On Probability, Measure, and Integration in HOL4

Aaron R. Coble

March 1, 2009

"You cannot avoid measure theory"
– David Williams, Probability with Martingales

Probability theory is inextricably connected to measure and integration theories; these two theories must be formalised before probability theory can be developed in a theorem prover. This text examines the shortcomings of previous formalisations in HOL4 and presents an new formalisation.

1 Motivation

As David Williams notes in *Probability with Martingales*, "you cannot avoid measure theory" in a mathematically rigourous treatment of probability theory. Measure theory is necessary for two reasons. Most importantly, it ensures mathematical consistency of the formalised definitions for probability theory. If measure theory were not used, inconsistencies based on the Banach-Tarski paradox could be introduced. Banach and Tarski [14] proved that it is possible to define a non-measurable set if the axiom of choice is assumed. A non-measurable set is a set for which no measure can be defined without resulting in a contradiction; non-measurable sets are outside the scope in which probability theory operates. Since the axiom of choice is assumed for most of conventional mathematics, as well as higher-order logic, Banach and Tarski's paradox must be considered. If we were to define a probability measure outright and then apply this definition to a non-measurable set, an inconsistency would arise. Any proofs involving such a definition would be meaningless, because a contradiction could be derived from the definition itself.

In order to avoid the Banach-Tarski paradox and ensure the correctness of proofs that use the formalisations in this text, probability measures must not

be applied to non-measurable sets. This can be achieved by defining probability theory as an extension of measure theory; this theory defines the sets that are measurable with respect to a given measure. Constructing new definitions incrementally as extensions of others prevents inconsistencies and maintains a straightforward correspondence between formalised theories and textbook definitions.

Another motivation for using measure theory in this chapter is to provide a mathematical link between definitions involving discrete probability measures and continuous probability measures. This unification of discrete and continuous probability under one general theory was a driving force behind the development of Kolmogorov's measure-theoretic treatment in the first half of the twentieth century [8]. Prior to his work, probability theory was marginalised by many mathematicians as lacking mathematical rigour. Using measure theory, definitions can be formalised that generalise the discrete and continuous cases, providing a concrete connection between the two.

General definitions also have practical implications for formalisation work. Often, they are easier to work with because mechanised proofs involving such definitions are not complicated by unnecessary details. On numerous occasions, I have struggled with a mechanised proof only to find that it could be proved easily in a more general form and then applied to the particular case of interest. By using more general definitions, the formalisation developed in the theorem prover can be applied to a wider range of future uses.

Let's revisit Williams's quote from *Probability with Martingales*, this time in its entirety:

> "You cannot avoid measure theory: an *event* in probability is a measurable set, a *random variable* is a measurable function on the sample space, the *expectation* of a random variable is its integral with respect to the probability measure; and so on."

Williams's words accurately capture the inseparability of the topics of probability, measure, and integration that form the core of this chapter. Some definitions in a measure-theoretic treatment of probability theory are constructed using (Lebesgue) integration. For example, both expectation and conditional expectation are defined using the Lebesgue integral and are needed to define concepts in information theory. Thus, just as measure theory must be formalised before information theory, so must integration theory.

The work presented here is not the first effort to formalise probability theory in a theorem prover nor has the importance of measure and integration theories gone unnoticed in previous work. However, the formalisations in this chapter

offer significant advantages over previous developments; in the next section, we will examine these contributions in comparison with past work.

2 Related work and novel contributions

Hurd's formalisations of measure and probability theories in HOL4 [7] and Richter's formalisation of Lebesgue integration in the Isabelle theorem-prover [13] have served as a guide for the work presented below. Also noteworthy are Białas's and Nędzusiak's [1, 2, 11, 12] formalisations of measure theory and probability theory in the Mizar theorem prover, Hasan's [5, 6] extensions of Hurd's work to include expected value, Harrison's [4] formalisation of the gauge integral in HOL4, and Lester's work on topology in the PVS system [9].

2.1 Hurd's measure and probability theories in HOL4

Hurd [7] developed a formalisation of measure theory in HOL4, upon which he constructed definitions for probability spaces and functions on them. Hurd then used his formalisation to verify the correctness of probabilistic algorithms; his most interesting application was the verification of the Miller-Rabin primality test.

While Hurd's work [7] was a major milestone for machine-verification of probabilistic algorithms, the scope of that work was limited. He formalised many definitions from probability theory and proved important results about independent functions on probability spaces. Despite these contributions, his work did not include some probability-theoretic concepts needed to define information theory. For example, Hurd proved that the probability of the empty event is 0, the probability of the set of all events is 1, and a probability measure is countably additive. However, he did not formalise definitions for random variables, expectation, or conditional expectation.

