

# Disintegration Theorem

Michikazu Hirata

April 21, 2024

## Abstract

We formalize mixture and disintegration of measures. This entry is a formalization of Chapter 14.D of the book by Baccelli et.al. [1]. The main result is the disintegration theorem: let  $(X, \Sigma_X)$  be a measurable space,  $(Y, \Sigma_Y)$  be a standard Borel space,  $\nu$  be a  $\sigma$ -finite measure on  $X \times Y$ , and  $\nu_X$  be the marginal measure on  $X$  defined by  $\nu_X(A) = \nu(A \times Y)$ . Assume that  $\nu_X$  is  $\sigma$ -finite, then there exists a probability kernel  $\kappa$  from  $X$  to  $Y$  such that

$$\nu(A \times B) = \int_A \kappa_x(B) \nu_X(dx), \quad A \in \Sigma_X, B \in \Sigma_Y.$$

Such a probability kernel is unique  $\nu_X$ -almost everywhere.

## Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Lemmas</b>   | <b>1</b> |
| 1.1      | Lemmas . . . . .  | 2        |
| 1.2      | Equivalence of Measures . . . . .                           | 8        |
| <b>2</b> | <b>Disintegration Theorem</b>                               | <b>9</b> |
| 2.1      | Definition 14.D.2. (Mixture and Disintegration) . . . . .   | 9        |
| 2.2      | Lemma 14.D.3. . . . .                                       | 15       |
| 2.3      | Theorem 14.D.4. (Measure Mixture Theorem) . . . . .         | 17       |
| 2.4      | Marginal Measure . . . . .                                  | 31       |
| 2.5      | Lemma 14.D.6. . . . .                                       | 34       |
| 2.6      | Theorem 14.D.10. (Measure Disintegration Theorem) . . . . . | 45       |
| 2.7      | Lemma 14.D.12. . . . .                                      | 60       |
| 2.8      | Lemma 14.D.13. . . . .                                      | 63       |
| 2.9      | Theorem 14.D.14. . . . .                                    | 64       |

## 1 Lemmas

```
theory Lemmas-Disintegration
  imports Standard-Borel-Spaces.StandardBorel
begin
```

## 1.1 Lemmas

**lemma** *semiring-of-sets-binary-product-sets*[simp]:

*semiring-of-sets* (space  $X \times$  space  $Y$ )  $\{a \times b \mid a \in \text{sets } X \wedge b \in \text{sets } Y\}$

**proof**

**show**  $\{a \times b \mid a \in \text{sets } X \wedge b \in \text{sets } Y\} \subseteq \text{Pow}(\text{space } X \times \text{space } Y)$

**using** *pair-measure-closed* **by** *blast*

**next**

**fix**  $c \ d$

**assume**  $c \in \{a \times b \mid a \in \text{sets } X \wedge b \in \text{sets } Y\}$   $d \in \{a \times b \mid a \in \text{sets } X \wedge b \in \text{sets } Y\}$

**then obtain**  $ac \ bc \ ad \ bd$  **where**

$c = ac \times bc$   $ac \in \text{sets } X$   $bc \in \text{sets } Y$   $d = ad \times bd$   $ad \in \text{sets } X$   $bd \in \text{sets } Y$

**by** *auto*

**thus**  $c \cap d \in \{a \times b \mid a \in \text{sets } X \wedge b \in \text{sets } Y\}$

**by**(*auto intro!*:  $exI$ [**where**  $x=ac \cap ad$ ]  $exI$ [**where**  $x=bc \cap bd$ ])

**next**

**fix**  $c \ d$

**assume**  $c \in \{a \times b \mid a \in \text{sets } X \wedge b \in \text{sets } Y\}$   $d \in \{a \times b \mid a \in \text{sets } X \wedge b \in \text{sets } Y\}$

**then obtain**  $ac \ bc \ ad \ bd$  **where** *cd*:

$c = ac \times bc$   $ac \in \text{sets } X$   $bc \in \text{sets } Y$   $d = ad \times bd$   $ad \in \text{sets } X$   $bd \in \text{sets } Y$

**by** *auto*

**then have**  $eq1:c - d = ((ac - ad) \times (bc - bd)) \cup ((ac - ad) \times (bc \cap bd)) \cup ((ac \cap ad) \times (bc - bd))$

**by** *blast*

**obtain**  $a1$  **where**  $a1: a1 \subseteq \text{sets } X$  *finite*  $a1$  *disjoint*  $a1$   $ac - ad = \bigcup a1$

**using** *cd sets.Diff-cover*[*of*  $ac \ X \ ad$ ] **by** *auto*

**obtain**  $a2$  **where**  $a2: a2 \subseteq \text{sets } Y$  *finite*  $a2$  *disjoint*  $a2$   $bc - bd = \bigcup a2$

**using** *cd sets.Diff-cover*[*of*  $bc \ Y \ bd$ ] **by** *auto*

**define**  $A1 \ A2 \ A3$

**where** *A1-def*:  $A1 \equiv \{a \times b \mid a \in a1 \wedge b \in a2\}$

**and** *A2-def*:  $A2 \equiv \{a \times (bc \cap bd) \mid a \in a1\}$

**and** *A3-def*:  $A3 \equiv \{(ac \cap ad) \times b \mid b \in a2\}$

**have** *disj*: *disjoint* ( $A1 \cup A2 \cup A3$ )

**proof** –

**have** [*simp*]: *disjoint*  $A1$

**proof**

**fix**  $x \ y$

**assume**  $x \in A1$   $y \in A1$   $x \neq y$

**then obtain**  $xa \ xb \ ya \ yb$  **where**  $xy: x = xa \times xb$   $xa \in a1$   $xb \in a2$   $y = ya \times yb$   $ya \in a1$   $yb \in a2$

**by**(*auto simp*: *A1-def*)

**with**  $\langle x \neq y \rangle$  **consider**  $xa \neq ya \mid xb \neq yb$  **by** *auto*

**thus** *disjnt*  $x \ y$

**proof** *cases*

**case**  $1$

**then have**  $xa \cap ya = \{\}$

**using**  $a1(3)$   $xy$  **by**(*auto simp*: *disjoint-def*)

**thus** *?thesis*

```

    by(auto simp: xy disjnt-def)
  next
  case 2
  then have  $xb \cap yb = \{\}$ 
    using a2(3) xy by(auto simp: disjoint-def)
  thus ?thesis
    by(auto simp: xy disjnt-def)
  qed
  have [simp]: disjoint A2
  proof
    fix x y
    assume  $x \in A2 \ y \in A2 \ x \neq y$ 
    then obtain xa ya where  $xy: x = xa \times (bc \cap bd) \ xa \in a1 \ y = ya \times (bc \cap$ 
  bd)  $ya \in a1$ 
      by(auto simp: A2-def)
    with a1(3)  $\langle x \neq y \rangle$  have  $xa \cap ya = \{\}$ 
      by(auto simp: disjoint-def)
    thus disjoint x y
      by(auto simp: xy disjnt-def)
  qed
  have [simp]: disjoint A3
  proof
    fix x y
    assume  $x \in A3 \ y \in A3 \ x \neq y$ 
    then obtain xb yb where  $xy: x = (ac \cap ad) \times xb \ xb \in a2 \ y = (ac \cap ad) \times$ 
  yb  $yb \in a2$ 
      by(auto simp: A3-def)
    with a2(3)  $\langle x \neq y \rangle$  have  $xb \cap yb = \{\}$ 
      by(auto simp: disjoint-def)
    thus disjoint x y
      by(auto simp: xy disjnt-def)
  qed
  show ?thesis
    by(auto intro!: disjoint-union) (insert a1 a2, auto simp: A1-def A2-def A3-def)
  qed
  have fin: finite (A1  $\cup$  A2  $\cup$  A3)
    using a1 a2 by (auto simp: A1-def A2-def A3-def finite-image-set2)
  have cdeq:  $c - d = \bigcup (A1 \cup A2 \cup A3)$ 
  proof -
    have [simp]:  $\bigcup a1 \times \bigcup a2 = \bigcup A1 \cup a1 \times (bc \cap bd) = \bigcup A2 (ac \cap ad)$ 
   $\times \bigcup a2 = \bigcup A3$ 
      by (auto simp: A1-def A2-def A3-def)
    show ?thesis
      using a1(4) a2(4) by(simp add: eq1)
  qed
  have  $A1 \cup A2 \cup A3 \subseteq \{a \times b \mid a \ b. \ a \in \text{sets } X \wedge b \in \text{sets } Y\}$ 
    using a1(1) a2(1) cd by(auto simp: A1-def A2-def A3-def)
  with fin disj cdeq show  $\exists C \subseteq \{a \times b \mid a \ b. \ a \in \text{sets } X \wedge b \in \text{sets } Y\}. \text{finite } C \wedge$ 

```

disjoint  $C \wedge c - d = \bigcup C$   
 by (auto intro!: exI[where x=A1  $\cup$  A2  $\cup$  A3])  
 qed auto

**lemma** sets-pair-restrict-space:

sets (restrict-space  $X A \otimes_M$  restrict-space  $Y B$ ) = sets (restrict-space ( $X \otimes_M Y$ ) ( $A \times B$ ))  
 (is ?lhs = ?rhs)

**proof** -

have ?lhs = sigma-sets (space (restrict-space  $X A$ )  $\times$  space (restrict-space  $Y B$ ))  
 $\{a \times b \mid a b. a \in \text{sets (restrict-space } X A) \wedge b \in \text{sets (restrict-space } Y B)\}$

by(simp add: sets-pair-measure)

also have ... = sigma-sets (space (restrict-space  $X A$ )  $\times$  space (restrict-space  $Y B$ ))  
 $\{a \times b \cap \text{space (restrict-space } X A) \times \text{space (restrict-space } Y B) \mid a b. a \in \text{sets } X \wedge b \in \text{sets } Y\}$

**proof** -

have  $\{a \times b \mid a b. a \in \text{sets (restrict-space } X A) \wedge b \in \text{sets (restrict-space } Y B)\}$   
 $= \{a \times b \cap \text{space (restrict-space } X A) \times \text{space (restrict-space } Y B) \mid a b. a \in \text{sets } X \wedge b \in \text{sets } Y\}$

unfolding space-restrict-space sets-restrict-space

**proof** safe

fix xa xb

show xa  $\in$  sets  $X \implies xb \in$  sets  $Y \implies$

$\exists a b. (A \cap xa) \times (B \cap xb) = a \times b \cap (A \cap \text{space } X) \times (B \cap \text{space } Y)$

$\wedge a \in \text{sets } X \wedge b \in \text{sets } Y$

by(auto intro!: exI[where x=xa] exI[where x=xb] dest:sets.sets-into-space)

next

fix a b

show a  $\in$  sets  $X \implies b \in$  sets  $Y \implies$

$\exists aa ba. a \times b \cap (A \cap \text{space } X) \times (B \cap \text{space } Y) = aa \times ba \wedge aa \in$

sets.restricted-space  $X A \wedge ba \in$  sets.restricted-space  $Y B$

by(auto intro!: exI[where x=a  $\cap$  A] exI[where x=b  $\cap$  B] dest:sets.sets-into-space)

qed

thus ?thesis by simp

qed

also have ... = sigma-sets (space (restrict-space  $X A$ )  $\times$  space (restrict-space  $Y B$ ))  
 $\{(\lambda x. x) -' c \cap \text{space (restrict-space } X A) \times \text{space (restrict-space } Y B) \mid c. c \in \{a \times b \mid a b. a \in \text{sets } X \wedge b \in \text{sets } Y\}\}$

**proof** -

have  $\{a \times b \cap \text{space (restrict-space } X A) \times \text{space (restrict-space } Y B) \mid a b. a \in \text{sets } X \wedge b \in \text{sets } Y\} = \{(\lambda x. x) -' c \cap \text{space (restrict-space } X A) \times \text{space (restrict-space } Y B) \mid c. c \in \{a \times b \mid a b. a \in \text{sets } X \wedge b \in \text{sets } Y\}\}$

by auto

thus ?thesis by simp

qed

also have ... =  $\{(\lambda x. x) -' c \cap \text{space (restrict-space } X A) \times \text{space (restrict-space } Y B) \mid c. c \in \text{sigma-sets (space } X \times \text{space } Y) \{a \times b \mid a b. a \in \text{sets } X \wedge b \in \text{sets } Y\}\}$

by(rule sigma-sets-vimage-commute[symmetric]) (auto simp: space-restrict-space)

**also have** ... =  $\{c \cap (A \cap \text{space } X) \times (B \cap \text{space } Y) \mid c. c \in \text{sets } (X \otimes_M Y)\}$   
**by**(simp add: space-restrict-space sets-pair-measure)  
**also have** ... =  $\{c \cap A \times B \mid c. c \in \text{sets } (X \otimes_M Y)\}$   
**using** sets.sets-into-space[of -  $X \otimes_M Y$ ,simplified space-pair-measure] **by** blast  
**also have** ... = ?rhs  
**by**(auto simp: sets-restrict-space)  
**finally show** ?thesis .  
**qed**

**lemma** restrict-space-space[simp]: restrict-space  $M$  (space  $M$ ) =  $M$   
**by**(auto intro!: measure-eqI simp: sets-restrict-space emeasure-restrict-space sets.sets-into-space)

**lemma** atMostq-Int-stable:  
Int-stable  $\{\{..r\} \mid r::\text{real}. r \in \mathbb{Q}\}$   
**by**(auto simp: Int-stable-def min-def)

**lemma** rborel-eq-atMostq:  
borel = sigma UNIV  $\{\{..r\} \mid r::\text{real}. r \in \mathbb{Q}\}$   
**proof**(safe intro!: borel-eq-sigmaII[OF borel-eq-atMost,where F=id,simplified])  
**fix** a :: real  
**interpret** s: sigma-algebra UNIV sigma-sets UNIV  $\{\{..r\} \mid r. r \in \mathbb{Q}\}$   
**by**(auto intro!: sigma-algebra-sigma-sets)  
**have** [simp]:  $\{..a\} = (\bigcap ((\lambda r. \{..r\}) ' \{r. r \in \mathbb{Q} \wedge a \leq r\}))$   
**by** auto (metis Rats-dense-in-real less-le-not-le nle-le)  
**show**  $\{..a\} \in \text{sigma-sets UNIV } \{\{..r\} \mid r. r \in \mathbb{Q}\}$   
**using** countable-Collect countable-rat Rats-no-top-le  
**by**(auto intro!: s.countable-INT')  
**qed** auto

**corollary** rborel-eq-atMostq-sets:  
sets borel = sigma-sets UNIV  $\{\{..r\} \mid r::\text{real}. r \in \mathbb{Q}\}$   
**by**(simp add: rborel-eq-atMostq)

**lemma** mono-absolutely-continuous:  
**assumes** sets  $\mu = \text{sets } \nu \wedge A. A \in \text{sets } \mu \implies \mu A \leq \nu A$   
**shows** absolutely-continuous  $\nu \mu$   
**by**(auto simp: absolutely-continuous-def) (metis assms(1) assms(2) fmeasurableD fmeasurableI-null-sets le-zero-eq null-setsD1 null-setsI)

**lemma** ex-measure-countable-space:  
**assumes** countable (space  $X$ )  
**and** sets  $X = \text{Pow}$  (space  $X$ )  
**shows**  $\exists \mu. \text{sets } \mu = \text{sets } X \wedge (\forall x \in \text{space } X. \mu \{x\} = f x)$   
**proof** –  
**define**  $\mu$  **where**  $\mu \equiv \text{extend-measure}$  (space  $X$ ) (space  $X$ )  $(\lambda x. \{x\}) f$   
**have** s:sets  $\mu = \text{sets } X$   
**using** sets-extend-measure[of  $\lambda x. \{x\}$  space  $X$  space  $X$ ] sigma-sets-singletons[OF assms(1)]  
**by**(auto simp add:  $\mu$ -def assms(2))

```

show ?thesis
proof(safe intro!: exI[where  $x=\mu$ ])
  fix  $x$ 
  assume  $x:x \in \text{space } X$ 
  show  $\mu \{x\} = f x$ 
  proof(cases finite (space X))
    case  $fin:True$ 
    then have sets-fin: $x \in \text{sets } \mu \implies \text{finite } x$  for  $x$ 
      by(auto intro!: rev-finite-subset[OF fin] sets.sets-into-space simp: s)
    define  $\mu'$  where  $\mu' \equiv (\lambda A. \sum_{x \in A}. f x)$ 
    show ?thesis
    proof(rule emeasure-extend-measure[of  $\mu$  space X space X - f  $\mu' x$ ])
      show countably-additive (sets  $\mu$ )  $\mu'$ 
      using fin sets-fin
      by(auto intro!: sets.countably-additiveI-finite simp: sets-eq-imp-space-eq[OF
s] positive-def  $\mu'$ -def additive-def comm-monoid-add-class.sum.union-disjoint)
      qed(auto simp:  $x$   $\mu$ -def  $\mu'$ -def positive-def)
    next
    case  $inf:False$ 
    define  $\mu'$  where  $\mu' \equiv (\lambda A. \sum n. \text{if from-nat-into (space X) } n \in A \text{ then } f$ 
(from-nat-into (space X)  $n$ ) else 0)
    show ?thesis
    proof(rule emeasure-extend-measure[of  $\mu$  space X space X - f  $\mu' x$ ])
      fix  $i$ 
      assume  $i \in \text{space } X$ 
      then obtain  $n$  where  $n:\text{from-nat-into (space X) } n = i$ 
      using bij-betw-from-nat-into[OF assms(1) inf] by (meson f-the-inv-into-f-bij-betw)
      then have  $\mu' \{i\} = (\sum m. \text{if } m = n \text{ then } f (\text{from-nat-into (space X) } n)$ 
else 0)
      using from-nat-into-inj-infinite[OF assms(1) inf]
      by(auto simp:  $\mu'$ -def) metis
      also have ... =  $(\sum m. \text{if } (m + (\text{Suc } n)) = n \text{ then } f (\text{from-nat-into (space$ 
X)  $n$ ) else 0) + (\sum  $m < \text{Suc } n. \text{if } m = n \text{ then } f (\text{from-nat-into (space X) } n)$ 
else 0)
      by(rule suminf-offset) auto
      also have ... =  $f i$ 
      by(auto simp:  $n$ )
      finally show  $\mu' \{i\} = f i$  .
    next
    show countably-additive (sets  $\mu$ )  $\mu'$ 
    proof(rule countably-additiveI)
      fix  $A :: \text{nat} \Rightarrow -$ 
      assume  $h:\text{range } A \subseteq \text{sets } \mu$  disjoint-family  $A \cup (\text{range } A) \in \text{sets } \mu$ 
      show  $(\sum i. \mu' (A i)) = \mu' (\bigcup (\text{range } A))$ 
      proof -
      have  $(\sum i. \mu' (A i)) = (\int^+ i. \mu' (A i) \partial(\text{count-space UNIV}))$ 
      by(simp add: nn-integral-count-space-nat)
      also have ... =  $(\int^+ i. (\sum n. \text{if from-nat-into (space X) } n \in A \text{ then } f$ 
(from-nat-into (space X)  $n$ ) else 0)  $\partial(\text{count-space UNIV}))$ 

```

```

      by(simp add: μ'-def)
      also have ... = (∑ n. (∫+ i. (if from-nat-into (space X) n ∈ A i then f
(from-nat-into (space X) n) else 0) ∂(count-space UNIV)))
      by(simp add: nn-integral-suminf)
      also have ... = (∑ n. (∫+ i. f (from-nat-into (space X) n) * indicator
(A i) (from-nat-into (space X) n) ∂(count-space UNIV)))
      by(auto intro!: suminf-cong nn-integral-cong)
      also have ... = (∑ n. (∑ i. f (from-nat-into (space X) n) * indicator
(A i) (from-nat-into (space X) n)))
      by(simp add: nn-integral-count-space-nat)
      also have ... = (∑ n. f (from-nat-into (space X) n) * indicator (∪
(range A)) (from-nat-into (space X) n))
      by(simp add: suminf-indicator[OF h(2)])
      also have ... = μ' (∪ (range A))
      by(auto simp: μ'-def intro!: suminf-cong)
      finally show ?thesis .
    qed
  qed
  qed(auto simp: x μ-def μ'-def positive-def)
  qed
  qed(simp-all add: s)
  qed

```

**lemma** *ex-prob-space-countable*:

```

  assumes space X ≠ {} countable (space X)
  and sets X = Pow (space X)
  shows ∃ μ. sets μ = sets X ∧ prob-space μ
proof(cases finite (space X))
  case fin: True
  define n where n ≡ card (space X)
  with fin assms(1) have n: 0 < n
  by(simp add: card-gt-0-iff)
  obtain μ where μ: sets μ = sets X ∧ x. x ∈ space X ⇒ μ {x} = ennreal (1 /
real n)
  using ex-measure-countable-space[OF assms(2,3)] by meson
  then have sets-fin: x ∈ sets μ ⇒ finite x for x
  by(auto intro!: rev-finite-subset[OF fin] sets.sets-into-space)
  show ?thesis
  proof(safe intro!: exI[where x=μ])
  show prob-space μ
  proof
  have emeasure μ (space μ) = (∑ a∈space μ. ennreal (1/n))
  using emeasure-eq-sum-singleton[OF sets-fin[OF sets.top], of μ] assms(3) μ
  by auto
  also have ... = of-nat n * ennreal (1 / real n)
  using μ(2) sets-eq-imp-space-eq[OF μ(1)] by(simp add: n-def)
  also have ... = 1
  using n by(auto simp: ennreal-of-nat-eq-real-of-nat) (metis ennreal-1 en-
nreal-mult'' mult commute nonzero-eq-divide-eq not-gr0 of-nat-0-eq-iff of-nat-0-le-iff)

```

```

    finally show emeasure  $\mu$  (space  $\mu$ ) = 1 .
  qed
qed(use  $\mu$  in auto)
next
  case inf:False
  obtain  $\mu$  where  $\mu$ : sets  $\mu$  = sets  $X \wedge x. x \in \text{space } X \implies \mu \{x\} = (1/2) \wedge \text{Suc}$ 
  (to-nat-on (space  $X$ )  $x$ )
  using ex-measure-countable-space[OF assms(2,3),of  $\lambda x. (1/2) \wedge \text{Suc}$  (to-nat-on
  (space  $X$ )  $x$ )] by auto
  show ?thesis
  proof(safe intro!: exI[where  $x=\mu$ ])
    show prob-space  $\mu$ 
  proof
    have emeasure  $\mu$  (space  $\mu$ ) = emeasure  $\mu$  ( $\bigcup n. \{\text{from-nat-into (space } X) n\}$ )
      by(simp add: sets-eq-imp-space-eq[OF  $\mu(1)$ ] UNION-singleton-eq-range
  assms(1) assms(2))
    also have ... = ( $\sum n. \mu \{\text{from-nat-into (space } X) n\}$ )
      using from-nat-into-inj-infinite[OF assms(2) inf] from-nat-into[OF assms(1)]
  assms(3)
    by(auto intro!: suminf-emeasure[symmetric] simp:  $\mu(1)$  disjoint-family-on-def)
    also have ... = ( $\sum n. (1/2) \wedge \text{Suc } n$ )
      by(simp add:  $\mu(2)$ [OF from-nat-into[OF assms(1)]] to-nat-on-from-nat-into-infinite[OF
  assms(2) inf])
    also have ... = ( $\sum i. \text{ennreal } ((1 / 2) \wedge \text{Suc } i)$ )
      by (metis (mono-tags, opaque-lifting) divide-ennreal divide-pos-pos en-
  nreal-numeral ennreal-power le-less power-0 zero-less-numeral zero-less-one)
    also have ... = 1
      using suminf-ennreal-eq[OF - power-half-series]
      by (metis ennreal-1 zero-le-divide-1-iff zero-le-numeral zero-le-power)
    finally show emeasure  $\mu$  (space  $\mu$ ) = 1 .
  qed
qed(use  $\mu$  in auto)
qed

```

**lemma** AE-I'':

```

  assumes  $N \in \text{null-sets } M$ 
  and  $\bigwedge x. x \in \text{space } M \implies x \notin N \implies P x$ 
  shows AE  $x$  in  $M. P x$ 
  by (metis (no-types, lifting) assms eventually-ae-filter mem-Collect-eq subsetI)

```

**lemma** absolutely-continuous-trans:

```

  assumes absolutely-continuous  $L M$  absolutely-continuous  $M N$ 
  shows absolutely-continuous  $L N$ 
  using assms by(auto simp: absolutely-continuous-def)

```

## 1.2 Equivalence of Measures

**abbreviation** equivalence-measure :: 'a measure  $\implies$  'a measure  $\implies$  bool (infix  $\sim_M$  60)



**where** *equivalence-measure*  $M N \equiv \text{absolutely-continuous } M N \wedge \text{absolutely-continuous } N M$

**lemma** *equivalence-measure-refl*:  $M \sim_M M$   
**by**(*auto simp: absolutely-continuous-def*)

**lemma** *equivalence-measure-sym*:  
**assumes**  $M \sim_M N$   
**shows**  $N \sim_M M$   
**using** *assms* **by** *simp*

**lemma** *equivalence-measure-trans*:  
**assumes**  $M \sim_M N$   $N \sim_M L$   
**shows**  $M \sim_M L$   
**using** *assms* **by**(*auto simp: absolutely-continuous-def*)

**lemma** *equivalence-measureI*:  
**assumes** *absolutely-continuous*  $M N$  *absolutely-continuous*  $N M$   
**shows**  $M \sim_M N$   
**by**(*simp add: assms*)

**end**

## 2 Disintegration Theorem

**theory** *Disintegration*  
**imports** *S-Finite-Measure-Monad.Kernels*  
*Lemmas-Disintegration*  
**begin**

### 2.1 Definition 14.D.2. (Mixture and Disintegration)

**context** *measure-kernel*  
**begin**

**definition** *mixture-of* :: [*'a*  $\times$  *'b*] *measure*, [*'a* *measure*]  $\Rightarrow$  *bool* **where**  
*mixture-of*  $\nu \mu \longleftrightarrow \text{sets } \nu = \text{sets } (X \otimes_M Y) \wedge \text{sets } \mu = \text{sets } X \wedge (\forall C \in \text{sets } (X \otimes_M Y). \nu C = (\int^+ x. \int^+ y. \text{indicator } C (x,y) \partial(\kappa x) \partial\mu))$

**definition** *disintegration* :: [*'a*  $\times$  *'b*] *measure*, [*'a* *measure*]  $\Rightarrow$  *bool* **where**  
*disintegration*  $\nu \mu \longleftrightarrow \text{sets } \nu = \text{sets } (X \otimes_M Y) \wedge \text{sets } \mu = \text{sets } X \wedge (\forall A \in \text{sets } X. \forall B \in \text{sets } Y. \nu (A \times B) = (\int^+ x \in A. (\kappa x B) \partial\mu))$

**lemma** *disintegrationI*:  
**assumes**  $\text{sets } \nu = \text{sets } (X \otimes_M Y)$   $\text{sets } \mu = \text{sets } X$   
**and**  $\bigwedge A B. A \in \text{sets } X \Longrightarrow B \in \text{sets } Y \Longrightarrow \nu (A \times B) = (\int^+ x \in A. (\kappa x B) \partial\mu)$   
**shows** *disintegration*  $\nu \mu$   
**by**(*simp add: disintegration-def assms*)

**lemma** *mixture-of-disintegration*:

**assumes** *mixture-of*  $\nu$   $\mu$   
**shows** *disintegration*  $\nu$   $\mu$   
**unfolding** *disintegration-def*

**proof** *safe*

**fix**  $A$   $B$

**assume** [*simp*]:  $A \in \text{sets } X$   $B \in \text{sets } Y$

**have** [*simp,measurable-cong*]: *sets*  $\mu = \text{sets } X$  *space*  $\mu = \text{space } X$

**using** *assms* **by**(*auto simp: mixture-of-def intro!: sets-eq-imp-space-eq*)

**have**  $A \times B \in \text{sets } (X \otimes_M Y)$  **by** *simp*

**with** *assms* **have**  $\nu (A \times B) = (\int^{+x}. \int^{+y}. \text{indicator } (A \times B) (x,y) \partial(\kappa x) \partial\mu)$

