On the Formalization of Martingales

Ata Keskin

March 29, 2024

Abstract

In the scope of this project, we present a formalization of martingales in arbitrary Banach spaces using Isabelle/HOL.

The current formalization of conditional expectation in the Isabelle library is limited to real-valued functions. To overcome this limitation, we extend the construction of conditional expectation to general Banach spaces, employing an approach similar to the one described in [2]. We use measure theoretic arguments to construct the conditional expectation using suitable limits of simple functions.

Subsequently, we define stochastic processes and introduce the concepts of adapted, progressively measurable and predictable processes using suitable locale definitions[1]. We show the relation

$$\text{adapted} \supseteq \text{progressive} \supseteq \text{predictable}$$

Furthermore, we show that progressive measurability and adaptedness are equivalent when the indexing set is discrete. We pay special attention to predictable processes in discrete-time, showing that $(X_n)_{n\in\mathbb{N}}$ is predictable if and only if $(X_{n+1})_{n\in\mathbb{N}}$ is adapted.

Moving forward, we rigorously define martingales, submartingales, and supermartingales, presenting their first consequences and corollaries[2]. Discrete-time martingales are given special attention in the formalization. In every step of our formalization, we make extensive use of the powerful locale system of Isabelle.

The formalization further contributes by generalizing concepts in Bochner integration by extending their application from the real numbers to arbitrary Banach spaces equipped with a second-countable topology. Induction schemes for integrable simple functions on Banach spaces are introduced, accommodating various scenarios with or without a real vector ordering[3]. Specifically, we formalize a powerful result called the "Averaging Theorem"[4] which allows us to show that densities are unique in Banach spaces.

In-depth information on the formalization and the proofs of the individual theorems can be found in [3].

[1] `Martingale.Stochastic_Process`
[2] `Martingale.Martingale`
[3] `Martingale.Bochner_Integration_Addendum`

Contents

theory *Measure-Space-Supplement*
  imports *HOL−Analysis.Measure-Space*
begin

1 Supplementary Lemmas for Measure Spaces

1.1 σ-Algebra Generated by a Family of Functions

definition *family-vimage-algebra* :: $'a$ *set* $\Rightarrow$ $('a \Rightarrow 'b)$ *set* $\Rightarrow$ $'b$ *measure* $\Rightarrow$ $'a$ *measure* **where**
  family-vimage-algebra Ω S M $\equiv$ *sigma* Ω $(\bigcup f \in S.\ \{f -` A \cap \Omega \mid A.\ A \in M\})$

For singleton S, i.e. $S = \{f\}$ for some f, the definition simplifies to that of *vimage-algebra*.

lemma *family-vimage-algebra-singleton*: *family-vimage-algebra* Ω $\{f\}$ M = *vimage-algebra* Ω f M ⟨*proof*⟩

lemma
 shows *sets-family-vimage-algebra*: *sets* (*family-vimage-algebra* Ω S M) = *sigma-sets* Ω $(\bigcup f \in S.\ \{f -` A \cap \Omega \mid A.\ A \in M\})$
  and *space-family-vimage-algebra*[*simp*]: *space* (*family-vimage-algebra* Ω S M) = Ω
  ⟨*proof*⟩

lemma *measurable-family-vimage-algebra*:
  assumes $f \in S$ $f \in \Omega \rightarrow$ *space* M
  shows $f \in$ *family-vimage-algebra* Ω S M $\rightarrow_M$ M
  ⟨*proof*⟩

lemma *measurable-family-vimage-algebra-singleton*:
  assumes $f \in \Omega \rightarrow$ *space* M
  shows $f \in$ *family-vimage-algebra* Ω $\{f\}$ M $\rightarrow_M$ M
  ⟨*proof*⟩

A collection of functions are measurable with respect to some σ-algebra N, if and only if the σ-algebra they generate is contained in N.

lemma *measurable-family-iff-sets*:
  shows $(S \subseteq N \rightarrow_M M) \longleftrightarrow S \subseteq$ *space* $N \rightarrow$ *space* M $\wedge$ *family-vimage-algebra* (*space* N) S $M \subseteq N$
⟨*proof*⟩

lemma *family-vimage-algebra-diff*:
  shows *family-vimage-algebra* Ω S M = *sigma* Ω (*sets* (*family-vimage-algebra* Ω $(S - I)$ M) $\cup$ *family-vimage-algebra* Ω $(S \cap I)$ M)
  ⟨*proof*⟩

end
theory *Elementary-Metric-Spaces-Supplement*
  imports *HOL−Analysis.Elementary-Metric-Spaces*
begin

2 Supplementary Lemmas for Elementary Metric Spaces

2.1 Diameter Lemma

lemma *diameter-comp-strict-mono*:
  fixes $s :: nat \Rightarrow 'a :: metric\text{-}space$
  assumes $strict\text{-}mono\ r\ bounded\ \{s\ i\ |i.\ r\ n \leq i\}$
  shows $diameter\ \{s\ (r\ i)\ |\ i.\ n \leq i\} \leq diameter\ \{s\ i\ |\ i.\ r\ n \leq i\}$
⟨*proof*⟩

lemma *diameter-bounded-bound′*:
  fixes $S :: 'a :: metric\text{-}space\ set$
  assumes S: $bdd\text{-}above\ (case\text{-}prod\ dist\ `\ (S{\times}S))\ x \in S\ y \in S$
  shows $dist\ x\ y \leq diameter\ S$
⟨*proof*⟩

lemma *bounded-imp-dist-bounded*:
  assumes $bounded\ (range\ s)$
  shows $bounded\ ((\lambda(i,\ j).\ dist\ (s\ i)\ (s\ j))\ `\ (\{n..\} \times \{n..\}))$
  ⟨*proof*⟩

A sequence is Cauchy, if and only if it is bounded and it's diameter tends to zero. The diameter is well-defined only if the sequence is bounded.

lemma *cauchy-iff-diameter-tends-to-zero-and-bounded*:
  fixes $s :: nat \Rightarrow 'a :: metric\text{-}space$
  shows $Cauchy\ s \longleftrightarrow ((\lambda n.\ diameter\ \{s\ i\ |\ i.\ i \geq n\}) \longrightarrow 0 \wedge bounded\ (range\ s))$
⟨*proof*⟩

end

theory *Bochner-Integration-Supplement*
  imports *HOL−Analysis.Bochner-Integration HOL−Analysis.Set-Integral Elementary-Metric-Spaces-Supplement*
begin

3 Supplementary Lemmas for Bochner Integration

3.1 Integrable Simple Functions

We restate some basic results concerning Bochner-integrable functions.

lemma *integrable-implies-simple-function-sequence*:
  fixes f :: $'a \Rightarrow 'b$::{*banach*, *second-countable-topology*}
  assumes *integrable M f*
  obtains s **where** $\bigwedge i.$ *simple-function M* $(s\ i)$
    and $\bigwedge i.$ *emeasure M* $\{y \in$ *space M*. $s\ i\ y \neq 0\} \neq \infty$
    and $\bigwedge x.\ x \in$ *space M* $\Longrightarrow (\lambda i.\ s\ i\ x) \longrightarrow f\ x$
    and $\bigwedge x\ i.\ x \in$ *space M* $\Longrightarrow$ *norm* $(s\ i\ x) \leq 2 *$ *norm* $(f\ x)$
⟨*proof*⟩

Simple functions can be represented by sums of indicator functions.

lemma *simple-function-indicator-representation*:
  fixes f ::$'a \Rightarrow 'b$ :: {*second-countable-topology*, *banach*}
  assumes *simple-function M f* $x \in$ *space M*
  shows $f\ x = (\sum y \in f$ ‘ *space M*. *indicator* $(f -$‘ $\{y\} \cap$ *space M*$)\ x *_R y)$
  (**is** *?l* = *?r*)
⟨*proof*⟩

lemma *simple-function-indicator-representation-AE*:
  fixes f ::$'a \Rightarrow 'b$ :: {*second-countable-topology*, *banach*}
  assumes *simple-function M f*
  shows *AE x in M*. $f\ x = (\sum y \in f$ ‘ *space M*. *indicator* $(f -$‘ $\{y\} \cap$ *space M*$)\ x$
$*_R\ y)$
  ⟨*proof*⟩

lemmas *simple-function-scaleR*[*intro*] = *simple-function-compose2*[**where** h=($*_R$)]
lemmas *integrable-simple-function* = *simple-bochner-integrable.intros*[*THEN has-bochner-integral-simple-bochr*
THEN integrable.intros]

Induction rule for simple integrable functions.

lemma *integrable-simple-function-induct*[*consumes 2*, *case-names cong indicator add*, *induct set*: *simple-function*]:
  fixes f :: $'a \Rightarrow 'b$ :: {*second-countable-topology*, *banach*}
  assumes f: *simple-function M f emeasure M* $\{y \in$ *space M*. $f\ y \neq 0\} \neq \infty$
  assumes *cong*: $\bigwedge f\ g.$ *simple-function M f* $\Longrightarrow$ *emeasure M* $\{y \in$ *space M*. $f\ y \neq$
$0\} \neq \infty$
        $\Longrightarrow$ *simple-function M g* $\Longrightarrow$ *emeasure M* $\{y \in$ *space M*. $g\ y \neq$
$0\} \neq \infty$
        $\Longrightarrow (\bigwedge x.\ x \in$ *space M* $\Longrightarrow f\ x = g\ x) \Longrightarrow P\ f \Longrightarrow P\ g$
  assumes *indicator*: $\bigwedge A\ y.\ A \in$ *sets M* $\Longrightarrow$ *emeasure M A* $< \infty \Longrightarrow P\ (\lambda x.$
indicator A x $*_R\ y)$
  assumes *add*: $\bigwedge f\ g.$ *simple-function M f* $\Longrightarrow$ *emeasure M* $\{y \in$ *space M*. $f\ y \neq$
$0\} \neq \infty \Longrightarrow$
        simple-function M g $\Longrightarrow$ *emeasure M* $\{y \in$ *space M*. $g\ y \neq 0\} \neq$
$\infty \Longrightarrow$
        $(\bigwedge z.\ z \in$ *space M* $\Longrightarrow$ *norm* $(f\ z + g\ z) =$ *norm* $(f\ z) +$ *norm*
$(g\ z)) \Longrightarrow$
        $P\ f \Longrightarrow P\ g \Longrightarrow P\ (\lambda x.\ f\ x + g\ x)$
  shows $P\ f$
⟨*proof*⟩

Induction rule for non-negative simple integrable functions

lemma *integrable-simple-function-induct-nn*[*consumes 3*, *case-names cong indicator add*, *induct set*: *simple-function*]:
fixes f :: $'a \Rightarrow 'b$ :: {*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*}
assumes f: *simple-function M f emeasure M* $\{y \in$ *space M*. $f\ y \neq 0\} \neq \infty$ $\bigwedge x.\ x \in$ *space M* $\longrightarrow f\ x \geq 0$
assumes *cong*: $\bigwedge f\ g$. *simple-function M f* $\Longrightarrow$ *emeasure M* $\{y \in$ *space M*. $f\ y \neq 0\} \neq \infty \Longrightarrow (\bigwedge x.\ x \in$ *space M* $\Longrightarrow f\ x \geq 0) \Longrightarrow$ *simple-function M g* $\Longrightarrow$ *emeasure M* $\{y \in$ *space M*. $g\ y \neq 0\} \neq \infty \Longrightarrow (\bigwedge x.\ x \in$ *space M* $\Longrightarrow g\ x \geq 0) \Longrightarrow (\bigwedge x.\ x \in$ *space M* $\Longrightarrow f\ x = g\ x) \Longrightarrow P\ f \Longrightarrow P\ g$
assumes *indicator*: $\bigwedge A\ y.\ y \geq 0 \Longrightarrow A \in$ *sets M* $\Longrightarrow$ *emeasure M A* $< \infty \Longrightarrow P\ (\lambda x.$ *indicator A x* $*_R\ y)$
assumes *add*: $\bigwedge f\ g.\ (\bigwedge x.\ x \in$ *space M* $\Longrightarrow f\ x \geq 0) \Longrightarrow$ *simple-function M f* $\Longrightarrow$ *emeasure M* $\{y \in$ *space M*. $f\ y \neq 0\} \neq \infty \Longrightarrow$
$(\bigwedge x.\ x \in$ *space M* $\Longrightarrow g\ x \geq 0) \Longrightarrow$ *simple-function M g* $\Longrightarrow$ *emeasure M* $\{y \in$ *space M*. $g\ y \neq 0\} \neq \infty \Longrightarrow$
$(\bigwedge z.\ z \in$ *space M* $\Longrightarrow$ *norm* $(f\ z + g\ z) =$ *norm* $(f\ z)$ + *norm* $(g\ z)) \Longrightarrow$
$P\ f \Longrightarrow P\ g \Longrightarrow P\ (\lambda x.\ f\ x + g\ x)$
shows $P\ f$
⟨*proof*⟩

lemma *finite-nn-integral-imp-ae-finite*:
fixes f :: $'a \Rightarrow$ *ennreal*
assumes $f \in$ *borel-measurable M* $(\int^+ x.\ f\ x\ \partial M) < \infty$
shows *AE x in M*. $f\ x < \infty$
⟨*proof*⟩

Convergence in L1-Norm implies existence of a subsequence which convergences almost everywhere. This lemma is easier to use than the existing one in *HOL−Analysis.Bochner-Integration*

lemma *cauchy-L1-AE-cauchy-subseq*:
fixes s :: *nat* $\Rightarrow 'a \Rightarrow 'b$::{*banach*, *second-countable-topology*}
assumes [*measurable*]: $\bigwedge n$. *integrable M* $(s\ n)$
and $\bigwedge e.\ e > 0 \Longrightarrow \exists N.\ \forall i \geq N.\ \forall j \geq N.$ *LINT x*$|M$. *norm* $(s\ i\ x - s\ j\ x) < e$
obtains r **where** *strict-mono r AE x in M*. *Cauchy* $(\lambda i.\ s\ (r\ i)\ x)$
⟨*proof*⟩

