

Doob's Upcrossing Inequality and Martingale Convergence Theorem

Ata Keskin

June 17, 2024

Abstract

In this entry, we formalize Doob's upcrossing inequality and subsequently prove Doob's first martingale convergence theorem. The upcrossing inequality is a fundamental result in the study of martingales. It provides a bound on the expected number of times a submartingale crosses a certain threshold within a given interval. Doob's martingale convergence theorem states that, if we have a submartingale where the supremum over the mean of the positive parts is finite, then the limit process exists almost surely and is integrable. Equivalent statements for martingales and supermartingales are also provided as corollaries.

The proofs provided are based mostly on the formalization done in the Lean mathematical library [1, 2].

Contents

1	Introduction	2
2	Stopping Times and Hitting Times	4
2.1	Stopping Time	4
2.2	σ -algebra of a Stopping Time	6
2.3	Hitting Time	11
3	Doob's Upcrossing Inequality and Martingale Convergence Theorems	16
3.1	Upcrossings and Downcrossings	17
3.2	Doob's Upcrossing Inequality	27
4	Doob's First Martingale Convergence Theorem	33

1 Introduction

Martingales, in the context of stochastic processes, are encountered in various real-world scenarios where outcomes are influenced by past events but are not entirely predictable due to randomness or uncertainty. A martingale is a stochastic process in which the expected value of the next observation, given all past observations, is equal to the current observation.

One real-world example can be encountered in environmental monitoring, particularly in the study of river flow rates. Consider a hydrologist tasked with monitoring the flow rate of a river to understand its behavior over time. The flow rate of a river is influenced by various factors such as rainfall, snowmelt, groundwater levels, and human activities like dam releases or water diversions. These factors contribute to the variability and unpredictability of the flow rate. In this scenario, the flow rate of the river can be modeled as a martingale. The flow rate at any given time is influenced by past events but is not entirely predictable due to the random nature of rainfall and other factors.

One concept that comes up frequently in the study of martingales are upcrossings and downcrossings. Upcrossings and downcrossings are random variables representing when the value of a stochastic process leaves a fixed interval. Specifically, an upcrossing occurs when the process moves from below the lower bound of the interval to above the upper bound [4], indicating a potential upward trend or positive movement. Conversely, a downcrossing happens when the process crosses below the lower bound of the interval, suggesting a potential downward trend or negative movement. By analyzing the frequency and timing of these crossings, researchers can infer information about the underlying dynamics of the process and detect shifts in its behavior.

For instance, consider tracking the movement of a stock price over time. The process representing the stock's price might cross above a certain threshold (upcrossing) or below it (downcrossing) multiple times during a trading session. The number of such crossings provides insights into the volatility and the trend of the stock.

Doob's upcrossing inequality is a fundamental result in the study of martingales. It provides a bound on the expected number of upcrossings a submartingale undertakes before some point in time.

Let's consider our example concerning river flow rates again. In this context, upcrossings represent instances where the flow rate of the river rises above a certain threshold. For example, the flow rate might cross a threshold indicating flood risk. Downcrossings, on the other hand, represent instances where the flow rate decreases below a certain threshold. This could indicate drought conditions or low-flow periods.

Doob's first martingale convergence theorem gives sufficient conditions for a submartingale to converge to a random variable almost surely. The proof is based on controlling the rate of growth or fluctuations of the submartingale,

which is where the *upcrossing inequality* comes into play. By bounding these fluctuations, we can ensure that the submartingale does not exhibit wild behavior or grow too quickly, which is essential for proving convergence.

Formally, the convergence theorem states that, if $(M_n)_{n \geq 0}$ is a submartingale with $\sup_n \mathbb{E}[M_n^+] < \infty$, where M_n^+ denotes the positive part of M_n , then the limit process $M_\infty := \lim_n M_n$ exists almost surely and is integrable. Furthermore, the limit process is measurable with respect to the smallest σ -algebra containing all of the σ -algebras in the filtration. In our formalization, we also show equivalent convergence statements for martingales and supermartingales. The theorem can be used to easily show convergence results for simple scenarios.

Consider the following example: Imagine a casino game where a player bets on the outcome of a random coin toss, where the coin comes up heads with odds $p \in [0, \frac{1}{2})$. Assume that the player goes bust when they have no money remaining. The player's wealth over time can be modeled as a supermartingale, where the value of their wealth at each time step depends only on the outcome of the previous coin toss. Doob's martingale convergence theorem assures us that the player will go bankrupt as the number of coin tosses increases.

The theorem that we have described here and formalized in the scope of our project is called *Doob's first martingale convergence theorem*. It is important to note that the convergence in this theorem is pointwise, not uniform, and is unrelated to convergence in mean square, or indeed in any L^p space. In order to obtain convergence in L^1 (i.e., convergence in mean), one requires uniform integrability of the random variables. In this form, the theorem is called *Doob's second martingale convergence theorem*. Since uniform integrability is not yet formalized in Isabelle/HOL, we have decided to confine our formalization to the first convergence theorem only.

2 Stopping Times and Hitting Times

In this section we formalize stopping times and hitting times. A stopping time is a random variable that represents the time at which a certain event occurs within a stochastic process. A hitting time, also known as first passage time or first hitting time, is a specific type of stopping time that represents the first time a stochastic process reaches a particular state or crosses a certain threshold.

```
theory Stopping-Time
imports Martingales.Stochastic-Process
begin
```

2.1 Stopping Time

The formalization of stopping times here is simply a rewrite of the document *HOL-Probability.Stopping-Time* [5]. We have adapted the document to use the locales defined in our formalization of filtered measure spaces [6] [7]. This way, we can omit the partial formalization of filtrations in the original document. Furthermore, we can include the initial time index t_0 that we introduced as well.

```
context linearly-filtered-measure
begin
```

— A stopping time is a measurable function from the measure space (possible events) into the time axis.

```
definition stopping-time :: ('a  $\Rightarrow$  'b)  $\Rightarrow$  bool where
  stopping-time T = ((T  $\in$  space M  $\rightarrow$  {t0..})  $\wedge$  ( $\forall$  t $\geq$ t0. Measurable.pred (F t)
  ( $\lambda$ x. T x  $\leq$  t)))
```

```
lemma stopping-time-cong:
```

```
  assumes  $\bigwedge$  t x. t  $\geq$  t0  $\implies$  x  $\in$  space (F t)  $\implies$  T x = S x
```

```
  shows stopping-time T = stopping-time S
```

```
proof (cases T  $\in$  space M  $\rightarrow$  {t0..})
```

```
  case True
```

```
    hence S  $\in$  space M  $\rightarrow$  {t0..} using assms space-F by force
```

```
    then show ?thesis unfolding stopping-time-def
```

```
      using assms arg-cong[where f=( $\lambda$ P.  $\forall$  t $\geq$ t0. P t)] measurable-cong[where
M=F - and f= $\lambda$ x. T x  $\leq$  - and g= $\lambda$ x. S x  $\leq$  -] True by metis
```

```
  next
```

```
    case False
```

```
      hence S  $\notin$  space M  $\rightarrow$  {t0..} using assms space-F by force
```

```
      then show ?thesis unfolding stopping-time-def using False by blast
```

```
qed
```

```
lemma stopping-time-ge-zero:
```

```
  assumes stopping-time T  $\omega \in$  space M
```

shows $T \omega \geq t_0$
using *assms* **unfolding** *stopping-time-def* **by** *auto*

lemma *stopping-timeD*:
assumes *stopping-time* $T t \geq t_0$
shows *Measurable.pred* ($F t$) ($\lambda x. T x \leq t$)
using *assms* **unfolding** *stopping-time-def* **by** *simp*

lemma *stopping-timeI*[*intro?*]:
assumes $\bigwedge x. x \in \text{space } M \implies T x \geq t_0$
 $(\bigwedge t. t \geq t_0 \implies \text{Measurable.pred } (F t) (\lambda x. T x \leq t))$
shows *stopping-time* T
using *assms* **by** (*auto simp: stopping-time-def*)

lemma *stopping-time-measurable*:
assumes *stopping-time* T
shows $T \in \text{borel-measurable } M$
proof (*rule borel-measurableI-le*)

- {
- fix** t **assume** $\neg t \geq t_0$
- hence** $\{x \in \text{space } M. T x \leq t\} = \{\}$ **using** *assms* *dual-order.trans*[*of - t t₀*]
- unfolding** *stopping-time-def* **by** *fastforce*
- hence** $\{x \in \text{space } M. T x \leq t\} \in \text{sets } M$ **by** (*metis sets.empty-sets*)
- }
- moreover**
- {
- fix** t **assume** *asm*: $t \geq t_0$
- hence** $\{x \in \text{space } M. T x \leq t\} \in \text{sets } M$ **using** *stopping-timeD*[*OF assms asm*]
- sets-F-subset* **unfolding** *Measurable.pred-def* *space-F*[*OF asm*] **by** *blast*
- }
- ultimately show** $\{x \in \text{space } M. T x \leq t\} \in \text{sets } M$ **for** t **by** *blast*

qed

lemma *stopping-time-const*:
assumes $t \geq t_0$
shows *stopping-time* ($\lambda x. t$) **using** *assms* **by** (*auto simp: stopping-time-def*)

lemma *stopping-time-min*:
assumes *stopping-time* T *stopping-time* S
shows *stopping-time* ($\lambda x. \min (T x) (S x)$)
using *assms* **by** (*auto simp: stopping-time-def min-le-iff-disj intro!: pred-intros-logic*)

lemma *stopping-time-max*:
assumes *stopping-time* T *stopping-time* S
shows *stopping-time* ($\lambda x. \max (T x) (S x)$)
using *assms* **by** (*auto simp: stopping-time-def intro!: pred-intros-logic max.coboundedI1*)

2.2 σ -algebra of a Stopping Time

Moving on, we define the σ -algebra associated with a stopping time T . It contains all the information up to time T , the same way $F t$ contains all the information up to time t .

definition *pre-sigma* :: ('a \Rightarrow 'b) \Rightarrow 'a *measure* **where**
pre-sigma $T = \text{sigma} (\text{space } M) \{A \in \text{sets } M. \forall t \geq t_0. \{\omega \in A. T \omega \leq t\} \in \text{sets } (F t)\}$

lemma *measure-pre-sigma[simp]*: $\text{emeasure} (\text{pre-sigma } T) = (\lambda \cdot. 0)$ **by** (*simp add: pre-sigma-def emeasure-sigma*)

lemma *sigma-algebra-pre-sigma*:

assumes *stopping-time* T

shows *sigma-algebra* ($\text{space } M$) $\{A \in \text{sets } M. \forall t \geq t_0. \{\omega \in A. T \omega \leq t\} \in \text{sets } (F t)\}$

proof –

let $\text{?}\Sigma = \{A \in \text{sets } M. \forall t \geq t_0. \{\omega \in A. T \omega \leq t\} \in \text{sets } (F t)\}$

{

fix A **assume** *asm*: $A \in \text{?}\Sigma$

{

fix t **assume** *asm'*: $t \geq t_0$

hence $\{\omega \in A. T \omega \leq t\} \in \text{sets } (F t)$ **using** *asm* **by** *blast*

then have $\{\omega \in \text{space } (F t). T \omega \leq t\} - \{\omega \in A. T \omega \leq t\} \in \text{sets } (F t)$

using *assms[THEN stopping-timeD, OF asm']* **by** *auto*

also have $\{\omega \in \text{space } (F t). T \omega \leq t\} - \{\omega \in A. T \omega \leq t\} = \{\omega \in \text{space } M - A. T \omega \leq t\}$ **using** *space-F[OF asm']* **by** *blast*

finally have $\{\omega \in (\text{space } M) - A. T \omega \leq t\} \in \text{sets } (F t)$.

}

hence $\text{space } M - A \in \text{?}\Sigma$ **using** *asm* **by** *blast*

}

moreover

{

fix A :: *nat* \Rightarrow 'a *set* **assume** *asm*: $\text{range } A \subseteq \text{?}\Sigma$

{

fix t **assume** $t \geq t_0$

then have $(\bigcup i. \{\omega \in A i. T \omega \leq t\}) \in \text{sets } (F t)$ **using** *asm* **by** *auto*

also have $(\bigcup i. \{\omega \in A i. T \omega \leq t\}) = \{\omega \in \bigcup (A \text{ ' UNIV}). T \omega \leq t\}$ **by**

auto

finally have $\{\omega \in \bigcup (\text{range } A). T \omega \leq t\} \in \text{sets } (F t)$.

}

hence $\bigcup (\text{range } A) \in \text{?}\Sigma$ **using** *asm* **by** *blast*

}

ultimately show *?thesis unfolding sigma-algebra-iff2* **by** (*auto intro!: sets.sets-into-space[THEN PowI, THEN subsetI]*)

qed

lemma *space-pre-sigma[simp]*: $\text{space} (\text{pre-sigma } T) = \text{space } M$ **unfolding** *pre-sigma-def* **by** (*intro space-measure-of-conv*)

lemma *sets-pre-sigma*:
assumes *stopping-time T*
shows $\text{sets } (\text{pre-sigma } T) = \{A \in \text{sets } M. \forall t \geq t_0. \{\omega \in A. T \omega \leq t\} \in F t\}$
unfolding *pre-sigma-def* **using** *sigma-algebra.sets-measure-of-eq* [*OF sigma-algebra-pre-sigma, OF assms*] **by** *blast*

lemma *sets-pre-sigmaI*:
assumes *stopping-time T*
and $\bigwedge t. t \geq t_0 \implies \{\omega \in A. T \omega \leq t\} \in F t$
shows $A \in \text{pre-sigma } T$
proof (*cases* $\exists t \geq t_0. \forall t'. t' \leq t$)
case *True*
then obtain t **where** $t \geq t_0$ $\{\omega \in A. T \omega \leq t\} = A$ **by** *blast*
hence $A \in M$ **using** *assms(2)* [*of t*] *sets-F-subset* [*of t*] **by** *fastforce*
thus *?thesis* **using** *assms(2)* **unfolding** *sets-pre-sigma* [*OF assms(1)*] **by** *blast*
next
case *False*
hence $*$: $\{t < ..\} \neq \{\}$ **if** $t \geq t_0$ **for** t **by** (*metis not-le empty-iff greaterThan-iff*)
obtain $D :: 'b \text{ set}$ **where** D : *countable* $D \wedge X$. *open* $X \implies X \neq \{\} \implies D \cap X \neq \{\}$ **by** (*metis countable-dense-setE disjoint-iff*)
have $**$: $D \cap \{t < ..\} \neq \{\}$ **if** $t \geq t_0$ **for** t **using** $*$ **that** **by** (*intro D(2)*) *auto*
{
fix ω
obtain t **where** $t: t \geq t_0$ $T \omega \leq t$ **using** *linorder-linear* **by** *auto*
moreover obtain t' **where** $t' \in D \cap \{t < ..\} \cap \{t_0.. \}$ **using** $**$ t **by** *fastforce*
moreover have $T \omega \leq t'$ **using** *calculation* **by** *fastforce*
ultimately have $\exists t. \exists t' \in D \cap \{t < ..\} \cap \{t_0.. \}. T \omega \leq t'$ **by** *blast*
}
hence $(\bigcup t' \in (\bigcup t. D \cap \{t < ..\} \cap \{t_0.. \}). \{\omega \in A. T \omega \leq t'\}) = A$ **by** *fast*
moreover have $(\bigcup t' \in (\bigcup t. D \cap \{t < ..\} \cap \{t_0.. \}). \{\omega \in A. T \omega \leq t'\}) \in M$
using D *assms(2)* *sets-F-subset* **by** (*intro sets.countable-UN''*, *fastforce*, *fast*)
ultimately have $A \in M$ **by** *argo*
thus *?thesis* **using** *assms(2)* **unfolding** *sets-pre-sigma* [*OF assms(1)*] **by** *blast*
qed

