# $L^{p}$ spaces in Isabelle 

Sebastien Gouezel


#### Abstract

$L^{p}$ is the space of functions whose $p$-th power is integrable. It is one of the most fundamental Banach spaces that is used in analysis and probability. We develop a framework for function spaces, and then implement the $L^{p}$ spaces in this framework using the existing integration theory in Isabelle/HOL. Our development contains most fundamental properties of $L^{p}$ spaces, notably the Hölder and Minkowski inequalities, completeness of $L^{p}$, duality, stability under almost sure convergence, multiplication of functions in $L^{p}$ and $L^{q}$, stability under conditional expectation.


## Contents

1 Functions as a real vector space ..... 1
2 Quasinorms on function spaces ..... 3
2.1 Definition of quasinorms ..... 4
2.2 The space and the zero space of a quasinorm ..... 7
2.3 An example: the ambient norm in a normed vector space ..... 11
2.4 An example: the space of bounded continuous functions from a topological space to a normed real vector space ..... 12
2.5 Continuous inclusions between functional spaces ..... 14
2.6 The intersection and the sum of two functional spaces ..... 17
2.7 Topology ..... 23
3 Conjugate exponents ..... 30
4 Convexity inequalities and integration ..... 33
$5 L^{p}$ spaces ..... 39
$5.1 \quad L^{\infty}$ ..... 40
$5.2 L^{p}$ for $0<p<\infty$ ..... 45
5.3 Specialization to $L^{1}$ ..... 50
$5.4 L^{0}$ ..... 51
5.5 Basic results on $L^{p}$ for general $p$ ..... 52
$5.6 L^{p}$ versions of the main theorems in integration theory ..... 56
5.7 Completeness of $L^{p}$ ..... 64
5.8 Multiplication of functions, duality ..... 68
5.9 Conditional expectations and $L^{p}$ ..... 80
theory Functional-Spaces
importsHOL-Analysis.AnalysisHOL-Library.Function-AlgebrasErgodic-Theory.SG-Library-Complement
begin

## 1 Functions as a real vector space

Many functional spaces are spaces of functions. To be able to use the following framework, spaces of functions thus need to be endowed with a vector space structure, coming from pointwise addition and multiplication.

Some instantiations for fun are already given in Function_Algebras.thy and Lattices.thy, we add several.
minus_fun is already defined, in Lattices.thy, but under the strange name fun_Compl_def. We restate the definition so that unfolding minus_fun_def works. Same thing for minus_fun_def. A better solution would be to have a coherent naming scheme in Lattices.thy.
lemmas uminus-fun-def $=$ fun-Compl-def
lemmas minus-fun-def $=$ fun-diff-def
lemma fun-sum-apply:
fixes $u:: ' i \Rightarrow{ }^{\prime} a \Rightarrow(' b::$ comm-monoid-add $)$
shows (sum uI) $x=\operatorname{sum}(\lambda i$. u $i x) I$
by (induction I rule: infinite-finite-induct, auto)
instantiation fun :: (type, real-vector) real-vector
begin
definition scaleR-fun::real $\Rightarrow\left({ }^{\prime} a \Rightarrow{ }^{\prime} b\right) \Rightarrow{ }^{\prime} a \Rightarrow{ }^{\prime} b$
where scaleR-fun $=\left(\lambda c f .\left(\lambda x . c *_{R} f x\right)\right)$
lemma scale $R$-apply [simp, code]: $\left(c *_{R} f\right) x=c *_{R}(f x)$
by (simp add: scaleR-fun-def)
instance by (standard, auto simp add: scaleR-add-right scaleR-add-left)
end
lemmas divide $R$-apply $=$ scale $R$-apply
lemma [measurable]:

```
    0 \in borel-measurable M
unfolding zero-fun-def by auto
lemma borel-measurable-const-scaleR' [measurable (raw)]:
    (f::(' }a=>\mathrm{ 'b::real-normed-vector )) }\in\mathrm{ borel-measurable M > c**R}f\in\mathrm{ borel-measurable
M
unfolding scaleR-fun-def using borel-measurable-add by auto
lemma borel-measurable-add'[measurable (raw)]:
    fixes fg ::' }a=>\mathrm{ ' 'b::{second-countable-topology,real-normed-vector}
    assumes f:f\in borel-measurable M
    assumes g: g\in borel-measurable M
    shows }f+g\in\mathrm{ borel-measurable M
unfolding plus-fun-def using assms by auto
lemma borel-measurable-uminus'[measurable (raw)]:
    fixes fg :: 'a = 'b::{second-countable-topology, real-normed-vector}
    assumes f: f\in borel-measurable M
    shows -f f borel-measurable M
unfolding fun-Compl-def using assms by auto
lemma borel-measurable-diff '[measurable (raw)]:
    fixes fg ::' }a>>'b::{\mathrm{ second-countable-topology, real-normed-vector}
    assumes f:f\inborel-measurable M
    assumes g: g\in borel-measurable M
    shows f-g\in borel-measurable M
unfolding fun-diff-def using assms by auto
lemma borel-measurable-sum'[measurable (raw)]:
    fixes f::'i=>'a=>'b::{second-countable-topology, real-normed-vector}
    assumes }\bigwedgei.i\inI\Longrightarrowfi\in\mathrm{ borel-measurable M
    shows (\sumi\inI.fi)\in borel-measurable M
using borel-measurable-sum[of If, OF assms] unfolding fun-sum-apply[symmetric]
by simp
lemma zero-applied-to [simp]:
    (0::('a m ('b::real-vector))) x = 0
unfolding zero-fun-def by simp
```


## 2 Quasinorms on function spaces

A central feature of modern analysis is the use of various functional spaces, and of results of functional analysis on them. Think for instance of $L^{p}$ spaces, of Sobolev or Besov spaces, or variations around them. Here are several relevant facts about this point of view:

- These spaces typically depend on one or several parameters. This makes it difficult to play with type classes in a system without depen-
dent types.
- The $L^{p}$ spaces are not spaces of functions (their elements are equivalence classes of functions, where two functions are identified if they coincide almost everywhere). However, in usual analysis proofs, one takes a definite representative and works with it, never going to the equivalence class point of view (which only becomes relevant when one wants to use the fact that one has a Banach space at our disposal, to apply functional analytic tools).
- It is important to describe how the spaces are related to each other, with respect to inclusions or compact inclusions. For instance, one of the most important theorems in analysis is Sobolev embedding theorem, describing when one Sobolev space is included in another one. One also needs to be able to take intersections or sums of Banach spaces, for instance to develop interpolation theory.
- Some other spaces play an important role in analysis, for instance the weak $L^{1}$ space. This space only has a quasi-norm (i.e., its norm satisfies the triangular inequality up to a fixed multiplicative constant). A general enough setting should also encompass this kind of space. (One could argue that one should also consider more general topologies such as Frechet spaces, to deal with Gevrey or analytic functions. This is true, but considering quasi-norms already gives a wealth of applications).

Given these points, it seems that the most effective way of formalizing this kind of question in Isabelle/HOL is to think of such a functional space not as an abstract space or type, but as a subset of the space of all functions or of all distributions. Functions that do not belong to the functional space under consideration will then have infinite norm. Then inclusions, intersections, and so on, become trivial to implement. Since the same object contains both the information about the norm and the space where the norm is finite, it conforms to the customary habit in mathematics of identifying the two of them, talking for instance about the $L^{p}$ space and the $L^{p}$ norm.
All in all, this approach seems quite promising for "real life analysis".

### 2.1 Definition of quasinorms

typedef (overloaded) ('a::real-vector) quasinorm $=\{(C::$ real, $N::(' a \Rightarrow$ ennreal $)$ ). ( $C \geq 1$ )
$\wedge\left(\forall x c . N\left(c *_{R} x\right)=\right.$ ennreal $\left.|c| * N(x)\right) \wedge(\forall x y . N(x+y) \leq C * N x+$ $C * N y)\}$
morphisms Rep-quasinorm quasinorm-of proof
show $(1,(\lambda x .0)) \in\{(C:$ :real, $N::(' a \Rightarrow$ ennreal $)) .(C \geq 1)$

```
        \wedge(\forallxc.N(c**R x) = ennreal |c|*Nx)^(\forallxy.N(x+y)\leqC*Nx+
C*Ny)}
    by auto
qed
definition eNorm::'a quasinorm }=>\mathrm{ ('a::real-vector) }=>\mathrm{ ennreal
    where eNorm Nx=(snd (Rep-quasinorm N)) x
definition defect::('a::real-vector) quasinorm }=>\mathrm{ real
    where defect N=fst(Rep-quasinorm N)
lemma eNorm-triangular-ineq:
    eNorm N (x+y)\leqdefect N*eNorm N x + defect N*eNorm Ny
unfolding eNorm-def defect-def using Rep-quasinorm[of N] by auto
lemma defect-ge-1:
    defect N\geq1
unfolding defect-def using Rep-quasinorm[of N] by auto
lemma eNorm-cmult:
    eNorm N (c**R x) = ennreal |c|* eNorm Nx
unfolding eNorm-def using Rep-quasinorm[of N] by auto
lemma eNorm-zero [simp]:
    eNorm N 0 = 0
by (metis eNorm-cmult abs-zero ennreal-0 mult-zero-left real-vector.scale-zero-left)
lemma eNorm-uminus [simp]:
    eNorm N (-x) = eNorm Nx
using eNorm-cmult[of N-1 x] by auto
lemma eNorm-sum:
    eNorm N(\sumi\in{..<n}.u i)\leq(\sumi\in{..<n}.(defect N)`(Suc i)*eNorm N(u
i))
proof (cases n=0)
    case True
    then show ?thesis by simp
next
    case False
    then obtain m}\mathrm{ where n=Suc m using not0-implies-Suc by blast
    have \v. eNorm N (\sumi\in{..n}.vi)\leq(\sumi\in{..<n}. (defect N)^(Suc i)*eNorm
N(vi))+(defect N)^n* eNorm N (v n) for n
    proof (induction n)
        case 0
        then show?case by simp
    next
        case (Suc n)
        have *: (defect N )` (Suc n) = (defect N)`n * ennreal(defect N)
            by (metis defect-ge-1 ennreal-le-iff ennreal-neg ennreal-power less-le not-less
```

not-one-le-zero semiring-normalization-rules(28))
fix $v:$ : nat $\Rightarrow{ }^{\prime} a$
define $w$ where $w=(\lambda i$. if $i=n$ then $v n+v$ (Suc n) else $v i)$
have $\left(\sum i \in\{.\right.$. Suc $\left.n\} . v i\right)=\left(\sum i \in\{. .<n\} . v i\right)+v n+v($ Suc $n)$ using lessThan-Suc-atMost sum.lessThan-Suc by auto
also have $\ldots=\left(\sum i \in\{. .<n\} . w i\right)+w n$ unfolding $w$-def by auto
finally have $\left(\sum i \in\{. . S u c n\} . v i\right)=\left(\sum i \in\{. . n\} . w i\right)$ by (metis lessThan-Suc-atMost sum.lessThan-Suc)
then have eNorm $N\left(\sum i \in\{. . S u c n\} . v i\right)=e \operatorname{Norm} N\left(\sum i \in\{. . n\} . w i\right)$ by simp
also have $\ldots \leq\left(\sum i \in\{. .<n\}\right.$. $($ defect $N) \uparrow($ Suc $\left.i) * e N o r m ~ N(w i)\right)+($ defect $N) \wedge n * \operatorname{Norm} N(w n)$ using Suc.IH by auto
also have $\ldots=\left(\sum i \in\{. .<n\} .(\right.$ defect $N) \uparrow($ Suc $i) *$ eNorm $\left.N(v i)\right)+($ defect $N) \widehat{ } n * \operatorname{eNorm} N(v n+v($ Suc $n))$
unfolding $w$-def by auto
also have $\ldots \leq\left(\sum i \in\{. .<n\}\right.$. $($ defect $N) \wedge($ Suc $i) *$ eNorm $\left.N(v i)\right)+$
$(\operatorname{defect} N) \widehat{n} *(\operatorname{defect} N *$ eNorm $N(v n)+\operatorname{defect} N * e N o r m ~ N(v($ Suc
n)))
by (rule add-mono, simp, rule mult-left-mono, auto simp add: eNorm-triangular-ineq)
also have $\ldots=\left(\sum i \in\{. .<n\}\right.$. $($ defect $N) \uparrow($ Suc $\left.i) * e \operatorname{Norm} N(v i)\right)$
$+($ defect $N) \uparrow($ Suc $n) * \operatorname{eNorm} N(v n)+(\operatorname{defect} N) \uparrow($ Suc $n) * e N o r m ~ N$ (v (Suc n))
unfolding $*$ by (simp add: distrib-left semiring-normalization-rules(18))
also have $\ldots=\left(\sum i \in\{. .<\right.$ Suc $n\} .($ defect $N) \uparrow($ Suc $\left.i) * \operatorname{eNorm} N(v i)\right)+$ $(\operatorname{defect} N) \uparrow($ Suc $n) *$ eNorm $N(v($ Suc $n))$
by auto
finally show eNorm $N\left(\sum i \in\{. . S u c n\} . v i\right)$
$\leq\left(\sum i<S u c\right.$ n. ennreal (defect $N^{\wedge}$ Suc $\left.i\right) *$ eNorm $\left.N(v i)\right)+$ ennreal $($ defect $N \wedge$ Suc $n) *$ eNorm $N(v($ Suc $n))$
by $\operatorname{simp}$
qed
then have $e \operatorname{Norm} N\left(\sum i \in\{. .<\right.$ Suc $\left.m\} . u i\right)$
$\leq\left(\sum i \in\{. .<m\} .(\right.$ defect $N) \uparrow($ Suc $\left.i) * \operatorname{eNorm} N(u i)\right)+(\operatorname{defect} N) \uparrow m *$ eNorm $N$ (u m)
using lessThan-Suc-atMost by auto
also have $\ldots \leq\left(\sum i \in\{. .<m\}\right.$. (defect $\left.N\right) \uparrow($ Suc $i) *$ eNorm $\left.N(u \quad i)\right)+($ defect
$N) \uparrow(S u c m) * e N o r m ~ N(u m)$
apply (rule add-mono, auto intro!: mult-right-mono ennreal-leI)
using defect-ge-1 by (metis atMost-iff le-less lessThan-Suc-atMost lessThan-iff power-Suc power-increasing)
also have $\ldots=\left(\sum i \in\{. .<\right.$ Suc $m\} .($ defect $N) \uparrow($ Suc $\left.i) * \operatorname{eNorm} N(u i)\right)$
by auto
finally show eNorm $N\left(\sum i \in\{. .<n\} . u i\right) \leq\left(\sum i<n\right.$. ennreal (defect $N^{\wedge}$ Suc $\left.i\right)$ * eNorm $N(u \quad i))$
unfolding $\langle n=$ Suc $m\rangle$ by auto
qed
Quasinorms are often defined by taking a meaningful formula on a vector
subspace, and then extending by infinity elsewhere. Let us show that this results in a quasinorm on the whole space.

```
definition quasinorm-on::('a set) \(\Rightarrow\) real \(\Rightarrow((' a::\) real-vector \() \Rightarrow\) ennreal \() \Rightarrow\) bool
    where quasinorm-on \(F C N=(\)
    \((\forall x y .(x \in F \wedge y \in F) \longrightarrow(x+y \in F) \wedge N(x+y) \leq C * N x+C * N y)\)
    \(\wedge\left(\forall c x . x \in F \longrightarrow c *_{R} x \in F \wedge N\left(c *_{R} x\right)=|c| * N x\right)\)
    \(\wedge C \geq 1 \wedge 0 \in F)\)
```

lemma quasinorm-of:
fixes $N::\left({ }^{\prime} a::\right.$ real-vector $) \Rightarrow$ ennreal and $C::$ real
assumes quasinorm-on UNIV C N
shows eNorm (quasinorm-of $(C, N)) x=N x$
defect (quasinorm-of $(C, N))=C$
using assms unfolding eNorm-def defect-def quasinorm-on-def by (auto simp
add: quasinorm-of-inverse)
lemma quasinorm-onI:
fixes $N::($ 'a::real-vector $) \Rightarrow$ ennreal and $C::$ real and $F::{ }^{\prime} a$ set
assumes $\bigwedge x y . x \in F \Longrightarrow y \in F \Longrightarrow x+y \in F$
$\bigwedge x y . x \in F \Longrightarrow y \in F \Longrightarrow N(x+y) \leq C * N x+C * N y$
$\bigwedge c x . c \neq 0 \Longrightarrow x \in F \Longrightarrow c *_{R} x \in F$
$\bigwedge c x . c \neq 0 \Longrightarrow x \in F \Longrightarrow N\left(c *_{R} x\right) \leq$ ennreal $|c| * N x$
$0 \in F N(0)=0 C \geq 1$
shows quasinorm-on $F C N$
proof -
have $N\left(c *_{R} x\right)=$ ennreal $|c| * N x$ if $x \in F$ for $c x$
proof (cases $c=0$ )
case True
then show ?thesis using $\langle N 0=0\rangle$ by auto
next
case False
have $N\left((1 / c) *_{R}\left(c *_{R} x\right)\right) \leq$ ennreal $(a b s(1 / c)) * N\left(c *_{R} x\right)$
apply (rule $\left\langle\bigwedge c x . c \neq 0 \Longrightarrow x \in F \Longrightarrow N\left(c *_{R} x\right) \leq\right.$ ennreal $\left.| c|* N x\rangle\right)$
using False assms that by auto
then have $N x \leq$ ennreal $(a b s(1 / c)) * N\left(c *_{R} x\right)$ using False by auto
then have ennreal $|c| * N x \leq$ ennreal $|c| *$ ennreal (abs $(1 / c)) * N\left(c *_{R} x\right)$
by (simp add: mult.assoc mult-left-mono)
also have $\ldots=N\left(c *_{R} x\right)$ using ennreal-mult' abs-mult False
by (metis abs-ge-zero abs-one comm-monoid-mult-class.mult-1 ennreal-1 eq-divide-eq-1
field-class.field-divide-inverse)
finally show ?thesis
using 〈 $\bigwedge c x . c \neq 0 \Longrightarrow x \in F \Longrightarrow N\left(c *_{R} x\right) \leq$ ennreal $\left.|c| * N x\right\rangle[$ OF False
$\langle x \in F\rangle]$ by auto
qed
then show ?thesis
unfolding quasinorm-on-def using assms by (auto, metis real-vector.scale-zero-left)
qed
lemma extend-quasinorm:

```
    assumes quasinorm-on FCN
    shows quasinorm-on UNIV C ( }\lambdax\mathrm{ . if }x\inF\mathrm{ then }Nx\mathrm{ else }\infty
proof -
    have *: (if }x+y\inF\mathrm{ then N(x+y) else }\infty
```



```
else \infty) for x y
    proof (cases x\inF^y\inF)
        case True
        then show ?thesis using assms unfolding quasinorm-on-def by auto
    next
        case False
        moreover have C\geq1 using assms unfolding quasinorm-on-def by auto
        ultimately have *: ennreal C * (if x f F then N x else \infty) + ennreal C * (if
y}\inF\mathrm{ then N y else }\infty)=
        using ennreal-mult-eq-top-iff by auto
        show ?thesis by (simp add:*)
    qed
    show ?thesis
        apply (rule quasinorm-onI)
    using assms * unfolding quasinorm-on-def apply (auto simp add: ennreal-top-mult
mult.commute)
    by (metis abs-zero ennreal-0 mult-zero-right real-vector.scale-zero-right)
qed
```


### 2.2 The space and the zero space of a quasinorm

The space of a quasinorm is the vector subspace where it is meaningful, i.e., finite.
definition space $_{N}::($ 'a::real-vector) quasinorm $\Rightarrow$ 'a set
where space $N=\{f$. eNorm $N f<\infty\}$
lemma space $N$-iff:
$x \in$ space $_{N} N \longleftrightarrow$ eNorm $N x<\infty$
unfolding space $_{N}-$ def by simp
lemma space $N$-cmult [simp]:
assumes $x \in$ space $_{N} N$
shows $c *_{R} x \in$ space $_{N} N$
using assms unfolding spaceN-iff using eNorm-cmult $[o f ~ N ~ c ~ x] ~ b y ~(s i m p ~ a d d: ~$ ennreal-mult-less-top)
lemma space $N$-add [simp]:
assumes $x \in$ space $_{N} N y \in$ space $_{N} N$
shows $x+y \in$ space $_{N} N$
proof -
have eNorm $N x<\infty$ eNorm $N y<\infty$ using assms unfolding space $_{N}$-def by auto
then have defect $N * e \operatorname{Norm} N x+\operatorname{defect} N * e N o r m ~ N y<\infty$ by (simp add: ennreal-mult-less-top)

```
    then show ?thesis
    unfolding space}\mp@subsup{N}{}{-def}\mathrm{ using eNorm-triangular-ineq[of N x y] le-less-trans by
blast
qed
lemma spaceN-diff [simp]:
    assumes x\in space}\mp@subsup{N}{N}{N}y\in\mp@subsup{\mathrm{ space}}{N}{}
    shows }x-y\in\mp@subsup{\mathrm{ space }}{N}{}
using spaceN-add[OF assms(1) spaceN-cmult[OF assms(2), of -1]] by auto
lemma spaceN-contains-zero [simp]:
    0}\in\mp@subsup{\mathrm{ space }}{N}{}
unfolding space}\mp@subsup{N}{N}{}-def by aut
lemma spaceN-sum [simp]:
    assumes \bigwedgei.i\inI\Longrightarrowxi\in space}\mp@subsup{N}{N}{N
    shows (\sumi\inI.xi)\in space N}
using assms by (induction I rule: infinite-finite-induct, auto)
The zero space of a quasinorm is the vector subspace of vectors with zero norm. If one wants to get a true metric space, one should quotient the space by the zero space.
```

```
definition zero-space \({ }_{N}::\left({ }^{\prime} a::\right.\) real-vector) quasinorm \(\Rightarrow\) 'a set
```

definition zero-space ${ }_{N}::\left({ }^{\prime} a::\right.$ real-vector) quasinorm $\Rightarrow$ 'a set
where zero-space }NN={f. eNorm Nf=0
lemma zero-spaceN-iff:
x\inzero-space }NN\longleftrightarrow eNorm Nx=
unfolding zero-space N
lemma zero-spaceN-cmult:
assumes x f zero-space }\mp@subsup{N}{}{N
shows c**R}x\in\mp@subsup{z}{R}{\prime2-o-space}\mp@subsup{N}{N}{}
using assms unfolding zero-spaceN-iff using eNorm-cmult[of N c x] by simp
lemma zero-spaceN-add:
assumes }x\in\mp@subsup{z\mp@code{ero-space }}{N}{}Ny\in\mp@subsup{z}{\mathrm{ zero-space }}{N}
shows }x+y\in\mp@subsup{zero-space}{N}{}
proof -
have eNorm Nx=0 eNorm Ny=0 using assms unfolding zero-space N-def
by auto
then have defect N*eNorm Nx+\operatorname{defect N*eNorm Ny=0 by auto}
then show ?thesis
unfolding zero-spaceN-iff using eNorm-triangular-ineq[of N x y y by auto
qed
lemma zero-spaceN-diff:
assumes x fzero-space }NN Ny\in\mp@subsup{zero-space }{N}{}
shows }x-y\in\mp@subsup{z}{}{\prime
using zero-spaceN-add[OF assms(1) zero-spaceN-cmult[OF assms(2), of -1]] by

```
auto
lemma zero-space \(N\)-subset-space \(N\) :
\(z^{\text {zero-space }}{ }_{N} N \subseteq\) space \(_{N} N\)
by (simp add: spaceN-iff zero-space \(N\)-iff subset-eq)
On the space, the norms are finite. Hence, it is much more convenient to work there with a real valued version of the norm. We use Norm with a capital N to distinguish it from norms in a (type class) banach space.
```

definition Norm::'a quasinorm }=>\mathrm{ ('a::real-vector) }=>\mathrm{ real
where Norm Nx = enn2real (eNorm Nx)
lemma Norm-nonneg [simp]:
Norm Nx\geq0
unfolding Norm-def by auto
lemma Norm-zero [simp]:
Norm N0 =0
unfolding Norm-def by auto
lemma Norm-uminus [simp]:
Norm N (-x) = Norm Nx
unfolding Norm-def by auto
lemma eNorm-Norm:
assumes x\in space}\mp@subsup{N}{N}{}
shows eNorm Nx = ennreal (Norm N x)
using assms unfolding Norm-def by (simp add: spaceN-iff)
lemma eNorm-Norm':
assumes }x\not\in\mp@subsup{\mathrm{ space N}}{N}{}
shows Norm N x = 0
using assms unfolding Norm-def apply (auto simp add: spaceN-iff)
using top.not-eq-extremum by fastforce
lemma Norm-cmult:
Norm N (c** x ) =abs c*Norm Nx
unfolding Norm-def unfolding eNorm-cmult by (simp add: enn2real-mult)
lemma Norm-triangular-ineq:
assumes }x\in\mp@subsup{\mathrm{ space }}{N}{}
shows Norm N (x+y)\leq\operatorname{defect N*Norm N x + defect N*Norm N y}
proof (cases y }\in\mp@subsup{\mathrm{ space N}}{N}{}N
case True
have *: defect N*Norm Nx+defect N*Norm Ny\geq1*0+1*0
apply (rule add-mono) by (rule mult-mono'[OF defect-ge-1 Norm-nonneg],
simp, simp)+
have ennreal (Norm N (x+y)) = eNorm N (x+y)
using eNorm-Norm[OF spaceN-add[OF assms True]] by auto

```
```

    also have \(\ldots \leq \operatorname{defect} N * \operatorname{eNorm} N x+\operatorname{defect} N * e N o r m ~ N y\)
        using eNorm-triangular-ineq[of \(N x y]\) by auto
    also have \(\ldots=\operatorname{defect} N * \operatorname{ennreal}(\operatorname{Norm} N x)+\operatorname{defect} N * \operatorname{ennreal}(\operatorname{Norm} N y)\)
    using eNorm-Norm assms True by metis
    also have \(\ldots=\operatorname{ennreal}(\operatorname{defect} N * \operatorname{Norm} N x+\operatorname{defect} N * \operatorname{Norm} N y)\)
    using ennreal-mult ennreal-plus Norm-nonneg defect-ge-1
        by (metis (no-types, opaque-lifting) ennreal-eq-0-iff less-le ennreal-ge-1 en-
    nreal-mult' le-less-linear not-one-le-zero semiring-normalization-rules(34))
finally show ?thesis
apply (subst ennreal-le-iff [symmetric]) using * by auto
next
case False
have $x+y \notin$ space $_{N} N$
proof (rule ccontr)
assume $\neg\left(x+y \notin\right.$ space $\left._{N} N\right)$
then have $x+y \in$ space $_{N} N$ by simp
have $y \in$ space $_{N} N$ using spaceN-diff $\left[O F\left\langle x+y \in\right.\right.$ space $\left._{N} N\right\rangle$ assms $]$ by auto
then show False using False by simp
qed
then have Norm $N(x+y)=0$ unfolding Norm-def using spaceN-iff top.not-eq-extremum
by force
moreover have defect $N * \operatorname{Norm} N x+\operatorname{defect} N * \operatorname{Norm} N y \geq 1 * 0+1 * 0$
apply (rule add-mono) by (rule mult-mono'[OF defect-ge-1 Norm-nonneg],
simp, simp)+
ultimately show ?thesis by simp
qed
lemma Norm-triangular-ineq-diff:
assumes $x \in$ space $_{N} N$
shows $\operatorname{Norm} N(x-y) \leq \operatorname{defect} N * \operatorname{Norm} N x+\operatorname{defect} N * N o r m ~ N y$
using Norm-triangular-ineq[OF assms, of $-y]$ by auto
lemma zero-space $N$-iff ':
$x \in$ zero-space $_{N} N \longleftrightarrow\left(x \in\right.$ space $_{N} N \wedge$ Norm $\left.N x=0\right)$
using eNorm-Norm unfolding space $_{N}$-def zero-space ${ }_{N}$-def by (auto simp add:
Norm-def, fastforce)
lemma Norm-sum:
assumes $\bigwedge i . i<n \Longrightarrow u i \in$ space $_{N} N$
shows Norm $N\left(\sum i \in\{. .<n\} . u i\right) \leq\left(\sum i \in\{. .<n\} .(\right.$ defect $N)$ 个(Suc $\left.i\right) *$ Norm
$N(u i))$
proof -
have *: $0 \leq \operatorname{defect} N * \operatorname{defect} N^{\wedge} i * \operatorname{Norm} N(u i)$ for $i$
by (meson Norm-nonneg defect-ge-1 dual-order.trans linear mult-nonneg-nonneg
not-one-le-zero zero-le-power)
have ennreal $\left(\operatorname{Norm} N\left(\sum i \in\{. .<n\} . u i\right)\right)=e \operatorname{Norm} N\left(\sum i \in\{. .<n\} . u i\right)$
apply (rule eNorm-Norm[symmetric], rule spaceN-sum) using assms by auto
also have $\ldots \leq\left(\sum i \in\{. .<n\} .(\right.$ defect $N) \uparrow($ Suc $i) *$ eNorm $\left.N(u i)\right)$

```
using eNorm-sum by simp
also have \(\ldots=\left(\sum i \in\{. .<n\} .(\right.\) defect \(N) \uparrow(\) Suc \(i) *\) ennreal \(\left.(\operatorname{Norm} N(u i))\right)\) using eNorm-Norm [OF assms] by auto
also have \(\ldots=\left(\sum i \in\{. .<n\}\right.\). ennreal \(\left.((\operatorname{defect} N) \uparrow(S u c i) * \operatorname{Norm} N(u i))\right)\) by (subst ennreal-mult \({ }^{\prime \prime}\), auto)
also have \(\ldots=\) ennreal \(\left(\sum i \in\{. .<n\} .(\right.\) defect \(N) \uparrow(\) Suc \(\left.i) * \operatorname{Norm} N(u i)\right)\)
by (auto intro!: sum-ennreal simp add: *)
finally have \(* *\) : ennreal \(\left(\operatorname{Norm} N\left(\sum i \in\{. .<n\} . u i\right)\right) \leq \operatorname{ennreal}\left(\sum i \in\{. .<n\}\right.\).
\((\operatorname{defect} N) \uparrow(S u c i) * \operatorname{Norm} N(u i))\) by \(\operatorname{simp}\)
show ?thesis
apply (subst ennreal-le-iff[symmetric], rule sum-nonneg) using \(* * *\) by auto qed

\subsection*{2.3 An example: the ambient norm in a normed vector space}
```

definition $N$-of-norm::'a::real-normed-vector quasinorm
where $N$-of-norm $=$ quasinorm-of $(1, \lambda f$. norm $f)$
lemma $N$-of-norm:
eNorm $N$-of-norm $f=\operatorname{ennreal}($ norm $f)$
Norm $N$-of-norm $f=$ norm $f$
$\operatorname{defect}(N$-of-norm $)=1$
proof -
have *: quasinorm-on UNIV 1 ( $\lambda f$. norm $f$ )
by (rule quasinorm-onI, auto simp add: ennreal-mult', metis ennreal-leI en-
nreal-plus norm-imp-pos-and-ge norm-triangle-ineq)
show eNorm $N$-of-norm $f=$ ennreal (norm $f$ )
$\operatorname{defect}(N$-of-norm $)=1$
unfolding $N$-of-norm-def using quasinorm-of $[O F *]$ by auto
then show Norm $N$-of-norm $f=$ norm $f$ unfolding Norm-def by auto
qed
lemma $N$-of-norm-space $[$ simp $]$ :
space $_{N} N$-of-norm $=U N I V$
unfolding space $_{N}$-def apply auto unfolding $N$-of-norm(1) by auto
lemma $N$-of-norm-zero-space [simp]:
zero-space $_{N} N$-of-norm $=\{0\}$
unfolding zero-space $_{N}$-def apply auto unfolding $N$-of-norm(1) by auto

```

\subsection*{2.4 An example: the space of bounded continuous functions from a topological space to a normed real vector space}

The Banach space of bounded continuous functions is defined in Bounded_Continuous_Function.thy as a type bcontfun. We import very quickly the results proved in this file to the current framework.
definition bcontfun \({ }_{N}::\left({ }^{( } a::\right.\) topological-space \(\Rightarrow\) ' \(b::\) real-normed-vector) quasinorm
where bcontfun \(_{N}=\) quasinorm-of ( \(1, \lambda f\). if \(f \in\) bcontfun then norm(Bcontfun f) else ( \(\infty\) ::ennreal \()\) )

\section*{lemma bcontfun \({ }_{N}\) :}
fixes \(f::(\) 'a::topological-space \(\Rightarrow\) 'b::real-normed-vector)
shows eNorm bcontfun \({ }_{N} f=(\) if \(f \in\) bcontfun then norm(Bcontfun \(f\) ) else ( \(\infty:\) :ennreal))

Norm bcontfun \(_{N} f=(\) if \(f \in\) bcontfun then norm \((\) Bcontfun \(f)\) else 0\()\)
defect \(\left(b \operatorname{contfun} N_{N}::\left(\left({ }^{\prime} a \Rightarrow\right.\right.\right.\) 'b) quasinorm \(\left.)\right)=1\)
proof -
have \(*\) : quasinorm-on bcontfun \(1\left(\lambda\left(f::\left({ }^{\prime} a \Rightarrow{ }^{\prime} b\right)\right)\right.\). norm \(\left.(B \operatorname{contfun} f)\right)\)
proof (rule quasinorm-onI, auto)
fix \(f g::^{\prime} a \Rightarrow\) ' \(b\) assume \(H: f \in b\) contfun \(g \in b\) contfun
then show \(f+g \in\) bcontfun unfolding plus-fun-def by (simp add: plus-cont)
have \(*\) : Bcontfun \((f+g)=B\) contfun \(f+B\) contfun \(g\) using \(H\)
by (auto simp: eq-onp-def plus-fun-def bcontfun-def intro!: plus-bcontfun.abs-eq[symmetric])
show ennreal (norm \((\) Bcontfun \((f+g))) \leq \operatorname{ennreal}(\) norm \((\) Bcontfun \(f))+\) ennreal (norm (Bcontfun g))
unfolding * using ennreal-leI[OF norm-triangle-ineq] by auto
next
fix \(c::\) real and \(f::^{\prime} a \Rightarrow\) ' \(b\) assume \(H: f \in b c o n t f u n\)
then show \(c *_{R} f \in\) bcontfun unfolding scaleR-fun-def by (simp add: scaleR-cont)
have \(*\) : \(\operatorname{Bcontfun}\left(c *_{R} f\right)=c *_{R} B\) contfun \(f\) using \(H\)
by (auto simp: eq-onp-def scaleR-fun-def bcontfun-def intro!: scaleR-bcontfun.abs-eq[symmetric])
show ennreal (norm (Bcontfun \(\left.\left(c *_{R} f\right)\right)\) ) \(\leq\) ennreal \(|c| *\) ennreal (norm
(Bcontfun f))
unfolding * by (simp add: ennreal-mult')
next
show \(\left(0::^{\prime} a \Rightarrow ' b\right) \in b c o n t f u n\) Bcontfun \(0=0\)
unfolding zero-fun-def zero-bcontfun-def by (auto simp add: const-bcontfun)
qed
have \(* *\) : quasinorm-on UNIV \(1\left(\lambda\left(f::^{\prime} a \Rightarrow^{\prime} b\right)\right.\). if \(f \in\) bcontfun then norm(Bcontfun
f) else ( \(\infty::\) ennreal \()\) )
by (rule extend-quasinorm[OF *])
show eNorm bcontfun \({ }_{N} f=(\) if \(f \in\) bcontfun then norm (Bcontfun \(f)\) else ( \(\infty:\) :ennreal \()\) )
defect (bcontfun \({ }_{N}::\left({ }^{\prime} a \Rightarrow\right.\) ' \(b\) ) quasinorm \()=1\)
using quasinorm-of \([O F * *]\) unfolding bcontfun \(N_{N}\)-def by auto
then show Norm bcontfun \(N_{N}=(\) if \(f \in\) bcontfun then norm(Bcontfun f) else 0\()\)
unfolding Norm-def by auto
qed
lemma bcontfun \({ }_{N}\)-space:
space \(_{N}\) bcontfun \(_{N}=\) bcontfun
using bcontfun \({ }_{N}(1)\) by (metis (no-types, lifting) Collect-cong bcontfun-def enn2real-top ennreal-0
ennreal-enn2real ennreal-less-top ennreal-zero-neq-top infinity-ennreal-def mem-Collect-eq
```

