightarrow |f x| \leq c \cdot \|x\|\) and ge: \( 0 \leq c\)
  shows \( \|f\| - V \leq c\)
proof
  interpret continuous V f norm by fact
  show \(?thesis\)
  proof (rule fn-norm-leastB [folded B-def fn-norm-def])
    fix \( b \) assume \( b: b \in B V f\)
    show \( b \leq c\)
    proof cases
      assume \( b = 0\)
        with ge show \(?thesis\) by simp
    next
      assume \( b \neq 0\)
        with \( b \) obtain \( x \) where \( b = |f x| \cdot \text{inverse} \ |\|x\||\)
          and x-neq: \( x \neq 0 \) and \( x: x \in V\)
          by (auto simp add: B-def divide-inverse)
        note \( b\)-rep
        also have \(|f x| \cdot \text{inverse} \ |\|x\|| \leq (c \cdot \|x\|) \cdot \text{inverse} \ |\|x\||\)
          proof (rule mul-right-mono)
            have \( 0 < \|x\|\) using \( x\)-neq ..
            then show \( 0 \leq \text{inverse} \ |\|x\||\) by simp
          qed
        finally show \(?thesis\).
      qed
    qed
30

9 ZORN’S LEMMA

from \( x \) show \(|f \cdot x| \leq c \cdot \|x\|\) by (rule ineq)
qed
also have \( \ldots = c \)
proof –
from \( x \neq \) and \( x \) have \( \|x\| \neq 0 \) by simp
then show ?thesis by simp
qed
finally show ?thesis .
qed
qed (insert \langle \text{continuous } V f \text{ norm} \rangle, simp-all add: continuous-def)
qed
end

9 Zorn’s Lemma

theory Zorn-Lemma
imports Main
begin

Zorn’s Lemmas states: if every linear ordered subset of an ordered set \( S \) has an upper bound in \( S \), then there exists a maximal element in \( S \). In our application, \( S \) is a set of sets ordered by set inclusion. Since the union of a chain of sets is an upper bound for all elements of the chain, the conditions of Zorn’s lemma can be modified: if \( S \) is non-empty, it suffices to show that for every non-empty chain \( c \) in \( S \) the union of \( c \) also lies in \( S \).

theorem Zorn’s-Lemma:
assumes \( r: \bigwedge c. c \in \text{chains } S \implies \exists x. x \in c \implies \bigcup c \in S \)
and \( aS: a \in S \)
shows \( \exists y \in S. \forall z \in S. y \subseteq z \implies z = y \)
proof (rule Zorn-Lemma2)
show \( \forall c \in \text{chains } S. \exists y \in S. \forall z \in c. z \subseteq y \)
proof
fix \( c \)
assume \( c \in \text{chains } S \)
show \( \exists y \in S. \forall z \in c. z \subseteq y \)
proof cases
If \( c \) is an empty chain, then every element in \( S \) is an upper bound of \( c \).

assume \( c = \{\} \)
with \( aS \) show ?thesis by fast
If \( c \) is non-empty, then \( \bigcup c \) is an upper bound of \( c \), lying in \( S \).

next
assume \( c \neq \{\} \)
show ?thesis
proof
show \( \forall z \in c. z \subseteq \bigcup c \) by fast
show \( \bigcup c \in S \)
proof (rule \( r \))
from \( c \neq \{\} \) show \( \exists x. x \in c \) by fast
show \( c \in \text{chains } S \) by fact
qed
qed
qed
qed
qed
qed

end
Part II

Lemmas for the Proof

10  The supremum w.r.t. the function order

theory Hahn-Banach-Sup-Lemmas
imports Function-Norm Zorn-Lemma
begin

This section contains some lemmas that will be used in the proof of the Hahn-Banach Theorem. In this section the following context is presumed. Let $E$ be a real vector space with a seminorm $p$ on $E$. $F$ is a subspace of $E$ and $f$ a linear form on $F$. We consider a chain $c$ of norm-preserving extensions of $f$, such that $\bigcup c = \text{graph } H h$. We will show some properties about the limit function $h$, i.e. the supremum of the chain $c$.

Let $c$ be a chain of norm-preserving extensions of the function $f$ and let $\text{graph } H h$ be the supremum of $c$. Every element in $H$ is member of one of the elements of the chain.

lemmas [dest?] = chainsD
lemmas [elim?] = chainsE2 [elim-format]

lemma some-H h′:
assumes $M$: $M = \text{norm-pres-extensions } E p F f$
and $cM$: $c \in \text{chains } M$
and $u$: $\text{graph } H h = \bigcup c$
and $x$: $x \in H$
shows $\exists H′ h′. \text{graph } H′ h′ \in c$
\hfill $\land (x, h x) \in \text{graph } H′ h′$
\hfill $\land \text{linearform } H′ h′ \land H′ \subseteq E$
\hfill $\land F \subseteq H′ \land \text{graph } F f \subseteq \text{graph } H′ h′$
\hfill $\land (\forall x \in H′. h′ x \leq p x)$

proof –

from $x$ have $(x, h x) \in \text{graph } H h$ ..
also from $u$ have $\ldots = \bigcup c$ .
finally obtain $g$ where $gc: g \in c$ and $gh: (x, h x) \in g$ by blast

from $cM$ have $c \subseteq M$ ..
with $gc$ have $g \in M$ ..
also from $M$ have $\ldots = \text{norm-pres-extensions } E p F f$ .
finally obtain $H′$ and $h′$ where $g: g = \text{graph } H′ h′$
\hfill $\land \ast: \text{linearform } H′ h′ \land H′ \subseteq E \land F \subseteq H′$
\hfill $\land \text{graph } F f \subseteq \text{graph } H′ h′ \land (\forall x \in H′. h′ x \leq p x)$ ..

from $gc$ and $g$ have $\text{graph } H′ h′ \in c$ by $(\text{simp only:})$
moreover from $gh$ and $g$ have $(x, h x) \in \text{graph } H′ h′$ by $(\text{simp only:})$
ultimately show $\ast$ using $\ast$ by blast
qed

Let $c$ be a chain of norm-preserving extensions of the function $f$ and let $\text{graph } H h$ be the supremum of $c$. Every element in the domain $H$ of the supremum

function is member of the domain $H'$ of some function $h'$, such that $h$ extends $h$.

**lemma some-$H'h'$:**

**assumes** $M: M = \text{norm-pres-extensions} E p F f$

and $cM: c \in \text{chains} M$

and $u: \text{graph} \ H h = \bigcup c$

and $x: x \in H$

**shows** $\exists H' h'. \ x \in H' \land \text{graph} \ H' h' \subseteq \text{graph} \ H h$

$\land \text{linearform} \ H' h' \land H' \subseteq E \land F \subseteq H'$

$\land \text{graph} \ F f \subseteq \text{graph} \ H' h' \land (\forall x \in H'. \ h' x \leq p x)$

**proof** –

from $M \ cM \ u \ c$ obtain $H' h'$ where

$x$-hx: $(x, h x) \in \text{graph} \ H' h'$

and $c: \text{graph} \ H' h' \subseteq c$

and $*: \text{linearform} \ H' h' \ H' \subseteq E \land F \subseteq H'$

$\land \text{graph} \ F f \subseteq \text{graph} \ H' h' \land (\forall x \in H'. \ h' x \leq p x)$

by (rule some-$H'h't$ [elim-formal]) blast

from $x$-hx have $x \in H'$.

moreover from $M \ cM \ u \ c$ have $\text{graph} \ H' h' \subseteq \text{graph} \ H h$ by blast

ultimately show $\textit{thesis}$ using $*$ by blast

qed

Any two elements $x$ and $y$ in the domain $H$ of the supremum function $h$ are both in the domain $H'$ of some function $h'$, such that $h$ extends $h'$.