3.2 Totally Ordered Banach Spaces

lemma *integrable-max*[*simp*, *intro*]:
fixes f :: $'a \Rightarrow 'b$ :: {*second-countable-topology*, *banach*, *linorder-topology*}
assumes *fg*[*measurable*]: *integrable M f integrable M g*
shows *integrable M* $(\lambda x.$ *max* $(f\ x)\ (g\ x))$
⟨*proof*⟩

lemma *integrable-min*[*simp*, *intro*]:
fixes f :: $'a \Rightarrow 'b$ :: {*second-countable-topology*, *banach*, *linorder-topology*}

assumes [*measurable*]: *integrable M f integrable M g*
shows *integrable M* ($\lambda x.$ *min* (*f x*) (*g x*))
⟨*proof*⟩

Restatement of *integral-nonneg-AE* for functions taking values in a Banach space.

lemma *integral-nonneg-AE-banach*:
fixes *f* :: *'a* ⇒ *'b* :: {*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*}
assumes [*measurable*]: *f* ∈ *borel-measurable M* **and** *nonneg*: *AE x in M*. $0 \le f\,x$
shows $0 \le$ $integral^L$ *M f*
⟨*proof*⟩

lemma *integral-mono-AE-banach*:
fixes *f g* :: *'a* ⇒ *'b* :: {*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*}
assumes *integrable M f integrable M g AE x in M*. $f\,x \le g\,x$
shows $integral^L$ *M f* ≤ $integral^L$ *M g*
⟨*proof*⟩

lemma *integral-mono-banach*:
fixes *f g* :: *'a* ⇒ *'b* :: {*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*}
assumes *integrable M f integrable M g* ⋀*x*. *x* ∈ *space M* ⟹ $f\,x \le g\,x$
shows $integral^L$ *M f* ≤ $integral^L$ *M g*
⟨*proof*⟩

3.3 Integrability and Measurability of the Diameter

context
fixes *s* :: *nat* ⇒ *'a* ⇒ *'b* :: {*second-countable-topology*, *banach*} **and** *M*
assumes *bounded*: ⋀*x*. *x* ∈ *space M* ⟹ *bounded* (*range* ($\lambda i.$ *s i x*))
begin

lemma *borel-measurable-diameter*:
assumes [*measurable*]: ⋀*i*. (*s i*) ∈ *borel-measurable M*
shows ($\lambda x.$ *diameter* {*s i x* |*i*. $n \le i$}) ∈ *borel-measurable M*
⟨*proof*⟩

lemma *integrable-bound-diameter*:
fixes *f* :: *'a* ⇒ *real*
assumes *integrable M f*
and [*measurable*]: ⋀*i*. (*s i*) ∈ *borel-measurable M*
and ⋀*x i*. *x* ∈ *space M* ⟹ *norm* (*s i x*) ≤ *f x*
shows *integrable M* ($\lambda x.$ *diameter* {*s i x* |*i*. $n \le i$})
⟨*proof*⟩
end

3.4 Auxiliary Lemmas for Set Integrals

lemma *set-integral-scaleR-left*:
  assumes $A \in sets\ M\ c \neq 0 \Longrightarrow integrable\ M\ f$
  shows $LINT\ t{:}A|M.\ f\ t *_R c = (LINT\ t{:}A|M.\ f\ t) *_R c$
  ⟨*proof*⟩

lemma *nn-set-integral-eq-set-integral*:
  assumes [*measurable*]:*integrable M f*
    and $AE\ x \in A\ in\ M.\ 0 \leq f\ x\ A \in sets\ M$
   shows $(\int^+ x{\in}A.\ f\ x\ \partial M) = (\int\ x \in A.\ f\ x\ \partial M)$
⟨*proof*⟩

lemma *set-integral-restrict-space*:
  fixes $f :: 'a \Rightarrow 'b{::}\{banach,\ second\text{-}countable\text{-}topology\}$
  assumes $\Omega \cap space\ M \in sets\ M$
  shows $set\text{-}lebesgue\text{-}integral\ (restrict\text{-}space\ M\ \Omega)\ A\ f = set\text{-}lebesgue\text{-}integral\ M\ A\ (\lambda x.\ indicator\ \Omega\ x *_R f\ x)$
  ⟨*proof*⟩

lemma *set-integral-const*:
  fixes $c :: 'b{::}\{banach,\ second\text{-}countable\text{-}topology\}$
  assumes $A \in sets\ M\ emeasure\ M\ A \neq \infty$
  shows $set\text{-}lebesgue\text{-}integral\ M\ A\ (\lambda\text{-}.\ c) = measure\ M\ A *_R c$
  ⟨*proof*⟩

lemma *set-integral-mono-banach*:
  fixes $f\ g :: 'a \Rightarrow 'b :: \{second\text{-}countable\text{-}topology,\ banach,\ linorder\text{-}topology,\ ordered\text{-}real\text{-}vector\}$
  assumes $set\text{-}integrable\ M\ A\ f\ set\text{-}integrable\ M\ A\ g$
   $\bigwedge x.\ x \in A \Longrightarrow f\ x \leq g\ x$
  shows $(LINT\ x{:}A|M.\ f\ x) \leq (LINT\ x{:}A|M.\ g\ x)$
  ⟨*proof*⟩

lemma *set-integral-mono-AE-banach*:
  fixes $f\ g :: 'a \Rightarrow 'b :: \{second\text{-}countable\text{-}topology,\ banach,\ linorder\text{-}topology,\ ordered\text{-}real\text{-}vector\}$
  assumes $set\text{-}integrable\ M\ A\ f\ set\text{-}integrable\ M\ A\ g\ AE\ x{\in}A\ in\ M.\ f\ x \leq g\ x$
  shows $set\text{-}lebesgue\text{-}integral\ M\ A\ f \leq set\text{-}lebesgue\text{-}integral\ M\ A\ g$ ⟨*proof*⟩

3.5 Averaging Theorem

We aim to lift results from the real case to arbitrary Banach spaces. Our fundamental tool in this regard will be the averaging theorem. The proof of this theorem is due to Serge Lang (Real and Functional Analysis) [4]. The theorem allows us to make statements about a functions value almost everywhere, depending on the value its integral takes on various sets of the measure space.

Before we introduce and prove the averaging theorem, we will first show the following lemma which is crucial for our proof. While not stated exactly in this manner, our proof makes use of the characterization of second-countable topological spaces given in the book General Topology by Ryszard Engelking (Theorem 4.1.15) [1].

lemma *balls-countable-basis*:
  obtains *D* :: *'a* :: {*metric-space*, *second-countable-topology*} *set*
  where *topological-basis* (*case-prod ball* ‘ (*D* × (ℚ ∩ {*0*<..})))
    and *countable D*
    and *D* ≠ {}
⟨*proof*⟩

context *sigma-finite-measure*
begin

To show statements concerning σ-finite measure spaces, one usually shows the statement for finite measure spaces and uses a limiting argument to show it for the σ-finite case. The following induction scheme formalizes this.

lemma *sigma-finite-measure-induct*[*case-names finite-measure*, *consumes 0*]:
  assumes ⋀(*N* :: *'a measure*) Ω. *finite-measure N*
    ⟹ *N* = *restrict-space M* Ω
    ⟹ Ω ∈ *sets M*
    ⟹ *emeasure N* Ω ≠ ∞
    ⟹ *emeasure N* Ω ≠ *0*
    ⟹ *almost-everywhere N Q*
    and [*measurable*]: *Measurable.pred M Q*
  shows *almost-everywhere M Q*
⟨*proof*⟩

The Averaging Theorem allows us to make statements concerning how a function behaves almost everywhere, depending on its behaviour on average.

lemma *averaging-theorem*:
  fixes *f*::- ⇒ *'b*::{*second-countable-topology*, *banach*}
  assumes [*measurable*]:*integrable M f*
    and *closed*: *closed S*
    and ⋀*A*. *A* ∈ *sets M* ⟹ *measure M A* > *0* ⟹ (*1* / *measure M A*) $*_R$ *set-lebesgue-integral M A f* ∈ *S*
  shows *AE x in M*. *f x* ∈ *S*
⟨*proof*⟩

lemma *density-zero*:
  fixes *f*::*'a* ⇒ *'b*::{*second-countable-topology*, *banach*}
  assumes *integrable M f*
    and *density-0*: ⋀*A*. *A* ∈ *sets M* ⟹ *set-lebesgue-integral M A f* = *0*
  shows *AE x in M*. *f x* = *0*
  ⟨*proof*⟩

The following lemma shows that densities are unique in Banach spaces.

lemma *density-unique-banach*:
fixes $f\ f'$::$'a \Rightarrow\ 'b$::{*second-countable-topology*, *banach*}
assumes *integrable M f integrable M f'*
and *density-eq*: $\bigwedge A.\ A \in$ *sets M* $\Longrightarrow$ *set-lebesgue-integral M A f* $=$ *set-lebesgue-integral M A f'*
shows *AE x in M. f x* $=$ *f' x*
⟨*proof*⟩

lemma *density-nonneg*:
fixes f::- $\Rightarrow\ 'b$::{*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*}
assumes *integrable M f*
and $\bigwedge A.\ A \in$ *sets M* $\Longrightarrow$ *set-lebesgue-integral M A f* $\geq$ *0*
shows *AE x in M. f x* $\geq$ *0*
⟨*proof*⟩

corollary *integral-nonneg-eq-0-iff-AE-banach*:
fixes f :: $'a \Rightarrow\ 'b$ :: {*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*}
assumes *f*[*measurable*]: *integrable M f* **and** *nonneg*: *AE x in M. 0* $\leq$ *f x*
shows $integral^L$ *M f* $=$ *0* $\longleftrightarrow$ (*AE x in M. f x* $=$ *0*)
⟨*proof*⟩

corollary *integral-eq-mono-AE-eq-AE*:
fixes $f\ g$ :: $'a \Rightarrow\ 'b$ :: {*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*}
assumes *integrable M f integrable M g* $integral^L$ *M f* $=$ $integral^L$ *M g AE x in M. f x* $\leq$ *g x*
shows *AE x in M. f x* $=$ *g x*
⟨*proof*⟩

end

end

theory *Conditional-Expectation-Banach*
imports *HOL−Probability.Conditional-Expectation HOL−Probability.Independent-Family Bochner-Integration-Supplement*
begin

4 Conditional Expectation in Banach Spaces

While constructing the conditional expectation operator, we have come up with the following approach, which is based on the construction in [2]. Both our approach, and the one in [2] are based on showing that the conditional expectation is a contraction on some dense subspace of the space of functions $L^1(E)$. In our approach, we start by constructing the conditional expectation explicitly for simple functions. Then we show that the conditional expectation is a contraction on simple functions, i.e. $\|E(s|F)(x)\| \leq E(\|s(x)\||F)$ for μ-almost all $x \in \Omega$ with $s : \Omega \to E$ simple and integrable. Using this, we can show that the conditional expectation of a convergent sequence of simple functions is again convergent. Finally, we show that this limit exhibits the properties of a conditional expectation. This approach has the benefit of being straightforward and easy to implement, since we could make use of the existing formalization for real-valued functions. To use the construction in [2] we need more tools from functional analysis, which Isabelle/HOL currently does not have.

Before we can talk about 'the' conditional expectation, we must define what it means for a function to have a conditional expectation.

definition *has-cond-exp* :: *'a measure* ⇒ *'a measure* ⇒ (*'a* ⇒ *'b*) ⇒ (*'a* ⇒ *'b*::{*real-normed-vector*, *second-countable-topology*}) ⇒ *bool* **where**
has-cond-exp M F f g = ((∀ *A* ∈ *sets F*. (∫ *x* ∈ *A*. *f x* ∂*M*) = (∫ *x* ∈ *A*. *g x* ∂*M*))
∧ *integrable M f*
∧ *integrable M g*
∧ *g* ∈ *borel-measurable F*)

This predicate precisely characterizes what it means for a function f to have a conditional expectation g, with respect to the measure M and the sub-σ-algebra F.

lemma *has-cond-expI'*:
assumes ⋀*A*. *A* ∈ *sets F* ⟹ (∫ *x* ∈ *A*. *f x* ∂*M*) = (∫ *x* ∈ *A*. *g x* ∂*M*)
integrable M f
integrable M g
g ∈ *borel-measurable F*
shows *has-cond-exp M F f g*
⟨*proof*⟩

lemma *has-cond-expD*:
assumes *has-cond-exp M F f g*
shows ⋀*A*. *A* ∈ *sets F* ⟹ (∫ *x* ∈ *A*. *f x* ∂*M*) = (∫ *x* ∈ *A*. *g x* ∂*M*)
integrable M f
integrable M g
g ∈ *borel-measurable F*
⟨*proof*⟩

Now we can use Hilberts ϵ-operator to define the conditional expectation, if it exists.

definition *cond-exp* :: *'a measure* ⇒ *'a measure* ⇒ (*'a* ⇒ *'b*) ⇒ (*'a* ⇒ *'b*::{*banach*, *second-countable-topology*}) **where**
cond-exp M F f = (*if* ∃ *g*. *has-cond-exp M F f g then* (*SOME g*. *has-cond-exp M F f g*) *else* (λ-. *0*))

lemma *borel-measurable-cond-exp*[*measurable*]: *cond-exp M F f* ∈ *borel-measurable F*

⟨*proof*⟩

lemma *integrable-cond-exp*[*intro*]: *integrable M* (*cond-exp M F f*)
⟨*proof*⟩

lemma *set-integrable-cond-exp*[*intro*]:
assumes *A* ∈ *sets M*
shows *set-integrable M A* (*cond-exp M F f*) ⟨*proof*⟩

lemma *has-cond-exp-self*:
assumes *integrable M f*
shows *has-cond-exp M* (*vimage-algebra* (*space M*) *f borel*) *f f*
⟨*proof*⟩

lemma *has-cond-exp-sets-cong*:
assumes *sets F* = *sets G*
shows *has-cond-exp M F* = *has-cond-exp M G*
⟨*proof*⟩

lemma *cond-exp-sets-cong*:
assumes *sets F* = *sets G*
shows *AE x in M*. *cond-exp M F f x* = *cond-exp M G f x*
⟨*proof*⟩

context *sigma-finite-subalgebra*
begin

lemma *borel-measurable-cond-exp′*[*measurable*]: *cond-exp M F f* ∈ *borel-measurable M*
⟨*proof*⟩

lemma *cond-exp-null*:
assumes ∄ *g*. *has-cond-exp M F f g*
shows *cond-exp M F f* = (λ-. *0*)
⟨*proof*⟩

We state the tower property of the conditional expectation in terms of the predicate *has-cond-exp*.