space N-def)

```
lemma bcontfun \(N_{N}\)-zero-space:
zero-space \(_{N}\) bcontfun \(_{N}=\{0\}\)
apply (auto simp add: zero-spaceN-iff)
by (metis Bcontfun-inject bcontfun \({ }_{N}(1)\) eNorm-zero ennreal-eq-zero-iff ennreal-zero-neq-top infinity-ennreal-def norm-eq-zero norm-imp-pos-and-ge)
lemma bcontfun \(_{N} D\) :
assumes \(f \in\) space \(_{N}\) bcontfun \(_{N}\)
shows continuous-on UNIV \(f\)
\(\bigwedge x . \operatorname{norm}(f x) \leq \operatorname{Norm}^{\operatorname{bcontfun}}{ }_{N} f\)
proof-
have \(f \in\) bcontfun using assms unfolding bcontfun \(_{N}\)-space by simp
then show continuous-on UNIV \(f\) unfolding bcontfun-def by auto
show \(\bigwedge x\). \(\operatorname{norm}(f x) \leq \operatorname{Norm}^{\text {bcontfun }}{ }_{N} f\)
using norm-bounded bcontfun \({ }_{N}\) (2) \(\langle f \in\) bcontfun» by (metis Bcontfun-inverse)
qed
lemma bcontfun \({ }_{N} I\) :
assumes continuous-on UNIV \(f\)
\[
\backslash x . \operatorname{norm}(f x) \leq C
\]
shows \(f \in\) space \(_{N}\) bcontfun \(_{N}\)
Norm bcontfun \(_{N} f \leq C\)
proof -
have \(f \in\) bcontfun using assms bcontfun-normI by blast
then show \(f \in\) space \(_{N}\) bcontfun \(_{N}\) unfolding bcontfun \({ }_{N}\)-space by simp
show Norm bcontfun \({ }_{N} f \leq C\) unfolding bcontfun \(_{N}(2)\) using \(\langle f \in\) bcontfun»
apply auto
using assms(2) by (metis apply-bcontfun-cases apply-bcontfun-inverse norm-bound)
qed

\subsection*{2.5 Continuous inclusions between functional spaces}

Continuous inclusions between functional spaces are now defined
```

instantiation quasinorm:: (real-vector) preorder
begin
definition less-eq-quasinorm::'a quasinorm }=>\mathrm{ 'a quasinorm }=>\mathrm{ bool
where less-eq-quasinorm N1 N2 = (\existsC\geq(0::real).}\forallf.eNorm N2 f\leqC*eNorm
N1 f)

```
definition less-quasinorm::'a quasinorm \(\Rightarrow\) 'a quasinorm \(\Rightarrow\) bool
    where less-quasinorm N1 N2 \(=(\) less-eq N1 N2 \(\wedge(\neg\) less-eq N2 N1 \()\) )
instance proof -
    have \(E: N \leq N\) for \(N::^{\prime}{ }^{\prime}\) quasinorm
        unfolding less-eq-quasinorm-def by (rule exI[of - 1], auto)
    have \(T: N 1 \leq N 3\) if \(N 1 \leq N 2\) N2 \(\leq N 3\) for N1 N2 N3::'a quasinorm
```

    proof -
    obtain C C'' where *: \f. eNorm N2 f}\leq\mathrm{ ennreal C * eNorm N1 f
                    \f.eNorm N3 f}\leq\mathrm{ ennreal C'* eNorm N2 f
                    C\geq0 C'\geq0
        using <N1 \leqN2〉<N2 \leqN3> unfolding less-eq-quasinorm-def by metis
    {
        fix f
        have eNorm N3 f}\leq\mathrm{ ennreal C'* ennreal C * eNorm N1 f
        by (metis*(1)[off]*(2)[of f] mult.commute mult.left-commute mult-left-mono
    order-trans zero-le)
also have ... = ennreal ( }\mp@subsup{C}{}{\prime}*C)*eNorm N1
using \langleC \geq0\rangle\langle\mp@subsup{C}{}{\prime}\geq0\rangle ennreal-mult by auto
finally have eNorm N3 f}\leq\operatorname{ennreal( (C'*C)*eNorm N1 f by simp
}
then show ?thesis
unfolding less-eq-quasinorm-def using <C \geq0\rangle\langleC''\geq0\rangle zero-le-mult-iff by
auto
qed
show OFCLASS('a quasinorm, preorder-class)
apply standard
unfolding less-quasinorm-def apply simp
using E apply fast
using T apply fast
done
qed
end
abbreviation quasinorm-subset :: ('a::real-vector) quasinorm = 'a quasinorm }
bool
where quasinorm-subset }\equiv\mathrm{ less
abbreviation quasinorm-subset-eq :: ('a::real-vector) quasinorm = 'a quasinorm
=> bool
where quasinorm-subset-eq \equiv less-eq
notation
quasinorm-subset ('(}\mp@subsup{\subset}{N}{}))\mathrm{ ) and
quasinorm-subset ((-/ \subset N -) [51, 51] 50) and
quasinorm-subset-eq ('( (\subseteqN')) and
quasinorm-subset-eq ((-/ \subseteq}\mp@subsup{\}{N}{}-)[51, 51] 50
lemma quasinorm-subsetD:
assumes N1 \subseteqN N2
shows \existsC\geq(0::real).}\forallf.eNorm N2 f\leqC*eNorm N1 f
using assms unfolding less-eq-quasinorm-def by auto
lemma quasinorm-subsetI:

```
```

    assumes \f.f\in space N N1 \Longrightarrow eNorm N2 f}\leq\mathrm{ ennreal C * eNorm N1f
    shows N1 \subseteq}\mp@subsup{\}{N}{N2
    proof -
have eNorm N2 f \leqennreal (max C 1)*eNorm N1 f for f
proof (cases f\in space}\mp@subsup{N}{N}{N}1
case True
then show ?thesis using assms[OF<f\in\mp@subsup{\mathrm{ space }}{N}{}N1\rangle]
by (metis (no-types, opaque-lifting) dual-order.trans ennreal-leI max.cobounded2
max.commute
mult.commute ordered-comm-semiring-class.comm-mult-left-mono zero-le)
next
case False
then show ?thesis using spaceN-iff
by (metis ennreal-ge-1 ennreal-mult-less-top infinity-ennreal-def max.cobounded1
max.commute not-le not-one-le-zero top.not-eq-extremum)
qed
then show ?thesis unfolding less-eq-quasinorm-def
by (metis ennreal-max-0' max.cobounded2)
qed
lemma quasinorm-subsetI':
assumes \f.f\in\mp@subsup{\operatorname{space}}{N}{}N1\Longrightarrowf\in\mp@subsup{\mathrm{ space N}}{N}{N2}
\f.f\in\mp@subsup{\operatorname{space}}{N}{}N1\LongrightarrowNorm N2 f}\leqC\mathrm{ C Norm N1 f
shows N1 \subseteq}\mp@subsup{\}{N}{N2
proof (rule quasinorm-subsetI)
fix f}\mathrm{ assume f}\in\mp@subsup{\operatorname{space}}{N}{}N
then have f}\in\mp@subsup{\mathrm{ space N}}{N}{}\mathrm{ N2 using assms(1) by simp
then have eNorm N2 f = ennreal(Norm N2 f) using eNorm-Norm by auto
also have .. \leq ennreal(C*Norm N1 f)
using assms(2)[OF<f\in\mp@subsup{\operatorname{space}}{N}{}N1\rangle] ennreal-leI by blast
also have ... = ennreal C*ennreal(Norm N1f)
using ennreal-mult/" by auto
also have ... = ennreal C * eNorm N1f
using eNorm-Norm[OF<f \in space}\mp@subsup{N}{N}{N1\rangle] by auto
finally show eNorm N2 f}\leq\mathrm{ ennreal C * eNorm N1 f
by simp
qed
lemma quasinorm-subset-space:
assumes N1 \subseteq}\mp@subsup{\}{N}{}N
shows space}\mp@subsup{N}{N1}{N}\subseteq\mp@subsup{\mathrm{ space N}}{N}{N2
using assms unfolding space }\mp@subsup{N}{N}{}\mathrm{ -def less-eq-quasinorm-def
by (auto, metis ennreal-mult-eq-top-iff ennreal-neq-top less-le top.extremum-strict
top.not-eq-extremum)
lemma quasinorm-subset-Norm-eNorm:
assumes f}\in\mp@subsup{\mathrm{ space N}}{N}{}N1\LongrightarrowNorm N2 f\leqC*Norm N1 f
N1 \subseteqN N2
C>0

```
```

    shows eNorm N2 f}\leq\mathrm{ ennreal C * eNorm N1 f
    proof (cases f\in\mp@subsup{\mathrm{ space }}{N}{}N1)
case True
then have f}\in\mp@subsup{\mathrm{ space N}}{N}{}N2\mathrm{ using quasinorm-subset-space[OF 〈N1 }\mp@subsup{\subseteq}{N}{}N2\] by

auto
then show ?thesis
using eNorm-Norm[OF True] eNorm-Norm assms(1)[OF True] by (metis
Norm-nonneg ennreal-leI ennreal-mult'')
next
case False
then show ?thesis using <C> 0\rangle
by (metis ennreal-eq-zero-iff ennreal-mult-eq-top-iff infinity-ennreal-def less-imp-le
neq-top-trans not-le spaceN-iff)
qed
lemma quasinorm-subset-zero-space:
assumes N1 \subseteq}\mp@subsup{\}{N}{}N
shows zero-space}\mp@subsup{N}{N}{N1}\subseteq\mp@subsup{z_ro-space}{N}{}\mathrm{ N2
using assms unfolding zero-space }\mp@subsup{N}{N}{}\mathrm{ -def less-eq-quasinorm-def
by (auto, metis le-zero-eq mult-zero-right)
We would like to define the equivalence relation associated to the above order, i.e., the equivalence between norms. This is not equality, so we do not have a true order, but nevertheless this is handy, and not standard in a preorder in Isabelle. The file Library/Preorder.thy defines such an equivalence relation, but including it breaks some proofs so we go the naive way.
definition quasinorm-equivalent::('a::real-vector) quasinorm $\Rightarrow{ }^{\prime}$ 'a quasinorm $\Rightarrow$ bool (infix $={ }_{N} 60$ )
where quasinorm-equivalent N1 N2 $=\left(\left(N 1 \subseteq_{N} N 2\right) \wedge\left(N 2 \subseteq_{N} N 1\right)\right)$
lemma quasinorm-equivalent-sym [sym]:
assumes $N 1={ }_{N} N 2$
shows $N 2={ }_{N} N 1$
using assms unfolding quasinorm-equivalent-def by auto
lemma quasinorm-equivalent-trans [trans]:
assumes $N 1=_{N} N 2$ N2 $=_{N} N 3$
shows $N 1={ }_{N} N 3$
using assms order-trans unfolding quasinorm-equivalent-def by blast

```

\subsection*{2.6 The intersection and the sum of two functional spaces}

In this paragraph, we define the intersection and the sum of two functional spaces. In terms of the order introduced above, this corresponds to the minimum and the maximum. More important, these are the first two examples of interpolation spaces between two functional spaces, and they are central as all the other ones are built using them.
definition quasinorm-intersection::('a::real-vector) quasinorm \(\Rightarrow\) 'a quasinorm \(\Rightarrow\) 'a quasinorm (infix \(\cap_{N} 70\) )
where quasinorm-intersection N1 N2 \(=\) quasinorm-of \((\max (\) defect \(N 1)\) (defect N2), \(\lambda f\). eNorm N1 \(f+e \operatorname{Norm}\) N2 \(f\) )
lemma quasinorm-intersection:
\(e \operatorname{Norm}\left(N 1 \cap_{N}\right.\) N2) \(f=e \operatorname{Norm} N 1 f+e N o r m\) N2 \(f\)
\(\operatorname{defect}\left(N 1 \cap_{N} N 2\right)=\max (\operatorname{defect} N 1)(\operatorname{defect} N 2)\)
proof -
have T: eNorm N1 \((x+y)+e N o r m ~ N 2 ~(x+y) \leq\)
ennreal (max (defect N1) (defect N2)) * (eNorm N1 \(x+e N o r m ~ N 2 ~ x)+\) ennreal \((\max (\) defect N1) \((\) defect N2) \() *(e N o r m ~ N 1 ~ y+e N o r m ~ N 2 ~ y) ~ f o r ~ x ~ y ~\)
proof -
have eNorm N1 \((x+y) \leq \operatorname{ennreal}(\max (\) defect \(N 1)(\) defect N2)) \(* e N o r m\) N1 \(x+\) ennreal \((\max (\) defect N1) (defect N2)) \(*\) eNorm N1 y
using eNorm-triangular-ineq[of N1 \(x y]\) by (metis (no-types) max-def dis-trib-left ennreal-leI mult-right-mono order-trans zero-le)
moreover have eNorm N2 \((x+y) \leq\) ennreal ( \(\max\) (defect N1) (defect N2))
* eNorm N2 \(x+\) ennreal \((\max (\) defect N1) (defect N2)) \(*\) eNorm N2 \(y\) using eNorm-triangular-ineq[of N2 \(x y\) ] by (metis (no-types) max-def max.commute distrib-left ennreal-leI mult-right-mono order-trans zero-le) ultimately have eNorm N1 \((x+y)+\) eNorm N2 \((x+y) \leq\) ennreal (max (defect N1) (defect N2)) * (eNorm N1 \(x+e N o r m ~ N 1 ~ y ~(e N o r m ~ N 2 ~ x ~ e N o r m ~\) N2 y) by (simp add: add-mono-thms-linordered-semiring(1) distrib-left) then show ?thesis by (simp add: ab-semigroup-add-class.add-ac(1) add.left-commute distrib-left) qed
have \(H\) : eNorm N1 \(\left(c *_{R} x\right)+e\) Norm N2 \(\left(c *_{R} x\right) \leq\) ennreal \(|c| *(e N o r m\) N1 \(x+e N o r m\) N2 \(x)\) for \(c x\) by (simp add: eNorm-cmult[of N1 c x] eNorm-cmult[of N2 c x] distrib-left)
have *: quasinorm-on UNIV (max (defect N1) (defect N2)) ( \(\lambda\) f. eNorm N1f+ eNorm N2 f)
apply (rule quasinorm-onI) using \(T H\) defect-ge-1[of N1] defect-ge-1[of N2] by auto
show defect \(\left(N 1 \cap_{N} N 2\right)=\max (\) defect N1) (defect N2)
eNorm ( \(N 1 \cap_{N}\) N2) \(f=e \operatorname{Norm} N 1 f+e N o r m\) N2 \(f\)
unfolding quasinorm-intersection-def using quasinorm-of[OF *] by auto qed
lemma quasinorm-intersection-commute:
\(N 1 \cap_{N} N 2=N 2 \cap_{N} N 1\)
unfolding quasinorm-intersection-def max.commute[of defect N1] add.commute[of eNorm N1 -] by simp
lemma quasinorm-intersection-space:
\(\operatorname{space}_{N}\left(N 1 \cap_{N} N 2\right)=\operatorname{space}_{N} N 1 \cap\) space \(_{N}\) N2
apply auto unfolding quasinorm-intersection(1) spaceN-iff by auto
lemma quasinorm-intersection-zero-space:
zero-space \(_{N}\left(N 1 \cap_{N}\right.\) N2 \()=\) zero-space \(_{N} N 1 \cap\) zero-space \(_{N}\) N2
apply auto unfolding quasinorm-intersection(1) zero-spaceN-iff by (auto simp add: add-eq-0-iff-both-eq-0)
lemma quasinorm-intersection-subset:
\(N 1 \cap_{N} N 2 \subseteq_{N} N 1 N 1 \cap_{N} N 2 \subseteq_{N} N 2\)
by (rule quasinorm-subsetI[of-1], auto simp add: quasinorm-intersection(1))+
lemma quasinorm-intersection-minimum:
assumes \(N \subseteq_{N} N 1 N \subseteq_{N} N 2\)
shows \(N \subseteq_{N} N 1 \cap_{N} N 2\)
proof -
obtain C1 C2:: real where \(*: \bigwedge f\). eNorm \(N 1 f \leq C 1 * e N o r m ~ N f\)
\f. eNorm N2 \(f \leq C 2 *\) eNorm \(N f\)
\(C 1 \geq 0 C 2 \geq 0\)
using quasinorm-subsetD[OF assms(1)] quasinorm-subsetD[OF assms(2)] by blast
have \(* *: \operatorname{eNorm}\left(N 1 \cap_{N} N 2\right) f \leq(C 1+C 2) * \operatorname{eNorm} N f\) for \(f\)
unfolding quasinorm-intersection(1) using add-mono[OF *(1) *(2)] by (simp add: distrib-right *)
show ?thesis
apply (rule quasinorm-subsetI) using \(* *\) by auto
qed
lemma quasinorm-intersection-assoc:
\(\left(N 1 \cap_{N} N 2\right) \cap_{N} N 3=_{N} N 1 \cap_{N}\left(N 2 \cap_{N} N 3\right)\)
unfolding quasinorm-equivalent-def by (meson order-trans quasinorm-intersection-minimum quasinorm-intersection-subset)
definition quasinorm-sum::('a::real-vector) quasinorm \(\Rightarrow\) 'a quasinorm \(\Rightarrow{ }^{\prime}\) 'a quasinorm (infix \(+_{N} 70\) )
where quasinorm-sum N1 N2 \(=\) quasinorm-of \((\max\) (defect N1) (defect N2), \(\lambda f\). Inf \(\{\) eNorm N1 f1 \(+e\) Norm N2 f2| \(f 1\) f2. \(f=f 1+f 2\})\)

\section*{lemma quasinorm-sum:}
\(e \operatorname{Norm}\left(N 1+_{N} N 2\right) f=\operatorname{Inf}\{e N o r m N 1 f 1+e N o r m\) N2 f2| f1 f2. \(f=f 1+\) f2\}
\(\operatorname{defect}\left(N 1+_{N} N 2\right)=\max (\operatorname{defect} N 1)(\operatorname{defect} N 2)\)
proof -
define \(N\) where \(N=(\lambda f\). Inf \(\{e N o r m\) N1 f1 \(+e \operatorname{Norm} N 2\) f2| f1 f2. \(f=f 1+\) f2 \})
have \(T: N(f+g) \leq\)
ennreal \((\max (\) defect \(N 1)(\) defect \(N 2)) * N f+\) ennreal \((\max (\) defect \(N 1)\) (defect N2)) * Ng for \(f g\)
proof -
have \(\exists u .(\forall n . u n \in\{e N o r m\) N1 f1 \(+e \operatorname{Norm}\) N2 f2| \(f 1 f 2 . f=f 1+f 2\}) \wedge\) \(u \longrightarrow \operatorname{Inf}\{e N o r m\) N1 f1 + eNorm N2 f2| f1 f2. \(f=f 1+f 2\}\)
by (rule Inf-as-limit, auto, rule exI \([o f-f]\), rule exI \([o f-0]\), auto)
then obtain \(u f\) where \(u f: \bigwedge n\). uf \(n \in\{e N o r m\) N1 f1 \(+e N o r m\) N2 f2|f1 f2. \(f=f 1+f 2\}\)
\[
u f \longrightarrow \operatorname{Inf}\{e N o r m \text { N1 f1 }+e \text { Norm N2 f2| f1 f2. } f=
\]
\(f 1+f 2\}\)
by blast
have \(\exists f 1\) f2. \(\forall n\). uf \(n=e \operatorname{Norm} N 1(f 1 n)+e \operatorname{Norm} N 2(f 2 n) \wedge f=f 1 n+\) f2 \(n\)
apply (rule SMT.choices(1)) using uf(1) by blast
then obtain \(f 1\) f2 where \(F: \bigwedge n\). uf \(n=e \operatorname{Norm} N 1\) ( \(f 1 n\) n \()+e \operatorname{Norm}\) N2 (f2 n) \(\bigwedge n . f=f 1 n+f 2 n\) by blast
have \(\exists u .(\forall n . u n \in\{e N o r m\) N1 \(g 1+e N o r m\) N2 \(g 2 \mid g 1 g 2 . g=g 1+g 2\})\) \(\wedge u \longrightarrow \operatorname{Inf}\{e N o r m\) N1 \(g 1+e N o r m\) N2 \(g 2 \mid g 1 g 2 . g=g 1+g 2\}\)
by (rule Inf-as-limit, auto, rule exI \([o f-g]\), rule exI \([o f-0]\), auto)
then obtain \(u g\) where \(u g: \bigwedge n . u g n \in\{e N o r m N 1 g 1+e N o r m\) N2 \(g 2 \mid g 1\) \(g 2 . g=g 1+g 2\}\)
\(u g \longrightarrow \operatorname{Inf}\{e N o r m\) N1 \(g 1+e N o r m\) N2 \(g 2 \mid g 1 g 2 \cdot g=\) \(g 1+g 2\}\)
by blast
have \(\exists g 1\) g2. \(\forall n . u g n=e N o r m\) N1 \((g 1 n)+e \operatorname{Norm} N 2(g 2 n) \wedge g=g 1 n\) \(+g 2 n\)
apply (rule SMT.choices(1)) using \(u g(1)\) by blast
then obtain g1 g2 where G: \(\bigwedge n\). ug \(n=e \operatorname{Norm} N 1(g 1 n)+e N o r m N 2(g 2\) n) \(\bigwedge n . g=g 1 n+g 2 n\)
by blast
define \(h 1\) where \(h 1=(\lambda n . f 1 n+g 1 n)\)
define \(h 2\) where \(h 2=(\lambda n\). f2 \(n+g 2 n)\)
have \(*: f+g=h 1 n+h 2 n\) for \(n\)
unfolding h1-def h2-def using \(F\) (2) \(G\) (2) by (auto simp add: algebra-simps)
have \(N(f+g) \leq\) ennreal \((\max (\) defect N1) \((\) defect N2) \() *(u f n+u g n)\) for \(n\)
proof -
have \(N(f+g) \leq e N o r m\) N1 (h1 n) \(+e \operatorname{Norm}\) N2 (h2 n)
unfolding \(N\)-def apply (rule Inf-lower, auto, rule exI[of - h1 n], rule exI[of - \(h 2\) n])
using * by auto
also have \(\ldots \leq\) ennreal \((\) defect \(N 1) *\) eNorm N1 \((f 1 n)+\) ennreal (defect N1) * eNorm N1 (g1 n)
\(+(\) ennreal \((\) defect N2) \(* \operatorname{eNorm}\) N2 (f2 \(n)+\) ennreal (defect N2)
* eNorm N2 (g2 n))
unfolding h1-def h2-def apply (rule add-mono) using eNorm-triangular-ineq by auto
also have \(\ldots \leq(\) ennreal \((\max (\operatorname{defect} N 1)(\) defect N2) \() * e N o r m ~ N 1(f 1 n)\) \(+\operatorname{ennreal}(\max (\) defect N1) (defect N2)) * eNorm N1 (g1 n))
\[
+(\text { ennreal }(\max (\text { defect N1 })(\text { defect N2 })) * \text { eNorm N2 }(\text { f2 } n)+
\]
```

ennreal (max (defect N1) (defect N2)) * eNorm N2 (g2 n))
by (auto intro!: add-mono mult-mono ennreal-leI)
also have ... = ennreal (max (defect N1) (defect N2)) * (uf n +ug n)
unfolding F(1)G(1) by (auto simp add: algebra-simps)
finally show ?thesis by simp
qed
moreover have ...\longrightarrow ennreal (max (defect N1) (defect NQ)) * (Nf +N
g)
unfolding N-def by (auto intro!: tendsto-intros simp add: uf(2) ug(2))
ultimately have N (f+g) \leq ennreal (max (defect N1) (defect N2)) * (Nf +
Ng)
using LIMSEQ-le-const by blast
then show ?thesis by (auto simp add: algebra-simps)
qed
have H:N(c**R}f)\leqennreal |c|*Nf for cf
proof -
have \existsu.(\foralln.u n \in{eNorm N1 f1 + eNorm N2 f2| f1 f2. f =f1 + f2}) ^
u\longrightarrowInf {eNorm N1 f1 + eNorm N2 f2| f1 f2. f = f1 + f2}
by (rule Inf-as-limit, auto, rule exI[of-f], rule exI[of-0], auto)
then obtain uf where uf: \n. uf n \in{eNorm N1 f1 + eNorm N2 f2| f1 f2.
f=f1 + f2 }
uf\longrightarrowInf {eNorm N1 f1 + eNorm N2 f2| f1 f2. f=
f1 + f2}
by blast
have \existsf1 f2. \foralln.uf n =eNorm N1 (f1 n) +eNorm N2 (f2 n) ^f =f1 n +
f2 n
apply (rule SMT.choices(1)) using uf(1) by blast
then obtain f1 f2 where F: \n. uf n =eNorm N1 (f1 n) +eNorm N2 (f2
n) }\n.f=f1n+f2
by blast
have N(c**R}f)\leq|c|*ufn\mathrm{ for n
proof -
have N(c**R}f)\leqeNorm N1 (c** f1 n) +eNorm N2 (c**R f2 n)
unfolding N-def apply (rule Inf-lower, auto, rule exI[of - c **R f1 n], rule
exI[of - c *R fo n])
using F(2)[of n] scaleR-add-right by auto
also have ... = |c|*(eNorm N1 (f1 n) +eNorm N2 (f2 n))
by (auto simp add: algebra-simps eNorm-cmult)
finally show ?thesis using F(1) by simp
qed
moreover have ... \longrightarrow
unfolding N-def by (auto intro!: tendsto-intros simp add: uf(2))
ultimately show ?thesis
using LIMSEQ-le-const by blast
qed

```
have Inf \(\{e N o r m\) N1 f1 \(+e N o r m\) N2 f2| f1 f2. \(0=f 1+f 2\} \leq 0\)

> by (rule Inf-lower, auto, rule exI[of-0], auto)
> then have \(Z\) : Inf \(\{\) eNorm N1 f1 + eNorm N2 f2| f1 f2. \(0=f 1+f 2\}=0\)
> by auto
have *: quasinorm-on UNIV (max (defect N1) (defect N2)) ( \(\lambda f\). Inf \{eNorm N1 \(f 1+e\) Norm N2 f2| f1 f2. \(f=f 1+f 2\})\)
apply (rule quasinorm-onI) using THZ defect-ge-1[of N1] defect-ge-1[of N2] unfolding \(N\)-def by auto
show defect \(\left(N 1+_{N} N 2\right)=\max (\) defect \(N 1)(\) defect \(N 2)\)
\(e \operatorname{Norm}\left(N 1+_{N}\right.\) N2) \(f=\operatorname{Inf}\{e N o r m N 1 f 1+e N o r m N 2 f 2 \mid f 1 f 2 . f=f 1\)
\(+f 2\}\)
unfolding quasinorm-sum-def using quasinorm-of \([O F *]\) by auto qed
lemma quasinorm-sum-limit:
\(\exists f 1\) f2. \((\forall n . f=f 1 n+f 2 n) \wedge(\lambda n . e N o r m \operatorname{N1}(f 1 n)+e N o r m\) N2 \((f 2 n))\)
\(\longrightarrow e N o r m\left(N 1+_{N} N 2\right) f\)
proof -
have \(\exists u .(\forall n . u n \in\{e N o r m\) N1 f1 + eNorm N2 f2| f1 f2. \(f=f 1+f 2\}) \wedge u\) \(\rightarrow\) Inf \(\{e N o r m\) N1 f1 \(+e N o r m\) N2 f2| \(\mathrm{f1}\) f2. \(f=f 1+f 2\}\)
by (rule Inf-as-limit, auto, rule exI[of-f], rule exI[of - 0], auto)
then obtain uf where \(u f: \bigwedge n\). uf \(n \in\{e N o r m\) N1 f1 \(+e\) Norm N2 f2|f1 f2. \(f\)
\(=f 1+f 2\}\)
\[
u f \longrightarrow \operatorname{Inf}\{e N o r m \text { N1 } f 1+e \text { Norm N2 f2| } f 1 \text { f2. } f=f 1
\]
\(+f 2\}\)
by blast
have \(\exists f 1\) f2. \(\forall n\). uf \(n=e \operatorname{Norm} N 1(f 1 n)+e \operatorname{Norm} N 2(f 2 n) \wedge f=f 1 n+f 2\) n
apply (rule SMT.choices(1)) using uf(1) by blast
then obtain f1 f2 where \(F: \backslash n\). uf \(n=e \operatorname{Norm} N 1(f 1 n)+e N o r m\) N2 (f2 \(n\) )
\(\wedge n . f=f 1 n+f 2 n\)
by blast
have \((\lambda n\). eNorm N1 \((f 1 n)+e \operatorname{Norm}\) N2 \((f 2 n)) \longrightarrow e N o r m ~\left(N 1+{ }_{N} N 2\right) f\)
using \(F(1)\) uf(2) unfolding quasinorm-sum(1) by presburger
then show ?thesis using \(F\) (2) by auto
qed
lemma quasinorm-sum-space:
\(\operatorname{space}_{N}\left(N 1+_{N} N 2\right)=\left\{f+g \mid f g . f \in\right.\) space \(_{N} N 1 \wedge g \in\) space \(\left._{N} N 2\right\}\)
proof (auto)
fix \(x\) assume \(x \in \operatorname{space}_{N}\left(N 1+_{N} N 2\right)\)
then have Inf \(\{e N o r m\) N1 \(f+e N o r m\) N2 \(g \mid f g . x=f+g\}<\infty\) unfolding quasinorm-sum(1) spaceN-iff.
then have \(\exists z \in\{e N o r m\) N1 \(f+e\) Norm N2 \(g \mid f g . x=f+g\} . z<\infty\) by (simp add: Inf-less-iff)
then show \(\exists f g . x=f+g \wedge f \in\) space \(_{N} N 1 \wedge g \in\) space \(_{N} N 2\)
using spaceN-iff by force
next
fix \(f g\) assume \(H: f \in\) space \(_{N} N 1 g \in\) space \(_{N} N 2\)
```

    have Inf \(\{e N o r m\) N1 \(u+e \operatorname{Norm} N 2 v \mid u v . f+g=u+v\} \leq e N o r m N 1 f+\)
    eNorm N2 $g$
by (rule Inf-lower, auto)
also have $\ldots<\infty$ using space $N$-iff $H$ by auto
finally show $f+g \in \operatorname{space}_{N}\left(N 1+_{N} N 2\right)$
unfolding space $N$-iff quasinorm-sum(1).
qed
lemma quasinorm-sum-zerospace:
$\left\{f+g \mid f g . f \in\right.$ zero-space $_{N} N 1 \wedge g \in$ zero-space $\left._{N} N 2\right\} \subseteq$ zero-space $_{N}\left(N 1+_{N}\right.$
N2)
proof (auto, unfold zero-spaceN-iff)
fix $f g$ assume $H$ : eNorm N1 $f=0$ eNorm N2 $g=0$
have Inf $\{e$ Norm N1 $f 1+e N o r m$ N2 f2| $f 1$ f2. $f+g=f 1+f 2\} \leq 0$
by (rule Inf-lower, auto, rule exI[of-f], auto simp add: $H$ )
then show eNorm ( $\left.N 1+_{N} N 2\right)(f+g)=0$ unfolding quasinorm-sum(1) by
auto
qed
lemma quasinorm-sum-subset:
$N 1 \subseteq_{N} N 1+_{N} N 2 N 2 \subseteq_{N} N 1+_{N} N 2$
by (rule quasinorm-subsetI[of--1], auto simp add: quasinorm-sum(1), rule Inf-lower,
auto,
metis add.commute add.left-neutral eNorm-zero)+
lemma quasinorm-sum-maximum:
assumes $N 1 \subseteq_{N} N N 2 \subseteq_{N} N$
shows $N 1+_{N} N 2 \subseteq_{N} N$
proof -
obtain C1 C2::real where $*: \bigwedge f$. eNorm $N f \leq C 1 *$ eNorm N1 $f$
\f. eNorm $N f \leq C 2 *$ eNorm N2 $f$
$C 1 \geq 0 C 2 \geq 0$
using quasinorm-subsetD[OF assms(1)] quasinorm-subsetD[OF assms(2)] by
blast
have $* *$ : eNorm $N f \leq(\operatorname{defect} N * \max C 1 C 2) * \operatorname{eNorm}\left(N 1+_{N} N 2\right) f$ for $f$
proof -
obtain $f 1$ f2 where $F$ : $\bigwedge n . f=f 1 n+f 2 n$
$(\lambda n$. eNorm N1 $(f 1 n)+e N o r m$ N2 $(f 2 n)) \longrightarrow e N o r m$
$\left(N 1+{ }_{N} N 2\right) f$
using quasinorm-sum-limit by blast
have eNorm $N f \leq$ ennreal (defect $N * \max C 1 C 2) *(e N o r m N 1(f 1 n)+$
eNorm N2 (f2 $n$ )) for $n$
proof -
have $e \operatorname{Norm} N f \leq \operatorname{ennreal}(\operatorname{defect} N) * \operatorname{eNorm} N(f 1 n)+\operatorname{ennreal}(\operatorname{defect} N)$