**lemma some-$H'h'2$:**

**assumes** $M: M = \text{norm-pres-extensions} E p F f$

and $cM: c \in \text{chains} M$

and $u: \text{graph} \ H h = \bigcup c$

and $x: x \in H$

and $y: y \in H$

**shows** $\exists H' h'. \ x \in H' \land y \in H'$

$\land \text{graph} \ H' h' \subseteq \text{graph} \ H h$

$\land \text{linearform} \ H' h' \land H' \subseteq E \land F \subseteq H'$

$\land \text{graph} \ F f \subseteq \text{graph} \ H' h' \land (\forall x \in H'. \ h' x \leq p x)$

**proof** –

$y$ is in the domain $H''$ of some function $h''$, such that $h$ extends $h''$.

from $M \ cM \ u$ and $y$ obtain $H' h'$ where

$y$-hy: $(y, h y) \in \text{graph} \ H' h'$

and $c': \text{graph} \ H' h' \subseteq c$

and $*: \text{linearform} \ H' h' \ H' \subseteq E \land F \subseteq H'$

$\land \text{graph} \ F f \subseteq \text{graph} \ H' h' \land (\forall x \in H'. \ h' x \leq p x)$

by (rule some-$H'h't$ [elim-formal]) blast

$x$ is in the domain $H$ of some function $h'$, such that $h$ extends $h'$.

from $M \ cM \ u$ and $x$ obtain $H'' h''$ where

$x$-hx: $(x, h x) \in \text{graph} \ H'' h''$

and $c'': \text{graph} \ H'' h'' \subseteq c$

and $**: \text{linearform} \ H'' h'' \ H'' \subseteq E \land F \subseteq H''$

$\land \text{graph} \ F f \subseteq \text{graph} \ H'' h'' \land (\forall x \in H''. \ h'' x \leq p x)$
by (rule some-H' h't [elim-format]) blast

Since both \( h' \) and \( h'' \) are elements of the chain, \( h'' \) is an extension of \( h' \) or vice versa. Thus both \( x \) and \( y \) are contained in the greater one.

From \( cM \ c' \) have graph \( H'' h' h'' \subseteq graph H' h' \vee graph H' h' \subseteq graph H'' h'' \) (in \(?case1 \vee ?case2\) ..

Then show \(?thesis\)

Proof

Assume \(?case1\)

Have \((x, h x) \in graph H'' h''\) by fact

Also have \(... \subseteq graph H' h'\) by fact

Finally have \(xh:(x, h x) \in graph H' h'\).

Then have \(x \in H'\) ..

Moreover from \(y-hy\) have \(y \in H'\) ..

Moreover from \(cM u\) and \(c'\) have graph \(H' h' \subseteq graph H h\) by blast

Ultimately show \(?thesis\) using * by blast

Next

Assume \(?case2\)

From \(x-hx\) have \(x \in H''\) ..

Moreover \{ have \((y, h y) \in graph H' h'\) by (rule \(y-hy\))

Also have \(... \subseteq graph H'' h''\) by fact

Finally have \((y, h y) \in graph H'' h''\).

\} then have \(y \in H''\) ..

Moreover from \(cM u\) and \(c''\) have graph \(H'' h'' \subseteq graph H h\) by blast

Ultimately show \(?thesis\) using ** by blast

Qed

Qed

The relation induced by the graph of the supremum of a chain \( c \) is definite, i.e. \( t \) is the graph of a function.

Lemma sup-definite:

Assumes \( M\)-def: \( M \equiv norm-pres-extensions E p f \)

And \( cM: c \in chains M \)

And \( xy: (x, y) \in \bigcup c \)

And \( xz: (x, z) \in \bigcup c \)

Shows \( z = y \)

Proof –

From \( cM \) have \( c: c \subseteq M \) ..

From \( xy \) obtain \( G1 \) where \( xy': (x, y) \in G1 \) and \( G1: G1 \subseteq c \) ..

From \( xz \) obtain \( G2 \) where \( xz': (x, z) \in G2 \) and \( G2: G2 \subseteq c \) ..

From \( G1 \) \( c \) have \( G1 \subseteq M \) ..

Then obtain \( H1 h1 \) where \( G1\)-rep: \( G1 = graph H1 h1 \)

Unfolding \( M\)-def by blast

From \( G2 \) \( c \) have \( G2 \subseteq M \) ..

Then obtain \( H2 h2 \) where \( G2\)-rep: \( G2 = graph H2 h2 \)

Unfolding \( M\)-def by blast

\( G1 \) is contained in \( G2 \) or vice versa, since both \( G_1 \) and \( G_2 \) are members of \( c \).

From \( cM \) \( G1 \) \( G2 \) have \( G1 \subseteq G2 \lor G2 \subseteq G1 \) (is \(?case1 \lor ?case2\) ..

Then show \(?thesis\)
proof
  assume ?case1
  with \( y' \) \( G2 \)-rep have \((x, y) \in \operatorname{graph} H2\) \( h2 \) by blast
  then have \( y = h2 x \).
  also from \( x' \) \( G2 \)-rep have \((x, z) \in \operatorname{graph} H2\) \( h2 \) by (simp only:)
  then have \( z = h2 x \).
  finally show ?thesis.

next
  assume ?case2
  with \( xz' \) \( G1 \)-rep have \((x, z) \in \operatorname{graph} H1\) \( h1 \) by blast
  then have \( z = h1 x \).
  also from \( xy' \) \( G1 \)-rep have \((x, y) \in \operatorname{graph} H1\) \( h1 \) by (simp only:)
  then have \( y = h1 x \).
  finally show ?thesis.

qed

qed

The limit function \( h \) is linear. Every element \( x \) in the domain of \( h \) is in the domain of a function \( h' \) in the chain of norm preserving extensions. Furthermore, \( h \) is an extension of \( h' \) so the function values of \( x \) are identical for \( h' \) and \( h \).

Finally, the function \( h' \) is linear by construction of \( M \).

lemma sup-lf:
  assumes \( M : M = \operatorname{norm-pres-extensions} E\) \( p\) \( F\) \( f \)
  and \( cM : c \in \operatorname{chains} M \)
  and \( u : \operatorname{graph} H\) \( h \) = \( \bigcup c \)
  shows \( \operatorname{linearform} H\) \( h \)

proof
  fix \( x\) \( y \) assume \( x : x \in H\) and \( y : y \in H \)
  with \( M\) \( cM\) \( u\) obtain \( H'\) \( h' \) where
  \( x' : x \in H'\) and \( y' : y \in H' \)
  and \( b : \operatorname{graph} H'\) \( h' \subseteq \operatorname{graph} H\) \( h \)
  and \( \operatorname{linearform} : \operatorname{linearform} H'\) \( h' \)
  and \( \operatorname{subspace} : H' \subseteq E \)
  by (rule some-H [elim-format]) blast
  show \( h\) \( (x + y) = h x + h y \)
  proof
    from \( \operatorname{linearform} x' \) \( y' \) have \( h'\) \( (x + y) = h' x + h' y \)
    by (rule \( \operatorname{linearform}.add \))
    also from \( b \) \( x' \) have \( h' x = h x \).
    also from \( b \) \( y' \) have \( h' y = h y \).
    also from \( \operatorname{subspace} x' \) \( y' \) have \( x + y \in H' \)
    by (rule \( \operatorname{subspace}.add-closed \))
    with \( b \) have \( h'\) \( (x + y) = h (x + y) \).
    finally show ?thesis.
  qed

