Coproduct Measure

Michikazu Hirata

March 19, 2025

Abstract

This entry formalizes the coproduct measure. Let I be a set and $\{M_i\}_{i\in I}$ measurable spaces. The σ-algebra on $\coprod_{i\in I} M_i = \{(i,x) \mid i \in I \wedge x \in M_i\}$ is defined as the least one making $(\lambda x.\ (i,x))$ measurable for all $i \in I$. Let μ_i be measures on M_i for all $i \in I$ and A a measurable set of $\coprod_{i\in I} M_i$. The coproduct measure $\coprod_{i\in I} \mu_i$ is defined as follows:

$$\left(\coprod_{i\in I} \mu_i\right)(A) = \sum_{i\in I} \mu_i(A_i), \quad \text{where } A_i = \{x \mid (i,x) \in A\}.$$

We also prove the relationship with coproduct quasi-Borel spaces: the functor $R : \mathbf{Meas} \to \mathbf{QBS}$ preserves countable coproducts.

Contents

1 Preliminaries

theory *Lemmas-Coproduct-Measure*
imports *HOL−Probability.Probability*
Standard-Borel-Spaces.Abstract-Metrizable-Topology
begin

1.1 Metrics and Metrizability

lemma *metrizable-space-metric-space*:
assumes *d:Metric-space UNIV d Metric-space.mtopology UNIV d = euclidean*
shows *class.metric-space d* (⨅ *e*∈{*0*<..}. *principal* {(*x*,*y*). *d x y* < *e*}) *open*
⟨*proof*⟩

corollary *metrizable-space-metric-space-ex*:
assumes *metrizable-space* (*euclidean* :: ′*a* :: *topological-space topology*)
shows ∃(*d* :: ′*a* ⇒ ′*a* ⇒ *real*) *F*. *class.metric-space d F open*
⟨*proof*⟩

lemma *completely-metrizable-space-metric-space*:
assumes *Metric-space* (*UNIV* :: ′*a* ::*topological-space set*) *d Metric-space.mtopology UNIV d = euclidean Metric-space.mcomplete UNIV d*
shows *class.complete-space d* (⨅ *e*∈{*0*<..}. *principal* {(*x*,*y*). *d x y* < *e*}) *open*
⟨*proof*⟩

lemma *completely-metrizable-space-metric-space-ex*:
assumes *completely-metrizable-space* (*euclidean* :: ′*a* :: *topological-space topology*)
shows ∃(*d* :: ′*a* ⇒ ′*a* ⇒ *real*) *F*. *class.complete-space d F open*
⟨*proof*⟩

1.2 Copy of Extended non-negative reals

In the proof of the change of ordering of the infinite sum (*infsum*) for *ennreal*, we use `infsum_Sigma` and `compact_uniformly_continuous`. Thus, we need to interpret *ennreal* as a metric space. However, there is no standard metric on *ennreal* even though it is a Polish space (thus, a metrizable space). Hence, we do not want to give a metric on *ennreal* globally. Instead of defining a metric on *ennreal*, we define a type copy of *ennreal*, then define a metric on the copy and prove the change of ordering of the infinite sum. Finally, we transfer the theorems to the ones for *ennreal*.

typedef *ennreal′* = *UNIV* :: *ennreal set*
⟨*proof*⟩

lemma *bij-Abs-ennreal′*: *bij Abs-ennreal′*
⟨*proof*⟩

lemma *inj-Abs-ennreal′*: *inj Abs-ennreal′*
⟨*proof*⟩

setup-lifting *type-definition-ennreal′*

instantiation *ennreal′* :: *complete-linorder*
begin

lift-definition *top-ennreal′* :: *ennreal′* **is** *top* ⟨*proof*⟩
lift-definition *bot-ennreal′* :: *ennreal′* **is** *0* ⟨*proof*⟩
lift-definition *sup-ennreal′* :: *ennreal′* ⇒ *ennreal′* ⇒ *ennreal′* **is** *sup* ⟨*proof*⟩
lift-definition *inf-ennreal′* :: *ennreal′* ⇒ *ennreal′* ⇒ *ennreal′* **is** *inf* ⟨*proof*⟩
lift-definition *Inf-ennreal′* :: *ennreal′ set* ⇒ *ennreal′* **is** *Inf* ⟨*proof*⟩
lift-definition *Sup-ennreal′* :: *ennreal′ set* ⇒ *ennreal′* **is** *sup 0* ∘ *Sup* ⟨*proof*⟩

lift-definition *less-eq-ennreal′* :: *ennreal′* ⇒ *ennreal′* ⇒ *bool* **is** (≤) ⟨*proof*⟩
lift-definition *less-ennreal′* :: *ennreal′* ⇒ *ennreal′* ⇒ *bool* **is** (<) ⟨*proof*⟩

instance
⟨*proof*⟩

end

instantiation *ennreal′* :: *infinity*
begin

definition *infinity-ennreal′* :: *ennreal′*
where
[*simp*]: ∞ = (*top*::*ennreal′*)

instance ⟨*proof*⟩

end

instantiation *ennreal′* :: {*semiring-1-no-zero-divisors*, *comm-semiring-1*}
begin

lift-definition *one-ennreal′* :: *ennreal′* **is** *1* ⟨*proof*⟩
lift-definition *zero-ennreal′* :: *ennreal′* **is** *0* ⟨*proof*⟩

lift-definition *plus-ennreal′* :: *ennreal′* ⇒ *ennreal′* ⇒ *ennreal′* **is** (+) ⟨*proof*⟩
lift-definition *times-ennreal′* :: *ennreal′* ⇒ *ennreal′* ⇒ *ennreal′* **is** (∗) ⟨*proof*⟩

instance
⟨*proof*⟩

end

instantiation *ennreal′* :: *minus*
begin

lift-definition *minus-ennreal′* :: *ennreal′* ⇒ *ennreal′* ⇒ *ennreal′* **is** *minus* ⟨*proof*⟩

instance ⟨*proof*⟩

end

instance *ennreal′* :: *numeral* ⟨*proof*⟩

instance *ennreal′* :: *ordered-comm-monoid-add*
⟨*proof*⟩

lemma *ennreal′-nonneg*[*simp*]: $\bigwedge r$ :: *ennreal′*. *0* ≤ *r*
⟨*proof*⟩