lemma *has-cond-exp-nested-subalg*:
fixes *f* :: ′*a* ⇒ ′*b*::{*second-countable-topology*, *banach*}
assumes *subalgebra G F has-cond-exp M F f h has-cond-exp M G f h′*
shows *has-cond-exp M F h′ h*
⟨*proof*⟩

The following lemma shows that the conditional expectation is unique as an element of L1, given that it exists.

lemma *has-cond-exp-charact*:
fixes *f* :: ′*a* ⇒ ′*b*::{*second-countable-topology*, *banach*}
assumes *has-cond-exp M F f g*

shows *has-cond-exp M F f* (*cond-exp M F f*)
AE x in M. cond-exp M F f x = g x
⟨*proof*⟩

corollary *cond-exp-charact*:
fixes *f* :: *'a* ⇒ *'b*::{*second-countable-topology*, *banach*}
assumes $\bigwedge A.\ A \in sets\ F \Longrightarrow (\int x \in A.\ f\ x\ \partial M) = (\int x \in A.\ g\ x\ \partial M)$
integrable M f
integrable M g
g ∈ *borel-measurable F*
shows *AE x in M. cond-exp M F f x = g x*
⟨*proof*⟩

Identity on F-measurable functions:

If an integrable function f is already F-measurable, then *cond-exp M F f* = f μ-a.e. This is a corollary of the lemma on the characterization of *cond-exp*.

corollary *cond-exp-F-meas*[*intro*, *simp*]:
fixes *f* :: *'a* ⇒ *'b*::{*second-countable-topology*, *banach*}
assumes *integrable M f*
f ∈ *borel-measurable F*
shows *AE x in M. cond-exp M F f x = f x*
⟨*proof*⟩

Congruence

lemma *has-cond-exp-cong*:
assumes *integrable M f* $\bigwedge x.\ x \in space\ M \Longrightarrow f\ x = g\ x$ *has-cond-exp M F g h*
shows *has-cond-exp M F f h*
⟨*proof*⟩

lemma *cond-exp-cong*:
fixes *f* :: *'a* ⇒ *'b*::{*second-countable-topology*,*banach*}
assumes *integrable M f integrable M g* $\bigwedge x.\ x \in space\ M \Longrightarrow f\ x = g\ x$
shows *AE x in M. cond-exp M F f x = cond-exp M F g x*
⟨*proof*⟩

lemma *has-cond-exp-cong-AE*:
assumes *integrable M f AE x in M. f x = g x has-cond-exp M F g h*
shows *has-cond-exp M F f h*
⟨*proof*⟩

lemma *has-cond-exp-cong-AE'*:
assumes *h* ∈ *borel-measurable F AE x in M. h x = h' x has-cond-exp M F f h'*
shows *has-cond-exp M F f h*
⟨*proof*⟩

lemma *cond-exp-cong-AE*:
fixes *f* :: *'a* ⇒ *'b*::{*second-countable-topology*,*banach*}
assumes *integrable M f integrable M g AE x in M. f x = g x*

shows *AE x in M. cond-exp M F f x = cond-exp M F g x*
⟨*proof*⟩

The conditional expectation operator on the reals, *real-cond-exp*, satisfies the conditions of the conditional expectation as we have defined it.

lemma *has-cond-exp-real*:
fixes *f* :: *'a ⇒ real*
assumes *integrable M f*
shows *has-cond-exp M F f* (*real-cond-exp M F f*)
⟨*proof*⟩

lemma *cond-exp-real*[*intro*]:
fixes *f* :: *'a ⇒ real*
assumes *integrable M f*
shows *AE x in M. cond-exp M F f x = real-cond-exp M F f x*
⟨*proof*⟩

lemma *cond-exp-cmult*:
fixes *f* :: *'a ⇒ real*
assumes *integrable M f*
shows *AE x in M. cond-exp M F* (*λx. c ∗ f x*) *x = c ∗ cond-exp M F f x*
⟨*proof*⟩

4.1 Existence

Showing the existence is a bit involved. Specifically, what we aim to show is that *has-cond-exp M F f* (*cond-exp M F f*) holds for any Bochner-integrable *f*. We will employ the standard machinery of measure theory. First, we will prove existence for indicator functions. Then we will extend our proof by linearity to simple functions. Finally we use a limiting argument to show that the conditional expectation exists for all Bochner-integrable functions.

Indicator functions

lemma *has-cond-exp-indicator*:
assumes *A ∈ sets M emeasure M A < ∞*
shows *has-cond-exp M F* (*λx. indicat-real A x* $*_R$ *y*) (*λx. real-cond-exp M F* (*indicator A*) *x* $*_R$ *y*)
⟨*proof*⟩

lemma *cond-exp-indicator*[*intro*]:
fixes *y* :: *'b*::{*second-countable-topology*,*banach*}
assumes [*measurable*]: *A ∈ sets M emeasure M A < ∞*
shows *AE x in M. cond-exp M F* (*λx. indicat-real A x* $*_R$ *y*) *x = cond-exp M F* (*indicator A*) *x* $*_R$ *y*
⟨*proof*⟩

Addition

lemma *has-cond-exp-add*:

fixes $f\ g$ :: $'a \Rightarrow 'b$::{*second-countable-topology*,*banach*}
assumes *has-cond-exp M F f f'* *has-cond-exp M F g g'*
shows *has-cond-exp M F* $(\lambda x.\ f\ x + g\ x)$ $(\lambda x.\ f'\ x + g'\ x)$
⟨*proof*⟩

lemma *has-cond-exp-scaleR-right*:
fixes f :: $'a \Rightarrow 'b$::{*second-countable-topology*,*banach*}
assumes *has-cond-exp M F f f'*
shows *has-cond-exp M F* $(\lambda x.\ c *_R f\ x)$ $(\lambda x.\ c *_R f'\ x)$
⟨*proof*⟩

lemma *cond-exp-scaleR-right*:
fixes f :: $'a \Rightarrow 'b$::{*second-countable-topology*,*banach*}
assumes *integrable M f*
shows *AE x in M. cond-exp M F* $(\lambda x.\ c *_R f\ x)$ $x = c *_R$ *cond-exp M F f x*
⟨*proof*⟩

lemma *cond-exp-uminus*:
fixes f :: $'a \Rightarrow 'b$::{*second-countable-topology*,*banach*}
assumes *integrable M f*
shows *AE x in M. cond-exp M F* $(\lambda x.\ -\ f\ x)$ $x = -$ *cond-exp M F f x*
⟨*proof*⟩

Together with the induction scheme *integrable-simple-function-induct*, we can show that the conditional expectation of an integrable simple function exists.

corollary *has-cond-exp-simple*:
fixes f :: $'a \Rightarrow 'b$::{*second-countable-topology*,*banach*}
assumes *simple-function M f emeasure M* $\{y \in$ *space M*. $f\ y \neq 0\} \neq \infty$
shows *has-cond-exp M F f* (*cond-exp M F f*)
⟨*proof*⟩

Now comes the most difficult part. Given a convergent sequence of integrable simple functions s, we must show that the sequence $\lambda n.$ *cond-exp M F* (s n) is also convergent. Furthermore, we must show that this limit satisfies the properties of a conditional expectation. Unfortunately, we will only be able to show that this sequence convergences in the L1-norm. Luckily, this is enough to show that the operator *cond-exp M F* preserves limits as a function from L1 to L1.

In anticipation of this result, we show that the conditional expectation operator is a contraction for simple functions. We first reformulate the lemma *real-cond-exp-abs*, which shows the statement for real-valued functions, using our definitions. Then we show the statement for simple functions via induction.

lemma *cond-exp-contraction-real*:
fixes f :: $'a \Rightarrow$ *real*
assumes *integrable*[*measurable*]: *integrable M f*

shows *AE x in M. norm (cond-exp M F f x) ≤ cond-exp M F (λx. norm (f x)) x*
⟨*proof*⟩

lemma *cond-exp-contraction-simple*:
fixes *f :: 'a ⇒ 'b::{second-countable-topology, banach}*
assumes *simple-function M f emeasure M {y ∈ space M. f y ≠ 0} ≠ ∞*
shows *AE x in M. norm (cond-exp M F f x) ≤ cond-exp M F (λx. norm (f x)) x*
⟨*proof*⟩

lemma *has-cond-exp-simple-lim*:
fixes *f :: 'a ⇒ 'b::{second-countable-topology, banach}*
assumes *integrable[measurable]: integrable M f*
and *⋀i. simple-function M (s i)*
and *⋀i. emeasure M {y ∈ space M. s i y ≠ 0} ≠ ∞*
and *⋀x. x ∈ space M ⟹ (λi. s i x) ⟶ f x*
and *⋀x i. x ∈ space M ⟹ norm (s i x) ≤ 2 ∗ norm (f x)*
obtains *r*
where *strict-mono r has-cond-exp M F f (λx. lim (λi. cond-exp M F (s (r i)) x))*
AE x in M. convergent (λi. cond-exp M F (s (r i)) x)
⟨*proof*⟩

Now, we can show that the conditional expectation is well-defined for all integrable functions.

corollary *has-cond-expI*:
fixes *f :: 'a ⇒ 'b::{second-countable-topology,banach}*
assumes *integrable M f*
shows *has-cond-exp M F f (cond-exp M F f)*
⟨*proof*⟩

4.2 Properties

The defining property of the conditional expectation now always holds, given that the function f is integrable.

lemma *cond-exp-set-integral*:
fixes *f :: 'a ⇒ 'b::{second-countable-topology,banach}*
assumes *integrable M f A ∈ sets F*
shows *(∫ x ∈ A. f x ∂M) = (∫ x ∈ A. cond-exp M F f x ∂M)*
⟨*proof*⟩

The following property of the conditional expectation is called the "Tower Property".

lemma *cond-exp-nested-subalg*:
fixes *f :: 'a ⇒ 'b::{second-countable-topology,banach}*
assumes *integrable M f subalgebra M G subalgebra G F*
shows *AE ξ in M. cond-exp M F f ξ = cond-exp M F (cond-exp M G f) ξ*
⟨*proof*⟩

The conditional expectation is linear.

lemma *cond-exp-add*:
fixes $f :: 'a \Rightarrow 'b::\{second\text{-}countable\text{-}topology,banach\}$
assumes *integrable M f integrable M g*
shows $AE\ x\ in\ M.\ cond\text{-}exp\ M\ F\ (\lambda x.\ f\ x + g\ x)\ x = cond\text{-}exp\ M\ F\ f\ x + cond\text{-}exp\ M\ F\ g\ x$
⟨*proof*⟩

lemma *cond-exp-diff*:
fixes $f :: 'a \Rightarrow 'b :: \{second\text{-}countable\text{-}topology,\ banach\}$
assumes *integrable M f integrable M g*
shows $AE\ x\ in\ M.\ cond\text{-}exp\ M\ F\ (\lambda x.\ f\ x - g\ x)\ x = cond\text{-}exp\ M\ F\ f\ x - cond\text{-}exp\ M\ F\ g\ x$
⟨*proof*⟩

lemma *cond-exp-diff'*:
fixes $f :: 'a \Rightarrow 'b :: \{second\text{-}countable\text{-}topology,\ banach\}$
assumes *integrable M f integrable M g*
shows $AE\ x\ in\ M.\ cond\text{-}exp\ M\ F\ (f - g)\ x = cond\text{-}exp\ M\ F\ f\ x - cond\text{-}exp\ M\ F\ g\ x$
⟨*proof*⟩

lemma *cond-exp-scaleR-left*:
fixes $f :: 'a \Rightarrow real$
assumes *integrable M f*
shows $AE\ x\ in\ M.\ cond\text{-}exp\ M\ F\ (\lambda x.\ f\ x *_R c)\ x = cond\text{-}exp\ M\ F\ f\ x *_R c$
⟨*proof*⟩

The conditional expectation operator is a contraction, i.e. a bounded linear operator with operator norm less than or equal to 1.

To show this we first obtain a subsequence $\lambda x\ i.\ s\ (r\ i)\ x$, such that $\lambda i.\ cond\text{-}exp\ M\ F\ (s\ (r\ i))\ x$ converges to $cond\text{-}exp\ M\ F\ f\ x$ a.e. Afterwards, we obtain a sub-subsequence $\lambda x\ i.\ s\ (r\ (r'\ i))\ x$, such that $\lambda i.\ cond\text{-}exp\ M\ F\ (\lambda x.\ norm\ (s\ (r\ i)))\ x$ converges to $cond\text{-}exp\ M\ F\ (\lambda x.\ norm\ (f\ x))\ x$ a.e. Finally, we show that the inequality holds by showing that the terms of the subsequences obey the inequality and the fact that a subsequence of a convergent sequence converges to the same limit.

lemma *cond-exp-contraction*:
fixes $f :: 'a \Rightarrow 'b::\{second\text{-}countable\text{-}topology,\ banach\}$
assumes *integrable M f*
shows $AE\ x\ in\ M.\ norm\ (cond\text{-}exp\ M\ F\ f\ x) \le cond\text{-}exp\ M\ F\ (\lambda x.\ norm\ (f\ x))\ x$
⟨*proof*⟩

The following lemmas are called "pulling out whats known". We first show the statement for real-valued functions using the lemma *real-cond-exp-intg*, which is already present. We then show it for arbitrary g using the lecture notes of Gordan Zitkovic for the course "Theory of Probability I" [5].

lemma *cond-exp-measurable-mult*:
fixes $f\ g :: 'a \Rightarrow real$
assumes [*measurable*]: $integrable\ M\ (\lambda x.\ f\ x * g\ x)\ integrable\ M\ g\ f \in borel\text{-}measurable\ F$
shows $integrable\ M\ (\lambda x.\ f\ x * cond\text{-}exp\ M\ F\ g\ x)$
$AE\ x\ in\ M.\ cond\text{-}exp\ M\ F\ (\lambda x.\ f\ x * g\ x)\ x = f\ x * cond\text{-}exp\ M\ F\ g\ x$
⟨*proof*⟩

lemma *cond-exp-measurable-scaleR*:
fixes $f :: 'a \Rightarrow real$ **and** $g :: 'a \Rightarrow 'b :: \{second\text{-}countable\text{-}topology,\ banach\}$
assumes [*measurable*]: $integrable\ M\ (\lambda x.\ f\ x *_R g\ x)\ integrable\ M\ g\ f \in borel\text{-}measurable\ F$
shows $integrable\ M\ (\lambda x.\ f\ x *_R cond\text{-}exp\ M\ F\ g\ x)$
$AE\ x\ in\ M.\ cond\text{-}exp\ M\ F\ (\lambda x.\ f\ x *_R g\ x)\ x = f\ x *_R cond\text{-}exp\ M\ F\ g\ x$
⟨*proof*⟩

lemma *cond-exp-sum* [*intro*, *simp*]:
fixes $f :: 't \Rightarrow 'a \Rightarrow 'b :: \{second\text{-}countable\text{-}topology,banach\}$
assumes [*measurable*]: $\bigwedge i.\ integrable\ M\ (f\ i)$
shows $AE\ x\ in\ M.\ cond\text{-}exp\ M\ F\ (\lambda x.\ \sum i{\in}I.\ f\ i\ x)\ x = (\sum i{\in}I.\ cond\text{-}exp\ M\ F\ (f\ i)\ x)$
⟨*proof*⟩