next
  fix \( x\) \( a \) assume \( x : x \in H \)
  with \( M\) \( cM\) \( u\) obtain \( H'\) \( h' \) where
  \( x' : x \in H'\)
  and \( b : \operatorname{graph} H'\) \( h' \subseteq \operatorname{graph} H\) \( h \)
and linearform: linearform $H' \ h'$
and subspace: $H' \subseteq E$
by (rule some-$H' \ h'$ [elim-format]) blast

show $h \ (a \cdot x) = a \ast h \ x$
proof
  from linearform $x'$ have $h' \ (a \cdot x) = a \ast h' \ x$
    by (rule linearform.mult)
  also from $b \ x' \ h' \ x = h \ x$
  also from subspace $x'$ have $a \cdot x \in H'$
    by (rule subspace.mult-closed)
  with $b$ have $h' \ (a \cdot x) = h \ (a \cdot x)$
  finally show $?thesis$.
qed

The limit of a non-empty chain of norm preserving extensions of $f$ is an extension of $f$, since every element of the chain is an extension of $f$ and the supremum is an extension for every element of the chain.

lemma sup-ext:
assumes graph: $\text{graph} \ H \ h = \bigcup c$
  and $M: \ M = \text{norm-pres-extensions} \ E \ p \ F \ f$
  and $cM: c \in \text{chains} \ M$
  and $\exists x. \ x \in c$
shows $\text{graph} \ F \ f \subseteq \text{graph} \ H \ h$
proof
  from $\exists x. \ x \in c$
  obtain $x$ where $x \in c$
  from $cM$ have $c \subseteq M$
  with $x$ have $x \in M$
  with $M$ have $x \in \text{norm-pres-extensions} \ E \ p \ F \ f$
    by (simp only)
  then obtain $G \ g$ where $x = \text{graph} \ G \ g$ and $\text{graph} \ F \ f \subseteq \text{graph} \ G \ g$
  then have $\text{graph} \ F \ f \subseteq x$ by (simp only)
  also from $x$ have $\ldots \subseteq \bigcup c$ by blast
  also from $\text{graph} \ have \ \ldots = \text{graph} \ H \ h$
  finally show $?thesis$.
qed

The domain $H$ of the limit function is a superspace of $F$, since $F$ is a subset of $H$. The existence of the $0$ element in $F$ and the closure properties follow from the fact that $F$ is a vector space.

lemma sup-supF:
assumes graph: $\text{graph} \ H \ h = \bigcup c$
  and $M: \ M = \text{norm-pres-extensions} \ E \ p \ F \ f$
  and $cM: c \in \text{chains} \ M$
  and $\exists x. \ x \in c$
  and $FE: \ F \subseteq E$
shows $F \subseteq H$
proof
  from $FE$ show $F \neq \{\}$ by (rule subspace.non-empty)
  from $\text{graph} \ M \ cM \ \exists x$ have $\text{graph} \ F \ f \subseteq \text{graph} \ H \ h$
    by (rule sup-ext)
  then show $F \subseteq H$.

fix $x$ $y$ assume $x \in F$ and $y \in F$
with $FE$ show $x + y \in F$ by (rule subspace.add-closed)
next
fix $x$ $a$ assume $x \in F$
with $FE$ show $a \cdot x \in F$ by (rule subspace.mult-closed)
qed

The domain $H$ of the limit function is a subspace of $E$.

**Lemma sup-subE:**

assumes graph: $\text{graph } H \ h = \bigcup c$
and $M; M = \text{norm-pres-extensions } E \ p \ f$
and $cM; c \in \text{chains } M$
and $\exists x. \ x \in c$
and $FE; F \subseteq E$
and $E; \text{vectorspace } E$
shows $H \subseteq E$

**proof**

show $H \neq \{\}$
proof –
from $FE \ E$ have $0 \in F$ by (rule subspace.zero)
also from graph $M \ cM \ \exists FE \ \text{have } F \subseteq H$ by (rule sup-supF)
then have $F \subseteq H$. ..
finally show $?\text{thesis }$ by blast
qed
show $H \subseteq E$

**proof**

fix $x$ assume $x \in H$
with $M \ cM$ graph
obtain $H'$ where $x; x \in H'$ and $H'E; H' \subseteq E$
by (rule some-H' h' [elim-format]) blast
from $H'E$ have $H' \subseteq E$. ..
with $x$ show $x \in E$. ..
qed

fix $x$ $y$ assume $x \in H$ and $y \in H$
show $x + y \in H$
proof –
from $M \ cM$ graph $x \ y$ obtain $H' \ h'$ where
$x'; x \in H'$ and $y'; y \in H'$ and $H'E; H' \subseteq E$
and graphs: $\text{graph } H' \ h' \subseteq \text{graph } H \ h$
by (rule some-H' h' [elim-format]) blast
from $H'E \ x' \ y'$ have $x + y \in H'$. ..
by (rule subspace.add-closed)
also from graphs have $H' \subseteq H$. ..
finally show $?\text{thesis }$.
qed

**next**

fix $x$ $a$ assume $x \in H$
show $a \cdot x \in H$
proof –
from $M \ cM$ graph $x$
obtain $H' \ h'$ where $x'; x \in H'$ and $H'E; H' \subseteq E$
and graphs: $\text{graph } H' \ h' \subseteq \text{graph } H \ h$
by (rule some-H' h' [elim-format]) blast
from $H'E \ x'$ have $a \cdot x \in H'$ by (rule subspace.mult-closed)
also from graphs have \( H' \subseteq H \).

finally show thesis.

\[ \text{qed} \]

\[ \text{qed} \]

The limit function is bounded by the norm \( p \) as well, since all elements in the chain are bounded by \( p \).

**lemma** sup-norm-pres:

assumes graph: \( \text{graph } H \ h = \bigcup c \)
and \( M: M = \text{norm-pres-extensions } E \ p \ f \)
and \( cM: c \in \text{chains } M \)
shows \( \forall x \in H, \ h x \leq p x \)

proof

fix \( x \) assume \( x \in H \)
with \( M \in M \) graph obtain \( H' h' \) where \( x': x \in H' \)
and graphs: \( \text{graph } H' h' \subseteq \text{graph } H h \)
and \( a: \forall x \in H'. \ h' x \leq p x \)
by (rule some-H' h' [elim-format]) blast
from graphs \( x' \) have [symmetric]: \( h' x = h x \).
also from \( a \) \( x' \) have \( h' x \leq p x \).

finally show \( h x \leq p x \).

\[ \text{qed} \]

The following lemma is a property of linear forms on real vector spaces. It will be used for the lemma abs-Hahn-Banach (see page ??). For real vector spaces the following inequations are equivalent:

\[ \forall x \in H. \ |h x| \leq p x \quad \text{and} \quad \forall x \in H. \ h x \leq p x \]