* eNorm $N$ (f2 n)
unfolding $\langle f=f 1 n+f 2 n\rangle$ using eNorm-triangular-ineq by auto
also have $\ldots \leq \operatorname{ennreal}(\operatorname{defect} N) *(C 1 * \operatorname{eNorm} N 1(f 1 n))+$ ennreal(defect
$N) *(C 2 *$ eNorm N2 (f2 n))
apply (rule add-mono) by (rule mult-mono, simp, simp add: *, simp, simp) +

```
also have \(\ldots \leq \operatorname{ennreal}(\) defect \(N) *(\max C 1 C 2 * e N o r m ~ N 1(f 1 n))+\) ennreal \((\operatorname{defect} N) *(\max C 1 C 2 * e N o r m ~ N 2(f 2 n))\)
by (auto intro!:add-mono mult-mono ennreal-leI)
also have \(\ldots=\) ennreal (defect \(N * \max C 1\) C2) \(*(e N o r m ~ N 1(f 1 n)+\) eNorm N2 (f2 n) )
apply (subst ennreal-mult') using defect-ge-1 order-trans zero-le-one apply
blast
by (auto simp add: algebra-simps)
finally show? ?thesis by simp
qed
moreover have \(\ldots \longrightarrow(\) defect \(N * \max C 1 C 2) * \operatorname{eNorm}\left(N 1+_{N} N 2\right) f\)
by (auto intro!:tendsto-intros F(2))
ultimately show ?thesis
using LIMSEQ-le-const by blast
qed
then show?thesis
using quasinorm-subsetI by force
qed
lemma quasinorm-sum-assoc:
\(\left(N 1+_{N} N 2\right)+_{N} N 3={ }_{N} N 1+_{N}\left(N 2+_{N} N 3\right)\)
unfolding quasinorm-equivalent-def by (meson order-trans quasinorm-sum-maximum quasinorm-sum-subset)

\subsection*{2.7 Topology}
definition topology \(n::(\) 'a::real-vector) quasinorm \(\Rightarrow\) 'a topology
where topology \(N=\) topology \((\lambda U . \forall x \in U . \exists e>0 . \forall y\). eNorm \(N(y-x)<e\) \(\longrightarrow y \in U)\)
lemma istopology-topology \({ }_{N}\) :
istopology \((\lambda U . \forall x \in U . \exists e>0 . \forall y\). eNorm \(N(y-x)<e \longrightarrow y \in U)\)
unfolding istopology-def by (auto, metis dual-order.strict-trans less-linear, meson)
lemma openin-topology \({ }_{N}\) :
openin \(\left(\right.\) topology \(\left.{ }_{N} N\right) U \longleftrightarrow(\forall x \in U . \exists e>0 . \forall y . e N o r m ~ N(y-x)<e \longrightarrow y \in\) U)
unfolding topology \(N_{N}\)-def using istopology-topology \({ }_{N}[\) of \(N]\) by (simp add: topol-ogy-inverse')
lemma openin-topology \({ }_{N}-I\) :
assumes \(\bigwedge x . x \in U \Longrightarrow \exists e>0 . \forall y . e N o r m N(y-x)<e \longrightarrow y \in U\)
shows openin \(\left(\right.\) topology \(\left._{N} N\right) U\)
using assms unfolding openin-topology \(y_{N}\) by auto
lemma openin-topology \({ }_{N}-D\) :
assumes openin \(\left(\right.\) topology \(\left._{N} N\right) U\)
\[
x \in U
\]
\[
\text { shows } \exists e>0 . \forall y . e N o r m ~ N(y-x)<e \longrightarrow y \in U
\]
using assms unfolding openin-topology \(N_{N}\) by auto
One should then use this topology to define limits and so on. This is not something specific to quasinorms, but to all topologies defined in this way, not using type classes. However, there is no such body of material (yet?) in Isabelle-HOL, where topology is essentially done with type classes. So, we do not go any further for now.
One exception is the notion of completeness, as it is so important in functional analysis. We give a naive definition, which will be sufficient for the proof of completeness of several spaces. Usually, the most convenient criterion to prove completeness of a normed vector space is in terms of converging series. This criterion is the only nontrivial thing we prove here. We will apply it to prove the completeness of \(L^{p}\) spaces.
```

definition cauchy-ine ${ }_{N}::\left({ }^{\prime} a::\right.$ real-vector $)$ quasinorm $\Rightarrow\left(n a t \Rightarrow{ }^{\prime} a\right) \Rightarrow$ bool
where cauchy-ine ${ }_{N} N u=(\forall e>0 . \exists M . \forall n \geq M . \forall m \geq M$. eNorm $N(u n-u$
$m)<e$ )

```
definition tendsto-ine \({ }_{N}::\left({ }^{\prime} a::\right.\) real-vector) quasinorm \(\Rightarrow\left(n a t \Rightarrow{ }^{\prime} a\right) \Rightarrow{ }^{\prime} a=>\) bool
    where tendsto-ine \(N\) N \(x=(\lambda n\). eNorm \(N(u n-x)) \longrightarrow 0\)
definition complete \({ }_{N}:\) ('a::real-vector) quasinorm \(\Rightarrow\) bool
    where complete \(_{N} N=\left(\forall u\right.\). cauchy-ine \(_{N} N u \longrightarrow\left(\exists x\right.\). tendsto-ine \(\left.\left._{N} N u x\right)\right)\)

The above definitions are in terms of eNorms, but usually the nice definitions only make sense on the space of the norm, and are expressed in terms of Norms. We formulate the same definitions with norms, they will be more convenient for the proofs.
```

definition cauchy-in}\mp@subsup{N}{N}{}::('a::real-vector) quasinorm = (nat => 'a) => boo
where cauchy-in}\mp@subsup{|}{N}{}Nu=(\foralle>0.\existsM.\foralln\geqM.\forallm\geqM.Norm N(un-um
<e)

```

```

    where tendsto-in}\mp@subsup{N}{N}{}Nux=(\lambdan.Norm N(un-x))\longrightarrow
    lemma cauchy-ine }\mp@subsup{N}{N}{}-I
assumes \bigwedgee.e>0\Longrightarrow(\existsM.\foralln\geqM.\forallm\geqM.eNorm N(un-um)<e)
shows cauchy-ine N}NN
using assms unfolding cauchy-ine N
lemma cauchy-in}\mp@subsup{N}{N}{}-I
assumes \bigwedgee. e>0\Longrightarrow(\existsM.\foralln\geqM.\forallm\geqM.Norm N(un-um)<e)
shows cauchy-in}\mp@subsup{n}{N}{}N
using assms unfolding cauchy-in}\mp@subsup{N}{N}{}-def by aut
lemma cauchy-ine-in:

```
```

    assumes \n.un < space}\mp@subsup{N}{N}{N
    shows cauchy-ine N}N|u\longleftrightarrowcauchy-in N N
    proof
assume cauchy-in}\mp@subsup{N}{N}{}N
show cauchy-ine }\mp@subsup{N}{N}{Nu
proof (rule cauchy-ine N
fix e::ennreal assume e>0
define e2 where e2 = min e 1
then obtain r where e2 = ennreal r r>0 unfolding e2-def using <e>0\rangle
by (metis ennreal-eq-1 ennreal-less-zero-iff le-ennreal-iff le-numeral-extra(1)
min-def zero-less-one)
then obtain M where *: \foralln\geqM. \forallm\geqM.Norm N (un-um)<r
using <cauchy-in}\mp@subsup{n}{N}{}Nu\rangle\langler>0\rangle\mathrm{ unfolding cauchy-in }\mp@subsup{N}{N}{}-def by aut
then have }\foralln\geqM.\forallm\geqM. eNorm N(un-um)<
by (auto simp add: assms eNorm-Norm <0<r〉 ennreal-lessI)
then have }\foralln\geqM.\forallm\geqM. eNorm N (un-um)<
unfolding <e2 = ennreal r>[symmetric] e2-def by auto
then show }\existsM.\foralln\geqM.\forallm\geqM. eNorm N(un-um)<
by auto
qed
next
assume cauchy-ine N}NN
show cauchy-in N}N
proof (rule cauchy-in N}-I
fix e::real assume e>0
then obtain M where *: }\foralln\geqM.\forallm\geqM. eNorm N (un-um)<
using <cauchy-ine N}N|\rangle\langlee>0\rangle ennreal-less-zero-iff unfolding cauchy-ine (N-def
by blast
then have }\foralln\geqM.\forallm\geqM.Norm N(un-um)<
by (auto, metis Norm-def {0<e\rangle eNorm-Norm eNorm-Norm' enn2real-nonneg
ennreal-less-iff)
then show }\existsM.\foralln\geqM.\forallm\geqM.Norm N(un-um)<
by auto
qed
qed
lemma tendsto-ine-in:
assumes \n.un\in space N}N\x\in\mp@subsup{\mathrm{ space }}{N}{}
shows tendsto-ine }NN|x\longleftrightarrow\mp@subsup{t}{N}{}Ndsto-i\mp@subsup{n}{N}{}Nu
proof -
have *: eNorm N (un-x) = Norm N (un-x) for n
using assms eNorm-Norm spaceN-diff by blast
show ?thesis unfolding tendsto-in}\mp@subsup{N}{N}{}\mathrm{ -def tendsto-ine }\mp@subsup{N}{N}{}\mathrm{ -def *
apply (auto)
apply (metis (full-types) Norm-nonneg ennreal-0 eventually-sequentiallyI or-
der-refl tendsto-ennreal-iff)
using tendsto-ennrealI by fastforce
qed

```

\section*{lemma complete \(_{N}-I\) ：}
assumes \(\bigwedge u\). cauchy－in \(_{N} N u \Longrightarrow\left(\forall n . u n \in\right.\) space \(\left._{N} N\right) \Longrightarrow\left(\exists x \in\right.\) space \(_{N} N\) ． tendsto－in \(\left.N_{N} N u x\right)\)
shows complete \(_{N} N\)
proof－
have \(\exists x\) ．tendsto－ine \(N_{N} N u x\) if cauchy－ine \(N_{N} N u\) for \(u\)
proof－
obtain \(M\) ：：nat where \(*: \bigwedge n m . n \geq M \Longrightarrow m \geq M \Longrightarrow e N o r m N(u n-u\)
\(m)<1\)
using 〈cauchy－ine \({ }_{N} N\) u〉 ennreal－zero－less－one unfolding cauchy－ine \({ }_{N}\)－def
by presburger
define \(v\) where \(v=(\lambda n . u(n+M)-u M)\)
have eNorm \(N(v n)<1\) for \(n\) unfolding \(v\)－def using \(*\) by auto
then have \(v n \in\) space \(_{N} N\) for \(n\) using space \(N\)－iff \([o f-N]\)
by（metis dual－order．strict－trans ennreal－1 ennreal－less－top infinity－ennreal－def）
have cauchy－ine \({ }_{N} N v\)
proof（rule cauchy－ine \(N_{N}-I\) ）
fix \(e\) ：：ennreal assume \(e>0\)
then obtain \(P:: n\) at where \(*: \bigwedge n m . n \geq P \Longrightarrow m \geq P \Longrightarrow e N o r m ~ N(u n\) \(-u m)<e\)
using 〈cauchy－ine \(N_{N} N u\) 〉 unfolding cauchy－ine \({ }_{N}\)－def by presburger
have \(e \operatorname{Norm} N(v n-v m)<e\) if \(n \geq P m \geq P\) for \(m n\)
unfolding \(v\)－def by（auto，rule \(*\) ，insert that，auto）
then show \(\exists M . \forall n \geq M . \forall m \geq M . e N o r m ~ N(v n-v m)<e\) by auto
qed
then have cauchy－in \(N_{N} N v\) using cauchy－ine－in［OF 〈 \(\backslash n . v n \in\) space \(\left.\left._{N} N\right\rangle\right]\)
by auto
then obtain \(y\) where tendsto－in \(N v y y \in\) space \(_{N} N\)
using assms 〈 \(\backslash n\) ．v \(n \in\) space \(_{N} N\) b by auto
then have \(*\) ：tendsto－ine \(N_{N} N v y\)
using tendsto－ine－in \(\left\langle\bigwedge n\right.\) ．v \(n \in\) space \(\left._{N} N\right\rangle\) by auto
have tendsto－ine \(N_{N} N u(y+u M)\)
unfolding tendsto－ine \({ }_{N}\)－def apply（rule LIMSEQ－offset \([o f-M]\) ）
using＊unfolding \(v\)－def tendsto－ine \(N_{N}\)－def by（auto simp add：algebra－simps）
then show ？thesis by auto
qed
then show ？thesis unfolding complete \(N_{N}\) def by auto
qed
lemma cauchy－tendsto－in－subseq：
assumes \(\bigwedge n . u n \in\) space \(_{N} N\)
cauchy－in \(_{N} N u\)
strict－mono \(r\)
tendsto－in \(N\)（u or \() x\)
shows tendsto－in \(N\) \(N x\)
proof－
have \(\exists M . \forall n \geq M\) ．Norm \(N(u n-x)<e\) if \(e>0\) for \(e\)
proof－
define \(f\) where \(f=e /(2 * \operatorname{defect} N)\)
have \(f>0\) unfolding \(f\)－def using \(\langle e>0\rangle\) defect－ge－ 1 ［of \(N\) ］by（auto simp add：divide－simps）
obtain \(M 1\) where \(M 1: \bigwedge m n . m \geq M 1 \Longrightarrow n \geq M 1 \Longrightarrow \operatorname{Norm} N(u n-u\) \(m)<f\)
using 〈cauchy－in \(\left.n_{N} N u\right\rangle\) unfolding cauchy－in \(n_{N}\) def using \(\langle f>0\rangle\) by meson
obtain M2 where M2：\(\wedge n . n \geq M 2 \Longrightarrow \operatorname{Norm} N((u\) or \() n-x)<f\)
using＜tendsto－in \(\left.\left.N_{N} N\left(\begin{array}{lll}u & o & r\end{array}\right) x\right\rangle\langle f\rangle 0\right\rangle\) unfolding tendsto－in \(N_{N}\)－def or－ der－tendsto－iff eventually－sequentially by blast
define \(M\) where \(M=\max M 1\) M2
have \(\operatorname{Norm} N(u n-x)<e\) if \(n \geq M\) for \(n\)
proof－
have \(\operatorname{Norm} N(u n-x)=\operatorname{Norm} N((u n-u(r M))+(u(r M)-x))\) by auto
also have \(\ldots \leq \operatorname{defect} N * \operatorname{Norm} N(u n-u(r M))+\operatorname{defect} N * \operatorname{Norm} N\) \((u(r M)-x)\)
apply（rule Norm－triangular－ineq）using 〈 \(\backslash n\) ．u \(n \in\) space \(\left._{N} N\right\rangle\) by simp
also have \(\ldots<\operatorname{defect} N * f+\operatorname{defect} N * f\)
apply（auto intro！：add－strict－mono mult－mono simp only：）
using defect－ge－1［of \(N]\langle n \geq M\rangle\) seq－suble \([O F\langle\) strict－mono r〉，of M］M1
M2 o－def unfolding \(M\)－def by auto
finally show ？thesis
unfolding \(f\)－def using \(\langle e>0\rangle\) defect－ge－1［of \(N]\) by（auto simp add：
divide－simps）
qed
then show ？thesis by auto
qed
then show ？thesis
unfolding tendsto－in \({ }_{N}\)－def order－tendsto－iff eventually－sequentially using Norm－nonneg less－le－trans by blast
qed
proposition complete \({ }_{N}-I^{\prime}\) ：
assumes \(\bigwedge n . c n>0\)
\(\bigwedge u .\left(\forall n . u n \in \operatorname{space}_{N} N\right) \Longrightarrow(\forall n\). Norm \(N(u n) \leq c n) \Longrightarrow \exists x \in\)
space \(_{N} N\). tendsto－in \(N_{N} N\left(\lambda n .\left(\sum i \in\{0 . .<n\} . u i\right)\right) x\)
shows complete \({ }_{N} N\)
proof（rule complete \({ }_{N}-I\) ）
fix \(v\) assume cauchy－in \(N v \forall n\) ．v \(n \in \operatorname{space}_{N} N\)
have \(*\) ：\(\exists y .(\forall m \geq y . \forall p \geq y\) ．Norm \(N(v m-v p)<c(\) Suc \(n)) \wedge x<y\) if
\(\forall m \geq x\) ．\(\forall p \geq x\) ．Norm \(N(v m-v p)<c n\) for \(x n\)
proof－
obtain \(M\) where \(i: \forall m \geq M . \forall p \geq M . \operatorname{Norm} N(v m-v p)<c(S u c n)\)
using 〈cauchy－in \(N\) v \({ }^{2}\langle c(S u c n)>0\rangle\) unfolding cauchy－in \(n_{N}\)－def by
（meson zero－less－power）
then show ？thesis
apply（intro exI［of \(-\max M(x+1)]\) ）by auto
qed
have \(\exists r . \forall n .(\forall m \geq r n . \forall p \geq r n . \operatorname{Norm} N(v m-v p)<c n) \wedge r n<r(S u c\) n）
apply（intro dependent－nat－choice）using \(\left\langle c a u c h y-i n_{N} N v\right\rangle\langle\backslash n . c n>0\rangle *\) unfolding cauchy－in \(n_{N}\)－def by auto
then obtain \(r\) where \(r\) ：strict－mono \(r \bigwedge n . \forall m \geq r n . \forall p \geq r n\) ．Norm \(N(v m-\) \(v p)<c n\)
by（auto simp：strict－mono－Suc－iff）
define \(u\) where \(u=(\lambda n . v(r(S u c n))-v(r n))\)
have \(u n \in\) space \(_{N} N\) for \(n\)
unfolding \(u\)－def using \(\left\langle\forall n . v n \in\right.\) space \(\left._{N} N\right\rangle\) by simp
moreover have Norm \(N(u n) \leq c n\) for \(n\)
unfolding \(u\)－def using \(r\) by（simp add：less－imp－le strict－mono－def）
ultimately obtain \(y\) where \(y: y \in\) space \(_{N} N\) tendsto－in \(_{N} N\left(\lambda n .\left(\sum i \in\{0 . .<n\}\right.\right.\) ． \(u\) i））\(y\)
using assms（2）by blast
define \(x\) where \(x=y+v(r 0)\)
have \(x \in\) space \(_{N} N\)
unfolding \(x\)－def using \(\left\langle y \in\right.\) space \(\left._{N} N\right\rangle\left\langle\forall n\right.\) ．v \(n \in\) space \(\left._{N} N\right\rangle\) by simp
have Norm \(N(v(r n)-x)=\operatorname{Norm} N\left(\left(\sum i \in\{0 . .<n\} . u i\right)-y\right)\) for \(n\)
proof－
have \(v(r n)=\left(\sum i \in\{0 . .<n\} . u i\right)+v(r 0)\) for \(n\) unfolding \(u\)－def by（induct \(n\) ，auto）
then show ？thesis unfolding \(x\)－def by（metis add－diff－cancel－right）
qed
then have \((\lambda n\) ．Norm \(N(v(r n)-x)) \longrightarrow 0\) using \(y\)（2）unfolding tendsto－in \(N_{N}\) def by auto
then have tendsto－in \(N\)（vor）x unfolding tendsto－in \(n_{N}\)－def comp－def by force
then have tendsto－in \(N_{N} N v x\)
using \(\left\langle\forall n . v n \in\right.\) space \(\left._{N} N\right\rangle\)
by（intro cauchy－tendsto－in－subseq［OF－〈cauchy－in \(N_{N} N\) v〉〈strict－mono \(\left.\left.r\right\rangle\right]\) ， auto）
then show \(\exists x \in\) space \(_{N} N\) ．tendsto－in \(N\) N vx using \(\left\langle x \in\right.\) space \(\left._{N} N\right\rangle\) by blast
qed
Next，we show when the two examples of norms we have introduced before， the ambient norm in a Banach space，and the norm on bounded continuous functions，are complete．We just have to translate in our setting the already known completeness of these spaces．
```

lemma complete-N-of-norm:
complete N (N-of-norm::'a::banach quasinorm)
proof (rule complete N}N\mathrm{ -I)
fix u::nat => 'a assume cauchy-in }\mp@subsup{N}{N}{}N\mathrm{ -of-norm u
then have Cauchy u unfolding Cauchy-def cauchy-in}\mp@subsup{N}{N}{}\mathrm{ -def N-of-norm(2) by
(simp add: dist-norm)
then obtain x where }u\longrightarrowx\mathrm{ using convergent-eq-Cauchy by blast
then have tendsto-in}\mp@subsup{N}{N}{}N\mathrm{ -of-norm u x unfolding tendsto-in}\mp@subsup{N}{N}{}\mathrm{ -def N-of-norm(2)
using Lim-null tendsto-norm-zero-iff by fastforce
moreover have x\in space}\mp@subsup{N}{N}{N
ultimately show \existsx\inspace N}NN\mathrm{ -of-norm. tendsto-in }\mp@subsup{N}{N}{}N\mathrm{ -of-norm u x by auto

```

\section*{qed}

In the next statement, the assumption that 'a is a metric space is not necessary, a topological space would be enough, but a statement about uniform convergence is not available in this setting. TODO: fix it.
```

lemma complete-bcontfunN:

```
    complete \(_{N}\left(\right.\) bcontfun \(_{N}::\left({ }^{\prime} a::\right.\) metric-space \(\Rightarrow\) 'b::banach) quasinorm)
proof (rule complete \({ }_{N}-I\) )
    fix \(u:: n a t \Rightarrow(' a \Rightarrow ' b)\) assume \(H:\) cauchy-in \(_{N} b^{\prime} \operatorname{contfun}_{N} u \forall n . u n \in\) space \(_{N}\)
bcontfun \({ }_{N}\)
    then have H2: \(u n \in\) bcontfun for \(n\) using bcontfun \(_{N}\)-space by auto
    then have \(* *\) : Bcontfun ( \(u n-u m\) ) \(=B\) contfun ( \(u n\) ) - Bcontfun ( \(u m\) ) for
\(m n\)
    unfolding minus-fun-def minus-bcontfun-def by (simp add: Bcontfun-inverse)
    have \(*\) : Norm bcontfun \(N_{N}(u n-u m)=\) norm (Bcontfun \((u n-u m)\) ) for \(n m\)
        unfolding bcontfun \(N_{N}\) (2) using \(H\) (2) bcontfun \(N_{N}\)-space by auto
    have Cauchy ( \(\lambda\) n. Bcontfun ( \(u n\) ) )
        using \(H(1)\) unfolding Cauchy-def cauchy-in \(n_{N}\)-def dist-norm \(* * *\) by simp
    then obtain \(v\) where \(v:(\lambda n\). Bcontfun \((u n)) \longrightarrow v\)
        using convergent-eq-Cauchy by blast
    have \(v\)-space: apply-bcontfun \(v \in\) space \(_{N}\) bcontfun \(_{N}\) unfolding bcontfun \(N_{N}\)-space
by (simp add: apply-bcontfun)
    have \(* * *\) : Norm bcontfun \(N_{N}(u n-v)=\operatorname{norm}(B \operatorname{contfun}(u n)-v)\) for \(n\)
    proof -
        have Norm bcontfun \(N_{N}(u n-v)=\) norm \((B c o n t f u n(u n-v))\)
            unfolding bcontfun \(N_{N}(2)\) using \(H(2)\) bcontfun \(_{N}\)-space \(v\)-space by auto
        moreover have Bcontfun ( \(u n-v\) ) \(=B\) contfun ( \(u n\) ) \(-v\)
            unfolding minus-fun-def minus-bcontfun-def by (simp add: Bcontfun-inverse
H2)
            ultimately show?thesis by simp
    qed
    have tendsto-in \({ }_{N}\) bcontfun \(n_{N} u v\)
            unfolding tendsto-in \(N_{N}\) def \(* * *\) using \(v\) Lim-null tendsto-norm-zero-iff by
fastforce
    then show \(\exists v \in\) space \(_{N}\) bcontfun \(_{N}\). tendsto-in \({ }_{N}\) bcontfun \(_{N} u v\) using \(v\)-space by
auto
qed
end
theory \(L p\)
imports Functional-Spaces
begin
The material in this file is essentially of analytic nature. However, one of the central proofs (the proof of Holder inequality below) uses a probability space, and Jensen's inequality there. Hence, we need to import Probability. Moreover, we use several lemmas from SG_Library_Complement.

\section*{3 Conjugate exponents}

Two numbers \(p\) and \(q\) are conjugate if \(1 / p+1 / q=1\). This relation keeps appearing in the theory of \(L^{p}\) spaces, as the dual of \(L^{p}\) is \(L^{q}\) where \(q\) is the conjugate of \(p\). This relation makes sense for real numbers, but also for ennreals (where the case \(p=1\) and \(q=\infty\) is most important). Unfortunately, manipulating the previous relation with ennreals is tedious as there is no good simproc involving addition and division there. To mitigate this difficulty, we prove once and for all most useful properties of such conjugates exponents in this paragraph.
```

lemma Lp-cases-1-PInf:
assumes p\geq(1::ennreal)
obtains (gr) p2 where p= ennreal p2 p2 > 1 p>1
| (one) p=1
| (PInf) p=\infty

```
using assms by (metis (full-types) antisym-conv ennreal-cases ennreal-le-1 infin-
ity-ennreal-def not-le)
lemma Lp-cases:
    obtains (real-pos) \(p 2\) where \(p=\) ennreal \(p 2 p 2>0 p>0\)
        \(\mid\) (zero) \(p=0\)
    | (PInf) \(p=\infty\)
by (metis enn2real-positive-iff ennreal-enn2real-if infinity-ennreal-def not-gr-zero
top.not-eq-extremum)

\section*{definition}
```

    conjugate-exponent \(p=1+1 /(p-1)\)
    ```
lemma conjugate-exponent-real:
    assumes \(p>(1::\) real \()\)
    shows \(1 / p+1 /(\) conjugate-exponent \(p)=1\)
        conjugate-exponent \(p>1\)
        conjugate-exponent(conjugate-exponent \(p)=p\)
        \((p-1) *\) conjugate-exponent \(p=p\)
    \(p-p /\) conjugate-exponent \(p=1\)
unfolding conjugate-exponent-def using assms by (auto simp add: algebra-simps
divide-simps)
lemma conjugate-exponent-real-iff:
    assumes \(p>(1::\) real \()\)
    shows \(q=\) conjugate-exponent \(p \longleftrightarrow(1 / p+1 / q=1)\)
unfolding conjugate-exponent-def using assms by (auto simp add: algebra-simps
divide-simps)
lemma conjugate-exponent-real-2 [simp]:
    conjugate-exponent \((2::\) real \()=2\)
    unfolding conjugate-exponent-def by (auto simp add: algebra-simps divide-simps)
lemma conjugate－exponent－realI：
assumes \(p>(0::\) real \() q>01 / p+1 / q=1\)
shows \(p>1 q=\) conjugate－exponent \(p q>1 p=\) conjugate－exponent \(q\)
unfolding conjugate－exponent－def using assms apply（auto simp add：algebra－simps divide－simps）
apply（metis assms（3）divide－less－eq－1－pos less－add－same－cancel1 zero－less－divide－1－iff） using mult－less－cancel－left－pos by fastforce
lemma conjugate－exponent－real－ennreal：
assumes \(p>(1::\) real \()\)
shows conjugate－exponent（ennreal p）\(=\) ennreal（conjugate－exponent \(p)\)
unfolding conjugate－exponent－def using assms
by（auto，metis diff－gt－0－iff－gt divide－ennreal ennreal－1 ennreal－minus zero－le－one）
lemma conjugate－exponent－ennreal－1－2－PInf \([\) simp \(]\) ：
conjugate－exponent \((1::\) ennreal \()=\infty\)
conjugate－exponent \((\infty::\) ennreal \()=1\)
conjugate－exponent \((\top\) ：：ennreal \()=1\)
conjugate－exponent \((2::\) ennreal \()=2\)
using conjugate－exponent－real－ennreal［of 2］by（auto simp add：conjugate－exponent－def）
lemma conjugate－exponent－ennreal：
assumes \(p \geq(1:\) ：ennreal \()\)
shows \(1 / p+1 /(\) conjugate－exponent \(p)=1\)
conjugate－exponent \(p \geq 1\)
conjugate－exponent（conjugate－exponent \(p)=p\)
proof－
have \((1 / p+1 /(\) conjugate－exponent \(p)=1) \wedge(\) conjugate－exponent \(p \geq 1) \wedge\)
conjugate－exponent \((\) conjugate－exponent \(p)=p\)
using \(\langle p \geq 1\rangle\) proof（cases rule：Lp－cases－1－PInf） case（ \(g r p 2\) ）
then have \(*\) ：conjugate－exponent \(p=\) ennreal（conjugate－exponent p2）using conjugate－exponent－real－ennreal \([O F\langle p 2>1\rangle]\) by auto
have \(a\) ：conjugate－exponent \(p \geq 1\) using \(*\) conjugate－exponent－real \([O F\langle p 2>\)
1）］by auto
have \(b\) ：conjugate－exponent（conjugate－exponent \(p)=p\)
using conjugate－exponent－real（3）［OF \(\langle p 2>1\rangle]\) conjugate－exponent－real－ennreal \([O F\) \(\langle p 2>1\rangle\) ］
conjugate－exponent－real－ennreal［OF conjugate－exponent－real（2）［OF \(\langle p 2>1\rangle]\) ］
unfolding \(*\langle p=\) ennreal \(p 2\rangle\) by auto
have \(1 / p+1 /\) conjugate－exponent \(p=\) ennreal \((1 / p 2+1 /(\) conjugate－exponent
p2））unfolding \(*\) unfolding \(\langle p=\) ennreal \(p 2\rangle\)
using conjugate－exponent－real（2）［OF〈p2＞1〉］〈p2＞1〉
apply（subst ennreal－plus，auto）apply（subst divide－ennreal［symmetric］，auto） using divide－ennreal－def inverse－ennreal inverse－eq－divide by auto
then have \(c: 1 / p+1 /\) conjugate－exponent \(p=1\) using conjugate－exponent－real［OF \(\langle p 2>1\rangle]\) by auto
show ？thesis using \(a b c\) by simp

> qed (auto)
then show \(1 / p+1 /(\) conjugate-exponent \(p)=1\)
conjugate-exponent \(p \geq 1\) conjugate-exponent(conjugate-exponent \(p)=p\)
by auto
qed
lemma conjugate-exponent-ennreal-iff:
assumes \(p \geq(1\) :: ennreal \()\)
shows \(q=\) conjugate-exponent \(p \longleftrightarrow(1 / p+1 / q=1)\)
using conjugate-exponent-ennreal[ OF assms]
by (auto, metis ennreal-add-diff-cancel-left ennreal-add-eq-top ennreal-top-neq-one one-divide-one-divide-ennreal)
lemma conjugate-exponent-ennrealI:
assumes \(1 / p+1 / q=(1:\) :ennreal \()\)
shows \(p \geq 1 q \geq 1 p=\) conjugate-exponent \(q\) q conjugate-exponent \(p\)
proof -
have \(1 / p \leq 1\) using assms using le-iff-add by fastforce
then show \(p \geq 1\)
by (metis assms divide-ennreal-def ennreal-add-eq-top ennreal-divide-self en-nreal-divide-zero ennreal-le-epsilon ennreal-one-neq-top mult.left-neutral mult-left-le zero-le)
then show \(q=\) conjugate-exponent \(p\) using conjugate-exponent-ennreal-iff assms by auto
then show \(q \geq 1\) using conjugate-exponent-ennreal \([O F\langle p \geq 1\rangle]\) by auto
show \(p=\) conjugate-exponent \(q\)
using conjugate-exponent-ennreal-iff \([O F\langle q \geq 1\rangle\), of \(p]\) assms by (simp add: add.commute)
qed

\section*{4 Convexity inequalities and integration}

In this paragraph, we describe the basic inequalities relating the integral of a function and of its \(p\)-th power, for \(p>0\). These inequalities imply in particular that the \(L^{p}\) norm satisfies the triangular inequality, a feature we will need when defining the \(L^{p}\) spaces below. In particular, we prove the Hölder and Minkowski inequalities. The Hölder inequality, especially, is the basis of all further inequalities for \(L^{p}\) spaces.
```

lemma (in prob-space) bound-L1-Lp:
assumes $p \geq(1::$ real $)$
$f \in$ borel-measurable $M$
integrable $M(\lambda x .|f x|$ powr $p)$
shows integrable $M f$
abs $\left(\int x . f x \partial M\right)$ powr $p \leq\left(\int x .|f x|\right.$ powr $\left.p \partial M\right)$
$\operatorname{abs}\left(\int x . f x \partial M\right) \leq\left(\int x .|f x|\right.$ powr $\left.p \partial M\right)$ powr $(1 / p)$
proof -

```
```

    have \(*\) : norm \(x \leq 1+(\) norm \(x)\) powr \(p\) for \(x:\) :real
    apply (cases norm \(x \leq 1\) )
    apply (meson le-add-same-cancel1 order.trans powr-ge-pzero)
    apply (metis add-le-same-cancel2 assms(1) less-le-trans linear not-less not-one-le-zero
    powr-le-cancel-iff powr-one-gt-zero-iff)
done
show $*$ : integrable $M f$
apply (rule Bochner-Integration.integrable-bound $[$ of $-\lambda x .1+|f x|$ powr $p]$,
auto simp add: assms) using $*$ by auto
show $\operatorname{abs}\left(\int x . f x \partial M\right)$ powr $p \leq\left(\int x .|f x|\right.$ powr $\left.p \partial M\right)$
by (rule jensens-inequality[OF *-assms(3) convex-abs-powr[OF $\langle p \geq 1\rangle]]$,
auto)
then have $\left(\operatorname{abs}\left(\int x . f x \partial M\right)\right.$ powr $\left.p\right)$ powr $(1 / p) \leq\left(\int x .|f x|\right.$ powr $\left.p \partial M\right)$ powr
( $1 / p$ )
using assms(1) powr-mono2 by auto
then show $\operatorname{abs}\left(\int x . f x \partial M\right) \leq\left(\int x .|f x|\right.$ powr $\left.p \partial M\right)$ powr $(1 / p)$
using $\langle p \geq 1\rangle$ by (auto simp add: powr-powr)
qed
theorem Holder-inequality:
assumes $p>(0::$ real $) q>01 / p+1 / q=1$
and [measurable]: $f \in$ borel-measurable $M g \in$ borel-measurable $M$
integrable $M(\lambda x .|f x|$ powr $p)$
integrable $M(\lambda x .|g x|$ powr $q)$
shows integrable $M(\lambda x . f x * g x)$
$\left(\int x .|f x * g x| \partial M\right) \leq\left(\int x .|f x|\right.$ powr $\left.p \partial M\right)$ powr $(1 / p) *\left(\int x .|g x|\right.$ powr
$q \partial M)$ powr (1/q)
$\operatorname{abs}\left(\int x . f x * g x \partial M\right) \leq\left(\int x .|f x|\right.$ powr $\left.p \partial M\right) \operatorname{powr}(1 / p) *\left(\int x .|g x|\right.$
powr $q \partial M)$ powr $(1 / q)$
proof -
have $p>1$ using conjugate-exponent-realI $(1)[O F\langle p>0\rangle\langle q>0\rangle\langle 1 / p+1 / q=1\rangle]$.
have $*: x * y \leq x$ powr $p+y$ powr $q$ if $x \geq 0 y \geq 0$ for $x y$
proof -
have $x * y=(x$ powr $p)$ powr $(1 / p) *(y$ powr $q)$ powr $(1 / q)$
using $\langle p>0\rangle\langle q>0\rangle$ powr-powr that(1) that(2) by auto
also have $\ldots \leq(\max (x$ powr $p)(y$ powr $q))$ powr $(1 / p) *(\max (x$ powr $p)(y$
powr $q)$ ) powr ( $1 / q$ )
apply (rule mult-mono, auto) using assms(1) assms(2) powr-mono2 by auto
also have $\ldots=\max (x$ powr $p)(y$ powr $q$ )
by (metis max-def mult.right-neutral powr-add powr-powr assms(3))
also have $\ldots \leq x$ powr $p+y$ powr $q$
by auto
finally show? ?thesis by simp
qed
show [simp]: integrable $M(\lambda x . f x * g x)$
apply (rule Bochner-Integration.integrable-bound $[$ of $-\lambda x$. $|f x|$ powr $p+|g x|$
powr q], auto)