4.3 Linearly Ordered Banach Spaces

In this subsection we show monotonicity results concerning the conditional expectation operator.

lemma *cond-exp-gr-c*:
fixes $f :: 'a \Rightarrow 'b :: \{second\text{-}countable\text{-}topology,\ banach,\ linorder\text{-}topology,\ ordered\text{-}real\text{-}vector\}$
assumes $integrable\ M\ f\quad AE\ x\ in\ M.\ f\ x > c$
shows $AE\ x\ in\ M.\ cond\text{-}exp\ M\ F\ f\ x > c$
⟨*proof*⟩

corollary *cond-exp-less-c*:
fixes $f :: 'a \Rightarrow 'b :: \{second\text{-}countable\text{-}topology,\ banach,\ linorder\text{-}topology,\ ordered\text{-}real\text{-}vector\}$
assumes $integrable\ M\ f\ AE\ x\ in\ M.\ f\ x < c$
shows $AE\ x\ in\ M.\ cond\text{-}exp\ M\ F\ f\ x < c$
⟨*proof*⟩

lemma *cond-exp-mono-strict*:
fixes $f :: 'a \Rightarrow 'b :: \{second\text{-}countable\text{-}topology,\ banach,\ linorder\text{-}topology,\ ordered\text{-}real\text{-}vector\}$
assumes $integrable\ M\ f\ integrable\ M\ g\ AE\ x\ in\ M.\ f\ x < g\ x$
shows $AE\ x\ in\ M.\ cond\text{-}exp\ M\ F\ f\ x < cond\text{-}exp\ M\ F\ g\ x$
⟨*proof*⟩

lemma *cond-exp-ge-c*:
fixes $f :: 'a \Rightarrow 'b ::$ {*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*}
assumes [*measurable*]: *integrable M f*
and *AE x in M*. $f\ x \geq c$
shows *AE x in M*. *cond-exp M F f x* $\geq c$
⟨*proof*⟩

corollary *cond-exp-le-c*:
fixes $f :: 'a \Rightarrow 'b ::$ {*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*}
assumes *integrable M f*
and *AE x in M*. $f\ x \leq c$
shows *AE x in M*. *cond-exp M F f x* $\leq c$
⟨*proof*⟩

corollary *cond-exp-mono*:
fixes $f :: 'a \Rightarrow 'b ::$ {*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*}
assumes *integrable M f integrable M g AE x in M*. $f\ x \leq g\ x$
shows *AE x in M*. *cond-exp M F f x* $\leq$ *cond-exp M F g x*
⟨*proof*⟩

corollary *cond-exp-min*:
fixes $f :: 'a \Rightarrow 'b ::$ {*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*}
assumes *integrable M f integrable M g*
shows *AE* ξ *in M*. *cond-exp M F* (λx. *min* ($f\ x$) ($g\ x$)) $\xi \leq$ *min* (*cond-exp M F f* ξ) (*cond-exp M F g* ξ)
⟨*proof*⟩

corollary *cond-exp-max*:
fixes $f :: 'a \Rightarrow 'b ::$ {*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*}
assumes *integrable M f integrable M g*
shows *AE* ξ *in M*. *cond-exp M F* (λx. *max* ($f\ x$) ($g\ x$)) $\xi \geq$ *max* (*cond-exp M F f* ξ) (*cond-exp M F g* ξ)
⟨*proof*⟩

corollary *cond-exp-inf*:
fixes $f :: 'a \Rightarrow 'b ::$ {*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*, *lattice*}
assumes *integrable M f integrable M g*
shows *AE* ξ *in M*. *cond-exp M F* (λx. *inf* ($f\ x$) ($g\ x$)) $\xi \leq$ *inf* (*cond-exp M F f* ξ) (*cond-exp M F g* ξ)
⟨*proof*⟩

corollary *cond-exp-sup*:

fixes *f* :: *'a* ⇒ *'b* :: {*second-countable-topology*, *banach*, *linorder-topology*, *ordered-real-vector*, *lattice*}
assumes *integrable M f integrable M g*
shows *AE ξ in M. cond-exp M F* (*λx. sup* (*f x*) (*g x*)) *ξ* ≥ *sup* (*cond-exp M F f ξ*) (*cond-exp M F g ξ*)
⟨*proof*⟩

end

4.4 Probability Spaces

lemma (**in** *prob-space*) *sigma-finite-subalgebra-restr-to-subalg*:
assumes *subalgebra M F*
shows *sigma-finite-subalgebra M F*
⟨*proof*⟩

lemma (**in** *prob-space*) *cond-exp-trivial*:
fixes *f* :: *'a* ⇒ *'b* :: {*second-countable-topology*, *banach*}
assumes *integrable M f*
shows *AE x in M. cond-exp M* (*sigma* (*space M*) {}) *f x* = *expectation f*
⟨*proof*⟩

The following lemma shows that independent σ-algebras don't matter for the conditional expectation. The proof is adapted from [5].

lemma (**in** *prob-space*) *cond-exp-indep-subalgebra*:
fixes *f* :: *'a* ⇒ *'b* :: {*second-countable-topology*, *banach*, *real-normed-field*}
assumes *subalgebra*: *subalgebra M F subalgebra M G*
and *independent*: *indep-set G* (*sigma* (*space M*) (*F* ∪ *vimage-algebra* (*space M*) *f borel*))
assumes [*measurable*]: *integrable M f*
shows *AE x in M. cond-exp M* (*sigma* (*space M*) (*F* ∪ *G*)) *f x* = *cond-exp M F f x*
⟨*proof*⟩

If a random variable is independent of a σ-algebra F, its conditional expectation *cond-exp M F f* is just its expectation.

lemma (**in** *prob-space*) *cond-exp-indep*:
fixes *f* :: *'a* ⇒ *'b* :: {*second-countable-topology*, *banach*, *real-normed-field*}
assumes *subalgebra*: *subalgebra M F*
and *independent*: *indep-set F* (*vimage-algebra* (*space M*) *f borel*)
and *integrable*: *integrable M f*
shows *AE x in M. cond-exp M F f x* = *expectation f*
⟨*proof*⟩

end

theory *Filtered-Measure*
imports *HOL−Probability.Conditional-Expectation*

begin

5 Filtered Measure Spaces

5.1 Filtered Measure

locale *filtered-measure* =
  fixes M F **and** t_0 :: $'b$ :: {*second-countable-topology*, *order-topology*, *t2-space*}
  assumes *subalgebras*: $\bigwedge i.\ t_0 \leq i \Longrightarrow$ *subalgebra* M $(F\ i)$
    and *sets-F-mono*: $\bigwedge i\ j.\ t_0 \leq i \Longrightarrow i \leq j \Longrightarrow$ *sets* $(F\ i) \leq$ *sets* $(F\ j)$
begin

lemma *space-F*[*simp*]:
  assumes $t_0 \leq i$
  shows *space* $(F\ i)$ = *space* M
  ⟨*proof*⟩

lemma *subalgebra-F*[*intro*]:
  assumes $t_0 \leq i$ $i \leq j$
  shows *subalgebra* $(F\ j)$ $(F\ i)$
  ⟨*proof*⟩

lemma *borel-measurable-mono*:
  assumes $t_0 \leq i$ $i \leq j$
  shows *borel-measurable* $(F\ i) \subseteq$ *borel-measurable* $(F\ j)$
  ⟨*proof*⟩

end

locale *linearly-filtered-measure* = *filtered-measure* M F t_0 **for** M **and** F :: - :: {*linorder-topology*} ⇒ - **and** t_0

locale *nat-filtered-measure* = *linearly-filtered-measure* M F *0* **for** M **and** F :: *nat* ⇒ -
locale *real-filtered-measure* = *linearly-filtered-measure* M F *0* **for** M **and** F :: *real* ⇒ -

5.2 σ-Finite Filtered Measure

The locale presented here is a generalization of the *sigma-finite-subalgebra* for a particular filtration.

locale *sigma-finite-filtered-measure* = *filtered-measure* +
  assumes *sigma-finite-initial*: *sigma-finite-subalgebra* M $(F\ t_0)$

lemma (**in** *sigma-finite-filtered-measure*) *sigma-finite-subalgebra-F*[*intro*]:
  assumes $t_0 \leq i$
  shows *sigma-finite-subalgebra* M $(F\ i)$
  ⟨*proof*⟩

locale *nat-sigma-finite-filtered-measure* = *sigma-finite-filtered-measure* *M F 0* :: *nat* **for** *M F*
locale *real-sigma-finite-filtered-measure* = *sigma-finite-filtered-measure* *M F 0* :: *real* **for** *M F*

sublocale *nat-sigma-finite-filtered-measure* ⊆ *sigma-finite-subalgebra M F i* ⟨*proof*⟩
sublocale *real-sigma-finite-filtered-measure* ⊆ *sigma-finite-subalgebra M F* |*i*| ⟨*proof*⟩

5.3 Finite Filtered Measure

locale *finite-filtered-measure* = *filtered-measure* + *finite-measure*

sublocale *finite-filtered-measure* ⊆ *sigma-finite-filtered-measure*
⟨*proof*⟩

locale *nat-finite-filtered-measure* = *finite-filtered-measure M F 0* :: *nat* **for** *M F*
locale *real-finite-filtered-measure* = *finite-filtered-measure M F 0* :: *real* **for** *M F*

sublocale *nat-finite-filtered-measure* ⊆ *nat-sigma-finite-filtered-measure* ⟨*proof*⟩
sublocale *real-finite-filtered-measure* ⊆ *real-sigma-finite-filtered-measure* ⟨*proof*⟩

5.4 Constant Filtration

lemma *filtered-measure-constant-filtration*:
assumes *subalgebra M F*
shows *filtered-measure M* (λ-. *F*) t_0
⟨*proof*⟩

sublocale *sigma-finite-subalgebra* ⊆ *constant-filtration*: *sigma-finite-filtered-measure* *M* λ- :: ′*t* :: {*second-countable-topology*, *linorder-topology*}. *F* t_0
⟨*proof*⟩

lemma (**in** *finite-measure*) *filtered-measure-constant-filtration*:
assumes *subalgebra M F*
shows *finite-filtered-measure M* (λ-. *F*) t_0
⟨*proof*⟩

end

theory *Stochastic-Process*
imports *Filtered-Measure Measure-Space-Supplement HOL−Probability.Independent-Family*
begin

6 Stochastic Processes

6.1 Stochastic Process

A stochastic process is a collection of random variables, indexed by a type $'b$.

locale *stochastic-process* =
fixes M t_0 **and** X :: $'b$ :: {*second-countable-topology*, *order-topology*, *t2-space*} ⇒ $'a$ ⇒ $'c$ :: {*second-countable-topology*, *banach*}
assumes *random-variable*[*measurable*]: $\bigwedge i.\ t_0 \le i \Longrightarrow X\ i \in$ *borel-measurable* M
begin

definition *left-continuous* **where** *left-continuous* = (*AE* ξ *in* M. ∀ t. *continuous* (*at-left* t) (λi. X i ξ))
definition *right-continuous* **where** *right-continuous* = (*AE* ξ *in* M. ∀ t. *continuous* (*at-right* t) (λi. X i ξ))

end

We specify the following locales to formalize discrete time and continuous time processes.

locale *nat-stochastic-process* = *stochastic-process* M *0* :: *nat* X **for** M X
locale *real-stochastic-process* = *stochastic-process* M *0* :: *real* X **for** M X

lemma *stochastic-process-const-fun*:
assumes $f \in$ *borel-measurable* M
shows *stochastic-process* M t_0 (λ-. f) ⟨*proof*⟩

lemma *stochastic-process-const*:
shows *stochastic-process* M t_0 (λi -. c i) ⟨*proof*⟩

In the following segment, we cover basic operations on stochastic processes.

context *stochastic-process*
begin

lemma *compose-stochastic*:
assumes $\bigwedge i.\ t_0 \le i \Longrightarrow f\ i \in$ *borel-measurable borel*
shows *stochastic-process* M t_0 (λi ξ. (f i) (X i ξ))
⟨*proof*⟩

lemma *norm-stochastic*: *stochastic-process* M t_0 (λi ξ. *norm* (X i ξ)) ⟨*proof*⟩

lemma *scaleR-right-stochastic*:
assumes *stochastic-process* M t_0 Y
shows *stochastic-process* M t_0 (λi ξ. (Y i ξ) $*_R$ (X i ξ))
⟨*proof*⟩

lemma *scaleR-right-const-fun-stochastic*:

assumes $f \in$ *borel-measurable M*
shows *stochastic-process* $M\ t_0\ (\lambda i\ \xi.\ f\ \xi *_R (X\ i\ \xi))$
⟨*proof*⟩

lemma *scaleR-right-const-stochastic*: *stochastic-process* $M\ t_0\ (\lambda i\ \xi.\ c\ i *_R (X\ i\ \xi))$
⟨*proof*⟩

lemma *add-stochastic*:
assumes *stochastic-process* $M\ t_0\ Y$
shows *stochastic-process* $M\ t_0\ (\lambda i\ \xi.\ X\ i\ \xi + Y\ i\ \xi)$
⟨*proof*⟩

lemma *diff-stochastic*:
assumes *stochastic-process* $M\ t_0\ Y$
shows *stochastic-process* $M\ t_0\ (\lambda i\ \xi.\ X\ i\ \xi - Y\ i\ \xi)$
⟨*proof*⟩

lemma *uminus-stochastic*: *stochastic-process* $M\ t_0\ (-X)$ ⟨*proof*⟩

lemma *partial-sum-stochastic*: *stochastic-process* $M\ t_0\ (\lambda n\ \xi.\ \sum i{\in}\{t_0..n\}.\ X\ i\ \xi)$
⟨*proof*⟩

lemma *partial-sum′-stochastic*: *stochastic-process* $M\ t_0\ (\lambda n\ \xi.\ \sum i{\in}\{t_0..{<}n\}.\ X\ i\ \xi)$ ⟨*proof*⟩

end

lemma *stochastic-process-sum*:
assumes $\bigwedge i.\ i \in I \Longrightarrow$ *stochastic-process* $M\ t_0\ (X\ i)$
shows *stochastic-process* $M\ t_0\ (\lambda k\ \xi.\ \sum i \in I.\ X\ i\ k\ \xi)$ ⟨*proof*⟩