**lemma** abs-ineq-iff:

assumes subspace \( H \ E \) and vectorspace \( E \) and seminorm \( E \ p \)
and linearform \( H \ h \)
shows \( (\forall x \in H. \ |h x| \leq p x) = (\forall x \in H. \ h x \leq p x) \) (is \( ?L = ?R \))

proof

interpret subspace \( H \ E \) by fact
interpret vectorspace \( E \) by fact
interpret seminorm \( E \ p \) by fact
interpret linearform \( H \ h \) by fact
have \( H: \text{vectorspace } H \) using (vectorspace \( E \)).

\[
\{ \\
\quad \text{assume } l: ?L \\
\quad \text{show } ?R \\
\quad \text{proof} \\
\quad \quad \text{fix } x \text{ assume } x: x \in H \\
\quad \quad \text{have } h x \leq |h x| \text{ by arith} \\
\quad \quad \text{also from } l \text{ have } \ldots \leq p x \ldots \\
\quad \quad \text{finally show } h x \leq p x . \\
\quad \text{qed} \\
\}
\]

next

\[
\text{assume } r: ?R \\
\text{show } ?L \\
\text{proof} \\
\quad \text{fix } x \text{ assume } x: x \in H \\
\]
show \( \forall a \ b :: \text{real.} - a \leq b \implies b \leq a \implies |b| \leq a \)
by arith
from (linearform H h) and H x
have \( - h x = h (- x) \) by (rule linearform.neg [symmetric])
also from H x have \(- x \in H \) by (rule vectorspace.neg-closed)
with \( r \) have \( h (- x) \leq p (- x) \) ..
also have \( \ldots = p x \)
using (seminorm E p) (vectorspace E)
proof (rule seminorm.minus)
from x show \( x \in E \) ..
qed
finally have \( - h x \leq p x \).
then show \( - p x \leq h x \) by simp
from \( r x \) show \( h x \leq p x \) ..
qed
}
qed
end

11 Extending non-maximal functions

theory Hahn-Banach-Ext-Lemmas
imports Function-Norm
begin

In this section the following context is presumed. Let \( E \) be a real vector space with a seminorm \( q \) on \( E \). \( F \) is a subspace of \( E \) and \( f \) a linear function on \( F \). We consider a subspace \( H \) of \( E \) that is a superspace of \( F \) and a linear form \( h \) on \( H \). \( H \) is not equal to \( E \) and \( x_0 \) is an element in \( E - H \). \( H \) is extended to the direct sum \( H' = H + \text{lin} \ x_0 \), so for any \( x \in H' \) the decomposition of \( x = y + a \cdot \xi \) with \( y \in H \) is unique. \( h' \) is defined on \( H' \) by \( h' x = h y + a \cdot \xi \) for a certain \( \xi \).

Subsequently we show some properties of this extension \( h' \) of \( h \).

This lemma will be used to show the existence of a linear extension of \( f \) (see page 47). It is a consequence of the completeness of \( \mathbb{R} \). To show

\[ \exists \xi. \forall y \in F. \ a \ y \leq \xi \ \land \ \xi \leq b \ y \]

it suffices to show that

\[ \forall u \in F. \forall v \in F. \ a \ u \leq b \ v \]

lemma ex-xi:
assumes vectorspace F
assumes \( r: \forall u \ v. \ u \in F \implies v \in F \implies a \ u \leq b \ v \)
shows \( \exists xi::\text{real.} \forall y \in F. \ a \ y \leq xi \ \land \ xi \leq b \ y \)
proof –
interpret vectorspace F by fact

From the completeness of the reals follows: The set \( S = \{ a \ u. \ u \in F \} \) has a supremum, if it is non-empty and has an upper bound.
let \( ?S = \{ a u | u. u \in F \} \)

have \( \exists x_1. \text{lub } ?S x_1 \)

proof (rule real-complete)

have \( a \ 0 \in \ ?S \) by blast

then show \( \exists \ X. \ X \in \ ?S \)

have \( \forall y \in \ ?S. \ y \leq b \ 0 \)

proof

fix \( y \) assume \( y \in \ ?S \)

then obtain \( u \) where \( u \in \ F \) and \( y = a \ u \) by blast

from \( u \) and \( \text{zero} \) have \( a \ u \leq b \ 0 \) by (rule r)

with \( y \) show \( y \leq b \ 0 \) by (simp only:)

qed

then show \( \exists x_1. \forall y \in ?S. \ y \leq u \)

qed

then obtain \( x_1 \) where \( x_1: \text{lub } ?S x_1 \)

\{
  fix \( y \) assume \( y \in \ F \)
  then have \( a \ y \in \ ?S \) by blast
  with \( x_1 \) have \( a \ y \leq x_1 \) by (rule lub.upper)
\}

moreover \{
  fix \( y \) assume \( y \in \ F \)
  from \( x_1 \) have \( x_1 \leq b \ y \)
  proof (rule lub.least)
    fix \( au \) assume \( au \in ?S \)
    then obtain \( u \) where \( u \in \ F \) and \( au = a \ u \) by blast
    from \( u \) \( y \) have \( a \ u \leq b \ y \) by (rule r)
    with \( au \) show \( au \leq b \ y \) by (simp only:)
    qed
  \}

ultimately show \( \exists x_1. \forall y \in ?S. \ y \leq x_1 \wedge x_1 \leq b \ y \) by blast

qed

The function \( h' \) is defined as \( h' \ x = h \ y + a \cdot \xi \) where \( x = y + a \cdot \xi \) is a linear extension of \( h \) to \( H' \).

lemma \( \ h'\)-lf:

assumes \( \ h'\)-def: \( h' \equiv \lambda x. \ \text{let } (y, a) = \ S \ O \ M E \ (y, a), \ x = y + a \cdot x0 \wedge y \in H \ \text{in } h \ y + a * \ x1 \)

and \( \ H'\)-def: \( H' \equiv H + \ \text{lin } x0 \)

and \( \ HE: \ H \subseteq E \)

assumes linearform \( H \ h \)

assumes \( x0: \ x0 \notin H \ x0 \in E \ x0 \neq 0 \)

assumes \( \ E: \ \text{vectorspace } E \)

shows linearform \( H' \ h' \)

proof —

interpret linearform \( H \ h \) by fact

interpret vectorspace \( E \) by fact

show \( ?\text{thesis} \)

proof

note \( E = \text{ (vectorspace } E) \)

have \( H': \text{vectorspace } H' \)

proof (unfold \( \ h'\)-def)

from \( (x0 \in E) \)

have \( \text{lin } x0 \subseteq E \)

with \( \ HE \) show vectorspace \( (H + \text{lin } x0) \) using \( E \)