lemma *pred-pre-sigmaI*:
assumes *stopping-time T*
shows $(\bigwedge t. t \geq t_0 \implies \text{Measurable.pred } (F t) (\lambda \omega. P \omega \wedge T \omega \leq t)) \implies \text{Measurable.pred } (\text{pre-sigma } T) P$
unfolding *pred-def space-pre-sigma* **using** *assms* **by** (*auto intro: sets-pre-sigmaI* [*OF assms(1)*])

lemma *sets-pre-sigmaD*:
assumes *stopping-time T* $A \in \text{pre-sigma } T$ $t \geq t_0$
shows $\{\omega \in A. T \omega \leq t\} \in \text{sets } (F t)$
using *assms* *sets-pre-sigma* **by** *auto*

lemma *borel-measurable-stopping-time-pre-sigma*:

assumes *stopping-time* T
shows $T \in \text{borel-measurable}$ (*pre-sigma* T)
proof (*intro borel-measurableI-le sets-pre-sigmaI[OF assms]*)
fix $t\ t'$ **assume** *asm*: $t \geq t_0$
{
 assume $\neg t' \geq t_0$
 hence $\{\omega \in \{x \in \text{space } (\text{pre-sigma } T). T x \leq t'\}. T \omega \leq t\} = \{\}$ **using** *assms*
dual-order.trans[of - t' t₀] **unfolding** *stopping-time-def* **by** *fastforce*
 hence $\{\omega \in \{x \in \text{space } (\text{pre-sigma } T). T x \leq t'\}. T \omega \leq t\} \in \text{sets } (F t)$ **by**
(metis sets.empty-sets)
}
moreover
{
 assume *asm'*: $t' \geq t_0$
 have $\{\omega \in \text{space } (F (\min t' t)). T \omega \leq \min t' t\} \in \text{sets } (F (\min t' t))$
 using *assms asm asm'* **unfolding** *pred-def[symmetric]* **by** (*intro stopping-timeD*) *auto*
 also have $\dots \subseteq \text{sets } (F t)$
 using *assms asm asm'* **by** (*intro sets-F-mono*) *auto*
 finally have $\{\omega \in \{x \in \text{space } (\text{pre-sigma } T). T x \leq t'\}. T \omega \leq t\} \in \text{sets } (F t)$
 using *asm asm'* **by** *simp*
}
ultimately show $\{\omega \in \{x \in \text{space } (\text{pre-sigma } T). T x \leq t'\}. T \omega \leq t\} \in \text{sets}$
 $(F t)$ **by** *blast*
qed

lemma *mono-pre-sigma*:

assumes *stopping-time* T *stopping-time* S
and $\bigwedge x. x \in \text{space } M \implies T x \leq S x$
shows *pre-sigma* $T \subseteq \text{pre-sigma } S$
proof
fix A **assume** $A \in \text{pre-sigma } T$
hence *asm*: $A \in \text{sets } M\ t \geq t_0 \implies \{\omega \in A. T \omega \leq t\} \in \text{sets } (F t)$ **for** t **using**
assms sets-pre-sigma **by** *blast+*
{
 fix t **assume** *asm'*: $t \geq t_0$
 then have $A \subseteq \text{space } M$ **using** *sets.sets-into-space asm* **by** *blast*
 have $\{\omega \in A. T \omega \leq t\} \cap \{\omega \in \text{space } (F t). S \omega \leq t\} \in \text{sets } (F t)$
 using *asm asm' stopping-timeD[OF assms(2)]* **by** (*auto simp: pred-def*)
 also have $\{\omega \in A. T \omega \leq t\} \cap \{\omega \in \text{space } (F t). S \omega \leq t\} = \{\omega \in A. S \omega \leq t\}$
 using *sets.sets-into-space[OF asm(1)] assms(3) order-trans asm'* **by** *fastforce*
 finally have $\{\omega \in A. S \omega \leq t\} \in \text{sets } (F t)$ **by** *simp*
}
thus $A \in \text{pre-sigma } S$ **by** (*intro sets-pre-sigmaI assms asm*) *blast*
qed

lemma *stopping-time-measurable-le*:

assumes *stopping-time* T $s \geq t_0$ $t \geq s$
shows *Measurable.pred* $(F t)$ $(\lambda \omega. T \omega \leq s)$

using *assms stopping-timeD*[of T] *sets-F-mono*[of - t] by (*auto simp: pred-def*)

lemma *stopping-time-measurable-less*:
assumes *stopping-time* $T s \geq t_0 t \geq s$
shows *Measurable.pred* ($F t$) ($\lambda\omega. T \omega < s$)
proof –
have *Measurable.pred* ($F t$) ($\lambda\omega. T \omega < t$) **if** *asm: stopping-time* $T t \geq t_0$ **for** $T t$
proof –
obtain $D :: 'b$ set **where** D : *countable* $D \wedge X$. *open* $X \implies X \neq \{\}$ $\implies D \cap X \neq \{\}$ **by** (*metis countable-dense-setE disjoint-iff*)
show *?thesis*
proof *cases*
assume *: $\forall t' \in \{t_0..<t\}. \{t'<..<t\} \neq \{\}$
hence **: $D \cap \{t'<..<t\} \neq \{\}$ **if** $t' \in \{t_0..<t\}$ **for** t' **using** *that* **by** (*intro D(2) fastforce+*)

show *?thesis*
proof (*rule measurable-cong[THEN iffD2]*)
show $T \omega < t \longleftrightarrow (\exists r \in D \cap \{t_0..<t\}. T \omega \leq r)$ **if** $\omega \in \text{space } (F t)$ **for** ω
using **[of $T \omega$] *that less-imp-le stopping-time-ge-zero asm* **by** *fastforce*
show *Measurable.pred* ($F t$) ($\lambda\omega. \exists r \in D \cap \{t_0..<t\}. T \omega \leq r$)
using *stopping-time-measurable-le asm D* **by** (*intro measurable-pred-countable*)
auto
qed
next
assume $\neg (\forall t' \in \{t_0..<t\}. \{t'<..<t\} \neq \{\})$
then obtain t' **where** $t' \in \{t_0..<t\} \{t'<..<t\} = \{\}$ **by** *blast*
show *?thesis*
proof (*rule measurable-cong[THEN iffD2]*)
show $T \omega < t \longleftrightarrow T \omega \leq t'$ **for** ω **using** t' **by** (*metis atLeastLessThan-iff emptyE greaterThanLessThan-iff linorder-not-less order.strict-trans1*)
show *Measurable.pred* ($F t$) ($\lambda\omega. T \omega \leq t'$) **using** t' **by** (*intro stopping-time-measurable-le[OF asm(1)] auto*)
qed
qed
thus *?thesis*
using *assms sets-F-mono*[of - t] **by** (*auto simp add: pred-def*)
qed

lemma *stopping-time-measurable-ge*:
assumes *stopping-time* $T s \geq t_0 t \geq s$
shows *Measurable.pred* ($F t$) ($\lambda\omega. T \omega \geq s$)
by (*auto simp: not-less[symmetric] intro: stopping-time-measurable-less[OF assms] Measurable.pred-intros-logic*)

lemma *stopping-time-measurable-gr*:
assumes *stopping-time* $T s \geq t_0 t \geq s$
shows *Measurable.pred* ($F t$) ($\lambda x. s < T x$)

by (*auto simp add: not-le[symmetric] intro: stopping-time-measurable-le[OF assms] Measurable.pred-intros-logic*)

lemma *stopping-time-measurable-eq*:

assumes *stopping-time* $T s \geq t_0 t \geq s$

shows *Measurable.pred* ($F t$) ($\lambda\omega. T \omega = s$)

unfolding *eq-iff* **using** *stopping-time-measurable-le[OF assms] stopping-time-measurable-ge[OF assms]*

by (*intro pred-intros-logic*)

lemma *stopping-time-less-stopping-time*:

assumes *stopping-time* T *stopping-time* S

shows *Measurable.pred* (*pre-sigma* T) ($\lambda\omega. T \omega < S \omega$)

proof (*rule pred-pre-sigmaI*)

fix t **assume** *asm*: $t \geq t_0$

obtain $D :: 'b$ **set** **where** D : *countable* D **and** *semidense-D*: $\bigwedge x y. x < y \implies (\exists b \in D. x \leq b \wedge b < y)$

using *countable-separating-set-linorder2* **by** *auto*

show *Measurable.pred* ($F t$) ($\lambda\omega. T \omega < S \omega \wedge T \omega \leq t$)

proof (*rule measurable-cong[THEN iffD2]*)

let $?f = \lambda\omega. \text{if } T \omega = t \text{ then } \neg S \omega \leq t \text{ else } \exists s \in D \cap \{t_0..t\}. T \omega \leq s \wedge \neg (S \omega \leq s)$

{

fix ω **assume** $\omega \in \text{space } (F t)$ $T \omega \leq t$ $T \omega \neq t$ $T \omega < S \omega$

hence $t_0 \leq T \omega$ $T \omega < \min t (S \omega)$ **using** *asm stopping-time-ge-zero[OF assms(1)]* **by** *auto*

then obtain r **where** $r \in D$ $t_0 \leq r$ $T \omega \leq r$ $r < \min t (S \omega)$ **using** *semidense-D order-trans* **by** *blast*

hence $\exists s \in D \cap \{t_0..t\}. T \omega \leq s \wedge s < S \omega$ **by** *auto*

}

thus $(T \omega < S \omega \wedge T \omega \leq t) = ?f \omega$ **if** $\omega \in \text{space } (F t)$ **for** ω **using** *that* **by** *force*

show *Measurable.pred* ($F t$) $?f$

using *assms asm D*

by (*intro pred-intros-logic measurable-If measurable-pred-countable countable-Collect*

stopping-time-measurable-le predE stopping-time-measurable-eq) *auto*

qed

qed (*intro assms*)

end

lemma (*in enat-filtered-measure*) *stopping-time-SUP-enat*:

fixes $T :: \text{nat} \Rightarrow ('a \Rightarrow \text{enat})$

shows $(\bigwedge i. \text{stopping-time } (T i)) \implies \text{stopping-time } (SUP i. T i)$

unfolding *stopping-time-def SUP-apply SUP-le-iff* **by** (*auto intro!: pred-intros-countable*)

lemma (*in enat-filtered-measure*) *stopping-time-Inf-enat*:

assumes $\bigwedge i. \text{Measurable.pred } (F i) (P i)$

shows *stopping-time* $(\lambda\omega. \text{Inf } \{i. P i \omega\})$
proof –
{
 fix $t :: \text{enat}$ **assume** $asm: t \neq \infty$
 moreover
 {
 fix $i \omega$ **assume** $\text{Inf } \{i. P i \omega\} \leq t$
 moreover **have** $a < eSuc b \iff (a \leq b \wedge a \neq \infty)$ **for** $a b$ **by** *(cases a) auto*
 ultimately **have** $(\exists i \leq t. P i \omega)$ **using** asm **unfolding** *Inf-le-iff* **by** *(cases t)*
(auto elim!: allE[of - eSuc t])
 }
 ultimately **have** $*$: $\bigwedge\omega. \text{Inf } \{i. P i \omega\} \leq t \iff (\exists i \in \{..t\}. P i \omega)$ **by** *(auto intro!: Inf-lower2)*
 have $\text{Measurable.pred } (F t)$ $(\lambda\omega. \text{Inf } \{i. P i \omega\} \leq t)$ **unfolding** $*$ **using**
sets-F-mono assms **by** *(intro pred-intros-countable-bounded) (auto simp: pred-def)*
 }
 moreover **have** $\text{Measurable.pred } (F t)$ $(\lambda\omega. \text{Inf } \{i. P i \omega\} \leq t)$ **if** $t = \infty$ **for** t
using *that* **by** *simp*
 ultimately **show** *?thesis* **by** *(blast intro: stopping-timeI[OF i0-lb])*
qed

lemma *(in nat-filtered-measure) stopping-time-Inf-nat:*
assumes $\bigwedge i. \text{Measurable.pred } (F i)$ $(P i)$
 $\bigwedge i \omega. \omega \in \text{space } M \implies \exists n. P n \omega$
shows *stopping-time* $(\lambda\omega. \text{Inf } \{i. P i \omega\})$
proof *(rule stopping-time-cong[THEN iffD2])*
show *stopping-time* $(\lambda x. \text{LEAST } n. P n x)$
proof
 fix t
 have $((\text{LEAST } n. P n \omega) \leq t) = (\exists i \leq t. P i \omega)$ **if** $\omega \in \text{space } M$ **for** ω **by** *(rule*
LeastI2-wellorder-ex[OF assms(2)[OF that]]) auto
 moreover **have** $\text{Measurable.pred } (F t)$ $(\lambda\omega. \exists i \in \{..t\}. P i \omega)$ **using** *sets-F-mono[of*
- t] assms **by** *(intro pred-intros-countable-bounded) (auto simp: pred-def)*
 ultimately **show** $\text{Measurable.pred } (F t)$ $(\lambda\omega. (\text{LEAST } n. P n \omega) \leq t)$ **by** *(subst*
measurable-cong[of F t]) auto
 qed *(simp)*
qed *(simp add: Inf-nat-def)*

definition *stopped-value* $:: ('b \Rightarrow 'a \Rightarrow 'c) \Rightarrow ('a \Rightarrow 'b) \Rightarrow ('a \Rightarrow 'c)$ **where**
stopped-value $X \tau \omega = X (\tau \omega) \omega$

2.3 Hitting Time

Given a stochastic process X and a borel set A , *hitting-time* $X A s t$ is the first time X is in A after time s and before time t . If X does not hit A after time s and before t then the hitting time is simply t . The definition presented here coincides with the definition of hitting times in mathlib [1].

context *linearly-filtered-measure*

begin

definition *hitting-time* :: ('b \Rightarrow 'a \Rightarrow 'c) \Rightarrow 'c set \Rightarrow 'b \Rightarrow 'b \Rightarrow ('a \Rightarrow 'b) **where**
hitting-time X A s t = ($\lambda\omega$. if $\exists i \in \{s..t\} \cap \{t_0..\}$. X i $\omega \in A$ then Inf ($\{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\}$) else max t₀ t)

lemma *hitting-time-def'*:

hitting-time X A s t = ($\lambda\omega$. Inf (insert (max t₀ t) ($\{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\}$)))

proof *cases*

assume *asm*: t₀ \leq s \wedge s \leq t

hence $\{s..t\} \cap \{t_0..\} = \{s..t\}$ **by** *simp*

{

fix ω

assume *: $\{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\} \neq \{\}$

then obtain i **where** i $\in \{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\}$ **by** *blast*

hence Inf ($\{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\}$) \leq t **by** (*intro cInf-lower*[of i,

THEN order-trans]) *auto*

hence Inf (insert (max t₀ t) ($\{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\}$)) = Inf ($\{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\}$) **using** *asm* * *inf-absorb2* **by** (*subst cInf-insert-If*) *force+*

also have ... = *hitting-time* X A s t ω **using** * **unfolding** *hitting-time-def* **by**

auto

finally have *hitting-time* X A s t $\omega =$ Inf (insert (max t₀ t) ($\{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\}$)) **by** *argo*

}

moreover

{

fix ω

assume $\{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\} = \{\}$

hence *hitting-time* X A s t $\omega =$ Inf (insert (max t₀ t) ($\{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\}$)) **unfolding** *hitting-time-def* **by** *auto*