**by**(*simp add: mixture-of-def*)

**also** **have**  $\dots = (\int^{+x}. \int^{+y}. \text{indicator } A x * \text{indicator } B y \partial(\kappa x) \partial\mu)$

**by**(*simp add: indicator-times*)

**also** **have**  $\dots = (\int^{+x \in A}. (\kappa x B) \partial\mu)$

**by**(*auto intro!: nn-integral-cong simp: kernel-sets nn-integral-cmult-indicator mult commute*)

**finally** **show** *emeasure*  $\nu (A \times B) = (\int^{+x \in A}. \text{emeasure } (\kappa x) B \partial\mu)$  .

**qed**(*use assms[simplified mixture-of-def] in auto*)

**lemma**

**shows** *mixture-of-sets-eq*: *mixture-of*  $\nu$   $\mu \implies \text{sets } \nu = \text{sets } (X \otimes_M Y)$  *mixture-of*  $\nu$   $\mu \implies \text{sets } \mu = \text{sets } X$

**and** *mixture-of-space-eq*: *mixture-of*  $\nu$   $\mu \implies \text{space } \nu = \text{space } (X \otimes_M Y)$  *mixture-of*  $\nu$   $\mu \implies \text{space } \mu = \text{space } X$

**and** *disintegration-sets-eq*: *disintegration*  $\nu$   $\mu \implies \text{sets } \nu = \text{sets } (X \otimes_M Y)$  *disintegration*  $\nu$   $\mu \implies \text{sets } \mu = \text{sets } X$

**and** *disintegration-space-eq*: *disintegration*  $\nu$   $\mu \implies \text{space } \nu = \text{space } (X \otimes_M Y)$  *disintegration*  $\nu$   $\mu \implies \text{space } \mu = \text{space } X$

**by**(*auto simp: mixture-of-def disintegration-def intro!: sets-eq-imp-space-eq*)

**lemma**

**shows** *mixture-ofD*: *mixture-of*  $\nu$   $\mu \implies C \in \text{sets } (X \otimes_M Y) \implies \nu C = (\int^{+x}. \int^{+y}. \text{indicator } C (x,y) \partial(\kappa x) \partial\mu)$

**and** *disintegrationD*: *disintegration*  $\nu$   $\mu \implies A \in \text{sets } X \implies B \in \text{sets } Y \implies \nu (A \times B) = (\int^{+x \in A}. (\kappa x B) \partial\mu)$

**by**(*auto simp: mixture-of-def disintegration-def*)

**lemma** *disintegration-restrict-space*:

**assumes** *disintegration*  $\nu$   $\mu$   $A \cap \text{space } X \in \text{sets } X$

**shows** *measure-kernel.disintegration* (*restrict-space*  $X$   $A$ )  $Y$   $\kappa$  (*restrict-space*  $\nu$  ( $A \times \text{space } Y$ )) (*restrict-space*  $\mu$   $A$ )

**proof**(*rule measure-kernel.disintegrationI[OF restrict-measure-kernel[of A]]*)

**have** *sets* (*restrict-space*  $\nu$  ( $A \times \text{space } Y$ )) = *sets* (*restrict-space* ( $X \otimes_M Y$ ) ( $A \times \text{space } Y$ ))

**by**(*auto simp: disintegration-sets-eq[OF assms(1)] intro!: sets-restrict-space-cong*)

**also** **have**  $\dots = \text{sets } (\text{restrict-space } X A \otimes_M Y)$

**using** *sets-pair-restrict-space[of X A Y space Y]*

```

    by simp
  finally show sets (restrict-space  $\nu$  (A  $\times$  space Y)) = sets (restrict-space X A
 $\otimes_M$  Y) .
next
  show sets (restrict-space  $\mu$  A) = sets (restrict-space X A)
  by(auto simp: disintegration-sets-eq[OF assms(1)] intro!: sets-restrict-space-cong)
next
  fix a b
  assume h:a  $\in$  sets (restrict-space X A) b  $\in$  sets Y
  then have restrict-space  $\nu$  (A  $\times$  space Y) (a  $\times$  b) =  $\nu$  (a  $\times$  b)
    using sets.sets-into-space
  by(auto intro!: emeasure-restrict-space simp: disintegration-space-eq[OF assms(1)]
  disintegration-sets-eq[OF assms(1)] sets-restrict-space)
    (metis Sigma-Int-distrib1 assms(2) pair-measureI sets.top space-pair-measure)
  also have ... = ( $\int^{+x \in a}$ . emeasure ( $\kappa$  x) b  $\partial\mu$ )
    using sets-restrict-space-iff[OF assms(2)] h assms(1)
  by(auto simp: disintegration-def)
  also have ... = ( $\int^{+x \in A}$ . (emeasure ( $\kappa$  x) b * indicator a x)  $\partial\mu$ )
    using h(1) by(auto intro!: nn-integral-cong simp: sets-restrict-space)
    (metis IntD1 indicator-simps(1) indicator-simps(2) mult.comm-neutral
  mult-zero-right)
  also have ... = ( $\int^{+x \in a}$ . emeasure ( $\kappa$  x) b  $\partial$ restrict-space  $\mu$  A)
    by (metis (no-types, lifting) assms disintegration-sets-eq(2) disintegration-space-eq(2)
  nn-integral-cong nn-integral-restrict-space)
  finally show restrict-space  $\nu$  (A  $\times$  space Y) (a  $\times$  b) = ( $\int^{+x \in a}$ . emeasure ( $\kappa$ 
  x) b  $\partial$ restrict-space  $\mu$  A) .
qed

end

context subprob-kernel
begin
lemma countable-disintegration-AE-unique:
  assumes countable (space Y) and [measurable-cong]:sets Y = Pow (space Y)
    and subprob-kernel X Y  $\kappa'$  sigma-finite-measure  $\mu$ 
    and disintegration  $\nu$   $\mu$  measure-kernel.disintegration X Y  $\kappa'$   $\nu$   $\mu$ 
  shows AE x in  $\mu$ .  $\kappa$  x =  $\kappa'$  x
proof -
  interpret  $\kappa'$ : subprob-kernel X Y  $\kappa'$  by fact
  interpret s: sigma-finite-measure  $\mu$  by fact
  have sets-eq[measurable-cong]: sets  $\mu$  = sets X sets  $\nu$  = sets (X  $\otimes_M$  Y)
    using assms(5) by(auto simp: disintegration-def)
  have 1:AE x in  $\mu$ .  $\forall y \in$  space Y.  $\kappa$  x {y} =  $\kappa'$  x {y}
    unfolding AE-ball-countable[OF assms(1)]
  proof
    fix y
    assume y: y  $\in$  space Y
    show AE x in  $\mu$ . emeasure ( $\kappa$  x) {y} = emeasure ( $\kappa'$  x) {y}
    proof(rule s.sigma-finite)

```

```

fix J :: nat => -
assume J:range J ⊆ sets μ ∪ (range J) = space μ ∧ i. emeasure μ (J i) ≠
∞
from y have [measurable]: (λx. κ x {y}) ∈ borel-measurable X (λx. κ' x {y})
∈ borel-measurable X
  using emeasure-measurable κ'.emeasure-measurable by auto
define A where A ≡ {x ∈ space μ. κ x {y} ≤ κ' x {y}}
have [measurable]: A ∈ sets μ
  by(auto simp: A-def)
have A: ∧x. x ∈ space μ ⇒ x ∉ A ⇒ κ' x {y} ≤ κ x {y}
  by(auto simp: A-def)
have 1: AE x∈A in μ. κ x {y} = κ' x {y}
proof -
  have AE x in μ. ∀n. (x ∈ A ∩ J n → κ x {y} = κ' x {y})
  unfolding AE-all-countable
proof
  fix n
  have ninf:(∫+x∈A ∩ J n. (κ x) {y} ∂μ) < ∞
  proof -
  have (∫+x∈A ∩ J n. (κ x) {y} ∂μ) ≤ (∫+x∈A ∩ J n. (κ x) (space Y)
∂μ)
    using kernel-sets y by(auto intro!: nn-integral-mono emeasure-mono
simp: indicator-def disintegration-space-eq(2)[OF assms(5)])
    also have ... ≤ (∫+x∈A ∩ J n. 1 ∂μ)
    using subprob-space by(auto intro!: nn-integral-mono simp: indicator-def
disintegration-space-eq(2)[OF assms(5)])
    also have ... = μ (A ∩ J n)
    using J by simp
    also have ... ≤ μ (J n)
    using J by (auto intro!: emeasure-mono)
    also have ... < ∞
    using J(3)[of n] by (simp add: top.not-eq-extremum)
    finally show ?thesis .
  qed
  have (∫+x. (κ' x) {y} * indicator (A ∩ J n) x - (κ x) {y} * indicator (A
∩ J n) x ∂μ) = (∫+x∈A ∩ J n. (κ' x) {y} ∂μ) - (∫+x∈A ∩ J n. (κ x) {y} ∂μ)
  using J ninf by(auto intro!: nn-integral-diff simp: indicator-def A-def)
  also have ... = 0
  proof -
  have 0: ν ((A ∩ J n) × {y}) = (∫+x∈A ∩ J n. (κ x) {y} ∂μ)
  using J y sets-eq by(auto intro!: disintegrationD[OF assms(5),of A ∩
J n {y}])
  have [simp]: (∫+x∈A ∩ J n. (κ' x) {y} ∂μ) = ν ((A ∩ J n) × {y})
  using J y sets-eq by(auto intro!: κ'.disintegrationD[OF assms(6),of A
∩ J n {y},symmetric])
  show ?thesis
  using ninf by (simp add: 0 diff-eq-0-iff-ennreal)
qed
finally have assm:AE x in μ. (κ' x) {y} * indicator (A ∩ J n) x - (κ x)

```

```

{y} * indicator (A ∩ J n) x = 0
  using J by (simp add: nn-integral-0-iff-AE)
  show AE x ∈ A ∩ J n in μ. (κ x) {y} = (κ' x) {y}
  proof (rule AE-mp[OF assm])
    show AE x in μ. emeasure (κ' x) {y} * indicator (A ∩ J n) x - emeasure
(κ x) {y} * indicator (A ∩ J n) x = 0 → x ∈ A ∩ J n → emeasure (κ x) {y}
= emeasure (κ' x) {y}
    proof -
      {
        fix x
        assume h: (κ' x) {y} - (κ x) {y} = 0 x ∈ A
        have (κ x) {y} = (κ' x) {y}
          using h(2) by (auto intro!: antisym ennreal-minus-eq-0[OF h(1)])
      }
    simp: A-def
  }
  thus ?thesis
    by (auto simp: indicator-def)
  qed
  qed
  hence AE x ∈ A ∩ (⋃ (range J)) in μ. κ x {y} = κ' x {y}
    by auto
  thus ?thesis
    using J(2) by auto
  qed
  have 2: AE x ∈ (space μ - A) in μ. κ x {y} = κ' x {y}
  proof -
    have AE x in μ. ∀ n. x ∈ (space μ - A) ∩ J n → κ x {y} = κ' x {y}
      unfolding AE-all-countable
    proof
      fix n
      have ninf: (∫+ x ∈ (space μ - A) ∩ J n. (κ' x) {y} ∂μ) < ∞
      proof -
        have (∫+ x ∈ (space μ - A) ∩ J n. (κ' x) {y} ∂μ) ≤ (∫+ x ∈ (space μ - A)
∩ J n. (κ' x) (space Y) ∂μ)
          using kernel-sets y by (auto intro!: nn-integral-mono emeasure-mono
simp: indicator-def disintegration-space-eq(2)[OF assms(5)])
        also have ... ≤ (∫+ x ∈ (space μ - A) ∩ J n. 1 ∂μ)
          using κ'.subprob-space by (auto intro!: nn-integral-mono simp: indica-
tor-def disintegration-space-eq(2)[OF assms(5)])
        also have ... = μ ((space μ - A) ∩ J n)
          using J by simp
        also have ... ≤ μ (J n)
          using J by (auto intro!: emeasure-mono)
        also have ... < ∞
          using J(3)[of n] by (simp add: top.not-eq-extremum)
        finally show ?thesis .
      }
    qed
  have (∫+ x. (κ x) {y} * indicator ((space μ - A) ∩ J n) x - (κ' x) {y} *

```

```

indicator ((space  $\mu - A$ )  $\cap$   $J$   $n$ )  $x$   $\partial\mu$ ) = ( $\int^{+x \in (\text{space } \mu - A) \cap J n. (\kappa x) \{y\}}$ 
 $\partial\mu$ ) - ( $\int^{+x \in (\text{space } \mu - A) \cap J n. (\kappa' x) \{y\}}$   $\partial\mu$ )
  using  $J$   $n$  inf  $A$  by(auto intro!: nn-integral-diff simp: indicator-def)
  also have ... = 0
  proof -
    have 0: ( $\int^{+x \in (\text{space } \mu - A) \cap J n. (\kappa' x) \{y\}}$   $\partial\mu$ ) =  $\nu$  (((space  $\mu - A$ )  $\cap$   $J$   $n$ )  $\times$   $\{y\}$ )
      using  $J$   $y$  sets-eq by(auto intro!:  $\kappa'$ .disintegrationD[OF assms(6),of (space  $\mu - A$ )  $\cap$   $J$   $n$   $\{y\}$ ,symmetric])
      have [simp]:  $\nu$  (((space  $\mu - A$ )  $\cap$   $J$   $n$ )  $\times$   $\{y\}$ ) = ( $\int^{+x \in (\text{space } \mu - A) \cap J n. (\kappa x) \{y\}}$   $\partial\mu$ )
        using  $J$   $y$  sets-eq by(auto intro!: disintegrationD[OF assms(5),of (space  $\mu - A$ )  $\cap$   $J$   $n$   $\{y\}$ ])
        show ?thesis
          using  $n$  inf by (simp add: 0 diff-eq-0-iff-ennreal)
    qed
    finally have  $assm: AE$   $x$  in  $\mu. (\kappa x) \{y\} * indicator ((\text{space } \mu - A) \cap J n) x - (\kappa' x) \{y\} * indicator ((\text{space } \mu - A) \cap J n) x = 0$ 
      using  $J$  by(simp add: nn-integral-0-iff-AE)
      show  $AE$   $x \in (\text{space } \mu - A) \cap J n$  in  $\mu. (\kappa x) \{y\} = (\kappa' x) \{y\}$ 
        proof(rule AE-mp[OF assm])
          show  $AE$   $x$  in  $\mu. emeasure (\kappa x) \{y\} * indicator ((\text{space } \mu - A) \cap J n) x - emeasure (\kappa' x) \{y\} * indicator ((\text{space } \mu - A) \cap J n) x = 0 \longrightarrow x \in (\text{space } \mu - A) \cap J n \longrightarrow emeasure (\kappa x) \{y\} = emeasure (\kappa' x) \{y\}$ 
            proof -
              {
                fix  $x$ 
                assume  $h: (\kappa x) \{y\} - (\kappa' x) \{y\} = 0$   $x \in \text{space } \mu$   $x \notin A$ 
                have  $(\kappa x) \{y\} = (\kappa' x) \{y\}$ 
                  using  $A$  [OF  $h(2,3)$ ] by(auto intro!: antisym ennreal-minus-eq-0[OF  $h(1)$ ] simp: A-def)
                }
              thus ?thesis
                by(auto simp: indicator-def)
            qed
          qed
        qed
      hence  $AE$   $x \in (\text{space } \mu - A) \cap (\bigcup (\text{range } J))$  in  $\mu. \kappa x \{y\} = \kappa' x \{y\}$ 
        by auto
      thus ?thesis
        using  $J(2)$  by auto
    qed
  show  $AE$   $x$  in  $\mu. \kappa x \{y\} = \kappa' x \{y\}$ 
    using 1 2 by(auto simp: A-def)
  qed
qed
show ?thesis
proof(rule AE-mp[OF 1])
  {

```

```

fix x
assume x: x ∈ space X
  and h: ∀ y ∈ space Y. κ x {y} = κ' x {y}
  have κ x = κ' x
  by (simp add: κ'.kernel-sets assms h kernel-sets measure-eqI-countable x)
}
thus AE x in μ. (∀ y ∈ space Y. emeasure (κ x) {y} = emeasure (κ' x) {y})
→ κ x = κ' x
by(auto simp: sets-eq-imp-space-eq[OF sets-eq(1)])
qed
qed

end

```

```

lemma(in subprob-kernel) nu-mu-spaceY-le:
  assumes disintegration ν μ A ∈ sets X
  shows ν (A × space Y) ≤ μ A
proof -
  have ν (A × space Y) = (∫+ x ∈ A. (κ x (space Y)) ∂μ)
  using assms by(simp add: disintegration-def)
  also have ... ≤ (∫+ x ∈ A. 1 ∂μ)
  using assms subprob-space by(auto intro!: nn-integral-mono simp: disintegration-space-eq) (metis dual-order.refl indicator-simps(1) indicator-simps(2) mult.commute mult-1 mult-zero-right)
  also have ... = μ A
  using assms by (simp add: disintegration-def)
  finally show ?thesis .
qed

```

```

context prob-kernel
begin

```

```

lemma countable-disintegration-AE-unique-prob:
  assumes countable (space Y) and [measurable-cong]:sets Y = Pow (space Y)
  and prob-kernel X Y κ' sigma-finite-measure μ
  and disintegration ν μ measure-kernel.disintegration X Y κ' ν μ
  shows AE x in μ. κ x = κ' x
  by(auto intro!: countable-disintegration-AE-unique[OF assms(1,2) - assms(4-6)]
  prob-kernel.subprob-kernel assms(3))

```

```

end

```

## 2.2 Lemma 14.D.3.

```

lemma(in prob-kernel) nu-mu-spaceY:
  assumes disintegration ν μ A ∈ sets X
  shows ν (A × space Y) = μ A
proof -
  have ν (A × space Y) = (∫+ x ∈ A. (κ x (space Y)) ∂μ)

```

```

    using assms by(simp add: disintegration-def)
  also have ... = ( $\int^{+x \in A. 1} \partial \mu$ )
    using assms by(auto intro!: nn-integral-cong simp: prob-space disintegration-space-eq)
  also have ... =  $\mu A$ 
    using assms by(simp add: disintegration-def)
  finally show ?thesis .
qed

```

```

corollary(in subprob-kernel) nu-finite:
  assumes disintegration  $\nu$   $\mu$  finite-measure  $\mu$ 
  shows finite-measure  $\nu$ 
proof
  have  $\nu$  (space  $\nu$ ) =  $\nu$  (space ( $X \otimes_M Y$ ))
    using assms by(simp add: disintegration-space-eq)
  also have ...  $\leq \mu$  (space  $\mu$ )
    using assms by(simp add: nu-mu-space-Y-le disintegration-space-eq space-pair-measure)
  finally show  $\nu$  (space  $\nu$ )  $\neq \infty$ 
    using assms(2) by (metis finite-measure.emeasure-finite infinity-enreal-def
neq-top-trans)
qed

```

```

corollary(in subprob-kernel) nu-subprob-space:
  assumes disintegration  $\nu$   $\mu$  subprob-space  $\mu$ 
  shows subprob-space  $\nu$ 
proof
  have  $\nu$  (space  $\nu$ ) =  $\nu$  (space ( $X \otimes_M Y$ ))
    using assms by(simp add: disintegration-space-eq)
  also have ...  $\leq \mu$  (space  $\mu$ )
    using assms by(simp add: nu-mu-space-Y-le disintegration-space-eq space-pair-measure)
  finally show  $\nu$  (space  $\nu$ )  $\leq 1$ 
    using assms(2) order.trans subprob-space.emeasure-space-le-1 by auto
next
  show space  $\nu$   $\neq \{\}$ 
    using Y-not-empty assms by(auto simp: disintegration-space-eq subprob-space-def
subprob-space-axioms-def space-pair-measure)
qed

```

```

corollary(in prob-kernel) nu-prob-space:
  assumes disintegration  $\nu$   $\mu$  prob-space  $\mu$ 
  shows prob-space  $\nu$ 
proof
  have  $\nu$  (space  $\nu$ ) =  $\nu$  (space ( $X \otimes_M Y$ ))
    using assms by(simp add: disintegration-space-eq)
  also have ... =  $\mu$  (space  $\mu$ )
    using assms by(simp add: nu-mu-space-Y disintegration-space-eq space-pair-measure)
  finally show  $\nu$  (space  $\nu$ ) = 1
    by (simp add: assms(2) prob-space.emeasure-space-1)
qed

```



**lemma**(in *subprob-kernel*) *nu-sigma-finite*:  
**assumes** *disintegration*  $\nu$   $\mu$  *sigma-finite-measure*  $\mu$   
**shows** *sigma-finite-measure*  $\nu$   
**proof**  
**obtain**  $A$  **where**  $A$ :countable  $A$   $A \subseteq \text{sets } \mu \cup A = \text{space } \mu \forall a \in A. \text{emeasure } \mu$   
 $a \neq \infty$   
**using** *assms*(2) **by** (*meson sigma-finite-measure.sigma-finite-countable*)  
**have** countable  $\{a \times \text{space } Y \mid a. a \in A\}$   
**using** *countable-image*[*OF*  $A(1)$ ,*of*  $\lambda a. a \times \text{space } Y$ ]  
**by** (*simp add: Setcompr-eq-image*)  
**moreover have**  $\{a \times \text{space } Y \mid a. a \in A\} \subseteq \text{sets } \nu$   
**using**  $A(2)$  *assms*(1) *disintegration-def* **by** *auto*  
**moreover have**  $\bigcup \{a \times \text{space } Y \mid a. a \in A\} = \text{space } \nu$   
**using** *assms*  $A(3)$  **by**(*simp add: disintegration-space-eq space-pair-measure*)  
*blast*  
**moreover have**  $\forall b \in \{a \times \text{space } Y \mid a. a \in A\}. \text{emeasure } \nu b \neq \infty$   
**using** *neq-top-trans*[*OF* - *nu-mu-space*  $Y$ -*le*[*OF* *assms*(1)]]  $A(2,4)$  *assms* *disintegration-sets-eq*(2) **by** *auto*  
**ultimately show**  $\exists B. \text{countable } B \wedge B \subseteq \text{sets } \nu \wedge \bigcup B = \text{space } \nu \wedge (\forall b \in B. \text{emeasure } \nu b \neq \infty)$   
**by** *blast*  
**qed**

## 2.3 Theorem 14.D.4. (Measure Mixture Theorem)

**lemma**(in *measure-kernel*) *exist-nu*:  
**assumes** *sets*  $\mu = \text{sets } X$   
**shows**  $\exists \nu. \text{disintegration } \nu \mu$   
**proof** –  
**define**  $\nu$  **where**  $\nu = \text{extend-measure } (\text{space } X \times \text{space } Y) \{(a, b). a \in \text{sets } X \wedge b \in \text{sets } Y\} (\lambda(a, b). a \times b) (\lambda(a, b). \int^{+x \in a. \text{emeasure } (\kappa x) b \partial \mu})$   
**have**  $1: \text{sets } \nu = \text{sets } (X \otimes_M Y)$   
**proof** –  
**have** *sets*  $\nu = \text{sigma-sets } (\text{space } X \times \text{space } Y) ((\lambda(a, b). a \times b) \text{ ‘ } \{(a, b). a \in \text{sets } X \wedge b \in \text{sets } Y\})$   
**unfolding**  $\nu$ -*def*  
**by**(*rule sets-extend-measure*) (*use sets.space-closed*[*of*  $X$ ] *sets.space-closed*[*of*  $Y$ ] *in blast*)  
**also have**  $\dots = \text{sigma-sets } (\text{space } X \times \text{space } Y) \{a \times b \mid a b. a \in \text{sets } X \wedge b \in \text{sets } Y\}$   
**by**(*auto intro!: sigma-sets-eqI*)  
**also have**  $\dots = \text{sets } (X \otimes_M Y)$   
**by**(*simp add: sets-pair-measure*)  
**finally show** *?thesis* .  
**qed**  
**have**  $2: \nu (A \times B) = (\int^{+x \in A. (\kappa x B) \partial \mu})$  **if**  $A \in \text{sets } X$   $B \in \text{sets } Y$  **for**  $A B$   
**proof**(*rule extend-measure-caratheodory-pair*[*OF*  $\nu$ -*def*])  
**fix**  $i j k l$   
**assume**  $i \in \text{sets } X \wedge j \in \text{sets } Y$   $k \in \text{sets } X \wedge l \in \text{sets } Y$   $i \times j = k \times l$

```

then consider  $i = k \ j = l \mid i \times j = \{\} \ k \times l = \{\}$  by blast
thus  $(\int^{+x \in i}. \text{emeasure } (\kappa \ x) \ j \ \partial\mu) = (\int^{+x \in k}. \text{emeasure } (\kappa \ x) \ l \ \partial\mu)$ 
by cases auto
next
fix  $A \ B \ j \ k$ 
assume  $h: \bigwedge n::\text{nat}. A \ n \in \text{sets } X \wedge B \ n \in \text{sets } Y \ j \in \text{sets } X \wedge k \in \text{sets } Y$ 
disjoint-family  $(\lambda n. A \ n \times B \ n) (\bigcup i. A \ i \times B \ i) = j \times k$ 
show  $(\sum n. \int^{+x \in A \ n}. \text{emeasure } (\kappa \ x) \ (B \ n) \ \partial\mu) = (\int^{+x \in j}. \text{emeasure } (\kappa \ x) \ k \ \partial\mu)$ 
(is ?lhs = ?rhs)
proof -
have  $?lhs = (\int^{+x}. (\sum n. \kappa \ x \ (B \ n) * \text{indicator } (A \ n) \ x) \ \partial\mu)$ 
proof(rule nn-integral-suminf[symmetric])
fix  $n$ 
have [measurable]:  $(\lambda x. \text{emeasure } (\kappa \ x) \ (B \ n)) \in \text{borel-measurable } \mu$  indicator
 $(A \ n) \in \text{borel-measurable } \mu$ 
using  $h(1)[\text{of } n]$  emeasure-measurable[of B n] assms(1) by auto
thus  $(\lambda x. \text{emeasure } (\kappa \ x) \ (B \ n) * \text{indicator } (A \ n) \ x) \in \text{borel-measurable } \mu$ 
by simp
qed
also have  $\dots = ?rhs$ 
proof(safe intro!: nn-integral-cong)
fix  $x$ 
assume  $x \in \text{space } \mu$ 
consider  $j = \{\} \mid k = \{\} \mid j \neq \{\} \ k \neq \{\}$  by auto
then show  $(\sum n. \text{emeasure } (\kappa \ x) \ (B \ n) * \text{indicator } (A \ n) \ x) = \text{emeasure}$ 
 $(\kappa \ x) \ k * \text{indicator } j \ x$ 
proof cases
case 1
then have  $\bigwedge n. A \ n \times B \ n = \{\}$ 
using  $h(4)$  by auto
have  $\text{emeasure } (\kappa \ x) \ (B \ n) * \text{indicator } (A \ n) \ x = 0$  for  $n$ 
using  $\langle A \ n \times B \ n = \{\} \rangle$  by(auto simp: Sigma-empty-iff)
thus ?thesis
by(simp only: 1, simp)
next
case 2
then have  $\bigwedge n. A \ n \times B \ n = \{\}$ 
using  $h(4)$  by auto
have  $\text{emeasure } (\kappa \ x) \ (B \ n) * \text{indicator } (A \ n) \ x = 0$  for  $n$ 
using  $\langle A \ n \times B \ n = \{\} \rangle$  by(auto simp: Sigma-empty-iff)
thus ?thesis
by(simp only: 2, simp)
next
case 3
then have  $x \text{injiff}: x \in j \longleftrightarrow (\exists i. \exists y \in B \ i. (x, y) \in A \ i \times B \ i)$ 
using  $h(4)$  by blast
have  $\text{bunk}: \bigcup (B \ \{i. x \in A \ i\}) = k$  if  $x \in j$ 
using that 3 h(4) by blast