qed
qed

\{ fix \ x1 \ x2 assume \ x1: \ x1 \in H' and \ x2: \ x2 \in H' show \ h'(x1 + x2) = h' x1 + h' x2 proof from H' x1 x2 have x1 + x2 \in H' by (rule vectorspace.add-closed) with x1 x2 obtain y y1 y2 a a1 a2 where x1x2: \ x1 + x2 = y + a \cdot x0 and y: y \in H and x1-rep: \ x1 = y1 + a1 \cdot x0 and y1: y1 \in H and x2-rep: \ x2 = y2 + a2 \cdot x0 and y2: y2 \in H unfolding H'-def sum-def lin-def by blast have ya: y1 + y2 = y \land a1 + a2 = a using E HE - y x0 proof (rule decomp-H') from HE y1 y2 show y1 + y2 \in H by (rule subspace.add-closed) from x0 and HE y y1 y2 have x0 \in E y \in E y1 \in E y2 \in E by auto with x1-rep x2-rep have (y1 + y2) + (a1 + a2) \cdot x0 = x1 + x2 by (simp add: add-ac add-mult-distrib2) also note x1x2 finally show (y1 + y2) + (a1 + a2) \cdot x0 = y + a \cdot x0 . qed from h'-def x1x2 E HE y x0 have h'(x1 + x2) = h y + a \cdot xi by (rule h'-definite) also have \ldots = h (y1 + y2) + (a1 + a2) \cdot xi by (simp only: ya) also from y1 y2 have h (y1 + y2) = h y1 + h y2 by simp also have \ldots + (a1 + a2) \cdot xi = (h y1 + a1 \cdot xi) + (h y2 + a2 \cdot xi) by (simp add: distrib-right) also from h'-def x1-rep E HE y1 x0 have h y1 + a1 \cdot xi = h' x1 by (rule h'-definite [symmetric]) also from h'-def x2-rep E HE y2 x0 have h y2 + a2 \cdot xi = h' x2 by (rule h'-definite [symmetric]) finally show ?thesis . qed next fix \ x1 \ c assume \ x1: \ x1 \in H' show h'(c \cdot x1) = c \cdot (h' x1) proof from H' x1 have ax1: c \cdot x1 \in H' by (rule vectorspace.mult-closed) with x1 obtain y a y1 a1 where cx1-rep: c \cdot x1 = y + a \cdot x0 and y: y \in H and x1-rep: x1 = y1 + a1 \cdot x0 and y1: y1 \in H unfolding H'-def sum-def lin-def by blast have ya: c \cdot y1 = y \land c \cdot a1 = a using E HE - y x0 proof (rule decomp-H')
The linear extension $h'$ of $h$ is bounded by the seminorm $p$.

**Lemma: $h'$-norm-pres:**
- **Assumptions:**
  - $h'$-def: $h' \equiv \lambda x. \text{let } (y, a) = \text{SOME } (y, a)$. $x = y + a \cdot x0 \land y \in H$ in $h \cdot y + a \cdot xi$
  - $H'$-def: $H' \equiv H + \text{lin } x0$
  - $x0$: $x0 \notin H \land x0 \in E \land x0 \neq 0$
  - $E$: vectorspace $E$ and $HE$: subspace $H E$
  - seminorm $E$ $p$ and linearform $H$ $h$
- **Shows:**
  - $\forall x \in H'. h' x \leq p x$

**Proof:**
- Interpret vectorspace $E$ by fact
- Interpret subspace $H E$ by fact
- Interpret seminorm $E$ $p$ by fact
- Interpret linearform $H$ $h$ by fact
- Show $\exists \text{thesis}$
- Proof
  - Fix $x$
  - Assume $x'$. $x \in H'$
  - Show $h' x \leq p x$
    - Proof
      - From $a'$ have $a1$:
        - $\forall ya \in H. - p (ya + x0) - h ya \leq xi$
        - and $a2$: $\forall ya \in H. xi \leq p (ya + x0) - h ya$ by auto
      - From $x'$ obtain $y$ $a$ where
        - $x$-rep: $x = y + a \cdot x0$ and $y \in H$
        - Unfolding $H'$-def sum-def lin-def by blast
      - From $y$ have $y'$:
        - $y \in E$ ..
      - From $y$ have $ay$: inverse $a \cdot y \in H$ by simp
In the case \( a < 0 \), we use \( a_1 \) with \( ya \) taken as \( y / a \):

Assume \( lz: a < 0 \) then have \( nz: a \neq 0 \) by simp.

From \( a1 ay \) have \(- p (inverse a \cdot y + x0) - h (inverse a \cdot y) \leq x1 \).

With \( lz \) have \( a * x1 \leq a * (- p (inverse a \cdot y + x0) - h (inverse a \cdot y)) \).

By simp: add-left-mono-neg order-less-imp-le.

Also have \( = a * (p (inverse a \cdot y + x0)) - a * (h (inverse a \cdot y)) \).

By simp: right-diff-distrib.

Also from \( lz x0 y' \) have \( a * p (inverse a \cdot y + x0) - a * h (inverse a \cdot y) \).

By simp: add-right-diff-distrib.

Also from \( nz x0 y' \) have \( a * p (inverse a \cdot y + x0) = p (a \cdot (inverse a \cdot y + x0)) \).

By simp: abs-homogenous.

Also from \( nz y \) have \( a * (h (inverse a \cdot y)) = h y \).

By simp.

Finally have \( a * x1 \leq p (y + a \cdot x0) - h y \).

Then show \( \text{thesis} \) by simp.

Next.

In the case \( a > 0 \), we use \( a_2 \) with \( ya \) taken as \( y / a \):

Assume \( gz: 0 < a \) then have \( nz: a \neq 0 \) by simp.

From \( a2 ay \) have \( x1 \leq p (inverse a \cdot y + x0) - h (inverse a \cdot y) \).

With \( gz \) have \( a * x1 \leq a * (p (inverse a \cdot y + x0) - h (inverse a \cdot y)) \).

By simp.

Also have \( = a * p (inverse a \cdot y + x0) - a * h (inverse a \cdot y) \).

By simp: right-diff-distrib.

Also from \( gz x0 y' \) have \( a * p (inverse a \cdot y + x0) = p (a \cdot (inverse a \cdot y + x0)) \).

By simp: abs-homogenous.

Also from \( nz x0 y' \) have \( = p (y + a \cdot x0) \).

By simp: add-right-diff-distrib.

Also from \( nz y \) have \( a * h (inverse a \cdot y) = h y \).

By simp.

Finally have \( a * x1 \leq p (y + a \cdot x0) - h y \).

Then show \( \text{thesis} \) by simp.

Qed.

Also from \( x-rep \) have \( = p x \) by (simp only:).
finally show $\theta$thesis .

qed

end
Part III
The Main Proof

12 The Hahn-Banach Theorem

theory Hahn-Banach
imports Hahn-Banach-Lemmas
begin

We present the proof of two different versions of the Hahn-Banach Theorem, closely following [1, §36].

12.1 The Hahn-Banach Theorem for vector spaces

Hahn-Banach Theorem. Let $F$ be a subspace of a real vector space $E$, let $p$ be a semi-norm on $E$, and $f$ be a linear form defined on $F$ such that $f$ is bounded by $p$, i.e. $\forall x \in F. \ f x \leq p x$. Then $f$ can be extended to a linear form $h$ on $E$ such that $h$ is norm-preserving, i.e. $h$ is also bounded by $p$.

Proof Sketch.

1. Define $M$ as the set of norm-preserving extensions of $f$ to subspaces of $E$. The linear forms in $M$ are ordered by domain extension.
2. We show that every non-empty chain in $M$ has an upper bound in $M$.
3. With Zorn’s Lemma we conclude that there is a maximal function $g$ in $M$.
4. The domain $H$ of $g$ is the whole space $E$, as shown by classical contradiction:
   - Assuming $g$ is not defined on whole $E$, it can still be extended in a norm-preserving way to a super-space $H'$ of $H$.
   - Thus $g$ can not be maximal. Contradiction!

theorem Hahn-Banach:
assumes $E$: vectorspace $E$ and subspace $F E$
and seminorm $E p$ and linearform $F f$
assumes $fp$: $\forall x \in F. \ f x \leq p x$
shows $\exists h. \ linearform E h \land (\forall x \in F. \ h x = f x) \land (\forall x \in E. \ h x \leq p x)$
   — Let $E$ be a vector space, $F$ a subspace of $E$, $p$ a seminorm on $E$,
   — and $f$ a linear form on $F$ such that $f$ is bounded by $p$,
   — then $f$ can be extended to a linear form $h$ on $E$ in a norm-preserving way.

proof –
interpret vectorspace $E$ by fact
interpret subspace $F E$ by fact
interpret seminorm $E p$ by fact
interpret linearform $F f$ by fact
def $M \equiv norm-pres-extensions E p F f$
then have $M$: $M = \ldots$ by (simp only:)
from $E$ have $F$: \textit{vectorspace} $F$ ..