Hurd's formalisations restrict the measure and probability spaces that can be constructed. A measure or probability space is a triple $(S, \mathbb{S}, \mu)$ consisting of a set called the space (S), a set of subsets of the space known as the measurable-sets or events ($\mathbb{S}$), and a (probability) measure (μ). Hurd's definitions do not explicitly include the space, which he specifies implicitly as the universal set of the appropriate HOL type. The universal set of a HOL type α is defined as

$$\texttt{UNIV} = \{\texttt{x} : \alpha \mid \top\},$$

i.e. all the elements of type α. Hurd's formalisations are restricted to probability and measure spaces of the form $(\texttt{UNIV}, \mathbb{S}, \mu)$. In fact, he formalised measure

and probability spaces as pairs $(\mathbb{S}, \mu)$; the space on which they are defined is implied by the HOL type of the pair. This approach does not allow measure and probability spaces where S is not the universal set of a HOL type.

As mentioned above, one of the primary motivations for using measure theory was to generalise the definitions for continuous and discrete probability-measures under a single theory.[1] By formalising probability theory as an extension of measure theory, such general definitions can be used in the theorem prover. These definitions can then be proved equivalent to simpler definitions for discrete (i.e. countable) and continuous spaces. This approach has been taken below so that simpler definitions can be used wherever applicable.

Although similar equivalences can be proved in Hurd's formalisation, they cannot be used easily. For example, one might be interested in a probability space $(S, \mathbb{S}, \mu)$, where S is a countable set of elements of HOL type α. If the universal set of α is countable, then a definition involving $(S, \mathbb{S}, \mu)$ can be simplified to its countable form. Although S is countable, the universal set of α might not be.[2] In these cases, Hurd's formalisation makes it difficult to simplify a definition to its countable form. Before such a reduction can be applied, a new HOL type must be defined for the elements of S. This requires a considerable effort, so it is not feasible to undertake for each countable space of interest. For example, in order to define a HOL type for a countable subset of the reals, it is necessary to redefine arithmetic operations on this set and prove properties of these operations. These complications are avoided in the formalisation below, because spaces are explicitly specified.

The formalisations of measure theory and probability theory presented in this chapter are modeled after Hurd's restricted formalisations. Some of Hurd's definitions and proofs could be generalised with only minor modifications; however, many of Hurd's proofs were not valid for the general case and had to be reproved. Despite overlap with Hurd's work, the formalisations presented here broaden the applicability of formal methods to probabilistic algorithms and ease the difficulty of that analysis.

2.2 Richter's integration theory in Isabelle/HOL

Recall that Lebesgue integration is needed to define a number of concepts in probability theory. Amongst these are expectation and conditional expectation,

[1] Measure theory even generalises measures that are neither continuous, nor discrete, nor a mixture of the two.

[2] For instance, α might be the real numbers or tuples representing program executions as in Chapter **??**

which feature heavily in subsequent chapters. Richter [13] formalised Lebesgue integration in the Isabelle/HOL theorem prover. The formalisation of Lebesgue integration developed below is modelled after his work.

Unfortunately, Richter's formalisations are insufficient for the applications in this text on several accounts. Richter's work was guided by Hurd's and suffers from the same restrictions. In addition, his work allows only integration of functions that are measurable from the real numbers to the real numbers. The applications below require integration of functions that are measurable from a space of an arbitrary HOL-type to any subset of the real numbers. Finally, Richter developed his formalisation in the Isabelle/HOL system, while the work in this text uses the HOL4 system; translation between the two systems is non-trivial.

Thus, a formalisation of Lebesgue integration that generalises Richter's had to be developed in HOL4. This approach allows equivalences to be proved between the general definition of the Lebesgue integral and simpler forms for specific classes of spaces. These simplifications are new contributions to the formalisation of Lebesgue integration. Furthermore, the work presented below is the only formalisation of Lebesgue integration that has been developed in the HOL4 system.

2.3 Białas and Nędzusiak's probability theories in Mizar

Białas and Nędzusiak developed formalisations of measure theory and probability theory in the Mizar theorem prover [1, 2, 11, 12]. Hurd [7] states that his formalisation supersedes that of Białas and Nędzusiak in a number of respects. Since the formalisation presented here generalises Hurd's, it must also extend the work of Białas and Nędzusiak. A review of publications by Białas and Nędzusiak was insufficient to determine if their formalisations are subject to the same restrictions as Hurd's. Although I suspect that they are not, a detailed examination of their formalisations would be necessary to comment with certainty.[3]

2.4 Hasan's expectation in HOL4

Building upon Hurd's work, Hasan [5, 6] formalised definitions of expected value for discrete and continuous probability distributions. He then used his formalisa-

[3] I "discovered" the restriction in Hurd's formalisation when I noticed an incongruence between Hurd's definitions and textbook definitions. Later, I found that this restriction had been noted as a comment in the code of the formalisation, though not in the literature related to it.

tions to prove properties of various probability distributions such as the bernoulli distribution. Hasan's definitions of expected value were restricted to the discrete space of the natural numbers (a countable HOL type) and the continuous space of the real numbers. His formalisations were not particularly affected by the limitations of Hurd's because he was only considering spaces that are the universal set of a HOL type.