```

```

show ?thesis
proof(cases x ∈ j)
  case False
  then have  $\bigwedge n. x \notin A\ n \vee B\ n = \{\}$ 
    using h(4)  $\exists$  xinjiff by auto
  have emeasure ( $\kappa$  x) (B n) * indicator (A n) x = 0 for n
    using  $\langle x \notin A\ n \vee B\ n = \{\} \rangle$  by auto
  thus ?thesis
    by(simp only:)(simp add: False)
next
  case True
  then have [simp]: emeasure ( $\kappa$  x) k * indicator j x = emeasure ( $\kappa$  x) k
    by simp
  have ( $\sum n. \text{emeasure } (\kappa\ x)\ (B\ n) * \text{indicator } (A\ n)\ x = (\sum n. \text{emeasure } (\kappa\ x)\ (\text{if } x \in A\ n \text{ then } B\ n \text{ else } \{\}))$ )
    by(auto intro!: suminf-cong)
  also have ... = emeasure ( $\kappa$  x) ( $\bigcup n. \text{if } x \in A\ n \text{ then } B\ n \text{ else } \{\}$ )
    proof(rule suminf-emeasure)
      show disjoint-family ( $\lambda i. \text{if } x \in A\ i \text{ then } B\ i \text{ else } \{\}$ )
      using disjoint-family-onD[OF h(3)] by (auto simp: disjoint-family-on-def)
    next
      show range ( $\lambda i. \text{if } x \in A\ i \text{ then } B\ i \text{ else } \{\}$ )  $\subseteq$  sets ( $\kappa$  x)
      using h(1) kernel-sets[of x]  $\langle x \in \text{space } \mu \rangle$  sets-eq-imp-space-eq[OF
assms(1)] by auto
    qed
  also have ... = emeasure ( $\kappa$  x) k
    using True by(simp add: bunk)
  finally show ?thesis by simp
  qed
qed
qed
qed
finally show ?thesis .
qed
qed(use that in auto)
show ?thesis
  using 1 2 assms
  by(auto simp: disintegration-def)
qed

lemma(in subprob-kernel) exist-unique-nu-sigma-finite':
  assumes sets  $\mu = \text{sets } X \text{ sigma-finite-measure } \mu$ 
  shows  $\exists! \nu. \text{disintegration } \nu \ \mu$ 
proof –
  obtain  $\nu$  where disi: disintegration  $\nu \ \mu$ 
    using exist-nu[OF assms(1)] by auto
  with assms(2) interpret sf: sigma-finite-measure  $\nu$ 
    by(simp add: nu-sigma-finite)
  interpret  $\mu$ : sigma-finite-measure  $\mu$  by fact
  show ?thesis

```

```

proof(rule ex1I[where a= $\nu$ ])
  fix  $\nu'$ 
  assume disi':disintegration  $\nu'$   $\mu$ 
  show  $\nu' = \nu$ 
  proof(rule  $\mu$ .sigma-finite-disjoint)
    fix  $A :: \text{nat} \Rightarrow -$ 
    assume  $A$ : range  $A \subseteq \text{sets } \mu \cup (\text{range } A) = \text{space } \mu \wedge i. \text{emeasure } \mu (A i) \neq \infty$  disjoint-family  $A$ 
    define  $B$  where  $B \equiv \lambda i. A i \times \text{space } Y$ 
    show  $\nu' = \nu$ 
    proof(rule measure-egI-generator-eg[where  $E = \{a \times b \mid a b. a \in \text{sets } X \wedge b \in \text{sets } Y\}$  and  $A=B$  and  $\Omega = \text{space } X \times \text{space } Y$ ])
      show  $\bigwedge C. C \in \{a \times b \mid a b. a \in \text{sets } X \wedge b \in \text{sets } Y\} \implies \text{emeasure } \nu' C = \text{emeasure } \nu C$ 
      sets  $\nu' = \text{sigma-sets } (\text{space } X \times \text{space } Y) \{a \times b \mid a b. a \in \text{sets } X \wedge b \in \text{sets } Y\}$ 
      sets  $\nu = \text{sigma-sets } (\text{space } X \times \text{space } Y) \{a \times b \mid a b. a \in \text{sets } X \wedge b \in \text{sets } Y\}$ 
    using disi disi' by(auto simp: disintegration-def sets-pair-measure)
  next
  show range  $B \subseteq \{a \times b \mid a b. a \in \text{sets } X \wedge b \in \text{sets } Y\} \cup (\text{range } B) = \text{space } X \times \text{space } Y$ 
  using  $A(1,2)$  by(auto simp: B-def assms(1) sets-eg-imp-space-eg[OF assms(1)])
  next
  fix  $i$ 
  show  $\text{emeasure } \nu' (B i) \neq \infty$ 
  using  $A(1)$  nu-mu-spaceY-le[OF disi',of  $A i$ ]  $A(3)$ [of  $i$ ] by(auto simp: B-def assms top.extremum-uniqueI)
  qed(simp-all add: Int-stable-pair-measure-generator pair-measure-closed)
qed
qed fact
qed

```

```

lemma(in subprob-kernel) exist-unique-nu-sigma-finite:
  assumes sets  $\mu = \text{sets } X$  sigma-finite-measure  $\mu$ 
  shows  $\exists! \nu. \text{disintegration } \nu \mu \wedge \text{sigma-finite-measure } \nu$ 
  using assms exist-unique-nu-sigma-finite' nu-sigma-finite by blast

```

```

lemma(in subprob-kernel) exist-unique-nu-finite:
  assumes sets  $\mu = \text{sets } X$  finite-measure  $\mu$ 
  shows  $\exists! \nu. \text{disintegration } \nu \mu \wedge \text{finite-measure } \nu$ 
  using assms nu-finite finite-measure.sigma-finite-measure[OF assms(2)] exist-unique-nu-sigma-finite' by blast

```

```

lemma(in subprob-kernel) exist-unique-nu-sub-prob-space:
  assumes sets  $\mu = \text{sets } X$  subprob-space  $\mu$ 
  shows  $\exists! \nu. \text{disintegration } \nu \mu \wedge \text{subprob-space } \nu$ 
  using assms nu-subprob-space subprob-space-imp-sigma-finite[OF assms(2)] ex-

```

*ist-unique-nu-sigma-finite'* **by** *blast*

**lemma**(*in prob-kernel*) *exist-unique-nu-prob-space*:

**assumes** *sets*  $\mu = \text{sets } X \text{ prob-space } \mu$

**shows**  $\exists! \nu. \text{disintegration } \nu \mu \wedge \text{prob-space } \nu$

**using** *assms nu-prob-space prob-space-imp-sigma-finite*[*OF assms(2)*] *exist-unique-nu-sigma-finite'*  
**by** *blast*

**lemma**(*in subprob-kernel*) *nn-integral-fst-finite'*:

**assumes**  $f \in \text{borel-measurable } (X \otimes_M Y) \text{ disintegration } \nu \mu \text{ finite-measure } \mu$

**shows**  $(\int^+ z. f z \partial \nu) = (\int^+ x. \int^+ y. f (x,y) \partial(\kappa x) \partial \mu)$

**using** *assms(1)*

**proof** *induction*

**case** (*cong f g*)

**have**  $\text{integral}^N \nu f = \text{integral}^N \nu g$

**using** *cong(3)* **by**(*auto intro!*: *nn-integral-cong simp: disintegration-space-eq(1)*)[*OF assms(2)*])

**with** *cong(3)* **show** *?case*

**by**(*auto simp: cong(4) kernel-space disintegration-space-eq(2)*)[*OF assms(2)*]  
*space-pair-measure intro!*: *nn-integral-cong*)

**next**

**case** (*set A*)

**show** *?case*

**proof**(*rule sigma-sets-induct-disjoint*[of  $\{a \times b \mid a \in \text{sets } X \wedge b \in \text{sets } Y\}$ ]  
*space X × space Y*])

**show**  $A \in \text{sigma-sets } (\text{space } X \times \text{space } Y) \{a \times b \mid a \in \text{sets } X \wedge b \in \text{sets } Y\}$

**using** *set* **by**(*simp add: sets-pair-measure*)

**next**

**fix** *A*

**assume**  $A \in \{a \times b \mid a \in \text{sets } X \wedge b \in \text{sets } Y\}$

**then obtain** *a b* **where**  $ab: A = a \times b \ a \in \text{sets } X \ b \in \text{sets } Y$

**by** *auto*

**with** *assms(2)* **have**  $\text{integral}^N \nu (\text{indicator } A) = (\int^+ x \in a. (\kappa x b) \partial \mu)$

**by**(*simp add: disintegration-def*)

**also have**  $\dots = (\int^+ x. \int^+ y. \text{indicator } A (x, y) \partial \kappa x \partial \mu)$

**by**(*auto simp: ab(1) indicator-times disintegration-space-eq(2)*)[*OF assms(2)*]  
*ab(3) kernel-sets mult.commute nn-integral-cmult-indicator intro!*: *nn-integral-cong*)

**finally show**  $\text{integral}^N \nu (\text{indicator } A) = (\int^+ x. \int^+ y. \text{indicator } A (x, y) \partial \kappa x \partial \mu)$  .

**next**

**fix** *A*

**assume**  $h: A \in \text{sigma-sets } (\text{space } X \times \text{space } Y) \{a \times b \mid a \in \text{sets } X \wedge b \in \text{sets } Y\}$   $\text{integral}^N \nu (\text{indicator } A) = (\int^+ x. \int^+ y. \text{indicator } A (x, y) \partial \kappa x \partial \mu)$

**show**  $\text{integral}^N \nu (\text{indicator } (\text{space } X \times \text{space } Y - A)) = (\int^+ x. \int^+ y. \text{indicator } (\text{space } X \times \text{space } Y - A) (x, y) \partial \kappa x \partial \mu)$  (**is** *?lhs = ?rhs*)

**proof** -

**have** *?lhs* =  $(\int^+ z. 1 - \text{indicator } A z \partial \nu)$

**by**(*auto intro!*: *nn-integral-cong simp: disintegration-space-eq(1)*)[*OF assms(2)*]  
*space-pair-measure indicator-def*  
**also have** ... =  $(\int^+ z. 1 \partial\nu) - (\int^+ z. \text{indicator } A \ z \ \partial\nu)$   
**proof**(*rule nn-integral-diff*)  
**show**  $\text{integral}^N \nu (\text{indicator } A) \neq \infty$   
**using**  $h(1)$ [*simplified sets-pair-measure[symmetric]*] *disintegration-sets-eq(1)* [*OF*  
*assms(2)*] *finite-measure.emeasure-finite* [*OF nu-finite* [*OF assms(2,3)*]]  
**by auto**  
**next**  
**show** *indicator*  $A \in \text{borel-measurable } \nu$   
**using**  $h(1)$ [*simplified sets-pair-measure[symmetric]*] *disintegration-sets-eq(1)* [*OF*  
*assms(2)*] **by simp**  
**qed**(*simp-all add: indicator-def*)  
**also have** ... =  $(\int^+ z. 1 \partial\nu) - (\int^+ x. \int^+ y. \text{indicator } A \ (x, y) \ \partial\kappa \ x \ \partial\mu)$   
**by**(*simp add: h(2)*)  
**also have** ... =  $\nu (\text{space } X \times \text{space } Y) - (\int^+ x. \int^+ y. \text{indicator } A \ (x, y)$   
 $\partial\kappa \ x \ \partial\mu)$   
**using** *nn-integral-indicator* [*OF sets.top[of \nu]*] **by**(*simp add: space-pair-measure*  
*disintegration-space-eq(1)* [*OF assms(2)*])  
**also have** ... =  $(\int^+ x. \kappa \ x (\text{space } Y) \ \partial\mu) - (\int^+ x. \int^+ y. \text{indicator } A \ (x, y)$   
 $\partial\kappa \ x \ \partial\mu)$   
**proof** –  
**have**  $\nu (\text{space } X \times \text{space } Y) = (\int^+ x. \kappa \ x (\text{space } Y) \ \partial\mu)$   
**using** *assms(2)* **by**(*auto simp: disintegration-def disintegration-space-eq(2)*)[*OF*  
*assms(2)*] *intro!*: *nn-integral-cong*)  
**thus** *?thesis* **by simp**  
**qed**  
**also have** ... =  $(\int^+ x. \int^+ y. \text{indicator } (\text{space } X \times \text{space } Y) \ (x, y) \ \partial\kappa \ x \ \partial\mu)$   
 $- (\int^+ x. \int^+ y. \text{indicator } A \ (x, y) \ \partial\kappa \ x \ \partial\mu)$   
**proof** –  
**have**  $(\int^+ x. \kappa \ x (\text{space } Y) \ \partial\mu) = (\int^+ x. \int^+ y. \text{indicator } (\text{space } X \times \text{space } Y)$   
 $(x, y) \ \partial\kappa \ x \ \partial\mu)$   
**using** *kernel-sets* **by**(*auto intro!*: *nn-integral-cong simp: indicator-times*  
*disintegration-space-eq(2)* [*OF assms(2)*])  
**thus** *?thesis* **by simp**  
**qed**  
**also have** ... =  $(\int^+ x. (\int^+ y. \text{indicator } (\text{space } X \times \text{space } Y) \ (x, y) \ \partial\kappa \ x)$   
 $- (\int^+ y. \text{indicator } A \ (x, y) \ \partial\kappa \ x \ \partial\mu)$   
**proof**(*rule nn-integral-diff[symmetric]*)  
**show**  $(\lambda x. \int^+ y. \text{indicator } (\text{space } X \times \text{space } Y) \ (x, y) \ \partial\kappa \ x) \in \text{borel-measurable}$   
 $\mu$   
 $(\lambda x. \int^+ y. \text{indicator } A \ (x, y) \ \partial\kappa \ x) \in \text{borel-measurable } \mu$   
**by**(*use disintegration-sets-eq* [*OF assms(2)*] *nn-integral-measurable-f*  
 $h(1)$ [*simplified sets-pair-measure[symmetric]*] **in auto**)+  
**next**  
**have**  $(\int^+ x. \int^+ y. \text{indicator } A \ (x, y) \ \partial\kappa \ x \ \partial\mu) \leq (\int^+ x. \int^+ y. 1 \ \partial\kappa \ x$   
 $\partial\mu)$   
**by**(*rule nn-integral-mono*) + (*simp add: indicator-def*)  
**also have** ...  $\leq (\int^+ x. 1 \ \partial\mu)$

```

by(rule nn-integral-mono) (simp add: subprob-spaces disintegration-space-eq(2))[OF
assms(2)] subprob-space.subprob-emeasure-le-1)
  also have ... < ∞
    using finite-measure.emeasure-finite[OF assms(3)]
    by (simp add: top.not-eq-extremum)
  finally show (∫+ x. ∫+ y. indicator A (x, y) ∂κ x ∂μ) ≠ ∞
    by auto
next
  have A ⊆ space X × space Y
    by (metis h(1) sets.sets-into-space sets-pair-measure space-pair-measure)
  hence ∧x. (∫+ y. indicator A (x, y) ∂κ x) ≤ (∫+ y. indicator (space X
× space Y) (x, y) ∂κ x)
    by(auto intro!: nn-integral-mono)
  thus AE x in μ. (∫+ y. indicator A (x, y) ∂κ x) ≤ (∫+ y. indicator (space
X × space Y) (x, y) ∂κ x)
    by simp
qed
  also have ... = (∫+ x. (∫+ y. indicator (space X × space Y) (x, y) -
indicator A (x, y) ∂κ x) ∂μ)
    proof(intro nn-integral-cong nn-integral-diff[symmetric])
      fix x
      assume x ∈ space μ
      then have x ∈ space X
        by(auto simp: disintegration-space-eq(2))[OF assms(2)])
      with kernel-sets[OF this] h(1)[simplified sets-pair-measure[symmetric]]
      show (λy. indicator (space X × space Y) (x, y)) ∈ borel-measurable (κ x)
        (λy. indicator A (x, y)) ∈ borel-measurable (κ x)
        by auto
    next
      fix x
      assume x ∈ space μ
      then have x ∈ space X
        by(auto simp: disintegration-space-eq(2))[OF assms(2)])
      have (∫+ y. indicator A (x, y) ∂κ x) ≤ (∫+ y. 1 ∂κ x)
        by(rule nn-integral-mono) (simp add: indicator-def)
      also have ... ≤ 1
      using subprob-spaces[OF ⟨x ∈ space X⟩] by (simp add: subprob-space.subprob-emeasure-le-1)
      also have ... < ∞
        by auto
      finally show (∫+ y. indicator A (x, y) ∂κ x) ≠ ∞
        by simp
      have A ⊆ space X × space Y
        by (metis h(1) sets.sets-into-space sets-pair-measure space-pair-measure)
      thus AE y in κ x. indicator A (x, y) ≤ (indicator (space X × space Y) (x,
y) :: ennreal)
        by auto
    qed
  also have ... = ?rhs
    by(auto simp: indicator-def intro!: nn-integral-cong)

```

```

    finally show ?thesis .
  qed
next
fix A
  assume h: disjoint-family A range A  $\subseteq$  sigma-sets (space X  $\times$  space Y) {a  $\times$ 
b | a b. a  $\in$  sets X  $\wedge$  b  $\in$  sets Y}
     $\bigwedge i::nat. \int^N \nu (\text{indicator } (A i)) = (\int^+ x. \int^+ y. \text{indicator } (A i)$ 
(x, y)  $\partial\kappa x \partial\mu$ )
    show  $\int^N \nu (\text{indicator } (\bigcup (\text{range } A))) = (\int^+ x. \int^+ y. \text{indicator } (\bigcup$ 
(range A)) (x, y)  $\partial\kappa x \partial\mu$ ) (is ?lhs = ?rhs)
  proof -
    have ?lhs =  $(\int^+ z. (\sum i. \text{indicator } (A i) z) \partial\nu)$ 
      by (simp add: suminf-indicator[OF h(1)])
    also have ... =  $(\sum i. (\int^+ z. \text{indicator } (A i) z \partial\nu))$ 
      by (rule nn-integral-suminf) (use disintegration-sets-eq(1)[OF assms(2)]
h(2)[simplified sets-pair-measure[symmetric]] in simp)
    also have ... =  $(\sum i. (\int^+ x. \int^+ y. \text{indicator } (A i) (x, y) \partial\kappa x \partial\mu))$ 
      by (simp add: h)
    also have ... =  $(\int^+ x. (\sum i. (\int^+ y. \text{indicator } (A i) (x, y) \partial\kappa x)) \partial\mu)$ 
      by (rule nn-integral-suminf[symmetric]) (use h(2)[simplified sets-pair-measure[symmetric]]
disintegration-sets-eq(2)[OF assms(2)] nn-integral-measurable-f in simp)
    also have ... =  $(\int^+ x. \int^+ y. (\sum i. \text{indicator } (A i) (x, y)) \partial\kappa x \partial\mu)$ 
      using h(2)[simplified sets-pair-measure[symmetric]] kernel-sets
      by (auto intro!: nn-integral-cong nn-integral-suminf[symmetric] simp: disin-
tegration-space-eq(2)[OF assms(2)])
    also have ... = ?rhs
      by (simp add: suminf-indicator[OF h(1)])
    finally show ?thesis .
  qed
qed (simp-all add: Int-stable-pair-measure-generator pair-measure-closed)
next
case (mult u c)
show ?case (is ?lhs = ?rhs)
proof -
  have ?lhs =  $c * (\int^+ z. u z \partial\nu)$ 
    using disintegration-sets-eq(1)[OF assms(2)] mult
    by (simp add: nn-integral-cmult)
  also have ... =  $c * (\int^+ x. \int^+ y. u (x, y) \partial\kappa x \partial\mu)$ 
    by (simp add: mult)
  also have ... =  $(\int^+ x. c * (\int^+ y. u (x, y) \partial\kappa x) \partial\mu)$ 
    using nn-integral-measurable-f'[OF mult(2)] disintegration-sets-eq(2)[OF
assms(2)]
    by (simp add: nn-integral-cmult)
  also have ... = ?rhs
    using mult by (auto intro!: nn-integral-cong nn-integral-cmult[symmetric] simp:
disintegration-space-eq(2)[OF assms(2)])
  finally show ?thesis .
qed
next

```



```

case (add u v)
show ?case (is ?lhs = ?rhs)
proof -
  have ?lhs = ( $\int^+ z. v z \partial\nu$ ) + ( $\int^+ z. u z \partial\nu$ )
  using add disintegration-sets-eq(1)[OF assms(2)] by (simp add: nn-integral-add)
  also have ... = ( $\int^+ x. \int^+ y. v (x, y) \partial\kappa x \partial\mu$ ) + ( $\int^+ x. \int^+ y. u (x, y) \partial\kappa x \partial\mu$ )
  by (simp add: add)
  also have ... = ( $\int^+ x. (\int^+ y. v (x, y) \partial\kappa x) + (\int^+ y. u (x, y) \partial\kappa x) \partial\mu$ )
  using nn-integral-measurable-f'[OF add(1)] nn-integral-measurable-f'[OF add(3)] disintegration-sets-eq[OF assms(2)]
  by (auto intro!: nn-integral-add[symmetric])
  also have ... = ( $\int^+ x. (\int^+ y. v (x, y) + u (x, y) \partial\kappa x) \partial\mu$ )
  using add by (auto intro!: nn-integral-add[symmetric] nn-integral-cong simp: disintegration-space-eq(2)[OF assms(2)])
  finally show ?thesis .
qed
next
case (seq fi)
have ( $\int^+ y. (\bigsqcup \text{range } fi) (x, y) \partial\kappa x$ ) = ( $\bigsqcup i. \int^+ y. fi i (x, y) \partial\kappa x$ ) (is ?lhs = ?rhs) if  $x \in \text{space } X$  for  $x$ 
proof -
  have ?lhs = ( $\int^+ y. (\bigsqcup i. fi i (x, y)) \partial\kappa x$ )
  by (metis SUP-apply)
  also have ... = ?rhs
  proof (rule nn-integral-monotone-convergence-SUP)
    show incseq ( $\lambda i y. fi i (x, y)$ )
    using seq mono-compose by blast
  next
  fix i
  show ( $\lambda y. fi i (x, y)$ )  $\in$  borel-measurable ( $\kappa x$ )
  using seq(1)[of i] that kernel-sets[OF that] by simp
  qed
  finally show ?thesis .
qed
have integralN  $\nu$  ( $\bigsqcup \text{range } fi$ ) = ( $\int^+ x. (\bigsqcup i. fi i x) \partial\nu$ )
by (metis SUP-apply)
also have ... = ( $\bigsqcup i. \text{integral}^N \nu (fi i)$ )
using disintegration-sets-eq(1)[OF assms(2)] seq(1,3)
by (auto intro!: nn-integral-monotone-convergence-SUP)
also have ... = ( $\bigsqcup i. \int^+ x. \int^+ y. fi i (x, y) \partial\kappa x \partial\mu$ )
by (simp add: seq)
also have ... = ( $\int^+ x. (\bigsqcup i. \int^+ y. fi i (x, y) \partial\kappa x) \partial\mu$ )
proof (safe intro!: nn-integral-monotone-convergence-SUP[symmetric])
  show incseq ( $\lambda i x. \int^+ y. fi i (x, y) \partial\kappa x$ )
  using le-funD[OF incseq-SucD[OF seq(3)]]
  by (auto intro!: incseq-SucI le-funI nn-integral-mono)
qed (use disintegration-sets-eq(2)[OF assms(2)] nn-integral-measurable-f'[OF seq(1)] in auto)

```

**also have** ... =  $(\int^+ x. \int^+ y. (\bigsqcup i. fi\ i\ (x, y))\ \partial\kappa\ x\ \partial\mu)$   
**using** *kernel-sets seq(1)*  
**by**(*auto intro!*: *nn-integral-cong nn-integral-monotone-convergence-SUP[symmetric]*  
*simp: disintegration-space-eq(2)[OF assms(2)] mono-compose seq(3)*)  
**also have** ... =  $(\int^+ x. \int^+ y. (\bigsqcup\ range\ fi)\ (x, y)\ \partial\kappa\ x\ \partial\mu)$   
**by**(*auto intro!*: *nn-integral-cong simp: image-image*)  
**finally show** ?*case* .  
**qed**

**lemma**(*in prob-kernel nn-integral-fst:*  
**assumes**  $f \in \text{borel-measurable } (X \otimes_M Y)$  *disintegration*  $\nu\ \mu$  *sigma-finite-measure*  
 $\mu$   
**shows**  $(\int^{+z}. f\ z\ \partial\nu) = (\int^+ x. \int^+ y. f\ (x,y)\ \partial(\kappa\ x)\ \partial\mu)$   
**proof**(*rule sigma-finite-measure.sigma-finite-disjoint[OF assms(3)]*)  
**fix**  $A$   
**assume**  $A:\text{range } A \subseteq \text{sets } \mu \cup (\text{range } A) = \text{space } \mu \wedge i::\text{nat. } \text{emeasure } \mu\ (A\ i)$   
 $\neq \infty$  *disjoint-family*  $A$   
**then have**  $A':\text{range } (\lambda i. A\ i \times \text{space } Y) \subseteq \text{sets } \nu \cup (\text{range } (\lambda i. A\ i \times \text{space } Y)) = \text{space } \nu$  *disjoint-family*  $(\lambda i. A\ i \times \text{space } Y)$   
**by**(*auto simp: disintegration-sets-eq[OF assms(2)] disjoint-family-on-def disintegration-space-eq[OF assms(2)] space-pair-measure*) *blast*  
**show** ?*thesis* (**is** ?*lhs* = ?*rhs*)  
**proof** –  
**have** ?*lhs* =  $(\int^{+z} z \in \bigcup (\text{range } (\lambda i. A\ i \times \text{space } Y)). f\ z\ \partial\nu)$   
**using**  $A'(2)$  **by** *auto*  
**also have** ... =  $(\sum i. \int^{+z} z \in A\ i \times \text{space } Y. f\ z\ \partial\nu)$   
**using**  $A'(1,3)$  *assms(1) disintegration-sets-eq[OF assms(2)]*  
**by**(*auto intro!*: *nn-integral-disjoint-family*)  
**also have** ... =  $(\sum i. \int^+ z. f\ z\ \partial\text{restrict-space } \nu\ (A\ i \times \text{space } Y))$   
**using**  $A'(1)$  **by**(*auto intro!*: *suminf-cong nn-integral-restrict-space[symmetric]*)  
**also have** ... =  $(\sum i. \int^+ x. \int^+ y. f\ (x,y)\ \partial(\kappa\ x)\ \partial\text{restrict-space } \mu\ (A\ i))$   
**proof**(*safe intro!*: *suminf-cong*)  
**fix**  $n$   
**interpret**  $pk:\text{prob-kernel restrict-space } X\ (A\ n)\ Y\ \kappa$   
**by**(*rule restrict-probability-kernel*)  
**have**  $An:A\ n \cap \text{space } X \in \text{sets } X\ A\ n \cap \text{space } X = A\ n$   
**using**  $A(1)$  **by**(*auto simp: disintegration-sets-eq[OF assms(2)]*)  
**have**  $f:f \in \text{borel-measurable } (\text{restrict-space } X\ (A\ n) \otimes_M Y)$   
**proof** –  
**have**  $1:\text{sets } (\text{restrict-space } X\ (A\ n) \otimes_M Y) = \text{sets } (\text{restrict-space } (X \otimes_M Y)\ (A\ n \times \text{space } Y))$   
**using** *sets-pair-restrict-space[where Y=Y and B=space Y]* **by** *simp*  
**show** ?*thesis*  
**using** *assms(1)* **by**(*simp add: measurable-cong-sets[OF 1 refl] measurable-restrict-space1*)  
**qed**  
**have** *fin*: *finite-measure*  $(\text{restrict-space } \mu\ (A\ n))$   
**by** (*metis*  $A(1)$   $A(3)$  *UNIV-I emeasure-restrict-space finite-measureI image-subset-iff space-restrict-space space-restrict-space2 subset-eq*)