}

ultimately show *?thesis* **by** *fast*

next

assume $\neg (t_0 \leq s \wedge s \leq t)$

moreover

{

assume *asm*: s < t₀ t \geq t₀

hence $\{s..t\} \cap \{t_0..\} = \{t_0..t\}$ **by** *simp*

{

fix ω

assume *: $\{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\} \neq \{\}$

then obtain i **where** i $\in \{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\}$ **by** *blast*

hence Inf ($\{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\}$) \leq t **by** (*intro cInf-lower*[of i,

THEN order-trans]) *auto*

hence Inf (insert (max t₀ t) ($\{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\}$)) = Inf ($\{s..t\} \cap \{t_0..\} \cap \{i$. X i $\omega \in A\}$) **using** *asm* * *inf-absorb2* **by** (*subst cInf-insert-If*) *force+*

also have ... = *hitting-time* X A s t ω **using** * **unfolding** *hitting-time-def*

by *auto*
finally have *hitting-time* $X A s t \omega = \text{Inf} (\text{insert} (\text{max } t_0 t) (\{s..t\} \cap \{t_0..\} \cap \{i. X i \omega \in A\}))$ **by** *argo*
}
moreover
{
fix ω
assume $\{s..t\} \cap \{t_0..\} \cap \{i. X i \omega \in A\} = \{\}$
hence *hitting-time* $X A s t \omega = \text{Inf} (\text{insert} (\text{max } t_0 t) (\{s..t\} \cap \{t_0..\} \cap \{i. X i \omega \in A\}))$ **unfolding** *hitting-time-def* **by** *auto*
}
ultimately have *?thesis* **by** *fast*
}
moreover have *?thesis* **if** $s < t_0$ $t < t_0$ **using** *that* **unfolding** *hitting-time-def* **by** *auto*
moreover have *?thesis* **if** $s > t$ **using** *that* **unfolding** *hitting-time-def* **by** *auto*
ultimately show *?thesis* **by** *fastforce*
qed

— The following lemma provides a sufficient condition for an injective function to preserve a hitting time.

lemma *hitting-time-inj-on*:
assumes *inj-on* $f S \wedge \omega t. t \geq t_0 \implies X t \omega \in S A \subseteq S$
shows *hitting-time* $X A = \text{hitting-time} (\lambda t \omega. f (X t \omega)) (f ' A)$
proof —
have $X t \omega \in A \longleftrightarrow f (X t \omega) \in f ' A$ **if** $t \geq t_0$ **for** $t \omega$ **using** *assms that inj-on-image-mem-iff* **by** *meson*
hence $\{t_0..\} \cap \{i. X i \omega \in A\} = \{t_0..\} \cap \{i. f (X i \omega) \in f ' A\}$ **for** ω **by** *blast*
thus *?thesis* **unfolding** *hitting-time-def'* *Int-assoc* **by** *presburger*
qed

lemma *hitting-time-translate*:
fixes $c :: - :: \text{ab-group-add}$
shows *hitting-time* $X A = \text{hitting-time} (\lambda n \omega. X n \omega + c) (((+) c) ' A)$
by (*subst hitting-time-inj-on[OF inj-on-add, of - UNIV]*) (*simp add: add.commute*)+

lemma *hitting-time-le*:
assumes $t \geq t_0$
shows *hitting-time* $X A s t \omega \leq t$
unfolding *hitting-time-def'* **using** *assms*
by (*intro cInf-lower[of max t_0 t, THEN order-trans]*) *auto*

lemma *hitting-time-ge*:
assumes $t \geq t_0$ $s \leq t$
shows $s \leq \text{hitting-time } X A s t \omega$
unfolding *hitting-time-def'* **using** *assms*
by (*intro le-cInf-iff[THEN iffD2]*) *auto*

lemma *hitting-time-mono*:
assumes $t \geq t_0$ $s \leq s'$ $t \leq t'$
shows *hitting-time* $X A s t \omega \leq$ *hitting-time* $X A s' t' \omega$
unfolding *hitting-time-def'* **using** *assms* **by** (*fastforce intro!*: *cInf-mono*)

end

context *nat-filtered-measure*
begin

— Hitting times are stopping times for adapted processes.

lemma *stopping-time-hitting-time*:
assumes *adapted-process* $M F 0 X A \in$ *borel*
shows *stopping-time* (*hitting-time* $X A s t$)
proof –
interpret *adapted-process* $M F 0 X$ **by** (*rule assms*)
have *insert* t ($\{s..t\} \cap \{i. X i \omega \in A\}$) = $\{i. i = t \vee i \in (\{s..t\} \cap \{i. X i \omega \in A\})\}$ **for** ω **by** *blast*
hence *hitting-time* $X A s t = (\lambda\omega. \text{Inf } \{i. i = t \vee i \in (\{s..t\} \cap \{i. X i \omega \in A\})\})$
unfolding *hitting-time-def'* **by** *simp*
thus *?thesis* **using** *assms* **by** (*auto intro: stopping-time-Inf-nat*)
qed

lemma *stopping-time-hitting-time'*:
assumes *adapted-process* $M F 0 X A \in$ *borel* *stopping-time* $s \wedge \omega. s \omega \leq t$
shows *stopping-time* $(\lambda\omega. \text{hitting-time } X A (s \omega) t \omega)$
proof –
interpret *adapted-process* $M F 0 X$ **by** (*rule assms*)
{
fix n
have $s \omega \leq$ *hitting-time* $X A (s \omega) t \omega$ **if** $s \omega > n$ **for** ω **using** *hitting-time-ge[OF*
- assms(4)] **by** *simp*
hence $(\bigcup_{i \in \{n<..\}}. \{\omega. s \omega = i\} \cap \{\omega. \text{hitting-time } X A i t \omega \leq n\}) = \{\}$ **by**
fastforce
hence $*$: $(\lambda\omega. \text{hitting-time } X A (s \omega) t \omega \leq n) = (\lambda\omega. \exists i \leq n. s \omega = i \wedge$
hitting-time $X A i t \omega \leq n)$ **by** *force*

have *Measurable.pred* $(F n)$ $(\lambda\omega. s \omega = i \wedge \text{hitting-time } X A i t \omega \leq n)$ **if** $i \leq$
 n **for** i

proof –
have *Measurable.pred* $(F i)$ $(\lambda\omega. s \omega = i)$ **using** *stopping-time-measurable-eq*
assms **by** *blast*
hence *Measurable.pred* $(F n)$ $(\lambda\omega. s \omega = i)$ **by** (*meson less-eq-nat.simps*
measurable-from-subalg subalgebra-F that)
moreover **have** *Measurable.pred* $(F n)$ $(\lambda\omega. \text{hitting-time } X A i t \omega \leq n)$
using *stopping-timeD[OF stopping-time-hitting-time, OF assms(1,2)]* **by** *blast*
ultimately show *?thesis* **by** *auto*
qed

hence *Measurable.pred* ($F\ n$) ($\lambda\omega. \exists i \leq n. s\ \omega = i \wedge \text{hitting-time } X\ A\ i\ t\ \omega \leq n$) **by** (*intro pred-intros-countable*) *auto*
hence *Measurable.pred* ($F\ n$) ($\lambda\omega. \text{hitting-time } X\ A\ (s\ \omega)\ t\ \omega \leq n$) **using** * **by** *argo*
}
thus *?thesis* **by** (*intro stopping-timeI*) *auto*
qed

— If X hits A at time $j \in \{s..t\}$, then the stopped value of X at the hitting time of A in the interval $\{s..t\}$ is an element of A .

lemma *stopped-value-hitting-time-mem*:
assumes $j \in \{s..t\}$ $X\ j\ \omega \in A$
shows *stopped-value* X (*hitting-time* $X\ A\ s\ t$) $\omega \in A$
proof –
have $\exists i \in \{s..t\} \cap \{0..\}$. $X\ i\ \omega \in A$ **using** *assms* **by** *blast*
moreover **have** *Inf* ($\{s..t\} \cap \{i. X\ i\ \omega \in A\}$) $\in \{s..t\} \cap \{i. X\ i\ \omega \in A\}$ **using** *assms* **by** (*blast intro!*: *Inf-nat-def1*)
ultimately **show** *?thesis* **unfolding** *hitting-time-def* *stopped-value-def* **by** *simp*
qed

lemma *hitting-time-le-iff*:
assumes $i < t$
shows *hitting-time* $X\ A\ s\ t\ \omega \leq i \iff (\exists j \in \{s..i\}. X\ j\ \omega \in A)$ (**is** *?lhs = ?rhs*)
proof
assume *?lhs*
moreover **have** *hitting-time* $X\ A\ s\ t\ \omega \in \text{insert } t\ (\{s..t\} \cap \{i. X\ i\ \omega \in A\})$
by (*metis hitting-time-def' Int-atLeastAtMostR2 inf-sup-aci(1) insertI1 max-0L wellorder-InfI*)
ultimately **have** *hitting-time* $X\ A\ s\ t\ \omega \in \{s..i\} \cap \{i. X\ i\ \omega \in A\}$ **using** *assms*
by *force*
thus *?rhs* **by** *blast*
next
assume *?rhs*
then **obtain** j **where** $j \in \{s..i\}$ $X\ j\ \omega \in A$ **by** *blast*
hence *hitting-time* $X\ A\ s\ t\ \omega \leq j$ **unfolding** *hitting-time-def'* **using** *assms* **by** (*auto intro: cInf-lower*)
thus *?lhs* **using** j **by** *simp*
qed

lemma *hitting-time-less-iff*:
assumes $i \leq t$
shows *hitting-time* $X\ A\ s\ t\ \omega < i \iff (\exists j \in \{s..<i\}. X\ j\ \omega \in A)$ (**is** *?lhs = ?rhs*)
proof
assume *?lhs*
moreover **have** *hitting-time* $X\ A\ s\ t\ \omega \in \text{insert } t\ (\{s..t\} \cap \{i. X\ i\ \omega \in A\})$
by (*metis hitting-time-def' Int-atLeastAtMostR2 inf-sup-aci(1) insertI1 max-0L wellorder-InfI*)

ultimately have *hitting-time* $X A s t \omega \in \{s..<i\} \cap \{i. X i \omega \in A\}$ **using** *assms*
by force
thus *?rhs* **by** *blast*
next
assume *?rhs*
then obtain j **where** $j: j \in \{s..<i\} X j \omega \in A$ **by** *blast*
hence *hitting-time* $X A s t \omega \leq j$ **unfolding** *hitting-time-def'* **using** *assms* **by**
(auto intro: cInf-lower)
thus *?lhs* **using** j **by** *simp*
qed

— If X already hits A in the interval $\{s..t\}$, then *hitting-time* $X A s t = \text{hitting-time } X A s t'$ for $t \leq t'$.

lemma *hitting-time-eq-hitting-time*:
assumes $t \leq t' j \in \{s..t\} X j \omega \in A$
shows *hitting-time* $X A s t \omega = \text{hitting-time } X A s t' \omega$ (**is** *?lhs = ?rhs*)
proof –
have *hitting-time* $X A s t \omega \in \{s..t'\}$ **using** *hitting-time-le*[*THEN order-trans, of t t' X A s*] *hitting-time-ge*[*of t s X A*] **assms** **by** *auto*
moreover have *stopped-value* $X (\text{hitting-time } X A s t) \omega \in A$ **by** (*blast intro: stopped-value-hitting-time-mem assms*)
ultimately have *hitting-time* $X A s t' \omega \leq \text{hitting-time } X A s t \omega$ **by** (*fastforce simp add: hitting-time-def'*[**where** $t=t'$] *stopped-value-def intro!: cInf-lower*)
thus *?thesis* **by** (*blast intro: le-antisym hitting-time-mono*[*OF - order-refl assms(1)*])
qed

end

end

3 Doob's Upcrossing Inequality and Martingale Convergence Theorems

In this section we formalize upcrossings and downcrossings. Following this, we prove Doob's upcrossing inequality and first martingale convergence theorem.

theory *Upcrossing*
imports *Martingales.Martingale Stopping-Time*
begin

lemma *real-embedding-borel-measurable*: $real \in \text{borel-measurable borel}$ **by** (*auto intro: borel-measurable-continuous-onI*)

lemma *limsup-lower-bound*:

fixes $u :: \text{nat} \Rightarrow \text{ereal}$
assumes $\text{limsup } u > l$
shows $\exists N > k. u \ N > l$
proof –
have $\text{limsup } u = - \text{liminf } (\lambda n. - u \ n)$ **using** $\text{liminf-ereal-cminus}[of \ 0 \ u]$ **by** simp
hence $\text{liminf } (\lambda n. - u \ n) < - l$ **using** $\text{assms } \text{ereal-less-uminus-reorder}$ **by**
 presburger
hence $\exists N > k. - u \ N < - l$ **using** $\text{liminf-upper-bound}$ **by** blast
thus $?thesis$ **using** $\text{ereal-less-uminus-reorder}$ **by** simp
qed

lemma $\text{ereal-abs-max-min}: |c| = \text{max } 0 \ c - \text{min } 0 \ c$ **for** $c :: \text{ereal}$
by $(\text{cases } c \geq 0)$ auto

3.1 Upcrossings and Downcrossings

Given a stochastic process X , real values a and b , and some point in time N , we would like to define a notion of "upcrossings" of X across the band $\{a..b\}$ which counts the number of times any realization of X crosses from below a to above b before time N . To make this heuristic rigorous, we inductively define the following hitting times.

context $\text{nat-filtered-measure}$
begin

context
fixes $X :: \text{nat} \Rightarrow 'a \Rightarrow \text{real}$
and $a \ b :: \text{real}$
and $N :: \text{nat}$
begin

primrec $\text{upcrossing} :: \text{nat} \Rightarrow 'a \Rightarrow \text{nat}$ **where**
 $\text{upcrossing } 0 = (\lambda \omega. 0) |$
 $\text{upcrossing } (\text{Suc } n) = (\lambda \omega. \text{hitting-time } X \ \{b..\}) (\text{hitting-time } X \ \{..a\}) (\text{upcrossing } n \ \omega) \ N \ \omega)$

definition $\text{downcrossing} :: \text{nat} \Rightarrow 'a \Rightarrow \text{nat}$ **where**
 $\text{downcrossing } n = (\lambda \omega. \text{hitting-time } X \ \{..a\}) (\text{upcrossing } n \ \omega) \ N \ \omega)$

lemma upcrossing-simps :
 $\text{upcrossing } 0 = (\lambda \omega. 0)$
 $\text{upcrossing } (\text{Suc } n) = (\lambda \omega. \text{hitting-time } X \ \{b..\}) (\text{downcrossing } n \ \omega) \ N \ \omega)$
by $(\text{auto } \text{simp } \text{add: } \text{downcrossing-def})$

lemma $\text{downcrossing-simps}$:
 $\text{downcrossing } 0 = \text{hitting-time } X \ \{..a\} \ 0 \ N$
 $\text{downcrossing } n = (\lambda \omega. \text{hitting-time } X \ \{..a\}) (\text{upcrossing } n \ \omega) \ N \ \omega)$
by $(\text{auto } \text{simp } \text{add: } \text{downcrossing-def})$

declare *upcrossing.simps*[*simp del*]

lemma *upcrossing-le*: *upcrossing* $n \omega \leq N$
by (*cases* n) (*auto simp add: upcrossing-simps hitting-time-le*)

lemma *downcrossing-le*: *downcrossing* $n \omega \leq N$
by (*cases* n) (*auto simp add: downcrossing-simps hitting-time-le*)

lemma *upcrossing-le-downcrossing*: *upcrossing* $n \omega \leq$ *downcrossing* $n \omega$
unfolding *downcrossing-simps* **by** (*auto intro: hitting-time-ge upcrossing-le*)

lemma *downcrossing-le-upcrossing-Suc*: *downcrossing* $n \omega \leq$ *upcrossing* (*Suc* n) ω
unfolding *upcrossing-simps* **by** (*auto intro: hitting-time-ge downcrossing-le*)

lemma *upcrossing-mono*:
assumes $n \leq m$
shows *upcrossing* $n \omega \leq$ *upcrossing* $m \omega$
using *order-trans*[*OF upcrossing-le-downcrossing downcrossing-le-upcrossing-Suc*]
assms
by (*rule lift-Suc-mono-le*)

lemma *downcrossing-mono*:
assumes $n \leq m$
shows *downcrossing* $n \omega \leq$ *downcrossing* $m \omega$
using *order-trans*[*OF downcrossing-le-upcrossing-Suc upcrossing-le-downcrossing*]
assms
by (*rule lift-Suc-mono-le*)