\textbf{note} $FE = (F \subseteq E)$

\{
  \begin{enumerate}
    \item \textbf{fix} $c$ \textbf{assume} $cM$: $c \in \text{chains} \ M$ \textbf{and} $\exists x. \ x \in c$
    \item \textbf{have} $\bigcup c \in M$
      \begin{enumerate}
        \item Show that every non-empty chain $c$ of $M$ has an upper bound in $M$:
        \item $\bigcup c$ is greater than any element of the chain $c$, so it suffices to show $\bigcup c \in M$.
      \end{enumerate}
    \end{enumerate}

unfolding $M$-def

\textbf{proof} (rule \textit{norm-pres-extensionI})

\begin{enumerate}
  \item \textbf{let} $\forall H = \text{domain} \ (\bigcup c)$
  \item \textbf{let} $\forall h = \text{funct} \ (\bigcup c)$
\end{enumerate}

\begin{enumerate}
  \item \textbf{have} $a$: $\text{graph} \ ?H \ ?h = \bigcup c$
    \begin{enumerate}
      \item \textbf{proof} (rule \textit{graph-domain-funct})
      \begin{enumerate}
        \item \textbf{fix} $x \ y \ z$ \textbf{assume} $(x, y) \in \bigcup c \ \text{and} \ (x, z) \in \bigcup c$
        \item \textbf{with} $M$-def $cM$ \textbf{show} $z = y$ by (rule \textit{sup-definite})
      \end{enumerate}
    \end{enumerate}
  \end{enumerate}

\textbf{qed}

moreover from $M$ $cM$ \textbf{a have} \textit{linearform} $?H \ ?h$

by (rule \textit{sup-if})

moreover from $a$ $M$ $cM$ \textbf{ex} $FE$ $E$ \textbf{have} $?H \subseteq E$

by (rule \textit{sup-subE})

moreover from $a$ $M$ $cM$ \textbf{ex} $FE$ \textbf{have} $F \subseteq ?H$

by (rule \textit{sup-supF})

moreover from $a$ $M$ $cM$ \textbf{have} $\text{graph} \ F \ f \subseteq \text{graph} \ ?H \ ?h$

by (rule \textit{sup-ext})

moreover from $a$ $M$ $cM$ \textbf{have} $\forall x \in ?H. \ ?h x \leq p x$

by (rule \textit{sup-norm-pres})

ultimately \textbf{show} $\exists H. h. \ \bigcup c = \text{graph} \ H \ h$

\begin{enumerate}
  \item $\land \ \textit{linearform} \ H \ h$
  \item $\land \ H \subseteq E$
  \item $\land \ F \subseteq H$
  \item $\land \ \text{graph} \ F \ f \subseteq \text{graph} \ H \ h$
  \item $\land \ (\forall x \in H. \ h \ x \leq p x)$ \textbf{by blast}
\end{enumerate}

\textbf{qed}

\}

\textbf{then have} $\exists g \in M. \ \forall x \in M. \ g \subseteq x \rightarrow x = g$

-- With \textit{Zorn}'s Lemma we can conclude that there is a maximal element in $M$.

\textbf{proof} (rule \textit{Zorn}'s-Lemma)

-- We show that $M$ is non-empty:

\textbf{show} $\text{graph} \ F \ f \in M$

unfolding $M$-def

\textbf{proof} (rule \textit{norm-pres-extensionI2})

\textbf{show} $\textit{linearform} \ F \ f$ \textbf{by fact}

\textbf{show} $F \subseteq E$ \textbf{by fact}

from $F$ \textbf{show} $\exists F \subseteq F$ \textbf{by} (rule \textit{vectorspace.subspace-refl})

\textbf{show} $\text{graph} \ F \ f \subseteq \text{graph} \ F \ f$ ..

\textbf{show} $\forall x \in F. \ f \ x \leq p x$ \textbf{by fact}

\textbf{qed}

\textbf{qed}

\textbf{then obtain} $g$ \textbf{where} $gM$: $g \in M$ \textbf{and} $gx: \forall x \in M. \ g \subseteq x \rightarrow g = x$

by blast

from $gM$ \textbf{obtain} $H \ h$ \textbf{where}

\textit{g-rep}: $g = \text{graph} \ H \ h$

and \textit{linearform}: $\textit{linearform} \ H \ h$
12.1 The Hahn-Banach Theorem for vector spaces

and $HE$: $H \subseteq E$ and $FH$: $F \subseteq H$

and graphs: $\text{graph } F \cap \text{graph } H$

and $hp$: $\forall x \in H. \, h \leq p \, x$ unfolding $M$-def ..

— $g$ is a norm-preserving extension of $f$, in other words:
— $g$ is the graph of some linear form $h$ defined on a subspace $H$ of $E$,
— and $h$ is an extension of $f$ that is again bounded by $p$.

from $HE$ have $H$: vectorspace $H$

by (rule subspace.vectorspace)

have $HE$-eq: $H = E$

— We show that $h$ is defined on whole $E$ by classical contradiction.

proof (rule classical)

assume $\neg H \neq E$

— Assume $h$ is not defined on whole $E$. Then show that $h$ can be extended
— in a norm-preserving way to a function $h'$ with the graph $g'$.

have $\exists g' \in M. \, g \subseteq g' \land g \neq g'$

proof —

from $HE$ have $H \subseteq E$ ..

with $\neg H$ obtain $x'$ where $x' \in E$ and $x' \notin H$ by blast

obtain $x': x' \neq 0$

proof

show $x' \neq 0$

proof

assume $x' = 0$

with $H$ have $x' \in H$ by (simp only: vectorspace.zero)

with ($x' \notin H$) show False by contradiction

qed

qed

def $H' \equiv H + \text{lin } x'$

— Define $H'$ as the direct sum of $H$ and the linear closure of $x'$.

have $HH'$: $H \subseteq H'$

proof (unfold $H'$-def)

from $x' \in E$ have vectorspace ($\text{lin } x'$) ..

with $H$ show $H \subseteq H + \text{lin } x'$ ..

qed

obtain $xi$ where

$xi$: $\forall y \in H. \, -p \, (y + x') - h \, y \leq xi$

$\land \, xi \leq p \, (y + x') - h \, y$

— Pick a real number $\xi$ that fulfills certain inequations; this will
— be used to establish that $h'$ is a norm-preserving extension of $h$.

proof —

from $H$ have $\exists \xi. \, \forall y \in H. \, -p \, (y + x') - h \, y \leq \xi$