**show**  $(\int^+ z. f z \partial \text{restrict-space } \nu (A \ n \times \text{space } Y)) = (\int^+ x. \int^+ y. f (x,y) \partial(\kappa x) \partial \text{restrict-space } \mu (A \ n))$   
**by**  $(\text{rule } pk.\text{nn-integral-fst-finite}'[OF \ \text{disintegration-restrict-space}[OF \ \text{assms}(2) \ An(1)] \ \text{fin}])$   
**qed**  
**also have**  $\dots = (\sum i. \int^+ x \in A \ i. \int^+ y. f (x,y) \partial(\kappa x) \partial \mu)$   
**using**  $A(1)$  **by**  $(\text{auto intro!} : \text{suminf-cong nn-integral-restrict-space})$   
**also have**  $\dots = (\int^+ x \in \bigcup (\text{range } A). \int^+ y. f (x,y) \partial(\kappa x) \partial \mu)$   
**using**  $A(1,4)$   $\text{nn-integral-measurable-f}'[OF \ \text{assms}(1)]$   $\text{disintegration-sets-eq}[OF \ \text{assms}(2)]$   
**by**  $(\text{auto intro!} : \text{nn-integral-disjoint-family[symmetric]})$   
**also have**  $\dots = ?rhs$   
**using**  $A(2)$  **by**  $\text{simp}$   
**finally show**  $?thesis$  .  
**qed**  
**qed**

**lemma**  $(\text{in } \text{prob-kernel}) \ \text{integrable-eq1} :$   
**fixes**  $f :: - \Rightarrow - :: \{\text{banach, second-countable-topology}\}$   
**assumes**  $[\text{measurable}] : f \in \text{borel-measurable } (X \otimes_M Y)$   
**and**  $\text{disintegration } \nu \ \mu \ \text{sigma-finite-measure } \mu$   
**shows**  $(\int^+ z. \text{ennreal } (\text{norm } (f z)) \partial \nu) < \infty \iff (\int^+ x. \int^+ y. \text{ennreal } (\text{norm } (f (x,y))) \partial(\kappa x) \partial \mu) < \infty$   
**by**  $(\text{simp add} : \text{nn-integral-fst}[OF - \ \text{assms}(2,3)])$

**lemma**  $(\text{in } \text{prob-kernel}) \ \text{integrable-kernel-integrable} :$   
**fixes**  $f :: - \Rightarrow - :: \{\text{banach, second-countable-topology}\}$   
**assumes**  $\text{integrable } \nu \ f \ \text{disintegration } \nu \ \mu \ \text{sigma-finite-measure } \mu$   
**shows**  $AE \ x \ \text{in } \mu. \ \text{integrable } (\kappa x) \ (\lambda y. f (x,y))$   
**proof** –  
**have**  $[\text{measurable}] : f \in \text{borel-measurable } (X \otimes_M Y)$   
**using**  $\text{integrable-iff-bounded assms}(1)$   $\text{disintegration-sets-eq}[OF \ \text{assms}(2)]$  **by**  $\text{simp}$   
**show**  $?thesis$   
**unfolding**  $\text{integrable-iff-bounded}$   
**proof** –  
**have**  $1 : (\int^+ x. \int^+ y. \text{ennreal } (\text{norm } (f (x,y))) \partial \kappa x \partial \mu) < \infty$   
**using**  $\text{assms}(1)$   $\text{integrable-eq1}[OF - \ \text{assms}(2,3), \text{of } f]$  **by**  $(\text{simp add} : \text{integrable-iff-bounded})$   
**have**  $AE \ x \ \text{in } \mu. (\int^+ y. \text{ennreal } (\text{norm } (f (x,y))) \partial \kappa x) \neq \infty$   
**by**  $(\text{rule } \text{nn-integral-PInf-AE})$   $(\text{use } 1 \ \text{disintegration-sets-eq}[OF \ \text{assms}(2)] \ \text{nn-integral-measurable-f in auto})$   
**thus**  $AE \ x \ \text{in } \mu. (\lambda y. f (x, y)) \in \text{borel-measurable } (\kappa x) \wedge (\int^+ y. \text{ennreal } (\text{norm } (f (x,y))) \partial \kappa x) < \infty$   
**using**  $\text{top.not-eq-extremum}$  **by**  $(\text{fastforce simp} : \text{disintegration-space-eq}[OF \ \text{assms}(2)])$   
**qed**  
**qed**

**lemma**(in *prob-kernel*) *integrable-lebesgue-integral-integrable'*:  
**fixes**  $f :: - \Rightarrow - :: \{\text{banach, second-countable-topology}\}$   
**assumes** *integrable*  $\nu$  *f disintegration*  $\nu$   $\mu$  *sigma-finite-measure*  $\mu$   
**shows** *integrable*  $\mu$   $(\lambda x. \int y. f(x,y) \partial(\kappa x))$   
**unfolding** *integrable-iff-bounded*  
**proof**  
**show**  $(\lambda x. \int y. f(x,y) \partial(\kappa x)) \in \text{borel-measurable } \mu$   
**using** *disintegration-sets-eq*[*OF assms(2)*] *assms(1)* *integral-measurable-f'*[*of f*]  
**by**(*auto simp: integrable-iff-bounded*)  
**next**  
**have**  $(\int^+ x. \text{ennreal}(\text{norm}(\int y. f(x,y) \partial(\kappa x))) \partial\mu) \leq (\int^+ x. \int^+ y. \text{ennreal}(\text{norm}(f(x,y))) \partial(\kappa x) \partial\mu)$   
**using** *integral-norm-bound-ennreal integrable-kernel-integrable*[*OF assms*]  
**by**(*auto intro!: nn-integral-mono-AE*)  
**also have**  $\dots < \infty$   
**using** *integrable-eq1*[*OF - assms(2,3), of f*] *assms(1)* *disintegration-sets-eq*[*OF assms(2)*]  
**by**(*simp add: integrable-iff-bounded*)  
**finally show**  $(\int^+ x. \text{ennreal}(\text{norm}(\int y. f(x,y) \partial(\kappa x))) \partial\mu) < \infty$ .  
**qed**

**lemma**(in *prob-kernel*) *integrable-lebesgue-integral-integrable*:  
**fixes**  $f :: - \Rightarrow - \Rightarrow - :: \{\text{banach, second-countable-topology}\}$   
**assumes** *integrable*  $\nu$   $(\lambda(x,y). f x y)$  *disintegration*  $\nu$   $\mu$  *sigma-finite-measure*  $\mu$   
**shows** *integrable*  $\mu$   $(\lambda x. \int y. f x y \partial(\kappa x))$   
**using** *integrable-lebesgue-integral-integrable'*[*OF assms*] **by** *simp*

**lemma**(in *prob-kernel*) *integral-fst*:  
**fixes**  $f :: - \Rightarrow - :: \{\text{banach, second-countable-topology}\}$   
**assumes** *integrable*  $\nu$  *f disintegration*  $\nu$   $\mu$  *sigma-finite-measure*  $\mu$   
**shows**  $(\int z. f z \partial\nu) = (\int x. \int y. f(x,y) \partial(\kappa x) \partial\mu)$   
**using** *assms(1)*  
**proof** *induct*  
**case**  $b:(\text{base } A \text{ } c)$   
**then have**  $0:\text{integrable } \nu$  (*indicat-real A*)  
**by** *blast*  
**then have**  $1[\text{measurable}]: \text{indicat-real } A \in \text{borel-measurable } (X \otimes_M Y)$   
**using** *disintegration-sets-eq*[*OF assms(2)*] **by** *auto*  
**have**  $\text{eq}:(\int z. \text{indicat-real } A z \partial\nu) = (\int x. \int y. \text{indicat-real } A(x,y) \partial\kappa x \partial\mu)$  (**is**  
*?lhs = ?rhs*)  
**proof** –  
**have**  $\text{?lhs} = \text{enn2real}(\int^+ z. \text{ennreal}(\text{indicat-real } A z) \partial\nu)$   
**by**(*rule integral-eq-nn-integral*) (*use b in auto*)  
**also have**  $\dots = \text{enn2real}(\int^+ x. \int^+ y. \text{ennreal}(\text{indicat-real } A(x,y)) \partial\kappa x \partial\mu)$   
**using** *nn-integral-fst*[*OF - assms(2,3)*] *b disintegration-sets-eq*[*OF assms(2)*]  
**by** *auto*  
**also have**  $\dots = \text{enn2real}(\int^+ x. \text{ennreal}(\int y. \text{indicat-real } A(x,y) \partial\kappa x) \partial\mu)$   
**proof** –  
**have**  $(\int^+ x. \int^+ y. \text{ennreal}(\text{indicat-real } A(x,y)) \partial\kappa x \partial\mu) = (\int^+ x. \text{ennreal}$

```

( $\int y. \text{indicat-real } A (x,y) \partial\kappa x \partial\mu$ )
proof(safe intro!: nn-integral-cong nn-integral-eq-integral)
  fix x
  assume x  $\in$  space  $\mu$ 
  then have x  $\in$  space X
    by(simp add: disintegration-space-eq[OF assms(2)])
  hence [simp]:prob-space ( $\kappa$  x) sets ( $\kappa$  x) = sets Y space ( $\kappa$  x) = space Y
    by(auto intro!: prob-spaces sets-eq-imp-space-eq kernel-sets)
  have [simp]:{y. (x, y)  $\in$  A}  $\in$  sets Y
  proof -
    have {y. (x, y)  $\in$  A} = ( $\lambda y. (x,y)$ ) -' A  $\cap$  space Y
      using b(1)[simplified disintegration-sets-eq[OF assms(2)]]
      by auto
    also have ...  $\in$  sets Y
      using b(1)[simplified disintegration-sets-eq[OF assms(2)]]  $\langle$ x  $\in$  space X $\rangle$ 
      by auto
    finally show ?thesis .
  qed
  have [simp]: ( $\lambda y. \text{indicat-real } A (x, y)$ ) = indicat-real {y. (x,y)  $\in$  A}
    by (auto simp: indicator-def)
  show integrable ( $\kappa$  x) ( $\lambda y. \text{indicat-real } A (x, y)$ )
    using prob-space.emeasure-le-1[of  $\kappa$  x {y. (x, y)  $\in$  A}]
    by(auto simp add: integrable-indicator-iff order-le-less-trans)
  qed simp
  thus ?thesis by simp
qed
also have ... = ?rhs
  using disintegration-sets-eq[OF assms(2)] integral-measurable-f'[OF 1]
  by(auto intro!: integral-eq-nn-integral[symmetric])
  finally show ?thesis .
qed
show ?case (is ?lhs = ?rhs)
proof -
  have ?lhs = ( $\int z. \text{indicat-real } A z \partial\nu$ ) *R c
    using 0 by auto
  also have ... = ( $\int x. \int y. \text{indicat-real } A (x,y) \partial\kappa x \partial\mu$ ) *R c
    by(simp only: eq)
  also have ... = ( $\int x. (\int y. \text{indicat-real } A (x,y) \partial\kappa x) *_{R} c \partial\mu$ )
    using integrable-lebesgue-integral-integrable'[OF 0 assms(2,3)]
    by(auto intro!: integral-scaleR-left[symmetric])
  also have ... = ?rhs
    using integrable-kernel-integrable[OF 0 assms(2,3)] integral-measurable-f'[of
indicat-real A] integral-measurable-f'[of  $\lambda z. \text{indicat-real } A z *_{R} c$ ] disintegration-sets-eq[OF
assms(2)]
    by(auto intro!: integral-cong-AE)
  finally show ?thesis .
qed
next
  case fg:(add f g)

```

```

note [measurable] = integrable-lebesgue-integral-integrable'[OF fg(1) assms(2,3)]
integrable-lebesgue-integral-integrable'[OF fg(3) assms(2,3)] integrable-lebesgue-integral-integrable'[OF
Bochner-Integration.integrable-add[OF fg(1,3)] assms(2,3)]
show ?case (is ?lhs = ?rhs)
proof -
  have ?lhs = (∫ x. (∫ y. f (x,y) ∂(κ x)) + (∫ y. g (x,y) ∂(κ x)) ∂μ)
  by(simp add: Bochner-Integration.integral-add[OF fg(1,3)] fg Bochner-Integration.integral-add[OF
integrable-lebesgue-integral-integrable'[OF fg(1) assms(2,3)] integrable-lebesgue-integral-integrable'[OF
fg(3) assms(2,3)]])
  also have ... = ?rhs
  using integrable-kernel-integrable[OF fg(1) assms(2,3)] integrable-kernel-integrable[OF
fg(3) assms(2,3)]
  by(auto intro!: integral-cong-AE)
  finally show ?thesis .
qed
next
case (lim f s)
then have [measurable]: f ∈ borel-measurable ν ∧ i. s i ∈ borel-measurable ν
  by auto
show ?case
proof (rule LIMSEQ-unique)
  show (λi. integralL ν (s i)) → integralL ν f
  proof (rule integral-dominated-convergence)
  show integrable ν (λx. 2 * norm (f x))
  using lim(5) by auto
qed(use lim in auto)
next
have (λi. ∫ x. ∫ y. s i (x, y) ∂(κ x) ∂μ) → ∫ x. ∫ y. f (x, y) ∂(κ x) ∂μ
proof (rule integral-dominated-convergence)
  have AE x in μ. ∀ i. integrable (κ x) (λy. s i (x, y))
  unfolding AE-all-countable using integrable-kernel-integrable[OF lim(1)
assms(2,3)] ..
  with AE-space integrable-kernel-integrable[OF lim(5) assms(2,3)]
  show AE x in μ. (λi. ∫ y. s i (x, y) ∂(κ x)) → ∫ y. f (x, y) ∂(κ x)
  proof eventually-elim
  fix x assume x: x ∈ space μ and
  s: ∀ i. integrable (κ x) (λy. s i (x, y)) and f: integrable (κ x) (λy. f (x, y))
  show (λi. ∫ y. s i (x, y) ∂(κ x)) → ∫ y. f (x, y) ∂(κ x)
  proof (rule integral-dominated-convergence)
  show integrable (κ x) (λy. 2 * norm (f (x, y)))
  using f by auto
  show AE xa in (κ x). (λi. s i (x, xa)) → f (x, xa)
  using x lim(3) kernel-space by (auto simp: space-pair-measure disinte-
gration-space-eq[OF assms(2)])
  show ∧ i. AE xa in (κ x). norm (s i (x, xa)) ≤ 2 * norm (f (x, xa))
  using x lim(4) kernel-space by (auto simp: space-pair-measure disinte-
gration-space-eq[OF assms(2)])
  qed (use x disintegration-sets-eq[OF assms(2)] disintegration-space-eq[OF
assms(2)] in auto)

```

```

qed
next
  show integrable  $\mu$  ( $\lambda x. (\int y. 2 * \text{norm } (f(x, y)) \partial(\kappa x))$ )
    using integrable-lebesgue-integral-integrable'[OF - assms(2,3)], of  $\lambda z. 2 * \text{norm } (f(\text{fst } z, \text{snd } z))$ ] lim(5)
    by auto
next
  fix i show AE x in  $\mu. \text{norm } (\int y. s\ i(x, y) \partial(\kappa x)) \leq (\int y. 2 * \text{norm } (f(x, y)) \partial(\kappa x))$ 
    using AE-space integrable-kernel-integrable[OF lim(1) assms(2,3)], of i]
integrable-kernel-integrable[OF lim(5) assms(2,3)]
    proof eventually-elim
      case sf:(elim x)
        from sf(2) have  $\text{norm } (\int y. s\ i(x, y) \partial(\kappa x)) \leq (\int^+ y. \text{norm } (s\ i(x, y)) \partial(\kappa x))$ 
          by (rule integrable-norm-bound-ennreal)
          also have  $\dots \leq (\int^+ y. 2 * \text{norm } (f(x, y)) \partial(\kappa x))$ 
            using sf lim kernel-space by (auto intro!: nn-integral-mono simp: space-pair-measure disintegration-space-eq[OF assms(2)])
            also have  $\dots = (\int y. 2 * \text{norm } (f(x, y)) \partial(\kappa x))$ 
              using sf by (intro nn-integral-eq-integral) auto
              finally show  $\text{norm } (\int y. s\ i(x, y) \partial(\kappa x)) \leq (\int y. 2 * \text{norm } (f(x, y)) \partial(\kappa x))$ 
                by simp
        qed
      qed(use integrable-lebesgue-integral-integrable'[OF lim(1) assms(2,3)] integrable-lebesgue-integral-integrable'[OF lim(5) assms(2,3)] disintegration-sets-eq[OF assms(2)] in auto)
      then show ( $\lambda i. \text{integral}^L \nu (s\ i)$ )  $\longrightarrow \int x. \int y. f(x, y) \partial(\kappa x) \partial\mu$ 
        using lim by simp
    qed
qed

```

## 2.4 Marginal Measure

**definition** *marginal-measure-on* :: [*'a* *measure*, *'b* *measure*, (*'a*  $\times$  *'b*) *measure*, *'b* *set*]  $\Rightarrow$  *'a* *measure* **where**  
*marginal-measure-on* *X* *Y*  $\nu$  *B* = *measure-of* (*space* *X*) (*sets* *X*) ( $\lambda A. \nu (A \times B)$ )

**abbreviation** *marginal-measure* :: [*'a* *measure*, *'b* *measure*, (*'a*  $\times$  *'b*) *measure*]  $\Rightarrow$  *'a* *measure* **where**  
*marginal-measure* *X* *Y*  $\nu$   $\equiv$  *marginal-measure-on* *X* *Y*  $\nu$  (*space* *Y*)

**lemma** *space-marginal-measure*: *space* (*marginal-measure-on* *X* *Y*  $\nu$  *B*) = *space* *X*  
**and** *sets-marginal-measure*: *sets* (*marginal-measure-on* *X* *Y*  $\nu$  *B*) = *sets* *X*  
**by** (*simp-all add: marginal-measure-on-def*)

**lemma** *emeasure-marginal-measure-on*:  
**assumes** *sets*  $\nu = \text{sets } (X \otimes_M Y)$  *B*  $\in$  *sets* *Y* *A*  $\in$  *sets* *X*

```

shows marginal-measure-on  $X\ Y\ \nu\ B\ A = \nu\ (A \times B)$ 
unfolding marginal-measure-on-def
proof(rule emeasure-measure-of-sigma)
show countably-additive (sets  $X$ ) ( $\lambda A.$  emeasure  $\nu\ (A \times B)$ )
proof(rule countably-additiveI)
  fix  $A :: \text{nat} \Rightarrow -$ 
  assume  $h:\text{range } A \subseteq \text{sets } X$  disjoint-family  $A \cup (\text{range } A) \in \text{sets } X$ 
  have [simp]:  $(\bigcup i. A\ i \times B) = (\bigcup (\text{range } A) \times B)$ 
    by blast
  have  $\text{range } (\lambda i. A\ i \times B) \subseteq \text{sets } \nu$  disjoint-family  $(\lambda i. A\ i \times B)$ 
    using  $h$  assms(1,2) by(auto simp: disjoint-family-on-def)
  from suminf-emeasure[OF this]
  show  $(\sum i. \nu\ (A\ i \times B)) = \nu\ (\bigcup (\text{range } A) \times B)$ 
    by simp
qed
qed(insert assms, auto simp: positive-def sets.sigma-algebra-axioms)

lemma emeasure-marginal-measure:
assumes sets  $\nu = \text{sets } (X \otimes_M Y)$   $A \in \text{sets } X$ 
shows marginal-measure  $X\ Y\ \nu\ A = \nu\ (A \times \text{space } Y)$ 
using emeasure-marginal-measure-on[OF assms(1) - assms(2)] by simp

lemma finite-measure-marginal-measure-on-finite:
assumes finite-measure  $\nu$  sets  $\nu = \text{sets } (X \otimes_M Y)$   $B \in \text{sets } Y$ 
shows finite-measure (marginal-measure-on  $X\ Y\ \nu\ B$ )
by (simp add: assms emeasure-marginal-measure-on finite-measure.emeasure-finite
finite-measureI space-marginal-measure)

lemma finite-measure-marginal-measure-finite:
assumes finite-measure  $\nu$  sets  $\nu = \text{sets } (X \otimes_M Y)$ 
shows finite-measure (marginal-measure  $X\ Y\ \nu$ )
by(rule finite-measure-marginal-measure-on-finite[OF assms sets.top])

lemma marginal-measure-restrict-space:
assumes sets  $\nu = \text{sets } (X \otimes_M Y)$   $B \in \text{sets } Y$ 
shows marginal-measure  $X$  (restrict-space  $Y\ B$ ) (restrict-space  $\nu$  (space  $X \times B$ ))
= marginal-measure-on  $X\ Y\ \nu\ B$ 
proof(rule measure-eqI)
  fix  $A$ 
  assume  $A \in \text{sets } (\text{marginal-measure } X\ (\text{restrict-space } Y\ B)\ (\text{restrict-space } \nu\ (\text{space } X \times B)))$ 
  then have  $A \in \text{sets } X$ 
    by(simp add: sets-marginal-measure)
  have  $1:\text{sets } (\text{restrict-space } \nu\ (\text{space } X \times B)) = \text{sets } (X \otimes_M \text{restrict-space } Y\ B)$ 
    by (metis assms(1) restrict-space-space sets-pair-restrict-space sets-restrict-space-cong)
  show emeasure (marginal-measure  $X$  (restrict-space  $Y\ B$ )) (restrict-space  $\nu$  (space
 $X \times B$ )))  $A = \text{emeasure } (\text{marginal-measure-on } X\ Y\ \nu\ B)\ A$ 
    apply(simp add: emeasure-marginal-measure-on[OF assms(1) assms(2)  $\langle A \in \text{sets } X \rangle$ ]
emeasure-marginal-measure[OF 1  $\langle A \in \text{sets } X \rangle$ ])

```



**apply**(*simp add: space-restrict-space*)  
**by** (*metis Sigma-cong Sigma-mono*  $\langle A \in \text{sets } X \rangle$  *assms(1) assms(2) emeasure-restrict-space inf-le1 pair-measureI sets.Int-space-eq2 sets.sets-into-space sets.top*)  
**qed**(*simp add: sets-marginal-measure*)

**lemma** *restrict-space-marginal-measure-on:*

**assumes** *sets*  $\nu = \text{sets } (X \otimes_M Y) B \in \text{sets } Y A \in \text{sets } X$   
**shows** *restrict-space* (*marginal-measure-on*  $X Y \nu B$ )  $A = \text{marginal-measure-on}$  (*restrict-space*  $X A$ )  $Y$  (*restrict-space*  $\nu (A \times \text{space } Y)$ )  $B$   
**proof**(*rule measure-eqI*)  
**fix**  $A'$   
**assume**  $A' \in \text{sets } (\text{restrict-space } (\text{marginal-measure-on } X Y \nu B) A)$   
**then have**  $h: A' \in \text{sets.restricted-space } X A$   
**by**(*simp add: sets-marginal-measure sets-restrict-space*)  
**show** *emeasure* (*restrict-space* (*marginal-measure-on*  $X Y \nu B$ )  $A$ )  $A' = \text{emeasure}$  (*marginal-measure-on* (*restrict-space*  $X A$ )  $Y$  (*restrict-space*  $\nu (A \times \text{space } Y)$ )  $B$ )  $A'$  (*is ?lhs = ?rhs*)  
**proof** -  
**have**  $1: \text{sets } (\text{restrict-space } \nu (A \times \text{space } Y)) = \text{sets } (\text{restrict-space } X A \otimes_M Y)$   
**by** (*metis assms(1) restrict-space-space sets-pair-restrict-space sets-restrict-space-cong*)  
**have**  $?lhs = \text{emeasure}$  (*marginal-measure-on*  $X Y \nu B$ )  $A'$   
**using**  $h$  **by**(*auto intro!: emeasure-restrict-space simp: space-marginal-measure sets-marginal-measure assms*)  
**also have**  $\dots = \nu (A' \times B)$   
**using** *emeasure-marginal-measure-on*[*OF assms(1,2), of A'*]  $h$  *assms(3)* **by** *auto*  
**also have**  $\dots = \text{restrict-space } \nu (A \times \text{space } Y) (A' \times B)$   
**using**  $h$  *assms sets.sets-into-space*  
**by**(*auto intro!: emeasure-restrict-space[symmetric]*)  
**also have**  $\dots = ?rhs$   
**using** *emeasure-marginal-measure-on*[*OF 1 assms(2), simplified sets-restrict-space, OF h*] ..  
**finally show** *?thesis* .  
**qed**  
**qed**(*simp add: sets-marginal-measure sets-restrict-space*)

**lemma** *restrict-space-marginal-measure:*

**assumes** *sets*  $\nu = \text{sets } (X \otimes_M Y) A \in \text{sets } X$   
**shows** *restrict-space* (*marginal-measure*  $X Y \nu$ )  $A = \text{marginal-measure}$  (*restrict-space*  $X A$ )  $Y$  (*restrict-space*  $\nu (A \times \text{space } Y)$ )  
**using** *restrict-space-marginal-measure-on*[*OF assms(1) - assms(2)*] **by** *simp*

**lemma** *marginal-measure-mono:*

**assumes** *sets*  $\nu = \text{sets } (X \otimes_M Y) A \in \text{sets } Y B \in \text{sets } Y A \subseteq B$   
**shows** *emeasure* (*marginal-measure-on*  $X Y \nu A$ )  $\leq \text{emeasure}$  (*marginal-measure-on*  $X Y \nu B$ )  
**proof**(*rule le-funI*)  
**fix**  $U$

```

show emeasure (marginal-measure-on X Y ν A) U ≤ emeasure (marginal-measure-on X Y ν B) U
proof –
  have  $1: U \times A \subseteq U \times B$  using assms(4) by auto
  show ?thesis
  proof(cases U ∈ sets X)
    case True
    then show ?thesis
      by (simp add: 1 assms emeasure-marginal-measure-on emeasure-mono)
    next
    case False
    then show ?thesis
      by (simp add: emeasure-notin-sets sets-marginal-measure)
  qed
qed
qed

```

**lemma** *marginal-measure-absolutely-continuous:*

```

assumes sets ν = sets (X ⊗M Y) A ∈ sets Y B ∈ sets Y A ⊆ B
shows absolutely-continuous (marginal-measure-on X Y ν B) (marginal-measure-on X Y ν A)
using emeasure-marginal-measure[OF assms(1)] assms(2,3) le-funD[OF marginal-measure-mono[OF assms]]
by(auto intro!: mono-absolutely-continuous simp: sets-marginal-measure)

```

**lemma** *marginal-measure-absolutely-continuous':*

```

assumes sets ν = sets (X ⊗M Y) A ∈ sets Y
shows absolutely-continuous (marginal-measure X Y ν) (marginal-measure-on X Y ν A)
by(rule marginal-measure-absolutely-continuous[OF assms sets.top sets.sets-into-space[OF assms(2)]])

```

## 2.5 Lemma 14.D.6.

**locale** *sigma-finite-measure-on-pair =*

```

fixes X :: 'a measure and Y :: 'b measure and ν :: ('a × 'b) measure
assumes nu-sets[measurable-cong]: sets ν = sets (X ⊗M Y)
and sigma-finite: sigma-finite-measure ν
begin

```

**abbreviation**  $\nu x \equiv$  *marginal-measure X Y ν*

**end**

**locale** *projection-sigma-finite =*

```

fixes X :: 'a measure and Y :: 'b measure and ν :: ('a × 'b) measure
assumes nu-sets[measurable-cong]: sets ν = sets (X ⊗M Y)
and marginal-sigma-finite: sigma-finite-measure (marginal-measure X Y ν)
begin