```
by (rule Bochner-Integration.integrable-add, auto simp add: assms * abs-mult)
The proof of the main inequality is done by applying the inequality \(\left(\int|h| d \mu \leq\right.\) \(\left.\int|h|^{p} d \mu\right)^{1 / p}\) to the right function \(h\) in the right probability space. One should take \(h=f \cdot|g|^{1-q}\), and \(d \mu=|g|^{q} d M / I\), where \(I=\int|g|^{q}\). This readily gives the result.
```

    show *: ( \(\left.\int x .|f x * g x| \partial M\right) \leq\left(\int x .|f x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p) *\left(\int x .|g x|\right.\)
    powr $q$ дM) powr ( $1 / q$ )
proof (cases $\left(\int x .|g x|\right.$ powr $\left.\left.q \partial M\right)=0\right)$
case True
then have $A E x$ in $M .|g x|$ powr $q=0$
by (subst integral-nonneg-eq-0-iff-AE[symmetric], auto simp add: assms)
then have *: AE $x$ in M. $f x * g x=0$
using $\langle q>0\rangle$ by auto
have $\left(\int x .|f x * g x| \partial M\right)=\left(\int x .0 \partial M\right)$
apply (rule integral-cong-AE) using * by auto
then show ?thesis by auto
next
case False
moreover have $\left(\int x .|g x|\right.$ powr $\left.q \partial M\right) \geq\left(\int x .0 \partial M\right)$ by (rule integral-mono,
auto simp add: assms)
ultimately have $*:\left(\int x .|g x|\right.$ powr $\left.q \partial M\right)>0$ by (simp add: le-less)
define $I$ where $I=\left(\int x .|g x|\right.$ powr $\left.q \partial M\right)$
have [simp]: I $>0$ unfolding $I$-def using $*$ by auto
define $M 2$ where $M 2=\operatorname{density} M(\lambda x .|g x|$ powr $q / I)$
interpret prob-space M2
apply (standard, unfold M2-def, auto, subst emeasure-density, auto)
apply (subst divide-ennreal[symmetric], auto, subst nn-integral-divide, auto)
apply (subst nn-integral-eq-integral, auto simp add: assms, unfold I-def)
using * by auto
have $[$ simp $]: p \geq 1 p \geq 0$ using $\langle p>1\rangle$ by auto
have $A: q+(1-q) * p=0$ using assms by (auto simp add: divide-simps
algebra-simps)
have $B: 1-1 / p=1 / q$ using $\langle 1 / p+1 / q=1\rangle$ by auto
define $f 2$ where $f 2=(\lambda x . f x *$ indicator $\{y \in$ space $M . g y \neq 0\} x)$
have [measurable]: $f 2 \in$ borel-measurable $M$ unfolding $f 2$-def by auto
define $h$ where $h=(\lambda x .|f 2 x| *|g x|$ powr (1-q))
have [measurable]: $h \in$ borel-measurable $M$ unfolding $h$-def by auto
have [measurable]: $h \in$ borel-measurable M2 unfolding M2-def by auto
have Eq: $(|g x|$ powr $q / I) *_{R}|h x|$ powr $p=|f 2 x|$ powr $p / I$ for $x$
apply (insert $\langle I\rangle 0\rangle$, auto simp add: divide-simps, unfold $h$-def)
apply (auto simp add: divide-nonneg-pos divide-simps powr-mult powr-powr
powr-add[symmetric] A)
unfolding f2-def by auto
have integrable M2 ( $\lambda x .|h x|$ powr $p$ )
unfolding M2-def apply (subst integrable-density, simp, simp simp add:
divide-simps)

```
apply（subst Eq，rule integrable－divide，rule Bochner－Integration．integrable－bound［of －\(\lambda x\) ．\(|f x|\) powr \(p]\) ，unfold f2－def）
by（unfold indicator－def，auto simp add：＜integrable \(M(\lambda x .|f x|\) powr \(p)\rangle)\)
then have integrable M2（ \(\lambda x .|h x|)\)
by（metis bound－L1－Lp（1）〈random－variable borel \(h\rangle\langle p>1\rangle\) integrable－abs le－less）
have \(\left(\int x .|h x|\right.\) powr \(p\) 万M2 \()=\left(\int x .(|g x|\right.\) powr \(q / I) *_{R}(|h x|\) powr \(\left.p) \partial M\right)\) unfolding M2－def by（rule integral－density［of \(\lambda x\) ．\(|h x|\) powr \(p M \lambda x .|g x|\)
powr \(q / I\) ］，auto simp add：divide－simps）
also have \(\ldots=\left(\int x . \mid\right.\) fo \(x \mid\) powr \(\left.p / I \partial M\right)\)
apply（rule Bochner－Integration．integral－cong）using Eq by auto
also have \(\ldots \leq\left(\int x .|f x|\right.\) powr \(\left.p / I \partial M\right)\)
apply（rule integral－mono＇，rule integrable－divide［OF＜integrable \(M(\lambda x .|f x|\) powr \(p)\rangle\) ］）
unfolding f2－def indicator－def using \(\langle I>0\rangle\) by（auto simp add：divide－simps）
finally have \(C:\left(\int x .|h x|\right.\) powr \(p\) 万M2 \() \leq\left(\int x .|f x|\right.\) powr \(\left.p / I \partial M\right)\) by simp
have \(\left(\int x .|f x * g x| \partial M\right) / I=\left(\int x .|f x * g x| / I \partial M\right)\)
by auto
also have \(\ldots=\left(\int x .|f 2 x * g x| / I \partial M\right)\)
by（auto simp add：divide－simps，rule Bochner－Integration．integral－cong，unfold f2－def indicator－def，auto）
also have \(\ldots=\left(\int x .|h x| \partial M 2\right)\)
apply（unfold M2－def，subst integral－density，simp，simp，simp add：di－ vide－simps）
by（rule Bochner－Integration．integral－cong，unfold h－def，auto simp add：di－ vide－simps algebra－simps powr－add［symmetric］abs－mult）
also have \(\ldots \leq a b s\left(\int x .|h x|\right.\) วM2）
by auto
also have \(\ldots \leq\left(\int x\right.\) ．abs \((|h x|)\) powr \(p\) дM2）powr \((1 / p)\)
apply（rule bound－L1－Lp（3）［of \(p \lambda x .|h x|])\)
by（auto simp add：＜integrable M2（ \(\lambda x .|h x|\) powr \(p)\rangle\) ）
also have \(\ldots \leq\left(\int x .|f x|\right.\) powr \(\left.p / I \partial M\right)\) powr \((1 / p)\)
by（rule powr－mono2，insert \(C\) ，auto）
also have \(\ldots \leq\left(\left(\int x .|f x|\right.\right.\) powr \(\left.\left.p \partial M\right) / I\right)\) powr \((1 / p)\)
apply（rule powr－mono2，auto simp add：divide－simps）using \(\langle p \geq 0\rangle\) by auto
also have \(\ldots=\left(\int x .|f x| \operatorname{powr} p \partial M\right) \operatorname{powr}(1 / p) * I \operatorname{powr}(-1 / p)\)
by（auto simp add：less－imp－le powr－divide powr－minus－divide）
finally have \(\left(\int x .|f x * g x| \partial M\right) \leq\left(\int x .|f x|\right.\) powr \(\left.p \partial M\right) \operatorname{powr}(1 / p) * I *\) I powr（ \(-1 / p\) ）
by（auto simp add：divide－simps algebra－simps）
also have \(\ldots=\left(\int x .|f x| \operatorname{powr} p \partial M\right) \operatorname{powr}(1 / p) * \operatorname{Ipowr}(1-1 / p)\)
by（auto simp add：powr－mult－base less－imp－le）
also have \(\ldots=\left(\int x .|f x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p) *\left(\int x .|g x|\right.\) powr \(\left.q \partial M\right)\) powr（1／q）
unfolding \(I\)－def using \(B\) by auto
finally show ？thesis
```

    by simp
    qed
    have abs(\intx.fx*gx\partialM)\leq(\intx. |fx*g x|\partialM) by auto
    then show abs(\intx.fx*gx\partialM)\leq(\intx. |fx| powr p \partialM) powr (1/p)*(\intx.
    |g x powr q \partialM) powr (1/q)
using * by linarith
qed
theorem Minkowski-inequality:
assumes p\geq(1::real)
and [measurable, simp]: f\in borel-measurable Mg
integrable M ( }\lambdax.|fx| powr p
integrable M ( }\lambdax.|gx| powr p
shows integrable M ( }\lambdax.|fx+gx| powr p
(\intx. |fx+gx| powr p \partialM) powr (1/p)
\leq(\intx. |f x powr p \partialM) powr (1/p)+(\intx. |g x| powr p \partialM) powr (1/p)
proof -
have *: |x + y| powr p\leq2 powr p * (|x| powr p + |y| powr p) for x y::real
proof -
have }|x+y|\leq|x|+|y| by aut
also have ... \leq(\operatorname{max}|x| |y|)+\operatorname{max}|x| |y| by auto
also have ... =2* max |x| |y| by auto
finally have }|x+y|\mathrm{ powr }p\leq(2*\operatorname{max}|x| |y|) powr
using powr-mono2 <p\geq1> by auto
also have ... = 2 powr p* (\operatorname{max}|x| |y|) powr p
using powr-mult by auto
also have .. \leq2 powr p * (|x| powr p + |y| powr p)
unfolding max-def by auto
finally show ?thesis by simp
qed
show [simp]: integrable M ( }\lambdax.|fx+gx| powr p
by (rule Bochner-Integration.integrable-bound[of - \lambdax. 2 powr p * (|f x powr p

+ |g x| powr p)], auto simp add: *)
show (\intx. |fx+g x| powr p \partialM) powr (1/p)\leq(\intx. |f x| powr p \partialM) powr
(1/p)+(\intx.|gx| powr p \partialM) powr (1/p)
proof (cases p=1)
case True
then show ?thesis
apply (auto, subst Bochner-Integration.integral-add[symmetric], insert assms(4)
assms(5), simp, simp)
by (rule integral-mono', auto)
next
case False
then have [simp]: p>1 p\geq1p>0 p\not=0 using assms(1) by auto
define q}\mathrm{ where q= conjugate-exponent p
have [simp]: q>1 q>0 1/p+1/q=1(p-1)*q=p
unfolding q-def using conjugate-exponent-real[OF <p>1\rangle] by auto
then have [simp]:(z powr (p-1)) powr q = z powr p for z

```
by (simp add: powr-powr)
have \(\left(\int x .|f x+g x|\right.\) powr p \(\left.\partial M\right)=\left(\int x .|f x+g x| *|f x+g x|\right.\) powr \((p-1)\) \(\partial M)\)
by (subst powr-mult-base, auto)
also have \(\ldots \leq\left(\int x .|f x| *|f x+g x|\right.\) powr \((p-1)+|g x| *|f x+g x|\) powr \((p-1) \partial M)\)
apply (rule integral-mono', rule Bochner-Integration.integrable-add)
apply (rule Holder-inequality (1) \([\) of \(p q]\), auto)
apply (rule Holder-inequality(1)[of p q], auto)
by (metis abs-ge-zero abs-triangle-ineq comm-semiring-class.distrib le-less mult-mono' powr-ge-pzero)
also have \(\ldots=\left(\int x .|f x| *|f x+g x| \operatorname{powr}(p-1) \partial M\right)+\left(\int x .|g x| * \mid f x+\right.\) \(g x \mid \operatorname{powr}(p-1) \partial M)\)
apply (rule Bochner-Integration.integral-add) by (rule Holder-inequality(1) [of p q], auto) +
also have \(\ldots \leq a b s\left(\int x .|f x| *|f x+g x| \operatorname{powr}(p-1) \partial M\right)+a b s\left(\int x .|g x|\right.\) * \(|f x+g x| \operatorname{powr}(p-1) \partial M)\)
by auto
also have \(\ldots \leq\left(\int x\right.\). abs \((|f x|)\) powr \(\left.p \partial M\right)\) powr \((1 / p) *\left(\int x . a b s(|f x+g x|\right.\) powr \((p-1))\) powr \(q \partial M)\) powr \((1 / q)\)
\[
+\left(\int x \cdot \operatorname{abs}(|g x|) \text { powr } p \partial M\right) \text { powr }(1 / p) *\left(\int x \cdot a b s(|f x+g x|\right.
\]
powr \((p-1))\) powr \(q\) дM) powr \((1 / q)\)
apply (rule add-mono)
apply (rule Holder-inequality(3)[of p \(q]\), simp, simp, simp, simp, simp, simp, simp)
apply (rule Holder-inequality(3)[of p q], simp, \(\operatorname{simp}, \operatorname{simp}, \operatorname{simp}, \operatorname{simp}, \operatorname{simp}\), \(\operatorname{simp}\) )
done
also have \(\ldots=\left(\int x .|f x+g x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / q) *\)
\(\left(\left(\int x . a b s(|f x|)\right.\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p)+\left(\int x . a b s(|g x|)\right.\) powr p \(\left.\partial M\right)\) powr \((1 / p))\)
by (auto simp add: algebra-simps)
finally have \(*:\left(\int x .|f x+g x|\right.\) powr \(\left.p \partial M\right) \leq\left(\int x .|f x+g x|\right.\) powr \(\left.p \partial M\right)\) powr ( \(1 / q\) ) *
(( \(\int x\). abs \((|f x|)\) powr \(\left.p \partial M\right)\) powr \((1 / p)+\left(\int x . a b s(|g x|)\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p))\)
by \(\operatorname{simp}\)
show ?thesis
proof (cases \(\left(\int x .|f x+g x|\right.\) powr \(\left.\left.p \partial M\right)=0\right)\)
case True
then show ?thesis by auto
next
case False
then have \(* *:\left(\int x .|f x+g x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / q)>0\)
by auto
have \(\left(\int x .|f x+g x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / q) *\left(\int x .|f x+g x|\right.\) powr \(\left.p \partial M\right)\) powr (1/p)
\[
=\left(\int x \cdot|f x+g x| \text { powr } p \partial M\right)
\]
by (auto simp add: powr-add[symmetric] add.commute)
```

    then have \(\left(\int x .|f x+g x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / q) *\left(\int x .|f x+g x|\right.\) powr
    p $\partial M)$ powr $(1 / p) \leq$
$\left(\int x .|f x+g x|\right.$ powr $\left.p \partial M\right)$ powr $(1 / q)$ *
$\left(\left(\int x . a b s(|f x|)\right.\right.$ powr p $\left.\partial M\right)$ powr $(1 / p)+\left(\int x . a b s(|g x|)\right.$ powr $\left.p \partial M\right)$
powr $(1 / p))$
using * by auto
then show ?thesis using ** by auto
qed
qed
qed

```

When \(p<1\), the function \(x \mapsto|x|^{p}\) is not convex any more. Hence, the \(L^{p}\) "norm" is not a norm any more, but a quasinorm. This is proved using a different convexity argument, as follows.
theorem Minkowski-inequality-le-1:
assumes \(p>(0::\) real \() p \leq 1\)
and [measurable, simp]: \(f \in\) borel-measurable \(M g \in\) borel-measurable \(M\)
integrable \(M(\lambda x .|f x|\) powr \(p)\)
integrable \(M(\lambda x .|g x|\) powr \(p)\)
shows integrable \(M(\lambda x .|f x+g x|\) powr \(p)\) ( \(\int x .|f x+g x|\) powr \(p \partial M\) ) powr ( \(1 / p\) )
\(\leq 2 \operatorname{powr}(1 / p-1) *\left(\int x .|f x| \operatorname{powr} p \partial M\right)\) powr \((1 / p)+2 \operatorname{powr}(1 / p-1)\)
* ( \(\int x .|g x|\) powr \(\left.p \partial M\right)\) powr \((1 / p)\)
proof -
have \(*:|a+b|\) powr \(p \leq|a|\) powr \(p+|b|\) powr \(p\) for \(a b\)
using \(x\)-plus- \(y\)-p-le-xp-plus-yp \([O F\langle p>0\rangle\langle p \leq 1\rangle\), of \(|a||b|]\)
by (auto, meson abs-ge-zero abs-triangle-ineq assms(1) le-less order.trans powr-mono2)
show integrable \(M(\lambda x .|f x+g x|\) powr \(p)\)
by (rule Bochner-Integration.integrable-bound \([\) of \(-\lambda x .|f x|\) powr \(p+|g x|\) powr p], auto simp add:*)
have \(\left(\int x .|f x+g x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p) \leq\left(\int x .|f x|\right.\) powr \(p+|g x|\) powr p \(\partial M)\) powr \((1 / p)\)
by (rule powr-mono2, simp add: \(\langle p>0\rangle\) less-imp-le, simp, rule integral-mono', auto simp add: *)
also have \(\ldots=2\) powr \((1 / p) *\left(\left(\left(\int x .|f x|\right.\right.\right.\) powr \(\left.p \partial M\right)+\left(\int x .|g x|\right.\) powr \(p\) \(\partial M)\) ) / 2) powr ( \(1 / p\) )
by (auto simp add: powr-mult[symmetric] add-divide-distrib)
also have \(\ldots \leq 2\) powr \((1 / p) *\left(\left(\left(\int x .|f x|\right.\right.\right.\) powr p \(\left.\partial M\right) \operatorname{powr}(1 / p)+\left(\int x . \mid g\right.\) \(x \mid\) powr \(p \partial M)\) powr \((1 / p)) / 2)\)
apply (rule mult-mono, simp, rule convex-on-mean-ineq[OF convex-powr[of \(1 / p]]\) )
using \(\langle p \leq 1\rangle\langle p>0\rangle\) by auto
also have \(\ldots=2\) powr \((1 / p-1) *\left(\left(\int x .|f x|\right.\right.\) powr \(p\) 2M \() \operatorname{powr}(1 / p)+\left(\int x\right.\).
\(|g x|\) powr p \(\partial M)\) powr \((1 / p))\)
by (simp add: powr-diff)
finally show \(\left(\int x .|f x+g x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p)\) \(\leq 2 \operatorname{powr}(1 / p-1) *\left(\int x .|f x| \operatorname{powr} p \partial M\right) \operatorname{powr}(1 / p)+2 \operatorname{powr}(1 / p-1)\)
* \(\left(\int x .|g x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p)\)
```

    by (auto simp add: algebra-simps)
    qed

```

\section*{\(5 \quad L^{p}\) spaces}

We define \(L^{p}\) spaces by giving their defining quasinorm. It is a norm for \(p \in[1, \infty]\), and a quasinorm for \(p \in(0,1)\). The construction of a quasinorm from a formula only makes sense if this formula is indeed a quasinorm, i.e., it is homogeneous and satisfies the triangular inequality with the given multiplicative defect. Thus, we have to show that this is indeed the case to be able to use the definition.
definition \(L p\)-space::ennreal \(\Rightarrow{ }^{\prime} a\) measure \(\Rightarrow\left({ }^{\prime} a \Rightarrow\right.\) real \()\) quasinorm where \(L p\)-space \(p M=(\)
if \(p=0\) then quasinorm-of \((1,(\lambda f\). if \((f \in\) borel-measurable \(M)\) then 0 else \(\infty)\) )
else if \(p<\infty\) then quasinorm-of (
if \(p<1\) then 2 powr \((1 /\) enn2real \(p-1)\) else 1 ,
( \(\lambda f\). if \((f \in\) borel-measurable \(M \wedge\) integrable \(M(\lambda x .|f x|\) powr (enn2real p)))
```

                    then \(\left(\int x .|f x|\right.\) powr (enn2real \(\left.\left.p\right) \partial M\right)\) powr (1/(enn2real p))
    ```
                    else \((\infty::\) ennreal \()\) )
else quasinorm-of \((1,(\lambda f\). if \(f \in\) borel-measurable \(M\) then esssup \(M(\lambda x\). ereal \(|f x|)\) else ( \(\infty::\) ennreal \()\) )))
abbreviation \(\mathfrak{L}==\) Lp-space

\section*{\(5.1 \quad L^{\infty}\)}

Let us check that, for \(L^{\infty}\), the above definition makes sense.
```

lemma L-infinity:
eNorm $(\mathfrak{L} \infty M) f=($ if $f \in$ borel-measurable $M$ then esssup $M(\lambda x$. ereal $|f x|)$
else ( $\infty::$ ennreal))
$\operatorname{defect}(\mathfrak{L} \infty M)=1$
proof -
have $T$ : esssup $M(\lambda x$. ereal $|(f+g) x|) \leq$ e2ennreal (esssup $M(\lambda x$. ereal $\mid f$
$x \mid))+\operatorname{esssup} M(\lambda x$. ereal $|g x|)$
if [measurable]: $f \in$ borel-measurable $M g \in$ borel-measurable $M$ for $f g$
proof (cases emeasure $M($ space $M)=0)$
case True
then have e2ennreal $(\operatorname{esssup} M(\lambda x$. ereal $|(f+g) x|))=0$
using esssup-zero-space[OF True] by (simp add: e2ennreal-neg)
then show ?thesis by simp
next
case False
have $*: \operatorname{esssup} M(\lambda x .|h x|) \geq 0$ for $h::^{\prime} a \Rightarrow$ real
proof -
have $\operatorname{esssup} M(\lambda x .0) \leq \operatorname{esssup} M(\lambda x .|h x|)$ by (rule esssup-mono, auto)

```
```

        then show ?thesis using esssup-const[OF False, of 0::ereal] by simp
    qed
    have esssup M (\lambdax. ereal |(f+g) x|) \leq esssup M ( }\lambdax\mathrm{ . ereal }|fx|+\operatorname{ereal}|gx|
    by (rule esssup-mono, auto simp add: plus-fun-def)
    also have ... \leq esssup M ( }\lambda\mathrm{ x. ereal |f x |) + esssup M ( }\lambdax\mathrm{ . ereal |gx|)
        by (rule esssup-add)
    finally show ?thesis
        using * by (simp add: e2ennreal-mono eq-onp-def plus-ennreal.abs-eq)
    qed
    ```
    have \(H: \operatorname{esssup} M\left(\lambda x\right.\). ereal \(\left.\left|\left(c *_{R} f\right) x\right|\right) \leq\) ennreal \(|c| * \operatorname{esssup} M(\lambda x\). ereal \(\mid f\)
\(x \mid)\) if \(c \neq 0\) for \(f c\)
    proof -
        have abs \(c>0\) ereal \(|c| \geq 0\) using that by auto
        have \(*: \operatorname{esssup} M\left(\lambda x . \operatorname{abs}\left(c *_{R} f x\right)\right)=a b s c * \operatorname{esssup} M(\lambda x .|f x|)\)
            apply (subst esssup-cmult \([O F\langle a b s ~ c>0\rangle\), of \(M \lambda\). ereal \(|f x|\), symmetric \(]\) )
            using times-ereal.simps(1) by (auto simp add: abs-mult)
    show ?thesis
        unfolding e2ennreal-mult \([\) OF <ereal \(|c| \geq 0\rangle] *\) scaleR-fun-def
        by \(\operatorname{simp}\)
    qed
have esssup \(M(\lambda x\). ereal 0\() \leq 0\) using esssup- \(I\) by auto
then have \(Z\) : e2ennreal (esssup \(M(\lambda\). ereal 0\())=0\) using e2ennreal-neg by auto
have *: quasinorm-on (borel-measurable M) \(1\left(\lambda\left(f::^{\prime} a \Rightarrow\right.\right.\) real \()\). e2ennreal(esssup \(M(\lambda x\). ereal \(|f x|)))\)
apply (rule quasinorm-onI) using \(T H Z\) by auto
have **: quasinorm-on UNIV \(1\left(\lambda\left(f::^{\prime} a \Rightarrow\right.\right.\) real \()\). if \(f \in\) borel-measurable \(M\) then e2ennreal(esssup \(M(\lambda x\). ereal \(|f x|))\) else \(\infty)\)
by (rule extend-quasinorm [OF *])
show eNorm \((\mathfrak{L} \infty M) f=(\) if \(f \in\) borel-measurable \(M\) then e2ennreal(esssup \(M\)
\((\lambda x .|f x|))\) else \(\infty)\)
defect \((\mathfrak{L} \infty M)=1\)
unfolding Lp-space-def using quasinorm-of \([O F * *]\) by auto
qed
lemma L-infinity-space:
space \(_{N}(\mathfrak{L} \infty M)=\{f \in\) borel-measurable \(M . \exists C . A E x\) in \(M .|f x| \leq C\}\)
proof (auto simp del: infinity-ennreal-def)
fix \(f\) assume \(H: f \in \operatorname{space}_{N}(\mathfrak{L} \infty M)\)
then show \(f \in\) borel-measurable \(M\)
unfolding space \(_{N}\)-def using L-infinity(1)[of \(\left.M\right]\) top.not-eq-extremum by fast-
force
then have \(*: \operatorname{esssup} M(\lambda x .|f x|)<\infty\)
using \(H\) unfolding space \(_{N}\)-def L-infinity(1)[of M] by (auto simp add: e2en-nreal-infty)
define \(C\) where \(C=\operatorname{real-of-\operatorname {ereal}(\operatorname {esssup}M(\lambda x.|fx|)),~(\lambda )}\)
have \(A E x\) in \(M\). ereal \(|f x| \leq\) ereal \(C\)
proof (cases emeasure \(M(\) space \(M)=0)\)
case True
then show ?thesis using emeasure- \(0-A E\) by simp
next
case False
then have esssup \(M(\lambda x .|f x|) \geq 0\)
using esssup-mono[of \(\lambda x .0 M \overline{(\lambda x} .|f x|)]\) esssup-const[OF False, of \(0::\) ereal] by auto
then have esssup \(M(\lambda x .|f x|)=\) ereal \(C\) unfolding \(C\)-def using \(*\) ereal-real by auto
then show ?thesis using esssup- \(A E[\) of ( \(\lambda\) x. ereal \(|f x|) M]\) by simp
qed
then have \(A E x\) in \(M .|f x| \leq C\) by auto
then show \(\exists C\). \(A E x\) in \(M .|f x| \leq C\) by blast
next
fix \(f::^{\prime} a \Rightarrow\) real and \(C::\) real
assume \(H: f \in\) borel-measurable \(M A E x\) in \(M .|f x| \leq C\)
then have esssup \(M(\lambda x .|f x|) \leq C\) using esssup- \(I\) by auto
then have \(e \operatorname{Norm}(\mathfrak{L} \infty M) f \leq C\) unfolding L-infinity(1) using \(H(1)\)
by auto (metis e2ennreal-ereal e2ennreal-mono)
then show \(f \in\) space \(_{N}(\mathfrak{L} \infty M)\)
using space \(N\)-iff le-less-trans by fastforce
qed
lemma L-infinity-zero-space:
zero-space \(_{N}(\mathfrak{L} \infty M)=\{f \in\) borel-measurable \(M\). AE x in M. fx=0\}
proof (auto simp del: infinity-ennreal-def)
fix \(f\) assume \(H: f \in\) zero-space \(_{N}(\mathfrak{L} \infty M)\)
then show \(f \in\) borel-measurable \(M\)
unfolding zero-space \({ }_{N}\)-def using L-infinity(1)[of \(\left.M\right]\) top.not-eq-extremum by fastforce
then have *: e2ennreal (esssup \(M(\lambda x .|f x|))=0\)
using \(H\) unfolding zero-space \(_{N}\)-def using L-infinity(1)[of \(M\) ] e2ennreal-infty by auto
then have esssup \(M(\lambda x .|f x|) \leq 0\)
by (metis e2ennreal-infty e2ennreal-mult ennreal-top-neq-zero ereal-mult-infty leI linear mult-zero-left)
then have \(f x=0\) if ereal \(|f x| \leq \operatorname{esssup} M(\lambda x .|f x|)\) for \(x\)
using that order.trans by fastforce
then show \(A E x\) in \(M\). \(f x=0\) using esssup \(-A E[\) of \(\lambda x\). ereal \(|f x| M]\) by auto next
fix \(f::^{\prime} a \Rightarrow\) real
assume \(H: f \in\) borel-measurable \(M A E x\) in \(M . f x=0\)
then have esssup \(M(\lambda x\). \(|f x|) \leq 0\) using esssup- \(I\) by auto
then have eNorm \((\mathfrak{L} \infty M) f=0\) unfolding L-infinity(1) using \(H(1)\)
by (simp add: e2ennreal-neg)
then show \(f \in\) zero-space \(_{N}(\mathfrak{L} \infty M)\)
using zero-space \(N\)-iff by auto
```

qed
lemma L-infinity-AE-ebound:
AEx in M. ennreal }|fx|\leqeNorm (L L M)
proof (cases f\in borel-measurable M)
case False
then have eNorm ({L }\inftyM)f=
unfolding L-infinity(1) by auto
then show ?thesis by simp
next
case True
then have ennreal |fx| \leqeNorm (\mathfrak{L}\inftyM)f if |fx| \leq esssup M ( \lambdax. |f x|) for
x
unfolding L-infinity(1) using that e2ennreal-mono
by fastforce
then show ?thesis using esssup-AE[of \lambdax. ereal |f x|] by force
qed
lemma L-infinity-AE-bound:
assumes f\in\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}\inftyM)
shows AE x in M. |fx| \leq Norm ({L\inftyM)f
using L-infinity-AE-ebound[of f M] unfolding eNorm-Norm[OF assms] by (simp)
In the next lemma, the assumption $C \geq 0$ that might seem useless is in fact necessary for the second statement when the space has zero measure. Indeed, any function is then almost surely bounded by any constant!

```
```

lemma L-infinity-I:

```
lemma L-infinity-I:
    assumes \(f \in\) borel-measurable \(M\)
    assumes \(f \in\) borel-measurable \(M\)
        \(A E x\) in \(M .|f x| \leq C\)
        \(A E x\) in \(M .|f x| \leq C\)
        \(C \geq 0\)
        \(C \geq 0\)
    shows \(f \in\) space \(_{N}(\mathfrak{L} \infty M)\)
    shows \(f \in\) space \(_{N}(\mathfrak{L} \infty M)\)
        Norm \((\mathfrak{L} \infty M) f \leq C\)
        Norm \((\mathfrak{L} \infty M) f \leq C\)
proof -
proof -
    show \(f \in\) space \(_{N}(\mathfrak{L} \infty M)\)
    show \(f \in\) space \(_{N}(\mathfrak{L} \infty M)\)
        using L-infinity-space assms(1) assms(2) by force
        using L-infinity-space assms(1) assms(2) by force
    have esssup \(M(\lambda x .|f x|) \leq C\) using assms(1) assms(2) esssup-I by auto
    have esssup \(M(\lambda x .|f x|) \leq C\) using assms(1) assms(2) esssup-I by auto
    then have \(\operatorname{eNorm}(\mathfrak{L} \infty M) f \leq\) ereal \(C\)
    then have \(\operatorname{eNorm}(\mathfrak{L} \infty M) f \leq\) ereal \(C\)
        unfolding L-infinity(1) using assms(1) e2ennreal-mono by force
        unfolding L-infinity(1) using assms(1) e2ennreal-mono by force
    then have ennreal \((\operatorname{Norm}(\mathfrak{L} \infty M) f) \leq\) ennreal \(C\)
    then have ennreal \((\operatorname{Norm}(\mathfrak{L} \infty M) f) \leq\) ennreal \(C\)
        using eNorm-Norm \(\left[O F\left\langle f \in \operatorname{space}_{N}(\mathfrak{L} \infty M)\right\rangle\right]\) assms(3) by auto
        using eNorm-Norm \(\left[O F\left\langle f \in \operatorname{space}_{N}(\mathfrak{L} \infty M)\right\rangle\right]\) assms(3) by auto
    then show \(\operatorname{Norm}(\mathfrak{L} \infty M) f \leq C\) using assms(3) by auto
    then show \(\operatorname{Norm}(\mathfrak{L} \infty M) f \leq C\) using assms(3) by auto
qed
qed
lemma L-infinity-I':
lemma L-infinity-I':
    assumes [measurable]: \(f \in\) borel-measurable \(M\)
    assumes [measurable]: \(f \in\) borel-measurable \(M\)
        and \(A E x\) in \(M\). ennreal \(|f x| \leq C\)
        and \(A E x\) in \(M\). ennreal \(|f x| \leq C\)
    shows \(e \operatorname{Norm}(\mathfrak{L} \infty M) f \leq C\)
    shows \(e \operatorname{Norm}(\mathfrak{L} \infty M) f \leq C\)
proof -
proof -
    have esssup \(M(\lambda x .|f x|) \leq\) enn2ereal \(C\)
```

    have esssup \(M(\lambda x .|f x|) \leq\) enn2ereal \(C\)
    ```
apply (rule esssup-I, auto) using assms(2) less-eq-ennreal.rep-eq by auto
then show ?thesis unfolding L-infinity using assms apply auto
using e2ennreal-mono by fastforce
qed
lemma L-infinity-pos-measure:
assumes [measurable]: \(f \in\) borel-measurable \(M\) and eNorm \((\mathfrak{L} \infty M) f>(C::\) real \()\)
shows emeasure \(M\{x \in\) space \(M .|f x|>C\}>0\)
proof -
have *: esssup \(M(\lambda x\). ereal \((|f x|))>\operatorname{ereal} C\) using \(\langle e N o r m ~(\mathfrak{L} \infty M) f>C\) 〉 unfolding L-infinity
proof (auto)
assume a1: ennreal \(C<\) e2ennreal (esssup \(M(\lambda x\). ereal \(|f x|)\) )
have \(\neg\) e2ennreal \((\operatorname{esssup} M(\lambda a\). ereal \(|f a|)) \leq e 2 e n n r e a l(\) ereal \(C)\) if \(\neg C<\) 0
using a1 that by (metis (no-types) e2ennreal-enn2ereal enn2ereal-ennreal leD

\section*{leI)}
then have e2ennreal \((\) esssup \(M(\lambda a\). ereal \(|f a|)) \leq e 2\) 2ennreal \((\) ereal \(C) \longrightarrow\) \((\exists e \leq\) esssup \(M(\lambda a\). ereal \(|f a|)\). ereal \(C<e)\) using a1 e2ennreal-neg by fastforce

\section*{then show?thesis} by (meson e2ennreal-mono leI less-le-trans)
qed
have emeasure \(M\{x \in\) space \(M\). ereal \((|f x|)>C\}>0\)
by (rule esssup-pos-measure[OF - *], auto)
then show ?thesis by auto
qed
lemma L-infinity-tendsto-AE:
assumes tendsto-in \({ }_{N}(\mathfrak{L} \infty M) f g\)
\(\bigwedge n . f n \in \operatorname{space}_{N}(\mathfrak{L} \infty M)\)
\(g \in \operatorname{space}_{N}(\mathfrak{L} \infty M)\)
shows \(A E x\) in \(M .(\lambda n . f n x) \longrightarrow g x\)
proof -
have \(*: A E x\) in \(M .|(f n-g) x| \leq \operatorname{Norm}(\mathfrak{L} \infty M)(f n-g)\) for \(n\)
apply (rule L-infinity-AE-bound) using assms spaceN-diff by blast
have \(A E x\) in \(M . \forall n .|(f n-g) x| \leq \operatorname{Norm}(\mathfrak{L} \infty M)(f n-g)\)
apply (subst AE-all-countable) using \(*\) by auto
moreover have \((\lambda n . f n x) \longrightarrow g x\) if \(\forall n .|(f n-g) x| \leq \operatorname{Norm}(\mathfrak{L} \infty M)\)
\((f n-g)\) for \(x\)
proof -
have \((\lambda n .|(f n-g) x|) \longrightarrow 0\)
apply (rule tendsto-sandwich[of \(\lambda n\). \(0-\lambda n . \operatorname{Norm}(\mathfrak{L} \infty M)(f n-g)])\)
using that <tendsto-in \(n_{N}(\mathfrak{L} \infty M) f g\) unfolding tendsto-in \(n_{N}\)-def by auto
then have \((\lambda n .|f n x-g x|) \longrightarrow 0\) by auto
then show ?thesis
by \((\) simp add: \(\langle(\lambda n .|f n x-g x|) \longrightarrow 0\rangle\) LIM-zero-cancel tendsto-rabs-zero-cancel \()\)
qed

\section*{ultimately show ?thesis by auto} qed

As an illustration of the mechanism of spaces inclusion, let us show that bounded continuous functions belong to \(L^{\infty}\).
```

lemma bcontfun-subset-L-infinity:
assumes sets M = sets borel
shows space}N\mp@code{bcontfun}NN\subseteq\mp@subsup{\mathrm{ space }}{N}{}(\mathfrak{L}\inftyM
\.f\in\mp@subsup{\mathrm{ space }}{N}{}\mp@subsup{b}{contfun}{N}\Longrightarrow\LongrightarrowNorm (\mathfrak{L}\inftyM)f\leqNorm bcontfun}\mp@subsup{N}{N}{}
\f.eNorm (\mathfrak{L}\inftyM)f\leqeNorm bcontfun}\mp@subsup{N}{N}{}

```