— The following lemmas help us make statements about when an upcrossing (resp. downcrossing) occurs, and the value that the process takes at that instant.

lemma *stopped-value-upcrossing*:
assumes *upcrossing* (*Suc* n) $\omega \neq N$
shows *stopped-value* X (*upcrossing* (*Suc* n)) $\omega \geq b$
proof —
have $*$: *upcrossing* (*Suc* n) $\omega < N$ **using** *le-neq-implies-less upcrossing-le assms*
by *presburger*
have $\exists j \in \{\text{downcrossing } n \omega.. \text{upcrossing } (\text{Suc } n) \omega\}$. $X j \omega \in \{b..\}$
using *hitting-time-le-iff*[*THEN iffD1, OF **] *upcrossing-simps* **by** *fastforce*
then obtain j **where** $j: j \in \{\text{downcrossing } n \omega.. N\}$ $X j \omega \in \{b..\}$ **using** $*$ **by**
(*meson atLeastatMost-subset-iff le-refl subsetD upcrossing-le*)
thus *?thesis* **using** *stopped-value-hitting-time-mem*[*of j - - X*] **unfolding** *upcrossing-simps stopped-value-def* **by** *blast*
qed

lemma *stopped-value-downcrossing*:
assumes *downcrossing* $n \omega \neq N$
shows *stopped-value* X (*downcrossing* n) $\omega \leq a$
proof —

have *: *downcrossing* $n \ \omega < N$ **using** *le-neq-implies-less downcrossing-le assms*
by *presburger*
have $\exists j \in \{\text{upcrossing } n \ \omega.. \text{downcrossing } n \ \omega\}$. $X \ j \ \omega \in \{..a\}$
using *hitting-time-le-iff[THEN iffD1, OF *] downcrossing-simps* **by** *fastforce*
then obtain j **where** $j: j \in \{\text{upcrossing } n \ \omega..N\}$ $X \ j \ \omega \in \{..a\}$ **using** * **by**
(meson atLeastatMost-subset-iff le-refl subsetD downcrossing-le)
thus *?thesis* **using** *stopped-value-hitting-time-mem[of j - - X]* **unfolding** *down-*
crossing-simps stopped-value-def **by** *blast*
qed

lemma *upcrossing-less-downcrossing*:
assumes $a < b$ *downcrossing* $(\text{Suc } n) \ \omega \neq N$
shows *upcrossing* $(\text{Suc } n) \ \omega < \text{downcrossing } (\text{Suc } n) \ \omega$
proof –
have *upcrossing* $(\text{Suc } n) \ \omega \neq N$ **using** *assms* **by** *(metis le-antisym downcrossing-le upcrossing-le-downcrossing)*
hence *stopped-value* $X \ (\text{downcrossing } (\text{Suc } n)) \ \omega < \text{stopped-value } X \ (\text{upcrossing } (\text{Suc } n)) \ \omega$
using *assms stopped-value-downcrossing stopped-value-upcrossing* **by** *force*
hence *downcrossing* $(\text{Suc } n) \ \omega \neq \text{upcrossing } (\text{Suc } n) \ \omega$ **unfolding** *stopped-value-def*
by *force*
thus *?thesis* **using** *upcrossing-le-downcrossing* **by** *(simp add: le-neq-implies-less)*
qed

lemma *downcrossing-less-upcrossing*:
assumes $a < b$ *upcrossing* $(\text{Suc } n) \ \omega \neq N$
shows *downcrossing* $n \ \omega < \text{upcrossing } (\text{Suc } n) \ \omega$
proof –
have *downcrossing* $n \ \omega \neq N$ **using** *assms* **by** *(metis le-antisym upcrossing-le downcrossing-le-upcrossing-Suc)*
hence *stopped-value* $X \ (\text{downcrossing } n) \ \omega < \text{stopped-value } X \ (\text{upcrossing } (\text{Suc } n)) \ \omega$
using *assms stopped-value-downcrossing stopped-value-upcrossing* **by** *force*
hence *downcrossing* $n \ \omega \neq \text{upcrossing } (\text{Suc } n) \ \omega$ **unfolding** *stopped-value-def*
by *force*
thus *?thesis* **using** *downcrossing-le-upcrossing-Suc* **by** *(simp add: le-neq-implies-less)*
qed

lemma *upcrossing-less-Suc*:
assumes $a < b$ *upcrossing* $n \ \omega \neq N$
shows *upcrossing* $n \ \omega < \text{upcrossing } (\text{Suc } n) \ \omega$
by *(metis assms upcrossing-le-downcrossing downcrossing-less-upcrossing order-le-less-trans le-neq-implies-less upcrossing-le)*

lemma *upcrossing-eq-bound*:
assumes $a < b$ $n \geq N$
shows *upcrossing* $n \ \omega = N$

proof –
have *: *upcrossing* $N \ \omega = N$
proof –
{
 assume *: *upcrossing* $N \ \omega \neq N$
 hence *asm*: *upcrossing* $n \ \omega < N$ **if** $n \leq N$ **for** n **using** *upcrossing-mono*
upcrossing-le **that** **by** (*metis le-antisym le-neq-implies-less*)
 {
 fix $i \ j$
 assume $i \leq N \ i < j$
 hence *upcrossing* $i \ \omega \neq$ *upcrossing* $j \ \omega$ **by** (*metis Suc-leI asm assms(1) leD*
upcrossing-less-Suc upcrossing-mono)
 }
 moreover
 {
 fix j
 assume $j \leq N$
 hence *upcrossing* $j \ \omega \leq$ *upcrossing* $N \ \omega$ **using** *upcrossing-mono* **by** *blast*
 hence *upcrossing* $(\text{Suc } N) \ \omega \neq$ *upcrossing* $j \ \omega$ **using** *upcrossing-less-Suc*[*OF*
*assms(1) **] **by** *simp*
 }
 ultimately have *inj-on* $(\lambda n. \text{upcrossing } n \ \omega) \ \{\dots \text{Suc } N\}$ **unfolding** *inj-on-def*
by (*metis atMost-iff le-SucE linorder-less-linear*)
 hence *card* $((\lambda n. \text{upcrossing } n \ \omega) \ \{\dots \text{Suc } N\}) = \text{Suc } (\text{Suc } N)$ **by** (*simp add:*
inj-on-iff-eq-card[*THEN iffD1*])
 moreover have $(\lambda n. \text{upcrossing } n \ \omega) \ \{\dots \text{Suc } N\} \subseteq \{\dots N\}$ **using** *upcrossing-le*
by *blast*
 moreover have *card* $((\lambda n. \text{upcrossing } n \ \omega) \ \{\dots \text{Suc } N\}) \leq \text{Suc } N$ **using**
card-mono[*OF - calculation(2)*] **by** *simp*
 ultimately have *False* **by** *linarith*
 }
 thus *?thesis* **by** *blast*
qed
thus *?thesis* **using** *upcrossing-mono*[*OF assms(2), of* ω] *upcrossing-le*[*of* $n \ \omega$] **by**
simp
qed

lemma *downcrossing-eq-bound*:
assumes $a < b \ n \geq N$
shows *downcrossing* $n \ \omega = N$
using *upcrossing-le-downcrossing*[*of* $n \ \omega$] *downcrossing-le*[*of* $n \ \omega$] *upcrossing-eq-bound*[*OF*
assms] **by** *simp*

lemma *stopping-time-crossings*:
assumes *adapted-process* $M \ F \ 0 \ X$
shows *stopping-time* (*upcrossing* n) *stopping-time* (*downcrossing* n)
proof –
have *stopping-time* (*upcrossing* n) \wedge *stopping-time* (*downcrossing* n)
proof (*induction* n)

```

    case 0
    then show ?case unfolding upcrossing-simps downcrossing-simps
      using stopping-time-const stopping-time-hitting-time[OF assms] by simp
    next
    case (Suc n)
    have stopping-time (upcrossing (Suc n)) unfolding upcrossing-simps
      using assms Suc downcrossing-le by (intro stopping-time-hitting-time') auto
    moreover have stopping-time (downcrossing (Suc n)) unfolding downcrossing-simps
      using assms calculation upcrossing-le by (intro stopping-time-hitting-time') auto
    ultimately show ?case by blast
  qed
  thus stopping-time (upcrossing n) stopping-time (downcrossing n) by blast+
qed

```

lemmas *stopping-time-upcrossing = stopping-time-crossings(1)*
lemmas *stopping-time-downcrossing = stopping-time-crossings(2)*

— We define *upcrossings-before* as the number of upcrossings which take place strictly before time N .

definition *upcrossings-before* :: 'a \Rightarrow nat **where**
upcrossings-before = ($\lambda\omega. \text{Sup } \{n. \text{upcrossing } n \ \omega < N\}$)

lemma *upcrossings-before-bdd-above*:
assumes $a < b$
shows *bdd-above* $\{n. \text{upcrossing } n \ \omega < N\}$
proof —
have $\{n. \text{upcrossing } n \ \omega < N\} \subseteq \{..<N\}$ **unfolding** *lessThan-def Collect-mono-iff*
using *upcrossing-eq-bound[OF assms] linorder-not-less order-less-irrefl* **by** *metis*
thus ?thesis **by** (*meson bdd-above-Iio bdd-above-mono*)
qed

lemma *upcrossings-before-less*:
assumes $a < b \ 0 < N$
shows *upcrossings-before* $\omega < N$
proof —
have *: $\{n. \text{upcrossing } n \ \omega < N\} \subseteq \{..<N\}$ **unfolding** *lessThan-def Collect-mono-iff*
using *upcrossing-eq-bound[OF assms(1)] linorder-not-less order-less-irrefl* **by** *metis*
have *upcrossing 0 $\omega < N$* **unfolding** *upcrossing-simps* **by** (*rule assms*)
moreover have *Sup $\{..<N\} < N$* **unfolding** *Sup-nat-def* **using** *assms* **by** *simp*
ultimately show ?thesis **unfolding** *upcrossings-before-def* **using** *cSup-subset-mono[OF - - *]* **by** *force*
qed

lemma *upcrossings-before-less-implies-crossing-eq-bound*:

assumes $a < b$ *upcrossings-before* $\omega < n$
shows *upcrossing* $n \omega = N$
downcrossing $n \omega = N$

proof –
have \neg *upcrossing* $n \omega < N$ **using** *assms* *upcrossings-before-bdd-above*[of ω]
upcrossings-before-def *bdd-above-nat* *finite-Sup-less-iff* **by** *fastforce*
thus *upcrossing* $n \omega = N$ **using** *upcrossing-le*[of $n \omega$] **by** *simp*
thus *downcrossing* $n \omega = N$ **using** *upcrossing-le-downcrossing*[of $n \omega$] *downcrossing-le*[of $n \omega$] **by** *simp*
qed

lemma *upcrossings-before-le*:
assumes $a < b$
shows *upcrossings-before* $\omega \leq N$
using *upcrossings-before-less* *assms* *less-le-not-le* *upcrossings-before-def*
by (*cases* N) *auto*

lemma *upcrossings-before-mem*:
assumes $a < b$ $0 < N$
shows *upcrossings-before* $\omega \in \{n. \text{upcrossing } n \omega < N\} \cap \{.. < N\}$

proof –
have *upcrossing* $0 \omega < N$ **using** *assms* **unfolding** *upcrossing-simps* **by** *simp*
hence $\{n. \text{upcrossing } n \omega < N\} \neq \{\}$ **by** *blast*
moreover **have** *finite* $\{n. \text{upcrossing } n \omega < N\}$ **using** *upcrossings-before-bdd-above*[OF *assms*(1)] **by** (*simp* *add*: *bdd-above-nat*)
ultimately show *?thesis* **using** *Max-in* *upcrossings-before-less*[OF *assms*(1,2)]
Sup-nat-def *upcrossings-before-def* **by** *auto*
qed

lemma *upcrossing-less-of-le-upcrossings-before*:
assumes $a < b$ $0 < N$ $n \leq \text{upcrossings-before } \omega$
shows *upcrossing* $n \omega < N$
using *upcrossings-before-mem*[OF *assms*(1,2), of ω] *upcrossing-mono*[OF *assms*(3), of ω] **by** *simp*

lemma *upcrossings-before-sum-def*:
assumes $a < b$
shows *upcrossings-before* $\omega = (\sum k \in \{1..N\}. \text{indicator } \{n. \text{upcrossing } n \omega < N\} k)$

proof (*cases* N)
case 0
then show *?thesis* **unfolding** *upcrossings-before-def* **by** *simp*
next
case (*Suc* N)
have *upcrossing* $0 \omega < N$ **using** *assms* *Suc* **unfolding** *upcrossing-simps* **by** *simp*
hence $\{n. \text{upcrossing } n \omega < N\} \neq \{\}$ **by** *blast*
hence $*$: \neg *upcrossing* $n \omega < N$ **if** $n \in \{\text{upcrossings-before } \omega < ..N\}$ **for** n
using *finite-Sup-less-iff*[THEN *iffD1*, OF *bdd-above-nat*[THEN *iffD1*, OF *upcrossings-before-bdd-above*], of ω n]

by (*metis that assms greaterThanAtMost-iff less-not-refl mem-Collect-eq upcrossings-before-def*)
have **: *upcrossing* $n \omega < N$ **if** $n \in \{1..upcrossings\text{-before } \omega\}$ **for** n
using *assms that Suc* **by** (*intro upcrossing-less-of-le-upcrossings-before*) *auto*
have *upcrossings-before* $\omega < N$ **using** *upcrossings-before-less Suc assms* **by** *simp*
hence $\{1..N\} - \{1..upcrossings\text{-before } \omega\} = \{upcrossings\text{-before } \omega <..N\}$
 $\{1..N\} \cap \{1..upcrossings\text{-before } \omega\} = \{1..upcrossings\text{-before } \omega\}$ **by** *force+*
hence $(\sum k \in \{1..N\}. indicator \{n. upcrossing n \omega < N\} k) =$
 $(\sum k \in \{1..upcrossings\text{-before } \omega\}. indicator \{n. upcrossing n \omega < N\} k) +$
 $(\sum k \in \{upcrossings\text{-before } \omega <..N\}. indicator \{n. upcrossing n \omega < N\} k)$
using *sum.Int-Diff[OF finite-atLeastAtMost, of - 1 N \{1..upcrossings-before \omega\}]* **by** *metis*
also have $\dots = upcrossings\text{-before } \omega$ **using** * ** **by** *simp*
finally show *?thesis* **by** *argo*
qed

lemma *upcrossings-before-measurable*:
assumes *adapted-process M F 0 X a < b*
shows *upcrossings-before* \in *borel-measurable M*
unfolding *upcrossings-before-sum-def*[*OF assms(2)*]
using *stopping-time-measurable*[*OF stopping-time-crossings(1), OF assms(1)*] **by**
simp

lemma *upcrossings-before-measurable'*:
assumes *adapted-process M F 0 X a < b*
shows $(\lambda \omega. real (upcrossings\text{-before } \omega)) \in$ *borel-measurable M*
using *real-embedding-borel-measurable upcrossings-before-measurable*[*OF assms*]
by *simp*