```

**sublocale**  $\nu x$  : *sigma-finite-measure marginal-measure X Y  $\nu$*   
**by**(*rule marginal-sigma-finite*)

**lemma**  $\nu$ -*sigma-finite*: *sigma-finite-measure  $\nu$*

**proof**(*rule  $\nu x$ .sigma-finite[simplified sets-marginal-measure space-marginal-measure]*)

**fix**  $A :: \text{nat} \Rightarrow -$

**assume**  $A$ :  $\text{range } A \subseteq \text{sets } X \cup (\text{range } A) = \text{space } X \wedge i. \text{marginal-measure } X$   
 $Y \nu (A \ i) \neq \infty$

**define**  $C$  **where**  $C \equiv \text{range } (\lambda n. A \ n \times \text{space } Y)$

**have**  $1$ :  $C \subseteq \text{sets } \nu \text{ countable } C \cup C = \text{space } \nu$

**using** *nu-sets A(1,2)* **by**(*auto simp: C-def sets-eq-imp-space-eq[OF nu-sets]*  
*space-pair-measure*)

**show** *sigma-finite-measure  $\nu$*

**unfolding** *sigma-finite-measure-def*

**proof**(*safe intro!*: *exI[where x=C,simplified C-def]*)

**fix**  $n$

**assume**  $\nu (A \ n \times \text{space } Y) = \infty$

**moreover have**  $\nu (A \ n \times \text{space } Y) \neq \infty$

**using**  $A(3)$ [*of n*] *emeasure-marginal-measure[OF nu-sets,of A n] A(1)* **by**

*auto*

**ultimately show** *False* **by** *auto*

**qed** (*use 1 C-def in auto*)

**qed**

**sublocale** *sigma-finite-measure-on-pair*

**using**  $\nu$ -*sigma-finite* **by**(*auto simp: sigma-finite-measure-on-pair-def nu-sets*)

**definition**  $\kappa' :: 'a \Rightarrow 'b \text{ set} \Rightarrow \text{ennreal}$  **where**

$\kappa' \ x \ B \equiv \text{RN-deriv } \nu x \ (\text{marginal-measure-on } X \ Y \ \nu \ B) \ x$

**lemma** *kernel-measurable[measurable]*:

( $\lambda x. \text{RN-deriv } (\text{marginal-measure } X \ Y \ \nu) \ (\text{marginal-measure-on } X \ Y \ \nu \ B) \ x) \in$   
*borel-measurable  $\nu x$*

**by** *simp*

**corollary**  $\kappa'$ -*measurable[measurable]*:

( $\lambda x. \kappa' \ x \ B) \in \text{borel-measurable } X$

**using** *sets-marginal-measure[of X Y  $\nu$  space Y]* **by**(*auto simp:  $\kappa'$ -def*)

**lemma** *kernel-RN-deriv*:

**assumes**  $A \in \text{sets } X \ B \in \text{sets } Y$

**shows**  $\nu (A \times B) = (\int^{+} x \in A. \kappa' \ x \ B \ \partial \nu x)$

**unfolding**  $\kappa'$ -*def*

**proof** –

**have** *emeasure  $\nu (A \times B) = \text{emeasure } (\text{density } \nu x \ (\text{RN-deriv } \nu x \ (\text{marginal-measure-on}$   
 $X \ Y \ \nu \ B))) \ A$*

**by** (*simp add:  $\nu x$ .density-RN-deriv assms emeasure-marginal-measure-on marginal-measure-absolutely-cont*)

*nu-sets sets-marginal-measure*)  
**then show**  $\text{emeasure } \nu (A \times B) = \text{set-nn-integral } \nu x A (\text{RN-deriv } \nu x (\text{marginal-measure-on } X Y \nu B))$   
**by** (*simp add: assms(1) emeasure-density sets-marginal-measure*)  
**qed**

**lemma** *empty-Y-bot*:  
**assumes**  $\text{space } Y = \{\}$   
**shows**  $\nu = \perp$   
**proof** –  
**have**  $\text{sets } \nu = \{\{\}\}$   
**using** *nu-sets space-empty-iff[of X  $\otimes_M$  Y, simplified space-pair-measure] assms*  
**by** *simp*  
**thus** *?thesis*  
**by**(*simp add: sets-eq-bot*)  
**qed**

**lemma** *empty-Y-nux*:  
**assumes**  $\text{space } Y = \{\}$   
**shows**  $\nu x A = 0$   
**proof**(*cases A  $\in$  sets X*)  
**case** *True*  
**from** *emeasure-marginal-measure[OF nu-sets this]*  
**show** *?thesis*  
**by**(*simp add: assms*)  
**next**  
**case** *False*  
**with** *sets-marginal-measure[of X Y  $\nu$  space Y]*  
**show** *?thesis*  
**by**(*auto intro!: emeasure-notin-sets*)  
**qed**

**lemma** *kernel-empty0-AE*:  
 $\text{AE } x \text{ in } \nu x. \kappa' x \{\} = 0$   
**unfolding**  $\kappa'$ -def **by**(*rule AE-symmetric[OF  $\nu x$ .RN-deriv-unique]*) (*auto intro!*:  
*measure-eqI simp: sets-marginal-measure emeasure-density emeasure-marginal-measure-on[OF nu-sets]*)

**lemma** *kernel-Y1-AE*:  
 $\text{AE } x \text{ in } \nu x. \kappa' x (\text{space } Y) = 1$   
**unfolding**  $\kappa'$ -def **by**(*rule AE-symmetric[OF  $\nu x$ .RN-deriv-unique]*) (*auto intro!*:  
*measure-eqI simp: emeasure-density*)

**lemma** *kernel-suminf-AE*:  
**assumes** *disjoint-family F*  
**and**  $\bigwedge i. F i \in \text{sets } Y$   
**shows**  $\text{AE } x \text{ in } \nu x. (\sum i. \kappa' x (F i)) = \kappa' x (\bigcup (\text{range } F))$   
**unfolding**  $\kappa'$ -def  
**proof**(*rule  $\nu x$ .RN-deriv-unique*)

**show**  $\text{density } \nu x (\lambda x. \sum i. \text{RN-deriv local.}\nu x (\text{marginal-measure-on } X Y \nu (F i)))$   
 $x) = \text{marginal-measure-on } X Y \nu (\bigcup (\text{range } F))$   
**proof** (*rule measure-eqI*)  
**fix**  $A$   
**assume** [*measurable*]:  $A \in \text{sets } (\text{density } \nu x (\lambda x. \sum i. \text{RN-deriv } \nu x (\text{marginal-measure-on } X Y \nu (F i)) x))$   
**then have** [*measurable*]:  $A \in \text{sets } \nu x A \in \text{sets } X$  **by** (*auto simp: sets-marginal-measure*)  
**show**  $(\text{density } \nu x (\lambda x. \sum i. \text{RN-deriv } \nu x (\text{marginal-measure-on } X Y \nu (F i)))$   
 $x)) A = (\text{marginal-measure-on } X Y \nu (\bigcup (\text{range } F))) A$   
**(is ?lhs = ?rhs)**  
**proof** –  
**have**  $?lhs = (\int^{+x \in A. (\sum i. \text{RN-deriv } \nu x (\text{marginal-measure-on } X Y \nu (F i)))$   
 $x) \partial \nu x)$   
**by** (*auto intro!: emeasure-density*)  
**also have**  $\dots = (\int^{+x. (\sum i. \text{RN-deriv } \nu x (\text{marginal-measure-on } X Y \nu (F i)))$   
 $x * \text{indicator } A x) \partial \nu x)$   
**by** *simp*  
**also have**  $\dots = (\sum i. (\int^{+x \in A. \text{RN-deriv } \nu x (\text{marginal-measure-on } X Y \nu (F$   
 $i)) x \partial \nu x))$   
**by** (*rule nn-integral-suminf*) *auto*  
**also have**  $\dots = (\sum i. \nu (A \times F i))$   
**using** *kernel-RN-deriv[of A F -]* *assms* **by** (*auto intro!: suminf-cong simp:*  
 $\kappa'$ -*def*)  
**also have**  $\dots = \nu (\bigcup i. A \times F i)$   
**using** *assms nu-sets* **by** (*fastforce intro!: suminf-emeasure simp: disjoint-family-on-def*)  
**also have**  $\dots = \nu (A \times (\bigcup i. F i))$   
**proof** –  
**have**  $(\bigcup i. A \times F i) = (A \times (\bigcup i. F i))$  **by** *blast*  
**thus**  $?thesis$  **by** *simp*  
**qed**  
**also have**  $\dots = ?rhs$   
**using** *nu-sets assms* **by** (*auto intro!: emeasure-marginal-measure-on[symmetric]*)  
**finally show**  $?thesis$  .  
**qed**  
**qed** (*simp add: sets-marginal-measure*)  
**qed** *auto*

**lemma** *kernel-finite-sum-AE*:  
**assumes** *disjoint-family-on F S finite S*  
**and**  $\bigwedge i. i \in S \implies F i \in \text{sets } Y$   
**shows**  $\text{AE } x \text{ in } \nu x. (\sum i \in S. \kappa' x (F i)) = \kappa' x (\bigcup i \in S. F i)$   
**proof** –  
**consider**  $S = \{\} \mid S \neq \{\}$  **by** *auto*  
**then show**  $?thesis$   
**proof** *cases*  
**case**  $1$   
**then show**  $?thesis$   
**by** (*simp add: kernel-empty0-AE*)  
**next**

```

case  $S:2$ 
define  $F'$  where  $F' \equiv (\lambda n. \text{if } n < \text{card } S \text{ then } F \text{ (from-nat-into } S \text{ } n) \text{ else } \{\})$ 
have  $F'[\text{simp}]: \bigwedge i. F' i \in \text{sets } Y$ 
using  $\text{assms}(3)$ 
by ( $\text{metis } F'\text{-def bot.extremum-strict bot-nat-def card.empty from-nat-into sets.empty-sets}$ )
have  $F'\text{-disj: disjoint-family } F'$ 
unfolding  $\text{disjoint-family-on-def}$ 
proof safe
fix  $m \ n \ x$ 
assume  $h: m \neq n \ x \in F' m \ x \in F' n$ 
consider  $n < \text{card } S \ m < \text{card } S \mid n \geq \text{card } S \mid m \geq \text{card } S$  by  $\text{arith}$ 
then show  $x \in \{\}$ 
proof cases
case 1
then have  $S \neq \{\}$ 
by  $\text{auto}$ 
with 1 have  $\text{from-nat-into } S \ n \in S \ \text{from-nat-into } S \ m \in S$ 
using  $\text{from-nat-into[of } S \ ]$  by  $\text{blast+}$ 
moreover have  $\text{from-nat-into } S \ n \neq \text{from-nat-into } S \ m$ 
by ( $\text{metis } 1(1) \ 1(2) \ \text{assms}(2) \ \text{bij-betw-def } h(1) \ \text{lessThan-iff to-nat-on-finite to-nat-on-from-nat-into}$ )
ultimately show  $?thesis$ 
using  $h \ \text{assms}(1) \ 1$  by ( $\text{auto simp: disjoint-family-on-def } F'\text{-def}$ )
qed( $\text{use } h \ F'\text{-def in simp-all}$ )
qed
have  $1: (\sum i \in S. \kappa' x (F' i)) = (\sum i < \text{card } S. \kappa' x (F' i))$  for  $x$ 
unfolding  $F'\text{-def}$  by  $\text{auto (metis (no-types, lifting) sum.card-from-nat-into sum.cong)}$ 
have  $2: (\bigcup (\text{range } F')) = (\bigcup i \in S. F' i)$ 
proof safe
fix  $x \ n$ 
assume  $h: x \in F' n$ 
then have  $S \neq \{\} \ n < \text{card } S$ 
by ( $\text{auto simp: } F'\text{-def (meson empty-iff)}$ )
with  $h$  show  $x \in \bigcup (F' ` S)$ 
by ( $\text{auto intro!: exI[where } x = \text{from-nat-into } S \ n] \ \text{simp: } F'\text{-def from-nat-into } \langle S \neq \{\} \rangle$ )
next
fix  $x \ s$ 
assume  $s \in S \ x \in F' s$ 
with  $\text{bij-betwE[OF to-nat-on-finite[OF assms}(2)]}$ 
show  $x \in \bigcup (\text{range } F')$ 
by ( $\text{auto intro!: exI[where } x = \text{to-nat-on } S \ s] \ \text{simp: } F'\text{-def from-nat-into-to-nat-on[OF countable-finite[OF assms}(2)]}$ )
qed
have  $AE \ x \ \text{in } \nu x. (\sum i < \text{card } S. \kappa' x (F' i)) = (\sum i. \kappa' x (F' i))$ 
proof -
have  $AE \ x \ \text{in } \nu x. \forall i \geq \text{card } S. \kappa' x (F' i) = 0$ 

```

```

using kernel-empty0-AE by(auto simp: F'-def)
hence AE x in νx. (∑ i. κ' x (F' i)) = (∑ i < card S. κ' x (F' i))
proof
  show AE x in νx. (∀ i ≥ card S. κ' x (F' i) = 0) → (∑ i. κ' x (F' i)) =
(∑ i < card S. κ' x (F' i))
  proof -
    {
      fix x
      assume ∀ i ≥ card S. κ' x (F' i) = 0
      then have (∑ i. κ' x (F' i)) = (∑ i < card S. κ' x (F' i))
        using suminf-offset[of λi. κ' x (F' i) card S]
        by(auto simp: F'-def)
    }
  thus ?thesis
  by auto
qed
qed
thus ?thesis
by auto
qed
moreover have AE x in νx. (∑ i. κ' x (F' i)) = κ' x (∪ (range F'))
  using kernel-suminf-AE[OF F'-disj] by simp
ultimately show ?thesis
by(auto simp: 1 2)
qed
qed

```

**lemma** *kernel-disjoint-sum-AE*:

```

assumes B ∈ sets Y C ∈ sets Y
and B ∩ C = {}
shows AE x in νx. κ' x (B ∪ C) = κ' x B + κ' x C
proof -
  define F where F ≡ λb. if b then B else C
  have [simp]: disjoint-family F ∧ i. F i ∈ sets Y ∧ x. (∑ i ∈ UNIV. κ' x (F i)) =
κ' x B + κ' x C ∪ (range F) = B ∪ C
  using assms by(auto simp: F-def disjoint-family-on-def comm-monoid-add-class.sum.Int-Diff[of
UNIV - {True}])
  show ?thesis
  using kernel-finite-sum-AE[of F UNIV] by auto
qed

```

**lemma** *kernel-mono-AE*:

```

assumes B ∈ sets Y C ∈ sets Y
and B ⊆ C
shows AE x in νx. κ' x B ≤ κ' x C
proof -
  have 1: B ∪ (C - B) = C using assms(3) by auto
  have AE x in νx. κ' x C = κ' x B + κ' x (C - B)
  using assms by(auto intro!: kernel-disjoint-sum-AE[of B C - B,simplified 1])

```

**thus** *?thesis*  
**by** *auto*  
**qed**

**lemma** *kernel-incseq-AE*:  
**assumes**  $\text{range } B \subseteq \text{sets } Y \text{ incseq } B$   
**shows**  $AE \ x \ \text{in } \nu x. \text{ incseq } (\lambda n. \kappa' \ x \ (B \ n))$   
**using** *assms(1)* **by** (*auto simp: incseq-Suc-iff AE-all-countable intro!: kernel-mono-AE[OF*  
*- - incseq-SucD[OF assms(2)]]*)

**lemma** *kernel-decseq-AE*:  
**assumes**  $\text{range } B \subseteq \text{sets } Y \text{ decseq } B$   
**shows**  $AE \ x \ \text{in } \nu x. \text{ decseq } (\lambda n. \kappa' \ x \ (B \ n))$   
**using** *assms(1)* **by** (*auto simp: decseq-Suc-iff AE-all-countable intro!: kernel-mono-AE[OF*  
*- - decseq-SucD[OF assms(2)]]*)

**corollary** *kernel-01-AE*:  
**assumes**  $B \in \text{sets } Y$   
**shows**  $AE \ x \ \text{in } \nu x. \ 0 \leq \kappa' \ x \ B \wedge \kappa' \ x \ B \leq 1$   
**proof** –  
**have**  $\{\} \subseteq B \ B \subseteq \text{space } Y$   
**using** *assms sets.sets-into-space* **by** *auto*  
**from** *kernel-empty0-AE kernel-Y1-AE kernel-mono-AE[OF - - this(1)] kernel-mono-AE[OF*  
*- - this(2)] assms*  
**show** *?thesis*  
**by** *auto*  
**qed**

**lemma** *kernel-get-0*:  $0 \leq \kappa' \ x \ B$   
**by** *simp*

**lemma** *kernel-le1-AE*:  
**assumes**  $B \in \text{sets } Y$   
**shows**  $AE \ x \ \text{in } \nu x. \ \kappa' \ x \ B \leq 1$   
**using** *kernel-01-AE[OF assms]* **by** *auto*

**corollary** *kernel-n-infty*:  
**assumes**  $B \in \text{sets } Y$   
**shows**  $AE \ x \ \text{in } \nu x. \ \kappa' \ x \ B \neq \top$   
**by** (*rule AE-mp[OF kernel-le1-AE[OF assms]],standard*) (*auto simp: neq-top-trans[OF*  
*ennreal-one-neq-top]*)

**corollary** *kernel-le-infty*:  
**assumes**  $B \in \text{sets } Y$   
**shows**  $AE \ x \ \text{in } \nu x. \ \kappa' \ x \ B < \top$   
**using** *kernel-n-infty[OF assms]* **by** (*simp add: top.not-eq-extremum*)

**lemma** *kernel-SUP-incseq*:  
**assumes**  $\text{range } B \subseteq \text{sets } Y \text{ incseq } B$



**shows**  $AE\ x\ in\ \nu x. \kappa' x (\bigcup (\text{range } B)) = (\bigsqcup n. \kappa' x (B\ n))$   
**proof** –  
**define**  $Bn$  **where**  $Bn \equiv (\lambda n. \text{if } n = 0 \text{ then } \{\} \text{ else } B\ (n - 1))$   
**have**  $incseq\ Bn$   
**using**  $assms(2)$  **by**( $auto\ simp: Bn\text{-def}\ incseq\text{-def}$ )  
**define**  $Cn$  **where**  $Cn \equiv (\lambda n. Bn\ (Suc\ n) - Bn\ n)$   
**have**  $Cn\text{-simp}: Cn\ 0 = B\ 0\ Cn\ (Suc\ n) = B\ (Suc\ n) - B\ n$  **for**  $n$   
**by**( $simp\text{-all}\ add: Cn\text{-def}\ Bn\text{-def}$ )  
**have**  $Cn\text{-sets}: Cn\ n \in \text{sets } Y$  **for**  $n$   
**using**  $assms(1)$  **by**( $induction\ n$ ) ( $auto\ simp: Cn\text{-simp}$ )  
**have**  $Cn\text{-disj}: \text{disjoint-family } Cn$   
**by**( $auto\ intro!: \text{disjoint-family-Suc}[OF\ ]\ incseq\text{-SucD}[OF\ \langle incseq\ Bn \rangle]\ simp:$   
 $Cn\text{-def}$ )  
**have**  $Cn\text{-un}: (\bigcup_{k < Suc\ n} Cn\ k) = B\ n$  **for**  $n$   
**using**  $incseq\text{-SucD}[OF\ assms(2)]$   
**by** ( $induction\ n$ ) ( $auto\ simp: Cn\text{-simp}\ lessThan\text{-Suc}\ sup\text{-commute}$ )  
**have**  $Cn\text{-sum-Bn}: AE\ x\ in\ \nu x. \forall n. (\sum_{i < Suc\ n} \kappa' x (Cn\ i)) = \kappa' x (B\ n)$   
**unfolding**  $AE\text{-all-countable}$   
**using**  $kernel\text{-finite-sum-AE}[OF\ \text{disjoint-family-on-mono}[OF\ -\ Cn\text{-disj}],\ of\ \{.. < Suc$   
 $\text{-}\}] Cn\text{-sets}$   
**by**( $auto\ simp: Cn\text{-un}$ )  
**have**  $Cn\text{-bn-un}: (\bigcup (\text{range } B)) = (\bigcup (\text{range } Cn))$  (**is**  $?lhs = ?rhs$ )  
**proof**  $safe$   
**fix**  $n\ x$   
**assume**  $x \in B\ n$   
**with**  $Cn\text{-un}[of\ n]$  **show**  $x \in \bigcup (\text{range } Cn)$   
**by**  $blast$   
**next**  
**fix**  $n\ x$   
**assume**  $x \in Cn\ n$   
**then** **show**  $x \in \bigcup (\text{range } B)$   
**by**( $cases\ n,\ auto\ simp: Cn\text{-simp}$ )  
**qed**  
**hence**  $AE\ x\ in\ \nu x. \kappa' x (\bigcup (\text{range } B)) = \kappa' x (\bigcup (\text{range } Cn))$   
**by**  $simp$   
**moreover** **have**  $AE\ x\ in\ \nu x. \kappa' x (\bigcup (\text{range } Cn)) = (\sum n. \kappa' x (Cn\ n))$   
**by**( $rule\ AE\text{-symmetric}[OF\ kernel\text{-suminf-AE}[OF\ Cn\text{-disj}]]$ ) ( $use\ Cn\text{-def}\ Bn\text{-def}$   
 $assms(1)$  **in**  $auto$ )  
**moreover** **have**  $AE\ x\ in\ \nu x. (\sum n. \kappa' x (Cn\ n)) = (\bigsqcup n. \sum_{i < n} \kappa' x (Cn\ i))$   
**by**( $auto\ simp: suminf\text{-eq-SUP}$ )  
**moreover** **have**  $AE\ x\ in\ \nu x. (\bigsqcup n. \sum_{i < n} \kappa' x (Cn\ i)) = (\bigsqcup n. \sum_{i < Suc\ n} \kappa'$   
 $x\ (Cn\ i))$   
**proof**( $intro\ AE\text{-I2}\ antisym$ )  
**fix**  $x$   
**show**  $(\bigsqcup n. \sum_{i < n} \kappa' x (Cn\ i)) \leq (\bigsqcup n. \sum_{i < Suc\ n} \kappa' x (Cn\ i))$   
**by**( $rule\ complete\text{-lattice-class.SUP}\text{-mono},\ auto,\ use\ le\text{-iff-add}$  **in**  $blast$ )  
**next**  
**fix**  $x$   
**show**  $(\bigsqcup n. \sum_{i < n} \kappa' x (Cn\ i)) \geq (\bigsqcup n. \sum_{i < Suc\ n} \kappa' x (Cn\ i))$

by(rule complete-lattice-class.Sup-mono) blast  
 qed  
 moreover have  $AE\ x\ in\ \nu x.\ (\bigsqcup n.\ \sum i < Suc\ n.\ \kappa'\ x\ (C\ n\ i)) = (\bigsqcup n.\ \kappa'\ x\ (B\ n))$   
 by(rule AE-mp[OF Cn-sum-Bn]) (standard+, auto)  
 ultimately show ?thesis by auto  
 qed

**lemma** kernel-lim-incseq:  
 assumes  $range\ B \subseteq sets\ Y\ incseq\ B$   
 shows  $AE\ x\ in\ \nu x.\ (\lambda n.\ \kappa'\ x\ (B\ n)) \longrightarrow \kappa'\ x\ (\bigcup (range\ B))$   
 by(rule AE-mp[OF AE-conjI[OF kernel-SUP-incseq[OF assms] kernel-incseq-AE[OF assms]]], auto simp: LIMSEQ-SUP)

**lemma** kernel-INF-decseq:  
 assumes  $range\ B \subseteq sets\ Y\ decseq\ B$   
 shows  $AE\ x\ in\ \nu x.\ \kappa'\ x\ (\bigcap (range\ B)) = (\prod n.\ \kappa'\ x\ (B\ n))$   
**proof** –  
 define  $C$  where  $C \equiv (\lambda k.\ space\ Y - B\ k)$   
 have  $C: range\ C \subseteq sets\ Y\ incseq\ C$   
 using assms by(auto simp: C-def decseq-def incseq-def)  
 have eq1:  $AE\ x\ in\ \nu x.\ 1 - \kappa'\ x\ (\bigcap (range\ B)) = \kappa'\ x\ (\bigcup (range\ C))$   
**proof** –  
 have  $AE\ x\ in\ \nu x.\ \kappa'\ x\ (\bigcup (range\ C)) + \kappa'\ x\ (\bigcap (range\ B)) - \kappa'\ x\ (\bigcap (range\ B)) = \kappa'\ x\ (\bigcup (range\ C))$   
 using assms(1) kernel-n-infty[of  $\bigcap (range\ B)$ ] by auto  
 moreover have  $AE\ x\ in\ \nu x.\ \kappa'\ x\ (\bigcup (range\ C)) + \kappa'\ x\ (\bigcap (range\ B)) = 1$   
**proof** –  
 have [simp]:  $(\bigcup (range\ C)) \cup (\bigcap (range\ B)) = space\ Y\ (\bigcup (range\ C)) \cap (\bigcap (range\ B)) = \{\}$   
 by(auto simp: C-def) (meson assms(1) range-subsetD sets.sets-into-space subsetD)  
 from kernel-disjoint-sum-AE[OF - - this(2)] C(1) assms(1) kernel-Y1-AE  
 show ?thesis by auto  
 qed  
 ultimately show ?thesis  
 by auto  
 qed

**lemma** kernel-SUP-incseq:  
 have eq2:  $AE\ x\ in\ \nu x.\ \kappa'\ x\ (\bigcup (range\ C)) = (\bigsqcup n.\ \kappa'\ x\ (C\ n))$   
 using kernel-SUP-incseq[OF C] by auto  
 have eq3:  $AE\ x\ in\ \nu x.\ (\bigsqcup n.\ \kappa'\ x\ (C\ n)) = (\bigsqcup n.\ 1 - \kappa'\ x\ (B\ n))$   
**proof** –  
 have  $AE\ x\ in\ \nu x.\ \forall n.\ \kappa'\ x\ (C\ n) = 1 - \kappa'\ x\ (B\ n)$   
 unfolding AE-all-countable  
**proof** safe  
 fix  $n$   
 have  $AE\ x\ in\ \nu x.\ \kappa'\ x\ (C\ n) + \kappa'\ x\ (B\ n) - \kappa'\ x\ (B\ n) = \kappa'\ x\ (C\ n)$   
 using assms(1) kernel-n-infty[of  $B\ n$ ] by auto  
 moreover have  $AE\ x\ in\ \nu x.\ \kappa'\ x\ (C\ n) + \kappa'\ x\ (B\ n) = 1$   
**proof** –

```

      have [simp]:  $C\ n \cup B\ n = \text{space } Y\ C\ n \cap B\ n = \{\}$ 
      by(auto simp: C-def) (meson assms(1) range-subsetD sets.sets-into-space
subsetD)
      thus ?thesis
      using kernel-disjoint-sum-AE[of C n B n] C(1) assms(1) kernel-Y1-AE
by fastforce
      qed
      ultimately show  $AE\ x\ \text{in } \nu x. \kappa' x\ (C\ n) = 1 - \kappa' x\ (B\ n)$  by auto
      qed
      thus ?thesis by auto
      qed
      have [simp]:  $(\bigsqcup n. 1 - \kappa' x\ (B\ n)) = 1 - (\prod n. \kappa' x\ (B\ n))$  for x
      by(auto simp: ennreal-INF-const-minus)
      have eq:  $AE\ x\ \text{in } \nu x. 1 - \kappa' x\ (\bigcap (\text{range } B)) = 1 - (\prod n. \kappa' x\ (B\ n))$ 
      using eq1 eq2 eq3 by auto
      have le1:  $AE\ x\ \text{in } \nu x. (\prod n. \kappa' x\ (B\ n)) \leq 1$ 
      proof -
      have  $AE\ x\ \text{in } \nu x. \forall n. \kappa' x\ (B\ n) \leq 1$ 
      using assms(1) by(auto intro!: kernel-le1-AE simp: AE-all-countable)
      thus ?thesis
      by (auto simp: INF-lower2)
      qed
      show ?thesis
      by(rule AE-mp[OF AE-conjI[OF AE-conjI[OF eq le1] kernel-le1-AE[of  $\bigcap$ 
(range B)]]]])
      (insert assms(1),auto simp: ennreal-minus-cancel[OF ennreal-one-neq-top])
      qed