lemma *sum-Rep-ennreal′*[*simp*]: ($\sum$ *i*∈*I*. *Rep-ennreal′* (*f i*)) = *Rep-ennreal′* (*sum f I*)
⟨*proof*⟩

lemma *transfer-sum-ennreal′* [*transfer-rule*]:
rel-fun (*rel-fun* (=) *pcr-ennreal′*) (*rel-fun* (=) *pcr-ennreal′*) *sum sum*
⟨*proof*⟩

lemma *pcr-ennreal′-eq*:*pcr-ennreal′ a b* ⟷ *b* = *Abs-ennreal′ a*
⟨*proof*⟩

lemma *rel-set-pcr-ennreal′-eq*:*rel-set pcr-ennreal′ A B* ⟷ *B* = *Abs-ennreal′* ‘ *A*
⟨*proof*⟩

lemma *transfer-lessThan-ennreal′*[*transfer-rule*]:
rel-fun pcr-ennreal′ (*rel-set pcr-ennreal′*) *lessThan lessThan*
⟨*proof*⟩

lemma *transfer-greaterThan-ennreal′*[*transfer-rule*]:
rel-fun pcr-ennreal′ (*rel-set pcr-ennreal′*) *greaterThan greaterThan*
⟨*proof*⟩

The transfer rule for *generate-topology*.

lemma *homeomorphism-generating-topology-imp*:

assumes *bj*:*bij f*
and *generate-topology S a*
shows *generate-topology* ((‘) *f* ‘ *S*) (*f* ‘ *a*)
⟨*proof*⟩

corollary *homeomorphism-generating-topology-eq*:
assumes *bjf*: *bij f*
shows *generate-topology S a* = *generate-topology* ((‘) *f* ‘ *S*) (*f* ‘ *a*)
⟨*proof*⟩

lemma *transfer-generate-topology-ennreal′*[*transfer-rule*]:
rel-fun (*rel-set* (*rel-set pcr-ennreal′*)) (*rel-fun* (*rel-set pcr-ennreal′*) (=)) *generate-topology generate-topology*
⟨*proof*⟩

instantiation *ennreal′* :: *topological-space*
begin

lift-definition *open-ennreal′* :: *ennreal′ set* ⇒ *bool* **is** *open* ⟨*proof*⟩

instance
⟨*proof*⟩

end

instance *ennreal′* :: *second-countable-topology*
⟨*proof*⟩

instance *ennreal′* :: *linorder-topology*
⟨*proof*⟩

lemma *continuous-map-Abs-ennreal′*:*continuous-on UNIV Abs-ennreal′*
⟨*proof*⟩

lemma *continuous-map-Rep-ennreal′*:*continuous-on UNIV Rep-ennreal′*
⟨*proof*⟩

corollary *continuous-map-ennreal′-eq*: *continuous-on A f* ⟷ *continuous-on A* (λ*x*. *Rep-ennreal′* (*f x*))
⟨*proof*⟩

lemma *ennreal-ennreal′-homeomorphic*:
(*euclidean* :: *ennreal topology*) *homeomorphic-space* (*euclidean* :: *ennreal′ topology*)
⟨*proof*⟩

corollary *Polish-space-ennreal′*: *Polish-space* (*euclidean* :: *ennreal′ topology*)
⟨*proof*⟩

instantiation *ennreal′* :: *metric-space*

begin

definition *dist-ennreal′* :: *ennreal′* ⇒ *ennreal′* ⇒ *real*
  where *dist-ennreal′* ≡ *SOME d. Metric-space UNIV d* ∧
    Metric-space.mtopology UNIV d = *euclidean* ∧
    Metric-space.mcomplete UNIV d

definition *uniformity-ennreal′* :: (*ennreal′* × *ennreal′*) *filter*
  where *uniformity-ennreal′* ≡ ⨅ *e*∈{*0*<..}. *principal* {(*x*,*y*). *dist x y* < *e*}

instance
⟨*proof*⟩

end

1.3 Lemmas for Infinite Sum

lemma *transfer-nhds-ennreal′*[*transfer-rule*]: *rel-fun pcr-ennreal′* (*rel-filter pcr-ennreal′*) *nhds nhds*
⟨*proof*⟩

lemmas *transfer-filtermap-ennreal′*[*transfer-rule*] = *filtermap-parametric*[**where** *A*=*HOL.eq* **and** *B*=*pcr-ennreal′*]

lemma *transfer-filterlim-ennreal′*[*transfer-rule*]:
rel-fun (*rel-fun* (=) *pcr-ennreal′*) (*rel-fun* (*rel-filter pcr-ennreal′*) (*rel-fun* (*rel-filter* (=)) (=))) *filterlim filterlim*
  ⟨*proof*⟩

lemma *transfer-The-ennreal*:
  assumes *P*:∃!*x*. *P x*
    and *rel-fun pcr-ennreal′* (=) *P P′*
  shows *The P′* = *Abs-ennreal′* (*The P*)
⟨*proof*⟩

lemma *transfer-infsum-ennreal′*[*transfer-rule*]:
rel-fun (*rel-fun* (=) *pcr-ennreal′*) (*rel-fun* (=) *pcr-ennreal′*) *infsum* (*infsum* :: (*′a* ⇒ -) ⇒ - ⇒ -)
⟨*proof*⟩

lemma *inf-sum-Rep-Abs-ennreal′*:*infsum f A* = *Rep-ennreal′* (*infsum* ((λ*x*. *Abs-ennreal′* (*f x*))) *A*)
  ⟨*proof*⟩

lemma *continuous-on-add-ennreal′*:
  fixes *f g* :: *′a*::*topological-space* ⇒ *ennreal′*
  shows *continuous-on A f* ⟹ *continuous-on A g* ⟹ *continuous-on A* (λ*x*. *f x* + *g x*)
  ⟨*proof*⟩

lemma *uniformly-continuous-add-ennreal'*: *isUCont* $(\lambda(x{::}ennreal', y).\ x + y)$
⟨*proof*⟩

lemma *infsum-eq-suminf*:
assumes *f summable-on UNIV*
shows $(\sum_\infty n{\in}UNIV.\ f\ n) = suminf\ f$
⟨*proof*⟩