```

proof -
have *: f\in\mp@subsup{\mathrm{ space }}{N}{}(\mathfrak{L}\inftyM)\wedgeNorm (\mathfrak{L}\inftyM)f\leqNorm bcontfun}\mp@subsup{N}{N}{}f\mathrm{ if }f
space}\mp@subsup{N}{N}{}\mp@subsup{b}{contfun}{N}\mathrm{ for f
proof -
have H: continuous-on UNIV f \x. abs(f x ) \leqNorm bcontfun}\mp@subsup{N}{N}{}
using bcontfun}\mp@subsup{N}{N}{}D[OF<f\in\mp@subsup{\mathrm{ space N}}{N}{}\mp@subsup{b}{contfun}{N}
then have f}\in\mathrm{ borel-measurable borel using borel-measurable-continuous-onI
by simp
then have f}\in\mathrm{ borel-measurable M using assms by auto
have *: AE x in M. |f x| \leqNorm bcontfun N f using H(2) by auto
show ?thesis using L-infinity-I[OF <f \in borel-measurable M>* Norm-nonneg]
by auto
qed
show space}\mp@subsup{N}{N bcontfun}{N}\subseteq\subseteq\mp@subsup{\mathrm{ space}}{N}{}(\mathfrak{L}\inftyM
\f.f\in\mp@subsup{\mathrm{ space }}{N}{}\mp@subsup{b}{contfun}{N}\Longrightarrow\LongrightarrowNorm (\mathfrak{L}\inftyM)f\leqNormbcontfun}\mp@subsup{N}{N}{}
using * by auto
show **: bcontfun}\mp@subsup{N}{N}{}\mp@subsup{\subseteq}{N}{}\mathfrak{L}\infty
apply (rule quasinorm-subsetI'[of - - 1]) using * by auto
have eNorm ( }\mathfrak{L}\inftyM)f\leq\mathrm{ ennreal 1 * eNorm bcontfun}\mp@subsup{N}{N}{}f\mathrm{ for }
apply (rule quasinorm-subset-Norm-eNorm) using * ** by auto
then show eNorm ({ L M ) f\leqeNorm bcontfun}\mp@subsup{N}{N}{}f\mathrm{ for f by simp
qed

```

\section*{5.2 \(L^{p}\) for \(0<p<\infty\)}
lemma \(L p\) :
    assumes \(p \geq(1::\) real \()\)
    shows eNorm \((\mathfrak{L} p M) f=(\) if \((f \in\) borel-measurable \(M \wedge\) integrable \(M(\lambda x . \mid f\)
\(x \mid\) powr \(p)\) )
            then \(\left(\int x .|f x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p)\)
            else ( \(\infty::\) ennreal) \()\)
        \(\operatorname{defect}(\mathfrak{L} p M)=1\)
proof -
    define \(F\) where \(F=\{f \in\) borel-measurable \(M\). integrable \(M(\lambda x .|f x|\) powr \(p)\}\)
    have \(*\) : quasinorm-on \(F 1\left(\lambda\left(f:^{\prime} a \Rightarrow\right.\right.\) real \()\). \(\left(\int x .|f x|\right.\) powr \(p\) dM) powr \(\left.(1 / p)\right)\)
    proof (rule quasinorm-onI)
    fix \(f g\) assume \(f \in F g \in F\)
    then show \(f+g \in F\)
unfolding \(F\)-def plus-fun-def apply (auto) by (rule Minkowski-inequality(1), auto simp add: \(\langle p \geq 1\rangle\) )
show ennreal \(\left(\left(\int x .|(f+g) x|\right.\right.\) powr \(\left.p \partial M\right)\) powr \(\left.(1 / p)\right)\)
\(\leq\) ennreal \(1 *\left(\int x .|f x|\right.\) powr p \(\left.\partial M\right)\) powr \((1 / p)+\) ennreal \(1 *\left(\int x .|g x|\right.\) powr \(p \partial M\) ) powr ( \(1 / p\) )
apply (auto, subst ennreal-plus[symmetric], simp, simp, rule ennreal-leI)
unfolding plus-fun-def apply (rule Minkowski-inequality(2) [of pf \(M\) g], auto simp add: \(\langle p \geq 1\rangle\) )
using \(\langle f \in F\rangle\langle g \in F\rangle\) unfolding \(F\)-def by auto
next
fix \(f\) and \(c:\) :real assume \(f \in F\)
show \(c *_{R} f \in F\) using \(\langle f \in F\rangle\) unfolding scaleR-fun-def F-def by (auto simp add: abs-mult powr-mult)
show \(\left(\int x .\left|\left(c *_{R} f\right) x\right|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p) \leq \operatorname{ennreal}(\operatorname{abs}(c)) *\left(\int x .|f x|\right.\) powr \(p \partial M)\) powr ( \(1 / p\) )
apply (rule eq-refl, subst ennreal-mult[symmetric], simp, simp, rule en-nreal-cong)
apply (unfold scaleR-fun-def, simp add: abs-mult powr-mult powr-powr) using \(\langle p \geq 1\rangle\) by auto
next
show \(0 \in F\) unfolding zero-fun-def \(F\)-def by auto
qed (auto)
have \(p \geq 0\) using \(\langle p \geq 1\) 〉 by auto
have \(* *: \mathfrak{L} p M=\) quasinorm-of ( 1 ,
\((\lambda f\). if \((f \in\) borel-measurable \(M \wedge\) integrable \(M(\lambda x .|f x|\) powr \(p))\)
then ( \(\int x .|f x|\) powr \(p \partial M\) ) powr ( \(1 / p\) )
else ( \(\infty:\) : ennreal) \()\) )
unfolding Lp-space-def using enn2real-ennreal[OF \(\langle p \geq 0\rangle]\langle p \geq 1\rangle\) apply auto
using enn2real-ennreal \([O F\langle p \geq 0\rangle]\) by presburger
show eNorm \((\mathfrak{L} p M) f=(\) if \((f \in\) borel-measurable \(M \wedge\) integrable \(M(\lambda x .|f x|\) powr \(p\) ))
then \(\left(\int x .|f x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p)\) else ( \(\infty::\) ennreal \()\) )
\(\operatorname{defect}(\mathfrak{L} p M)=1\)
unfolding \({ }^{* *}\) using quasinorm-of[OF extend-quasinorm[OF *]] unfolding \(F\)-def by auto
qed
lemma \(L p-l e-1\) :
assumes \(p>0 p \leq(1:\) :real \()\)
shows \(e \operatorname{Norm}(\mathfrak{L} p M) f=(\) if \((f \in\) borel-measurable \(M \wedge\) integrable \(M(\lambda x . \mid f\) \(x \mid\) powr \(p)\) )
then \(\left(\int x .|f x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p)\)
else ( \(\infty::\) ennreal \()\) )
\(\operatorname{defect}(\mathfrak{L} p M)=2 \operatorname{powr}(1 / p-1)\)
proof -
define \(F\) where \(F=\{f \in\) borel-measurable \(M\). integrable \(M(\lambda x .|f x|\) powr \(p)\}\)
```

    have \(*\) : quasinorm-on \(F\) (2 powr ( \(1 / p-1\) ) ) \(\left(\lambda\left(f::^{\prime} a \Rightarrow\right.\right.\) real \() .\left(\int x .|f x|\right.\) powr \(p\)
    ```
\(\partial M)\) powr \((1 / p))\)
    proof (rule quasinorm-onI)
    fix \(f g\) assume \(f \in F g \in F\)
    then show \(f+g \in F\)
    unfolding F-def plus-fun-def apply (auto) by (rule Minkowski-inequality-le-1 (1),
auto simp add: \(\langle p>0\rangle\langle p \leq 1\rangle)\)
    show ennreal \(\left(\left(\int x .|(f+g) x|\right.\right.\) powr \(\left.p \partial M\right)\) powr \(\left.(1 / p)\right)\)
            \(\leq\) ennreal \((2\) powr \((1 / p-1)) *\left(\int x .|f x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p)+\) ennreal
(2 powr \((1 / p-1)) *\left(\int x .|g x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p)\)
            apply (subst ennreal-mult [symmetric], auto)+
            apply (subst ennreal-plus[symmetric], simp, simp)
            apply (rule ennreal-leI)
            unfolding plus-fun-def apply (rule Minkowski-inequality-le-1(2)[of pf Mg ],
auto simp add: \(\langle p>0\rangle\langle p \leq 1\rangle\) )
            using \(\langle f \in F\rangle\langle g \in F\rangle\) unfolding \(F\)-def by auto
    next
    fix \(f\) and \(c:\) :real assume \(f \in F\)
        show \(c *_{R} f \in F\) using \(\langle f \in F\rangle\) unfolding scaleR-fun-def F-def by (auto
simp add: abs-mult powr-mult)
    show \(\left(\int x .\left|\left(c *_{R} f\right) x\right|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p) \leq \operatorname{ennreal}(\operatorname{abs}(c)) *\left(\int x .|f x|\right.\)
powr \(p \partial M\) ) powr ( \(1 / p\) )
        apply (rule eq-refl, subst ennreal-mult[symmetric], simp, simp, rule en-
nreal-cong)
    apply (unfold scaleR-fun-def, simp add: abs-mult powr-mult powr-powr) using
\(\langle p>0\rangle\) by auto
    next
        show \(0 \in F\) unfolding zero-fun-def \(F\)-def by auto
        show \(1 \leq 2\) powr \((1 / p-1)\) using \(\langle p>0\rangle\langle p \leq 1\rangle\) by (auto simp add:
ge-one-powr-ge-zero)
    qed (auto)
    have \(p \geq 0\) using \(\langle p>0\rangle\) by auto
    have \(* *: \mathfrak{L} p M=\) quasinorm-of (2 powr ( \(1 / p-1\) ),
                            \((\lambda f\). if \((f \in\) borel-measurable \(M \wedge\) integrable \(M(\lambda x .|f x|\) powr \(p))\)
                then ( \(\int x .|f x|\) powr \(p \partial M\) ) powr ( \(1 / p\) )
                else ( \(\infty::\) ennreal \()\) )
    unfolding Lp-space-def using \(\langle p>0\rangle\langle p \leq 1\rangle\) using enn2real-ennreal[ \(O F\langle p\)
\(\geq 0\rangle\) ] apply auto
    by (insert enn2real-ennreal \([O F\langle p \geq 0\rangle]\), presburger) +
    show eNorm \((\mathfrak{L} p M) f=(\) if \((f \in\) borel-measurable \(M \wedge\) integrable \(M(\lambda x .|f x|\)
powr \(p\) ))
```

                                    then \(\left(\int x .|f x|\right.\) powr \(\left.p \partial M\right)\) powr \((1 / p)\)
                                    else ( \(\infty::\) ennreal \()\) )
    ```
    \(\operatorname{defect}(\mathfrak{L} p M)=2\) powr \((1 / p-1)\)
    unfolding \(* *\) using quasinorm-of[OF extend-quasinorm \([O F *]\) ] unfolding
\(F\)-def by auto
qed
```

lemma Lp-space:
assumes }p>(0::real
shows space}\mp@subsup{N}{N}{}(\mathfrak{L}pM)={f\in\mathrm{ borel-measurable M. integrable M ( }\lambdax.|fx|\mathrm{ powr
p)}
apply (auto simp add: spaceN-iff)
using Lp(1) Lp-le-1(1) <p> 0\rangle apply (metis infinity-ennreal-def less-le not-less)
using Lp(1) Lp-le-1(1)\langlep>0\rangle apply (metis infinity-ennreal-def less-le not-less)
using Lp(1) Lp-le-1(1)\langlep>0\rangle by (metis ennreal-neq-top linear top.not-eq-extremum)
lemma Lp-I:
assumes p>(0::real)
f\in borel-measurable M integrable M ( }\lambdax.|fx| powr p
shows f\in\mp@subsup{\mathrm{ space}}{N}{}(\mathfrak{L}pM)
Norm (\mathscr{L}pM)f=(\intx.|fx| powr p \partialM) powr (1/p)
eNorm (LL p M) f=( ( x. |fx| powr p \partialM) powr (1/p)
proof -
have *: eNorm ({ p M) f=( { x. |f x| powr p \partialM) powr (1/p)
by (cases p\leq1, insert assms, auto simp add: Lp-le-1(1) Lp(1))
then show **: f\in\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}pM)\mathrm{ unfolding space }
show Norm ({L p M) f=( {x. |f x | powr p \partialM) powr (1/p) using * unfolding
Norm-def by auto
then show eNorm ({ p M) f=( (\intx. |f x| powr p \partialM) powr (1/p) using
eNorm-Norm[OF **] by auto
qed
lemma Lp-D:
assumes p>0f\in \mp@subsup{\operatorname{sace}}{N}{}(\mathcal{L}pM)
shows f}\in\mathrm{ borel-measurable M
integrable M ( }\lambdax.|fx| powr p
Norm ({ p M) f=(\intx. |f x| powr p \partialM) powr (1/p)
eNorm({) p M) f=(\intx. |fx| powr p \partialM) powr (1/p)
proof -
show *: f\in borel-measurable M
integrable M ( }\lambdax.|fx| powr p
using Lp-space[OF <p> 0\rangle] assms(2) by auto
then show Norm ({) pM)f=(\intx. |f x| powr p \partialM) powr (1/p)
eNorm ({ p M) f=(\intx. |fx| powr p \partialM) powr (1/p)
using Lp-I[OF<p>0`] by auto
qed
lemma Lp-Norm:
assumes p>(0::real)
f\in borel-measurable M
shows Norm ({ p M) f=(\intx. |f x| powr p \partialM) powr (1/p)
(Norm (\mathfrak{L p M) f) powr p = ( f x. |f x| powr p \partialM)})
proof -
show *: Norm (L p M) f=(\intx. |f x| powr p \partialM) powr (1/p)
proof (cases integrable M (\lambdax. |f x| powr p))
case True

```
```

    then show ?thesis using Lp-I[OF assms True] by auto
    next
    case False
    then have f}\not\in\mp@subsup{\mathrm{ space N}}{N}{}(\mathfrak{L}pM)\mathrm{ using Lp-space[OF}\langlep>0\rangle, of M] by aut
    then have *: Norm ( }\mathfrak{L}pM)f=0\mathrm{ using eNorm-Norm' by auto
    have (\intx.|fx| powr p \partialM) = 0 using False by (simp add: not-integrable-integral-eq)
    then have ( }\intx.|fx| powr p \partialM) powr (1/p)=0 by aut
    then show ?thesis using * by auto
    qed
    then show (Norm ({ p M) f) powr p = (\intx. |f x | powr p \partialM)
    unfolding * using powr-powr }\langlep>0\rangle\mathrm{ by auto
    qed
lemma Lp-zero-space:
assumes p> (0::real)
shows zero-space N}(\mathfrak{L}pM)={f\in\mathrm{ borel-measurable M. AE x in M.fx=0}
proof (auto)
fix f assume H:f\in\mp@subsup{zero-space}{N}{}(\mathfrak{L}pM)
then have *: f\in{f\in borel-measurable M. integrable M (\lambdax. |fx| powr p)}
using Lp-space[OF assms] zero-spaceN-subset-spaceN by auto
then show }f\in\mathrm{ borel-measurable M by auto
have eNorm ({ p M) f=( \intx. |f x| powr p \partialM) powr (1/p)
by (cases p\leq1, insert * <p> 0\rangle, auto simp add: Lp-le-1(1) Lp(1))
then have ( }\intx.|fx| powr p \partialM)=0 using H unfolding zero-space N-def by
auto
then have AE x in M. |f x powr p=0
by (subst integral-nonneg-eq-0-iff-AE[symmetric], insert *, auto)
then show AE x in M.fx=0 by auto
next
fix f::'a m real
assume H [measurable]: f\in borel-measurable M AE x in M. fx=0
then have *:AEx in M. |f x powr p=0 by auto
have integrable M ( }\lambdax.|fx| powr p
using integrable-cong-AE[OF - *] by auto
have **: (\int x. |f x| powr p \partialM)=0
using integral-cong-AE[OF-*] by auto
have eNorm ({ p M) f=( ( x. |f x| powr p \partialM) powr (1/p)
by (cases p\leq1, insert H(1)<integrable M (\lambdax. |f x| powr p)\rangle\langlep>0\rangle, auto
simp add: Lp-le-1(1) Lp(1))
then have eNorm ({ L PM)f=0 using ** by simp
then show }f\in\mp@subsup{\mathrm{ zero-spaceN}}{N}{}(\mathfrak{L}pM
using zero-spaceN-iff by auto
qed
lemma Lp-tendsto-AE-subseq:
assumes p>(0::real)
tendsto-in}\mp@subsup{N}{N}{}(\mathfrak{L}pM)f
\n.fn \in space}N({~LpM
g\in space}N({\mathfrak{L}pM

```
shows \(\exists r\). strict-mono \(r \wedge(A E x\) in \(M .(\lambda n . f(r n) x) \longrightarrow g x)\) proof -
have \(f n-g \in \operatorname{space}_{N}(\mathfrak{L} p M)\) for \(n\)
using spaceN-diff \(\left[O F\left\langle\bigwedge n\right.\right.\).f \(\left.\left.n \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\left\langle g \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\right]\) by simp
have int: integrable \(M(\lambda x\). \(|f n x-g x|\) powr \(p)\) for \(n\)
using \(L p-D(2)\left[O F\langle p>0\rangle\left\langle f n-g \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\right]\) by auto
have \((\lambda n\). Norm \((\mathfrak{L} p M)(f n-g)) \longrightarrow 0\) using \(\left\langle\right.\) tendsto-in \(\left.n_{N}(\mathfrak{L} p M) f g\right\rangle\) unfolding tendsto-in \(n_{N}\)-def by auto
then have \(*:\left(\lambda n\right.\). \(\left(\int x .|f n x-g x|\right.\) powr \(\left.p \partial M\right)\) powr \(\left.(1 / p)\right) \longrightarrow 0\)
using \(L p-D(3)\left[O F\langle p>0\rangle\left\langle\bigwedge n\right.\right.\). \(f n-g \in\) space \(\left.\left._{N}(\mathfrak{L} p M)\right\rangle\right]\) by auto
have \(\left(\lambda n\right.\). \(\left(\left(\int x .|f n x-g x|\right.\right.\) powr \(\left.p \partial M\right)\) powr \(\left.(1 / p)\right)\) powr \(\left.p\right) \longrightarrow 0\)
apply (rule tendsto-zero-powrI \([\) of - - p]) using \(\langle p>0\rangle *\) by auto
then have \(* *:\left(\lambda n .\left(\int x .|f n x-g x|\right.\right.\) powr \(\left.\left.p \partial M\right)\right) \longrightarrow 0\) using powr-powr \(\langle p>0\rangle\) by auto
have \(\exists r\). strict-mono \(r \wedge(A E x\) in \(M .(\lambda n .|f(r n) x-g x|\) powr \(p) \longrightarrow 0)\) apply (rule tendsto-L1-AE-subseq) using int \(* *\) by auto
then obtain \(r\) where strict-mono \(r A E x\) in \(M .(\lambda n .|f(r n) x-g x|\) powr \(p)\)
\(\qquad\)
by blast
moreover have \((\lambda n . f(r n) x) \longrightarrow g x\) if \((\lambda n .|f(r n) x-g x|\) powr \(p)\)
\(\qquad\)
proof -
have \((\lambda n .(|f(r n) x-g x|\) powr \(p)\) powr \((1 / p)) \longrightarrow 0\)
apply (rule tendsto-zero-powr \(I[\) of \(--1 / p]\) ) using \(\langle p>0\rangle\) that by auto
then have \((\lambda n .|f(r n) x-g x|) \longrightarrow 0\)
using powr-powr \(\langle p>0\rangle\) by auto
show ?thesis
by (simp add: \(\langle(\lambda n .|f(r n) x-g x|) \longrightarrow 0\rangle\) Limits.LIM-zero-cancel tendsto-rabs-zero-cancel)
qed
ultimately have \(A E x\) in \(M .(\lambda n . f(r n) x) \longrightarrow g x\) by auto
then show ?thesis using «strict-mono \(r\) 〉 by auto
qed

\subsection*{5.3 Specialization to \(L^{1}\)}

\section*{lemma L1-space:}
space \(_{N}(\mathfrak{L} 1 M)=\{f\). integrable \(M f\}\)
unfolding one-ereal-def using Lp-space[of 1 M\(]\) integrable-abs-iff by auto
lemma L1-I:
assumes integrable \(M f\)
shows \(f \in\) space \(_{N}(\mathfrak{L} 1 M)\)
\(\operatorname{Norm}(\mathfrak{L} 1 M) f=\left(\int x .|f x| \partial M\right)\)
\(\operatorname{eNorm}(\mathfrak{L} 1 M) f=\left(\int x .|f x| \partial M\right)\)
unfolding one-ereal-def using Lp-I[of 1, OF - borel-measurable-integrable[OF assms]] assms powr-to-1 by auto
```

lemma L1-D:
assumes f\in \mp@subsup{\operatorname{sace}}{N}{}(\mathfrak{L}1M)
shows f\in borel-measurable M
integrable Mf
Norm ({ 1 M) f=( { x. |f x| \partialM)
eNorm (L 1 M) f=( \intx. |fx|\partialM)
using assms by (auto simp add: L1-space L1-I)
lemma L1-int-ineq:
abs(\intx.fx\partialM)\leqNorm ({) 1M)f
proof (cases integrable M f)
case True
then show ?thesis using L1-I(2)[OF True] by auto
next
case False
then have (\intx.fx\partialM)=0 by (simp add: not-integrable-integral-eq)
then show ?thesis using Norm-nonneg by auto
qed

```

In \(L^{1}\), one can give a direct formula for the eNorm of a measurable function, using a nonnegative integral. The same formula holds in \(L^{p}\) for \(p>0\), with additional powers \(p\) and \(1 / p\), but one can not write it down since powr is not defined on ennreal.
```

lemma L1-Norm:
assumes [measurable]: $f \in$ borel-measurable $M$
shows $\operatorname{Norm}(\mathfrak{L} 1 M) f=\left(\int x .|f x| \partial M\right)$
$\operatorname{eNorm}(\mathfrak{L} 1 M) f=\left(\int{ }^{+} x .|f x| \partial M\right)$
proof -
show *: $\operatorname{Norm}(\mathfrak{L} 1 M) f=\left(\int x .|f x| \partial M\right)$
using Lp-Norm[of 1, OF - assms] unfolding one-ereal-def by auto
show eNorm $(\mathfrak{L} 1 M) f=\left(\int{ }^{+} x .|f x| \partial M\right)$
proof (cases integrable $M f$ )
case True
then have $f \in \operatorname{space}_{N}(\mathfrak{L} 1 M)$ using L1-space by auto
then have eNorm $(\mathfrak{L} 1 M) f=\operatorname{ennreal}(\operatorname{Norm}(\mathfrak{L} 1 M) f)$
using eNorm-Norm by auto
then show ?thesis
by (metis (mono-tags) * AE-I2 True abs-ge-zero integrable-abs nn-integral-eq-integral)
next
case False
then have eNorm ( $\mathfrak{L} 1 M) f=\infty$ using L1-space space ${ }_{N}$-def
by (metis ennreal-add-eq-top infinity-ennreal-def le-iff-add le-less-linear mem-Collect-eq)
moreover have $\left(\int^{+} x .|f x| \partial M\right)=\infty$
apply (rule nn-integral-nonneg-infinite) using False by (auto simp add:
integrable-abs-iff)
ultimately show ?thesis by simp
qed
qed

```

\section*{lemma L1-indicator:}
assumes [measurable]: \(A \in\) sets \(M\)
shows \(e \operatorname{Norm}(\mathfrak{L} 1 M)(\) indicator \(A)=\) emeasure \(M A\)
by (subst L1-Norm, auto, metis assms ennreal-indicator nn-integral-cong nn-integral-indicator)
lemma L1-indicator': assumes [measurable]: \(A \in\) sets \(M\)
and emeasure \(M A \neq \infty\)
shows indicator \(A \in\) space \(_{N}(\mathfrak{L} 1 M)\)
\(\operatorname{Norm}(\mathfrak{L} 1 M)(\) indicator \(A)=\) measure \(M A\)
unfolding space \(_{N}\)-def Norm-def using L1-indicator[OF \(\langle A \in\) sets \(M\rangle\) ] 〈emeasure \(M A \neq \infty\) >
by (auto simp add: top.not-eq-extremum Sigma-Algebra.measure-def)

\section*{\(5.4 L^{0}\)}

We have defined \(L^{p}\) for all exponents \(p\), although it does not really make sense for \(p=0\). We have chosen a definition in this case (the space of all measurable functions) so that many statements are true for all exponents. In this paragraph, we show the consistency of this definition.
```

lemma L-zero:
eNorm ( {L 0M) f=(if f\inborel-measurable M then 0 else }\infty
defect ({L OM)=1
proof -
have *: quasinorm-on UNIV 1 ( }\lambda(f::''a=>\mathrm{ real). (if }f\in\mathrm{ borel-measurable }M\mathrm{ then 0
else \infty))
by (rule extend-quasinorm, rule quasinorm-onI, auto)
show eNorm ({) OM) f=(if f\inborel-measurable M then 0 else }\infty\mathrm{ )
defect ({) OM)=1
using quasinorm-of[OF *] unfolding Lp-space-def by auto
qed
lemma L-zero-space [simp]:
space}\mp@subsup{N}{}{\prime}(\mathfrak{L}0M)=\mathrm{ borel-measurable M
zero-space }\mp@subsup{N}{N}{}(\mathfrak{L}0M)=\mathrm{ borel-measurable M
apply (auto simp add: spaceN-iff zero-spaceN-iff L-zero(1))
using top.not-eq-extremum by force+

```

\subsection*{5.5 Basic results on \(L^{p}\) for general \(p\)}
```

lemma Lp-measurable-subset:

```
lemma Lp-measurable-subset:
    space}N({~~M)\subseteq\mathrm{ borel-measurable M
    space}N({~~M)\subseteq\mathrm{ borel-measurable M
proof (cases rule: Lp-cases[of p])
proof (cases rule: Lp-cases[of p])
    case zero
    case zero
    then show ?thesis using L-zero-space by auto
    then show ?thesis using L-zero-space by auto
next
next
    case (real-pos p2)
```

    case (real-pos p2)
    ```
```

    then show ?thesis using Lp-space[OF<p2>0\rangle] by auto
    next
case PInf
then show ?thesis using L-infinity-space by auto
qed
lemma Lp-measurable:
assumes f\in \mp@subsup{\operatorname{sace}}{N}{}(\mathfrak{L}pM)
shows f}\in\mathrm{ borel-measurable M
using assms Lp-measurable-subset by auto
lemma Lp-infinity-zero-space:
assumes p>(0::ennreal)
shows zero-space
proof (cases rule: Lp-cases[of p])
case PInf
then show ?thesis using L-infinity-zero-space by auto
next
case (real-pos p2)
then show ?thesis using Lp-zero-space[OF <p2>0\rangle] unfolding < p = ennreal
p2> by auto
next
case zero
then have False using assms by auto
then show ?thesis by simp
qed
lemma (in prob-space) Lp-subset-Lq:
assumes p\leqq
shows }\f\mathrm{ .eNorm ({)
L qM \subseteq
\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}qM)\subseteq\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}pM)
\f.f\in\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}qM)\Longrightarrow\operatorname{Norm}(\mathfrak{L}pM)f\leqNorm({L qM)f
proof -
show eNorm ({L pM)f\leqeNorm (\mathfrak{L}qM)f for f
proof (cases eNorm (\mathcal{L q M) f<\infty)}
case True
then have f}\in\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}qM)\mathrm{ using spaceN-iff by auto
then have f-meas [measurable]: f}\in\mathrm{ borel-measurable M using Lp-measurable
by auto
consider p=0 | p=q| |>0^p<\infty\wedgeq=\infty| p>0\wedge p<q\wedgeq<\infty
using <p\leqq\rangle apply (simp add: top.not-eq-extremum)
using not-less-iff-gr-or-eq order.order-iff-strict by fastforce
then show ?thesis
proof (cases)
case 1
then show ?thesis by (simp add: L-zero(1))
next
case 2

```
```

    then show ?thesis by auto
    next
    case 3
    then have q=\infty by simp
    obtain p2 where p=ennreal p2 p2 > 0
        using 3 enn2real-positive-iff[of p] by (cases p) auto
    have *: AE x in M. |fx|\leqNorm (L }\inftyM)
        using L-infinity-AE-bound}\langlef\in\mp@subsup{\mathrm{ space }}{N}{}(\mathfrak{L}qM)\rangle\langleq=\infty> by aut
    have **: integrable M ( }\lambdax.|fx|\mathrm{ powr p2)
        apply (rule Bochner-Integration.integrable-bound[of - \lambdax. (Norm ( }\mathfrak{L}\inftyM
    f) powr p2], auto)
using * powr-mono2 <p2 > 0> by force
then have eNorm ({ p2 M) f=(\intx. |fx| powr p2 \partialM) powr (1/p2)
using Lp-I(3)[OF <p2 > 0`f-meas] by simp     also have \ldots\leq (\intx.(Norm (L L M) f) powr p2 \partialM) powr (1/p2)         apply (rule ennreal-leI, rule powr-mono2, simp add: <p2 > 0\rangle less-imp-le, simp)         apply (rule integral-mono-AE, auto simp add:**)         using * powr-mono2 <p2 > 0> by force     also have ... = Norm (L }\inftyM)         using <p2 > 0` by (auto simp add: prob-space powr-powr)
finally show ?thesis
using <p = ennreal p2\rangle\langleq=\infty> eNorm-Norm[OF}\langlef\in\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}qM)\rangle
by auto
next
case }
then have 0<pp<\infty by auto
then obtain p2 where p= ennreal p2 p2 >0
using enn2real-positive-iff[of p] by (cases p) auto
have 0<qq<\infty using 4 by auto
then obtain q2 where q=ennreal q2 q2 >0
using enn2real-positive-iff[of q] by (cases q) auto
have p2 < q2 using 4 <p = ennreal p2` \langleq= ennreal q2`
using ennreal-less-iff by auto
define r2 where r2 = q2 / p2
have r2 \geq1 unfolding r2-def using <p2 < q2> <p2 > 0> by auto
have *: abs (|z| powr p2) powr r2 = |z| powr q2 for z::real
unfolding r2-def using <p2 > 0` by (simp add: powr-powr)     have I: integrable M ( }\lambdax\mathrm{ . abs( |f x| powr p2) powr r2)         unfolding * using <f\in \mp@subsup{\mathrm{ space }}{N}{}(\mathfrak{L}qM)\rangle\langleq= ennreal q2` Lp-D(\mathscr{)}[OF
<q2> 0\rangle] by auto
have J: integrable M ( }\lambdax.|fx| powr p2
by (rule bound-L1-Lp(1)[OF<r2 \geq1>-I],auto)
have f}\in\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}p\mathcal{Z}M
by (rule Lp-I(1)[OF <p2>0\rangle-J], simp)
have (\intx. |f x| powr p2 \partialM) powr (1/p2) =abs(\intx. |f x| powr p2 \partialM) powr
(1/p2)
by auto
also have ... \leq((\int x. abs (|f x| powr p2) powr r2 \partialM) powr (1/r2)) powr

```
```

(1/p2)
apply (subst powr-mono2, simp add: <p2 > 0\rangle less-imp-le, simp)
apply (rule bound-L1-Lp, simp add: <r2 \geq 1\rangle, simp)
unfolding * using <f\in space
<q2 > 0`] by auto     also have ... = (\int x. |f x| powr q2 \partialM) powr (1/q\mathcal{Z}         unfolding * using <p2 > 0> by (simp add: powr-powr r2-def)     finally show ?thesis         using <f \in space}\mp@subsup{N}{N}{}(\mathfrak{L}qM)\rangleLp-D(4)[OF<q2 > 0`] ennreal-leI
unfolding <p = ennreal p2\rangle\langleq= ennreal q2\rangle Lp-D(4)[OF<p2>0\rangle\langlef\in
space}NN({~p2 M)>] by forc
qed
next
case False
then have eNorm ( L q M)f=\infty
using top.not-eq-extremum by fastforce
then show ?thesis by auto
qed
then show }\mathfrak{L}qM\mp@subsup{\subseteq}{N}{}\mathfrak{L}pM\mathrm{ using quasinorm-subsetI[of - - 1] by auto
then show space}\mp@subsup{N}{N}{}(\mathfrak{L}qM)\subseteq\mp@subsup{\mathrm{ space }}{N}{}(\mathfrak{L}pM)\mathrm{ using quasinorm-subset-space
by auto
then show Norm ({~pM)f\leqNorm (\mathfrak{L qM)f if f\in space}N({~
using eNorm-Norm that <eNorm ( }\mathfrak{L}pM)f\leqeNorm ({LqM)f> ennreal-le-iff
Norm-nonneg
by (metis rev-subsetD)
qed
proposition Lp-domination:
assumes [measurable]: g borel-measurable M
and f\in space}N({~LPM
AE x in M. |gx | \leq |f x
shows g\in space}N({) (L M
Norm ({ p M)g\leqNorm ({ p M )f
proof -
have [measurable]: f\in borel-measurable M using Lp-measurable[OF assms(2)]
by simp
have g\in space}N(\mathfrak{L}pM)\wedgeNorm ({~pM)g\leqNorm ({~ pM)
proof (cases rule: Lp-cases[of p])
case zero
then have Norm ({L pM)g=0
unfolding Norm-def using L-zero(1)[of M] by auto
then have Norm ( L p M)g\leqNorm ({L pM)f using Norm-nonneg by auto
then show ?thesis unfolding }\langlep=0\rangleL\mathrm{ -zero-space by auto
next
case (real-pos p2)
have *: integrable M ( }\lambdax.|fx| powr p2
using}\langlef\in\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}pM)\rangle\mathrm{ unfolding <p= ennreal p2` using Lp-D(2)<p2
>0> by auto
have **: integrable M ( }\lambdax.|g\mathrm{ x| powr p2)