```

```

lemma kernel-lim-decseq:
  assumes  $\text{range } B \subseteq \text{sets } Y\ \text{decseq } B$ 
  shows  $AE\ x\ \text{in } \nu x. (\lambda n. \kappa' x\ (B\ n)) \longrightarrow \kappa' x\ (\bigcap (\text{range } B))$ 
  by(rule AE-mp[OF AE-conjI[OF kernel-INF-decseq[OF assms] kernel-decseq-AE[OF
assms]]],standard,auto simp: LIMSEQ-INF)

```

end

```

lemma qlim-eq-lim-mono-at-bot:
  fixes  $g :: \text{rat} \Rightarrow 'a :: \text{linorder-topology}$ 
  assumes  $\text{mono } f\ (g \longrightarrow a)\ \text{at-bot } \bigwedge r :: \text{rat}. f\ (\text{real-of-rat } r) = g\ r$ 
  shows  $(f \longrightarrow a)\ \text{at-bot}$ 
  proof -
  have mono g
  by(metis assms(1,3) mono-def of-rat-less-eq)
  have  $ga: \bigwedge r. g\ r \geq a$ 
  proof(rule ccontr)
  fix r
  assume  $\neg a \leq g\ r$ 
  then have  $g\ r < a$  by simp
  from order-topology-class.order-tendstoD(1)[OF assms(2) this]

```

```

obtain  $Q :: \text{rat}$  where  $q: \bigwedge q. q \leq Q \implies g r < g q$ 
  by(auto simp: eventually-at-bot-linorder)
define  $q$  where  $q \equiv \min r Q$ 
show False
  using  $q[\text{of } q] \langle \text{mono } g \rangle$ 
  by(auto simp: q-def mono-def) (meson linorder-not-less min.cobounded1)
qed
show ?thesis
proof(rule decreasing-tendsto)
  show  $\forall_F n$  in at-bot.  $a \leq f n$ 
    unfolding eventually-at-bot-linorder
    by(rule exI[where x=undefined], auto) (metis Ratreal-def assms(1,3) dual-order.trans
ga less-eq-real-def lt-ex monoD of-rat-dense)
  next
    fix  $x$ 
    assume  $a < x$ 
    with topological-space-class.topological-tendstoD[OF assms(2), of {.. $x$ }]
    obtain  $Q :: \text{rat}$  where  $q: \bigwedge q. q \leq Q \implies g q < x$ 
      by(auto simp: eventually-at-bot-linorder)
    show  $\forall_F n$  in at-bot.  $f n < x$ 
      using  $q$  assms(1,3) by(auto intro!: exI[where x=real-of-rat Q] simp: even-
tually-at-bot-linorder) (metis dual-order.refl monoD order-le-less-trans)
    qed
  qed

lemma qlim-eq-lim-mono-at-top:
  fixes  $g :: \text{rat} \Rightarrow 'a :: \text{linorder-topology}$ 
  assumes mono  $f (g \longrightarrow a)$  at-top  $\bigwedge r :: \text{rat}. f (\text{real-of-rat } r) = g r$ 
  shows  $(f \longrightarrow a)$  at-top
proof –
  have mono  $g$ 
    by(metis assms(1,3) mono-def of-rat-less-eq)
  have  $ga: \bigwedge r. g r \leq a$ 
  proof(rule ccontr)
    fix  $r$ 
    assume  $\neg g r \leq a$ 
    then have  $a < g r$  by simp
    from order-topology-class.order-tendstoD(2)[OF assms(2) this]
    obtain  $Q :: \text{rat}$  where  $q: \bigwedge q. Q \leq q \implies g q < g r$ 
      by(auto simp: eventually-at-top-linorder)
    define  $q$  where  $q \equiv \max r Q$ 
    show False
      using  $q[\text{of } q] \langle \text{mono } g \rangle$  by(auto simp: q-def mono-def leD)
    qed
  show ?thesis
proof(rule increasing-tendsto)
  show  $\forall_F n$  in at-top.  $f n \leq a$ 
    unfolding eventually-at-top-linorder
    by(rule exI[where x=undefined], auto) (metis (no-types, opaque-lifting) assms(1))

```

```

assms( $\beta$ ) dual-order.trans ga gt-ex monoD of-rat-dense order-le-less)
next
  fix  $x$ 
  assume  $x < a$ 
  with topological-space-class.topological-tendstoD[OF assms( $\beta$ ),of { $x < ..$ }]
  obtain  $Q :: \text{rat}$  where  $q: \bigwedge q. Q \leq q \implies x < g\ q$ 
    by(auto simp: eventually-at-top-linorder)
  show  $\forall_F n$  in at-top.  $x < f\ n$ 
    using  $q$  assms( $1, \beta$ ) by(auto simp: eventually-at-top-linorder intro!: exI[where
x=real-of-rat Q]) (metis dual-order.refl monoD order-less-le-trans)
  qed
qed

```

## 2.6 Theorem 14.D.10. (Measure Disintegration Theorem)

```

locale projection-sigma-finite-standard = projection-sigma-finite + standard-borel-ne
   $Y$ 
begin

```

**theorem** *measure-disintegration*:

```

   $\exists \kappa. \text{prob-kernel } X\ Y\ \kappa \wedge \text{measure-kernel.disintegration } X\ Y\ \kappa\ \nu\ \nu x \wedge$ 
  ( $\forall \kappa''. \text{prob-kernel } X\ Y\ \kappa'' \longrightarrow \text{measure-kernel.disintegration } X\ Y\ \kappa''\ \nu\ \nu x$ 
 $\longrightarrow (AE\ x\ \text{in } \nu x. \kappa\ x = \kappa''\ x))$ )

```

**proof** –

```

  have  $*$ :  $\exists \kappa. \text{prob-kernel } X\ (\text{borel} :: \text{real measure})\ \kappa \wedge \text{measure-kernel.disintegration}$ 
 $X\ \text{borel}\ \kappa\ \nu\ (\text{marginal-measure } X\ \text{borel}\ \nu) \wedge$ 
  ( $\forall \kappa''. \text{prob-kernel } X\ \text{borel}\ \kappa'' \longrightarrow \text{measure-kernel.disintegration } X\ \text{borel}$ 
 $\kappa''\ \nu\ (\text{marginal-measure } X\ \text{borel}\ \nu) \longrightarrow (AE\ x\ \text{in } (\text{marginal-measure } X\ \text{borel}\ \nu). \kappa$ 
 $x = \kappa''\ x))$ )

```

**if** *nu-sets'*: *sets*  $\nu = \text{sets } (X \otimes_M \text{borel})$  **and** *marginal-sigma-finite'*:  
*sigma-finite-measure* (*marginal-measure*  $X$  *borel*  $\nu$ ) **for**  $X :: 'a \text{ measure and } \nu$

**proof** –

**interpret**  $r$ : *projection-sigma-finite*  $X$  *borel*  $\nu$

**using** *that* **by**(*auto simp: projection-sigma-finite-def*)

**define**  $\varphi :: 'a \Rightarrow \text{rat} \Rightarrow \text{real}$

**where**  $\varphi \equiv (\lambda x\ r. \text{enn2real } (r.\kappa'\ x \ \{.. \text{real-of-rat } r\}))$

**have** *as1*:  $AE\ x\ \text{in } r.\nu x. \forall r\ s. r \leq s \longrightarrow \varphi\ x\ r \leq \varphi\ x\ s$

**unfolding** *AE-all-countable*

**proof**(*safe intro!*: *AE-impI*)

**fix**  $r\ s :: \text{rat}$

**assume**  $r \leq s$

**have**  $AE\ x\ \text{in } r.\nu x. r.\kappa'\ x \ \{.. \text{real-of-rat } k\} < \text{top}$  **for**  $k$

**using** *atMost-borel r.kernel-le-infity* **by** *blast*

**from** *this*[*of*  $s$ ] *r.kernel-mono-AE*[*of* { $.. \text{real-of-rat } r$ }] { $.. \text{real-of-rat } s$ }] ( $r \leq s$ )

**show**  $AE\ x\ \text{in } r.\nu x. \varphi\ x\ r \leq \varphi\ x\ s$

**by**(*auto simp: phi-def of-rat-less-eq enn2real-mono*)

**qed**

**have** *as2*:  $AE\ x\ \text{in } r.\nu x. \forall r. (\lambda n. \varphi\ x\ (r + 1 / \text{rat-of-nat } (\text{Suc } n))) \longrightarrow \varphi$

```

x r
  unfolding AE-all-countable
proof safe
  fix r
  have 1:( $\bigcap n. \{..real-of-rat (r + 1 / rat-of-nat (Suc n))\}$ ) =  $\{..real-of-rat r\}$ 
proof safe
  fix x
  assume h:  $x \in (\bigcap n. \{..real-of-rat (r + 1 / rat-of-nat (Suc n))\})$ 
  show  $x \leq real-of-rat r$ 
  proof(rule ccontr)
    assume  $\neg x \leq real-of-rat r$ 
    then have  $0 < x - real-of-rat r$  by simp
    then obtain n where  $(1 / (real (Suc n))) < x - real-of-rat r$ 
      using nat-approx-posE by blast
    hence  $real-of-rat (r + 1 / (1 + rat-of-nat n)) < x$ 
      by (simp add: of-rat-add of-rat-divide)
    with h show False
      using linorder-not-le by fastforce
  qed
next
  fix x n
  assume  $x \leq real-of-rat r$ 
  then show  $x \leq real-of-rat (r + 1 / rat-of-nat (Suc n))$ 
    by (metis le-add-same-cancel1 of-nat-0-le-iff of-rat-less-eq order-trans
zero-le-divide-1-iff)
  qed
  have AE x in r.v x. ( $\lambda n. r.\kappa' x \{..real-of-rat (r + 1 / rat-of-nat (Suc n))\}$ )
   $\longrightarrow r.\kappa' x \{..real-of-rat r\}$ 
    unfolding 1[symmetric] by(rule r.kernel-lim-decseq) (auto simp: dec-
seq-Suc-iff of-rat-less-eq frac-le)
  from AE-conjI[OF r.kernel-le-infty[of  $\{..real-of-rat r\}$ ,simplified] this]
  show AE x in r.v x. ( $\lambda n. \varphi x (r + 1 / (rat-of-nat (Suc n)))$ )  $\longrightarrow \varphi x r$ 
    unfolding  $\varphi$ -def by eventually-elim (rule tendsto-enn2real, auto)
  qed

  have as3: AE x in r.v x. ( $\varphi x \longrightarrow 0$ ) at-bot
  proof -
    have 0: range ( $\lambda n. \{..- real n\}$ )  $\subseteq$  sets borel decseq ( $\lambda n. \{..- real n\}$ )
      by(auto simp: decseq-def)
    show ?thesis
  proof(safe intro!: AE-I2[THEN AE-mp[OF AE-conjI[OF r.kernel-empty0-AE
AE-conjI[OF r.kernel-lim-decseq[OF 0] as1]]]])
    fix x
    assume h:  $r.\kappa' x \{ \} = 0$  ( $\lambda n. r.\kappa' x \{..- real n\}$ )  $\longrightarrow r.\kappa' x (\bigcap n. \{..-
real n\}) \forall r s. r \leq s \longrightarrow \varphi x r \leq \varphi x s$ 
    have [simp]:  $(\bigcap n. \{..- real n\}) = \{ \}$  by auto (meson le-minus-iff linorder-not-less
reals-Archimedean2)
    show ( $\varphi x \longrightarrow 0$ ) at-bot
      proof(rule decreasing-tendsto)

```

```

fix  $r :: \text{real}$ 
assume  $0 < r$ 
with  $h(2)$  eventually-sequentially
obtain  $N$  where  $N: \bigwedge n. n \geq N \implies r.\kappa' x \{.. \text{real } n\} < r$ 
  by(fastforce simp: order-tendsto-iff h(1))
show  $\forall_F q$  in at-bot.  $\varphi x q < r$ 
  unfolding eventually-at-bot-linorder
proof(safe intro!: exI[where x=- rat-of-nat N])
  fix  $q$ 
  assume  $q \leq - \text{rat-of-nat } N$ 
  with  $h(3)$  have  $\varphi x q \leq \varphi x (- \text{rat-of-nat } N)$  by simp
  also have  $\dots < r$ 
    by(auto simp:  $\varphi$ -def) (metis N[OF order-refl]  $\langle 0 < r \rangle$  enn2real-less-iff
enn2real-top of-rat-minus of-rat-of-nat-eq top.not-eq-extremum)
  finally show  $\varphi x q < r$  .
  qed
qed(simp add:  $\varphi$ -def)
qed
qed

have  $as_4: AE x$  in  $r.\nu x.$  ( $\varphi x \longrightarrow 1$ ) at-top
proof -
  have  $0: \text{range } (\lambda n. \{.. \text{real } n\}) \subseteq \text{sets borel incseq } (\lambda n. \{.. \text{real } n\})$ 
    by(auto simp: incseq-def)
  have [simp]:  $(\bigcup n. \{.. \text{real } n\}) = UNIV$  by (auto simp: real-arch-simple)
  have  $1: AE x$  in  $r.\nu x.$   $\forall n. r.\kappa' x \{.. \text{real } n\} \leq 1$   $AE x$  in  $r.\nu x.$   $\forall q. r.\kappa' x$ 
 $\{.. \text{real-of-rat } q\} \leq 1$ 
    by(auto simp: AE-all-countable intro!: r.kernel-le1-AE)
  show ?thesis
    proof(safe intro!: AE-I2[THEN AE-mp[OF AE-conjI[OF AE-conjI[OF 1]
AE-conjI[OF r.kernel-Y1-AE AE-conjI[OF r.kernel-lim-incseq[OF 0] as1]]],simplified])
    fix  $x$ 
    assume  $h: \forall q. r.\kappa' x \{.. \text{real-of-rat } q\} \leq 1 \forall n. r.\kappa' x \{.. \text{real } n\} \leq 1$ 
       $(\lambda n. r.\kappa' x \{.. \text{real } n\}) \longrightarrow r.\kappa' x UNIV \forall r s. r \leq s \longrightarrow \varphi x r \leq$ 
 $\varphi x s$   $r.\kappa' x UNIV = 1$ 
    then have  $h3: (\lambda n. r.\kappa' x \{.. \text{real } n\}) \longrightarrow 1$ 
      by auto
    show ( $\varphi x \longrightarrow 1$ ) at-top
    proof(rule increasing-tendsto)
    fix  $r :: \text{real}$ 
    assume  $r < 1$ 
    with  $h3$  eventually-sequentially
    obtain  $N$  where  $N: \bigwedge n. n \geq N \implies r < r.\kappa' x \{.. \text{real } n\}$ 
      by(fastforce simp: order-tendsto-iff)
    show  $\forall_F n$  in at-top.  $r < \varphi x n$ 
      unfolding eventually-at-top-linorder
    proof(safe intro!: exI[where x=rat-of-nat N])
    fix  $q$ 
    assume  $\text{rat-of-nat } N \leq q$ 

```

```

      have r < φ x (rat-of-nat N)
      by(auto simp: φ-def) (metis N[OF order-refl] h(2) enn2real-1
enn2real-ennreal enn2real-positive-iff ennreal-cases ennreal-leI linorder-not-less zero-less-one)
      also have ... ≤ φ x q
      using h(4) ⟨rat-of-nat N ≤ q⟩ by simp
      finally show r < φ x q .
    qed
  qed(use h(1) enn2real-leI φ-def in auto)
  qed
  qed
  from AE-E3[OF AE-conjI[OF as1 AE-conjI[OF as2 AE-conjI[OF as3 as4]]],simplified
space-marginal-measure]
  obtain N where N: N ∈ null-sets r.νx ∧ x r s. x ∈ space X - N ⇒ r ≤ s
⇒ φ x r ≤ φ x s
      ∧ x r. x ∈ space X - N ⇒ (λn. φ x (r + 1 / rat-of-nat (Suc
n))) → φ x r
      ∧ x. x ∈ space X - N ⇒ (φ x → 0) at-bot ∧ x. x ∈ space X
- N ⇒ (φ x → 1) at-top
  by metis
  define F where F ≡ (λx y. indicat-real (space X - N) x * Inf {φ x r | r. y ≤
real-of-rat r} + indicat-real N x * indicat-real {0..} y)
  have [simp]: {φ x r | r. y ≤ real-of-rat r} ≠ {} for x y
  by auto (meson gt-ex less-eq-real-def of-rat-dense)
  have [simp]: bdd-below {φ x r | r. y ≤ real-of-rat r} if x ∈ space X - N for x y
  proof -
    obtain r' where real-of-rat r' ≤ y
    by (metis less-eq-real-def lt-ex of-rat-dense)
    from order-trans[OF this] of-rat-less-eq show ?thesis
    by(auto intro!: bdd-belowI[of - φ x r'] N(2)[OF that])
  qed
  have Feg: F x (real-of-rat r) = φ x r if x ∈ space X - N for x r
  using that N(2)[OF that] by(auto intro!: cInf-eq-minimum simp: of-rat-less-eq
F-def)
  have Fmono: mono (F x) if x ∈ space X for x
  by(auto simp: F-def mono-def indicator-def intro!: cInf-superset-mono) (meson
gt-ex less-eq-real-def of-rat-dense)

  have F1: (F x → 0) at-bot if x ∈ space X for x
  proof(cases x ∈ N)
    case True
    with that show ?thesis
    by(auto simp: F-def tendsto-iff eventually-at-bot-dense indicator-def intro!:
exI[where x=0])
  next
    case False
    with qlim-eq-lim-mono-at-bot[OF Fmono[OF that] N(4)] Feg that
    show ?thesis by auto
  qed
  have F2: (F x → 1) at-top if x ∈ space X for x

```



```

proof(cases  $x \in N$ )
  case True
    with that show ?thesis
      by(auto simp: F-def tendsto-iff eventually-at-top-dense indicator-def intro!:
exI[where  $x=0$ ])
  next
    case False
      with qlim-eq-lim-mono-at-top[OF Fmono[OF that]  $N(5)$ ] Feq that
      show ?thesis by auto
qed
have F3: continuous (at-right a) (F x) if  $x \in \text{space } X$  for  $x a$ 
proof(cases  $x \in N$ )
  case  $x:\textit{True}$ 
    {
      fix  $e :: \textit{real}$ 
      assume  $e:0 < e$ 
      consider  $a \geq 0 \mid a < 0$  by fastforce
      then have  $\exists d>0. \textit{indicat-real } \{0..\} (a + d) - \textit{indicat-real } \{0..\} a < e$ 
      proof cases
        case 1
          with  $e$  show ?thesis
            by(auto intro!: exI[where  $x=1$ ])
        next
          case 2
            then obtain  $b$  where  $b > 0 \ a + b < 0$ 
            by (metis add-less-same-cancel2 of-rat-dense real-add-less-0-iff)
            with  $e$  2 show ?thesis
              by(auto intro!: exI[where  $x=b$ ])
          qed
        }
      with  $x$  show ?thesis
      unfolding continuous-at-right-real-increasing[of F x, OF monoD[OF Fmono[OF
that]],simplified]
      by(auto simp: F-def)
  next
    case  $x:\textit{False}$ 
      {
        fix  $e :: \textit{real}$ 
        assume  $e: e > 0$ 
        have  $\exists k. a \leq \textit{real-of-rat } k \wedge \bigcap \{\varphi x r \mid r. a \leq \textit{real-of-rat } r\} + e / 2 \geq \varphi$ 
x k
        proof(rule ccontr)
          assume  $\nexists k. a \leq \textit{real-of-rat } k \wedge \varphi x k \leq \bigcap \{\varphi x r \mid r. a \leq \textit{real-of-rat } r\}$ 
           $+ e / 2$ 
          then have cont:  $\bigwedge k. a \leq \textit{real-of-rat } k \implies \varphi x k - e / 2 > \bigcap \{\varphi x r \mid r. a \leq \textit{real-of-rat } r\}$ 
          by auto
          hence  $a \leq \textit{real-of-rat } k \implies \exists r. a \leq \textit{real-of-rat } r \wedge \varphi x r < \varphi x k - e / 2$ 
          for  $k$ 
      }

```

```

    using cont ⟨x ∈ space X⟩ x cInf-less-iff [of {φ x r | r. a ≤ real-of-rat r}
φ x k - e / 2]
    by auto
    then obtain r where r: ∧k. a ≤ real-of-rat k ⇒ a ≤ real-of-rat (r k)
∧k. a ≤ real-of-rat k ⇒ φ x (r k) < φ x k - e / 2
    by metis
    obtain k where k: a ≤ real-of-rat k
    by (meson gt-ex less-eq-real-def of-rat-dense)
    define f where f ≡ rec-nat k (λn fn. r fn)
    have f-simp: f 0 = k f (Suc n) = r (f n) for n
    by(auto simp: f-def)
    have f1: a ≤ real-of-rat (f n) for n
    using r(1) k by(induction n) (auto simp: f-simp)
    have f2: n ≥ 1 ⇒ φ x (f n) < φ x k - real n * e / 2 for n
    proof(induction n)
    case ih:(Suc n)
    consider n = 0 | n ≥ 1 by fastforce
    then show ?case
    proof cases
    case 1
    with r k show ?thesis
    by(simp add: f-simp)
    next
    case 2
    show ?thesis
    using less-trans[OF r(2)[OF f1[of n]] diff-strict-right-mono[OF
ih(1)[OF 2],of e / 2]]
    by(auto simp: f-simp ring-distrib(2) add-divide-distrib)
    qed
    qed simp
    have ¬ bdd-below {φ x r | r. a ≤ real-of-rat r}
    unfolding bdd-below-def
    proof safe
    fix M
    obtain n where φ x k - M < real n * e / 2
    using f2 e reals-Archimedean3 by fastforce
    then have φ x k - M < real (Suc n) * e / 2
    using divide-strict-right-mono pos-divide-less-eq e by fastforce
    thus Ball {φ x r | r. a ≤ real-of-rat r} ((≤) M) ⇒ False
    using f2[of Suc n] f1[of Suc n] by(auto intro!: exI[where x=φ x (f
(Suc n))])
    qed
    with that x show False
    by simp
    qed
    then obtain k where k: a ≤ real-of-rat k ∧ {φ x r | r. a ≤ real-of-rat r}
+ e / 2 ≥ φ x k
    by auto
    obtain no where no: ∧n. n ≥ no ⇒ (φ x (k + 1 / rat-of-nat (Suc n))) -

```

```

( $\varphi x k$ ) <  $e / 2$ 
  using  $\langle x \in \text{space } X \rangle x \text{ metric-LIMSEQ-D}[OF N(3)[of x k], of e/2] e N(2)[of$ 
 $x k k + 1 / \text{rat-of-nat } (Suc -)]$ 
  by(auto simp: dist-real-def)
  have  $\exists d > 0. \sqcap \{ \varphi x r \mid r. a + d \leq \text{real-of-rat } r \} - \sqcap \{ \varphi x r \mid r. a \leq$ 
 $\text{real-of-rat } r \} < e$ 
  proof(safe intro!: exI[where  $x = \text{real-of-rat } (1 / \text{rat-of-nat } (Suc no))$ ])
    have  $\varphi x (k + 1 / \text{rat-of-nat } (Suc no)) - e < \varphi x k - e / 2$ 
      using no[OF order-refl] by simp
    also have  $\dots \leq \sqcap \{ \varphi x r \mid r. a \leq \text{real-of-rat } r \}$ 
      using k by simp
    finally have  $\varphi x (k + 1 / \text{rat-of-nat } (Suc no)) - \sqcap \{ \varphi x r \mid r. a \leq$ 
 $\text{real-of-rat } r \} < e$  by simp
    moreover have  $\sqcap \{ \varphi x r \mid r. a + \text{real-of-rat } (1 / (1 + \text{rat-of-nat } no)) \leq$ 
 $\text{real-of-rat } r \} \leq \varphi x (k + 1 / \text{rat-of-nat } (Suc no))$ 
      using k that x by(auto intro!: cInf-lower simp: of-rat-add)
    ultimately show  $\sqcap \{ \varphi x r \mid r. a + \text{real-of-rat } (1 / (\text{rat-of-nat } (Suc no)))$ 
 $\leq \text{real-of-rat } r \} - \sqcap \{ \varphi x r \mid r. a \leq \text{real-of-rat } r \} < e$ 
      by simp
    qed simp
  }
  with that x show ?thesis
  unfolding continuous-at-right-real-increasing[of F x, OF monoD[OF Fmono[OF
that]],simplified]
  by(auto simp: F-def)
qed

define  $\kappa$  where  $\kappa \equiv (\lambda x. \text{interval-measure } (F x))$ 

have  $\kappa: \bigwedge x. x \in \text{space } X \implies \kappa x \text{ UNIV} = 1$ 
   $\bigwedge x r. x \in \text{space } X \implies \kappa x \{..r\} = \text{ennreal } (F x r)$ 
and[simp]:  $\bigwedge x. \text{sets } (\kappa x) = \text{sets borel } \bigwedge x. \text{space } (\kappa x) = \text{UNIV}$ 
  using emeasure-interval-measure-Iic[OF - F3 F1] interval-measure-UNIV[OF
- F3 F1 F2] Fmono
  by(auto simp: mono-def  $\kappa$ -def)

interpret  $\kappa$ : prob-kernel X borel  $\kappa$ 
  unfolding prob-kernel-def'
proof(rule measurable-prob-algebra-generated[OF - atMostq-Int-stable, of - UNIV])
  show  $\bigwedge a. a \in \text{space } X \implies \text{prob-space } (\kappa a)$ 
    by(auto intro!: prob-spaceI  $\kappa(1)$ )
next
fix A
  assume  $A \in \{ \{..r\} \mid r::\text{real}. r \in \mathbb{Q} \}$ 
  then obtain r where  $r: A = \{.. \text{real-of-rat } r \}$ 
    using Rats-cases by blast
  have  $(\lambda x. \text{ennreal } (\text{indicat-real } (\text{space } X - N) x * \varphi x r + \text{indicat-real } N x$ 
 $* \text{indicat-real } \{0..\} (\text{real-of-rat } r))) \in \text{borel-measurable } X$ 
  proof -