lemma *infsum-Sigma-ennreal'*:
fixes f :: - ⇒ *ennreal'*
shows *infsum f* $(Sigma\ A\ B) = infsum\ (\lambda x.\ infsum\ (\lambda y.\ f\ (x, y))\ (B\ x))\ A$
⟨*proof*⟩

lemma *infsum-swap-ennreal'*:
fixes f :: - ⇒ - ⇒ *ennreal'*
shows *infsum* $(\lambda x.\ infsum\ (\lambda y.\ f\ x\ y)\ B)\ A = infsum\ (\lambda y.\ infsum\ (\lambda x.\ f\ x\ y)\ A)$
B
⟨*proof*⟩

lemma *infsum-Sigma-ennreal*:
fixes f :: - ⇒ *ennreal*
shows *infsum f* $(Sigma\ A\ B) = infsum\ (\lambda x.\ infsum\ (\lambda y.\ f\ (x, y))\ (B\ x))\ A$
⟨*proof*⟩

lemma *infsum-swap-ennreal*:
fixes f :: - ⇒ - ⇒ *ennreal*
shows *infsum* $(\lambda x.\ infsum\ (\lambda y.\ f\ x\ y)\ B)\ A = infsum\ (\lambda y.\ infsum\ (\lambda x.\ f\ x\ y)\ A)$
B
⟨*proof*⟩

lemma *has-sum-cmult-right-ennreal*:
fixes f :: - ⇒ *ennreal*
assumes $c < \top$ $(f\ has\text{-}sum\ a)\ A$
shows $((\lambda x.\ c * f\ x)\ has\text{-}sum\ c * a)\ A$
⟨*proof*⟩

lemma *infsum-cmult-right-ennreal*:
fixes f :: - ⇒ *ennreal*
assumes $c < \top$
shows $(\sum_\infty x{\in}A.\ c * f\ x) = c * infsum\ f\ A$
⟨*proof*⟩

lemma *ennreal-sum-SUP-eq*:
fixes f :: *nat* ⇒ - ⇒ *ennreal*
assumes *finite A* $\bigwedge x.\ x \in A \Longrightarrow incseq\ (\lambda j.\ f\ j\ x)$
shows $(\sum i{\in}A.\ \bigsqcup n.\ f\ n\ i) = (\bigsqcup n.\ \sum i{\in}A.\ f\ n\ i)$
⟨*proof*⟩

lemma *ennreal-infsum-Sup-eq*:
fixes f :: $nat \Rightarrow$ - $\Rightarrow$ $ennreal$
assumes $\bigwedge x.\ x \in A \Longrightarrow incseq\ (\lambda j.\ f\ j\ x)$
shows $(\sum_\infty x{\in}A.\ (SUP\ j.\ f\ j\ x)) = (SUP\ j.\ (\sum_\infty x{\in}A.\ f\ j\ x))$ (**is** $?lhs = ?rhs$)
⟨*proof*⟩

lemma *bounded-infsum-summable*:
assumes $\bigwedge x.\ x \in A \Longrightarrow f\ x \geq 0$ $(\sum_\infty x{\in}A.\ ennreal\ (f\ x)) < top$
shows f *summable-on* A
⟨*proof*⟩

lemma *infsum-less-top-dest*:
fixes f :: - $\Rightarrow$ -::{*ordered-comm-monoid-add*, *topological-comm-monoid-add*, *t2-space*, *complete-linorder*, *linorder-topology*}
assumes $(\sum_\infty x{\in}A.\ f\ x) < top$ $\bigwedge x.\ x \in A \Longrightarrow f\ x \geq 0$ $x \in A$
shows $f\ x < top$
⟨*proof*⟩

lemma *infsum-ennreal-eq*:
assumes f *summable-on* A $\bigwedge x.\ x \in A \Longrightarrow f\ x \geq 0$
shows $(\sum_\infty x{\in}A.\ ennreal\ (f\ x)) = ennreal\ (\sum_\infty x{\in}A.\ f\ x)$
⟨*proof*⟩

lemma *abs-summable-on-integrable-iff*:
fixes f :: - $\Rightarrow$ - :: {*banach*, *second-countable-topology*}
shows *Infinite-Sum.abs-summable-on* f A $\longleftrightarrow$ *integrable* (*count-space* A) f
⟨*proof*⟩

lemma *infsum-eq-integral*:
fixes f :: - $\Rightarrow$ - :: {*banach*, *second-countable-topology*}
assumes *Infinite-Sum.abs-summable-on* f A
shows *infsum* f A = $integral^L$ (*count-space* A) f
⟨*proof*⟩

end

theory *Coproduct-Measure*
imports *Lemmas-Coproduct-Measure*
HOL−Analysis.Analysis
begin

2 Binary Coproduct Measures

definition *copair-measure* :: [*'a measure*, *'b measure*] $\Rightarrow$ (*'a* + *'b*) *measure* (**infixr** ‹$\bigoplus_M$› *65*) **where**
$M \bigoplus_M N$ = *measure-of* (*space* M <+> *space* N)
({*Inl* ‘ A |A. $A \in$ *sets* M} ∪ {*Inr* ‘ A|A. $A \in$ *sets* N})
(λA. *emeasure* M (*Inl* −‘ A) + *emeasure* N (*Inr* −‘ A))

2.1 The Measurable Space and Measurability

lemma
shows *space-copair-measure*: *space* (*copair-measure M N*) = *space M* <+> *space N*
and *sets-copair-measure-sigma*:
sets (*copair-measure M N*)
= *sigma-sets* (*space M* <+> *space N*) ($\{$*Inl* ‘ *A* $|$*A*. *A* $\in$ *sets M*$\}$ $\cup$ $\{$*Inr* ‘ *A*$|$*A*. *A* $\in$ *sets N*$\}$)
and *Inl-measurable*[*measurable*]: *Inl* $\in M \to_M M \bigoplus_M N$
and *Inr-measurable*[*measurable*]: *Inr* $\in N \to_M M \bigoplus_M N$
⟨*proof*⟩

lemma *sets-copair-measure-cong*:
sets M1 = *sets M2* $\Longrightarrow$ *sets N1* = *sets N2* $\Longrightarrow$ *sets* ($M1 \bigoplus_M N1$) = *sets* ($M2 \bigoplus_M N2$)
⟨*proof*⟩