end

lemma *crossing-eq-crossing*:
assumes $N \leq N'$
and *downcrossing* $X a b N n \omega < N$
shows *upcrossing* $X a b N n \omega = upcrossing X a b N' n \omega$
 $downcrossing X a b N n \omega = downcrossing X a b N' n \omega$
proof –
have *upcrossing* $X a b N n \omega = upcrossing X a b N' n \omega \wedge downcrossing X a b N n \omega = downcrossing X a b N' n \omega$ **using** *assms(2)*
proof (*induction n*)
case 0
show *?case* **by** (*metis (no-types, lifting) upcrossing-simps(1) assms atLeast-0 bot-nat-0.extremum hitting-time-def hitting-time-eq-hitting-time inf-top.right-neutral leD downcrossing-mono downcrossing-simps(1) max-nat.left-neutral*)
next
case (*Suc n*)
hence *upper-less*: *upcrossing* $X a b N (Suc n) \omega < N$ **using** *upcrossing-le-downcrossing Suc order.strict-trans1* **by** *blast*
hence *lower-less*: *downcrossing* $X a b N n \omega < N$ **using** *downcrossing-le-upcrossing-Suc*

order.strict-trans1 **by** *blast*

obtain j **where** $j \in \{\text{downcrossing } X \ a \ b \ N \ n \ \omega..<N\} \ X \ j \ \omega \in \{b..\}$
using *hitting-time-less-iff*[*THEN iffD1*, *OF order-refl*] *upper-less* **by** (*force simp add: upcrossing-simps*)
hence *upper-eq*: $\text{upcrossing } X \ a \ b \ N \ (\text{Suc } n) \ \omega = \text{upcrossing } X \ a \ b \ N' \ (\text{Suc } n) \ \omega$
using *Suc(1)*[*OF lower-less*] *assms(1)*
by (*auto simp add: upcrossing-simps intro!: hitting-time-eq-hitting-time*)
obtain j **where** $j: j \in \{\text{upcrossing } X \ a \ b \ N \ (\text{Suc } n) \ \omega..<N\} \ X \ j \ \omega \in \{..a\}$
using *Suc(2)* *hitting-time-less-iff*[*THEN iffD1*, *OF order-refl*] **by** (*force simp add: downcrossing-simps*)
thus *?case unfolding downcrossing-simps upper-eq* **by** (*force intro: hitting-time-eq-hitting-time assms*)
qed
thus $\text{upcrossing } X \ a \ b \ N \ n \ \omega = \text{upcrossing } X \ a \ b \ N' \ n \ \omega \ \text{downcrossing } X \ a \ b \ N \ n \ \omega = \text{downcrossing } X \ a \ b \ N' \ n \ \omega$ **by** *auto*
qed

lemma *crossing-eq-crossing'*:

assumes $N \leq N'$
and $\text{upcrossing } X \ a \ b \ N \ (\text{Suc } n) \ \omega < N$
shows $\text{upcrossing } X \ a \ b \ N \ (\text{Suc } n) \ \omega = \text{upcrossing } X \ a \ b \ N' \ (\text{Suc } n) \ \omega$
 $\text{downcrossing } X \ a \ b \ N \ n \ \omega = \text{downcrossing } X \ a \ b \ N' \ n \ \omega$
proof –
show *lower-eq*: $\text{downcrossing } X \ a \ b \ N \ n \ \omega = \text{downcrossing } X \ a \ b \ N' \ n \ \omega$
using *downcrossing-le-upcrossing-Suc*[*THEN order.strict-trans1*] *crossing-eq-crossing assms* **by** *fast*
have $\exists j \in \{\text{downcrossing } X \ a \ b \ N \ n \ \omega..<N\}. \ X \ j \ \omega \in \{b..\}$ **using** *assms(2)* **by** (*intro hitting-time-less-iff*[*OF order-refl*, *THEN iffD1*]) (*simp add: upcrossing-simps lower-eq*)
then obtain j **where** $j \in \{\text{downcrossing } X \ a \ b \ N \ n \ \omega..N\} \ X \ j \ \omega \in \{b..\}$ **by** *fastforce*
thus $\text{upcrossing } X \ a \ b \ N \ (\text{Suc } n) \ \omega = \text{upcrossing } X \ a \ b \ N' \ (\text{Suc } n) \ \omega$
unfolding *upcrossing-simps stopped-value-def* **using** *hitting-time-eq-hitting-time*[*OF assms(1)*] *lower-eq* **by** *metis*
qed

lemma *upcrossing-eq-upcrossing*:

assumes $N \leq N'$
and $\text{upcrossing } X \ a \ b \ N \ n \ \omega < N$
shows $\text{upcrossing } X \ a \ b \ N \ n \ \omega = \text{upcrossing } X \ a \ b \ N' \ n \ \omega$
using *crossing-eq-crossing'*[*OF assms(1)*] *assms(2)* *upcrossing-simps*
by (*cases n*) (*presburger, fast*)

lemma *upcrossings-before-zero*: $\text{upcrossings-before } X \ a \ b \ 0 \ \omega = 0$

unfolding *upcrossings-before-def* **by** *simp*

lemma *upcrossings-before-less-exists-upcrossing*:

assumes $a < b$

and *upcrossing*: $N \leq L \ X \ L \ \omega < a \ L \leq U \ b < X \ U \ \omega$
shows *upcrossings-before* $X \ a \ b \ N \ \omega < \text{upcrossings-before } X \ a \ b \ (\text{Suc } U) \ \omega$
proof –
have *upcrossing* $X \ a \ b \ (\text{Suc } U) \ (\text{upcrossings-before } X \ a \ b \ N \ \omega) \ \omega \leq L$
using *assms upcrossing-le*[*THEN order-trans, OF upcrossing(1)*]
by (*cases* $0 < N$, *subst upcrossing-eq-upcrossing*[*of N Suc U, symmetric, OF -*
upcrossing-less-of-le-upcrossings-before])
(auto simp add: upcrossings-before-zero upcrossing-simps)
hence *downcrossing* $X \ a \ b \ (\text{Suc } U) \ (\text{upcrossings-before } X \ a \ b \ N \ \omega) \ \omega \leq U$
unfolding *downcrossing-simps using upcrossing by* (*force intro: hitting-time-le-iff*[*THEN*
iffD2])
hence *upcrossing* $X \ a \ b \ (\text{Suc } U) \ (\text{Suc } (\text{upcrossings-before } X \ a \ b \ N \ \omega)) \ \omega < \text{Suc}$
 U
unfolding *upcrossing-simps using upcrossing by* (*force intro: hitting-time-less-iff*[*THEN*
iffD2])
thus *?thesis using cSup-upper*[*OF - upcrossings-before-bdd-above*[*OF assms(1)*]]
upcrossings-before-def by fastforce
qed

lemma *crossings-translate*:

upcrossing $X \ a \ b \ N = \text{upcrossing } (\lambda n \ \omega. (X \ n \ \omega + c)) \ (a + c) \ (b + c) \ N$
downcrossing $X \ a \ b \ N = \text{downcrossing } (\lambda n \ \omega. (X \ n \ \omega + c)) \ (a + c) \ (b + c) \ N$

proof –

have *upper*: *upcrossing* $X \ a \ b \ N \ n = \text{upcrossing } (\lambda n \ \omega. (X \ n \ \omega + c)) \ (a + c) \ (b$
 $+ c) \ N \ n$ **for** n

proof (*induction n*)

case 0

then show *?case by* (*simp only: upcrossing.simps*)

next

case $(\text{Suc } n)$

have $((+) \ c \ \{..a\}) = \{..a + c\}$ **by** *simp*

moreover have $((+) \ c \ \{b..\}) = \{b + c..\}$ **by** *simp*

ultimately show *?case unfolding upcrossing.simps using hitting-time-translate*[*of*
 $X \ \{b..\} \ c$] *hitting-time-translate*[*of* $X \ \{..a\} \ c$] *Suc by presburger*

qed

thus *upcrossing* $X \ a \ b \ N = \text{upcrossing } (\lambda n \ \omega. (X \ n \ \omega + c)) \ (a + c) \ (b + c) \ N$
by *blast*

have $((+) \ c \ \{..a\}) = \{..a + c\}$ **by** *simp*

thus *downcrossing* $X \ a \ b \ N = \text{downcrossing } (\lambda n \ \omega. (X \ n \ \omega + c)) \ (a + c) \ (b$
 $+ c) \ N$ **using** *upper downcrossing-simps hitting-time-translate*[*of* $X \ \{..a\} \ c$] **by**
presburger

qed

lemma *upcrossings-before-translate*:

upcrossings-before $X \ a \ b \ N = \text{upcrossings-before } (\lambda n \ \omega. (X \ n \ \omega + c)) \ (a + c) \ (b$
 $+ c) \ N$

using *upcrossings-before-def crossings-translate by simp*

lemma *crossings-pos-eq*:

assumes $a < b$
shows $\text{upcrossing } X \ a \ b \ N = \text{upcrossing } (\lambda n \ \omega. \max 0 \ (X \ n \ \omega - a)) \ 0 \ (b - a) \ N$
 $\text{downcrossing } X \ a \ b \ N = \text{downcrossing } (\lambda n \ \omega. \max 0 \ (X \ n \ \omega - a)) \ 0 \ (b - a) \ N$
proof –
have $*$: $\max 0 \ (x - a) \in \{..0\} \iff x - a \in \{..0\}$ $\max 0 \ (x - a) \in \{b - a..\}$
 $\iff x - a \in \{b - a..\}$ **for** x **using** *assms* **by** *auto*
have $\text{upcrossing } X \ a \ b \ N = \text{upcrossing } (\lambda n \ \omega. X \ n \ \omega - a) \ 0 \ (b - a) \ N$ **using**
crossings-translate[of $X \ a \ b \ N - a$] **by** *simp*
thus *upper*: $\text{upcrossing } X \ a \ b \ N = \text{upcrossing } (\lambda n \ \omega. \max 0 \ (X \ n \ \omega - a)) \ 0 \ (b - a) \ N$ **unfolding** *upcrossing-def hitting-time-def'* **using** $*$ **by** *presburger*

thus $\text{downcrossing } X \ a \ b \ N = \text{downcrossing } (\lambda n \ \omega. \max 0 \ (X \ n \ \omega - a)) \ 0 \ (b - a) \ N$
unfolding *downcrossing-simps hitting-time-def'* **using** *upper* $*$ **by** *simp*
qed

lemma *upcrossings-before-mono*:
assumes $a < b \ N \leq N'$
shows $\text{upcrossings-before } X \ a \ b \ N \ \omega \leq \text{upcrossings-before } X \ a \ b \ N' \ \omega$
proof (*cases* N)
case 0
then show *?thesis* **unfolding** *upcrossings-before-def* **by** *simp*
next
case (*Suc* N')
hence $\text{upcrossing } X \ a \ b \ N \ 0 \ \omega < N$ **unfolding** *upcrossing-simps* **by** *simp*
thus *?thesis* **unfolding** *upcrossings-before-def* **using** *upcrossings-before-bdd-above*
upcrossing-eq-upcrossing *assms* **by** (*intro* *cSup-subset-mono*) *auto*
qed

lemma *upcrossings-before-pos-eq*:
assumes $a < b$
shows $\text{upcrossings-before } X \ a \ b \ N = \text{upcrossings-before } (\lambda n \ \omega. \max 0 \ (X \ n \ \omega - a)) \ 0 \ (b - a) \ N$
using *upcrossings-before-def crossings-pos-eq*[*OF* *assms*] **by** *simp*

— We define *upcrossings* to be the total number of upcrossings a stochastic process completes as $N \longrightarrow \infty$.

definition *upcrossings* :: $(\text{nat} \Rightarrow 'a \Rightarrow \text{real}) \Rightarrow \text{real} \Rightarrow \text{real} \Rightarrow 'a \Rightarrow \text{ennreal}$ **where**
 $\text{upcrossings } X \ a \ b = (\lambda \omega. (\text{SUP } N. \text{ennreal } (\text{upcrossings-before } X \ a \ b \ N \ \omega)))$

lemma *upcrossings-measurable*:
assumes *adapted-process* $M \ F \ 0 \ X \ a < b$
shows $\text{upcrossings } X \ a \ b \in \text{borel-measurable } M$
unfolding *upcrossings-def*
using *upcrossings-before-measurable'*[*OF* *assms*] **by** (*auto* *intro!*: *borel-measurable-SUP*)

end

lemma (in *nat-finite-filtered-measure*) *integrable-upcrossings-before*:
assumes *adapted-process* $M F 0 X a < b$
shows *integrable* $M (\lambda\omega. \text{real} (\text{upcrossings-before } X a b N \omega))$
proof –
have $(\int^+ x. \text{ennreal} (\text{norm} (\text{real} (\text{upcrossings-before } X a b N x))) \partial M) \leq (\int^+ x. \text{ennreal } N \partial M)$ **using** *upcrossings-before-le[OF assms(2)]* **by** (*intro nn-integral-mono simp*)
also have $\dots = \text{ennreal } N * \text{emeasure } M (\text{space } M)$ **by** *simp*
also have $\dots < \infty$ **by** (*metis emeasure-real ennreal-less-top ennreal-mult-less-top infinity-ennreal-def*)
finally show *?thesis* **by** (*intro integrableI-bounded upcrossings-before-measurable' assms*)
qed

3.2 Doob's Upcrossing Inequality

Doob's upcrossing inequality provides a bound on the expected number of upcrossings a submartingale completes before some point in time. The proof follows the proof presented in the paper *A Formalization of Doob's Martingale Convergence Theorems in mathlib* [1] [2].

context *nat-finite-filtered-measure*
begin

theorem *upcrossing-inequality*:
fixes $a b :: \text{real}$ **and** $N :: \text{nat}$
assumes *submartingale* $M F 0 X$
shows $(b - a) * (\int \omega. \text{real} (\text{upcrossings-before } X a b N \omega) \partial M) \leq (\int \omega. \text{max } 0 (X N \omega - a) \partial M)$
proof –
interpret *submartingale-linorder* $M F 0 X$ **unfolding** *submartingale-linorder-def*
by (*intro assms*)
show *?thesis*
proof (*cases* $a < b$)
case *True*
– We show the statement first for $X 0$ non-negative and $X N$ greater than or equal to a .
have $*$: $(b - a) * (\int \omega. \text{real} (\text{upcrossings-before } X a b N \omega) \partial M) \leq (\int \omega. X N \omega \partial M)$
if *asm*: *submartingale* $M F 0 X a < b \wedge \omega. X 0 \omega \geq 0 \wedge \omega. X N \omega \geq a$
for $a b X$
proof –
interpret *subm*: *submartingale* $M F 0 X$ **by** (*intro asm*)
define $C :: \text{nat} \Rightarrow 'a \Rightarrow \text{real}$ **where** $C = (\lambda n \omega. \sum k < N. \text{indicator} \{ \text{downcrossing } X a b N k \omega .. < \text{upcrossing } X a b N (Suc k) \omega \} n)$
have *C-values*: $C n \omega \in \{0, 1\}$ **for** $n \omega$
proof (*cases* $\exists j < N. n \in \{ \text{downcrossing } X a b N j \omega .. < \text{upcrossing } X a b N (Suc j) \omega \}$)

case *True*
then obtain j **where** $j: j \in \{..<N\}$ $n \in \{\text{downcrossing } X \text{ a b } N \ j \ \omega..<\text{upcrossing } X \text{ a b } N \ (\text{Suc } j) \ \omega\}$ **by** *blast*
{
fix $k \ l :: \text{nat}$ **assume** $k\text{-less-}l: k < l$
hence $\text{Suc-}k\text{-le-}l: \text{Suc } k \leq l$ **by** *simp*
have $\{\text{downcrossing } X \text{ a b } N \ k \ \omega..<\text{upcrossing } X \text{ a b } N \ (\text{Suc } k) \ \omega\} \cap$
 $\{\text{downcrossing } X \text{ a b } N \ l \ \omega..<\text{upcrossing } X \text{ a b } N \ (\text{Suc } l) \ \omega\} =$
 $\{\text{downcrossing } X \text{ a b } N \ l \ \omega..<\text{upcrossing } X \text{ a b } N \ (\text{Suc } k) \ \omega\}$
using $k\text{-less-}l$ *upcrossing-mono downcrossing-mono* **by** *simp*
moreover have $\text{upcrossing } X \text{ a b } N \ (\text{Suc } k) \ \omega \leq \text{downcrossing } X \text{ a b } N \ l \ \omega$
using *upcrossing-le-downcrossing downcrossing-mono[OF Suc-k-le-l]*
order-trans **by** *blast*
ultimately have $\{\text{downcrossing } X \text{ a b } N \ k \ \omega..<\text{upcrossing } X \text{ a b } N \ (\text{Suc } k) \ \omega\} \cap$
 $\{\text{downcrossing } X \text{ a b } N \ l \ \omega..<\text{upcrossing } X \text{ a b } N \ (\text{Suc } l) \ \omega\} = \{\}$ **by** *simp*
}
hence *disjoint-family-on* $(\lambda k. \{\text{downcrossing } X \text{ a b } N \ k \ \omega..<\text{upcrossing } X \text{ a b } N \ (\text{Suc } k) \ \omega\}) \ \{..<N\}$
unfolding *disjoint-family-on-def*
by *(metis Int-commute linorder-less-linear)*
hence $C \ n \ \omega = 1$ **unfolding** *C-def* **using** *sum-indicator-disjoint-family[where ?f= $\lambda-. 1$]* j **by** *fastforce*
thus *?thesis* **by** *blast*
next
case *False*
hence $C \ n \ \omega = 0$ **unfolding** *C-def* **by** *simp*
thus *?thesis* **by** *simp*
qed
hence *C-interval*: $C \ n \ \omega \in \{0..1\}$ **for** $n \ \omega$ **by** *(metis atLeastAtMost-iff empty-iff insert-iff order.refl zero-less-one-class.zero-le-one)*