```

**have**  $N \in \text{sets } X$   
**using** *null-setsD2*[*OF*  $N(1)$ ] **by**(*auto simp: sets-marginal-measure*)  
**thus** *?thesis* **by**(*auto simp:  $\varphi$ -def*)  
**qed**  
**moreover have** *indicat-real (space  $X - N$ )  $x * \varphi x r + \text{indicat-real } N x * \text{indicat-real } \{0..\}$  (real-of-rat  $r$ ) = *emeasure* ( $\kappa x$ )  $A$  **if**  $x \in \text{space } X$  **for**  $x$   
**using** *Feq*[*of*  $x r$ ]  $\kappa(2)$ [*OF that, of real-of-rat r*]  
**by**(*cases  $x \in N$* ) (*auto simp: r indicator-def F-def*)  
**ultimately show** ( $\lambda x. \text{emeasure} (\kappa x) A$ )  $\in$  *borel-measurable*  $X$   
**using** *measurable-cong*[*of -  $\lambda x. \text{emeasure} (\kappa x) A \lambda x. \text{ennreal (indicat-real (space } X - N) x * \varphi x r + \text{indicat-real } N x * \text{indicat-real } \{0..\}$  (real-of-rat  $r$ ))*]  
**by** *simp*  
**qed**(*auto simp: rborel-eq-atMostq*)  
**have**  $\kappa$ -*AE*: *AE*  $x$  *in*  $r.\nu x. \kappa x \{.. \text{real-of-rat } r\} = r.\kappa' x \{.. \text{real-of-rat } r\}$  **for**  $r$   
**proof -**  
**have** *AE*  $x$  *in*  $r.\nu x. \kappa x \{.. \text{real-of-rat } r\} = \text{ennreal} (F x (\text{real-of-rat } r))$   
**by**(*auto simp: space-marginal-measure  $\kappa(2)$* )  
**moreover have** *AE*  $x$  *in*  $r.\nu x. \text{ennreal} (F x (\text{real-of-rat } r)) = \text{ennreal} (\varphi x r)$   
**using** *Feq*[*of - r*] **by**(*auto simp add: space-marginal-measure intro!: AE-I''[OF*  
 $N(1)]$ )  
**moreover have** *AE*  $x$  *in*  $r.\nu x. \text{ennreal} (\varphi x r) = \text{ennreal} (\text{enn2real} (r.\kappa' x$   
 $\{.. \text{real-of-rat } r\}))$   
**by**(*simp add:  $\varphi$ -def*)  
**moreover have** *AE*  $x$  *in*  $r.\nu x. \text{ennreal} (\text{enn2real} (r.\kappa' x \{.. \text{real-of-rat } r\})) =$   
 $r.\kappa' x \{.. \text{real-of-rat } r\}$   
**using** *r.kernel-le-infty*[*of*  $\{.. \text{real-of-rat } r\}, \text{simplified}$ ]  
**by**(*auto simp: ennreal-enn2real-if*)  
**ultimately show** *?thesis* **by** *auto*  
**qed**  
**have**  $\kappa$ -*dis*:  $\kappa$ .*disintegration*  $\nu r.\nu x$   
**proof -**  
**interpret**  $D$ : *Dynkin-system*  $UNIV \{B \in \text{sets borel}. \forall A \in \text{sets } X. \nu (A \times B)$   
 $= (\int^{+x \in A. (\kappa x) B \partial r.\nu x})\}$   
**proof**  
**{**  
**fix**  $A$   
**assume**  $h: A \in \text{sets } X$   
**then have**  $\nu (A \times UNIV) = (\int^{+x \in A. 1 \partial r.\nu x})$   
**using** *emeasure-marginal-measure*[*OF nu-sets' h*] *sets-marginal-measure*[*of*  
 $X \text{ borel } \nu \text{ space borel}$ ] **by** *auto*  
**also have**  $\dots = (\int^{+x \in A. (\kappa x) UNIV \partial r.\nu x})$   
**by**(*auto intro!: nn-integral-cong simp:  $\kappa$  space-marginal-measure*)  
**finally have**  $\nu (A \times UNIV) = (\int^{+x \in A. \text{emeasure} (\kappa x) UNIV \partial r.\nu x}) .$   
**}**  
**thus**  $UNIV \in \{B \in \text{sets borel}. \forall A \in \text{sets } X. \text{emeasure } \nu (A \times B) = (\int^{+x \in A.}$   
 $\text{emeasure} (\kappa x) B \partial r.\nu x)\}$   
**by** *auto*  
**hence** *univ*:  $\bigwedge A. A \in \text{sets } X \implies \nu (A \times UNIV) = (\int^{+x \in A. \text{emeasure} (\kappa}$   
 $x) UNIV \partial r.\nu x)$  **by** *auto**

```

show  $\bigwedge B. B \in \{B \in \text{sets borel. } \forall A \in \text{sets } X. \text{emeasure } \nu (A \times B) = (\int^{+x \in A. \text{emeasure } (\kappa x) B \partial r. \nu x)\}$ 
 $\implies UNIV - B \in \{B \in \text{sets borel. } \forall A \in \text{sets } X. \text{emeasure } \nu (A \times B) = (\int^{+x \in A. \text{emeasure } (\kappa x) B \partial r. \nu x)\}$ 
proof(rule r.nu.x.sigma-finite-disjoint)
  fix  $B$  and  $J :: \text{nat} \implies -$ 
  assume  $B \in \{B \in \text{sets borel. } \forall A \in \text{sets } X. \text{emeasure } \nu (A \times B) = (\int^{+x \in A. \text{emeasure } (\kappa x) B \partial r. \nu x)\}$ 
   $\text{range } J \subseteq \text{sets } r. \nu x \cup (\text{range } J) = \text{space } r. \nu x$ 
   $(\bigwedge i. \text{emeasure } r. \nu x (J i) \neq \infty)$  disjoint-family  $J$ 
  then have  $B: B \in \text{sets borel } \forall A \in \text{sets } X. \nu (A \times B) = (\int^{+x \in A. (\kappa x) B \partial r. \nu x}$ 
  and  $J: \text{range } J \subseteq \text{sets } X \cup (\text{range } J) = \text{space } X \bigwedge i. \text{emeasure } r. \nu x (J i) \neq \infty$  disjoint-family  $J$ 
  by (auto simp: sets-marginal-measure space-marginal-measure)
  {
    fix  $A$ 
    assume  $A: A \in \text{sets } X$ 
    have  $\nu (A \times (UNIV - B)) = (\int^{+x \in A. (\kappa x) (UNIV - B) \partial r. \nu x)$  (is
     $?lhs = ?rhs$ )
    proof -
      have  $AJi1: \text{disjoint-family } (\lambda i. (A \cap J i) \times (UNIV - B))$ 
      using  $B(1) J(4)$  by (fastforce simp: disjoint-family-on-def)
      have  $AJi2[\text{simp}]: (\bigcup i. ((A \cap J i) \times (UNIV - B))) = A \times (UNIV - B)$ 
      using  $J(2)$  sets.sets-into-space[OF  $A$ ] by blast
      have  $AJi3: (\text{range } (\lambda i. (A \cap J i) \times (UNIV - B))) \subseteq \text{sets } \nu$ 
      using  $B(1) J(1) A$  by (auto simp: nu-sets')

      have  $?lhs = (\sum i. \nu ((A \cap J i) \times (UNIV - B)))$ 
      by (simp add: suminf-emeasure[OF  $AJi3 AJi1$ ])
      also have  $\dots = (\sum i. (\int^{+x \in A \cap J i. (\kappa x) (UNIV - B) \partial r. \nu x))$ 
      proof (safe intro!: suminf-cong)
        fix  $n$ 
        have  $Jn: J n \in \text{sets } X$ 
        using  $J$  by auto
        have  $fin: \nu ((A \cap J n) \times C) \neq \infty$  for  $C$ 
        proof (cases  $(A \cap J n) \times C \in \text{sets } \nu$ )
          case True
          then have  $\nu ((A \cap J n) \times C) \leq \nu ((A \cap J n) \times UNIV)$ 
          using  $Jn$  nu-sets' A by (intro emeasure-mono) auto
          also have  $\nu ((A \cap J n) \times UNIV) \leq \nu (J n \times UNIV)$ 
          using  $Jn$  nu-sets' by (intro emeasure-mono) auto
          also have  $\dots = r. \nu x (J n)$ 
          using emeasure-marginal-measure[OF  $nu\text{-sets}' Jn$ ] by simp
          finally show  $?thesis$ 
          by (metis  $J(3)$ [of  $n$ ] infinity-ennreal-def neq-top-trans)
        qed (simp add: emeasure-notin-sets)
        show  $\nu ((A \cap J n) \times (UNIV - B)) = (\int^{+x \in A \cap J n. (\kappa x) (UNIV - B) \partial r. \nu x)$  (is
         $?lhs = ?rhs$ )

```

```

proof –
  have ?lhs =  $\nu ((A \cap J n) \times UNIV) - \nu ((A \cap J n) \times B)$ 
  proof –
    have [simp]: ?lhs +  $\nu ((A \cap J n) \times B) = \nu ((A \cap J n) \times UNIV)$ 
    proof –
      have [simp]:  $((A \cap J n) \times (UNIV - B)) \cup ((A \cap J n) \times B) =$ 
 $((A \cap J n) \times UNIV)$  by blast
      show ?thesis
      using B(1) A Jn nu-sets' by(intro plus-emeasure[of (A  $\cap$  J n)
 $\times$  (UNIV - B) - (A  $\cap$  J n)  $\times$  B,simplified]) auto
    qed
    have ?lhs = ?lhs +  $\nu ((A \cap J n) \times B) - \nu ((A \cap J n) \times B)$ 
    by(simp only: ennreal-add-diff-cancel[OF fin[of B]])
    also have ... =  $\nu ((A \cap J n) \times UNIV) - \nu ((A \cap J n) \times B)$ 
    by simp
    finally show ?thesis .
  qed
  also have ... =  $(\int^{+x \in A \cap J n. (\kappa x) UNIV} \partial r. \nu x) - (\int^{+x \in A$ 
 $\cap J n. (\kappa x) B} \partial r. \nu x)$ 
    using B(2) A Jn univ by auto
    also have ... =  $(\int^{+x. ((\kappa x) UNIV * indicator (A \cap J n) x) - (\kappa$ 
 $x) B * indicator (A \cap J n) x} \partial r. \nu x)$ 
    proof(rule nn-integral-diff[symmetric])
    show  $(\lambda x. (\kappa x) UNIV * indicator (A \cap J n) x) \in$  borel-measurable
 $r. \nu x$   $(\lambda x. (\kappa x) B * indicator (A \cap J n) x) \in$  borel-measurable  $r. \nu x$ 
    using sets-marginal-measure[of X borel  $\nu$  space borel]
 $\kappa.emeasure-measurable$ [OF B(1)]  $\kappa.emeasure-measurable$ [of UNIV] A Jn
    by(auto simp del: space-borel)
  next
    show  $(\int^{+x \in A \cap J n. (\kappa x) B} \partial r. \nu x) \neq \infty$ 
    using B(2) A Jn univ fin[of B] by auto
  next
    show  $A E x$  in  $r. \nu x. (\kappa x) B * indicator (A \cap J n) x \leq (\kappa x)$ 
 $UNIV * indicator (A \cap J n) x$ 
    by(standard, auto simp: space-marginal-measure indicator-def
intro!: emeasure-mono)
  qed
  also have ... =  $(\int^{+x \in A \cap J n. ((\kappa x) UNIV - (\kappa x) B)} \partial r. \nu x)$ 
    by(auto intro!: nn-integral-cong simp: indicator-def)
  also have ... = ?rhs
  proof(safe intro!: nn-integral-cong)
    fix x
    assume  $x \in$  space  $r. \nu x$ 
    then have  $x \in$  space X
    by(simp add: space-marginal-measure)
    show  $((\kappa x) UNIV - (\kappa x) B) * indicator (A \cap J n) x = (\kappa x)$ 
 $(UNIV - B) * indicator (A \cap J n) x$ 
    by(auto intro!: emeasure-compl[of B  $\kappa x$ ,simplified,symmetric] simp:
B  $\kappa$ .prob-spaces  $\langle x \in$  space X  $\rangle$  prob-space-imp-subprob-space subprob-space.emeasure-subprob-space-less-top

```

```

indicator-def)
  qed
  finally show ?thesis .
  qed
  qed
  also have ... = ( $\int^{+x}. (\sum i. (\kappa x) (UNIV - B) * indicator (A \cap J i) x) \partial r. \nu x$ )
  using  $\kappa$ .emeasure-measurable[of UNIV - B] B(1) sets-marginal-measure[of X borel  $\nu$  space borel] A J
  by(intro nn-integral-suminf[symmetric]) (auto simp del: space-borel)
  also have ... = ( $\int^{+x}. (\kappa x) (UNIV - B) * indicator A x * (\sum i. indicator (A \cap J i) x) \partial r. \nu x$ )
  by(auto simp: indicator-def intro!: nn-integral-cong)
  also have ... = ( $\int^{+x}. (\kappa x) (UNIV - B) * indicator A x * (indicator (\bigcup i. A \cap J i) x) \partial r. \nu x$ )
  proof -
    have ( $\sum i. indicator (A \cap J i) x$ ) = (indicator ( $\bigcup i. A \cap J i$ ) x :: ennreal) for x
  using J(4) by(intro suminf-indicator) (auto simp: disjoint-family-on-def)
  thus ?thesis
  by(auto intro!: nn-integral-cong)
  qed
  also have ... = ?rhs
  using J(2) by(auto simp: indicator-def space-marginal-measure intro!: nn-integral-cong)
  finally show ?thesis .
  qed
}
thus UNIV - B  $\in$  {B  $\in$  sets borel.  $\forall A \in$  sets X. emeasure  $\nu$  (A  $\times$  B) = ( $\int^{+x \in A}. \kappa x B \partial r. \nu x$ )}
using B by auto
qed
next
fix J :: nat  $\Rightarrow$  -
assume J1: disjoint-family J range J  $\subseteq$  {B  $\in$  sets borel.  $\forall A \in$  sets X.  $\nu$  (A  $\times$  B) = ( $\int^{+x \in A}. (\kappa x) B \partial r. \nu x$ )}
then have J2: range J  $\subseteq$  sets borel  $\bigcup$  (range J)  $\in$  sets borel  $\bigwedge n$  A. A  $\in$  sets X  $\implies \nu$  (A  $\times$  (J n)) = ( $\int^{+x \in A}. (\kappa x) (J n) \partial r. \nu x$ )
by auto
show  $\bigcup$  (range J)  $\in$  {B  $\in$  sets borel.  $\forall A \in$  sets X.  $\nu$  (A  $\times$  B) = ( $\int^{+x \in A}. (\kappa x) B \partial r. \nu x$ )}
proof -
{
  fix A
  assume A: A  $\in$  sets X
  have  $\nu$  (A  $\times$   $\bigcup$  (range J)) = ( $\int^{+x \in A}. (\kappa x) (\bigcup$  (range J))  $\partial r. \nu x$ ) (is ?lhs = ?rhs)
  proof -
    have ?lhs =  $\nu$  ( $\bigcup n. A \times J n$ )

```

```

proof –
  have  $(A \times \bigcup (\text{range } J)) = (\bigcup n. A \times J n)$  by blast
  thus ?thesis by simp
qed
also have  $\dots = (\sum n. \nu (A \times J n))$ 
using J1(1) J2(1) A nu-sets' by (fastforce intro!: suminf-emeasure[symmetric])
simp: disjoint-family-on-def
also have  $\dots = (\int^{+x \in A. (\kappa x) (J n) \partial r. \nu x})$ 
by (simp add: J2(3)[OF A])
also have  $\dots = (\int^{+x. (\sum n. (\kappa x) (J n) * \text{indicator } A x) \partial r. \nu x})$ 
using  $\kappa$ .emeasure-measurable J2(1) A sets-marginal-measure[of X
borel  $\nu$  space borel]
by (intro nn-integral-suminf[symmetric]) auto
also have  $\dots = (\int^{+x \in A. (\sum n. (\kappa x) (J n)) \partial r. \nu x})$ 
by auto
also have  $\dots = (\int^{+x \in A. (\kappa x) (\bigcup (\text{range } J)) \partial r. \nu x})$ 
using J1 J2 by (auto intro!: nn-integral-cong suminf-emeasure simp:
space-marginal-measure indicator-def)
finally show ?thesis .
qed
}
thus ?thesis
using J2(2) by auto
qed
qed auto
have  $\{ \{..r\} \mid r::\text{real. } r \in \mathbf{Q} \} \subseteq \{ B \in \text{sets borel. } \forall A \in \text{sets } X. \nu (A \times B) =$ 
 $(\int^{+x \in A. (\kappa x) B \partial r. \nu x}) \}$ 
proof –
{
fix r ::real and A
assume h: r ∈ Q A ∈ sets X
then obtain r' where r':r = real-of-rat r'
using Rats-cases by blast
have  $\nu (A \times \{..r\}) = (\int^{+x \in A. (\kappa x) \{..r\} \partial r. \nu x)$  (is ?lhs = ?rhs)
proof –
have ?lhs  $= (\int^{+x \in A. r. \kappa' x \{..r\} \partial r. \nu x)$ 
using h by (simp add: r.kernel-RN-deriv)
also have  $\dots = ?rhs$ 
using  $\kappa$ -AE[of r'] by (auto intro!: nn-integral-cong-AE simp: r' simp
del: space-borel)
finally show ?thesis .
qed
}
thus ?thesis
by auto
qed
from D.Dynkin-subset[OF this] rborel-eq-atMostq[symmetric]
show ?thesis
by (auto simp:  $\kappa$ .disintegration-def sets-marginal-measure nu-sets' sigma-eq-Dynkin[OF

```



```

- atMostq-Int-stable,of UNIV,simplified,symmetric] rborel-eq-atMostq-sets simp del:
space-borel)
  qed
  show ?thesis
  proof(intro exI conjI strip)
    fix  $\kappa''$ 
    assume prob-kernel X (borel :: real measure)  $\kappa''$ 
    interpret  $\kappa''$ : prob-kernel X borel  $\kappa''$  by fact
    assume disi:  $\kappa''$ .disintegration  $\nu$   $r.\nu x$ 
    have eq-atMostr-AE:AE x in  $r.\nu x$ .  $\forall r. \kappa x \{..real-of-rat\ r\} = \kappa'' x \{..real-of-rat$ 
r}
      unfolding AE-all-countable
      proof safe
        fix r
        have AE x in  $r.\nu x$ . ( $\kappa'' x \{..real-of-rat\ r\} = r.\kappa' x \{..real-of-rat\ r\}$ )
        proof(safe intro!:  $r.\nu x$ .RN-deriv-unique[of  $\lambda x. \kappa'' x \{..real-of-rat\ r\}$  marginal-measure-on
X borel  $\nu \{..real-of-rat\ r\}$ ,simplified  $r.\kappa'$ -def[of -  $\{..real-of-rat\ r\}$ ,symmetric]])
          show 1:( $\lambda x$ . emeasure ( $\kappa'' x \{..real-of-rat\ r\}) \in$  borel-measurable  $r.\nu x$ )
          using  $\kappa''$ .emeasure-measurable[of  $\{..real-of-rat\ r\}$ ] sets-marginal-measure[of
X borel  $\nu$  space borel] by simp
          show density  $r.\nu x$  ( $\lambda x$ . emeasure ( $\kappa'' x \{..real-of-rat\ r\}) =$  marginal-measure-on
X borel  $\nu \{..real-of-rat\ r\}$ )
          proof(rule measure-eqI)
            fix A
            assume  $A \in$  sets (density  $r.\nu x$  ( $\lambda x$ . ( $\kappa'' x \{..real-of-rat\ r\}$ )))
            then have A [measurable]: $A \in$  sets X
              by(simp add: sets-marginal-measure)
            show emeasure (density  $r.\nu x$  ( $\lambda x$ . emeasure ( $\kappa'' x \{..real-of-rat\ r\}$ ))) A
= emeasure (marginal-measure-on X borel  $\nu \{..real-of-rat\ r\}) A$  (is ?lhs = ?rhs)
            proof -
              have ?lhs = ( $\int^{+x \in A. (\kappa'' x \{..real-of-rat\ r\}) \partial r.\nu x$ )
              using emeasure-density[OF 1,of A] A
              by(simp add: sets-marginal-measure)
              also have ... =  $\nu (A \times \{..real-of-rat\ r\})$ 
              using disi A by(auto simp:  $\kappa''$ .disintegration-def)
              also have ... = ?rhs
              by(simp add: emeasure-marginal-measure-on[OF nu-sets' - A])
            finally show ?thesis .
          qed
        qed(simp add: sets-marginal-measure)
      qed
    with  $\kappa$ -AE[of r]
    show AE x in  $r.\nu x$ .  $\kappa x \{..real-of-rat\ r\} = \kappa'' x \{..real-of-rat\ r\}$ 
      by auto
  qed
  { fix x
    assume h: $x \in$  space  $r.\nu x \ \forall r$ . ( $\kappa x \{..real-of-rat\ r\} = \kappa'' x \{..real-of-rat$ 
r}
    then have x:  $x \in$  space X

```

```

    by(simp add: space-marginal-measure)
  have  $\kappa x = \kappa'' x$ 
  proof(rule measure-eqI-generator-eq[OF atMostq-Int-stable, of UNIV - -  $\lambda n$ .
  {..real n}])
    show  $\bigwedge A. A \in \{\{..r\} \mid r. r \in \mathbf{Q}\} \implies (\kappa x) A = (\kappa'' x) A$ 
      using h(2) Rats-cases by auto
    next
    show  $(\bigcup n. \{..real n\}) = UNIV$ 
      by (simp add: real-arch-simple subsetI subset-antisym)
    next
    fix  $n$ 
    have  $(\kappa x) \{..real n\} \leq \kappa x UNIV$ 
      by(auto intro!: emeasure-mono)
    also have  $\dots = 1$ 
      by(rule  $\kappa(1)[OF x]$ )
    finally show  $(\kappa x) \{..real n\} \neq \infty$ 
      using linorder-not-le by fastforce
    next
    show  $range (\lambda n. \{..real n\}) \subseteq \{\{..r\} \mid r. r \in \mathbf{Q}\}$ 
      using Rats-of-nat by blast
  qed(auto simp:  $\kappa$ .kernel-sets[OF x]  $\kappa''$ .kernel-sets[OF x] rborel-eq-atMostq-sets)
}
then show AE x in  $r.\nu x$ .  $\kappa x = \kappa'' x$ 
  using eq-atMostr-AE by fastforce
qed(auto simp del: space-borel simp add:  $\kappa$ -dis  $\kappa$ .prob-kernel-axioms)
qed

show ?thesis
proof -
  define  $\nu'$  where  $\nu' = distr \nu (X \otimes_M borel) (\lambda(x,y). (x, to-real y))$ 
  have  $\nu$ -distr: $\nu = distr \nu' (X \otimes_M Y) (\lambda(x,y). (x, from-real y))$ 
    using nu-sets sets-eq-imp-space-eq[OF nu-sets] from-real-to-real
  by(auto simp:  $\nu'$ -def distr-distr space-pair-measure intro!: distr-id'[symmetric])
  have  $\nu x$ -eq:(marginal-measure X borel  $\nu'$ ) =  $\nu x$ 
  using emeasure-marginal-measure[of  $\nu' X borel$ ] emeasure-marginal-measure[OF
  nu-sets] sets-eq-imp-space-eq[OF nu-sets]
  by(auto intro!: measure-eqI simp: sets-marginal-measure  $\nu'$ -def emeasure-distr
  map-prod-vimage[of id to-real, simplified map-prod-def id-def] space-pair-measure
  Times-Int-Times)
  interpret  $\nu'$ : projection-sigma-finite X borel  $\nu'$ 
  by(auto simp: projection-sigma-finite-def  $\nu x$ -eq  $\nu x$ .sigma-finite-measure-axioms
  simp del: space-borel, auto simp add:  $\nu'$ -def)
  obtain  $\kappa'$  where  $\kappa'$ : prob-kernel X borel  $\kappa'$  measure-kernel.disintegration X
  borel  $\kappa' \nu' \nu'.\nu x$ 
     $\bigwedge \kappa''. prob-kernel X borel \kappa'' \implies measure-kernel.disintegration X borel \kappa''
  \nu' \nu'.\nu x \implies (AE x in  $\nu'.\nu x$ .  $\kappa' x = \kappa'' x$ )$ 
    using *[of  $\nu' X$ ]  $\nu'$ .nu-sets  $\nu'.\nu x$ .sigma-finite-measure-axioms by blast
  interpret  $\kappa'$ : prob-kernel X borel  $\kappa'$  by fact
  define  $\kappa$  where  $\kappa \equiv (\lambda x. distr (\kappa' x) Y from-real)$ 

```

```

interpret  $\kappa$ : prob-kernel  $X Y \kappa$ 
  by(auto simp: prob-kernel-def'  $\kappa$ -def)
have disi:  $\kappa$ .disintegration  $\nu \nu x$ 
proof(rule  $\kappa$ .disintegrationI)
  fix  $A B$ 
  assume  $A$ [measurable]:  $A \in \text{sets } X$  and  $B$ [measurable]:  $B \in \text{sets } Y$ 
  have [measurable]: from-real -'  $B \in \text{sets borel}$ 
    by(simp add: measurable-sets-borel[OF - B])
  show  $\nu (A \times B) = (\int^{+x \in A. \kappa x B} \partial \nu x)$  (is ?lhs = ?rhs)
  proof -
    have ?lhs =  $\nu' (A \times (\text{from-real -' } B))$ 
    by(auto simp:  $\nu$ -distr emeasure-distr map-prod-vimage[of id from-real,simplified
map-prod-def id-def])
    also have ... =  $(\int^{+x \in A. \kappa' x (\text{from-real -' } B)} \partial \nu x)$ 
      using  $\kappa'$ .disintegrationD[OF  $\kappa'(\varnothing)$ ,of A from-real -' B]
      by(auto simp add:  $\nu x$ -eq simp del: space-borel)
    also have ... = ?rhs
    by(auto intro!: nn-integral-cong simp: space-marginal-measure  $\kappa$ -def emeasure-distr)
  finally show ?thesis .
qed
qed(simp-all add: sets-marginal-measure nu-sets)

show ?thesis
proof(safe intro!: exI[where x= $\kappa$ ])
  fix  $\kappa''$ 
  assume  $h$ :prob-kernel  $X Y \kappa''$ 
    measure-kernel.disintegration  $X Y \kappa'' \nu \nu x$ 
  interpret  $\kappa''$ : prob-kernel  $X Y \kappa''$  by fact
  show  $A E x$  in  $\nu x. \kappa x = \kappa'' x$ 
  proof -
    define  $\kappa'''$  where  $\kappa''' \equiv (\lambda x. \text{distr } (\kappa'' x) \text{ borel to-real})$ 
    interpret  $\kappa'''$ : prob-kernel  $X \text{ borel } \kappa'''$ 
      by(auto simp: prob-kernel-def'  $\kappa'''$ -def)
    have  $\kappa''$ -def:  $\kappa'' x = \text{distr } (\kappa''' x) Y \text{ from-real if } x \in \text{space } X \text{ for } x$ 
      using distr-distr[of from-real borel Y to-real  $\kappa'' x$ ,simplified measurable-cong-sets[OF  $\kappa''$ .kernel-sets[OF that] refl,of borel]]
      by(auto simp:  $\kappa'''$ -def comp-def  $\kappa''$ .kernel-sets[OF that] measurable-cong-sets[OF  $\kappa''$ .kernel-sets[OF that]  $\kappa''$ .kernel-sets[OF that]] sets-eq-imp-space-eq[OF  $\kappa''$ .kernel-sets[OF that]] intro!: distr-id'[symmetric])
    have  $\kappa'''$ -disi:  $\kappa'''$ .disintegration  $\nu' \nu'.\nu x$ 
    proof(rule  $\kappa'''$ .disintegrationI)
      fix  $A$  and  $B :: \text{real set}$ 
      assume  $A$ [measurable]:  $A \in \text{sets } X$  and  $B$ [measurable]:  $B \in \text{sets borel}$ 
      show  $\nu' (A \times B) = (\int^{+x \in A. (\kappa''' x) B} \partial \nu'.\nu x)$  (is ?lhs = ?rhs)
      proof -
        have ?lhs =  $\nu (A \times (\text{to-real -' } B \cap \text{space } Y))$ 
        by(auto simp:  $\nu'$ -def emeasure-distr map-prod-vimage[of id to-real,simplified
map-prod-def id-def] sets-eq-imp-space-eq[OF nu-sets] space-pair-measure Times-Int-Times)

```

**also have** ... =  $(\int^{+} x \in A. (\kappa'' x) (to-real - ' B \cap space Y) \partial \nu x)$   
**using**  $\kappa''.disintegrationD[OF h(2) A, of to-real - ' B \cap space Y]$  **by**  
*auto*  
**also have** ... = ?rhs  
**by**(*auto simp:  $\nu x$ -eq[symmetric] space-marginal-measure  $\kappa'''$ -def emea-*  
*sure-distr sets-eq-imp-space-eq[OF  $\kappa''$ .kernel-sets] intro!: nn-integral-cong*)  
**finally show** ?thesis .  
**qed**  
**qed**(*auto simp:  $\nu'$ -def sets-marginal-measure*)  
**show** ?thesis  
**by**(*rule AE-mp[OF  $\kappa'(\mathcal{B})$ ][OF  $\kappa'''$ .prob-kernel-axioms  $\kappa'''$ -disi,simplified*  
 *$\nu x$ -eq]],standard) (auto simp: space-marginal-measure  $\kappa''$ -def  $\kappa$ -def)  
**qed**  
**qed**(*simp-all add: disi  $\kappa$ .prob-kernel-axioms*)  
**qed**  
**qed**  
**end***