lemma *measurable-image-Inl*[*measurable*]: *A* $\in$ *sets M* $\Longrightarrow$ *Inl* ‘ *A* $\in$ *sets* ($M \bigoplus_M N$)
⟨*proof*⟩

lemma *measurable-image-Inr*[*measurable*]: *A* $\in$ *sets N* $\Longrightarrow$ *Inr* ‘ *A* $\in$ *sets* ($M \bigoplus_M N$)
⟨*proof*⟩

lemma *measurable-vimage-Inl*:
assumes [*measurable*]:*A* $\in$ *sets* ($M \bigoplus_M N$)
shows *Inl* $-$‘ *A* $\in$ *sets M*
⟨*proof*⟩

lemma *measurable-vimage-Inr*:
assumes [*measurable*]:*A* $\in$ *sets* ($M \bigoplus_M N$)
shows *Inr* $-$‘ *A* $\in$ *sets N*
⟨*proof*⟩

lemma *in-sets-copair-measure-iff*:
A $\in$ *sets* (*copair-measure M N*) $\longleftrightarrow$ *Inl* $-$‘ *A* $\in$ *sets M* $\wedge$ *Inr* $-$‘ *A* $\in$ *sets N*
⟨*proof*⟩

lemma *measurable-copair-Inl-Inr*:
assumes [*measurable*]:(λx. *f* (*Inl x*)) $\in M \to_M L$ (λx. *f* (*Inr x*)) $\in N \to_M L$
shows $f \in M \bigoplus_M N \to_M L$
⟨*proof*⟩

corollary *measurable-copair-measure-iff*:
$f \in M \bigoplus_M N \to_M L \longleftrightarrow$ (λx. *f* (*Inl x*)) $\in M \to_M L \wedge$ (λx. *f* (*Inr x*)) $\in N \to_M L$
⟨*proof*⟩

lemma *measurable-copair-dest1*:
assumes [*measurable*]:$f \in L \to_M M \bigoplus_M N$ **and** $f -` (Inl ` space\ M) \cap space\ L = space\ L$
obtains f' **where** $f' \in L \to_M M$ $\bigwedge x.\ x \in space\ L \Longrightarrow f\ x = Inl\ (f'\ x)$
⟨*proof*⟩

lemma *measurable-copair-dest2*:
assumes [*measurable*]:$f \in L \to_M M \bigoplus_M N$ **and** $f -` (Inr ` space\ N) \cap space\ L = space\ L$
obtains f' **where** $f' \in L \to_M N$ $\bigwedge x.\ x \in space\ L \Longrightarrow f\ x = Inr\ (f'\ x)$
⟨*proof*⟩

lemma *measurable-copair-dest3*:
assumes [*measurable*]:$f \in L \to_M M \bigoplus_M N$
and $f -` (Inl ` space\ M) \cap space\ L \subset space\ L$ $f -` (Inr ` space\ N) \cap space\ L \subset space\ L$
obtains f' f'' **where** $f' \in L \to_M M$ $f'' \in L \to_M N$
$\bigwedge x.\ x \in space\ L \Longrightarrow x \in f -` Inl ` space\ M \Longrightarrow f\ x = Inl\ (f'\ x)$
$\bigwedge x.\ x \in space\ L \Longrightarrow x \notin f -` Inl ` space\ M \Longrightarrow f\ x = Inr\ (f''\ x)$
⟨*proof*⟩

2.2 Measures

lemma *emeasure-copair-measure*:
assumes [*measurable*]: $A \in sets\ (M \bigoplus_M N)$
shows $emeasure\ (M \bigoplus_M N)\ A = emeasure\ M\ (Inl -` A) + emeasure\ N\ (Inr -` A)$
⟨*proof*⟩

lemma *emeasure-copair-measure-space*:
$emeasure\ (M \bigoplus_M N)\ (space\ (M \bigoplus_M N)) = emeasure\ M\ (space\ M) + emeasure\ N\ (space\ N)$
⟨*proof*⟩

corollary
shows *emeasure-copair-measure-Inl*: $A \in sets\ M \Longrightarrow emeasure\ (M \bigoplus_M N)\ (Inl ` A) = emeasure\ M\ A$
and *emeasure-copair-measure-Inr*: $B \in sets\ N \Longrightarrow emeasure\ (M \bigoplus_M N)\ (Inr ` B) = emeasure\ N\ B$
⟨*proof*⟩

lemma *measure-copair-measure*:
assumes [*measurable*]:$A \in sets\ (M \bigoplus_M N)$ $emeasure\ (M \bigoplus_M N)\ A < \infty$
shows $measure\ (M \bigoplus_M N)\ A = measure\ M\ (Inl -` A) + measure\ N\ (Inr -` A)$
⟨*proof*⟩

lemma
shows *measure-copair-measure-Inl*: $A \in sets\ M \Longrightarrow measure\ (M \bigoplus_M N)\ (Inl `$

$A) = measure\ M\ A$
and *measure-copair-measure-Inr*: $B \in sets\ N \Longrightarrow measure\ (M \bigoplus_M N)\ (Inr$ ‘
$B) = measure\ N\ B$
⟨*proof*⟩

2.3 Finiteness

lemma *finite-measure-copair-measure*: $finite\text{-}measure\ M \Longrightarrow finite\text{-}measure\ N \Longrightarrow finite\text{-}measure\ (M \bigoplus_M N)$
⟨*proof*⟩

2.4 σ-Finiteness

lemma *sigma-finite-measure-copair-measure*:
assumes *sigma-finite-measure M sigma-finite-measure N*
shows $sigma\text{-}finite\text{-}measure\ (M \bigoplus_M N)$
⟨*proof*⟩

2.5 Non-Negative Integral

lemma *nn-integral-copair-measure*:
assumes $f \in borel\text{-}measurable\ (M \bigoplus_M N)$
shows $(\int^+ x.\ f\ x\ \partial(M \bigoplus_M N)) = (\int^+ x.\ f\ (Inl\ x)\ \partial M) + (\int^+ x.\ f\ (Inr\ x)\ \partial N)$
⟨*proof*⟩