Hasan did not formalise a general definition of expected value using Lebesgue integration, so he could not formalise a mathematical connection between his definitions for the discrete case and the continuous case. The formalisations developed below generalise his by allowing arbitrary spaces (rather than only the naturals or the reals) and by defining expected value using Lebesgue integration; this definition generalises those for the discrete and continuous cases.

Hasan's work devoted to proving properties of well-known probability distributions is tangential to aims of this text. However, future work could be to reproduce those proofs using the more general framework developed in this chapter.

2.5 Harrison's gauge integral in HOL4

Harrison's formalisation of the real numbers in HOL4 [4] included a definition of the gauge integral for functions over the reals. His formalisation is not sufficiently general for the work in this chapter, which requires a definition of integration for functions from an arbitrary type to the real numbers. Moreover, Lebesgue integration is the natural choice for developing probability theory from measure theory and is used in most textbooks [3, 10, 15]. Thus, the Lebesgue integral has been selected for the formalisations developed below. Future work could include a proof of equivalence between the formalisation of integration in this chapter and Harrison's when considering functions over the reals.

2.6 Lester's topology and probability in PVS

Also noteworthy is Lester's work on topology in the PVS theorem prover [9]. Lester developed formalisations for measure theory and Lebesgue integration in PVS. He then built on that work to formalise some portion of probability theory. At the time that the formalisations in this chapter were completed, I was unaware of Lester's work. There is some overlap between his formalisations and the work presented here. However, a number of important contributions of this chapter do not appear in Lester's work; amongst these are the formalisation of conditional expectation and proofs of equivalence between general definitions

and simpler definitions for discrete spaces.

It is difficult to say precisely how much Lester's work overlaps with the formalisations developed below. One difficulty in comparing these two formalisations is that different approaches were taken for each. In this chapter, measures must be finite-valued (i.e. take values from the reals), but measurable functions are real-valued and may take negative values. Lester allowed measures to be infinite (i.e. take values from the extended-reals), but required measurable functions to be positive. Below, we will see that Lester's restrictions simplified the formalisation process; however, some generality was lost because functions taking both positive and negative values cannot be integrated. Whether Lester's restriction to positive measurable-functions or my restriction to finite measures is a greater limitation would depend on the specific application. Another difficulty in comparing Lester's work with this chapter is that a different proof assistant was used for each (PVS and HOL4 respectively).

$\star\star\star$

The remainder of this text describes the formalisation of measure theory, Lebesgue integration, and probability theory, within the context of the HOL4 theorem prover. Measure theory will be examined first, followed by Lebesgue integration, and ultimately probability theory. In general, the presentation will progress from the simplest to the most complex definitions for each theory. A number of properties of these formalisations have been proved in the theorem prover, some of which appear in this chapter. These proofs should reassure the reader that the formalisations behave as expected and appropriately capture the textbook definitions.

Definitions from measure, integration, and probability theories will be reviewed prior to their formalisations in order to facilitate easier reading. However, these definitions will not be discussed in great detail; more detailed presentations can be found in standard textbooks on those subjects. Doob's and Williams's books are an excellent starting point [3, 15]. Doob focuses more on the development of measure theory with applications to probability theory. Williams emphasises probability theory with a basis in measure theory. In general, the definitions from measure theory, Lebesgue integration, and probability theory found in this chapter are Doob's [3] restated to respect the notational conventions used here.

The presentation of the formalisations in this chapter avoids any unnecessary details about their implementation in the HOL4 theorem prover. However, implementation details are included where necessary or of particular interest.

In such cases, sufficient explanation is provided for someone unfamiliar with the syntax of HOL4. A glossary of HOL4 notation can be found in Appendix A.

3 Measure theory formalised in HOL4

This section is devoted to the formalisation of measure theory in higher-order logic, upon which the formalisations of Lebesgue integration (Section 4) and probability theory (Section 5) are built. Knowledge of this section will aid in understanding subsequent sections. As mentioned above, the formalisations developed here are modeled after and generalise Hurd's [7].