$\land \, \xi \leq p \, (y + x') - h \, y$

proof (rule ex-xi)

fix $u v$ assume $u: u \in H$ and $v: v \in H$

with $HE$ have $u \in E$ and $v \in E$ by auto

from $H \, u \, v$ linearform have $h \, v - h \, u = h \, (v - u)$

by (simp add: linearform.diff)

also from $hp$ and $H \, u \, v$ have $\ldots \leq p \, (v - u)$

by (simp only: vectorspace.diff-closed)

also from $x' \in E \, u \, v \in E$ have $v - u = x' + - x' + v + - u$
by (simp add: diff-eq1)
also from \( x' \in E \) \( u \in E \) \( v \in E \) have \( \ldots = v + x' + - (u + x') \)
by (simp add: add-ac)
also from \( x' \in E \) \( u \in E \) \( v \in E \) have \( \ldots = (v + x') - (u + x') \)
by (simp add: diff-eq1)
also from \( x' \in E \) \( u \in E \) \( E \) have \( p \ldots \leq p (v + x') + p (u + x') \)
by (simp add: diff-subadditive)
finally have \( h v - h u \leq p (v + x') - h u \) by simp
qed
then show \( thesis \) by (blast intro: that)
qed

def \( h' \equiv \lambda x. \) let \( \langle y, a \rangle = \) SOME \( (y, a) \) \( x = y + a \cdot x' \land y \in H \) in \( h y \) 
— Define the extension \( h' \) of \( h \) to \( H' \) using \( \xi \).

have \( g \subseteq \text{graph } H' h' \land g \neq \text{graph } H' h' \)
— \( h' \) is an extension of \( h \) . . .
proof
show \( g \subseteq \text{graph } H' h' \)
proof —
\[ \begin{aligned}
& \text{have } \text{graph } H h \subseteq \text{graph } H' h' \\
& \text{proof } \text{rule graph-extI} \\
& \text{fix } t \text{ assume } t: t \in H \\
& \text{from } E HE t \text{ have } (\text{SOME } (y, a), t = y + a \cdot x' \land y \in H) = (t, 0) \\
& \text{using } (x' \notin H) \quad (x' \in E \land x' \neq 0) \text{ by } \text{rule } \text{decomp-H'-H} \\
& \text{with } h'\text{-def show } h t = h' t \text{ by } (\text{simp add: Let-def}) \\
& \text{next }
\end{aligned} \]
then show \( H \subseteq H' \) ..
qed
with g-rep show \( \text{thesis by } (\text{simp only;}) \)
qed

show \( g \neq \text{graph } H' h' \)
proof —
\[ \begin{aligned}
& \text{have } \text{graph } H h \neq \text{graph } H' h' \\
& \text{proof } \\
& \text{assume eq: graph } H h = \text{graph } H' h' \\
& \text{have } x' \in H' \\
& \text{unfolding } H'\text{-def} \\
& \text{proof } \\
& \text{from } H \text{ show } 0 \in H \text{ by } (\text{rule vectorspace.zero}) \\
& \text{from } x' \in \text{lin } x' \text{ by } (\text{rule } x\text{-lin-x}) \\
& \text{from } x' \in \text{lin } x' \text{ by } \text{simp} \\
& \text{qed} \\
& \text{then have } (x', h' x') \in \text{graph } H' h' .. \\
& \text{with eq have } (x', h' x') \in \text{graph } H h \text{ by } (\text{simp only;}) \\
& \text{then have } x' \in H .. \\
& \text{with } (x' \notin H) \text{ show False by contradiction} \\
& \text{qed} \\
& \text{with g-rep show } \text{thesis by simp} \\
& \text{qed} \\
& \text{qed} \\
& \text{qed}
\end{aligned} \]
Moreover have graph $H' h' \in M$

— and $h'$ is norm-preserving.

**proof** (unfold $M$-def)

show graph $H' h' \in \text{norm-pres-extensions } E p F f$

**proof** (rule $\text{norm-pres-extensionI2}$)

show linearform $H' h'$

using $h'$-def $H'$-def $HE$ linearform $\langle x' \notin H \rangle \langle x' \in E \rangle \langle x' \neq 0 \rangle E$

by (rule $h'$-lf)

show $H' \subseteq E$

unfolding $H'$-def

**proof**

show $H \leq E$ by fact

show vectorspace $E$ by fact

from $x' E$ show $\text{lin } x' \leq E$ ..

qed

from $H \langle F \leq H \rangle \langle HH' \rangle$ show $FH' \subseteq F \leq H'$

by (rule vectorspace.subspace-trans)

show graph $F f \subseteq \text{graph } H' h'$

**proof** (rule graph-extI)

fix $x$ assume $x : x \in F$

with graphs have $f x = h x$ ..

also have $\ldots = h x + 0 * xi$ by simp

also have $\ldots = (\text{let } (y, a) = (x, 0)) \in h y + a * xi$

by (simp add: Let-def)

also have $(x, 0) = (\text{SOME } (y, a). x = y + a \cdot x' \land y \in H)$

using $E HE$

**proof** (rule $\text{decomp-H'} \langle H \rangle \langle \text{symmetric} \rangle$

from $FH x$ show $x \in H$ ..

from $x' \notin H$ by fact

show $x' \in E$ by fact

qed

also have $(\text{let } (y, a) = (\text{SOME } (y, a). x = y + a \cdot x' \land y \in H))$

in $h y + a * xi = h' x$ by (simp only: $h'$-def)

finally show $f x = h' x$.

next

from $FH' \subseteq H'$ ..

qed

show $\forall x \in H'. h' x \leq p x$

using $h'$-def $H'$-def $\langle x' \notin H \rangle \langle x' \in E \rangle \langle x' \neq 0 \rangle E HE$

$\langle \text{seminorm } E p \rangle$ linearform and $hp xi$

by (rule $h'$-norm-pres)

qed

qed

ultimately show ?thesis ..

qed

then have $\lnot (\forall x \in M. g \subseteq x \longrightarrow g = x)$ by simp

— So the graph $g$ of $h$ cannot be maximal. Contradiction!

with $gx$ show $H = E$ by contradiction

qed

from $HE$-eq and linearform have linearform $E h$
by \((\text{simp only:})\)
moreover have \(\forall x \in F. \, h x = f x\)
proof
fix \(x\) assume \(x \in F\)
with graphs have \(f x = h x\)
then show \(h x = f x\)
qed
moreover from HE-eq and hp have \(\forall x \in E. \, h x \leq p x\)
by \((\text{simp only:})\)
ultimately show ?thesis by blast
qed

12.2 Alternative formulation

The following alternative formulation of the Hahn-Banach Theorem uses the fact that for a real linear form \(f\) and a seminorm \(p\) the following inequations are equivalent:\(^1\)

\[
\forall x \in H. \, |h x| \leq p x \quad \text{and} \quad \forall x \in H. \, h x \leq p x
\]

**Theorem abs-Hahn-Banach:**
assumes \(E: \text{vectorspace } E\) and \(FE: \text{subspace } F E\)
and \(lf: \text{linearform } F f\) and \(sn: \text{seminorm } E p\)
assumes \(fp: \forall x \in F. \, \|f x\| \leq p x\)
shows \(\exists g. \, \text{linearform } E g\)
\(\land (\forall x \in F. \, g x = f x)\)
\(\land (\forall x \in E. \, |g x| \leq p x)\)
proof (rule Hahn-Banach)
show \(\forall x \in F. \, f x \leq p x\)
using \(FE E sn\) if
proof (rule abs-ineq-iff \([\text{THEN iffD1}]\))
qed
then obtain \(g\) where \(lg: \text{linearform } E g\) and \(\ast: \forall x \in F. \, g x = f x\)
\(\land \ast\ast: \forall x \in E. \, g x \leq p x\) by blast
have \(\forall x \in E. \, |g x| \leq p x\)
using \(- E sn lg \ast\ast\)
proof (rule abs-ineq-iff \([\text{THEN iffD2}]\))
show \(E \leq E\)
qed
with \(lg \ast\) show ?thesis by blast
qed

12.3 The Hahn-Banach Theorem for normed spaces

Every continuous linear form \(f\) on a subspace \(F\) of a norm space \(E\), can be extended to a continuous linear form \(g\) on \(E\) such that \(\|f\| = \|g\|\).