2.6 Integrability

lemma *integrable-copair-measure-iff*:
fixes $f :: 'a + 'b \Rightarrow 'c::\{banach,\ second\text{-}countable\text{-}topology\}$
shows $integrable\ (M \bigoplus_M N)\ f \longleftrightarrow integrable\ M\ (\lambda x.\ f\ (Inl\ x)) \wedge integrable\ N\ (\lambda x.\ f\ (Inr\ x))$
⟨*proof*⟩

corollary *interable-copair-measureI*:
fixes $f :: 'a + 'b \Rightarrow 'c::\{banach,\ second\text{-}countable\text{-}topology\}$
shows $integrable\ M\ (\lambda x.\ f\ (Inl\ x)) \Longrightarrow integrable\ N\ (\lambda x.\ f\ (Inr\ x)) \Longrightarrow integrable\ (M \bigoplus_M N)\ f$
⟨*proof*⟩

2.7 The Lebesgue Integral

lemma *integral-copair-measure*:
fixes $f :: 'a + 'b \Rightarrow 'c::\{banach,\ second\text{-}countable\text{-}topology\}$
assumes $integrable\ (M \bigoplus_M N)\ f$
shows $(\int x.\ f\ x\ \partial(M \bigoplus_M N)) = (\int x.\ f\ (Inl\ x)\ \partial M) + (\int x.\ f\ (Inr\ x)\ \partial N)$
⟨*proof*⟩

3 Coproduct Measures

definition $coPiM :: ['i\ set,\ 'i \Rightarrow 'a\ measure] \Rightarrow ('i \times 'a)\ measure$ **where**

coPiM I Mi ≡ *measure-of*
(*SIGMA i*:*I*. *space* (*Mi i*))
{*A*. *A*⊆(*SIGMA i*:*I*. *space* (*Mi i*)) ∧ (∀ *i*∈*I*. *Pair i* −‘ *A* ∈ *sets* (*Mi i*))}
(λ*A*. ($\sum_\infty$ *i*∈*I*. *emeasure* (*Mi i*) (*Pair i* −‘ *A*)))

syntax
-coPiM :: *pttrn* ⇒ ′*i set* ⇒ ′*a measure* ⇒ (′*i* × ′*a*) *measure* (‹(*3*$\amalg_M$ -∈-./ -)› *10*)
translations
$\amalg_M$ *x*∈*I*. *M* ⇌ *CONST coPiM I* (λ*x*. *M*)

3.1 The Measurable Space and Measurability

lemma
shows *space-coPiM*: *space* (*coPiM I Mi*) = (*SIGMA i*:*I*. *space* (*Mi i*))
and *sets-coPiM*:
sets (*coPiM I Mi*) = *sigma-sets* (*SIGMA i*:*I*. *space* (*Mi i*)) {*A*. *A*⊆(*SIGMA i*:*I*. *space* (*Mi i*)) ∧ (∀ *i*∈*I*. *Pair i* −‘ *A* ∈ *sets* (*Mi i*))}
and *sets-coPiM-eq*:*sets* (*coPiM I Mi*) = {*A*. *A*⊆(*SIGMA i*:*I*. *space* (*Mi i*)) ∧ (∀ *i*∈*I*. *Pair i* −‘ *A* ∈ *sets* (*Mi i*))}
⟨*proof*⟩

lemma *sets-coPiM-cong*:
I = *J* ⟹ (⋀*i*. *i* ∈ *I* ⟹ *sets* (*Mi i*) = *sets* (*Ni i*)) ⟹ *sets* (*coPiM I Mi*) = *sets* (*coPiM J Ni*)
⟨*proof*⟩

lemma *measurable-coPiM2*:
assumes [*measurable*]:⋀*i*. *i* ∈ *I* ⟹ *f i* ∈ *Mi i* $\to_M$ *N*
shows (λ(*i*,*x*). *f i x*) ∈ *coPiM I Mi* $\to_M$ *N*
⟨*proof*⟩

lemma *measurable-Pair-coPiM*[*measurable* (*raw*)]:
assumes *i* ∈ *I*
shows *Pair i* ∈ *Mi i* $\to_M$ *coPiM I Mi*
⟨*proof*⟩

lemma *measurable-Pair-coPiM′*:
assumes *i* ∈ *I* (λ(*i*,*x*). *f i x*) ∈ *coPiM I Mi* $\to_M$ *N*
shows *f i* ∈ *Mi i* $\to_M$ *N*
⟨*proof*⟩

lemma *measurable-copair-iff*: (λ(*i*,*x*). *f i x*) ∈ *coPiM I Mi* $\to_M$ *N* ⟷ (∀ *i*∈*I*. *f i* ∈ *Mi i* $\to_M$ *N*)
⟨*proof*⟩

lemma *measurable-copair-iff′*: *f* ∈ *coPiM I Mi* $\to_M$ *N* ⟷ (∀ *i*∈*I*. (λ*x*. *f* (*i*, *x*)) ∈ *Mi i* $\to_M$ *N*)
⟨*proof*⟩

lemma *coPair-inverse-space-unit*:
$i \in I \Longrightarrow A \in sets\ (coPiM\ I\ Mi) \Longrightarrow Pair\ i\ -`\ A \cap space\ (Mi\ i) = Pair\ i\ -`\ A$
⟨*proof*⟩

lemma *measurable-Pair-vimage*:
assumes $i \in I\ A \in sets\ (coPiM\ I\ Mi)$
shows $Pair\ i\ -`\ A \in sets\ (Mi\ i)$
⟨*proof*⟩

lemma *measurable-Sigma-singleton*[*measurable* (*raw*)]:
$\bigwedge i\ A.\ i \in I \Longrightarrow A \in sets\ (Mi\ i) \Longrightarrow \{i\} \times A \in sets\ (coPiM\ I\ Mi)$
⟨*proof*⟩

lemma *sets-coPiM-countable*:
assumes *countable I*
shows $sets\ (coPiM\ I\ Mi) = sigma\text{-}sets\ (SIGMA\ i{:}I.\ space\ (Mi\ i))\ (\bigcup i{\in}I.\ (\times)\ \{i\}\ `\ (sets\ (Mi\ i)))$
⟨*proof*⟩