6.1.1 Natural Filtration

The natural filtration induced by a stochastic process X is the filtration generated by all events involving the process up to the time index t, i.e. $F\ t = \sigma(\{X\ s \mid s.\ s \leq t\})$.

definition *natural-filtration* :: $'a$ *measure* $\Rightarrow$ $'b \Rightarrow ('b \Rightarrow 'a \Rightarrow 'c$ :: *topological-space*) $\Rightarrow$ $'b$ :: {*second-countable-topology*, *order-topology*} $\Rightarrow$ $'a$ *measure* **where**
natural-filtration $M\ t_0\ Y = (\lambda t.$ *family-vimage-algebra* (*space* M) $\{Y\ i \mid i.\ i \in \{t_0..t\}\}$ *borel*)

abbreviation *nat-natural-filtration* $\equiv \lambda M.$ *natural-filtration* M (*0* :: *nat*)
abbreviation *real-natural-filtration* $\equiv \lambda M.$ *natural-filtration* M (*0* :: *real*)

lemma *space-natural-filtration*[*simp*]: *space* (*natural-filtration* $M\ t_0\ X\ t$) = *space* M ⟨*proof*⟩

lemma *sets-natural-filtration*: *sets* (*natural-filtration* $M\ t_0\ X\ t$) = *sigma-sets* (*space* M) $(\bigcup i{\in}\{t_0..t\}.\ \{X\ i$ -‘ $A \cap$ *space* $M \mid A.\ A \in$ *borel*$\})$

⟨*proof*⟩

lemma *sets-natural-filtration′*:
assumes *borel* = *sigma UNIV S*
shows *sets* (*natural-filtration M* t_0 *X t*) = *sigma-sets* (*space M*) (⋃ *i*∈{t_0..*t*}. {*X i* −' *A* ∩ *space M* | *A*. *A* ∈ *S*})
⟨*proof*⟩

lemma *sets-natural-filtration-open*:
sets (*natural-filtration M* t_0 *X t*) = *sigma-sets* (*space M*) (⋃ *i*∈{t_0..*t*}. {*X i* −' *A* ∩ *space M* | *A*. *open A*})
⟨*proof*⟩

lemma *sets-natural-filtration-oi*:
sets (*natural-filtration M* t_0 *X t*) = *sigma-sets* (*space M*) (⋃ *i*∈{t_0..*t*}. {*X i* −' *A* ∩ *space M* | *A* :: - :: {*linorder-topology*, *second-countable-topology*} *set*. *A* ∈ *range greaterThan*})
⟨*proof*⟩

lemma *sets-natural-filtration-io*:
sets (*natural-filtration M* t_0 *X t*) = *sigma-sets* (*space M*) (⋃ *i*∈{t_0..*t*}. {*X i* −' *A* ∩ *space M* | *A* :: - :: {*linorder-topology*, *second-countable-topology*} *set*. *A* ∈ *range lessThan*})
⟨*proof*⟩

lemma *sets-natural-filtration-ci*:
sets (*natural-filtration M* t_0 *X t*) = *sigma-sets* (*space M*) (⋃ *i*∈{t_0..*t*}. {*X i* −' *A* ∩ *space M* | *A* :: *real set*. *A* ∈ *range atLeast*})
⟨*proof*⟩

context *stochastic-process*
begin

lemma *subalgebra-natural-filtration*:
shows *subalgebra M* (*natural-filtration M* t_0 *X i*)
⟨*proof*⟩

lemma *filtered-measure-natural-filtration*:
shows *filtered-measure M* (*natural-filtration M* t_0 *X*) t_0
⟨*proof*⟩

In order to show that the natural filtration constitutes a filtered σ-finite measure, we need to provide a countable exhausting set in the preimage of $X\ t_0$.

lemma *sigma-finite-filtered-measure-natural-filtration*:
assumes *exhausting-set*: *countable A* (⋃ *A*) = *space M* ⋀*a*. *a* ∈ *A* ⟹ *emeasure M a* ≠ ∞ ⋀*a*. *a* ∈ *A* ⟹ ∃ *b* ∈ *borel*. *a* = *X* t_0 −' *b* ∩ *space M*
shows *sigma-finite-filtered-measure M* (*natural-filtration M* t_0 *X*) t_0
⟨*proof*⟩

lemma *finite-filtered-measure-natural-filtration*:
assumes *finite-measure M*
shows *finite-filtered-measure M* (*natural-filtration M* t_0 *X*) t_0
⟨*proof*⟩

end

Filtration generated by independent variables.

lemma (**in** *prob-space*) *indep-set-natural-filtration*:
assumes $t_0 \leq s$ $s < t$ *indep-vars* (λ-. *borel*) *X* {t_0..}
shows *indep-set* (*natural-filtration M* t_0 *X s*) (*vimage-algebra* (*space M*) (*X t*) *borel*)
⟨*proof*⟩

6.2 Adapted Process

We call a collection a stochastic process *X* adapted if *X i* is *F i*-borel-measurable for all indices *i*.

locale *adapted-process* = *filtered-measure M F* t_0 **for** *M F* t_0 **and** *X* :: - ⇒ - ⇒ - :: {*second-countable-topology*, *banach*} +
assumes *adapted*[*measurable*]: $\bigwedge i.\ t_0 \leq i \Longrightarrow X\ i \in$ *borel-measurable* (*F i*)
begin

lemma *adaptedE*[*elim*]:
assumes $[\![\bigwedge j\ i.\ t_0 \leq j \Longrightarrow j \leq i \Longrightarrow X\ j \in$ *borel-measurable* $(F\ i)]\!] \Longrightarrow P$
shows *P*
⟨*proof*⟩

lemma *adaptedD*:
assumes $t_0 \leq j$ $j \leq i$
shows *X j* ∈ *borel-measurable* (*F i*) ⟨*proof*⟩

end

locale *nat-adapted-process* = *adapted-process M F 0* :: *nat X* **for** *M F X*
locale *real-adapted-process* = *adapted-process M F 0* :: *real X* **for** *M F X*

sublocale *nat-adapted-process* ⊆ *nat-filtered-measure* ⟨*proof*⟩
sublocale *real-adapted-process* ⊆ *real-filtered-measure* ⟨*proof*⟩

lemma (**in** *filtered-measure*) *adapted-process-const-fun*:
assumes *f* ∈ *borel-measurable* (*F* t_0)
shows *adapted-process M F* t_0 (λ-. *f*)
⟨*proof*⟩

lemma (**in** *filtered-measure*) *adapted-process-const*:
shows *adapted-process M F* t_0 (λ*i* -. *c i*) ⟨*proof*⟩

Again, we cover basic operations.

context *adapted-process*
begin

lemma *compose-adapted*:
assumes $\bigwedge i.\ t_0 \leq i \Longrightarrow f\ i \in$ *borel-measurable borel*
shows *adapted-process* $M\ F\ t_0\ (\lambda i\ \xi.\ (f\ i)\ (X\ i\ \xi))$
⟨*proof*⟩

lemma *norm-adapted*: *adapted-process* $M\ F\ t_0\ (\lambda i\ \xi.\ norm\ (X\ i\ \xi))$ ⟨*proof*⟩

lemma *scaleR-right-adapted*:
assumes *adapted-process* $M\ F\ t_0\ R$
shows *adapted-process* $M\ F\ t_0\ (\lambda i\ \xi.\ (R\ i\ \xi) *_R (X\ i\ \xi))$
⟨*proof*⟩

lemma *scaleR-right-const-fun-adapted*:
assumes $f \in$ *borel-measurable* $(F\ t_0)$
shows *adapted-process* $M\ F\ t_0\ (\lambda i\ \xi.\ f\ \xi *_R (X\ i\ \xi))$
⟨*proof*⟩

lemma *scaleR-right-const-adapted*: *adapted-process* $M\ F\ t_0\ (\lambda i\ \xi.\ c\ i *_R (X\ i\ \xi))$
⟨*proof*⟩

lemma *add-adapted*:
assumes *adapted-process* $M\ F\ t_0\ Y$
shows *adapted-process* $M\ F\ t_0\ (\lambda i\ \xi.\ X\ i\ \xi + Y\ i\ \xi)$
⟨*proof*⟩

lemma *diff-adapted*:
assumes *adapted-process* $M\ F\ t_0\ Y$
shows *adapted-process* $M\ F\ t_0\ (\lambda i\ \xi.\ X\ i\ \xi - Y\ i\ \xi)$
⟨*proof*⟩

lemma *uminus-adapted*: *adapted-process* $M\ F\ t_0\ (-X)$ ⟨*proof*⟩

lemma *partial-sum-adapted*: *adapted-process* $M\ F\ t_0\ (\lambda n\ \xi.\ \sum i{\in}\{t_0..n\}.\ X\ i\ \xi)$
⟨*proof*⟩

lemma *partial-sum′-adapted*: *adapted-process* $M\ F\ t_0\ (\lambda n\ \xi.\ \sum i{\in}\{t_0..{<}n\}.\ X\ i\ \xi)$

⟨*proof*⟩

end

In the dicrete time case, we have the following lemma which will be useful later on.

lemma (**in** *nat-adapted-process*) *partial-sum-Suc-adapted*: *nat-adapted-process* $M\ F\ (\lambda n\ \xi.\ \sum i{<}n.\ X\ (Suc\ i)\ \xi)$

⟨*proof*⟩

lemma (**in** *filtered-measure*) *adapted-process-sum*:
 assumes $\bigwedge i.\ i \in I \Longrightarrow$ *adapted-process* $M\ F\ t_0\ (X\ i)$
 shows *adapted-process* $M\ F\ t_0\ (\lambda k\ \xi.\ \sum i \in I.\ X\ i\ k\ \xi)$
⟨*proof*⟩

An adapted process is necessarily a stochastic process.

sublocale *adapted-process* ⊆ *stochastic-process* ⟨*proof*⟩

sublocale *nat-adapted-process* ⊆ *nat-stochastic-process* ⟨*proof*⟩
sublocale *real-adapted-process* ⊆ *real-stochastic-process* ⟨*proof*⟩

A stochastic process is always adapted to the natural filtration it generates.

lemma (**in** *stochastic-process*) *adapted-process-natural-filtration*: *adapted-process* M (*natural-filtration* $M\ t_0\ X$) $t_0\ X$
 ⟨*proof*⟩

6.3 Progressively Measurable Process

locale *progressive-process* = *filtered-measure* $M\ F\ t_0$ **for** $M\ F\ t_0$ **and** X :: - ⇒ - ⇒ - :: {*second-countable-topology*, *banach*} +
 assumes *progressive*[*measurable*]: $\bigwedge t.\ t_0 \le t \Longrightarrow (\lambda(i, x).\ X\ i\ x) \in$ *borel-measurable* (*restrict-space borel* $\{t_0..t\} \bigotimes_M F\ t$)
begin

lemma *progressiveD*:
 assumes $S \in$ *borel*
 shows $(\lambda(j, \xi).\ X\ j\ \xi)$ −‘ $S \cap (\{t_0..i\} \times$ *space* $M) \in$ (*restrict-space borel* $\{t_0..i\} \bigotimes_M F\ i$)
 ⟨*proof*⟩

end

locale *nat-progressive-process* = *progressive-process* $M\ F$ *0* :: *nat* X **for** $M\ F\ X$
locale *real-progressive-process* = *progressive-process* $M\ F$ *0* :: *real* X **for** $M\ F\ X$

lemma (**in** *filtered-measure*) *progressive-process-const-fun*:
 assumes $f \in$ *borel-measurable* $(F\ t_0)$
 shows *progressive-process* $M\ F\ t_0\ (\lambda\text{-}.\ f)$
⟨*proof*⟩

lemma (**in** *filtered-measure*) *progressive-process-const*:
 assumes $c \in$ *borel-measurable borel*
 shows *progressive-process* $M\ F\ t_0\ (\lambda i\ \text{-}.\ c\ i)$
 ⟨*proof*⟩

context *progressive-process*
begin

lemma *compose-progressive*:
  assumes *case-prod f* $\in$ *borel-measurable borel*
  shows *progressive-process M F* t_0 $(\lambda i\ \xi.\ (f\ i)\ (X\ i\ \xi))$
⟨*proof*⟩

lemma *norm-progressive*: *progressive-process M F* t_0 $(\lambda i\ \xi.\ norm\ (X\ i\ \xi))$ ⟨*proof*⟩

lemma *scaleR-right-progressive*:
  assumes *progressive-process M F* t_0 *R*
  shows *progressive-process M F* t_0 $(\lambda i\ \xi.\ (R\ i\ \xi) *_R (X\ i\ \xi))$
  ⟨*proof*⟩

lemma *scaleR-right-const-fun-progressive*:
  assumes $f \in$ *borel-measurable* $(F\ t_0)$
  shows *progressive-process M F* t_0 $(\lambda i\ \xi.\ f\ \xi *_R (X\ i\ \xi))$
  ⟨*proof*⟩

lemma *scaleR-right-const-progressive*:
  assumes $c \in$ *borel-measurable borel*
  shows *progressive-process M F* t_0 $(\lambda i\ \xi.\ c\ i *_R (X\ i\ \xi))$
  ⟨*proof*⟩

lemma *add-progressive*:
  assumes *progressive-process M F* t_0 *Y*
  shows *progressive-process M F* t_0 $(\lambda i\ \xi.\ X\ i\ \xi + Y\ i\ \xi)$
  ⟨*proof*⟩

lemma *diff-progressive*:
  assumes *progressive-process M F* t_0 *Y*
  shows *progressive-process M F* t_0 $(\lambda i\ \xi.\ X\ i\ \xi - Y\ i\ \xi)$
  ⟨*proof*⟩

lemma *uminus-progressive*: *progressive-process M F* t_0 $(-X)$ ⟨*proof*⟩

end

A progressively measurable process is also adapted.