— We consider the discrete stochastic integral of C and $\lambda n \ \omega. 1 - C \ n \ \omega$.

define C' **where** $C' = (\lambda n \ \omega. \sum k < n. C \ k \ \omega *_{\mathbb{R}} (X \ (\text{Suc } k) \ \omega - X \ k \ \omega))$
define *one-minus-C'* **where** *one-minus-C'* $= (\lambda n \ \omega. \sum k < n. (1 - C \ k \ \omega) *_{\mathbb{R}} (X \ (\text{Suc } k) \ \omega - X \ k \ \omega))$

— We use the fact that the crossing times are stopping times to show that C is predictable.

have *adapted-C*: *adapted-process* $M \ F \ 0 \ C$
proof
fix i
have $(\lambda \omega. \text{indicat-real } \{\text{downcrossing } X \text{ a b } N \ k \ \omega..<\text{upcrossing } X \text{ a b } N \ (\text{Suc } k) \ \omega\} \ i) \in \text{borel-measurable } (F \ i)$ **for** k
unfolding *indicator-def*
using *stopping-time-upcrossing[OF subm.adapted-process-axioms, THEN stopping-time-measurable-gr]*
stopping-time-downcrossing[OF subm.adapted-process-axioms, THEN stopping-time-measurable-le]
by *force*

thus $C i \in \text{borel-measurable } (F i)$ **unfolding** $C\text{-def}$ **by** simp
qed
hence $\text{adapted-process } M F 0 (\lambda n \omega. 1 - C n \omega)$ **by** $(\text{intro adapted-process.diff-adapted adapted-process-const})$
hence $\text{submartingale-one-minus-}C'$: $\text{submartingale } M F 0 \text{ one-minus-}C'$
unfolding $\text{one-minus-}C'\text{-def}$ **using** $C\text{-interval}$
by $(\text{intro submartingale-partial-sum-scaleR}[of - - 1] \text{submartingale-linorder.intro asm}) \text{ auto}$

have $C n \in \text{borel-measurable } M$ **for** n
using $\text{adapted-}C \text{ adapted-process.adapted measurable-from-subalg subalg}$ **by**
 blast

have $\text{integrable-}C'$: $\text{integrable } M (C' n)$ **for** n **unfolding** $C'\text{-def}$ **using**
 $C\text{-interval}$
by $(\text{intro submartingale-partial-sum-scaleR}[THEN \text{submartingale.integrable}] \text{submartingale-linorder.intro adapted-}C \text{asm}) \text{ auto}$

— We show the following inequality, by using the fact that $\text{one-minus-}C'$ is a submartingale.
have $\text{integral}^L M (C' n) \leq \text{integral}^L M (X n)$ **for** n
proof –
interpret subm' : $\text{submartingale-linorder } M F 0 \text{ one-minus-}C'$ **unfolding**
 $\text{submartingale-linorder-def}$ **by** $(\text{rule submartingale-one-minus-}C')$
have $0 \leq \text{integral}^L M (\text{one-minus-}C' n)$
using $\text{subm}'.\text{set-integral-le}[OF \text{sets.top, where } i=0 \text{ and } j=n] \text{space-}F$
 $\text{subm}'.\text{integrable}$ **by** $(\text{fastforce simp add: set-integral-space one-minus-}C'\text{-def})$
moreover **have** $\text{one-minus-}C' n \omega = (\sum k < n. X (\text{Suc } k) \omega - X k \omega) -$
 $C' n \omega$ **for** ω
unfolding $\text{one-minus-}C'\text{-def } C'\text{-def}$ **by** $(\text{simp only: scaleR-diff-left sum-subtractf scale-one})$
ultimately **have** $0 \leq (LINT \omega | M. (\sum k < n. X (\text{Suc } k) \omega - X k \omega)) -$
 $\text{integral}^L M (C' n)$
using $\text{subm}.integrable \text{integrable-}C'$
by $(\text{subst Bochner-Integration.integral-diff}[symmetric]) (\text{auto simp add: one-minus-}C'\text{-def})$
moreover **have** $(LINT \omega | M. (\sum k < n. X (\text{Suc } k) \omega - X k \omega)) \leq (LINT$
 $\omega | M. X n \omega)$ **using** $\text{asm sum-lessThan-telescope}[of \lambda i. X i - n] \text{subm}.integrable$
by $(\text{intro integral-mono}) \text{ auto}$
ultimately **show** $?thesis$ **by** linarith
qed
moreover **have** $(b - a) * (\int \omega. \text{real } (\text{upcrossings-before } X a b N \omega) \partial M) \leq$
 $\text{integral}^L M (C' N)$
proof $(\text{cases } N)$
case 0
then **show** $?thesis$ **using** $C'\text{-def upcrossings-before-zero}$ **by** simp
next
case $(\text{Suc } N')$
{

fix ω
have *dc-not-N*: *downcrossing* $X a b N k \omega \neq N$ **if** $k < \text{upcrossings-before } X a b N \omega$ **for** k
by (*metis Suc Suc-leI asm(2) downcrossing-le-upcrossing-Suc leD that upcrossing-less-of-le-upcrossings-before zero-less-Suc*)
have *uc-not-N*: *upcrossing* $X a b N (\text{Suc } k) \omega \neq N$ **if** $k < \text{upcrossings-before } X a b N \omega$ **for** k
by (*metis Suc Suc-leI asm(2) order-less-irrefl that upcrossing-less-of-le-upcrossings-before zero-less-Suc*)

have *subset-lessThan-N*: $\{\text{downcrossing } X a b N i \omega \dots < \text{upcrossing } X a b N (\text{Suc } i) \omega\} \subseteq \{.. < N\}$ **if** $i < N$ **for** i **using** *that*
by (*simp add: lessThan-atLeast0 upcrossing-le*)

— First we rewrite the sum as follows:

have $C' N \omega = (\sum k < N. \sum i < N. \text{indicator } \{\text{downcrossing } X a b N i \omega \dots < \text{upcrossing } X a b N (\text{Suc } i) \omega\} k * (X (\text{Suc } k) \omega - X k \omega))$
unfolding *C'-def C-def* **by** (*simp add: sum-distrib-right*)
also have $\dots = (\sum i < N. \sum k < N. \text{indicator } \{\text{downcrossing } X a b N i \omega \dots < \text{upcrossing } X a b N (\text{Suc } i) \omega\} k * (X (\text{Suc } k) \omega - X k \omega))$
using *sum.swap* **by** *fast*
also have $\dots = (\sum i < N. \sum k \in \{.. < N\} \cap \{\text{downcrossing } X a b N i \omega \dots < \text{upcrossing } X a b N (\text{Suc } i) \omega\}. X (\text{Suc } k) \omega - X k \omega)$
by (*subst Indicator-Function.sum-indicator-mult*) *simp+*
also have $\dots = (\sum i < N. \sum k \in \{\text{downcrossing } X a b N i \omega \dots < \text{upcrossing } X a b N (\text{Suc } i) \omega\}. X (\text{Suc } k) \omega - X k \omega)$
using *subset-lessThan-N[THEN Int-absorb1]* **by** *simp*
also have $\dots = (\sum i < N. X (\text{upcrossing } X a b N (\text{Suc } i) \omega) \omega - X (\text{downcrossing } X a b N i \omega) \omega)$
by (*subst sum-Suc-diff'[OF downcrossing-le-upcrossing-Suc]*) *blast*
finally have $*$: $C' N \omega = (\sum i < N. X (\text{upcrossing } X a b N (\text{Suc } i) \omega) \omega - X (\text{downcrossing } X a b N i \omega) \omega)$.

— For $k \leq N$, we consider three cases:

- 1. If $k < \text{upcrossings-before } X a b N \omega$, then $X (\text{upcrossing } X a b N (\text{Suc } k) \omega) \omega - X (\text{downcrossing } X a b N k \omega) \omega \geq b - a$
- 2. If $\text{upcrossings-before } X a b N \omega < k$, then $X (\text{upcrossing } X a b N (\text{Suc } k) \omega) \omega = X (\text{downcrossing } X a b N k \omega) \omega$
- 3. If $k = \text{upcrossings-before } X a b N \omega$, then $X (\text{upcrossing } X a b N (\text{Suc } k) \omega) \omega - X (\text{downcrossing } X a b N k \omega) \omega \geq 0$

have *summand-zero-if*: $X (\text{upcrossing } X a b N (\text{Suc } k) \omega) \omega - X (\text{downcrossing } X a b N k \omega) \omega = 0$ **if** $k > \text{upcrossings-before } X a b N \omega$ **for** k
using *that upcrossings-before-less-implies-crossing-eq-bound[OF asm(2)]*
by *simp*

have *summand-nonneg-if*: $X (\text{upcrossing } X a b N (\text{Suc } (\text{upcrossings-before } X a b N \omega)) \omega) \omega - X (\text{downcrossing } X a b N (\text{upcrossings-before } X a b N \omega) \omega)$

$\omega \geq 0$
using *upcrossings-before-less-implies-crossing-eq-bound(1)[OF asm(2) lessI]*
stopped-value-downcrossing[of X a b N - ω , THEN order-trans, OF -asm(4)[of ω]]
by (*cases downcrossing X a b N (upcrossings-before X a b N ω) $\omega \neq N$*)
(simp add: stopped-value-def)+

have *interval: {upcrossings-before X a b N $\omega..<N$ } - {upcrossings-before X a b N ω } = {upcrossings-before X a b N $\omega<..<N$ }*
using *Diff-insert atLeastSucLessThan-greaterThanLessThan lessThan-Suc lessThan-minus-lessThan* **by** *metis*

have $(b - a) * \text{real } (\text{upcrossings-before } X \ a \ b \ N \ \omega) = (\sum \text{..} < \text{upcrossings-before } X \ a \ b \ N \ \omega. \ b - a)$ **by** *simp*
also have $\dots \leq (\sum k < \text{upcrossings-before } X \ a \ b \ N \ \omega. \text{stopped-value } X \ (\text{upcrossing } X \ a \ b \ N \ (\text{Suc } k)) \ \omega - \text{stopped-value } X \ (\text{downcrossing } X \ a \ b \ N \ k) \ \omega)$
using *stopped-value-downcrossing[OF dc-not-N] stopped-value-upcrossing[OF uc-not-N]* **by** (*force intro!: sum-mono*)
also have $\dots = (\sum k < \text{upcrossings-before } X \ a \ b \ N \ \omega. X \ (\text{upcrossing } X \ a \ b \ N \ (\text{Suc } k) \ \omega) \ \omega - X \ (\text{downcrossing } X \ a \ b \ N \ k \ \omega) \ \omega)$ **unfolding** *stopped-value-def* **by** *blast*
also have $\dots \leq (\sum k < \text{upcrossings-before } X \ a \ b \ N \ \omega. X \ (\text{upcrossing } X \ a \ b \ N \ (\text{Suc } k) \ \omega) \ \omega - X \ (\text{downcrossing } X \ a \ b \ N \ k \ \omega) \ \omega)$
 $+ (\sum k \in \{\text{upcrossings-before } X \ a \ b \ N \ \omega\}. X \ (\text{upcrossing } X \ a \ b \ N \ (\text{Suc } k) \ \omega) \ \omega - X \ (\text{downcrossing } X \ a \ b \ N \ k \ \omega) \ \omega)$
 $+ (\sum k \in \{\text{upcrossings-before } X \ a \ b \ N \ \omega < .. < N\}. X \ (\text{upcrossing } X \ a \ b \ N \ (\text{Suc } k) \ \omega) \ \omega - X \ (\text{downcrossing } X \ a \ b \ N \ k \ \omega) \ \omega)$
using *summand-zero-if summand-nonneg-if* **by** *auto*
also have $\dots = (\sum k < N. X \ (\text{upcrossing } X \ a \ b \ N \ (\text{Suc } k) \ \omega) \ \omega - X \ (\text{downcrossing } X \ a \ b \ N \ k \ \omega) \ \omega)$
using *upcrossings-before-le[OF asm(2)]*
by (*subst sum.subset-diff[where A={..<N} and B={..<upcrossings-before X a b N ω }]*, *simp*, *simp*,
subst sum.subset-diff[where A={..<N} - {..<upcrossings-before X a b N ω } and B={upcrossings-before X a b N ω }])
(simp add: Suc asm(2) upcrossings-before-less, simp, simp add: interval)
finally have $(b - a) * \text{real } (\text{upcrossings-before } X \ a \ b \ N \ \omega) \leq C' N \ \omega$
using *** **by** *presburger*
}
thus *?thesis using integrable-upcrossings-before subm.adapted-process-axioms asm integrable-C'*
by (*subst integral-mult-right-zero[symmetric], intro integral-mono*) *auto*
qed
ultimately show *?thesis using order-trans* **by** *blast*
qed

have $(b - a) * (\int \omega. \text{real } (\text{upcrossings-before } X \ a \ b \ N \ \omega) \ \partial M) = (b - a) * (\int \omega. \text{real } (\text{upcrossings-before } (\lambda n \ \omega. \max \ 0 \ (X \ n \ \omega - a)) \ 0 \ (b - a) \ N \ \omega) \ \partial M)$

using *upcrossings-before-pos-eq*[*OF True*] **by** *simp*
also have $\dots \leq (\int \omega. \max 0 (X N \omega - a) \partial M)$
using $*$ [*OF submartingale-linorder.max-0*[*OF submartingale-linorder.intro*,
OF submartingale.diff, *OF assms supermartingale-const*], of $0 b - a$] **True by**
simp
finally show *?thesis* .
next
case *False*
have $0 \leq (\int \omega. \max 0 (X N \omega - a) \partial M)$ **by** *simp*
moreover have $0 \leq (\int \omega. \text{real } (\text{upcrossings-before } X a b N \omega) \partial M)$ **by** *simp*
moreover have $b - a \leq 0$ **using** *False* **by** *simp*
ultimately show *?thesis* **using** *mult-nonpos-nonneg order-trans* **by** *meson*
qed
qed