lemma *measurable-coPiM1'*:
assumes *countable I*
and [*measurable*]: $a \in N \to_M count\text{-}space\ I\ \bigwedge i.\ i \in a\ `\ (space\ N) \Longrightarrow g\ i \in N \to_M Mi\ i$
shows $(\lambda x.\ (a\ x,\ g\ (a\ x)\ x)) \in N \to_M coPiM\ I\ Mi$
⟨*proof*⟩

lemma *measurable-coPiM1*:
assumes *countable I*
and $a \in N \to_M count\text{-}space\ I\ \bigwedge i.\ i \in I \Longrightarrow g\ i \in N \to_M Mi\ i$
shows $(\lambda x.\ (a\ x,\ g\ (a\ x)\ x)) \in N \to_M coPiM\ I\ Mi$
⟨*proof*⟩

lemma *measurable-coPiM1-elements*:
assumes *countable I* **and** [*measurable*]:$f \in N \to_M coPiM\ I\ Mi$
obtains *a g*
where $a \in N \to_M count\text{-}space\ I$
$\bigwedge i.\ i \in I \Longrightarrow space\ (Mi\ i) \neq \{\} \Longrightarrow g\ i \in N \to_M Mi\ i$
$f = (\lambda x.\ (a\ x,\ g\ (a\ x)\ x))$
⟨*proof*⟩

3.2 Measures

lemma *emeasure-coPiM*:
assumes $A \in sets\ (coPiM\ I\ Mi)$
shows $emeasure\ (coPiM\ I\ Mi)\ A = (\sum_\infty i{\in}I.\ emeasure\ (Mi\ i)\ (Pair\ i\ -`\ A))$
⟨*proof*⟩

corollary *emeasure-coPiM-space*:

emeasure (*coPiM I Mi*) (*space* (*coPiM I Mi*)) = ($\sum_\infty$*i*∈*I*. *emeasure* (*Mi i*) (*space* (*Mi i*)))
⟨*proof*⟩

lemma *emeasure-coPiM-coproj*:
assumes [*measurable*]: *i* ∈ *I A* ∈ *sets* (*Mi i*)
shows *emeasure* (*coPiM I Mi*) ({*i*} × *A*) = *emeasure* (*Mi i*) *A*
⟨*proof*⟩

lemma *measure-coPiM-coproj*: *i* ∈ *I* ⟹ *A* ∈ *sets* (*Mi i*) ⟹ *measure* (*coPiM I Mi*) ({*i*} × *A*) = *measure* (*Mi i*) *A*
⟨*proof*⟩

lemma *emeasure-coPiM-less-top-summable*:
assumes [*measurable*]:*A* ∈ *sets* (*coPiM I Mi*) *emeasure* (*coPiM I Mi*) *A* < ∞
shows(λ*i*. *measure* (*Mi i*) (*Pair i* −' *A*)) *summable-on I*
⟨*proof*⟩

lemma *measure-coPiM*:
assumes [*measurable*]:*A* ∈ *sets* (*coPiM I Mi*) *emeasure* (*coPiM I Mi*) *A* < ∞
shows *measure* (*coPiM I Mi*) *A* = ($\sum_\infty$*i*∈*I*. *measure* (*Mi i*) (*Pair i* −' *A*))
⟨*proof*⟩

3.3 Non-Negative Integral

lemma *nn-integral-coPiM*:
assumes *f* ∈ *borel-measurable* (*coPiM I Mi*)
shows ($\int^+$*x*. *f x* ∂*coPiM I Mi*) = ($\sum_\infty$*i*∈*I*. ($\int^+$*x*. *f* (*i*, *x*) ∂*Mi i*))
⟨*proof*⟩

3.4 Integrability

lemma
fixes *f* :: - ⇒ ′*b*::{*banach*, *second-countable-topology*}
assumes *integrable* (*coPiM I Mi*) *f*
shows *integrable-coPiM-dest-sum*:($\sum_\infty$*i*∈*I*. ($\int^+$ *x*. *norm* (*f* (*i*, *x*)) ∂*Mi i*)) < ∞
and *integrable-coPiM-dest-integrable*: ⋀*i*. *i* ∈ *I* ⟹ *integrable* (*Mi i*) (λ*x*. *f* (*i*, *x*))
and *integrable-coPiM-summable-norm*: (λ*i*. (∫ *x*. *norm* (*f* (*i*, *x*)) ∂*Mi i*)) *summable-on I*
and *integrable-coPiM-abs-summable*: *Infinite-Sum.abs-summable-on* (λ*i*. (∫ *x*. *f* (*i*, *x*) ∂*Mi i*)) *I*
and *integrable-coPiM-summable*: (λ*i*. (∫ *x*. *f* (*i*, *x*) ∂*Mi i*)) *summable-on I*
⟨*proof*⟩

3.5 The Lebesgue Integral

lemma *integral-coPiM*:
fixes *f* :: - ⇒ ′*b*::{*banach*, *second-countable-topology*}

assumes *integrable* (*coPiM I Mi*) *f*
shows ($\int x.\ f\ x\ \partial coPiM\ I\ Mi$) = ($\sum_\infty i{\in}I.$ ($\int x.\ f\ (i,\ x)\ \partial Mi\ i$))
⟨*proof*⟩

3.6 Finite Coproduct Measures

lemma *emeasure-coPiM-finite*:
assumes *finite I A* ∈ *sets* (*coPiM I Mi*)
shows *emeasure* (*coPiM I Mi*) *A* = ($\sum i{\in}I.$ *emeasure* (*Mi i*) (*Pair i* −' *A*))
⟨*proof*⟩

lemma *emeasure-coPiM-finite-space*:
finite I ⟹ *emeasure* (*coPiM I Mi*) (*space* (*coPiM I Mi*)) = ($\sum i{\in}I.$ *emeasure* (*Mi i*) (*space* (*Mi i*)))
⟨*proof*⟩

lemma *measure-coPiM-finite*:
assumes *finite I A* ∈ *sets* (*coPiM I Mi*) *emeasure* (*coPiM I Mi*) *A* < ∞
shows *measure* (*coPiM I Mi*) *A* = ($\sum i{\in}I.$ *measure* (*Mi i*) (*Pair i* −' *A*))
⟨*proof*⟩

lemma *nn-integral-coPiM-finite*:
assumes *finite I f* ∈ *borel-measurable* (*coPiM I Mi*)
shows ($\int^+ x.\ f\ x\ \partial(coPiM\ I\ Mi)$) = ($\sum i{\in}I.$ ($\int^+ x.\ f\ (i,\ x)\ \partial(Mi\ i)$))
⟨*proof*⟩