```
apply (rule Bochner-Integration.integrable-bound[of - \(\lambda x .|f x|\) powr p2]) using * apply auto
using assms(3) powr-mono2 \(\langle p 2>0\rangle\) by (auto simp add: less-imp-le)
then have \(g \in \operatorname{space}_{N}(\mathfrak{L} p M)\)
unfolding \(\langle p=\) ennreal \(p 2\rangle\) using \(L p\)-space \([O F\langle p 2>0\rangle\), of \(M]\) by auto
have \(\operatorname{Norm}(\mathfrak{L} p M) g=\left(\int x .|g x|\right.\) powr p2 \(\left.\partial M\right)\) powr (1/p2)
unfolding \(\langle p=\) ennreal \(p 2\rangle\) by (rule \(L p-I(2)[O F\langle p 2>0\rangle-* *]\), simp)
also have \(\ldots \leq\left(\int x .|f x|\right.\) powr p2 \(\left.\partial M\right)\) powr (1/p2)
apply (rule powr-mono2, simp add: \(\langle p 2>0\rangle\) less-imp-le, simp)
apply (rule integral-mono-AE, auto simp add: \(* * *\) )
using \(\langle p 2>0\rangle\) less-imp-le powr-mono2 assms(3) by auto
also have \(\ldots=\operatorname{Norm}(\mathfrak{L} p M) f\)
 simp)
finally show ?thesis using \(\left\langle g \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\) by auto
next
case PInf
have \(A E x\) in \(M .|f x| \leq \operatorname{Norm}(\mathfrak{L} p M) f\)
using \(\left\langle f \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle L\)-infinity-AE-bound unfolding \(\langle p=\infty\rangle\) by auto
then have \(*: A E x\) in \(M .|g x| \leq \operatorname{Norm}(\mathfrak{L} p M) f\)
using assms(3) by auto
show ?thesis
using L-infinity-I[OF assms(1)*] Norm-nonneg \(\langle p=\infty\) by auto
qed
then show \(g \in\) space \(_{N}(\mathfrak{L} p M) \operatorname{Norm}(\mathfrak{L} p M) g \leq \operatorname{Norm}(\mathfrak{L} p M) f\)
by auto
qed
lemma Lp-Banach-lattice:
assumes \(f \in\) space \(_{N}(\mathfrak{L} p M)\)
shows \((\lambda x .|f x|) \in \operatorname{space}_{N}(\mathfrak{L} p M)\)
\(\operatorname{Norm}(\mathfrak{L} p M)(\lambda x .|f x|)=\operatorname{Norm}(\mathfrak{L} p M) f\)
proof -
have [measurable]: \(f \in\) borel-measurable \(M\) using Lp-measurable[OF assms] by simp
show \((\lambda x .|f x|) \in \operatorname{space}_{N}(\mathfrak{L} p M)\)
by (rule Lp-domination (1)[OF - assms], auto)
have \(\operatorname{Norm}(\mathfrak{L} p M)(\lambda x .|f x|) \leq \operatorname{Norm}(\mathfrak{L} p M) f\)
by (rule Lp-domination[OF - assms], auto)
moreover have \(\operatorname{Norm}(\mathfrak{L} p M) f \leq \operatorname{Norm}(\mathfrak{L} p M)(\lambda x .|f x|)\)
by (rule Lp-domination \(\left[O F-\left\langle(\lambda x .|f x|) \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\right]\), auto)
finally show \(\operatorname{Norm}(\mathfrak{L} p M)(\lambda x .|f x|)=\operatorname{Norm}(\mathfrak{L} p M) f\) by auto
qed
lemma Lp-bounded-bounded-support:
assumes [measurable]: \(f \in\) borel-measurable \(M\)
and \(A E x\) in \(M .|f x| \leq C\)
emeasure \(M\{x \in\) space \(M . f x \neq 0\} \neq \infty\)
```

    shows f\in space}N(\mathfrak{L}pM
    proof (cases rule: Lp-cases[of p])
case zero
then show ?thesis using L-zero-space assms by blast
next
case PInf
then show ?thesis using L-infinity-space assms by blast
next
case (real-pos p2)
have *: integrable M ( }\lambdax.|fx| powr p2)
apply (rule integrableI-bounded-set[of {x\in space M. fx\not=0} - C powr p2])
using assms powr-mono2[OF less-imp-le[OF<p2 > 0〉]] by (auto simp add:
top.not-eq-extremum)
show ?thesis
unfolding <p = ennreal p2` apply (rule Lp-I[OF<p2 > 0`]) using * by auto
qed

```

\section*{5．6 \(L^{p}\) versions of the main theorems in integration theory}

The space \(L^{p}\) is stable under almost sure convergence，for sequence with bounded norm．This is a version of Fatou＇s lemma（and it indeed follows from this lemma in the only nontrivial situation where \(p \in(0,+\infty)\) ．
proposition Lp－AE－limit：
assumes［measurable］：\(g \in\) borel－measurable \(M\)
and \(A E x\) in \(M\) ．\((\lambda n . f n x) \longrightarrow g x\)
shows \(e \operatorname{Norm}(\mathfrak{L} p M) g \leq \liminf (\lambda n . \operatorname{eNorm}(\mathfrak{L} p M)(f n))\)
proof \((\) cases liminf \((\lambda n\) ．eNorm \((\mathfrak{L} p M)(f n))=\infty)\)
case True
then show ？thesis by auto
next
case False
define \(l e\) where \(l e=\liminf (\lambda n . \operatorname{eNorm}(\mathfrak{L} p M)(f n))\)
then have le \(<\infty\) using False by（simp add：top．not－eq－extremum）
obtain \(r 0\) where \(r 0\) ：strict－mono r0 \((\lambda n\) ．eNorm \((\mathfrak{L} p M)(f(r 0 n))) \longrightarrow l e\)
using liminf－subseq－lim［of \(\lambda n\) ．eNorm（ \(\mathfrak{L} p M)(f n)]\)
unfolding comp－def le－def
by blast
then have eventually \((\lambda n\) ．eNorm \((\mathfrak{L} p M)(f(r 0 n))<\infty)\) sequentially using False unfolding order－tendsto－iff le－def by（simp add：top．not－eq－extremum）
then obtain \(N\) where \(N: \wedge n . n \geq N \Longrightarrow \operatorname{eNorm}(\mathfrak{L} p M)(f(r 0 n))<\infty\)
unfolding eventually－sequentially by blast
define \(r\) where \(r=(\lambda n . r 0(n+N))\)
have strict－mono \(r\) unfolding \(r\)－def using 〈strict－mono r0〉
by（simp add：strict－mono－Suc－iff）
have \(*:(\lambda n . e \operatorname{Norm}(\mathfrak{L} p M)(f(r n))) \longrightarrow l e\)
unfolding \(r\)－def using LIMSEQ－ignore－initial－segment［OF r0（2），of \(N]\) ．
have \(f(r n) \in \operatorname{space}_{N}(\mathfrak{L} p M)\) for \(n\)
using \(N\) space \(N\)－iff unfolding \(r\)－def by force
then have［measurable］：\(f(r n) \in\) borel－measurable \(M\) for \(n\)
using Lp-measurable by auto
define \(l\) where \(l=\) enn2real le
have \(l \geq 0\) unfolding \(l\)-def by auto
have \(l e=\) ennreal \(l\) using \(\langle l e<\infty\rangle\) unfolding \(l\)-def by auto
have [tendsto-intros]: \((\lambda n . \operatorname{Norm}(\mathfrak{L} p M)(f(r n))) \longrightarrow l\)
apply (rule tendsto-ennrealD)
using \(*\langle l e<\infty\rangle\) unfolding eNorm-Norm[OF \(\left\langle\bigwedge n . f(r n) \in\right.\) space \(_{N}(\mathfrak{L} p\) M) >] l-def by auto
show ?thesis
proof (cases rule: Lp-cases[of p])
case zero
then have \(e \operatorname{Norm}(\mathfrak{L} p M) g=0\)
using assms(1) by (simp add: L-zero(1))
then show ?thesis by auto
next
case (real-pos p2)
then have \(f(r n) \in \operatorname{space}_{N}(\mathfrak{L} p \mathcal{Z} M)\) for \(n\)
using \(\left\langle\bigwedge n . f(r n) \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\) by auto
have liminf \((\lambda n\). ennreal \((|f(r n) x|\) powr p2 \())=|g x|\) powr p2 if \((\lambda n . f n x)\) \(g x\) for \(x\)
apply (rule lim-imp-Liminf, auto intro!: tendsto-intros simp add: \(\langle p 2>0\rangle\) )
using LIMSEQ-subseq-LIMSEQ[OF that 〈strict-mono r〉] unfolding comp-def by auto
then have \(*: A E x\) in \(M . \liminf (\lambda n\). ennreal \((|f(r n) x|\) powr \(p 2))=|g x|\) powr \({ }^{2} 2\)
using \(\langle A E x\) in \(M .(\lambda n . f n x) \longrightarrow g x\rangle\) by auto
have \(\left(\int{ }^{+}\right.\)x. ennreal \((|f(r n) x|\) powr \(\left.p \mathcal{Z}) \partial M\right)=\operatorname{ennreal}((\operatorname{Norm}(\mathfrak{L} p M)(f(r\) \(n)\) )) powr \(p\) 2) for \(n\)
proof -
have \(\left(\int{ }^{+} x\right.\). ennreal \((|f(r n) x|\) powr p2 \(\left.) \partial M\right)=\) ennreal \(\left(\int x .|f(r n) x|\right.\) powr p2 \(\partial M\) )
by (rule nn-integral-eq-integral, auto simp add: Lp-D(2)[OF \(\langle p 2>0\rangle\langle f(r\) \(\left.\left.\left.n) \in \operatorname{space}_{N}(\mathfrak{L} p 2 M)\right\rangle\right]\right)\)
also have \(\ldots=\operatorname{ennreal}((\operatorname{Norm}(\mathfrak{L} p 2 M)(f(r n)))\) powr p2 \()\)
unfolding \(L p-D(3)\left[O F\langle p 2>0\rangle\left\langle f(r n) \in \operatorname{space}_{N}(\mathfrak{L} p 2 M)\right\rangle\right]\) using powr-powr \(\langle p 2>0\rangle\) by auto
finally show ?thesis using \(\langle p=\) ennreal \(p 2\rangle\) by simp
qed
moreover have \((\lambda n\). ennreal \(((\operatorname{Norm}(\mathfrak{L} p M)(f(r n)))\) powr \(p \mathcal{Z})) \longrightarrow\) ennreal(l powr p2)
by (auto intro!:tendsto-intros simp add: \(\langle p 2>0\rangle\) )
ultimately have \(* *: \liminf \left(\lambda n .\left(\int^{+} x\right.\right.\). ennreal \((|f(r n) x|\) powr p2 \(\left.\left.) \partial M\right)\right)=\) ennreal(l powr p2)
using lim-imp-Liminf by force
have \(\left(\int{ }^{+} x .|g x|\right.\) powr p2 \(\left.\partial M\right)=\left(\int{ }^{+} x\right.\). liminf \((\lambda n\). ennreal \((|f(r n) x|\) powr p2)) \(\partial M\) )
apply (rule nn-integral-cong-AE) using * by auto
also have \(\ldots \leq \liminf \left(\lambda n . \int{ }^{+}\right.\)x. ennreal \((|f(r n) x|\) powr \(p\) 2) \(\partial M)\)
by (rule nn-integral-liminf, auto)
finally have \(\left(\int{ }^{+} x .|g x|\right.\) powr \(\left.p 2 \partial M\right) \leq\) ennreal (l powr p2) using \(* *\) by auto
then have \(\left(\int^{+}{ }^{+} x .|g x|\right.\) powr p2 \(\left.\partial M\right)<\infty\) using le-less-trans by fastforce
then have intg: integrable \(M(\lambda x .|g x|\) powr \(p 2)\)
apply (intro integrableI-nonneg) by auto
then have \(g \in \operatorname{space}_{N}(\mathfrak{L} p 2 M)\) using \(L p-I(1)[O F\langle p 2>0\rangle\), of \(-M]\) by fastforce
have ennreal \(\left(\left(\operatorname{Norm}(\mathfrak{L}\right.\right.\) p2 M) g) powr p2 \()=\operatorname{ennreal}\left(\int x .|g x|\right.\) powr p2 \(\left.\partial M\right)\) unfolding \(L p-D(3)\left[O F\langle p 2>0\rangle\left\langle g \in\right.\right.\) space \(\left.\left._{N}(\mathfrak{L} p 2 M)\right\rangle\right]\) using powr-powr \(\langle p 2>0\rangle\) by auto
also have \(\ldots=\left(\int^{+} x .|g x|\right.\) powr p2 \(\left.\partial M\right)\)
by (rule nn-integral-eq-integral[symmetric], auto simp add: intg)
finally have ennreal ((Norm \((\mathfrak{L} p 2 M) g)\) powr p2) \(\leq\) ennreal (l powr p2)
using \(\left\langle\left(\int^{+} x . \mid g\right.\right.\) x| powr p2 \(\left.\partial M\right) \leq\) ennreal(l powr p2) > by auto
then have \(((\operatorname{Norm}(\mathfrak{L}\) p2 M) g) powr p2) powr \((1 / p 2) \leq(l\) powr 2 2) powr (1/p2)
using ennreal-le-iff \(\langle l \geq 0\rangle\langle p 2>0\rangle\) powr-mono2 by auto
then have \(\operatorname{Norm}(\mathfrak{L} p 2 M) g \leq l\)
using \(\langle p 2>0\rangle\langle l \geq 0\rangle\) by (auto simp add: powr-powr)
then have \(e \operatorname{Norm}(\mathfrak{L} p 2 M) g \leq l e\)
unfolding eNorm-Norm \(\left[\right.\) OF \(\left\langle g \in\right.\) space \(_{N}(\mathfrak{L}\) p2 \(\left.M)\right\rangle\) 〕 \(\langle l e=\) ennreal l \(\rangle\) using ennreal-leI by auto
then show ?thesis unfolding \(l e-d e f\langle p=\) ennreal \(p 2\rangle\) by \(\operatorname{simp}\)
next
case PInf
then have \(A E x\) in \(M . \forall n .|f(r n) x| \leq \operatorname{Norm}(\mathfrak{L} \infty M)(f(r n))\)
apply (subst AE-all-countable) using L-infinity-AE-bound \(\langle\bigwedge n . f(r n) \in\) space \(_{N}(\mathfrak{L} p M)\) > by blast
moreover have \(|g x| \leq l\) if \(\forall n .|f(r n) x| \leq \operatorname{Norm}(\mathfrak{L} \infty M)(f(r n))(\lambda n . f\) \(n x) \longrightarrow g x\) for \(x\)
proof -
have \((\lambda n . f(r n) x) \longrightarrow g x\)
using that LIMSEQ-subseq-LIMSEQ[OF - 〈strict-mono \(r\rangle]\) unfolding comp-def by auto
then have \(*:(\lambda n .|f(r n) x|) \longrightarrow|g x|\)
by (auto intro!:tendsto-intros)
show ?thesis
apply (rule LIMSEQ-le[OF *]) using that(1) \(\langle(\lambda n\). Norm ( \(\mathfrak{L} p M)(f(r\) \(n))\) ) \(l>\) unfolding PInf by auto
qed
ultimately have \(A E x\) in \(M .|g x| \leq l\) using \(<A E x\) in \(M .(\lambda n . f n x) \longrightarrow\) \(g x>\) by auto
then have \(g \in\) space \(_{N}(\mathfrak{L} \infty M) \operatorname{Norm}(\mathfrak{L} \infty M) g \leq l\)
using L-infinity-I[OF \(\langle g \in\) borel-measurable \(M\rangle-\langle l \geq 0\rangle]\) by auto
then have \(e \operatorname{Norm}(\mathfrak{L} \infty M) g \leq l e\)
unfolding eNorm-Norm[OF \(\left.\left\langle g \in \operatorname{space}_{N}(\mathfrak{L} \infty M)\right\rangle\right]\langle l e=\) ennreal \(l\rangle\) using ennreal-leI by auto
```

    then show ?thesis unfolding le-def \langlep=\infty> by simp
    qed
    qed
lemma Lp-AE-limit':
assumes g \in borel-measurable M
\n.fn \in space
AE x in M. (\lambdan.fnx)\longrightarrowgx
(\lambdan.Norm (L p M) (fn))\longrightarrowl
shows g\in space}\mp@subsup{N}{N}{}(\mathfrak{L}pM
Norm ({ p M)g\leql
proof -
have l \geq 0 by (rule LIMSEQ-le-const[OF}\langle(\lambdan.Norm ({ p M) (fn))
l>], auto)
have (\lambdan. eNorm ({ p M) (f n)) \longrightarrow ennreal l
unfolding eNorm-Norm[OF<\n.fn \in space}\mp@subsup{N}{N}{}(\mathfrak{L}pM)>] using <(\lambdan. Norm
( L p M) (fn)) \longrightarrowl> by auto
then have *: ennreal l = liminf ( }\lambdan.eNorm ({)pM)(fn)
using lim-imp-Liminf[symmetric] trivial-limit-sequentially by blast
have eNorm ( }\mathfrak{L}pM)g\leqennreal
unfolding * apply (rule Lp-AE-limit) using assms by auto
then have eNorm ( }\mathfrak{L}pM)g<\infty\mathrm{ using le-less-trans by fastforce
then show g\in \mp@subsup{\mathrm{ sace N}}{N}{}(\mathfrak{L}pM)\mathrm{ using spaceN-iff by auto}
show Norm ({ p M ) g\leql
using<eNorm (L }pM)g\leqennreal l> ennreal-le-iff[OF <l\geq0>]
unfolding eNorm-Norm[OF}\langleg\in\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}pM)\rangle]\mathrm{ by auto
qed
lemma Lp-AE-limit'':
assumes g\in borel-measurable M
\n.fn \in space}\mp@subsup{N}{N}{({LpM)
AEx in M. (\lambdan.fnx)\longrightarrowgx
\n.Norm ({ p M) (fn)\leqC
shows g\in space}N({\mathfrak{L}pM
Norm ({ p M)g\leqC
proof -
have C\geq0 by (rule order-trans[OF Norm-nonneg[of L p Mf 0]<Norm({) p
M) (f0) \leqC>])
have *: liminf }(\lambdan. ennreal C)= ennreal C
using Liminf-const trivial-limit-at-top-linorder by blast
have eNorm ({ pM) (fn)\leqennreal C for n
unfolding eNorm-Norm[OF<f n\in space
using <Norm ({ p M) (fn)\leqC> by (auto simp add: ennreal-leI)
then have liminf ( }\lambda\mathrm{ n. eNorm ({ L p M) (fn)) < ennreal C
using Liminf-mono[of (\lambdan. eNorm ({) pM) (fn)) \lambda-.C sequentially]* by auto
then have eNorm ( L p M)g\leqennreal C using
Lp-AE-limit[OF}\langleg\in\mathrm{ borel-measurable M><AE x in M. ( }\lambdan.fnx)\longrightarrow
x>, of p] by auto
then have eNorm ({ p M)g<\infty}\mathrm{ ( using le-less-trans by fastforce

```
```

    then show g\in space}\mp@subsup{N}{N}{}(\mathfrak{L}pM)\mathrm{ using spaceN-iff by auto
    show Norm ({ p M)g\leqC
    using <eNorm ({ p M)g\leqennreal C` ennreal-le-iff[OF 〈C\geq0`]
    unfolding eNorm-Norm[OF <g\in \mp@subsup{\mathrm{ space N}}{N}{}(\mathfrak{L}pM)\rangle] by auto
    qed

```

We give the version of Lebesgue dominated convergence theorem in the setting of \(L^{p}\) spaces.
proposition Lp-domination-limit:
fixes \(p\) ::real
assumes [measurable]: \(g \in\) borel-measurable \(M\)
\n.f \(n \in\) borel-measurable \(M\)
and \(m \in\) space \(_{N}(\mathfrak{L} p M)\)
\(A E x\) in \(M .(\lambda n . f n x) \longrightarrow g x\)
\(\wedge n . A E x\) in \(M .|f n x| \leq m x\)
shows \(g \in \operatorname{space}_{N}(\mathfrak{L} p M)\)
tendsto-in \(_{N}(\mathfrak{L} p M) f g\)
proof -
have [measurable]: \(m \in\) borel-measurable \(M\) using Lp-measurable[OF \(\langle m \in\) space \(\left.\left._{N}(\mathfrak{L} p M)\right\rangle\right]\) by auto
have \(f n \in \operatorname{space}_{N}(\mathfrak{L} p M)\) for \(n\)
apply (rule Lp-domination[OF \(\left.\left.-\left\langle m \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\right]\right)\) using \(\langle A E x\) in \(M\). \(|f n x| \leq m x\rangle\) by auto
have \(A E x\) in \(M . \forall n .|f n x| \leq m x\)
apply (subst AE-all-countable) using <\} n \text { . AE } x \text { in } M \text { . } | f n x | \leq m x \rangle \text { by auto }
moreover have \(|g x| \leq m x\) if \(\forall n\). \(|f n x| \leq m x(\lambda n\). \(f n x) \longrightarrow g x\) for \(x\)
apply (rule LIMSEQ-le-const2 [of \(\lambda n .|f n x|]\) ) using that by (auto intro!:tendsto-intros)
ultimately have \(*: A E x\) in \(M .|g x| \leq m x\) using \(\langle A E x\) in \(M\). \((\lambda n . f n x)\)
\(\longrightarrow g x\rangle\) by auto
show \(g \in \operatorname{space}_{N}(\mathfrak{L} p M)\)
apply (rule Lp-domination \(\left.\left[O F-\left\langle m \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\right]\right)\) using \(*\) by auto
```

have $(\lambda n . \operatorname{Norm}(\mathfrak{L} p M)(f n-g)) \longrightarrow 0$
proof (cases $p \leq 0$ )
case True
then have ennreal $p=0$ by (simp add: ennreal-eq- 0 -iff)
then show ?thesis unfolding Norm-def by (auto simp add: L-zero(1))
next
case False
then have $p>0$ by auto
have $\left(\lambda n .\left(\int x .|f n x-g x|\right.\right.$ powr $\left.\left.p \partial M\right)\right) \longrightarrow\left(\int x .|0|\right.$ powr p $\left.\partial M\right)$
proof (rule integral-dominated-convergence[of $-(\lambda x .|2 * m x|$ powr $p)]$,
auto)
show integrable $M(\lambda x .|2 * m x|$ powr $p)$
unfolding abs-mult apply (subst powr-mult)
using $L p-D(2)\left[O F\langle p>0\rangle\left\langle m \in\right.\right.$ space $\left.\left._{N}(\mathfrak{L} p M)\right\rangle\right]$ by auto
have $(\lambda n .|f n x-g x|$ powr $p) \longrightarrow|0|$ powr $p$ if $(\lambda n$. $f n x) \longrightarrow g x$
for $x$

```
apply (rule tendsto-powr') using \(\langle p>0\rangle\) that apply (auto)
using Lim-null tendsto-rabs-zero-iff by fastforce
then show \(A E x\) in \(M .(\lambda n .|f n x-g x|\) powr \(p) \longrightarrow 0\)
using \(\langle A E x\) in \(M\). \((\lambda n\). \(f n x) \longrightarrow g x\rangle\) by auto
have \(|f n x-g x|\) powr \(p \leq|2 * m x|\) powr \(p\) if \(|f n x| \leq m x|g x| \leq m x\) for \(n x\)
using powr-mono2 \(\langle p>0\rangle\) that by auto
then show \(A E x\) in \(M\). \(|f n x-g x|\) powr \(p \leq|2 * m x|\) powr \(p\) for \(n\)
using \(\langle A E x\) in \(M\). \(| f n x|\leq m x\rangle\langle A E x\) in \(M| g x.|\leq m x\rangle\) by auto
qed
then have \((\lambda n .(\operatorname{Norm}(\mathfrak{L} p M)(f n-g))\) powr \(p) \longrightarrow(\operatorname{Norm}(\mathfrak{L} p M) 0)\) powr \(p\)
unfolding \(L p-D\left[O F\langle p>0\rangle\right.\) spaceN-diff \(\left[O F\left\langle\bigwedge n . f n \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\langle g\right.\) \(\left.\left.\left.\in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\right]\right]\)
using \(\langle p>0\rangle\) by (auto simp add: powr-powr)
then have \((\lambda n\). \(((\operatorname{Norm}(\mathfrak{L} p M)(f n-g))\) powr \(p)\) powr \((1 / p)) \longrightarrow\) ( \(\operatorname{Norm}(\mathfrak{L} p M) 0)\) powr \(p)\) powr \((1 / p)\)
by (rule tendsto-powr \({ }^{\prime}\), auto simp add: \(\langle p>0\rangle\) )
then show ?thesis using powr-powr \(\langle p>0\rangle\) by auto
qed
then show tendsto-in \({ }_{N}(\mathfrak{L} p M) f g\)
unfolding tendsto-in \({ }_{N}\)-def by auto
qed
We give the version of the monotone convergence theorem in the setting of \(L^{p}\) spaces.
proposition Lp-monotone-limit:
fixes \(f::\) nat \(\Rightarrow{ }^{\prime} a \Rightarrow\) real
assumes \(p>(0::\) ennreal \()\)
\(A E x\) in M. incseq \((\lambda n . f n x)\)
\(\bigwedge n\). Norm \((\mathfrak{L} p M)(f n) \leq C\)
\(\bigwedge n . f n \in \operatorname{space}_{N}(\mathfrak{L} p M)\)
shows \(A E x\) in \(M\). convergent \((\lambda n . f n x)\)
\((\lambda x \cdot \lim (\lambda n . f n x)) \in \operatorname{space}_{N}(\mathfrak{L} p M)\)
\(\operatorname{Norm}(\mathfrak{L} p M)(\lambda x . \lim (\lambda n . f n x)) \leq C\)
proof -
have [measurable]: \(f n \in\) borel-measurable \(M\) for \(n\) using Lp-measurable[OF \(\operatorname{assms}(4)]\).
show \(A E x\) in \(M\). convergent \((\lambda n . f n x)\)
proof (cases rule: Lp-cases[of p])
case PInf
have \(A E x\) in \(M\). \(|f n x| \leq C\) for \(n\)
using L-infinity-AE-bound \([\) of \(f n M]\langle\operatorname{Norm}(\mathfrak{L} p M)(f n) \leq C\rangle\left\langle f n \in\right.\) space \(_{N}\)
\((\mathfrak{L} p M)\) >
unfolding \(\langle p=\infty\) 〉 by auto
then have *: AE \(x\) in \(M . \forall n .|f n x| \leq C\)
by (subst AE-all-countable, auto)
have \((\lambda n . f n x) \longrightarrow(S U P n . f n x)\) if incseq \((\lambda n . f n x) \bigwedge n .|f n x| \leq C\) for \(x\)
apply（rule LIMSEQ－incseq－SUP［OF－〈incseq（ \(\lambda n\) ．f \(n x)\rangle]\) ）using that（2） abs－le－D1 by fastforce
then have convergent \((\lambda n\) ．\(f n x)\) if incseq \((\lambda n\) ．\(f n x) \bigwedge n\) ．\(|f n x| \leq C\) for \(x\) unfolding convergent－def using that by auto
then show ？thesis using \(\langle A E x\) in \(M\) ．incseq \((\lambda n . f n x)\rangle *\) by auto

\section*{next}
case（real－pos p2）
define \(g\) where \(g=(\lambda n . f n-f 0)\)
have \(A E x\) in \(M\) ．incseq \((\lambda n . g n x)\)
unfolding \(g\)－def using \(\langle A E x\) in \(M\) ．incseq \((\lambda n . f n x)\rangle\) by（simp add：
incseq－def）
have \(g n \in\) space \(_{N}(\mathfrak{L} p 2 M)\) for \(n\)
unfolding \(g\)－def using \(\left\langle\bigwedge n . f n \in\right.\) space \(\left._{N}(\mathfrak{L} p M)\right\rangle\) unfolding \(\langle p=\) ennreal p2）by auto
then have［measurable］：\(g n \in\) borel－measurable \(M\) for \(n\) using Lp－measurable by auto
define \(D\) where \(D=\operatorname{defect}(\mathfrak{L} p 2 M) * C+\operatorname{defect}(\mathfrak{L} p 2 M) * C\)
have \(\operatorname{Norm}(\mathfrak{L} p 2 M)(g n) \leq D\) for \(n\)
proof－
have \(f n \in \operatorname{space}_{N}(\mathfrak{L} p 2 M)\) using \(\left\langle f n \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\) unfolding \(\langle p=\) ennreal p2＞by auto
have \(\operatorname{Norm}(\mathfrak{L} p 2 M)(g n) \leq \operatorname{defect}(\mathfrak{L} p 2 M) * \operatorname{Norm}(\mathfrak{L} p 2 M)(f n)+\) \(\operatorname{defect}(\mathfrak{L} p 2 M) * \operatorname{Norm}(\mathfrak{L} p \mathcal{2} M)(f 0)\)
unfolding \(g\)－def using Norm－triangular－ineq－diff \(\left[O F<f n \in \operatorname{space}_{N}(\mathfrak{L}\right.\) p2 \(M)\rangle\) ］by auto
also have \(\ldots \leq D\)
unfolding \(D\)－def apply（rule add－mono）
using mult－left－mono defect－ge－1［of \(\mathfrak{L} p 2 M]<\backslash n . \operatorname{Norm}(\mathfrak{L} p M)(f n) \leq\)
\(C 〉\) unfolding \(\langle p=\) ennreal \(p 2\rangle\) by auto
finally show ？thesis by simp
qed
have \(g\)－bound：\(\left(\int{ }^{+} x .|g n x|\right.\) powr p2 \(\left.\left.\partial M\right) \leq \operatorname{ennreal(D~powr~} p 2\right)\) for \(n\)
proof－
have \(\left(\int{ }^{+} x .|g n x|\right.\) powr p2 \(\left.\partial M\right)=\operatorname{ennreal}\left(\int x .|g n x|\right.\) powr p2 \(\left.\partial M\right)\)
apply（rule nn－integral－eq－integral）using \(L p-D(2)[O F\langle p 2>0\rangle\langle g n \in\) space \(\left.\left._{N}(\mathfrak{L} p 2 M)\right\rangle\right]\) by auto
also have \(\ldots=\operatorname{ennreal}((\operatorname{Norm}(\mathfrak{L}\) p2 \(M)(g n))\) powr p2 \()\)
apply（subst Lp－Norm（2）［OF \(\langle p 2>0\rangle\) ，of \(g n\) ，symmetric］）by auto
also have \(\ldots \leq \operatorname{ennreal(D~powr~p2)~}\)
by（auto intro！：powr－mono2 simp add：less－imp－le［OF \(\langle p 2>0\rangle]\langle\operatorname{Norm}(\mathfrak{L}\) p2 M）\((g n) \leq D\) ）\()\)
finally show ？thesis by simp
qed
have \(\forall n . g n x \geq 0\) if \(\operatorname{incseq}(\lambda n . f n x)\) for \(x\)
unfolding \(g\)－def using that by（auto simp add：incseq－def）
then have \(A E x\) in \(M . \forall n . g n x \geq 0\) using \(\langle A E x\) in \(M\) ．incseq \((\lambda n . f n x)\) 〉 by auto
define \(h\) where \(h=(\lambda n x\) ．ennreal \((|g n x|\) powr p2 \()\) ）
have [measurable]: \(h n \in\) borel-measurable \(M\) for \(n\) unfolding \(h\)-def by auto define \(H\) where \(H=(\lambda x\). (SUP n. \(h n x))\)
have [measurable]: \(H \in\) borel-measurable \(M\) unfolding \(H\)-def by auto
have \(\bigwedge n\). \(h n x \leq h(\) Suc \(n) x\) if \(\forall n\).g \(n x \geq 0 \operatorname{incseq}(\lambda n\).g \(n x\) ) for \(x\) unfolding \(h\)-def apply (auto intro!:powr-mono2)
apply (auto simp add: less-imp-le[OF \(\langle p 2>0\rangle]\) ) using that incseq-SucD by auto
then have \(*: A E x\) in \(M . h n x \leq h(S u c n) x\) for \(n\)
using \(\langle A E x\) in \(M . \forall n . g n x \geq 0\rangle\langle A E x\) in \(M . \operatorname{incseq}(\lambda n . g n x)\rangle\) by auto
have \(\left(\int{ }^{+} x . H x \partial M\right)=\left(S U P n . \int{ }^{+} x . h n x \partial M\right)\)
unfolding \(H\)-def by (rule nn-integral-monotone-convergence-SUP-AE, auto simp add: *)
also have \(\ldots \leq \operatorname{ennreal}(D\) powr \(p 2)\)
unfolding \(H\)-def \(h\)-def using \(g\)-bound by (simp add: SUP-least)
finally have \(\left(\int{ }^{+} x . H \times \partial M\right)<\infty\) by (simp add: le-less-trans)
then have \(A E x\) in \(M . H x \neq \infty\)
by (metis (mono-tags, lifting) \(\langle H \in\) borel-measurable \(M\rangle\) infinity-ennreal-def nn-integral-noteq-infinite top.not-eq-extremum)
have convergent \((\lambda n . f n x)\) if \(H x \neq \infty \operatorname{incseq}(\lambda n . f n x)\) for \(x\)
proof -
define \(A\) where \(A=\operatorname{enn2real}(H x)\)
then have \(H x=\) ennreal \(A\) using \(\langle H x \neq \infty\rangle\) by (simp add: ennreal-enn2real-if)
have \(f n x \leq f 0 x+A\) powr \((1 / p 2)\) for \(n\)
proof -
have ennreal \((|g n x|\) powr p2 \() \leq\) ennreal \(A\)
unfolding \(\langle H x=\) ennreal \(A\rangle[\) symmetric \(] H\)-def \(h\)-def by (meson SUP-upper2
UNIV-I order-refl)
then have \(|g n x|\) powr \(p 2 \leq A\)
by (subst ennreal-le-iff [symmetric], auto simp add: A-def)
have \(|g n x|=(|g n x|\) powr p2) powr (1/p2)
using \(\langle p 2>0\rangle\) by (simp add: powr-powr)
also have \(\ldots \leq A\) powr (1/p2)
apply (rule powr-mono2) using \(\langle p 2>0\rangle\langle | g n x \mid\) powr \(p 2 \leq A\rangle\) by auto
finally have \(|g n x| \leq A\) powr (1/p2) by simp
then show ?thesis unfolding \(g\)-def by auto
qed
then show convergent ( \(\lambda n\). \(f n x\) )
using LIMSEQ-incseq-SUP[OF - 〈incseq ( \(\lambda n\). f \(n x)\rangle\) ] convergent-def by
(metis bdd-aboveI2)
qed
then show \(A E x\) in \(M\). convergent \((\lambda n . f n x)\)
using \(\langle A E x\) in \(M . H x \neq \infty\rangle\langle A E x\) in \(M\). incseq \((\lambda n\). \(f n x)\rangle\) by auto
qed (insert \(\langle p>0\rangle\), simp)
then have lim: \(A E x\) in \(M .(\lambda n . f n x) \longrightarrow \lim (\lambda n . f n x)\)
using convergent-LIMSEQ-iff by auto
show \((\lambda x . \lim (\lambda n . f n x)) \in \operatorname{space}_{N}(\mathfrak{L} p M)\)
apply (rule Lp-AE-limit'" \(\left[\right.\) of \(--f, O F-\left\langle\bigwedge n . f n \in\right.\) space \(\left._{N}(\mathfrak{L} p M)\right\rangle \lim \langle\bigwedge n\). \(\operatorname{Norm}(\mathfrak{L} p M)(f n) \leq C>])\)
```

    by auto
    show \(\operatorname{Norm}(\mathfrak{L} p M)(\lambda x . \lim (\lambda n . f n x)) \leq C\)
    apply (rule Lp-AE-limit' \({ }^{\prime \prime}\left[o f--f\right.\), OF \(-\left\langle\bigwedge n . f n \in\right.\) space \(\left._{N}(\mathfrak{L} p M)\right\rangle \lim \langle\bigwedge n\).
    $\operatorname{Norm}(\mathfrak{L} p M)(f n) \leq C>])$
by auto
qed