sublocale *progressive-process* $\subseteq$ *adapted-process* ⟨*proof*⟩

sublocale *nat-progressive-process* $\subseteq$ *nat-adapted-process* ⟨*proof*⟩
sublocale *real-progressive-process* $\subseteq$ *real-adapted-process* ⟨*proof*⟩

In the discrete setting, adaptedness is equivalent to progressive measurability.

theorem *nat-progressive-iff-adapted*: *nat-progressive-process M F X* $\longleftrightarrow$ *nat-adapted-process M F X*
⟨*proof*⟩

6.4 Predictable Process

We introduce the constant Σ_P to denote the predictable σ-algebra.

context *linearly-filtered-measure*
begin

definition Σ_P :: $('b \times 'a)$ *measure* **where** *predictable-sigma*: $\Sigma_P \equiv$ *sigma* $(\{t_0..\} \times$ *space* $M)$ $(\{\{s<..t\} \times A \mid A\ s\ t.\ A \in F\ s \wedge t_0 \leq s \wedge s < t\} \cup \{\{t_0\} \times A \mid A.\ A \in F\ t_0\})$

lemma *space-predictable-sigma*[*simp*]: *space* $\Sigma_P = (\{t_0..\} \times$ *space* $M)$ ⟨*proof*⟩

lemma *sets-predictable-sigma*: *sets* $\Sigma_P =$ *sigma-sets* $(\{t_0..\} \times$ *space* $M)$ $(\{\{s<..t\} \times A \mid A\ s\ t.\ A \in F\ s \wedge t_0 \leq s \wedge s < t\} \cup \{\{t_0\} \times A \mid A.\ A \in F\ t_0\})$
⟨*proof*⟩

lemma *measurable-predictable-sigma-snd*:
assumes *countable* $\mathcal{I}$ $\mathcal{I} \subseteq \{\{s<..t\} \mid s\ t.\ t_0 \leq s \wedge s < t\}$ $\{t_0<..\} \subseteq (\bigcup \mathcal{I})$
shows *snd* $\in \Sigma_P \rightarrow_M F\ t_0$
⟨*proof*⟩

lemma *measurable-predictable-sigma-fst*:
assumes *countable* $\mathcal{I}$ $\mathcal{I} \subseteq \{\{s<..t\} \mid s\ t.\ t_0 \leq s \wedge s < t\}$ $\{t_0<..\} \subseteq (\bigcup \mathcal{I})$
shows *fst* $\in \Sigma_P \rightarrow_M$ *borel*
⟨*proof*⟩

end

locale *predictable-process* = *linearly-filtered-measure* $M\ F\ t_0$ **for** $M\ F\ t_0$ **and** X :: - ⇒ - ⇒ - :: {*second-countable-topology*, *banach*} +
assumes *predictable*: $(\lambda(t, x).\ X\ t\ x) \in$ *borel-measurable* Σ_P
begin

lemmas *predictableD* = *measurable-sets*[*OF predictable*, *unfolded space-predictable-sigma*]

end

locale *nat-predictable-process* = *predictable-process* $M\ F\ 0$ :: *nat* X **for** $M\ F\ X$
locale *real-predictable-process* = *predictable-process* $M\ F\ 0$ :: *real* X **for** $M\ F\ X$

lemma (**in** *nat-filtered-measure*) *measurable-predictable-sigma-snd'*:
shows *snd* $\in \Sigma_P \rightarrow_M F\ 0$
⟨*proof*⟩

lemma (**in** *nat-filtered-measure*) *measurable-predictable-sigma-fst'*:
shows *fst* $\in \Sigma_P \rightarrow_M$ *borel*
⟨*proof*⟩

lemma (**in** *real-filtered-measure*) *measurable-predictable-sigma-snd'*:
shows *snd* $\in \Sigma_P \rightarrow_M F\ 0$
⟨*proof*⟩

lemma (**in** *real-filtered-measure*) *measurable-predictable-sigma-fst'*:
shows *fst* $\in \Sigma_P \rightarrow_M$ *borel*
⟨*proof*⟩

We show sufficient conditions for functions constant in one argument to constitute a predictable process. In contrast to the cases before, this is not a triviality.

lemma (**in** *linearly-filtered-measure*) *predictable-process-const-fun*:
assumes *snd* $\in \Sigma_P \rightarrow_M F\ t_0$ $f \in$ *borel-measurable* $(F\ t_0)$
shows *predictable-process* $M\ F\ t_0$ $(\lambda\text{-}.\ f)$
⟨*proof*⟩

lemma (**in** *nat-filtered-measure*) *predictable-process-const-fun'*[*intro*]:
assumes $f \in$ *borel-measurable* $(F\ 0)$
shows *nat-predictable-process* $M\ F$ $(\lambda\text{-}.\ f)$
⟨*proof*⟩

lemma (**in** *real-filtered-measure*) *predictable-process-const-fun'*[*intro*]:
assumes $f \in$ *borel-measurable* $(F\ 0)$
shows *real-predictable-process* $M\ F$ $(\lambda\text{-}.\ f)$
⟨*proof*⟩

lemma (**in** *linearly-filtered-measure*) *predictable-process-const*:
assumes *fst* $\in$ *borel-measurable* Σ_P $c \in$ *borel-measurable borel*
shows *predictable-process* $M\ F\ t_0$ $(\lambda i\ \text{-}.\ c\ i)$
⟨*proof*⟩

lemma (**in** *linearly-filtered-measure*) *predictable-process-const-const*[*intro*]:
shows *predictable-process* $M\ F\ t_0$ $(\lambda\text{-}\ \text{-}.\ c)$
⟨*proof*⟩

lemma (**in** *nat-filtered-measure*) *predictable-process-const'*[*intro*]:
assumes $c \in$ *borel-measurable borel*
shows *nat-predictable-process* $M\ F$ $(\lambda i\ \text{-}.\ c\ i)$
⟨*proof*⟩

lemma (**in** *real-filtered-measure*) *predictable-process-const'*[*intro*]:
assumes $c \in$ *borel-measurable borel*
shows *real-predictable-process* $M\ F$ $(\lambda i\ \text{-}.\ c\ i)$
⟨*proof*⟩

context *predictable-process*
begin

lemma *compose-predictable*:
assumes *fst* $\in$ *borel-measurable* Σ_P *case-prod f* $\in$ *borel-measurable borel*
shows *predictable-process* $M\ F\ t_0\ (\lambda i\ \xi.\ (f\ i)\ (X\ i\ \xi))$
⟨*proof*⟩

lemma *norm-predictable*: *predictable-process* $M\ F\ t_0\ (\lambda i\ \xi.\ norm\ (X\ i\ \xi))$ ⟨*proof*⟩

lemma *scaleR-right-predictable*:
assumes *predictable-process* $M\ F\ t_0\ R$
shows *predictable-process* $M\ F\ t_0\ (\lambda i\ \xi.\ (R\ i\ \xi) *_R (X\ i\ \xi))$
⟨*proof*⟩

lemma *scaleR-right-const-fun-predictable*:
assumes *snd* $\in \Sigma_P \rightarrow_M F\ t_0$ *f* $\in$ *borel-measurable* $(F\ t_0)$
shows *predictable-process* $M\ F\ t_0\ (\lambda i\ \xi.\ f\ \xi *_R (X\ i\ \xi))$
⟨*proof*⟩

lemma *scaleR-right-const-predictable*:
assumes *fst* $\in$ *borel-measurable* Σ_P *c* $\in$ *borel-measurable borel*
shows *predictable-process* $M\ F\ t_0\ (\lambda i\ \xi.\ c\ i *_R (X\ i\ \xi))$
⟨*proof*⟩

lemma *scaleR-right-const′-predictable*: *predictable-process* $M\ F\ t_0\ (\lambda i\ \xi.\ c *_R (X\ i\ \xi))$
⟨*proof*⟩

lemma *add-predictable*:
assumes *predictable-process* $M\ F\ t_0\ Y$
shows *predictable-process* $M\ F\ t_0\ (\lambda i\ \xi.\ X\ i\ \xi + Y\ i\ \xi)$
⟨*proof*⟩

lemma *diff-predictable*:
assumes *predictable-process* $M\ F\ t_0\ Y$
shows *predictable-process* $M\ F\ t_0\ (\lambda i\ \xi.\ X\ i\ \xi - Y\ i\ \xi)$
⟨*proof*⟩

lemma *uminus-predictable*: *predictable-process* $M\ F\ t_0\ (-X)$ ⟨*proof*⟩

end

Every predictable process is also progressively measurable.

sublocale *predictable-process* $\subseteq$ *progressive-process*
⟨*proof*⟩

sublocale *nat-predictable-process* $\subseteq$ *nat-progressive-process* ⟨*proof*⟩
sublocale *real-predictable-process* $\subseteq$ *real-progressive-process* ⟨*proof*⟩

The following lemma characterizes predictability in a discrete-time setting.

lemma (**in** *nat-filtered-measure*) *sets-in-filtration*:

assumes $(\bigcup i.\ \{i\} \times A\ i) \in \Sigma_P$
shows $A\ (\mathit{Suc}\ i) \in F\ i\ A\ 0 \in F\ 0$
⟨*proof*⟩

This leads to the following useful fact.

lemma (**in** *nat-predictable-process*) *adapted-Suc*: *nat-adapted-process* $M\ F\ (\lambda i.\ X\ (\mathit{Suc}\ i))$
⟨*proof*⟩

The following lemma characterizes predictability in the discrete setting.

theorem *nat-predictable-process-iff*: *nat-predictable-process* $M\ F\ X \longleftrightarrow$ *nat-adapted-process* $M\ F\ (\lambda i.\ X\ (\mathit{Suc}\ i)) \wedge X\ 0 \in$ *borel-measurable* $(F\ 0)$
⟨*proof*⟩

end

theory *Martingale*
imports *Stochastic-Process Conditional-Expectation-Banach*
begin

7 Martingales

The following locales are necessary for defining martingales.

7.1 Additional Locale Definitions

locale *sigma-finite-adapted-process* = *sigma-finite-filtered-measure* $M\ F\ t_0$ + *adapted-process* $M\ F\ t_0\ X$ **for** $M\ F\ t_0\ X$

locale *nat-sigma-finite-adapted-process* = *sigma-finite-adapted-process* $M\ F\ 0$ :: *nat* X **for** $M\ F\ X$
locale *real-sigma-finite-adapted-process* = *sigma-finite-adapted-process* $M\ F\ 0$ :: *real* X **for** $M\ F\ X$

sublocale *nat-sigma-finite-adapted-process* ⊆ *nat-sigma-finite-filtered-measure* ⟨*proof*⟩
sublocale *real-sigma-finite-adapted-process* ⊆ *real-sigma-finite-filtered-measure* ⟨*proof*⟩

locale *finite-adapted-process* = *finite-filtered-measure* $M\ F\ t_0$ + *adapted-process* $M\ F\ t_0\ X$ **for** $M\ F\ t_0\ X$

sublocale *finite-adapted-process* ⊆ *sigma-finite-adapted-process* ⟨*proof*⟩

locale *nat-finite-adapted-process* = *finite-adapted-process* $M\ F\ 0$ :: *nat* X **for** $M\ F\ X$
locale *real-finite-adapted-process* = *finite-adapted-process* $M\ F\ 0$ :: *real* X **for** $M\ F\ X$

sublocale *nat-finite-adapted-process* ⊆ *nat-sigma-finite-adapted-process* ⟨*proof*⟩
sublocale *real-finite-adapted-process* ⊆ *real-sigma-finite-adapted-process* ⟨*proof*⟩

locale *sigma-finite-adapted-process-order* = *sigma-finite-adapted-process* M F t_0 X **for** M F t_0 **and** X :: - ⇒ - ⇒ - :: {*order-topology*, *ordered-real-vector*}

locale *nat-sigma-finite-adapted-process-order* = *sigma-finite-adapted-process-order* M F *0* :: *nat* X **for** M F X
locale *real-sigma-finite-adapted-process-order* = *sigma-finite-adapted-process-order* M F *0* :: *real* X **for** M F X

sublocale *nat-sigma-finite-adapted-process-order* ⊆ *nat-sigma-finite-adapted-process* ⟨*proof*⟩
sublocale *real-sigma-finite-adapted-process-order* ⊆ *real-sigma-finite-adapted-process* ⟨*proof*⟩

locale *finite-adapted-process-order* = *finite-adapted-process* M F t_0 X **for** M F t_0 **and** X :: - ⇒ - ⇒ - :: {*order-topology*, *ordered-real-vector*}

locale *nat-finite-adapted-process-order* = *finite-adapted-process-order* M F *0* :: *nat* X **for** M F X
locale *real-finite-adapted-process-order* = *finite-adapted-process-order* M F *0* :: *real* X **for** M F X

sublocale *nat-finite-adapted-process-order* ⊆ *nat-sigma-finite-adapted-process-order* ⟨*proof*⟩
sublocale *real-finite-adapted-process-order* ⊆ *real-sigma-finite-adapted-process-order* ⟨*proof*⟩

locale *sigma-finite-adapted-process-linorder* = *sigma-finite-adapted-process-order* M F t_0 X **for** M F t_0 **and** X :: - ⇒ - ⇒ - :: {*linorder-topology*}

locale *nat-sigma-finite-adapted-process-linorder* = *sigma-finite-adapted-process-linorder* M F *0* :: *nat* X **for** M F X
locale *real-sigma-finite-adapted-process-linorder* = *sigma-finite-adapted-process-linorder* M F *0* :: *real* X **for** M F X

sublocale *nat-sigma-finite-adapted-process-linorder* ⊆ *nat-sigma-finite-adapted-process-order* ⟨*proof*⟩
sublocale *real-sigma-finite-adapted-process-linorder* ⊆ *real-sigma-finite-adapted-process-order* ⟨*proof*⟩

locale *finite-adapted-process-linorder* = *finite-adapted-process-order* M F t_0 X **for** M F t_0 **and** X :: - ⇒ - ⇒ - :: {*linorder-topology*}

locale *nat-finite-adapted-process-linorder* = *finite-adapted-process-linorder M F 0* :: *nat X* **for** *M F X*
locale *real-finite-adapted-process-linorder* = *finite-adapted-process-linorder M F 0* :: *real X* **for** *M F X*

sublocale *nat-finite-adapted-process-linorder* ⊆ *nat-sigma-finite-adapted-process-linorder* ⟨*proof*⟩
sublocale *real-finite-adapted-process-linorder* ⊆ *real-sigma-finite-adapted-process-linorder* ⟨*proof*⟩