lemma *integrable-coPiM-finite-iff*:
fixes *f* :: - ⇒ *'c*::{*banach*, *second-countable-topology*}
shows *finite I* ⟹ *integrable* (*coPiM I Mi*) *f* ⟷ (∀ *i*∈*I*. *integrable* (*Mi i*) (λ*x*. *f* (*i*, *x*)))
⟨*proof*⟩

lemma *integral-coPiM-finite*:
fixes *f* :: - ⇒ *'c*::{*banach*, *second-countable-topology*}
assumes *finite I integrable* (*coPiM I Mi*) *f*
shows ($\int x.\ f\ x\ \partial(coPiM\ I\ Mi)$) = ($\sum i{\in}I.$ ($\int x.\ f\ (i,\ x)\ \partial(Mi\ i)$))
⟨*proof*⟩

3.7 Countable Infinite Coproduct Measures

lemma *emeasure-coPiM-countable-infinite*:
assumes [*measurable*]: *bij-betw from-n* (*UNIV* :: *nat set*) *I A* ∈ *sets* (*coPiM I Mi*)
shows *emeasure* (*coPiM I Mi*) *A* = ($\sum n.$ *emeasure* (*Mi* (*from-n n*)) (*Pair* (*from-n n*) −' *A*))
⟨*proof*⟩

lemmas *emeasure-coPiM-countable-infinite'* = *emeasure-coPiM-countable-infinite*[*OF bij-betw-from-nat-into*]
lemmas *emeasure-coPiM-nat* = *emeasure-coPiM-countable-infinite*[*OF bij-id*,*simplified*]

lemma *measure-coPiM-countable-infinite*:
assumes [*measurable,simp*]: *bij-betw from-n* (*UNIV* :: *nat set*) *I A* ∈ *sets* (*coPiM I Mi*)
and *emeasure* (*coPiM I Mi*) *A* < ∞
shows *measure* (*coPiM I Mi*) *A* = (∑ *n*. *measure* (*Mi* (*from-n n*)) (*Pair* (*from-n n*) −‘ *A*)) (**is** *?lhs* = *?rhs*)
and *summable* (λ*n*. *measure* (*Mi* (*from-n n*)) (*Pair* (*from-n n*) −‘ *A*))
⟨*proof*⟩

lemmas *measure-coPiM-countable-infinite′* = *measure-coPiM-countable-infinite*[*OF bij-betw-from-nat-into*]
lemmas *measure-coPiM-nat* = *measure-coPiM-countable-infinite*[*OF bij-id*,*simplified id-apply*]

lemma *nn-integral-coPiM-countable-infinite*:
assumes [*measurable*]:*bij-betw from-n* (*UNIV* :: *nat set*) *I f* ∈ *borel-measurable* (*coPiM I Mi*)
shows (∫ ⁺*x*. *f x* ∂(*coPiM I Mi*)) = (∑ *n*. (∫ ⁺*x*. *f* (*from-n n*, *x*) ∂(*Mi* (*from-n n*)))) (**is** - = *?rhs*)
⟨*proof*⟩
lemmas *nn-integral-coPiM-countable-infinite′* = *nn-integral-coPiM-countable-infinite*[*OF bij-betw-from-nat-into*]
lemmas *nn-integral-coPiM-nat* = *nn-integral-coPiM-countable-infinite*[*OF bij-id*,*simplified*]

lemma
fixes *f* :: - ⇒ ′*b*::{*banach*, *second-countable-topology*}
assumes *bij-betw from-n* (*UNIV* :: *nat set*) *I integrable* (*coPiM I Mi*) *f*
shows *integrable-coPiM-countable-infinite-dest-sum*:(∑ *n*. (∫ ⁺ *x*. *norm* (*f* (*from-n n*, *x*)) ∂(*Mi* (*from-n n*)))) < ∞
and *integrable-coPiM-countable-infinite-dest′*: ⋀*n*. *integrable* (*Mi* (*from-n n*)) (λ*x*. *f* (*from-n n*, *x*))
⟨*proof*⟩

lemmas *integrable-coPiM-countable-infinite-dest-sum′* = *integrable-coPiM-countable-infinite-dest-sum*[*OF bij-betw-from-nat-into*]
lemmas *integrable-coPiM-countable-infinite-dest′′* = *integrable-coPiM-countable-infinite-dest′*[*OF bij-betw-from-nat-into*]
lemmas *integrable-coPiM-nat-dest-sum* = *integrable-coPiM-countable-infinite-dest-sum*[*OF bij-id*,*simplified id-apply*]
lemmas *integrable-coPiM-nat-dest* = *integrable-coPiM-countable-infinite-dest′*[*OF bij-id*,*simplified id-apply*]

lemma
fixes *f* :: - ⇒ ′*b*::{*banach*, *second-countable-topology*}
assumes *bij-betw from-n* (*UNIV* :: *nat set*) *I integrable* (*coPiM I Mi*) *f*
shows *integrable-coPiM-countable-infinite-summable-norm*: *summable* (λ*n*. (∫ *x*. *norm* (*f* (*from-n n*, *x*)) ∂(*Mi* (*from-n n*))))
and *integrable-coPiM-countable-infinite-summable-norm′*: *summable* (λ*n*. *norm*