## 2.7 Lemma 14.D.12.

**lemma** *ex-finite-density-measure:*

**fixes**  $A :: nat \Rightarrow -$   
**assumes**  $A: range A \subseteq sets M \cup (range A) = space M \wedge i. emeasure M (A i)$   
 $\neq \infty$  *disjoint-family A*  
**defines**  $h \equiv (\lambda x. (\sum n. (1/2) \wedge (Suc n) * (1 / (1 + M (A n))) * indicator (A n) x))$   
**shows**  $h \in borel-measurable M$   
 $\wedge x. x \in space M \implies 0 < h x$   
 $\wedge x. x \in space M \implies h x < 1$   
*finite-measure (density M h)*

**proof** –

**have** *less1:  $0 < 1 / (1 + M (A n)) \ 1 / (1 + M (A n)) \leq 1$  for  $n$*   
**using**  $A(\mathcal{B})[of n]$  *ennreal-zero-less-divide[of 1 1 + M (A n)]*  
**by** (*auto intro!: divide-le-posI-ennreal simp: add-pos-nonneg*)  
**show** [*measurable*]:  $h \in borel-measurable M$   
**using**  $A$  **by**(*simp add: h-def*)  
{  
**fix**  $x$   
**assume**  $x: x \in space M$   
**then obtain**  $i$  **where**  $i: x \in A i$   
**using**  $A(2)$  **by** *auto*  
**show**  $0 < h x$   
**using**  $A(\mathcal{B})[of i]$  *less1[of i]*  
**by**(*auto simp: h-def suminf-pos-iff i ennreal-divide-times ennreal-zero-less-divide*  
*power-divide-distrib-ennreal power-less-top-ennreal intro!: exI[where  $x=i$ ]*)  
**have**  $h x = (\sum n. (1/2) \wedge (Suc n + 2) * (1 / (1 + M (A (n + 2)))) * indicator$   
 $(A (n + 2)) x) + (1/2) * (1 / (1 + M (A 0))) * indicator (A 0) x + (1/2) \wedge 2$   
 $* (1 / (1 + M (A 1))) * indicator (A 1) x$

```

    by(auto simp: h-def suminf-split-head suminf-offset[of λn. (1/2)^(Suc n) * (1
/ (1 + M (A n))) * indicator (A n) x 2] simp del: power-Suc sum-mult-indicator)
(auto simp: numeral-2-eq-2)
    also have ... ≤ 1/4 + (1/2) * (1 / (1 + M (A 0))) * indicator (A 0) x +
(1/2)^2 * (1 / (1 + M (A 1))) * indicator (A 1) x
    proof -
      have (∑ n. (1/2)^(Suc n + 2) * (1 / (1 + M (A (n + 2))))) * indicator (A
(n + 2)) x ≤ (∑ n. (1/2)^(Suc n + 2))
      using less1(2)[of Suc (Suc -)] by(intro suminf-le, auto simp: indicator-def)
(metis mult.right-neutral mult-left-mono zero-le)
      also have ... = (∑ n. ennreal ((1 / 2) ^ (Suc n + 2)))
      by(simp only: ennreal-power[of 1/2, symmetric]) (metis divide-ennreal en-
nreal-1 ennreal-numeral linorder-not-le not-one-less-zero zero-less-numeral)
      also have ... = ennreal (∑ n. (1 / 2) ^ (Suc n + 2))
      by(rule suminf-ennreal2) auto
      also have ... = ennreal (1/4)
      using nsum-of-r'[of 1/2 Suc (Suc (Suc 0)) 1] by auto
      also have ... = 1 / 4
      by (metis ennreal-divide-numeral ennreal-numeral numeral-One zero-less-one-class.zero-le-one)
      finally show ?thesis by simp
    qed
  also have ... < 1 (is ?lhs < -)
  proof(cases x ∈ A 0)
    case True
      then have x ∉ A 1
      using A(4) by (auto simp: disjoint-family-on-def)
      hence ?lhs = 1 / 4 + 1 / 2 * (1 / (1 + emeasure M (A 0)))
      by(simp add: True)
      also have ... ≤ 1 / 4 + 1 / 2
      using less1(2)[of 0] by (simp add: divide-right-mono-ennreal ennreal-divide-times)
      also have ... = 1 / 4 + 2 / 4
      using divide-mult-eq[of 2 1 2] by simp
      also have ... = 3 / 4
      by(simp add: add-divide-distrib-ennreal[symmetric])
      also have ... < 1
      by(simp add: divide-less-ennreal)
      finally show ?thesis .
    case False
      then have ?lhs = 1 / 4 + (1 / 2)^2 * (1 / (1 + emeasure M (A 1))) *
indicator (A 1) x
      by simp
      also have ... ≤ 1 / 4 + (1 / 2)^2
      by (metis less1(2)[of 1] add-left-mono indicator-eq-0-iff indicator-eq-1-iff
mult.right-neutral mult-eq-0-iff mult-left-mono zero-le)
      also have ... = 2 / 4
      by(simp add: power-divide-distrib-ennreal add-divide-distrib-ennreal[symmetric])
      also have ... < 1
      by(simp add: divide-less-ennreal)
  end

```

```

    finally show ?thesis .
  qed
  finally show  $h\ x < 1$  .
}
show finite-measure (density M h)
proof
  show emeasure (density M h) (space (density M h))  $\neq \infty$ 
  proof -
    have integralN M h  $\neq \top$  (is ?lhs  $\neq -$ )
    proof -
      have ?lhs =  $(\sum n. (\int^+ x \in A\ n. ((1/2) \wedge (Suc\ n)) * (1 / (1 + M (A\ n))))$ 
 $\partial M)$ 
      using A by (simp add: h-def nn-integral-suminf)
      also have ... =  $(\sum n. (1/2) \wedge (Suc\ n)) * (1 / (1 + M (A\ n))) * M (A\ n)$ 
      by (rule suminf-cong, rule nn-integral-cmult-indicator) (use A in auto)
      also have ... =  $(\sum n. (1/2) \wedge (Suc\ n)) * ((1 / (1 + M (A\ n))) * M (A\ n))$ 
      by (simp add: mult.assoc)
      also have ...  $\leq (\sum n. (1/2) \wedge (Suc\ n))$ 
      proof -
        have  $(1 / (1 + M (A\ n))) * M (A\ n) \leq 1$  for n
        using A(3)[of n] by (simp add: add-pos-nonneg divide-le-posI-ennreal
ennreal-divide-times)
        thus ?thesis
        by (intro suminf-le) (metis mult.right-neutral mult-left-mono zero-le, auto)
      qed
      also have ... =  $(\sum n. ennreal ((1/2) \wedge (Suc\ n)))$ 
      by (simp only: ennreal-power[of 1/2, symmetric]) (metis divide-ennreal
ennreal-1 ennreal-numeral linorder-not-le not-one-less-zero zero-less-numeral)
      also have ... =  $ennreal (\sum n. (1/2) \wedge (Suc\ n))$ 
      by (rule suminf-ennreal2) auto
      also have ... = 1
      using nsum-of-r'[of 1/2 1 1] by auto
      finally show ?thesis
      using nle-le by fastforce
    qed
  qed
  thus ?thesis
  by (simp add: emeasure-density)
qed
qed
qed

```

```

lemma (in sigma-finite-measure) finite-density-measure:
  obtains h where h  $\in$  borel-measurable M
     $\bigwedge x. x \in \text{space } M \implies 0 < h\ x$ 
     $\bigwedge x. x \in \text{space } M \implies h\ x < 1$ 
    finite-measure (density M h)
  by (metis (no-types, lifting) sigma-finite-disjoint ex-finite-density-measure)

```

## 2.8 Lemma 14.D.13.

**lemma** (in *measure-kernel*)  
**assumes** *disintegration*  $\nu$   $\mu$   
**defines**  $\nu x \equiv$  *marginal-measure*  $X$   $Y$   $\nu$   
**shows** *disintegration-absolutely-continuous*: *absolutely-continuous*  $\mu$   $\nu x$   
**and** *disintegration-density*:  $\nu x =$  *density*  $\mu$  ( $\lambda x. \kappa x$  (*space*  $Y$ ))  
**and** *disintegration-absolutely-continuous-iff*:  
*absolutely-continuous*  $\nu x$   $\mu \longleftrightarrow$  (*AE*  $x$  in  $\mu. \kappa x$  (*space*  $Y$ )  $> 0$ )  
**proof** –  
**note** *sets-eq*[*measurable-cong*] = *disintegration-sets-eq*[*OF assms*(1)]  
**note** [*measurable*] = *emeasure-measurable*[*OF sets.top*]  
**have**  $\nu x$ -eq:  $\nu x$   $A =$  ( $\int^{+x \in A. (\kappa x$  (*space*  $Y$ ))  $\partial \mu$ ) **if**  $A: A \in$  *sets*  $X$  **for**  $A$   
**by**(*simp add: disintegrationD*[*OF assms*(1)  $A$  *sets.top*] *emeasure-marginal-measure*[*OF sets-eq*(1)  $A$ ]  $\nu x$ -def)  
**thus**  $1: \nu x =$  *density*  $\mu$  ( $\lambda x. \kappa x$  (*space*  $Y$ ))  
**by**(*auto intro!: measure-eqI simp: sets-marginal-measure  $\nu x$ -def sets-eq emeasure-density*)  
**hence** *sets- $\nu x$ :sets*  $\nu x =$  *sets*  $X$   
**using** *sets-eq* **by** *simp*  
**show** *absolutely-continuous*  $\mu$   $\nu x$   
**unfolding** *absolutely-continuous-def*  
**proof** *safe*  
**fix**  $A$   
**assume**  $A: A \in$  *null-sets*  $\mu$   
**have**  $0 =$  ( $\int^{+x \in A. (\kappa x$  (*space*  $Y$ ))  $\partial \mu$ )  
**by**(*simp add: A nn-integral-null-set*)  
**also have**  $\dots = \nu x$   $A$   
**using**  $A$   $\nu x$ -eq[*of A, simplified sets-eq*(2)] [*symmetric*]  
**by** *auto*  
**finally show**  $A \in$  *null-sets*  $\nu x$   
**using**  $A$  **by**(*auto simp: null-sets-def  $\nu x$ -def sets-marginal-measure sets-eq*)  
**qed**  
**show** *absolutely-continuous*  $\nu x$   $\mu \longleftrightarrow$  (*AE*  $x$  in  $\mu. \kappa x$  (*space*  $Y$ )  $> 0$ )  
**proof**  
**assume**  $h: \text{absolutely-continuous } \nu x \mu$   
**define**  $N$  **where**  $N = \{x \in \text{space } \mu. (\kappa x) (\text{space } Y) = 0\}$   
**have**  $N \in$  *null-sets*  $\mu$   
**proof** –  
**have**  $\nu x$   $N =$  ( $\int^{+x \in N. (\kappa x$  (*space*  $Y$ ))  $\partial \mu$ )  
**using**  $\nu x$ -eq[*of N*] **by**(*simp add: N-def sets-eq-imp-space-eq*[*OF sets-eq*(2)])  
**also have**  $\dots =$  ( $\int^{+x \in N. 0 \partial \mu$ )  
**by**(*rule nn-integral-cong*) (*auto simp: N-def indicator-def*)  
**also have**  $\dots = 0$  **by** *simp*  
**finally have**  $N \in$  *null-sets*  $\nu x$   
**by**(*auto simp: null-sets-def 1 N-def*)  
**thus** *?thesis*  
**using**  $h$  **by**(*auto simp: absolutely-continuous-def*)  
**qed**  
**then show** *AE*  $x$  in  $\mu. 0 < (\kappa x)$  (*space*  $Y$ )

```

    by(auto intro!: AE-I'[OF - subset-refl] simp: N-def)
  next
    assume AE x in  $\mu$ .  $0 < (\kappa x)$  (space Y)
    then show absolutely-continuous  $\nu x \mu$ 
      using  $\nu x$ -eq by(auto simp: absolutely-continuous-def intro!: null-if-pos-func-has-zero-nn-int[where
 $f=\lambda x. \text{emeasure } (\kappa x)$  (space Y)]) (auto simp: null-sets-def sets- $\nu x$ )
    qed
  qed

```

## 2.9 Theorem 14.D.14.

**locale** *sigma-finite-measure-on-pair-standard* = *sigma-finite-measure-on-pair* + *standard-borel-ne Y*

**sublocale** *projection-sigma-finite-standard*  $\subseteq$  *sigma-finite-measure-on-pair-standard*  
**by** (*simp add: sigma-finite-measure-on-pair-axioms sigma-finite-measure-on-pair-standard-def standard-borel-ne-axioms*)

**context** *sigma-finite-measure-on-pair-standard*  
**begin**

**lemma** *measure-disintegration-extension*:

$\exists \mu \kappa. \text{finite-measure } \mu \wedge \text{measure-kernel } X Y \kappa \wedge \text{measure-kernel.disintegration } X Y \kappa \nu \mu \wedge$   
 $(\forall x \in \text{space } X. \text{sigma-finite-measure } (\kappa x)) \wedge$   
 $(\forall x \in \text{space } X. \kappa x (\text{space } Y) > 0) \wedge$   
 $\mu \sim_M \nu x$  (**is** ?goal)

**proof**(*rule sigma-finite-measure.sigma-finite-disjoint[OF sigma-finite]*)

**fix**  $A :: \text{nat} \Rightarrow -$   
**assume**  $A:\text{range } A \subseteq \text{sets } \nu \cup (\text{range } A) = \text{space } \nu \wedge i. \text{emeasure } \nu (A i) \neq \infty$   
*disjoint-family A*  
**define**  $h$  **where**  $h \equiv (\lambda x. \sum n. (1 / 2) \wedge \text{Suc } n * (1 / (1 + \text{emeasure } \nu (A n))))$   
 $* \text{indicator } (A n) x$   
**have**  $h: h \in \text{borel-measurable } \nu \wedge x y. x \in \text{space } X \implies y \in \text{space } Y \implies 0 < h$   
 $(x,y) \wedge x y. x \in \text{space } X \implies y \in \text{space } Y \implies h(x,y) < 1$  *finite-measure (density*  
 $\nu h)$

**using** *ex-finite-density-measure[OF A]* **by**(*auto simp: sets-eq-imp-space-eq[OF nu-sets] h-def space-pair-measure*)

**interpret** *psfs- $\nu x$ : finite-measure marginal-measure X Y (density  $\nu h$ )*  
**by**(*rule finite-measure-marginal-measure-finite[OF h(4),simplified,OF nu-sets]*)

**interpret** *psfs: projection-sigma-finite-standard X Y density  $\nu h$*   
**by**(*auto simp: projection-sigma-finite-standard-def projection-sigma-finite-def standard-borel-ne-axioms nu-sets finite-measure.sigma-finite-measure[OF finite-measure-marginal-measure-finite h(4),simplified,OF nu-sets]*)  
**from** *psfs.measure-disintegration*  
**obtain**  $\kappa'$  **where**  $\kappa': \text{prob-kernel } X Y \kappa' \text{measure-kernel.disintegration } X Y \kappa'$   
*(density  $\nu h$ ) psfs. $\nu x$  by auto*



```

interpret pk: prob-kernel X Y κ' by fact
define κ where κ ≡ (λx. density (κ' x) (λy. 1 / h (x,y)))
have κB: κ x B = (∫+ y∈B. (1 / h (x, y)) ∂κ' x) if x ∈ space X and [measurable]: B
∈ sets Y for x B
  using nu-sets pk.kernel-sets[OF that(1)] that h(1) by(auto simp: κ-def emea-
sure-density)
interpret mk: measure-kernel X Y κ
proof
  fix B
  assume [measurable]: B ∈ sets Y
  have 1: (λx. ∫+ y∈B. (1 / h (x, y)) ∂κ' x) ∈ borel-measurable X
  using h(1) nu-sets by(auto intro!: pk.nn-integral-measurable-f'[of λz. (1 / h
z) * indicator B (snd z),simplified])
  show (λx. (κ x) B) ∈ borel-measurable X
  by(rule measurable-cong[THEN iffD1,OF - 1],simp add: κB)
qed(simp-all add: κ-def pk.kernel-sets space-ne)

have disi: mk.disintegration ν psfs.νx
proof(rule mk.disintegrationI)
  fix A B
  assume A[measurable]: A ∈ sets X and B[measurable]: B ∈ sets Y
  show ν (A × B) = (∫+ x∈A. (κ x) B ∂psfs.νx) (is ?lhs = ?rhs)
  proof -
    have ?lhs = (∫+ z∈A × B. 1 ∂ν)
    by auto
    also have ... = (∫+ z∈A × B. (1 / h z * h z) ∂ν)
    proof -
      have 1: a * (1 / a) = 1 if 0 < a a < 1 for a :: ennreal
      proof -
        have a * (1 / a) = ennreal (enn2real a * 1 / (enn2real a))
        by (simp add: divide-eq-1-ennreal enn2real-eq-0-iff ennreal-times-divide)
        also have ... = ennreal 1
        using enn2real-eq-0-iff that by fastforce
        finally show ?thesis
        using ennreal-1 by simp
      qed
    show ?thesis
    by(rule nn-integral-cong,auto simp add: sets-eq-imp-space-eq[OF nu-sets]
space-pair-measure ennreal-divide-times indicator-def 1[OF h(2,3)])
  qed
  also have ... = (∫+ z. h z * ((1 / h z) * indicator (A × B) z) ∂ν)
  by(auto intro!: nn-integral-cong simp: indicator-def mult commute)
  also have ... = (∫+ z∈A × B. (1 / h z) ∂(density ν h))
  using h(1) by(simp add: nn-integral-density)
  also have ... = (∫+ x. ∫+ y. (1 / h (x,y) * indicator (A × B) (x,y)) ∂κ' x
∂psfs.νx)
  using h(1) by(simp add: pk.nn-integral-fst-finite'[OF - κ'(2) psfs-νx.finite-measure-axioms])
  also have ... = (∫+ x∈A. (∫+ y∈B. (1 / h (x,y)) ∂κ' x) ∂psfs.νx)
  by(auto intro!: nn-integral-cong simp: indicator-def)

```

```

    also have ... = ?rhs
      by(auto intro!: nn-integral-cong simp:  $\kappa B[OF - B]$  space-marginal-measure)
    finally show ?thesis .
  qed
qed(simp-all add: nu-sets sets-marginal-measure)
have geq0:  $0 < (\kappa x)$  (space Y) if  $x \in$  space X for x
proof -
  have  $0 = (\int^+ y. 0 \partial\kappa' x)$  by simp
  also have  $\dots < (\int^+ y. (1 / h(x,y)) \partial\kappa' x)$ 
  proof(rule nn-integral-less)
    show  $\neg (AE y \text{ in } \kappa' x. 1 / h(x,y) \leq 0)$ 
    proof
      assume  $AE y \text{ in } \kappa' x. 1 / h(x,y) \leq 0$ 
      moreover have  $h(x,y) \neq \top$  if  $y \in$  space  $(\kappa' x)$  for y
      using  $h(\beta)[OF \langle x \in \text{space } X \rangle \text{ that}[simplified \text{ sets-eq-imp-space-eq}[OF$ 
 $pk.kernel-sets[OF \langle x \in \text{space } X \rangle]]]] \text{ top.not-eq-extremum}$ 
      by fastforce
      ultimately show False
      using  $prob-space.AE-False[OF pk.prob-spaces[OF that]]$  by simp
    qed
  qed(use  $h(1)$   $pk.kernel-sets[OF that]$  that in auto)
  also have  $\dots = (\kappa x)$  (space Y)
  by(simp add:  $\kappa B[OF that \text{ sets.top}]$ ) (simp add:  $\text{sets-eq-imp-space-eq}[OF$ 
 $pk.kernel-sets[OF that], \text{symmetric}]$ )
  finally show ?thesis .
qed

show ?goal
proof(safe intro!:  $exI[\text{where } x=psfs.\nu x]$   $exI[\text{where } x=\kappa]$  disi)
  show absolutely-continuous  $\nu x$   $psfs.\nu x$ 
    unfolding  $mk.\text{disintegration-absolutely-continuous-iff}[OF \text{disi}]$ 
    by standard (simp add: space-marginal-measure geq0)
next
fix x
assume  $x: x \in$  space X
define C where  $C \equiv \text{range } (\lambda n. \text{Pair } x -' (A n) \cap \text{space } Y)$ 
have 1: countable C  $C \subseteq$  sets Y
  using  $A(1,2)$  x by (auto simp: nu-sets sets-eq-imp-space-eq[OF nu-sets]
space-pair-measure C-def)
have 2:  $\bigcup C =$  space Y
  using  $A(1,2)$  by (auto simp: sets-eq-imp-space-eq[OF nu-sets] space-pair-measure
C-def) (use x in auto)

show sigma-finite-measure  $(\kappa x)$ 
  unfolding sigma-finite-measure-def
proof(safe intro!:  $exI[\text{where } x=C]$ )
  fix c
  assume  $c \in C$   $(\kappa x)$   $c = \infty$ 
  then obtain n where  $c: c = \text{Pair } x -' (A n) \cap \text{space } Y$  by (auto simp: C-def)

```

```

have ( $\kappa$   $x$ )  $c = (\int^+ y \in c. (1 / h(x, y)) \partial \kappa' x)$ 
  using  $\kappa B[OF\ x, of\ c] 1 \langle c \in C \rangle$  by auto
also have ... =  $(\int^+ y \in Pair\ x - ' (A\ n). (1 / h(x, y)) \partial \kappa' x)$ 
  by (auto intro!: nn-integral-cong simp: c indicator-def sets-eq-imp-space-eq[OF pk.kernel-sets[OF x]])
also have ... =  $(\int^+ y \in Pair\ x - ' (A\ n). (1 / ((1 / 2) \wedge Suc\ n * (1 / (1 + emeasure\ \nu\ (A\ n)))) \partial \kappa' x)$ 
proof -
  {
    fix  $y$ 
    assume  $xy:(x, y) \in A\ n$ 
    have  $1 / h(x, y) = 1 / ((1 / 2) \wedge Suc\ n * (1 / (1 + emeasure\ \nu\ (A\ n))))$ 
    proof -
      have  $h(x, y) = (1 / 2) \wedge Suc\ n * (1 / (1 + emeasure\ \nu\ (A\ n)))$  (is
         $?lhs = ?rhs$ )
      proof -
        have  $?lhs = (\sum m. (1 / 2) \wedge Suc\ m * (1 / (1 + emeasure\ \nu\ (A\ m))))$ 
           $* indicator\ (A\ m)\ (x, y)$ 
        by (simp add: h-def)
        also have ... =  $(\sum m. if\ m = n\ then\ (1 / 2) \wedge Suc\ n * (1 / (1 + emeasure\ \nu\ (A\ n)))$ 
           $else\ 0)$ 
        using  $xy\ A(4)$  by (fastforce intro!: suminf-cong simp: disjoint-family-on-def indicator-def)
        also have ... =  $(\sum j. if\ j + Suc\ n = n\ then\ (1 / 2) \wedge Suc\ n * (1 / (1 + emeasure\ \nu\ (A\ n)))$ 
           $else\ 0) + (\sum j < Suc\ n. if\ j = n\ then\ (1 / 2) \wedge Suc\ n * (1 / (1 + emeasure\ \nu\ (A\ n)))$ 
           $else\ 0)$ 
        by (auto simp: suminf-offset[of  $\lambda m. if\ m = n\ then\ (1 / 2) \wedge Suc\ n * (1 / (1 + emeasure\ \nu\ (A\ n)))$ 
           $else\ 0\ Suc\ n]$  simp del: power-Suc)
        also have ... =  $?rhs$ 
        by simp
        finally show  $?thesis$  .
      qed
      thus  $?thesis$  by simp
    qed
  }
thus  $?thesis$ 
by (intro nn-integral-cong) (auto simp: sets-eq-imp-space-eq[OF pk.kernel-sets[OF x]] indicator-def simp del: power-Suc)
qed
also have ...  $\leq (\int^+ y. (1 / ((1 / 2) \wedge Suc\ n * (1 / (1 + emeasure\ \nu\ (A\ n)))) \partial \kappa' x)$ 
by (rule nn-integral-mono) (auto simp: indicator-def)
also have ... =  $(1 / ((1 / 2) \wedge Suc\ n * (1 / (1 + emeasure\ \nu\ (A\ n))))$ 
by (simp add: prob-space.emeasure-space-1[OF pk.prob-spaces[OF x]])
also have ...  $< \infty$ 
by (metis A(3) ennreal-add-eq-top ennreal-divide-eq-0-iff ennreal-divide-eq-top-iff ennreal-top-neq-one infinity-ennreal-def mult-eq-0-iff power-eq-0-iff top.not-eq-extremum top-neq-numeral)
finally show False

```

```

    using ⟨(κ x) c = ∞⟩ by simp
  qed(insert 1 2, auto simp: mk.kernel-sets[OF x] sets-eq-imp-space-eq[OF mk.kernel-sets[OF
x]])
  qed(auto simp: psfs-νx.finite-measure-axioms geq0 mk.measure-kernel-axioms mk.disintegration-absolutely-con
disi])
qed

end

lemma(in sigma-finite-measure-on-pair) measure-disintegration-extension-AE-unique:
  assumes sigma-finite-measure μ sigma-finite-measure μ'
    measure-kernel X Y κ measure-kernel X Y κ'
    measure-kernel.disintegration X Y κ ν μ measure-kernel.disintegration X
Y κ' ν μ'
  and absolutely-continuous μ μ' B ∈ sets Y
  shows AE x in μ. κ' x B * RN-deriv μ μ' x = κ x B
proof -
  interpret s1: sigma-finite-measure μ by fact
  interpret s2: sigma-finite-measure μ' by fact
  interpret mk1: measure-kernel X Y κ by fact
  interpret mk2: measure-kernel X Y κ' by fact
  have sets[measurable-cong]:sets μ = sets X sets μ' = sets X
  using assms(5,6) by(auto dest: mk1.disintegration-sets-eq mk2.disintegration-sets-eq)
  have 1:AE x in μ. κ x B = RN-deriv μ (marginal-measure-on X Y ν B) x
  using sets mk1.emmeasure-measurable[OF assms(8)] mk1.disintegrationD[OF
assms(5) - assms(8)]
  by(auto intro!: measure-eqI s1.RN-deriv-unique simp: emeasure-density emea-
sure-marginal-measure-on[OF nu-sets assms(8)] sets sets-marginal-measure)
  have 2:AE x in μ. κ' x B * RN-deriv μ μ' x = RN-deriv μ (marginal-measure-on
X Y ν B) x
  proof -
    {
      fix A
      assume A: A ∈ sets X
      have (∫+x∈A. ((κ' x) B * RN-deriv μ μ' x) ∂μ) = (∫+x. RN-deriv μ μ' x
* (κ' x B * indicator A x)∂μ)
      by(auto intro!: nn-integral-cong simp: indicator-def mult.commute)
      also have ... = (∫+x∈A. κ' x B ∂μ')
      using mk2.emmeasure-measurable[OF assms(8)] sets A
      by(auto intro!: s1.RN-deriv-nn-integral[OF assms(7),symmetric])
      also have ... = ν (A × B)
      by(simp add: mk2.disintegrationD[OF assms(6) A assms(8)])
      finally have (∫+x∈A. ((κ' x) B * RN-deriv μ μ' x) ∂μ) = ν (A × B) .
    }
  thus ?thesis
  using sets mk2.emmeasure-measurable[OF assms(8)]
  by(auto intro!: measure-eqI s1.RN-deriv-unique simp: emeasure-density emea-
sure-marginal-measure-on[OF nu-sets assms(8)]sets sets-marginal-measure)

```

```
qed
show ?thesis
  using 1 2 by auto
qed
end
```

## References

- [1] F. Baccelli, B. Blaszczyzyn, and M. Karray. *Random Measures, Point Processes, and Stochastic Geometry*. Inria, Jan. 2020.