7.2 Martingale

A martingale is an adapted process where the expected value of the next observation, given all past observations, is equal to the current value.

locale *martingale* = *sigma-finite-adapted-process* +
assumes *integrable*: $\bigwedge i.\ t_0 \leq i \Longrightarrow$ *integrable* $M\ (X\ i)$
and *martingale-property*: $\bigwedge i\ j.\ t_0 \leq i \Longrightarrow i \leq j \Longrightarrow AE\ \xi\ in\ M.\ X\ i\ \xi =$ *cond-exp* $M\ (F\ i)\ (X\ j)\ \xi$

locale *martingale-order* = *martingale* $M\ F\ t_0\ X$ **for** $M\ F\ t_0$ **and** X :: - ⇒ - ⇒ - :: {*order-topology*, *ordered-real-vector*}
locale *martingale-linorder* = *martingale* $M\ F\ t_0\ X$ **for** $M\ F\ t_0$ **and** X :: - ⇒ - ⇒ - :: {*linorder-topology*, *ordered-real-vector*}
sublocale *martingale-linorder* ⊆ *martingale-order* ⟨*proof*⟩

lemma (**in** *sigma-finite-filtered-measure*) *martingale-const-fun*[*intro*]:
assumes *integrable* $M\ f\ f \in$ *borel-measurable* $(F\ t_0)$
shows *martingale* $M\ F\ t_0\ (\lambda\text{-}.\ f)$
⟨*proof*⟩

lemma (**in** *sigma-finite-filtered-measure*) *martingale-cond-exp*[*intro*]:
assumes *integrable* $M\ f$
shows *martingale* $M\ F\ t_0\ (\lambda i.$ *cond-exp* $M\ (F\ i)\ f)$
⟨*proof*⟩

corollary (**in** *sigma-finite-filtered-measure*) *martingale-zero*[*intro*]: *martingale* $M\ F\ t_0\ (\lambda\text{- -}.\ 0)$ ⟨*proof*⟩

corollary (**in** *finite-filtered-measure*) *martingale-const*[*intro*]: *martingale* $M\ F\ t_0\ (\lambda\text{- -}.\ c)$ ⟨*proof*⟩

7.3 Submartingale

A submartingale is an adapted process where the expected value of the next observation, given all past observations, is greater than or equal to the current value.

locale *submartingale* = *sigma-finite-adapted-process-order* +
assumes *integrable*: $\bigwedge i.\ t_0 \leq i \Longrightarrow$ *integrable* $M\ (X\ i)$

and *submartingale-property*: $\bigwedge i\ j.\ t_0 \leq i \Longrightarrow i \leq j \Longrightarrow AE\ \xi\ in\ M.\ X\ i\ \xi \leq$ *cond-exp* $M\ (F\ i)\ (X\ j)\ \xi$

locale *submartingale-linorder* = *submartingale* $M\ F\ t_0\ X$ **for** $M\ F\ t_0$ **and** X :: - ⇒ - ⇒ - :: {*linorder-topology*}

sublocale *martingale-order* ⊆ *submartingale* ⟨*proof*⟩
sublocale *martingale-linorder* ⊆ *submartingale-linorder* ⟨*proof*⟩

7.4 Supermartingale

A supermartingale is an adapted process where the expected value of the next observation, given all past observations, is less than or equal to the current value.

locale *supermartingale* = *sigma-finite-adapted-process-order* +
assumes *integrable*: $\bigwedge i.\ t_0 \leq i \Longrightarrow$ *integrable* $M\ (X\ i)$
and *supermartingale-property*: $\bigwedge i\ j.\ t_0 \leq i \Longrightarrow i \leq j \Longrightarrow AE\ \xi\ in\ M.\ X\ i\ \xi \geq$ *cond-exp* $M\ (F\ i)\ (X\ j)\ \xi$

locale *supermartingale-linorder* = *supermartingale* $M\ F\ t_0\ X$ **for** $M\ F\ t_0$ **and** X :: - ⇒ - ⇒ - :: {*linorder-topology*}

sublocale *martingale-order* ⊆ *supermartingale* ⟨*proof*⟩
sublocale *martingale-linorder* ⊆ *supermartingale-linorder* ⟨*proof*⟩

A stochastic process is a martingale, if and only if it is both a submartingale and a supermartingale.

lemma *martingale-iff*:
shows *martingale* $M\ F\ t_0\ X \longleftrightarrow$ *submartingale* $M\ F\ t_0\ X \land$ *supermartingale* $M\ F\ t_0\ X$
⟨*proof*⟩

7.5 Martingale Lemmas

In the following segment, we cover basic properties of martingales.

context *martingale*
begin

lemma *cond-exp-diff-eq-zero*:
assumes $t_0 \leq i\ i \leq j$
shows $AE\ \xi\ in\ M.$ *cond-exp* $M\ (F\ i)\ (\lambda\xi.\ X\ j\ \xi - X\ i\ \xi)\ \xi = 0$
⟨*proof*⟩

lemma *set-integral-eq*:
assumes $A \in F\ i\ t_0 \leq i\ i \leq j$
shows *set-lebesgue-integral* $M\ A\ (X\ i)$ = *set-lebesgue-integral* $M\ A\ (X\ j)$
⟨*proof*⟩

lemma *scaleR-const*[*intro*]:
shows *martingale* M F t_0 (λi x. $c *_R X$ i x)
⟨*proof*⟩

lemma *uminus*[*intro*]:
shows *martingale* M F t_0 ($-$ X)
⟨*proof*⟩

lemma *add*[*intro*]:
assumes *martingale* M F t_0 Y
shows *martingale* M F t_0 (λi ξ. X i ξ + Y i ξ)
⟨*proof*⟩

lemma *diff*[*intro*]:
assumes *martingale* M F t_0 Y
shows *martingale* M F t_0 (λi x. X i x $-$ Y i x)
⟨*proof*⟩

end

Using properties of the conditional expectation, we present the following alternative characterizations of martingales.

lemma (**in** *sigma-finite-adapted-process*) *martingale-of-cond-exp-diff-eq-zero*:
assumes *integrable*: $\bigwedge i$. $t_0 \leq i \Longrightarrow$ *integrable* M (X i)
and *diff-zero*: $\bigwedge i$ j. $t_0 \leq i \Longrightarrow i \leq j \Longrightarrow$ *AE* x *in* M. *cond-exp* M (F i) ($\lambda \xi$. X j ξ $-$ X i ξ) x = *0*
shows *martingale* M F t_0 X
⟨*proof*⟩

lemma (**in** *sigma-finite-adapted-process*) *martingale-of-set-integral-eq*:
assumes *integrable*: $\bigwedge i$. $t_0 \leq i \Longrightarrow$ *integrable* M (X i)
and $\bigwedge A$ i j. $t_0 \leq i \Longrightarrow i \leq j \Longrightarrow A \in F$ $i \Longrightarrow$ *set-lebesgue-integral* M A (X i) = *set-lebesgue-integral* M A (X j)
shows *martingale* M F t_0 X
⟨*proof*⟩

7.6 Submartingale Lemmas

context *submartingale*
begin

lemma *cond-exp-diff-nonneg*:
assumes $t_0 \leq i$ $i \leq j$
shows *AE* x *in* M. *cond-exp* M (F i) ($\lambda \xi$. X j ξ $-$ X i ξ) $x \geq$ *0*
⟨*proof*⟩

lemma *add*[*intro*]:
assumes *submartingale* M F t_0 Y

shows *submartingale* $M\ F\ t_0\ (\lambda i\ \xi.\ X\ i\ \xi + Y\ i\ \xi)$
⟨*proof*⟩

lemma *diff*[*intro*]:
assumes *supermartingale* $M\ F\ t_0\ Y$
shows *submartingale* $M\ F\ t_0\ (\lambda i\ \xi.\ X\ i\ \xi - Y\ i\ \xi)$
⟨*proof*⟩

lemma *scaleR-nonneg*:
assumes $c \geq 0$
shows *submartingale* $M\ F\ t_0\ (\lambda i\ \xi.\ c *_R X\ i\ \xi)$
⟨*proof*⟩

lemma *scaleR-le-zero*:
assumes $c \leq 0$
shows *supermartingale* $M\ F\ t_0\ (\lambda i\ \xi.\ c *_R X\ i\ \xi)$
⟨*proof*⟩

lemma *uminus*[*intro*]:
shows *supermartingale* $M\ F\ t_0\ (-\ X)$
⟨*proof*⟩

end

context *submartingale-linorder*
begin

lemma *set-integral-le*:
assumes $A \in F\ i\ t_0 \leq i\ i \leq j$
shows *set-lebesgue-integral* $M\ A\ (X\ i) \leq$ *set-lebesgue-integral* $M\ A\ (X\ j)$
⟨*proof*⟩

lemma *max*:
assumes *submartingale-linorder* $M\ F\ t_0\ Y$
shows *submartingale-linorder* $M\ F\ t_0\ (\lambda i\ \xi.\ max\ (X\ i\ \xi)\ (Y\ i\ \xi))$
⟨*proof*⟩

lemma *max-0*:
shows *submartingale-linorder* $M\ F\ t_0\ (\lambda i\ \xi.\ max\ 0\ (X\ i\ \xi))$
⟨*proof*⟩

end

lemma (**in** *sigma-finite-adapted-process-order*) *submartingale-of-cond-exp-diff-nonneg*:
assumes *integrable*: $\bigwedge i.\ t_0 \leq i \Longrightarrow$ *integrable* $M\ (X\ i)$
and *diff-nonneg*: $\bigwedge i\ j.\ t_0 \leq i \Longrightarrow i \leq j \Longrightarrow AE\ x\ in\ M.\ cond\text{-}exp\ M\ (F\ i)$
$(\lambda\xi.\ X\ j\ \xi - X\ i\ \xi)\ x \geq 0$
shows *submartingale* $M\ F\ t_0\ X$
⟨*proof*⟩

lemma (**in** *sigma-finite-adapted-process-linorder*) *submartingale-of-set-integral-le*:
assumes *integrable*: $\bigwedge i.\ t_0 \leq i \Longrightarrow$ *integrable* $M\ (X\ i)$
and $\bigwedge A\ i\ j.\ t_0 \leq i \Longrightarrow i \leq j \Longrightarrow A \in F\ i \Longrightarrow$ *set-lebesgue-integral* $M\ A\ (X\ i) \leq$ *set-lebesgue-integral* $M\ A\ (X\ j)$
shows *submartingale* $M\ F\ t_0\ X$
⟨*proof*⟩

7.7 Supermartingale Lemmas

The following lemmas are exact duals of the ones for submartingales.

context *supermartingale*
begin

lemma *cond-exp-diff-nonneg*:
assumes $t_0 \leq i\ i \leq j$
shows *AE x in M*. *cond-exp* $M\ (F\ i)\ (\lambda\xi.\ X\ i\ \xi - X\ j\ \xi)\ x \geq 0$
⟨*proof*⟩

lemma *add*[*intro*]:
assumes *supermartingale* $M\ F\ t_0\ Y$
shows *supermartingale* $M\ F\ t_0\ (\lambda i\ \xi.\ X\ i\ \xi + Y\ i\ \xi)$
⟨*proof*⟩

lemma *diff*[*intro*]:
assumes *submartingale* $M\ F\ t_0\ Y$
shows *supermartingale* $M\ F\ t_0\ (\lambda i\ \xi.\ X\ i\ \xi - Y\ i\ \xi)$
⟨*proof*⟩

lemma *scaleR-nonneg*:
assumes $c \geq 0$
shows *supermartingale* $M\ F\ t_0\ (\lambda i\ \xi.\ c *_R X\ i\ \xi)$
⟨*proof*⟩

lemma *scaleR-le-zero*:
assumes $c \leq 0$
shows *submartingale* $M\ F\ t_0\ (\lambda i\ \xi.\ c *_R X\ i\ \xi)$
⟨*proof*⟩

lemma *uminus*[*intro*]:
shows *submartingale* $M\ F\ t_0\ (-\ X)$
⟨*proof*⟩

end

context *supermartingale-linorder*
begin

lemma *set-integral-ge*:

assumes $A \in F\ i\ t_0 \leq i\ i \leq j$
shows *set-lebesgue-integral* $M\ A\ (X\ i) \geq$ *set-lebesgue-integral* $M\ A\ (X\ j)$
⟨*proof*⟩

lemma *min*:
assumes *supermartingale-linorder* $M\ F\ t_0\ Y$
shows *supermartingale-linorder* $M\ F\ t_0\ (\lambda i\ \xi.\ min\ (X\ i\ \xi)\ (Y\ i\ \xi))$
⟨*proof*⟩

lemma *min-0*:
shows *supermartingale-linorder* $M\ F\ t_0\ (\lambda i\ \xi.\ min\ 0\ (X\ i\ \xi))$
⟨*proof*⟩

end

lemma (**in** *sigma-finite-adapted-process-order*) *supermartingale-of-cond-exp-diff-le-zero*:
assumes *integrable*: $\bigwedge i.\ t_0 \leq i \Longrightarrow$ *integrable* $M\ (X\ i)$
and *diff-le-zero*: $\bigwedge i\ j.\ t_0 \leq i \Longrightarrow i \leq j \Longrightarrow AE\ x\ in\ M.$ *cond-exp* $M\ (F\ i)$ $(\lambda\xi.\ X\ j\ \xi - X\ i\ \xi)\ x \leq 0$
shows *supermartingale* $M\ F\ t_0\ X$
⟨*proof*⟩

lemma (**in** *sigma-finite-adapted-process-linorder*) *supermartingale-of-set-integral-ge*:
assumes *integrable*: $\bigwedge i.\ t_0 \leq i \Longrightarrow$ *integrable* $M\ (X\ i)$
and $\bigwedge A\ i\ j.\ t_0 \leq i \Longrightarrow i \leq j \Longrightarrow A \in F\ i \Longrightarrow$ *set-lebesgue-integral* $M\ A\ (X\ j) \leq$ *set-lebesgue-integral* $M\ A\ (X\ i)$
shows *supermartingale* $M\ F\ t_0\ X$
⟨*proof*⟩

Many of the statements we have made concerning martingales can be simplified when the indexing set is the natural numbers. Given a point in time $i \in \mathbb{N}$, it suffices to consider the successor $i + (1::'a)$, instead of all future times $i \leq j$.