($\int x.\ f\ (from\text{-}n\ n,\ x)\ \partial(Mi\ (from\text{-}n\ n)))$)
and *integrable-coPiM-countable-infinite-summable*: *summable* ($\lambda n.\ (\int x.\ f\ (from\text{-}n\ n,\ x)\ \partial(Mi\ (from\text{-}n\ n)))$)
⟨*proof*⟩

lemmas *integrable-coPiM-countable-infinite-summable-norm''*
= *integrable-coPiM-countable-infinite-summable-norm*[*OF bij-betw-from-nat-into*]
lemmas *integrable-coPiM-countable-infinite-summable-norm'''*
= *integrable-coPiM-countable-infinite-summable-norm'*[*OF bij-betw-from-nat-into*]
lemmas *integrable-coPiM-countable-infinite-summable'*
= *integrable-coPiM-countable-infinite-summable*[*OF bij-betw-from-nat-into*]
lemmas *integrable-coPiM-nat-summable-norm*
= *integrable-coPiM-countable-infinite-summable-norm*[*OF bij-id*,*simplified id-apply*]
lemmas *integrable-coPiM-nat-summable-norm'*
= *integrable-coPiM-countable-infinite-summable-norm'*[*OF bij-id*,*simplified id-apply*]
lemmas *integrable-coPiM-nat-summable*
= *integrable-coPiM-countable-infinite-summable*[*OF bij-id*,*simplified id-apply*]

lemma
fixes f :: - ⇒ *'b*::{*banach*, *second-countable-topology*}
assumes *countable I infinite I integrable* (*coPiM I Mi*) *f*
shows *integrable-coPiM-countable-infinite-dest*: $\bigwedge i.\ i \in I \Longrightarrow$ *integrable* (*Mi i*) ($\lambda x.\ f\ (i,\ x)$)
⟨*proof*⟩

lemma *integrable-coPiM-countable-infiniteI*:
fixes f :: - ⇒ *'b*::{*banach*, *second-countable-topology*}
assumes *bij-betw from-n* (*UNIV* :: *nat set*) *I* $\bigwedge i.\ i \in I \Longrightarrow (\lambda x.\ f\ (i,x)) \in$ *borel-measurable* (*Mi i*)
and $(\sum n.\ (\int^{+} x.\ norm\ (f\ (from\text{-}n\ n,\ x))\ \partial(Mi\ (from\text{-}n\ n)))) < \infty$
shows *integrable* (*coPiM I Mi*) *f*
⟨*proof*⟩

lemmas *integrable-coPiM-countable-infiniteI'* = *integrable-coPiM-countable-infiniteI*[*OF bij-betw-from-nat-into*]
lemmas *integrable-coPiM-natI* = *integrable-coPiM-countable-infiniteI*[*OF bij-id*, *simplified id-apply*]

lemma *integral-coPiM-countable-infinite*:
fixes f :: - ⇒ *'b*::{*banach*, *second-countable-topology*}
assumes *bij-betw from-n* (*UNIV* :: *nat set*) *I integrable* (*coPiM I Mi*) *f*
shows $(\int x.\ f\ x\ \partial(coPiM\ I\ Mi)) = (\sum n.\ (\int x.\ f\ (from\text{-}n\ n,\ x)\ \partial(Mi\ (from\text{-}n\ n))))$ (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemmas *integral-coPiM-countable-infinite'* = *integral-coPiM-countable-infinite*[*OF bij-betw-from-nat-into*]
lemmas *integral-coPiM-nat* = *integral-coPiM-countable-infinite*[*OF bij-id*,*simplified id-apply*]

3.8 Finiteness

lemma *finite-measure-coPiM*:
  assumes *finite I* $\bigwedge i.\ i \in I \Longrightarrow$ *finite-measure* (*Mi i*)
  shows *finite-measure* (*coPiM I Mi*)
  ⟨*proof*⟩

3.9 σ-Finiteness

lemma *sigma-finite-measure-coPiM*:
  assumes *countable I* $\bigwedge i.\ i \in I \Longrightarrow$ *sigma-finite-measure* (*Mi i*)
  shows *sigma-finite-measure* (*coPiM I Mi*)
⟨*proof*⟩

end

4 Additional Properties

theory *Coproduct-Measure-Additional*
  imports *Coproduct-Measure*
    Standard-Borel-Spaces.StandardBorel
    S-Finite-Measure-Monad.Kernels
    S-Finite-Measure-Monad.Measure-QuasiBorel-Adjunction
begin

4.1 s-Finiteness

lemma *s-finite-measure-copair-measure*:
  assumes *s-finite-measure M s-finite-measure N*
  shows *s-finite-measure* (*copair-measure M N*)
⟨*proof*⟩

lemma *s-finite-measure-coPiM*:
  assumes *countable I* $\bigwedge i.\ i \in I \Longrightarrow$ *s-finite-measure* (*Mi i*)
  shows *s-finite-measure* (*coPiM I Mi*)
⟨*proof*⟩

4.2 Standardness

lemma *standard-borel-copair-measure*:
  assumes *standard-borel M standard-borel N*
  shows *standard-borel* ($M \bigoplus_M N$)
⟨*proof*⟩

corollary
 shows *standard-borel-ne-copair-measure1*: *standard-borel-ne M* $\Longrightarrow$ *standard-borel* $N \Longrightarrow$ *standard-borel-ne* ($M \bigoplus_M N$)
  and *standard-borel-ne-copair-measure2*: *standard-borel M* $\Longrightarrow$ *standard-borel-ne* $N \Longrightarrow$ *standard-borel-ne* ($M \bigoplus_M N$)

and *standard-borel-ne-copair-measure*: *standard-borel-ne* $M \Longrightarrow$ *standard-borel-ne* $N \Longrightarrow$ *standard-borel-ne* $(M \bigoplus_M N)$
⟨*proof*⟩

lemma *standard-borel-coPiM*:
assumes *countable* I $\bigwedge i.\ i \in I \Longrightarrow$ *standard-borel* (*Mi* i)
shows *standard-borel* (*coPiM I Mi*)
⟨*proof*⟩

lemma *standard-borel-ne-coPiM*:
assumes *countable* I $\bigwedge i.\ i \in I \Longrightarrow$ *standard-borel* (*Mi* i)
and $i \in I$ *space* (*Mi* i) $\neq \{\}$
shows *standard-borel-ne* (*coPiM I Mi*)
⟨*proof*⟩

4.3 Relationships with Quasi-Borel Spaces

Proposition19(3) [1]

lemma *r-preserve-copair*: *measure-to-qbs* (*copair-measure M N*) = *measure-to-qbs* $M \bigoplus_Q$ *measure-to-qbs* N
⟨*proof*⟩

lemma *r-preserve-coproduct*:
assumes *countable* I
shows *measure-to-qbs* (*coPiM I M*) = ($\amalg_Q$ $i \in I$. *measure-to-qbs* (M i))
⟨*proof*⟩

end

References

[1] C. Heunen, O. Kammar, S. Staton, and H. Yang. A convenient category for higher-order probability theory. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '17. IEEE Press, 2017.