theorem *upcrossing-inequality-Sup*:

fixes $a b :: \text{real}$
assumes *submartingale M F 0 X*
shows $(b - a) * (\int^{+\omega}. \text{upcrossings } X a b \omega \partial M) \leq (\text{SUP } N. (\int^{+\omega}. \max 0 (X N \omega - a) \partial M))$
proof –
interpret *submartingale M F 0 X* **by** (*intro assms*)
show *?thesis*
proof (*cases a < b*)
case *True*
have $(\int^{+\omega}. \text{upcrossings } X a b \omega \partial M) = (\text{SUP } N. (\int^{+\omega}. \text{real } (\text{upcrossings-before } X a b N \omega) \partial M))$
unfolding *upcrossings-def*
using *upcrossings-before-mono True upcrossings-before-measurable'*[*OF adapted-process-axioms*]
by (*auto intro: nn-integral-monotone-convergence-SUP simp add: mono-def le-funI*)
hence $(b - a) * (\int^{+\omega}. \text{upcrossings } X a b \omega \partial M) = (\text{SUP } N. (b - a) * (\int^{+\omega}. \text{real } (\text{upcrossings-before } X a b N \omega) \partial M))$
by (*simp add: SUP-mult-left-ennreal*)
moreover
{
fix N
have $(\int^{+\omega}. \text{real } (\text{upcrossings-before } X a b N \omega) \partial M) = (\int \omega. \text{real } (\text{upcrossings-before } X a b N \omega) \partial M)$
by (*force intro!: nn-integral-eq-integral integrable-upcrossings-before True adapted-process-axioms*)
moreover have $(\int^{+\omega}. \max 0 (X N \omega - a) \partial M) = (\int \omega. \max 0 (X N \omega - a) \partial M)$
using *Bochner-Integration.integrable-diff*[*OF integrable integrable-const*]
by (*force intro!: nn-integral-eq-integral*)
ultimately have $(b - a) * (\int^{+\omega}. \text{real } (\text{upcrossings-before } X a b N \omega) \partial M) \leq (\int^{+\omega}. \max 0 (X N \omega - a) \partial M)$
using *upcrossing-inequality*[*OF assms, of b a N*] **True ennreal-mult'**[*symmetric*]
by *simp*


```

    }
    ultimately show ?thesis by (force intro!: Sup-mono)
qed (simp add: ennreal-neg)
qed
end
end

```

4 Doob's First Martingale Convergence Theorem

```

theory Doob-Convergence
  imports Upcrossing
begin

context nat-finite-filtered-measure
begin

```

Doob's martingale convergence theorem states that, if we have a submartingale where the supremum over the mean of the positive parts is finite, then the limit process exists almost surely and is integrable. Furthermore, the limit process is measurable with respect to the smallest σ -algebra containing all of the σ -algebras in the filtration. The argumentation below is taken mostly from [3].

```

theorem submartingale-convergence-AE:
  fixes X :: nat  $\Rightarrow$  'a  $\Rightarrow$  real
  assumes submartingale M F 0 X
    and  $\bigwedge n. (\int \omega. \max 0 (X n \omega) \partial M) \leq C$ 
  obtains  $X_{lim}$  where AE  $\omega$  in M.  $(\lambda n. X n \omega) \longrightarrow X_{lim} \omega$ 
    integrable M  $X_{lim}$ 
     $X_{lim} \in \text{borel-measurable } (F_\infty)$ 

```

```

proof -
  interpret submartingale-linorder M F 0 X unfolding submartingale-linorder-def
  by (rule assms)

```

— We first show that the number of upcrossings has to be finite using the upcrossing inequality we proved above.

```

  have finite-upcrossings: AE  $\omega$  in M. upcrossings X a b  $\omega \neq \infty$  if  $a < b$  for a b
  proof -
    have C-nonneg:  $C \geq 0$  using assms(2) by (meson Bochner-Integration.integral-nonneg
linorder-not-less max.cobounded1 order-less-le-trans)
    {
      fix n
      have  $(\int^{+\omega}. \max 0 (X n \omega - a) \partial M) \leq (\int^{+\omega}. \max 0 (X n \omega) + |a| \partial M)$ 
    }
  by (fastforce intro: nn-integral-mono ennreal-leI)
    also have ... =  $(\int^{+\omega}. \max 0 (X n \omega) \partial M) + |a| * \text{emeasure } M (\text{space } M)$ 
  by (simp add: nn-integral-add)

```

also have ... = $(\int \omega. \max 0 (X n \omega) \partial M) + |a| * \text{emeasure } M \text{ (space } M)$
using *integrable by (simp add: nn-integral-eq-integral)*
also have ... $\leq C + |a| * \text{emeasure } M \text{ (space } M)$ **using** *assms(2) ennreal-leI*
by *simp*
finally have $(\int^+ \omega. \max 0 (X n \omega - a) \partial M) \leq C + |a| * \text{enn2real (emeasure } M \text{ (space } M))$ **using** *finite-emeasure-space C-nonneg by (simp add: ennreal-enn2real-if ennreal-mult)*
}
hence $(\text{SUP } N. \int^+ x. \text{ennreal (max 0 (X N x - a)) } \partial M) / (b - a) \leq \text{ennreal (C + |a| * enn2real (emeasure } M \text{ (space } M)) / (b - a)$ **by** *(fast intro: divide-right-mono-ennreal Sup-least)*
moreover have $\text{ennreal (C + |a| * enn2real (emeasure } M \text{ (space } M)) / (b - a) < \infty$ **using** *that C-nonneg by (subst divide-ennreal) auto*
moreover have $\text{integral}^N M \text{ (upcrossings } X a b) \leq (\text{SUP } N. \int^+ x. \text{ennreal (max 0 (X N x - a)) } \partial M) / (b - a)$
using *upcrossing-inequality-Sup[OF assms(1), of b a, THEN divide-right-mono-ennreal, of b - a]*
ennreal-mult-divide-eq mult.commute[of ennreal (b - a)] **that by** *simp*
ultimately show *?thesis using upcrossings-measurable adapted-process-axioms that by (intro nn-integral-noteq-infinite) auto*
qed

— Since the number of upcrossings are finite, limsup and liminf have to agree almost everywhere. To show this we consider the following countable set, which has zero measure.

define *S* **where** $S = ((\lambda(a :: \text{real}, b). \{\omega \in \text{space } M. \text{liminf } (\lambda n. \text{ereal } (X n \omega)) < \text{ereal } a \wedge \text{ereal } b < \text{limsup } (\lambda n. \text{ereal } (X n \omega))\}) ' \{(a, b) \in \mathbb{Q} \times \mathbb{Q}. a < b\})$

have $(0, 1) \in \{(a :: \text{real}, b). (a, b) \in \mathbb{Q} \times \mathbb{Q} \wedge a < b\}$ **unfolding** *Rats-def* **by** *simp*

moreover have *countable* $\{(a, b). (a, b) \in \mathbb{Q} \times \mathbb{Q} \wedge a < b\}$ **by** *(blast intro: countable-subset[OF - countable-SIGMA[OF countable-rat countable-rat]])*

ultimately have *from-nat-into-S: range (from-nat-into S) = S* *from-nat-into S* $n \in S$ **for** *n*

unfolding *S-def*

by *(auto intro!: range-from-nat-into from-nat-into simp only: Rats-def)*

{

fix *a b :: real*

assume *a-less-b: a < b*

then obtain *N* **where** $N: x \in \text{space } M - N \implies \text{upcrossings } X a b x \neq \infty$ $N \in \text{null-sets } M$ **for** *x* **using** *AE-E3[OF finite-upcrossings]* **by** *blast*

{

fix *ω*

assume *liminf-limsup: liminf (λn. X n ω) < a < b < limsup (λn. X n ω)*

have *upcrossings X a b ω = ∞*

proof —

{

fix *n*

```

have  $\exists m. \text{upcrossings-before } X \ a \ b \ m \ \omega \geq n$ 
proof (induction n)
  case 0
    have  $\text{Sup } \{n. \text{upcrossing } X \ a \ b \ 0 \ n \ \omega < 0\} = 0$  by simp
    then show ?case unfolding upcrossings-before-def by blast
  next
    case (Suc n)
      then obtain m where  $m: n \leq \text{upcrossings-before } X \ a \ b \ m \ \omega$  by blast
      obtain l where  $l: l \geq m \ X \ l \ \omega < a$  using liminf-upper-bound[OF
liminf-limsup(1), of m] nless-le by auto
      obtain u where  $u: u \geq l \ X \ u \ \omega > b$  using limsup-lower-bound[OF
liminf-limsup(2), of l] nless-le by auto
      show ?case using upcrossings-before-less-exists-upcrossing[OF a-less-b,
where ?X=X, OF l u] m by (metis Suc-leI le-neq-implies-less)
      qed
    }
  thus ?thesis unfolding upcrossings-def by (simp add: ennreal-SUP-eq-top)
  qed
}
hence  $\{\omega \in \text{space } M. \text{liminf } (\lambda n. \text{ereal } (X \ n \ \omega)) < \text{ereal } a \wedge \text{ereal } b < \text{limsup}$ 
 $(\lambda n. \text{ereal } (X \ n \ \omega))\} \subseteq N$  using N by blast
  moreover have  $\{\omega \in \text{space } M. \text{liminf } (\lambda n. \text{ereal } (X \ n \ \omega)) < \text{ereal } a \wedge \text{ereal } b$ 
 $< \text{limsup } (\lambda n. \text{ereal } (X \ n \ \omega))\} \cap N \in \text{null-sets } M$  by (force intro: null-set-Int1[OF
N(2)])
  ultimately have  $\text{emeasure } M \ \{\omega \in \text{space } M. \text{liminf } (\lambda n. \text{ereal } (X \ n \ \omega)) < a$ 
 $\wedge b < \text{limsup } (\lambda n. \text{ereal } (X \ n \ \omega))\} = 0$  by (simp add: Int-absorb1 Int-commute
null-setsD1)
  }
hence  $\text{emeasure } M \ (\text{from-nat-into } S \ n) = 0$  for n using from-nat-into-S(2)[of
n] unfolding S-def by force
  moreover have  $S \subseteq M$  unfolding S-def by force
  ultimately have  $\text{emeasure } M \ (\bigcup (\text{range } (\text{from-nat-into } S))) = 0$  using from-nat-into-S
by (intro emeasure-UN-eq-0) auto
  moreover have  $(\bigcup S) = \{\omega \in \text{space } M. \text{liminf } (\lambda n. \text{ereal } (X \ n \ \omega)) \neq \text{limsup}$ 
 $(\lambda n. \text{ereal } (X \ n \ \omega))\}$  (is ?L = ?R)
  proof –
    {
      fix  $\omega$ 
      assume asm:  $\omega \in ?L$ 
      then obtain a b :: real where  $a < b \ \text{liminf } (\lambda n. \text{ereal } (X \ n \ \omega)) < \text{ereal } a \wedge$ 
 $\text{ereal } b < \text{limsup } (\lambda n. \text{ereal } (X \ n \ \omega))$  unfolding S-def by blast
      hence  $\text{liminf } (\lambda n. \text{ereal } (X \ n \ \omega)) \neq \text{limsup } (\lambda n. \text{ereal } (X \ n \ \omega))$  using
ereal-less-le order.asym by fastforce
      hence  $\omega \in ?R$  using asm unfolding S-def by blast
    }
  moreover
  {
    fix  $\omega$ 
    assume asm:  $\omega \in ?R$ 

```

hence $\liminf (\lambda n. \text{ereal } (X \ n \ \omega)) < \limsup (\lambda n. \text{ereal } (X \ n \ \omega))$ **using**
Liminf-le-Limsup[of sequentially] *less-eq-ereal-def* **by** *auto*
then obtain a' **where** $a': \liminf (\lambda n. \text{ereal } (X \ n \ \omega)) < \text{ereal } a' \ \text{ereal } a' <$
 $\limsup (\lambda n. \text{ereal } (X \ n \ \omega))$ **using** *ereal-dense2* **by** *blast*
then obtain b' **where** $b': \text{ereal } a' < \text{ereal } b' \ \text{ereal } b' < \limsup (\lambda n. \text{ereal } (X$
 $n \ \omega))$ **using** *ereal-dense2* **by** *blast*
hence $a' < b'$ **by** *simp*
then obtain a **where** $a: a \in \mathbb{Q} \ a' < a \ a < b'$ **using** *Rats-dense-in-real* **by**
blast
then obtain b **where** $b: b \in \mathbb{Q} \ a < b \ b < b'$ **using** *Rats-dense-in-real* **by**
blast
have $\liminf (\lambda n. \text{ereal } (X \ n \ \omega)) < \text{ereal } a$ **using** $a \ a'$ *le-ereal-less or-*
der-less-imp-le **by** *meson*
moreover have $\text{ereal } b < \limsup (\lambda n. \text{ereal } (X \ n \ \omega))$ **using** $b \ b'$ *or-*
der-less-imp-le *ereal-less-le* **by** *meson*
ultimately have $\omega \in ?L$ **unfolding** *S-def* **using** $a \ b$ *asm* **by** *blast*
}
ultimately show *?thesis* **by** *blast*
qed
ultimately have $\text{emeasure } M \ \{\omega \in \text{space } M. \liminf (\lambda n. \text{ereal } (X \ n \ \omega)) \neq \limsup$
 $(\lambda n. \text{ereal } (X \ n \ \omega))\} = 0$ **using** *from-nat-into-S* **by** *argo*
hence *liminf-limsup-AE*: $AE \ \omega$ *in* $M. \liminf (\lambda n. X \ n \ \omega) = \limsup (\lambda n. X \ n \ \omega)$
by (*intro* *AE-iff-measurable*[*THEN* *iffD2*, *OF* - *refl*]) *auto*
hence *convergent-AE*: $AE \ \omega$ *in* $M. \text{convergent } (\lambda n. \text{ereal } (X \ n \ \omega))$ **using** *con-*
vergent-ereal **by** *fastforce*

— Hence the limit exists almost everywhere.

have *bounded-pos-part*: $\text{ennreal } (\int \omega. \text{max } 0 \ (X \ n \ \omega) \ \partial M) \leq \text{ennreal } C$ **for** n
using *assms(2)* *ennreal-leI* **by** *blast*

— Integral of positive part is $< \infty$.