7.8 Discrete Time Martingales

locale *nat-martingale* = *martingale* $M\ F\ 0 :: nat\ X$ **for** $M\ F\ X$
locale *nat-submartingale* = *submartingale* $M\ F\ 0 :: nat\ X$ **for** $M\ F\ X$
locale *nat-supermartingale* = *supermartingale* $M\ F\ 0 :: nat\ X$ **for** $M\ F\ X$

locale *nat-submartingale-linorder* = *submartingale-linorder* $M\ F\ 0 :: nat\ X$ **for** $M\ F\ X$
locale *nat-supermartingale-linorder* = *supermartingale-linorder* $M\ F\ 0 :: nat\ X$ **for** $M\ F\ X$

sublocale *nat-submartingale-linorder* ⊆ *nat-submartingale* ⟨*proof*⟩
sublocale *nat-supermartingale-linorder* ⊆ *nat-supermartingale* ⟨*proof*⟩

A predictable martingale is necessarily constant.

lemma (**in** *nat-martingale*) *predictable-const*:
assumes *nat-predictable-process M F X*
shows $AE\ \xi\ in\ M.\ X\ i\ \xi = X\ j\ \xi$
⟨*proof*⟩

lemma (**in** *nat-sigma-finite-adapted-process*) *martingale-of-set-integral-eq-Suc*:
assumes *integrable*: $\bigwedge i.$ *integrable* $M\ (X\ i)$
and $\bigwedge A\ i.\ A \in F\ i \Longrightarrow$ *set-lebesgue-integral* $M\ A\ (X\ i) =$ *set-lebesgue-integral* $M\ A\ (X\ (Suc\ i))$
shows *nat-martingale M F X*
⟨*proof*⟩

lemma (**in** *nat-sigma-finite-adapted-process*) *martingale-nat*:
assumes *integrable*: $\bigwedge i.$ *integrable* $M\ (X\ i)$
and $\bigwedge i.\ AE\ \xi\ in\ M.\ X\ i\ \xi =$ *cond-exp* $M\ (F\ i)\ (X\ (Suc\ i))\ \xi$
shows *nat-martingale M F X*
⟨*proof*⟩

lemma (**in** *nat-sigma-finite-adapted-process*) *martingale-of-cond-exp-diff-Suc-eq-zero*:
assumes *integrable*: $\bigwedge i.$ *integrable* $M\ (X\ i)$
and $\bigwedge i.\ AE\ \xi\ in\ M.$ *cond-exp* $M\ (F\ i)\ (\lambda\xi.\ X\ (Suc\ i)\ \xi - X\ i\ \xi)\ \xi = 0$
shows *nat-martingale M F X*
⟨*proof*⟩

7.9 Discrete Time Submartingales

lemma (**in** *nat-submartingale*) *predictable-mono*:
assumes *nat-predictable-process M F X* $i \le j$
shows $AE\ \xi\ in\ M.\ X\ i\ \xi \le X\ j\ \xi$
⟨*proof*⟩

lemma (**in** *nat-sigma-finite-adapted-process-linorder*) *submartingale-of-set-integral-le-Suc*:
assumes *integrable*: $\bigwedge i.$ *integrable* $M\ (X\ i)$
and $\bigwedge A\ i.\ A \in F\ i \Longrightarrow$ *set-lebesgue-integral* $M\ A\ (X\ i) \le$ *set-lebesgue-integral* $M\ A\ (X\ (Suc\ i))$
shows *nat-submartingale M F X*
⟨*proof*⟩

lemma (**in** *nat-sigma-finite-adapted-process-linorder*) *submartingale-nat*:
assumes *integrable*: $\bigwedge i.$ *integrable* $M\ (X\ i)$
and $\bigwedge i.\ AE\ \xi\ in\ M.\ X\ i\ \xi \le$ *cond-exp* $M\ (F\ i)\ (X\ (Suc\ i))\ \xi$
shows *nat-submartingale M F X*
⟨*proof*⟩

lemma (**in** *nat-sigma-finite-adapted-process-linorder*) *submartingale-of-cond-exp-diff-Suc-nonneg*:
assumes *integrable*: $\bigwedge i.$ *integrable* $M\ (X\ i)$
and $\bigwedge i.\ AE\ \xi\ in\ M.$ *cond-exp* $M\ (F\ i)\ (\lambda\xi.\ X\ (Suc\ i)\ \xi - X\ i\ \xi)\ \xi \ge 0$
shows *nat-submartingale M F X*
⟨*proof*⟩

lemma (**in** *nat-submartingale-linorder*) *partial-sum-scaleR*:
assumes *nat-adapted-process M F C* $\bigwedge i.$ *AE* ξ *in M.* $0 \leq C\ i\ \xi$ $\bigwedge i.$ *AE* ξ *in M.* $C\ i\ \xi \leq R$
shows *nat-submartingale M F* $(\lambda n\ \xi.\ \sum i{<}n.\ C\ i\ \xi *_R (X\ (Suc\ i)\ \xi - X\ i\ \xi))$
⟨*proof*⟩

lemma (**in** *nat-submartingale-linorder*) *partial-sum-scaleR'*:
assumes *nat-predictable-process M F C* $\bigwedge i.$ *AE* ξ *in M.* $0 \leq C\ i\ \xi$ $\bigwedge i.$ *AE* ξ *in M.* $C\ i\ \xi \leq R$
shows *nat-submartingale M F* $(\lambda n\ \xi.\ \sum i{<}n.\ C\ (Suc\ i)\ \xi *_R (X\ (Suc\ i)\ \xi - X\ i\ \xi))$
⟨*proof*⟩

7.10 Discrete Time Supermartingales

lemma (**in** *nat-supermartingale*) *predictable-mono*:
assumes *nat-predictable-process M F X* $i \leq j$
shows *AE* ξ *in M.* $X\ i\ \xi \geq X\ j\ \xi$
⟨*proof*⟩

lemma (**in** *nat-sigma-finite-adapted-process-linorder*) *supermartingale-of-set-integral-ge-Suc*:
assumes *integrable*: $\bigwedge i.$ *integrable M* $(X\ i)$
and $\bigwedge A\ i.\ A \in F\ i \Longrightarrow$ *set-lebesgue-integral M A* $(X\ i) \geq$ *set-lebesgue-integral M A* $(X\ (Suc\ i))$
shows *nat-supermartingale M F X*
⟨*proof*⟩

lemma (**in** *nat-sigma-finite-adapted-process-linorder*) *supermartingale-nat*:
assumes *integrable*: $\bigwedge i.$ *integrable M* $(X\ i)$
and $\bigwedge i.$ *AE* ξ *in M.* $X\ i\ \xi \geq$ *cond-exp M* $(F\ i)\ (X\ (Suc\ i))\ \xi$
shows *nat-supermartingale M F X*
⟨*proof*⟩

lemma (**in** *nat-sigma-finite-adapted-process-linorder*) *supermartingale-of-cond-exp-diff-Suc-le-zero*:
assumes *integrable*: $\bigwedge i.$ *integrable M* $(X\ i)$
and $\bigwedge i.$ *AE* ξ *in M.* *cond-exp M* $(F\ i)\ (\lambda\xi.\ X\ (Suc\ i)\ \xi - X\ i\ \xi)\ \xi \leq 0$
shows *nat-supermartingale M F X*
⟨*proof*⟩

end

theory *Example-Coin-Toss*
imports *Martingale HOL−Probability.Stream-Space HOL−Probability.Probability-Mass-Function*
begin

8 Example: Coin Toss

Consider a coin-tossing game, where the coin lands on heads with probability $p \in [0, 1]$. Assume that the gambler wins a fixed amount $c > 0$ on a heads outcome and loses the same amount c on a tails outcome. Let $(X_n)_{n \in \mathbb{N}}$ be a stochastic process, where X_n denotes the gamblers fortune after the n-th coin toss. Then, we have the following three cases.

1. If $p = 1/2$, it means the coin is fair and has an equal chance of landing heads or tails. In this case, the gambler, on average, neither wins nor loses money over time. The expected value of the gamblers fortune stays the same over time. Therefore, $(X_n)_{n \in \mathbb{N}}$ is a martingale.

2. If $p \geq 1/2$, it means the coin is biased in favor of heads. In this case, the gambler is more likely to win money on each bet. Over time, the gamblers fortune tends to increase on average. Therefore, $(X_n)_{n \in \mathbb{N}}$ is a submartingale.

3. If $p \leq 1/2$, it means the coin is biased in favor of tails. In this scenario, the gambler is more likely to lose money on each bet. Over time, the gamblers fortune decreases on average. Therefore, $(X_n)_{n \in \mathbb{N}}$ is a supermartingale.

To formalize this example, we first consider a probability space consisting of infinite sequences of coin tosses.

definition *bernoulli-stream* :: *real* ⇒ (*bool stream*) *measure* **where**
bernoulli-stream p = *stream-space* (*measure-pmf* (*bernoulli-pmf p*))

lemma *space-bernoulli-stream*[*simp*]: *space* (*bernoulli-stream p*) = *UNIV* ⟨*proof*⟩

We define the fortune of the player at time n to be the number of heads minus number of tails.

definition *fortune* :: *nat* ⇒ *bool stream* ⇒ *real* **where**
fortune n = ($\lambda s.$ $\sum$ *b* ← *stake* (*Suc n*) *s*. *if b then 1 else* −*1*)

definition *toss* :: *nat* ⇒ *bool stream* ⇒ *real* **where**
toss n = ($\lambda s.$ *if snth s n then 1 else* −*1*)

lemma *toss-indicator-def*: *toss n* = *indicator* {*s*. *s* !! *n*} − *indicator* {*s*. ¬ *s* !! *n*}
⟨*proof*⟩

lemma *range-toss*: *range* (*toss n*) = {−*1*, *1*}
⟨*proof*⟩

lemma *vimage-toss*: *toss n* −‘ *A* = (*if 1* ∈ *A then* {*s*. *s* !! *n*} *else* {}) ∪ (*if* −*1* ∈ *A then* {*s*. ¬*s* !! *n*} *else* {})
⟨*proof*⟩

lemma *fortune-Suc*: *fortune* (*Suc n*) *s* = *fortune n s* + *toss* (*Suc n*) *s*

⟨*proof*⟩

lemma *fortune-toss-sum*: *fortune n s* = ($\sum$ *i* ∈ {..*n*}. *toss i s*)
⟨*proof*⟩

lemma *fortune-bound*: *norm* (*fortune n s*) ≤ *Suc n* ⟨*proof*⟩

Our definition of *bernoulli-stream* constitutes a probability space.

interpretation *prob-space bernoulli-stream p* ⟨*proof*⟩

abbreviation *toss-filtration p* ≡ *nat-natural-filtration* (*bernoulli-stream p*) *toss*

The stochastic process *toss* is adapted to the filtration it generates.

interpretation *toss*: *nat-adapted-process bernoulli-stream p nat-natural-filtration* (*bernoulli-stream p*) *toss toss*
⟨*proof*⟩

Similarly, the stochastic process *fortune* is adapted to the filtration generated by the tosses.

interpretation *fortune*: *nat-finite-adapted-process-linorder bernoulli-stream p nat-natural-filtration* (*bernoulli-stream p*) *toss fortune*
⟨*proof*⟩

lemma *integrable-toss*: *integrable* (*bernoulli-stream p*) (*toss n*)
⟨*proof*⟩

lemma *integrable-fortune*: *integrable* (*bernoulli-stream p*) (*fortune n*) ⟨*proof*⟩

We provide the following lemma to explicitly calculate the probability of events in this probability space.

lemma *measure-bernoulli-stream-snth-pred*:
assumes *0* ≤ *p* **and** *p* ≤ *1* **and** *finite J*
shows *prob p* {*w* ∈ *space* (*bernoulli-stream p*). ∀ *j*∈*J*. *P j* = *w* !! *j*} = *p* ^ *card* (*J* ∩ *Collect P*) ∗ (*1* − *p*) ^ (*card* (*J* − *Collect P*))
⟨*proof*⟩

lemma
assumes *0* ≤ *p* **and** *p* ≤ *1*
shows *measure-bernoulli-stream-snth*: *prob p* {*w* ∈ *space* (*bernoulli-stream p*). *w* !! *i*} = *p*
and *measure-bernoulli-stream-neg-snth*: *prob p* {*w* ∈ *space* (*bernoulli-stream p*). ¬*w* !! *i*} = *1* − *p*
⟨*proof*⟩

Now we can express the expected value of a single coin toss.

lemma *integral-toss*:
assumes *0* ≤ *p p* ≤ *1*
shows *expectation p* (*toss n*) = *2* ∗ *p* − *1*

⟨*proof*⟩

Now, we show that the tosses are independent from one another.

lemma *indep-vars-toss*:
  assumes *0* ≤ *p p* ≤ *1*
  shows *indep-vars p* (λ-. *borel*) *toss* {*0*..}
⟨*proof*⟩

The fortune of a player is a martingale (resp. sub- or supermartingale) with respect to the filtration generated by the coin tosses.

theorem *fortune-martingale*:
  assumes *p* = *1/2*
  shows *nat-martingale* (*bernoulli-stream p*) (*toss-filtration p*) *fortune*
  ⟨*proof*⟩

theorem *fortune-submartingale*:
  assumes *1/2* ≤ *p p* ≤ *1*
  shows *nat-submartingale* (*bernoulli-stream p*) (*toss-filtration p*) *fortune*
⟨*proof*⟩

theorem *fortune-supermartingale*:
  assumes *0* ≤ *p p* ≤ *1/2*
  shows *nat-supermartingale* (*bernoulli-stream p*) (*toss-filtration p*) *fortune*
⟨*proof*⟩

end

References

[1] R. Engelking. *General Topology*. Sigma series in pure mathematics. Heldermann, 1989.

[2] T. Hytönen, J. v. Neerven, M. Veraar, and L. Weis. *Analysis in Banach Spaces Volume I: Martingales and Littlewood-Paley theory*. Springer International Publishing, 2016.

[3] A. Keskin. A formalization of martingales in Isabelle/HOL. 2023.

[4] S. Lang. *Real and Functional Analysis*. Springer, 1993.

[5] G. Zitkovic. Lecture notes on conditional expectation, theory of probability I, UT Austin, Jan 2015. https://web.ma.utexas.edu/users/gordanz/notes/conditional_expectation.pdf.