\(^1\)This was shown in lemma abs-ineq-iff (see page 38).
12.3 The Hahn-Banach Theorem for normed spaces

**Theorem** norm-Hahn-Banach:

`fixes V and norm (∥∥)
fixes B defines ∩ V f. B V f ≡ {0} ∪ {∥x∥ / x. x ≠ 0 ∧ x ∈ V}
defines ∩ V f. ∥∥-∥∥ V f ≡ 0 ∪ {∥f x∥ / x. x ≠ 0 ∧ x ∈ V}
defines ∩ V f. ∥∥-F ≡ ∪ (B V f)
assumes E-norm: normed-vectorspace E norm and FE: subspace F E
and linearform: linearform F f and continuous F f norm
shows ∃ g. linearform E g ∧ continuous E g norm ∧ (∀ x ∈ F. g x = f x) ∧ ∥g∥-E = ∥f∥-F

**Proof**

`interpret normed-vectorspace E norm by fact`
`interpret normed-vectorspace-with-fn-norm E norm B fn-norm by (auto simp: B-def fn-norm-def intro-locales)
interpret subspace F E by fact
interpret linearform F f by fact
interpret continuous F f norm by fact
have E: vectorspace E by intro-locales
have F: vectorspace F by rule intro-locales
have F-norm: normed-vectorspace F norm
using FE E-norm by (rule subspace-normed-vs)
have ge-zero: 0 ≤ ∥f∥-F by (rule normed-vectorspace-with-fn-norm.fn-norm-ge-zero
  [OF normed-vectorspace-with-fn-norm.intro,
   OF F-norm ⟨continuous F f norm⟩, folded B-def fn-norm-def])

We define a function p on E as follows: p x = ∥f∥-F * ∥x∥

`def p ≡ λx. ∥f∥-F * ∥x∥`

p is a seminorm on E:

`have q: seminorm E p
proof
  fix x y a assume x: x ∈ E and y: y ∈ E

p is positive definite:

  have 0 ≤ ∥f∥-F by (rule ge-zero)
  moreover from x have 0 ≤ ∥x∥ ..
  ultimately show 0 ≤ p x
    by (simp add: p-def zero-le-mult-iff)

p is absolutely homogeneous:

  show p (a · x) = |a| * p x
  proof
    have p (a · x) = ∥f∥-F * ∥a · x∥ by (simp only: p-def)
    also from x have ∥a · x∥ = |a| * ∥x∥ by (rule abs-homogenous)
    also have ∥∥-F * (|a| * ∥x∥) = |a| * (∥f∥-F * ∥x∥) by simp
    also have ... = |a| * p x by (simp only: p-def)
    finally show thesis .
    qed

Furthermore, p is subadditive:

  show p (x + y) ≤ p x + p y`
proof
have \( p(x + y) = \|f\| - F \ast \|x + y\| \) by (simp only: p-def)
also have \( a : 0 \leq \|f\| - F \ast \|x\| \) by (rule ge-zero)
from \( x \) \( y \) have \( \|x + y\| \leq \|x\| + \|y\| \) ..
with \( a \) have \( \|f\| - F \ast \|x\| \leq \|f\| - F \ast (\|x\| + \|y\|) \)
by (simp add: mult-left-mono)
also have \( \ldots = \|f\| - F \ast \|x\| + \|f\| - F \ast \|y\| \) by (simp only: distrib-left)
also have \( \ldots = p \) \( x \) \( + \) \( p \) \( y \) by (simp only: p-def)
finally show \( \text{thesis} \) .
qed
qed

\( f \) is bounded by \( p \).

have \( \forall x \in F. \ |f| x \leq p \) \( x \)
proof
fix \( x \) assume \( x \in F \)
with \( \langle \text{continuous } F \ f \ \text{norm} \rangle \) and \( \text{linearform} \)
show \( |f| x \leq p \) \( x \)
unfolding \( p\)-def by (rule normed-vectorspace-with-fn-norm.fn-norm-le-cong
\[ \text{OF normed-vectorspace-with-fn-norm.fn-norm-le-cong} \]
\[ \text{OF F-norm, folded B-def fn-norm-def} \])
qed

Using the fact that \( p \) is a seminorm and \( f \) is bounded by \( p \) we can apply the Hahn-Banach Theorem for real vector spaces. So \( f \) can be extended in a norm-preserving way to some function \( g \) on the whole vector space \( E \).

with \( E \) \( F \) \( E \) \text{linearform} \( q \) obtain \( g \) where
\( \text{linearformE}: \text{linearform} E \ g \)
and \( a : \forall x \in F. \ g \ x = f \ x \)
and \( b : \forall x \in E. \ |g \ x| \leq p \) \( x \)
by (rule abs-Hahn-Banach [elim-format]) iprover

We furthermore have to show that \( g \) is also continuous:

have \( g\)-cont: \( \text{continuous } E \ g \ \text{norm} \) using \( \text{linearformE} \)
proof
fix \( x \) assume \( x \in E \)
with \( b \) show \( |g \ x| \leq \|f\| - F \ast \|x\| \)
by (simp only: p-def)
qed

To complete the proof, we show that \( \|g\| = \|f\| \).

have \( \|g\|-E = \|f\|-F \)
proof (rule order-antisym)
First we show \( \|g\| \leq \|f\| \). The function norm \( \|g\| \) is defined as the smallest \( c \in \mathbb{R} \) such that
\[ \forall x \in E. \ |g \ x| \leq c \cdot \|x\| \]
Furthermore holds
\[ \forall x \in E. \ |g \ x| \leq \|f\| \cdot \|x\| \]
from \( g\)-cont - ge-zero
show $\|g\|_E \leq \|f\|_F$  
proof  
  fix $x$ assume $x \in E$  
  with $b$ show $|g x| \leq \|f\|_F \ast \|x\|$  
  by (simp only: $p$-def)  
qed  

The other direction is achieved by a similar argument.  

show $\|f\|_F \leq \|g\|_E$  
proof (rule normed-vectorspace-with-fn-norm-fn-norm-least  
  [OF normed-vectorspace-with-fn-norm.intro,  
  OF F-norm, folded B-def fn-norm-def])  
fix $x$ assume $x : x \in F$  
show $|f x| \leq \|g\|_E \ast \|x\|$  
proof  
  from $a \ x$ have $g x = f x$ ..  
  then have $|f x| = |g x|$ by (simp only:)  
  also from $g$-cont  
  have $\ldots \leq \|g\|_E \ast \|x\|$  
  proof (rule fn-norm-le-cong [OF $\cdot$ linearformE, folded B-def fn-norm-def])  
  from $FE \ x$ show $x \in E$ ..  
  qed  
  finally show $?thesis$ .  
  qed  
next  
show $0 \leq \|g\|_E$  
using $g$-cont  
by (rule fn-norm-ge-zero [of $g$, folded B-def fn-norm-def])  
show continuous $F \ f \ norm$ by fact  
qed  
qed  
with linearformE $a \ g$-cont show $?thesis$ by blast  
qed  

end

References