{
fix ω
assume *asm*: $\text{convergent } (\lambda n. \text{ereal } (X \ n \ \omega))$
hence $(\lambda n. \text{max } 0 \ (\text{ereal } (X \ n \ \omega))) \longrightarrow \text{max } 0 \ (\lim (\lambda n. \text{ereal } (X \ n \ \omega)))$
using *convergent-LIMSEQ-iff isCont-tendsto-compose continuous-max contin-*
uous-const continuous-ident continuous-at-e2ennreal
by *fast*
hence $(\lambda n. \text{e2ennreal } (\text{max } 0 \ (\text{ereal } (X \ n \ \omega)))) \longrightarrow \text{e2ennreal } (\text{max } 0 \ (\lim$
 $(\lambda n. \text{ereal } (X \ n \ \omega))))$
using *isCont-tendsto-compose continuous-at-e2ennreal* **by** *blast*
moreover have $\lim (\lambda n. \text{e2ennreal } (\text{max } 0 \ (\text{ereal } (X \ n \ \omega)))) = \text{e2ennreal } (\text{max}$
 $0 \ (\lim (\lambda n. \text{ereal } (X \ n \ \omega))))$ **using** *limI* *calculation* **by** *blast*
ultimately have $\text{e2ennreal } (\text{max } 0 \ (\liminf (\lambda n. \text{ereal } (X \ n \ \omega)))) = \liminf$
 $(\lambda n. \text{e2ennreal } (\text{max } 0 \ (\text{ereal } (X \ n \ \omega))))$ **using** *convergent-liminf-cl* **by** (*metis* *asm*
convergent-def *limI*)
}

hence $(\int^{+\omega}. e2ennreal (\max 0 (\liminf (\lambda n. ereal (X n \omega)))) \partial M) = (\int^{+\omega}. \liminf (\lambda n. e2ennreal (\max 0 (ereal (X n \omega)))) \partial M)$ **using** *convergent-AE* **by** (*fast intro: nn-integral-cong-AE*)
moreover have $(\int^{+\omega}. \liminf (\lambda n. e2ennreal (\max 0 (ereal (X n \omega)))) \partial M) \leq \liminf (\lambda n. (\int^{+\omega}. e2ennreal (\max 0 (ereal (X n \omega)))) \partial M)$
by (*intro nn-integral-liminf*) *auto*
moreover have $(\int^{+\omega}. e2ennreal (\max 0 (ereal (X n \omega)))) \partial M = ennreal (\int \omega. \max 0 (X n \omega) \partial M)$ **for** n
using *e2ennreal-ereal ereal-max-0*
by (*subst nn-integral-eq-integral[symmetric]*) (*fastforce intro!: nn-integral-cong integrable | presburger*)
moreover have *liminf-pos-part-finite*: $\liminf (\lambda n. ennreal (\int \omega. \max 0 (X n \omega) \partial M)) < \infty$
unfolding *liminf-SUP-INF*
using *Inf-lower2[OF - bounded-pos-part]*
by (*intro order.strict-trans1[OF Sup-least, of - ennreal C]*) (*metis (mono-tags, lifting) atLeast-iff imageE image-eqI order.refl, simp*)
ultimately have *pos-part-finite*: $(\int^{+\omega}. e2ennreal (\max 0 (\liminf (\lambda n. ereal (X n \omega)))) \partial M) < \infty$ **by** *force*

— Integral of negative part is $< \infty$.

$\{$
fix ω
assume *asm: convergent* $(\lambda n. ereal (X n \omega))$
hence $(\lambda n. - \min 0 (ereal (X n \omega))) \longrightarrow - \min 0 (\lim (\lambda n. ereal (X n \omega)))$
using *convergent-LIMSEQ-iff isCont-tendsto-compose continuous-min continuous-const continuous-ident continuous-at-e2ennreal*
by *fast*
hence $(\lambda n. e2ennreal (- \min 0 (ereal (X n \omega)))) \longrightarrow e2ennreal (- \min 0 (\lim (\lambda n. ereal (X n \omega))))$
using *isCont-tendsto-compose continuous-at-e2ennreal* **by** *blast*
moreover have $\lim (\lambda n. e2ennreal (- \min 0 (ereal (X n \omega)))) = e2ennreal (- \min 0 (\lim (\lambda n. ereal (X n \omega))))$ **using** *limI calculation* **by** *blast*
ultimately have $e2ennreal (- \min 0 (\liminf (\lambda n. ereal (X n \omega)))) = \liminf (\lambda n. e2ennreal (- \min 0 (ereal (X n \omega))))$ **using** *convergent-liminf-cl* **by** (*metis asm convergent-def limI*)
 $\}$
hence $(\int^{+\omega}. e2ennreal (- \min 0 (\liminf (\lambda n. ereal (X n \omega)))) \partial M) = (\int^{+\omega}. \liminf (\lambda n. e2ennreal (- \min 0 (ereal (X n \omega)))) \partial M)$ **using** *convergent-AE* **by** (*fast intro: nn-integral-cong-AE*)
moreover have $(\int^{+\omega}. \liminf (\lambda n. e2ennreal (- \min 0 (ereal (X n \omega)))) \partial M) \leq \liminf (\lambda n. (\int^{+\omega}. e2ennreal (- \min 0 (ereal (X n \omega)))) \partial M)$
by (*intro nn-integral-liminf*) *auto*
moreover have $(\int^{+\omega}. e2ennreal (- \min 0 (ereal (X n \omega)))) \partial M = (\int \omega. \max 0 (X n \omega) \partial M) - (\int \omega. X n \omega \partial M)$ **for** n
proof —
have $*$: $(- \min 0 c) = \max 0 c - c$ **if** $c \neq \infty$ **for** $c :: ereal$ **using** *that* **by**

(cases $c \geq 0$) auto

hence $(\int^+ \omega. e2ennreal (- \min 0 (ereal (X n \omega))) \partial M) = (\int^+ \omega. e2ennreal (max 0 (ereal (X n \omega)) - (ereal (X n \omega))) \partial M)$ **by simp**

also have $\dots = (\int^+ \omega. ennreal (max 0 (X n \omega) - (X n \omega)) \partial M)$ **using** $e2ennreal-ereal \text{ereal-max-0} \text{ereal-minus}(1)$ **by** (intro nn-integral-cong) presburger

also have $\dots = (\int \omega. max 0 (X n \omega) - (X n \omega) \partial M)$ **using integrable by** (intro nn-integral-eq-integral) auto

finally show ?thesis **using** Bochner-Integration.integral-diff integrable **by simp qed**

moreover have $\liminf (\lambda n. ennreal ((\int \omega. max 0 (X n \omega) \partial M) - (\int \omega. X n \omega \partial M))) < \infty$

proof -

{

fix $n A$

assume $asm: ennreal ((\int \omega. max 0 (X n \omega) \partial M) - (\int \omega. X n \omega \partial M)) \in A$

have $(\int \omega. X 0 \omega \partial M) \leq (\int \omega. X n \omega \partial M)$ **using** set-integral-le[OF sets.top order-refl, of n] space-F **by** (simp add: integrable set-integral-space)

hence $(\int \omega. max 0 (X n \omega) \partial M) - (\int \omega. X n \omega \partial M) \leq C - (\int \omega. X 0 \omega \partial M)$ **using** $assms(2)[of n]$ **by argo**

hence $ennreal ((\int \omega. max 0 (X n \omega) \partial M) - (\int \omega. X n \omega \partial M)) \leq ennreal (C - (\int \omega. X 0 \omega \partial M))$ **using** $ennreal-leI$ **by blast**

hence $Inf A \leq ennreal (C - (\int \omega. X 0 \omega \partial M))$ **by** (rule Inf-lower2[OF asm])

}

thus ?thesis

unfolding $\liminf\text{-SUP-INF}$

by (intro order.strict-trans1[OF Sup-least, of - $ennreal (C - (\int \omega. X 0 \omega \partial M))$]) (metis (no-types, lifting) atLeast-iff imageE image-eqI order.refl order-trans, simp)

qed

ultimately have $neg\text{-part-finite}: (\int^+ \omega. e2ennreal (- (\min 0 (\liminf (\lambda n. ereal (X n \omega)))) \partial M) < \infty$ **by simp**

— Putting it all together now to show that the limit is integrable and $< \infty$ a.e.

have $e2ennreal |\liminf (\lambda n. ereal (X n \omega))| = e2ennreal (max 0 (\liminf (\lambda n. ereal (X n \omega)))) + e2ennreal (- (\min 0 (\liminf (\lambda n. ereal (X n \omega)))))$ **for** ω

unfolding $ereal\text{-abs-max-min}$

by (simp add: eq-onp-same-args max-def plus-ennreal.abs-eq)

hence $(\int^+ \omega. e2ennreal |\liminf (\lambda n. ereal (X n \omega))| \partial M) = (\int^+ \omega. e2ennreal (max 0 (\liminf (\lambda n. ereal (X n \omega)))) \partial M) + (\int^+ \omega. e2ennreal (- (\min 0 (\liminf (\lambda n. ereal (X n \omega)))) \partial M)$ **by** (auto intro: nn-integral-add)

hence $nn\text{-integral-finite}: (\int^+ \omega. e2ennreal |\liminf (\lambda n. ereal (X n \omega))| \partial M) \neq \infty$ **using** $pos\text{-part-finite} \text{neg-part-finite}$ **by auto**

hence $finite\text{-AE}: AE \omega \text{ in } M. e2ennreal |\liminf (\lambda n. ereal (X n \omega))| \neq \infty$ **by** (intro nn-integral-noteq-infinite) auto

moreover

{

fix ω

assume $asm: \liminf (\lambda n. X n \omega) = \limsup (\lambda n. X n \omega) |\liminf (\lambda n. ereal (X$

$n \omega)) \neq \infty$
hence $(\lambda n. X n \omega) \longrightarrow \text{real-of-ereal } (\text{liminf } (\lambda n. X n \omega))$ **using** *lim-sup-le-liminf-real ereal-real'* **by** *simp*
}
ultimately have *converges: AE ω in M . $(\lambda n. X n \omega) \longrightarrow \text{real-of-ereal } (\text{liminf } (\lambda n. X n \omega))$* **using** *liminf-limsup-AE* **by** *fastforce*
{
fix ω
assume *e2ennreal $|\text{liminf } (\lambda n. \text{ereal } (X n \omega))| \neq \infty$*
hence *$|\text{liminf } (\lambda n. \text{ereal } (X n \omega))| \neq \infty$* **by** *force*
hence *e2ennreal $|\text{liminf } (\lambda n. \text{ereal } (X n \omega))| = \text{ennreal } (\text{norm } (\text{real-of-ereal } (\text{liminf } (\lambda n. \text{ereal } (X n \omega))))))$* **by** *fastforce*
}
hence $(\int^+ \omega. \text{e2ennreal } |\text{liminf } (\lambda n. \text{ereal } (X n \omega))| \partial M) = (\int^+ \omega. \text{ennreal } (\text{norm } (\text{real-of-ereal } (\text{liminf } (\lambda n. \text{ereal } (X n \omega)))))) \partial M$ **using** *finite-AE* **by** *(fast intro: nn-integral-cong-AE)*
hence $(\int^+ \omega. \text{ennreal } (\text{norm } (\text{real-of-ereal } (\text{liminf } (\lambda n. \text{ereal } (X n \omega)))))) \partial M < \infty$ **using** *nn-integral-finite* **by** *(simp add: order-less-le)*
hence *integrable M $(\lambda \omega. \text{real-of-ereal } (\text{liminf } (\lambda n. X n \omega)))$* **by** *(intro integrableI-bounded) auto*
moreover have $(\lambda \omega. \text{real-of-ereal } (\text{liminf } (\lambda n. X n \omega))) \in \text{borel-measurable } F_\infty$
using *borel-measurable-liminf[OF F-infinity-measurableI]* **adapted by** *measurable*
ultimately show *?thesis using that converges by presburger*
qed

— We state the theorem again for martingales and supermartingales.

corollary *supermartingale-convergence-AE:*

fixes $X :: \text{nat} \Rightarrow 'a \Rightarrow \text{real}$
assumes *supermartingale $M F 0 X$*
and $\bigwedge n. (\int \omega. \text{max } 0 \ (- X n \omega) \partial M) \leq C$
obtains X_{lim} **where** *AE ω in M . $(\lambda n. X n \omega) \longrightarrow X_{lim} \omega$*
integrable $M X_{lim}$
 $X_{lim} \in \text{borel-measurable } (F_\infty)$

proof —

obtain Y **where** **: AE ω in M . $(\lambda n. - X n \omega) \longrightarrow Y \omega$* *integrable $M Y Y$*
 $\in \text{borel-measurable } (F_\infty)$
using *supermartingale.uminus[OF assms(1), THEN submartingale-convergence-AE]*
assms(2) **by** *auto*
hence *AE ω in M . $(\lambda n. X n \omega) \longrightarrow (- Y) \omega$* *integrable $M (- Y) - Y \in$*
borel-measurable (F_∞)
using *isCont-tendsto-compose[OF isCont-minus, OF continuous-ident]* *integrable-minus borel-measurable-uminus* **unfolding** *fun-Compl-def* **by** *fastforce+*
thus *?thesis using that[of $- Y$]* **by** *blast*
qed

corollary *martingale-convergence-AE:*

fixes $X :: \text{nat} \Rightarrow 'a \Rightarrow \text{real}$

assumes *martingale M F 0 X*
and $\bigwedge n. (\int \omega. |X\ n\ \omega| \partial M) \leq C$
obtains X_{lim} **where** *AE ω in M.* $(\lambda n. X\ n\ \omega) \longrightarrow X_{lim}\ \omega$
integrable M X_{lim}
 $X_{lim} \in \text{borel-measurable}(F_\infty)$

proof –

interpret *martingale-linorder M F 0 X* **unfolding** *martingale-linorder-def* **by**
(rule assms)

have $\max 0 (X\ n\ \omega) \leq |X\ n\ \omega|$ **for** n **by** *linarith*

hence $(\int \omega. \max 0 (X\ n\ \omega) \partial M) \leq C$ **for** n **using** *assms(2)[THEN dual-order.trans, OF integral-mono, OF integrable-max]* **integrable** **by** *fast*

thus *?thesis using that submartingale-convergence-AE[OF submartingale-axioms]*
by *blast*

qed

corollary *martingale-nonneg-convergence-AE:*

fixes $X :: \text{nat} \Rightarrow 'a \Rightarrow \text{real}$

assumes *martingale M F 0 X* $\bigwedge n. \text{AE } \omega \text{ in } M. X\ n\ \omega \geq 0$

obtains X_{lim} **where** *AE ω in M.* $(\lambda n. X\ n\ \omega) \longrightarrow X_{lim}\ \omega$
integrable M X_{lim}
 $X_{lim} \in \text{borel-measurable}(F_\infty)$

proof –

interpret *martingale-linorder M F 0 X* **unfolding** *martingale-linorder-def* **by**
(rule assms)

have *AE ω in M.* $\max 0 (- X\ n\ \omega) = 0$ **for** n **using** *assms(2)[of n]* **by** *force*

hence $(\int \omega. \max 0 (- X\ n\ \omega) \partial M) \leq 0$ **for** n **by** *(simp add: integral-eq-zero-AE)*

thus *?thesis using that supermartingale-convergence-AE[OF supermartingale-axioms]*
by *blast*

qed

end

end

References

- [1] R. Degenne and K. Ying. A Formalization of Doob’s Martingale Convergence Theorems in mathlib. In *Proceedings of the 12th ACM SIG-PLAN International Conference on Certified Programs and Proofs*. Association for Computing Machinery, New York, United States, 2022. [arXiv:2212.05578](https://arxiv.org/abs/2212.05578).
- [2] R. Degenne and K. Ying. probability.martingale.basic - mathlib, 2022. https://leanprover-community.github.io/mathlib_docs/probability/martingale/basic.html, Last Accessed: 15 Feb 2024.

- [3] R. Durrett. *Probability: Theory and Examples*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019. URL: <https://books.google.de/books?id=b22MDwAAQBAJ>.
- [4] G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, 2020.
- [5] J. Hölzl and A. Heller. Three Chapters of Measure Theory in Isabelle/HOL. In M. C. J. D. van Eekelen, H. Geuvers, J. Schmaltz, and F. Wiedijk, editors, *Interactive Theorem Proving (ITP 2011)*, volume 6898 of *LNCS*, pages 135–151, 2011. doi:10.1007/978-3-642-22863-6_12.
- [6] A. Keskin. A Formalization of Martingales in Isabelle/HOL. Bachelor’s thesis, Technical University of Munich, 2023. arXiv:2311.06188.
- [7] A. Keskin. Martingales. *Archive of Formal Proofs*, November 2023. <https://isa-afp.org/entries/Martingales.html>, Formal proof development.