```

\subsection*{5.7 Completeness of \(L^{p}\)}

We prove the completeness of \(L^{p}\).
theorem Lp-complete:
complete \(_{N}(\mathfrak{L} p M)\)
proof (cases rule: Lp-cases[of p])
case zero
show ?thesis
proof (rule complete \({ }_{N}-I\) )
fix \(u\) assume \(\forall(n:: n a t) . u n \in \operatorname{space}_{N}(\mathfrak{L} p M)\)
then have tendsto-in \({ }_{N}(\mathfrak{L} p M)\) u 0
unfolding tendsto-in \({ }_{N}\)-def Norm-def \(\langle p=0\rangle L\)-zero(1) L-zero-space by auto
then show \(\exists x \in\) space \(_{N}(\mathfrak{L} p M)\). tendsto-in \(N_{N}(\mathfrak{L} p M) u x\)
by auto
qed
next
case (real-pos p2)
show ?thesis
proof \(\left(\right.\) rule complete \(_{N}-I^{\prime}[\) of \(\lambda n .(1 / 2) \widehat{n} *(1 /(\operatorname{defect}(\mathfrak{L} p M)) \uparrow(\) Suc \(n))]\), unfold \(\langle p=\) ennreal \(p 2\rangle\) )
show \(0<(1 / \mathcal{Z}){ }^{\wedge} n *\left(1 / \operatorname{defect}(\mathfrak{L}(\right.\) ennreal \(p \mathcal{Z}) M){ }^{\text {^ Suc } n) \text { for } n}\)
using defect-ge-1[of \(\mathfrak{L}\) (ennreal p2) \(M\) ] by (auto simp add: divide-simps)
fix \(u\) assume \(\forall(n:: n a t) . u n \in \operatorname{space}_{N}(\mathfrak{L} p 2 M) \forall n . \operatorname{Norm}(\mathfrak{L} p 2 M)(u n) \leq\) \((1 / 2) \uparrow n *(1 /(\operatorname{defect}(\mathfrak{L} p 2 M)) \uparrow(S u c n))\)
then have \(H: \bigwedge n\). u \(n \in \operatorname{space}_{N}(\mathfrak{L} p 2 M)\)
\[
\wedge n . \operatorname{Norm}(\mathfrak{L} p 2 M)(u n) \leq(1 / \mathcal{Z}) \wedge n *(1 /(\operatorname{defect}(\mathfrak{L} p 2 M)) \uparrow(S u c
\]
n))
unfolding \(\langle p=\) ennreal \(p 2\) 〉 by auto
have [measurable]: \(u n \in\) borel-measurable \(M\) for \(n\) using Lp-measurable[OF \(H(1)]\).
define \(w\) where \(w=\left(\lambda N x .\left(\sum n \in\{. .<N\} .|u n x|\right)\right)\)
have \(w 2: w=(\lambda N\). sum \((\lambda n x .|u n x|)\{. .<N\})\) unfolding \(w\)-def apply (rule ext) +
by (metis (mono-tags, lifting) sum.cong fun-sum-apply)
have incseq \((\lambda N . w N x)\) for \(x\) unfolding w2 by (rule incseq-SucI, auto)
then have \(w N\)-inc: \(A E x\) in \(M\). incseq \((\lambda N . w N x)\) by simp
have abs-u-space: \((\lambda x .|u n x|) \in\) space \(_{N}(\mathfrak{L} p 2 M)\) for \(n\)
by (rule Lp-Banach-lattice \(\left[O F\left\langle u n \in\right.\right.\) space \(_{N}(\mathfrak{L}\) p2 M) \(\rangle\) ])
then have \(w N\)-space: \(w N \in \operatorname{space}_{N}(\mathfrak{L} p 2 M)\) for \(N\) unfolding w2 using
\(H(1)\) by auto
have abs-u-Norm: Norm \((\mathfrak{L}\) p2 \(M)(\lambda x .|u n x|) \leq(1 / 2){ }^{\wedge} n *(1 /(\operatorname{defect}(\mathfrak{L}\) \(p 2 M)\) ) (Suc n)) for \(n\)
using Lp-Banach-lattice(2)[OF \(\left\langle u n \in\right.\) space \(\left.\left._{N}(\mathfrak{L} p 2 \mathrm{M})\right\rangle\right] H(2)\) by auto
have \(w N\)-Norm: Norm \((\mathfrak{L} p 2 M)(w N) \leq 2\) for \(N\)
proof -
have \(*:(\operatorname{defect}(\mathfrak{L} p \mathcal{Z} M)) \uparrow(\) Suc \(n) \geq 0(\operatorname{defect}(\mathfrak{L} p \mathcal{Z} M)) \uparrow(\) Suc \(n)>0\) for \(n\) using defect-ge-1[of \(\mathfrak{L} p 2 M]\) by auto
have \(\operatorname{Norm}(\mathfrak{L} p \mathfrak{2} M)(w N) \leq\left(\sum n<N .(\operatorname{defect}(\mathfrak{L} p 2 M)) \uparrow(\right.\) Suc \(n) * N o r m\) \((\mathfrak{L} p \mathcal{Z} M)(\lambda x .|u n x|))\)
unfolding w2 lessThan-Suc-atMost[symmetric] by (rule Norm-sum, simp add: abs-u-space)
also have \(\ldots \leq\left(\sum n<N .(\operatorname{defect}(\mathfrak{L} p \mathscr{2} M)) \uparrow(S u c n) *((1 / \mathcal{Z}) \wedge n *(1 /(\right.\) defect \(\left.\left.\left.(\mathfrak{L} p 2 M))^{\wedge}(S u c n)\right)\right)\right)\)
apply (rule sum-mono, rule mult-left-mono) using abs-u-Norm * by auto also have \(\ldots=\left(\sum n<N .(1 / 2)^{\wedge} n\right)\)
using \(*\) (2) defect-ge-1 \([\) of \(\mathfrak{L}\) p2 \(M]\) by (auto simp add: algebra-simps) also have \(\ldots \leq\left(\sum n .(1 / 2)^{\wedge} n\right)\)
unfolding lessThan-Suc-atMost[symmetric] by (rule sum-le-suminf, rule summable-geometric[of 1/2], auto)
also have \(\ldots=2\) using suminf-geometric[of 1/2] by auto
finally show? ?thesis by simp
qed
have \(A E x\) in \(M\). convergent \((\lambda N . w N x)\)
apply (rule Lp-monotone-limit[OF \(\langle p>0\rangle\), of - 2], unfold \(\langle p=\) ennreal p2>)
using \(w N\)-inc \(w N\)-Norm \(w N\)-space by auto
define \(m\) where \(m=(\lambda x \cdot \lim (\lambda N . w N x))\)
have \(m\)-space: \(m \in\) space \(_{N}(\mathfrak{L}\) p2 \(M)\)
unfolding \(m\)-def \(\langle p=\) ennreal \(p 2\rangle[\) symmetric] apply (rule Lp-monotone-limit[OF \(\langle p>0\rangle\), of - 2], unfold \(\langle p=\) ennreal \(p 2\rangle\) )
using \(w N\)-inc \(w N\)-Norm \(w N\)-space by auto
define \(v\) where \(v=\left(\lambda x\right.\). \(\left.\left(\sum n . u n x\right)\right)\)
have \(v\)-meas: \(v \in\) borel-measurable \(M\) unfolding \(v\)-def by auto
have u-meas: \(\bigwedge n .(\) sum \(u\{0 . .<n\}) \in\) borel-measurable \(M\) by auto \{
fix \(x\) assume convergent \((\lambda N . w N x)\)
then have \(S\) : summable ( \(\lambda n\). \(|u n x|\) ) unfolding \(w\)-def using summable-iff-convergent by auto
 inf-eq-lim)
have summable ( \(\lambda n . u n x)\) using \(S\) by (rule summable-rabs-cancel)
then have \(*:(\lambda n .(\operatorname{sum} u\{. .<n\}) x) \longrightarrow v x\)
unfolding \(v\)-def fun-sum-apply by (metis convergent-LIMSEQ-iff sum-inf-eq-lim summable-iff-convergent)
```

    have |(sum u{..<n}) x| \leqmx for n
    proof -
    have |(sum u {..<n}) x| \leq (\sumi\in{..<n}. |u ix|)
        unfolding fun-sum-apply by auto
    also have ...\leq(\sumi.|uix|)
        apply (rule sum-le-suminf) using S by auto
        finally show ?thesis using <mx = (\sumn. |unx|)\rangle by simp
    qed
    then have }(\foralln.|(\operatorname{sum}u{0..<n})x|\leqmx)\wedge(\lambdan.(sum u {0..<n})x
    |x
unfolding atLeastOLessThan using * by auto
}
then have m-bound: \n. AE x in M. |(sum u {0..<n}) x|\leqmx
and u-conv:AE x in M. (\lambdan. (sum u{0..<n}) x)\longrightarrowv
using <AE x in M. convergent ( }\lambdaN.wNx)\rangle\mathrm{ by auto
have tendsto-in
by (rule Lp-domination-limit[OF v-meas u-meas m-space u-conv m-bound])
moreover have v\in space}N({~L2 M
by (rule Lp-domination-limit[OF v-meas u-meas m-space u-conv m-bound])
ultimately show }\existsv\in\mp@subsup{\mathrm{ space }}{N}{}(\mathfrak{L}\mathrm{ p2 M). tendsto-in}\mp@subsup{N}{N}{\prime}(\mathfrak{L}\mathrm{ p2 M) ( \n. sum u
{0..<n})v
by auto
qed
next
case PInf
show ?thesis
proof (rule complete N
fix }u\mathrm{ assume }\forall(n::nat).un\in\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}pM)\foralln.Norm ({) { M) (un)
(1/2) ^}
then have H: \n.un { \mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}\inftyM)\bigwedgen.Norm (\mathfrak{L}\inftyM) (un)\leq(1/\mathcal{Q})
`n}\mathrm{ using PInf by auto     have [measurable]: u n \in borel-measurable M for n using Lp-measurable[OF H(1)] by auto     define v}\mathrm{ where v=( }\lambdax.\sumn.unx     have [measurable]: v\in borel-measurable M unfolding v-def by auto     define w where w=( }\lambdaNx.(\sumn\in{0..<N}. u n x)      have [measurable]: w N \in borel-measurable M for N unfolding w-def by auto     have AE x in M. |u n x| \leq (1/2)^n for n         using L-infinity-AE-bound[OF H(1), of n] H(2)[of n] by auto     then have AE x in M.\foralln. |u n x| \leq (1/2)`n
by (subst AE-all-countable, auto)

```

```

N x
proof -
have *: \n. |u n x | \leq (1/2)^n using that by auto
have **: summable (\lambdan. |u n x|)
apply (rule summable-norm-cancel, rule summable-comparison-test'[OF

```
```

summable-geometric[of 1/2]])
using * by auto
have |wN x-v x| = |(\sumn.u(n+N) x)|
unfolding v-def w-def
apply (subst suminf-split-initial-segment[OF summable-rabs-cancel[OF
<summable (\lambdan. || n x|)>], of N])
by (simp add: lessThan-atLeast0)
also have ... \leq (\sumn. |u (n+N)x|)
apply (rule summable-rabs, subst summable-iff-shift) using ** by auto
also have ... \leq (\sumn.(1/2)`(n+N))
proof (rule suminf-le)
show \n. |u (n+N) x| \leq (1/2)^ (n+N)
using *[of - + N] by simp
show summable (\lambdan. |u(n+N)x|)
using ** by (subst summable-iff-shift) simp
show summable (\lambdan. (1/2::real) ^ (n+N))
using summable-geometric [of 1/2] by (subst summable-iff-shift) simp
qed
also have ... = (1/2)^N*(\sumn.(1/2)^n)
by (subst power-add, subst suminf-mult2[symmetric], auto simp add:
summable-geometric[of 1/2])
also have ... = (1/2)^N*2
by (subst suminf-geometric, auto)
finally show ?thesis by simp
qed
ultimately have *: AE x in M. |wN x-vx|\leq(1/2)^N*2 for N by auto
have **:wN-v\in \mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}\inftyM)\operatorname{Norm}(\mathfrak{L}\inftyM)(wN-v)\leq(1/2)^N

* 2 for N
unfolding fun-diff-def using L-infinity-I[OF - *] by auto
have l:(\lambdaN.((1/2)^N)*(2::real))\longrightarrow0*2
by (rule tendsto-mult, auto simp add: LIMSEQ-realpow-zero[of 1/2])
have tendsto-in}N({\mathfrak{L}\inftyM)wv unfolding tendsto-in N-def
apply (rule tendsto-sandwich[of \lambda-. 0- - \lambdan. (1/2)^n * 2]) using l **(2)
by auto
have v=-( w 0 - v) unfolding w-def by auto
then have v}\in\mp@subsup{\mathrm{ space N}}{N}{}(\mathfrak{L}\inftyM)\mathrm{ using **(1)[of 0] spaceN-add spaceN-diff by
fastforce
then show \existsv\in \mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}pM).tendsto-in}N(\mathfrak{L}pM)(\lambdan.sum u {0..<n}
v
using <tendsto-in
by auto
qed (simp)
qed

```

\subsection*{5.8 Multiplication of functions, duality}

The next theorem asserts that the multiplication of two functions in \(L^{p}\) and \(L^{q}\) belongs to \(L^{r}\), where \(r\) is determined by the equality \(1 / r=1 / p+1 / q\).

This is essentially a case by case analysis, depending on the kind of \(L^{p}\) space we are considering. The only nontrivial case is when \(p, q\) (and \(r\) ) are finite and nonzero. In this case, it reduces to Hölder inequality.
theorem \(L p\)-Lq-mult:
fixes \(p q r\) ::ennreal
assumes \(1 / p+1 / q=1 / r\)
and \(f \in\) space \(_{N}(\mathfrak{L} p M) g \in \operatorname{space}_{N}(\mathfrak{L} q M)\)
shows \((\lambda x . f x * g x) \in\) space \(_{N}(\mathfrak{L} r M)\)
\(\operatorname{Norm}(\mathfrak{L} r M)(\lambda x . f x * g x) \leq \operatorname{Norm}(\mathfrak{L} p M) f * \operatorname{Norm}(\mathfrak{L} q M) g\)
proof -
have [measurable]: \(f \in\) borel-measurable \(M g \in\) borel-measurable \(M\) using Lp-measurable assms by auto
have \((\lambda x . f x * g x) \in \operatorname{space}_{N}(\mathfrak{L} r M) \wedge \operatorname{Norm}(\mathfrak{L} r M)(\lambda x . f x * g x) \leq N o r m\) \((\mathfrak{L} p M) f * \operatorname{Norm}(\mathfrak{L} q M) g\)
proof (cases rule: Lp-cases[of r])
case zero
have \(*:(\lambda x . f x * g x) \in\) borel-measurable \(M\) by auto
then have \(\operatorname{Norm}(\mathfrak{L} r M)(\lambda x . f x * g x)=0\) using L-zero[of \(M]\) unfolding
Norm-def zero by auto
then have \(\operatorname{Norm}(\mathfrak{L} r M)(\lambda x . f x * g x) \leq \operatorname{Norm}(\mathfrak{L} p M) f * \operatorname{Norm}(\mathfrak{L} q M)\) \(g\) using Norm-nonneg by auto
then show ?thesis unfolding zero using * L-zero-space [of \(M\) ] by auto
next
case (real-pos r2)
have \(p>0 q>0\) using \(\langle 1 / p+1 / q=1 / r\rangle\langle r>0\rangle\)
by (metis ennreal-add-eq-top ennreal-divide-eq-top-iff ennreal-top-neq-one gr-zeroI zero-neq-one)+
consider \(p=\infty|q=\infty| p<\infty \wedge q<\infty\) using top.not-eq-extremum by force
then show ?thesis
proof (cases)
case 1
then have \(q=r\) using \(\langle 1 / p+1 / q=1 / r\rangle\)
by (metis ennreal-divide-top infinity-ennreal-def one-divide-one-divide-ennreal semiring-normalization-rules(5))
have \(A E x\) in \(M .|f x| \leq \operatorname{Norm}(\mathfrak{L} p M) f\)
using \(\left\langle f \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\) L-infinity-AE-bound unfolding \(\langle p=\infty\rangle\) by auto
then have \(*: A E x\) in \(M .|f x * g x| \leq|\operatorname{Norm}(\mathfrak{L} p M) f * g x|\)
unfolding abs-mult using Norm-nonneg[of \(\mathfrak{L} p M f]\) mult-right-mono by fastforce
have \(* *:(\lambda x . \operatorname{Norm}(\mathfrak{L} p M) f * g x) \in\) space \(_{N}(\mathfrak{L} r M)\)
using space \(N\)-cmult \(\left[O F\left\langle g \in\right.\right.\) space \(\left.\left._{N}(\mathfrak{L} q M)\right\rangle\right]\) unfolding \(\langle q=r\rangle\) scaleR-fun-def by simp
have \(* * *: \operatorname{Norm}(\mathfrak{L} r M)(\lambda x . \operatorname{Norm}(\mathfrak{L} p M) f * g x)=\operatorname{Norm}(\mathfrak{L} p M) f *\) \(\operatorname{Norm}(\mathfrak{L} q M) g\)
using Norm-cmult[of \(\mathfrak{L} r M]\) unfolding \(\langle q=r\rangle\) scale \(R\)-fun-def by auto
then show? ?hesis
using Lp-domination[of \(\lambda x . f x * g x M \lambda x\). Norm \((\mathfrak{L} p M) f * g x r]\)
unfolding \(\langle q=r\rangle\)
using \(* * * * * *\) by auto
next
case 2
then have \(p=r\) using \(\langle 1 / p+1 / q=1 / r\rangle\)
by (metis add.right-neutral ennreal-divide-top infinity-ennreal-def one-divide-one-divide-ennreal)
have \(A E x\) in \(M .|g x| \leq \operatorname{Norm}(\mathfrak{L} q M) g\)
using \(\left\langle g \in\right.\) space \(\left._{N}(\mathfrak{L} q M)\right\rangle\) L-infinity-AE-bound unfolding \(\langle q=\infty\rangle\) by auto
then have \(*: A E x\) in \(M .|f x * g x| \leq|\operatorname{Norm}(\mathfrak{L} q M) g * f x|\)
apply (simp only: mult.commute[of Norm \((\mathfrak{L} q M) g-])\)
unfolding abs-mult using mult-left-mono Norm-nonneg[of \(\mathfrak{L} q M g]\) by fastforce
have \(* *:(\lambda x . \operatorname{Norm}(\mathfrak{L} q M) g * f x) \in \operatorname{space}_{N}(\mathfrak{L} r M)\)
using spaceN-cmult \(\left[O F\left\langle f \in\right.\right.\) space \(\left.\left._{N}(\mathfrak{L} p M)\right\rangle\right]\) unfolding \(\langle p=r\rangle\) scaleR-fun-def by simp
have \(* * *: \operatorname{Norm}(\mathfrak{L} r M)(\lambda x . \operatorname{Norm}(\mathfrak{L} q M) g * f x)=\operatorname{Norm}(\mathfrak{L} p M) f *\) Norm \((\mathfrak{L} q M) g\)
using Norm-cmult[of \(\mathfrak{L} r M\) ] unfolding \(\langle p=r\rangle\) scaleR-fun-def by auto
then show ?thesis
using Lp-domination[of \(\lambda x . f x * g x M \lambda x . \operatorname{Norm}(\mathfrak{L} q M) g * f x r]\) unfolding \(\langle p=r\rangle\)
using \(* * * * * *\) by auto
next
case 3
obtain \(p^{2}\) where \(p=\) ennreal p2 p2 \(>0\)
using enn2real-positive-iff \([\) of \(p] 3\langle p>0\rangle\) by (cases \(p\) ) auto
obtain \(q 2\) where \(q=\) ennreal \(q 2 q 2>0\)
using enn2real-positive-iff \([\) of \(q] 3\langle q>0\rangle\) by (cases q) auto
have \(\operatorname{ennreal}(1 / r\) 2) \(=1 / r\)
using \(\langle r=\) ennreal r2〉 \(\langle r 2>0\rangle\) divide-ennreal zero-le-one by fastforce
also have \(\ldots=1 / p+1 / q\) using assms by auto
also have \(\ldots=\operatorname{ennreal}(1 / p 2+1 / q 2)\) using \(\langle p=\) ennreal \(p 2\rangle\langle p 2>0\rangle\langle q\) \(=\) ennreal \(q 2\rangle\langle q 2>0\rangle\)
apply (simp only: divide-ennreal ennreal-1[symmetric]) using ennreal-plus[of 1/p2 1/q2, symmetric] by auto
finally have \(*: 1 / r 2=1 / p 2+1 / q^{2}\)
using ennreal-inj \(\langle p 2>0\rangle\langle q 2>0\rangle\langle r 2>0\rangle\) by (metis divide-pos-pos ennreal-less-zero-iff le-less zero-less-one)
define \(P\) where \(P=p 2 / r 2\)
define \(Q\) where \(Q=q^{2} / r^{2}\)
have \([\) simp \(]: P>0 Q>0\) and \(1 / P+1 / Q=1\)
using \(\langle p 2>0\rangle\langle q 2>0\rangle\langle r 2>0\rangle *\) unfolding \(P\)-def \(Q\)-def by (auto simp add: divide-simps algebra-simps)
have Pa: (|z| powr r2) powr \(P=|z|\) powr p2 for \(z\)
unfolding \(P\)-def powr-powr using \(\langle r 2>0\rangle\) by auto
have Qa: (|z| powr r2) powr \(Q=|z|\) powr q2 for \(z\)
unfolding \(Q\)-def powr-powr using \(\langle r 2>0\rangle\) by auto
have \(*\) : integrable \(M(\lambda x .|f x|\) powr r2 \(*|g x|\) powr r2)
apply (rule Holder-inequality[OF \(\langle P>0\rangle\langle Q>0\rangle\langle 1 / P+1 / Q=1\rangle]\), auto simp add: Pa Qa)
using \(\left\langle f \in\right.\) space \(\left._{N}(\mathfrak{L} p M)\right\rangle\) unfolding \(\langle p=\) ennreal \(p 2\rangle\) using Lp-space \([O F\) \(\langle p 2>0\rangle\) ] apply auto
using \(\left\langle g \in\right.\) space \(\left._{N}(\mathfrak{L} q M)\right\rangle\) unfolding \(\langle q=\) ennreal \(q 2\rangle\) using \(L p\)-space \([O F\) \(\langle q 2>0\rangle\) ] by auto
have \((\lambda x . f x * g x) \in \operatorname{space}_{N}(\mathfrak{L} r M)\)
unfolding \(\langle r=\) ennreal \(r 2\rangle\) using \(L p\)-space \([O F\langle r 2>0\rangle\), of \(M]\) by (auto simp add: * abs-mult powr-mult)
have \(\operatorname{Norm}(\mathfrak{L} r M)(\lambda x . f x * g x)=\left(\int x .|f x * g x|\right.\) powr r2 \(\left.\partial M\right)\) powr (1/r2)
unfolding \(\langle r=\) ennreal \(r\) 2 \(\rangle\) using \(L p-N o r m[O F\langle r 2>0\rangle\), of \(-M]\) by auto
also have \(\ldots=a b s\left(\int x .|f x|\right.\) powr r2 * \(|g x|\) powr r2 \(\partial M\) ) powr (1/r2)
by (auto simp add: powr-mult abs-mult)
also have \(\ldots \leq\left(\left(\int x .||f x|\right.\right.\) powr r2 \(|\) powr \(\left.P \partial M\right)\) powr \((1 / P) *\left(\int x .| | g\right.\) \(x \mid\) powr r2 | powr \(Q \partial M)\) powr \((1 / Q))\) powr \((1 /\) r2)
apply (rule powr-mono2, simp add: 〈r2 \(>0\rangle\) less-imp-le, simp)
apply (rule Holder-inequality[OF \(\langle P>0\rangle\langle Q>0\rangle\langle 1 / P+1 / Q=1\rangle]\), auto simp add: Pa Qa)
using \(\left\langle f \in\right.\) space \(\left._{N}(\mathfrak{L} p M)\right\rangle\) unfolding \(\langle p=\) ennreal \(p 2\rangle\) using \(L p\)-space \([O F\) \(\langle p 2>0\rangle\) ] apply auto
using \(\left\langle g \in \operatorname{space}_{N}(\mathfrak{L} q M)\right\rangle\) unfolding \(\langle q=\) ennreal \(q 2\rangle\) using Lp-space \([O F\) \(\langle q 2>0\rangle]\) by auto
also have \(\ldots=\left(\int x .|f x|\right.\) powr p2 \(\left.\partial M\right)\) powr \((1 / p 2) *\left(\int x .|g x|\right.\) powr q2 \(\partial M)\) powr (1/q2)
apply (auto simp add: powr-mult powr-powr) unfolding \(P\)-def \(Q\)-def using \(\langle r 2>0\rangle\) by auto
also have \(\ldots=\operatorname{Norm}(\mathfrak{L} p M) f * \operatorname{Norm}(\mathfrak{L} q M) g\)
unfolding \(\langle p=\) ennreal \(p 2\rangle\langle q=\) ennreal \(q 2\rangle\)
using \(L p-\operatorname{Norm}[O F\langle p 2>0\rangle\), of \(-M] L p-N o r m[O F\langle q 2>0\rangle\), of \(-M]\) by auto
finally show ?thesis using \(\left\langle(\lambda x . f x * g x) \in \operatorname{space}_{N}(\mathfrak{L} r M)\right\rangle\) by auto qed
next
case PInf
then have \(p=\infty q=r\) using \(\langle 1 / p+1 / q=1 / r\rangle\)
by (metis add-eq-0-iff-both-eq-0 ennreal-divide-eq-0-iff infinity-ennreal-def not-one-le-zero order.order-iff-strict)+
have \(A E x\) in \(M .|f x| \leq \operatorname{Norm}(\mathfrak{L} p M) f\)
using \(\left\langle f \in\right.\) space \(\left._{N}(\mathfrak{L} p M)\right\rangle\) L-infinity-AE-bound unfolding \(\langle p=\infty\rangle\) by auto
then have \(*: A E x\) in \(M .|f x * g x| \leq|\operatorname{Norm}(\mathfrak{L} p M) f * g x|\)
unfolding abs-mult using Norm-nonneg[of \(\mathfrak{L} p M f]\) mult-right-mono by fastforce
have \(* *:(\lambda x . \operatorname{Norm}(\mathfrak{L} p M) f * g x) \in \operatorname{space}_{N}(\mathfrak{L} r M)\)
using spaceN-cmult \(\left[O F\left\langle g \in\right.\right.\) space \(\left.\left._{N}(\mathfrak{L} q M)\right\rangle\right]\) unfolding \(\langle q=r\rangle\) scale \(R\)-fun-def by simp
have \(* * *: \operatorname{Norm}(\mathfrak{L} r M)(\lambda x . \operatorname{Norm}(\mathfrak{L} p M) f * g x)=\operatorname{Norm}(\mathfrak{L} p M) f *\) \(\operatorname{Norm}(\mathfrak{L} q M) g\)
using Norm-cmult[of \(\mathfrak{L} r M]\) unfolding \(\langle q=r\rangle\) scale \(R\)-fun-def by auto
then show ?thesis
using Lp-domination[of \(\lambda x . f x * g x M \lambda x . \operatorname{Norm}(\mathfrak{L} p M) f * g x r]\) unfolding \(\langle q=r\rangle\)
using \(* * * * * *\) by auto
qed
then show \((\lambda x . f x * g x) \in \operatorname{space}_{N}(\mathfrak{L} r M)\)
\(\operatorname{Norm}(\mathfrak{L} r M)(\lambda x . f x * g x) \leq \operatorname{Norm}(\mathfrak{L} p M) f * \operatorname{Norm}(\mathfrak{L} q M) g\)
by auto
qed
The previous theorem admits an eNorm version in which one does not assume a priori that the functions under consideration belong to \(L^{p}\) or \(L^{q}\).
```

theorem Lp-Lq-emult:
fixes $p$ q $r$ ::ennreal
assumes $1 / p+1 / q=1 / r$
$f \in$ borel-measurable $M g \in$ borel-measurable $M$
shows $\operatorname{eNorm}(\mathfrak{L} r M)(\lambda x . f x * g x) \leq \operatorname{eNorm}(\mathfrak{L} p M) f * \operatorname{eNorm}(\mathfrak{L} q M) g$
proof (cases $r=0$ )
case True
then have $e \operatorname{Norm}(\mathfrak{L} r M)(\lambda x . f x * g x)=0$
using assms by (simp add: L-zero(1))
then show ?thesis by auto
next
case False
then have $r>0$ using not-gr-zero by blast
then have $p>0 q>0$ using $\langle 1 / p+1 / q=1 / r\rangle$
by (metis ennreal-add-eq-top ennreal-divide-eq-top-iff ennreal-top-neq-one gr-zeroI
zero-neq-one)+
then have $Z$ : zero-space $_{N}(\mathfrak{L} p M)=\{f \in$ borel-measurable $M$. $A E x$ in $M . f x$
$=0\}$
zero-space $_{N}(\mathfrak{L} q M)=\{f \in$ borel-measurable M. AE x in M. $f x=0\}$
zero-space $_{N}(\mathfrak{L} r M)=\{f \in$ borel-measurable $M . A E x$ in $M . f x=0\}$
using $\langle r>0\rangle$ Lp-infinity-zero-space by auto
have [measurable]: $(\lambda x . f x * g x) \in$ borel-measurable $M$ using assms by auto
consider eNorm $(\mathfrak{L} p M) f=0 \vee e \operatorname{Norm}(\mathfrak{L} q M) g=0$
$\mid(e \operatorname{Norm}(\mathfrak{L} p M) f>0 \wedge \operatorname{eNorm}(\mathfrak{L} q M) g=\infty) \vee(e \operatorname{Norm}(\mathfrak{L} p M) f$
$=\infty \wedge e \operatorname{Norm}(\mathfrak{L} q M) g>0)$
$\mid \operatorname{eNorm}(\mathfrak{L} p M) f<\infty \wedge \operatorname{eNorm}(\mathfrak{L} q M) g<\infty$
using less-top by fastforce
then show ?thesis
proof (cases)
case 1
then have $(A E x$ in $M . f x=0) \vee(A E x$ in $M . g x=0)$ using $Z$ unfolding
zero-space $_{N}$-def by auto

```
```

    then have AEx in M. fx*gx=0 by auto
    then have eNorm ({)rM)}(\lambdax.fx*gx)=0 using Z unfolding zero-space N-def
    by auto
then show ?thesis by simp
next
case 2
then have eNorm (\mathfrak{L}pM)f*eNorm ({)qM)g=\infty using ennreal-mult-eq-top-iff
by force
then show ?thesis by auto
next
case 3
then have *:f\in space}\mp@subsup{N}{}{({LpM)g\in space}N(\mathfrak{L}qM)\mathrm{ unfolding space}\mp@subsup{N}{N}{}\mathrm{ -def
by auto
then have (\lambdax.fx*gx)\in \mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}rM)\mathbf{using}\operatorname{Lp-Lq-mult(1)[OF assms(1)]}
by auto
then show ?thesis
using Lp-Lq-mult(2)[OF assms(1) *] by (simp add: eNorm-Norm * en-
nreal-mult'[symmetric])
qed
qed
lemma Lp-Lq-duality-bound:
fixes p q::ennreal
assumes 1/p+1/q=1
f\in\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}pM)
g\in \mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}qM)
shows integrable M (\lambdax.fx*g x)
abs(\intx.fx*gx\partialM)\leqNorm ({LpM)f*Norm ({LqM)g
proof -
have (\lambdax.fx*gx)\in \mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}1M)
apply (rule Lp-Lq-mult[OF-\langlef\in space}\mp@subsup{N}{N}{}(\mathfrak{L}pM)\rangle\langleg\in\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}qM)\rangle]
using <1/p+1/q=1> by auto
then show integrable M ( \lambdax.fx*g x) using L1-space by auto
have abs(\intx.fx*gx\partialM)\leqNorm({L 1 M) (\lambdax.fx*gx) using L1-int-ineq
by auto
also have ... \leqNorm ({ p M) f*Norm ( }\mathfrak{L}qM)
apply (rule Lp-Lq-mult[OF - <f\in space}\mp@subsup{N}{N}{}(\mathfrak{L}pM)\rangle\langleg\in\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}qM)\rangle]
using <1/p+1/q=1\rangle by auto
finally show abs(\intx.fx*gx\partialM)\leqNorm ({ p M)f*Norm ({ { qM)g}\mathrm{ by
simp
qed

```

The next theorem asserts that the norm of an \(L^{p}\) function \(f\) can be obtained by estimating the integrals of \(f g\) over all \(L^{q}\) functions \(g\), where \(1 / p+1 / q=1\). When \(p=\infty\), it is necessary to assume that the space is sigma-finite: for instance, if the space is one single atom of infinite mass, then there is no nonzero \(L^{1}\) function, so taking for \(f\) the constant function equal to 1 , it has \(L^{\infty}\) norm equal to 1 , but \(\int f g=0\) for all \(L^{1}\) function \(g\).
```

theorem Lp-Lq-duality:
fixes p q::ennreal
assumes f\in space N}(\mathfrak{L}pM
1/p+1/q=1
p=\infty\Longrightarrow sigma-finite-measure M
shows bdd-above ((\lambdag. (\intx.fx*g x \partialM))`{g\in space }\mp@subsup{N}{N}{}(\mathfrak{L}qM).Norm (\mathfrak{L}
M) g < 1})
Norm (\mathfrak{L p M) f=(SUP g\in{g\in space}
(\intx.fx*gx\partialM))
proof -
have [measurable]: f \in borel-measurable M using Lp-measurable[OF assms(1)]
by auto
have B:(\intx.fx*gx\partialM)\leqNorm ({ p M)f if g\in{g\in space N ({NqM).
Norm ({ q M) g\leq1} for g
proof -
have g:g\in space}N(\mathfrak{L}qM)\operatorname{Norm}(\mathfrak{L}qM)g\leq1 using that by aut
have (\intx.fx*gx\partialM)\leqabs(\intx.fx*gx\partialM) by auto
also have ... \leqNorm ( L pM)f*Norm ({LqM)g
using Lp-Lq-duality-bound(2)[OF <1/p+1/q=1\rangle\langlef\in \mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}pM)\rangle
g(1)] by auto
also have .. \leqNorm ({ p M)f
using g(2) Norm-nonneg[of \mathfrak{L p M f] mult-left-le by blast}
finally show (\intx.fx*gx\partialM)\leqNorm ({ L pM)f by simp
qed

```

```

qM) g < 1})
by (meson bdd-aboveI2)
show Norm ({L p M) f=(SUP g\in{g\in space}\mp@subsup{N}{N}{}(\mathfrak{L}qM).Norm ({LqM)g\leq
1}. (\intx.fx*gx\partialM))
proof (rule antisym)
show (SUP g\in{g\in space N (\mathfrak{LqM}).Norm ({ qqM)g\leq1}. \intx.fx*gx
\partialM) \leqNorm (L p M)f
by (rule cSUP-least, auto, rule exI[of - 0], auto simp add: B)
have p\geq1 using conjugate-exponent-ennrealI(1)[OF<1/p+1/q=1\rangle] by
simp
show Norm (\mathfrak{L p M) f \leq (SUP g\in{g\in space}
1}.(\intx.fx*gx\partialM))
using <p\geq1> proof (cases rule: Lp-cases-1-PInf)
case PInf
then have f\in\mp@subsup{\mathrm{ space }}{N}{}(\mathfrak{L}\inftyM)
using <f\in space}\mp@subsup{N}{N}{}(\mathfrak{L}pM)\rangle\mathrm{ by simp
have q=1 using }\langle1/p+1/q=1\rangle\langlep=\infty> by (simp add: divide-eq-1-ennreal
have c}\leq(SUPg\in{g\in\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}qM).Norm({\mathfrak{L}qM)g\leq1}.(\intx.fx*
gx\partialM)) if c<Norm ({ p M) f for c
proof (cases c<<0)
case True
then have c\leq(\intx.fx*0x\partialM) by auto

```
also have \(\ldots \leq\left(S U P g \in\left\{g \in \operatorname{space}_{N}(\mathfrak{L} q M) . \operatorname{Norm}(\mathfrak{L} q M) g \leq 1\right\} .\left(\int x\right.\right.\). \(f x * g x \partial M)\) ）
apply（rule cSUP－upper，auto simp add：zero－fun－def［symmetric］）using \(B\) by（meson bdd－aboveI2）
finally show ？thesis by simp
next
case False
then have ennreal \(c<\operatorname{eNorm}(\mathfrak{L} \infty M) f\) using eNorm－Norm \(\left[\right.\) OF \(\left\langle f \in\right.\) space \(\left.\left._{N}(\mathfrak{L} p \quad M)\right\rangle\right]\) that ennreal－less－iff unfolding \(\langle p=\infty\) 〉 by auto
then have \(*\) ：emeasure \(M\{x \in\) space \(M .|f x|>c\}>0\) using L－infinity－pos－measure \([\) of \(f M c]\) by auto
obtain \(A\) where［measurable］：\(\bigwedge(n:: n a t) . A n \in\) sets \(M\) and \((\bigcup i . A i)=\) space \(M \bigwedge i\) ．emeasure \(M(A i) \neq \infty\)
using sigma－finite－measure．sigma－finite \([O F<p=\infty \Longrightarrow\) sigma－finite－measure \(M\rangle[O F\langle p=\infty\rangle]]\) by（metis UNIV－I sets－range）
define \(Y\) where \(Y=(\lambda n:: n a t .\{x \in A n .|f x|>c\})\)
have［measurable］：\(Y n \in\) sets \(M\) for \(n\) unfolding \(Y\)－def by auto
have \(\{x \in\) space \(M .|f x|>c\}=(\bigcup n\) ．\(Y n)\) unfolding \(Y\)－def using \(\langle(\bigcup i\) ． A i）\(=\) space \(M>\) by auto
then have emeasure \(M(\bigcup n\) ．\(Y n)>0\) using＊by auto
then obtain \(n\) where emeasure \(M(Y n)>0\)
using emeasure－pos－unionE［of \(Y, O F\langle\bigwedge n\) ．\(Y n \in\) sets \(M>]\) by auto
have emeasure \(M(Y n) \leq\) emeasure \(M(A n)\) apply（rule emeasure－mono） unfolding \(Y\)－def by auto
then have emeasure \(M(Y n) \neq \infty\) using＜emeasure \(M(A n) \neq \infty\) 〉
by（metis infinity－ennreal－def neq－top－trans）
then have measure \(M(Y n)>0\) using \(\langle\) emeasure \(M(Y n)>0\rangle\) unfolding measure－def
by（simp add：enn2real－positive－iff top．not－eq－extremum）
have \(|f x| \geq c\) if \(x \in Y n\) for \(x\) using that less－imp－le unfolding \(Y\)－def by auto
define \(g\) where \(g=(\lambda x\) ．indicator \((Y n) x * \operatorname{sgn}(f x)) /{ }_{R}\) measure \(M(Y\) n）
have \(g \in \operatorname{space}_{N}(\mathfrak{L} 1 M)\)
apply（rule Lp－domination［of－－indicator \((Y n) / R\) measure \(M(Y n)])\) unfolding \(g\)－def
using L1－indicator＇ ［OF〈Yn sets \(M\rangle\langle\) emeasure \(M(Y n) \neq \infty\rangle]\) by （auto simp add：abs－mult indicator－def abs－sgn－eq）
have \(\operatorname{Norm}(\mathfrak{L} 1 M) g=\operatorname{Norm}(\mathfrak{L} 1 M)(\lambda x\) ．indicator \((Y n) x * \operatorname{sgn}(f x))\) ／abs（measure \(M(Y n)\) ）
unfolding \(g\)－def Norm－cmult by（simp add：divide－inverse）
also have \(\ldots \leq \operatorname{Norm}(\mathfrak{L} 1 M)(\) indicator \((Y n)) / \operatorname{abs}(\) measure \(M(Y n))\)
using \(\langle\) measure \(M(Y n)>0\rangle\) apply（auto simp add：divide－simps）apply （rule Lp－domination）
using L1－indicator \({ }^{\prime}[\) OF \(\langle Y n \in\) sets \(M\rangle\langle e m e a s u r e ~ M(Y n) \neq \infty\rangle]\) by （auto simp add：abs－mult indicator－def abs－sgn－eq）
also have \(\ldots=\) measure \(M(Y n) /\) abs（measure \(M(Y n))\)
using L1-indicator \({ }^{[ }[O F\langle Y n \in\) sets \(M\rangle\langle\) emeasure \(M(Y n) \neq \infty\rangle]\) by (auto simp add: abs-mult indicator-def abs-sgn-eq)
also have \(\ldots=1\) using \(\langle\) measure \(M(Y n)>0\) 〉 by auto
finally have \(\operatorname{Norm}(\mathfrak{L} 1 M) g \leq 1\) by \(\operatorname{simp}\)
have \(c *\) measure \(M(Y n)=\left(\int x . c *\right.\) indicator \(\left.(Y n) x \partial M\right)\)
using <measure \(M(Y n)>0\rangle\langle\) emeasure \(M(Y n) \neq \infty\) by auto
also have \(\ldots \leq\left(\int x .|f x| *\right.\) indicator \(\left.(Y n) x \partial M\right)\)
apply (rule integral-mono)
using <emeasure \(M(Y n) \neq \infty\rangle\langle 0<\) Sigma-Algebra.measure \(M(Y n)\rangle\) not-integrable-integral-eq apply fastforce
apply (rule Bochner-Integration.integrable-bound \([\) of \(-\lambda x\). Norm \((\mathfrak{L} \infty M)\)
\(f *\) indicator \((Y n) x]\) )
using <emeasure \(M(Y n) \neq \infty\rangle\langle 0<\) Sigma-Algebra.measure \(M(Y n)\rangle\) not-integrable-integral-eq apply fastforce
using L-infinity-AE-bound \(\left[O F\left\langle f \in\right.\right.\) space \(\left.\left._{N}(\mathfrak{L} \infty M)\right\rangle\right]\) by (auto simp add: indicator-def \(Y\)-def)
finally have \(c \leq\left(\int x .|f x| *\right.\) indicator \(\left.(Y n) x \partial M\right) /\) measure \(M(Y n)\)
using <measure \(M(Y n)>0\rangle\) by (auto simp add: divide-simps)
also have \(\ldots=\left(\int x . f x *\right.\) indicator \((Y n) x * \operatorname{sgn}(f x) /\) measure \(M(Y n)\) \(\partial M)\)
using <measure \(M(Y n)>0\rangle\) by (simp add: abs-sgn mult.commute mult.left-commute)
also have \(\ldots=\left(\int x . f x * g x \partial M\right)\)
unfolding divide-inverse \(g\)-def divideR-apply by (auto simp add: alge-bra-simps)
also have \(\ldots \leq\left(S U P g \in\left\{g \in\right.\right.\) space \(\left._{N}(\mathfrak{L} q M) . \operatorname{Norm}(\mathfrak{L} q M) g \leq 1\right\} .\left(\int x\right.\). \(f x * g x \partial M))\)
unfolding \(\langle q=1\rangle\) apply (rule cSUP-upper, auto)
using \(\left\langle g \in \operatorname{space}_{N}(\mathfrak{L} 1 M)\right\rangle\langle\operatorname{Norm}(\mathfrak{L} 1 M) g \leq 1\rangle\) apply auto using \(B\langle p=\infty\rangle\langle q=1\rangle\) by (meson bdd-aboveI2)
finally show ?thesis by simp
qed
then show ?thesis using dense-le by auto
next
case one
then have \(q=\infty\) using \(\langle 1 / p+1 / q=1\rangle\) by simp
define \(g\) where \(g=(\lambda x\). sgn \((f x))\)
have [measurable]: \(g \in \operatorname{space}_{N}(\mathfrak{L} \infty M)\)
apply (rule L-infinity-I[of \(g\) M 1]) unfolding \(g\)-def by (auto simp add: \(a b s-s g n-e q\) )
have \(\operatorname{Norm}(\mathfrak{L} \infty M) g \leq 1\)
apply (rule L-infinity-I[of \(g\) M 1]) unfolding \(g\)-def by (auto simp add: \(a b s-s g n-e q)\)
have \(\operatorname{Norm}(\mathfrak{L} p M) f=\left(\int x .|f x| \partial M\right)\)
unfolding \(\langle p=1\rangle\) apply \((\) rule \(L 1-D(3))\) using \(\left\langle f \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\)
unfolding \(\langle p=1\rangle\) by auto
also have \(\ldots=\left(\int x . f x * g x \partial M\right)\)
unfolding \(g\)-def by (simp add: abs-sgn)
also have \(\ldots \leq\left(S U P g \in\left\{g \in \operatorname{space}_{N}(\mathfrak{L} q M) . \operatorname{Norm}(\mathfrak{L} q M) g \leq 1\right\} .\left(\int x\right.\right.\). \(f x * g x \partial M)\) )
unfolding \(\langle q=\infty\) apply (rule cSUP-upper, auto)
using \(\left\langle g \in \operatorname{space}_{N}(\mathfrak{L} \infty M)\right\rangle\langle\operatorname{Norm}(\mathfrak{L} \infty M) g \leq 1\rangle\) apply auto
using \(B\langle q=\infty\rangle\) by fastforce
finally show ?thesis by simp

\section*{next}
case ( \(g r p 2\) )
then have \(p 2>0\) by simp
have \(f \in \operatorname{space}_{N}(\mathfrak{L} p 2 M)\) using \(\left\langle f \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\langle p=\) ennreal \(p 2\rangle\) by auto
define \(q 2\) where \(q 2=\) conjugate-exponent \(p 2\)
have \(q^{2}>1 q^{2}>0\) using conjugate-exponent-real(2)[OF \(\left.\langle p 2>1\rangle\right]\) unfolding \(q 2-d e f\) by auto
have \(q=\) ennreal \(q 2\)
unfolding q2-def conjugate-exponent-real-ennreal[OF \(\langle p 2>1\rangle\), symmetric] \(\langle p=\) ennreal \(p\rangle\rangle[\) symmetric \(]\)
using conjugate-exponent-ennreal-iff \([O F\langle p \geq 1\rangle]\langle 1 / p+1 / q=1\rangle\) by auto
show ?thesis
proof (cases Norm ( \(\mathfrak{L} p M) f=0\) )
case True
then have \(\operatorname{Norm}(\mathfrak{L} p M) f \leq\left(\int x . f x * 0 x \partial M\right)\) by auto
also have \(\ldots \leq\left(S U P g \in\left\{g \in \operatorname{space}_{N}(\mathfrak{L} q M) . \operatorname{Norm}(\mathfrak{L} q M) g \leq 1\right\} .\left(\int x\right.\right.\). \(f x * g x \partial M))\)
apply (rule cSUP-upper, auto simp add: zero-fun-def[symmetric]) using \(B\) by (meson bdd-aboveI2)
finally show ?thesis by simp
next
case False
then have \(\operatorname{Norm}(\mathfrak{L} p \mathfrak{2} M) f>0\)
unfolding \(\langle p=\) ennreal p2 \(\langle\) using Norm-nonneg \([\) of \(\mathfrak{L} p 2 M f]\) by linarith
define \(h\) where \(h=(\lambda x . \operatorname{sgn}(f x) *|f x| \operatorname{powr}(p 2-1))\)
have [measurable]: \(h \in\) borel-measurable \(M\) unfolding \(h\)-def by auto
have \(\left(\int{ }^{+} x .|h x|\right.\) powr q2 \(\left.\partial M\right)=\left(\int{ }^{+}\right.\)x. \((|f x| \operatorname{powr}(p 2-1))\) powr q2 \(\left.\partial M\right)\)
unfolding \(h\)-def by (rule nn-integral-cong, auto simp add: abs-mult \(a b s-s g n-e q)\)
also have \(\ldots=\left(\int^{+} x .|f x|\right.\) powr p2 \(\left.\partial M\right)\)
unfolding powr-powr q2-def using conjugate-exponent-real(4)[OF \(\langle\) p2 \(>\) 1)] by auto
also have \(\ldots=(\operatorname{Norm}(\mathfrak{L} p 2 M) f)\) powr p2
apply (subst Lp-Norm(2), auto simp add: \(\langle p 2>0\rangle\) )
by (rule nn-integral-eq-integral, auto simp add: Lp-D(2)[OF \(\langle p 2>0\rangle\langle f\) \(\left.\left.\left.\in \operatorname{space}_{N}(\mathfrak{L} p 2 M)\right\rangle\right]\right)\)
finally have \(*:\left(\int^{+} x .|h x|\right.\) powr q2 \(\left.\partial M\right)=(\operatorname{Norm}(\mathfrak{L}\) p2 \(M) f)\) powr p2 by \(\operatorname{simp}\)
have integrable \(M(\lambda x .|h x|\) powr q2)
apply (rule integrableI-bounded, auto) using * by auto
```

    then have (\int x. |h x| powr q2 \partialM) = ( \int + }x.|hx| powr q2 \partialM)
    by (rule nn-integral-eq-integral[symmetric], auto)
    then have **:(\intx. |h x| powr q2 \partialM) =(Norm ({ p2 M) f) powr p2
    using * by auto
define g where g=( \lambdax.hx / (Norm ({ p2 M) f) powr (p2 / q2))
have [measurable]: g}\in\mathrm{ borel-measurable }M\mathrm{ unfolding g-def by auto
have intg: integrable M ( }\lambdax.|gx| powr q2
unfolding g-def using<Norm ({ p2 M) f>0〉\langleq2 > 1` apply (simp add: abs-mult powr-divide powr-powr)             using <integrable M ( }\lambdax.|hx| powr q2)> integrable-divide-zero by blas     have g \in space N     have (\intx. |gx| powr q2 \partialM)=1             unfolding g-def using <Norm ({ p2 M) f>0\rangle\langleq2 > 1\rangle by (simp add: abs-mult powr-divide powr-powr **)     then have Norm ({~q2 M) g=1         apply (subst Lp-D[OF<q2 > 0`]) using <g\in space N
have (\intx.fx*gx \partialM) =(\intx.fx*\operatorname{sgn}(fx)*|fx| powr (p2 - 1) /
(Norm ({ p2 M) f) powr (p2 / q2) \partialM)
unfolding g-def h-def by (simp add: mult.assoc)
also have ... = (\intx. |fx|*|fx| powr (p2-1) \partialM)/(Norm ({ p2 M)f)
powr (p2 / q2)
by (auto simp add: abs-sgn)
also have ... = (\intx. |f x | powr p2 \partialM) / (Norm ({ p2 M) f) powr (p2 /
q2)
by (subst powr-mult-base, auto)
also have ... = (Norm ({ p2 M)f) powr p2 / (Norm ({ p2 M) f) powr
(p2 / q2)
by (subst Lp-Norm(2)[OF<p2> 0>], auto)
also have ... =(Norm (L p2 M) f) powr (p2 - p2/q2)
by (simp add: powr-diff [symmetric] )
also have ... = Norm (L p2 M)f
unfolding q2-def using conjugate-exponent-real(5)[OF<p2 > 1〉] by auto
finally have Norm ({)pM)f=(\intx.fx*gx\partialM)
unfolding <p = ennreal p2\rangle by simp
also have ···.\leq(SUP g\in{g\in \mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}qM).Norm ({~qM)g\leq1}.(\intx.
fx*g x \partialM))
unfolding <q = ennreal q2> apply (rule cSUP-upper, auto)
using <g\in space}\mp@subsup{N}{N}{}(\mathfrak{L}q\mathcal{2}M)\rangle\langleNorm (\mathfrak{L q2 M) g= 1` apply auto
using B<q= ennreal q2\rangle by fastforce
finally show ?thesis by simp
qed
qed
qed
qed

```

The previous theorem admits a version in which one does not assume a priori that the function under consideration belongs to \(L^{p}\). This gives an
efficient criterion to check if a function is indeed in \(L^{p}\) ．In this case，it is always necessary to assume that the measure is sigma－finite．
Note that，in the statement，the Bochner integral \(\int f g\) vanishes by definition if \(f g\) is not integrable．Hence，the statement really says that the eNorm can be estimated using functions \(g\) for which \(f g\) is integrable．It is precisely the construction of such functions \(g\) that requires the space to be sigma－finite．
```

theorem Lp-Lq-duality':
fixes $p$ ::ennreal
assumes $1 / p+1 / q=1$
sigma-finite-measure $M$
and [measurable]: $f \in$ borel-measurable $M$
shows $\operatorname{eNorm}(\mathfrak{L} p M) f=\left(S U P g \in\left\{g \in \operatorname{space}_{N}(\mathfrak{L} q M) . \operatorname{Norm}(\mathfrak{L} q M) g \leq\right.\right.$
1\}. ennreal ( $\left.\left.\int x . f x * g x \partial M\right)\right)$
proof (cases eNorm ( $\mathfrak{L} p M) f \neq \infty)$
case True
then have $f \in$ space $_{N}(\mathfrak{L} p M)$ unfolding space $_{N}$-def by (simp add: top.not-eq-extremum)
show ?thesis
unfolding eNorm-Norm $\left[O F\left\langle f \in\right.\right.$ space $\left.\left._{N}(\mathfrak{L} p M)\right\rangle\right] \operatorname{Lp-Lq-duality[OF}\langle f \in$
space $\left._{N}(\mathfrak{L} p M)\right\rangle\langle 1 / p+1 / q=1\rangle\langle$ sigma-finite-measure $\left.M\rangle\right]$
apply (rule SUP-real-ennreal $[$ symmetric], auto, rule exI $[o f-0]$, auto)
by $\left(\right.$ rule $L p$-Lq-duality $\left[O F\left\langle f \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\langle 1 / p+1 / q=1\rangle\langle\right.$ sigma-finite-measure
M>])
next
case False
have $B: \exists g \in\left\{g \in \operatorname{space}_{N}(\mathfrak{L} q M) . \operatorname{Norm}(\mathfrak{L} q M) g \leq 1\right\} .\left(\int x . f x * g x \partial M\right)$
$\geq C$ if $C<\infty$ for $C::$ ennreal
proof -
obtain $C r$ where $C=$ ennreal $C r C r \geq 0$ using 〈 $C<\infty$ 〉ennreal-cases
less-irrefl by auto
obtain $A$ where $A: \bigwedge n:: n a t . A n \in$ sets $M$ incseq $A(\bigcup n . A n)=$ space $M$
$\wedge n$. emeasure $M(A n) \neq \infty$
using sigma-finite-measure.sigma-finite-incseq[OF «sigma-finite-measure $M$ 〉]
by (metis range-subsetD)
define $Y$ where $Y=(\lambda n .\{x \in A n .|f x| \leq n\})$
have [measurable]: $\bigwedge n$. $Y n \in$ sets $M$ unfolding $Y$-def using 〈 $\bigwedge n:: n a t$. A $n$
$\in$ sets $M>$ by auto
have incseq $Y$
apply (rule incseq-SucI) unfolding $Y$-def using incseq-SucD $[O F\langle$ incseq $A\rangle]$
by auto
have $*: \exists N . \forall n \geq N . f x *$ indicator $(Y n) x=f x$ if $x \in$ space $M$ for $x$
proof -
obtain $n 0$ where $n 0: x \in A$ n0 using $\langle x \in$ space $M\rangle\langle(\bigcup n$. A n) $=$ space
$M>$ by auto
obtain $n 1::$ nat where $n 1:|f x| \leq n 1$ using real-arch-simple by blast
have $x \in Y(\max n 0 n 1)$
unfolding $Y$-def using $n 1$ apply auto
using $n 0$ 〈incseq $A$ 〉incseq-def max.cobounded1 by blast
then have $*: x \in Y n$ if $n \geq \max n 0 n 1$ for $n$

```
using 〈incseq \(Y\) 〉that incseq-def by blast show ?thesis by (rule exI[of - max n0 n1], auto simp add: *)
qed
have \(*:(\lambda n . f x *\) indicator \((Y n) x) \longrightarrow f x\) if \(x \in\) space \(M\) for \(x\) using \(*[\) OF that \(]\) unfolding eventually-sequentially[symmetric] by (simp add: tendsto-eventually)
have \(\liminf (\lambda n . \operatorname{eNorm}(\mathfrak{L} p M)(\lambda x . f x * \operatorname{indicator}(Y n) x)) \geq e \operatorname{Norm}(\mathfrak{L}\) \(p M) f\)
apply (rule Lp-AE-limit) using \(*\) by auto
then have liminf \((\lambda n . e \operatorname{Norm}(\mathfrak{L} p M)(\lambda x . f x * \operatorname{indicator}(Y n) x))>C r\) using False neq-top-trans by force
then have limsup \((\lambda n . \operatorname{eNorm}(\mathfrak{L} p M)(\lambda x . f x * \operatorname{indicator}(Y n) x))>C r\) using Liminf-le-Limsup less-le-trans trivial-limit-sequentially by blast
then obtain \(n\) where \(n\) : eNorm \((\mathfrak{L} p M)(\lambda x . f x *\) indicator \((Y n) x)>C r\) using Limsup-obtain by blast
have \((\lambda x . f x *\) indicator \((Y n) x) \in \operatorname{space}_{N}(\mathfrak{L} p M)\)
apply (rule Lp-bounded-bounded-support \([\) of - \(-n]\), auto)
unfolding \(Y\)-def indicator-def apply auto
by (metis (mono-tags, lifting) A(1) A(4) emeasure-mono infinity-ennreal-def mem-Collect-eq neq-top-trans subsetI)
have \(\operatorname{Norm}(\mathfrak{L} p M)(\lambda x . f x *\) indicator \((Y n) x)>C r\)
using \(n\) unfolding eNorm-Norm \(\left[O F\left\langle(\lambda x . f x *\right.\right.\) indicator \((Y n) x) \in\) space \(_{N}\) \((\mathfrak{L} p M)\rangle]\)
by (meson ennreal-leI not-le)
then have \(\left(S U P g \in\left\{g \in \operatorname{space}_{N}(\mathfrak{L} q M) . \operatorname{Norm}(\mathfrak{L} q M) g \leq 1\right\} .\left(\int x . f x *\right.\right.\) indicator \((Y n) x * g x \partial M))>C r\)
using Lp-Lq-duality(2) \(\left[\right.\) OF \(\left\langle(\lambda x . f x *\right.\) indicator \(\left.(Y n) x) \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\) \(\langle 1 / p+1 / q=1\rangle\langle\) sigma-finite-measure \(M\rangle\) ]
by auto
then have \(\exists g \in\left\{g \in \operatorname{space}_{N}(\mathfrak{L} q M) . \operatorname{Norm}(\mathfrak{L} q M) g \leq 1\right\} .\left(\int x . f x *\right.\)
indicator \((Y n) x * g x \partial M)>C r\)
apply (subst less-cSUP-iff[symmetric])
using Lp-Lq-duality \((1)\left[O F<(\lambda x . f x *\right.\) indicator \(\left.(Y n) x) \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\) \(\langle 1 / p+1 / q=1\rangle\langle\) sigma-finite-measure \(M\rangle\) ] apply auto by (rule exI[of - 0], auto)
then obtain \(g\) where \(g: g \in \operatorname{space}_{N}(\mathfrak{L} q M) \operatorname{Norm}(\mathfrak{L} q M) g \leq 1\left(\int x . f x\right.\) * indicator \((Y n) x * g x \partial M)>C r\)
by auto
then have [measurable]: \(g \in\) borel-measurable \(M\) using Lp-measurable by auto
define \(h\) where \(h=(\lambda x\). indicator \((Y n) x * g x)\)
have \(\operatorname{Norm}(\mathfrak{L} q M) h \leq \operatorname{Norm}(\mathfrak{L} q M) g\)
apply (rule Lp-domination \([o f--g]\) ) unfolding \(h\)-def indicator-def using \(\langle g\) \(\in \operatorname{space}_{N}(\mathfrak{L} q M)\) > by auto
then have \(a: \operatorname{Norm}(\mathfrak{L} q M) h \leq 1\) using \(\langle\operatorname{Norm}(\mathfrak{L} q M) g \leq 1\rangle\) by auto
have \(b: h \in\) space \(_{N}(\mathfrak{L} q M)\)
apply (rule Lp-domination \([o f-g]\) ) unfolding \(h\)-def indicator-def using \(\langle g\) \(\in\) space \(_{N}(\mathfrak{L} q M)\) > by auto
have \(\left(\int x . f x * h x \partial M\right)>C r\) unfolding \(h\)-def using \(g(3)\) by (auto simp
add: mult.assoc)
then have \(\left(\int x . f x * h x \partial M\right)>C\)
unfolding \(\langle C=\) ennreal \(C r\rangle\) using \(\langle C r \geq 0\rangle\) by (simp add: ennreal-less-iff)
then show ?thesis using \(a b\) by auto
qed
have \(\left(S U P g \in\left\{g \in \operatorname{space}_{N}(\mathfrak{L} q M) . \operatorname{Norm}(\mathfrak{L} q M) g \leq 1\right\}\right.\).ennreal \(\left(\int x . f x *\right.\) \(g x \partial M)) \geq \infty\)
apply (rule dense-le) using \(B\) by (meson SUP-upper2)
then show ?thesis using False neq-top-trans by force
qed

\subsection*{5.9 Conditional expectations and \(L^{p}\)}

The \(L^{p}\) space with respect to a subalgebra is included in the whole \(L^{p}\) space.
lemma Lp-subalgebra:
assumes subalgebra M F
shows \(\bigwedge f\).eNorm \((\mathfrak{L} p M) f \leq e \operatorname{Norm}(\mathfrak{L} p(\) restr-to-subalg \(M F)) f\)
\((\mathfrak{L} p(\) restr-to-subalg \(M F)) \subseteq_{N} \mathfrak{L} p M\)
space \(_{N}((\mathfrak{L} p(\) restr-to-subalg M F \())) \subseteq\) space \(_{N}(\mathfrak{L} p M)\)
\(\bigwedge f . f \in \operatorname{space}_{N}((\mathfrak{L} p(\) restr-to-subalg \(M F))) \Longrightarrow \operatorname{Norm}(\mathfrak{L} p M) f=\) Norm \((\mathfrak{L} p(\) restr-to-subalg \(M F)) f\)
proof -
have \(*: f \in \operatorname{space}_{N}(\mathfrak{L} p M) \wedge \operatorname{Norm}(\mathfrak{L} p M) f=\operatorname{Norm}(\mathfrak{L} p\) (restr-to-subalg \(M F)) f\)
if \(f \in \operatorname{space}_{N}(\mathfrak{L} p(\) restr-to-subalg \(M F))\) for \(f\)

\section*{proof -}
have [measurable]: \(f \in\) borel-measurable (restr-to-subalg \(M F\) ) using that Lp-measurable by auto
then have [measurable]: \(f \in\) borel-measurable \(M\)
using assms measurable-from-subalg measurable-in-subalg' by blast
show ?thesis
proof (cases rule: Lp-cases[of \(p]\) )
case zero
then show ?thesis using that unfolding \(\langle p=0\rangle L\)-zero-space Norm-def L-zero by auto
next
case PInf
have [measurable]: \(f \in\) borel-measurable (restr-to-subalg \(M F\) ) using that Lp-measurable by auto
then have [measurable]: \(f \in\) borel-measurable \(F\) using assms measur-able-in-subalg' by blast
then have [measurable]: \(f \in\) borel-measurable \(M\) using assms measur-able-from-subalg by blast
have \(A E x\) in (restr-to-subalg MF). \(|f x| \leq \operatorname{Norm}(\mathfrak{L} \infty\) (restr-to-subalg \(M\) F)) \(f\)
using L-infinity-AE-bound that unfolding \(\langle p=\infty\) by auto
then have \(a: A E x\) in \(M .|f x| \leq \operatorname{Norm}(\mathfrak{L} \infty(\) restr-to-subalg \(M F)) f\)
using assms AE-restr-to-subalg by blast
have \(*: f \in\) space \(_{N}(\mathfrak{L} \infty M)\) Norm \((\mathfrak{L} \infty M) f \leq \operatorname{Norm}(\mathfrak{L} \infty\) (restr-to-subalg
```

MF))f
using L-infinity-I[OF< }f\in\mathrm{ borel-measurable M>a] by auto
then have b:AE x in M. |fx|\leqNorm ({~ \inftyM)f
using L-infinity-AE-bound by auto
have c:AE x in (restr-to-subalg M F). |f x | \leqNorm ({ L M ) f
apply (rule AE-restr-to-subalg2[OF assms]) using b by auto
have Norm ({~\infty(restr-to-subalg MF)) f}\leq\operatorname{Norm}(\mathfrak{L}\inftyM)
using L-infinity-I[OF}\langlef\in\mathrm{ borel-measurable (restr-to-subalg M F)>c] by
auto
then show ?thesis using * unfolding <p=\infty> by auto
next
case (real-pos p2)
then have a [measurable]: f\in space N ({L p2 (restr-to-subalg M F))
using that unfolding <p= ennreal p2\rangle by auto
then have b [measurable]: f\in space}\mp@subsup{N}{N}{({L p2 M)
unfolding Lp-space[OF <p2 > 0`] using integrable-from-subalg[OF assms] by auto     show ?thesis         unfolding <p = ennreal p2` Lp-D[OF\langlep2 > 0\ranglea] Lp-D[OF\langlep2 > 0\rangleb]
using integral-subalgebra2[OF assms, symmetric, of f] apply (auto simp
add: b)
by (metis (mono-tags, lifting)<integrable (restr-to-subalg M F) (\lambdax. |f x|
powr p2)> assms integrableD(1) integral-subalgebra2 measurable-in-subalg')
qed
qed
show space}N(({\mathfrak{L}p(\mathrm{ restr-to-subalg M F))) }\subseteq\mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}pM)\mathrm{ using * by auto
show Norm (\mathfrak{L p M ) f = Norm ( }\mathfrak{L}p(restr-to-subalg M F)) f if f\in space}\mp@subsup{N}{N}{}((\mathfrak{L
p(restr-to-subalg M F))) for f
using * that by auto
show eNorm ( L p M) f\leqeNorm ({ L p (restr-to-subalg M F)) f for f
by (metis * eNorm-Norm eq-iff infinity-ennreal-def less-imp-le spaceN-iff top.not-eq-extremum)
then show ( }\mathfrak{L}p(\mathrm{ restr-to-subalg M F)) }\mp@subsup{\subseteq}{N}{}\mathfrak{L}p
by (metis ennreal-1 mult.left-neutral quasinorm-subsetI)
qed

```

For \(p \geq 1\), the conditional expectation of an \(L^{p}\) function still belongs to \(L^{p}\), with an \(L^{p}\) norm which is bounded by the norm of the original function. This is wrong for \(p<1\). One can prove this separating the cases and using the conditional version of Jensen's inequality, but it is much more efficient to do it with duality arguments, as follows.
```

proposition Lp-real-cond-exp:
assumes [simp]: subalgebra M F
and p\geq(1::ennreal)
sigma-finite-measure (restr-to-subalg M F)
f\in spaceN
shows real-cond-exp MFf\in \mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}p(restr-to-subalg M F))
Norm ({L p (restr-to-subalg M F)) (real-cond-exp M F f) \leqNorm ({ p M)f
proof -
have [measurable]: f\in borel-measurable M using Lp-measurable assms by auto

```
define \(q\) where \(q=\) conjugate-exponent \(p\)
have \(1 / p+1 / q=1\) unfolding \(q\)-def using conjugate-exponent-ennreal \([O F<p\) \(\geq 1\rangle\) ] by simp
have eNorm ( \(\mathfrak{L} p\) (restr-to-subalg MF)) (real-cond-exp MFf)
\(=\left(S U P g \in\left\{g \in\right.\right.\) space \(_{N}(\mathfrak{L} q(\) restr-to-subalg MF)\()\). Norm \((\mathfrak{L} q\) (restr-to-subalg \(M F)) g \leq 1\}\). ennreal \(\left(\int x\right.\). (real-cond-exp \(\left.M F f\right) x * g x \partial(\) restr-to-subalg \(\left.\left.M F)\right)\right)\)
by (rule Lp-Lq-duality \({ }^{\prime}[O F\langle 1 / p+1 / q=1\rangle\langle\) sigma-finite-measure (restr-to-subalg MF) \()\) ], simp)
also have \(\ldots \leq\left(S U P g \in\left\{g \in \operatorname{space}_{N}(\mathfrak{L} q M) . \operatorname{Norm}(\mathfrak{L} q M) g \leq 1\right\}\right.\).ennreal \(\left(\int x\right.\). \(f x * g x \partial M)\) )
proof (rule SUP-mono, auto)
fix \(g\) assume \(H: g \in \operatorname{space}_{N}(\mathfrak{L} q\) (restr-to-subalg MF))
\(\operatorname{Norm}(\mathfrak{L} q(\) restr-to-subalg \(M F)) g \leq 1\)
then have H2: \(g \in \operatorname{space}_{N}(\mathfrak{L} q M) \operatorname{Norm}(\mathfrak{L} q M) g \leq 1\)
using Lp-subalgebra[OF «subalgebra \(M\) F \(\rangle\) ] by (auto simp add: subset-iff)
have [measurable]: \(g \in\) borel-measurable \(M g \in\) borel-measurable \(F\)
using Lp-measurable[OF H(1)] Lp-measurable[OF H2(1)] by auto
have int: integrable \(M(\lambda x . f x * g x)\)
using Lp-Lq-duality-bound (1)[OF \(\langle 1 / p+1 / q=1\rangle\left\langle f \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\) H2 (1)].
have \(\left(\int x .(\right.\) real-cond-exp \(M F f) x * g x \partial(\) restr-to-subalg \(\left.M F)\right)=\left(\int x . g x *\right.\) (real-cond-exp MFf) x \(\partial M\) )
by (subst mult.commute, rule integral-subalgebra2[OF «subalgebra M F \(\upharpoonright\) ], auto)
also have \(\ldots=\left(\int x . g x * f x \partial M\right)\)
apply (rule sigma-finite-subalgebra.real-cond-exp-intg, auto simp add: int mult.commute)
unfolding sigma-finite-subalgebra-def using assms by auto
finally have ennreal \(\left(\int x\right.\). (real-cond-exp \(\left.M F f\right) x * g x \partial\) (restr-to-subalg \(M\) \(F)) \leq \operatorname{ennreal}\left(\int x . f x * g x \partial M\right)\)
by (auto intro!: ennreal-leI simp add: mult.commute)
then show \(\exists m . m \in \operatorname{space}_{N}(\mathfrak{L} q M) \wedge \operatorname{Norm}(\mathfrak{L} q M) m \leq 1\)
\(\wedge\) ennreal (LINT x \(\mid\) restr-to-subalg MF. real-cond-exp MFfx*gx) \(\leq\) ennreal (LINT \(x \mid M . f x * m x)\)
using H2 by blast
qed
also have \(\ldots=\operatorname{eNorm}(\mathfrak{L} p M) f\)
apply (rule \(L p\)-Lq-duality' \([O F<1 / p+1 / q=1\rangle\), symmetric], auto intro!: sigma-finite-subalgebra-is-sigma-finite \([o f-F])\)
unfolding sigma-finite-subalgebra-def using assms by auto
finally have \(*: \operatorname{eNorm}(\mathfrak{L} p(\) restr-to-subalg \(M F))(\) real-cond-exp \(M F f) \leq\) \(\operatorname{eNorm}(\mathfrak{L} p M) f\)
by \(\operatorname{simp}\)
then show a: real-cond-exp \(M F f \in \operatorname{space}_{N}(\mathfrak{L} p(\) restr-to-subalg MF))
apply (subst space \(N\)-iff) using \(\left\langle f \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\) by (simp add: space \({ }_{N}\)-def) show \(\operatorname{Norm}(\mathfrak{L} p(\) restr-to-subalg \(M F))(\) real-cond-exp MFf)\(\leq \operatorname{Norm}(\mathfrak{L} p M)\) \(f\)
using \(*\) unfolding eNorm-Norm[OF \(\left.\left\langle f \in \operatorname{space}_{N}(\mathfrak{L} p M)\right\rangle\right]\) eNorm-Norm[OF a] by \(\operatorname{simp}\) qed
```

lemma Lp-real-cond-exp-eNorm:
assumes [simp]: subalgebra MF
and }p\geq(1::\mathrm{ ennreal )
sigma-finite-measure (restr-to-subalg M F)
shows eNorm ({ p (restr-to-subalg M F)) (real-cond-exp M F f) \leqeNorm ({ p
M) }
proof (cases eNorm (L p M)f=\infty)
case False
then have *: f\in space}N( ({ p M)
unfolding spaceN-iff by (simp add: top.not-eq-extremum)
show ?thesis
using Lp-real-cond-exp[OF assms <f \in space }\mp@subsup{N}{N}{}(\mathfrak{L}pM)\rangle]\mathrm{ by (subst eNorm-Norm,
auto simp:<f \in \mp@subsup{\operatorname{space}}{N}{}(\mathfrak{L}pM)\rangle)+
qed (simp)
end

```
```