3.1 Subset classes and σ-algebras

Our investigation of measure theory in HOL begins with a study of important classes of subsets of a *space*. Before identifying particular classes of subsets, it is necessary to characterise what it means for a set to be a class of subsets of a particular space. To that end, we will review a definition for a *subset class* and then examine the formalisation of that definition in HOL.

Definition 1 (subset class)**.** $\mathbb{S}$ *is a* **subset class** *of a space S iff all the elements of $\mathbb{S}$ are subsets of S.*

The formalisation of Definition 1 in higher-order logic is straightforward as can be seen below.

Formalisation 1 (subset class)**.**

$$\texttt{subset\_class sp sts} \;=\; \forall\, \texttt{x}.\; \texttt{x IN sts} \;\Rightarrow\; \texttt{x SUBSET sp}.$$

We will now review a textbook definition for an important type of subset class known as an *algebra*; afterwards we will examine the formalisation of this definition in higher-order logic.

Definition 2 (algebra)**.** *A class $\mathbb{S}$ of subsets of a space S define an* **algebra** *iff*

(i) $\mathbb{S}$ *contains the empty set,*
(ii) $\mathbb{S}$ *is closed under complementation (within S), and*
(iii) $\mathbb{S}$ *is closed under finite unions.*

Note that conditions (ii) and (iii) together are equivalent to condition (ii) and the condition that $\mathbb{S}$ is closed under finite intersections. Thus, an algebra is necessarily closed under both finite unions and finite intersections. Definition 2 has been formalised in higher-order logic as

Formalisation 2 (algebra)**.**

$$\begin{aligned}\texttt{algebra}\ (\texttt{sp},\texttt{sts}) \ =\ & \texttt{subset\_class sp sts} \wedge \\ & \{\}\ \texttt{IN sts} \wedge (\forall\ \texttt{s}.\ \texttt{s IN sts} \Rightarrow \texttt{sp DIFF s IN sts}) \wedge \\ & (\forall\ \texttt{s t}.\ \texttt{s IN sts} \wedge \texttt{t IN sts} \Rightarrow \texttt{s UNION t IN sts}),\end{aligned}$$

where $\{\}$ *represents the empty set and* DIFF *and* UNION *are the set-difference and -union operations respectively. Note that within a space* S*, the complement of a set* $s \subseteq S$ *is the difference of* S *and* s*.*

A simple line-by-line comparison reveals that Formalisation 2 captures Definition 2. The first line of the formalisation captures the implicit condition in Definition 2 that sp and sts form a subset class. A straightforward correspondence between textbook definitions and their formalisations has been maintained wherever possible.

Let's digress for a moment to examine a concrete example of Hurd's restriction on spaces in his formalisation of measure theory [7]. As explained in Section 2, the differences between Hurd's formalisations and those in this chapter are subtle, but they have a substantial impact on the respective formalisations. Hurd's formalisation requires that all spaces considered are the universal set of some HOL type (i.e. the set of all elements of that type). For example, Hurd's formalisation of Definition 2 above is

$$\begin{aligned}\texttt{algebra sts} \ =\ & \{\}\ \texttt{IN sts} \wedge \\ & (\forall\ \texttt{s}.\ \texttt{s IN sts} \Rightarrow \texttt{COMPL s IN sts}) \wedge \\ & (\forall\ \texttt{s t}.\ \texttt{s IN sts} \wedge \texttt{t IN sts} \Rightarrow \texttt{s UNION t IN sts}),\end{aligned}$$

where COMPL s is the set-complementation operation equal to UNIV DIFF s.[4] Notice that Hurd's formalisation does not require that sts is a class of subsets of UNIV. That requirement is trivially true in his formalisation because all sets of a given type are subsets of the universal set of that type.

We now continue looking at important classes of subsets by reviewing a definition for a *σ-algebra* followed by its formalisation in higher-order logic.

Definition 3 (σ-algebra)**.** *A subset class* $\mathbb{S}$ *of a space* S *defines a* **σ-algebra** *iff* $(S, \mathbb{S})$ *defines an algebra and* $\mathbb{S}$ *is closed under* countable *(possibly infinite) unions.*

Formalisation 3 (σ-algebra)**.**

$$\begin{aligned}\texttt{sigma\_algebra}\ (\texttt{sp},\texttt{sts}) \ =\ & \texttt{algebra}\ (\texttt{sp},\texttt{sts}) \wedge \\ & (\forall\ \texttt{c}.\ \texttt{countable c} \wedge \texttt{c SUBSET sts} \Rightarrow \texttt{BIGUNION c IN sts}),\end{aligned}$$

[4] i.e. the difference between the universal set of the type of s and s

where `BIGUNION S` *is the union operation applied over the elements of a set of sets* `S` *and is equivalent to the mathematical notation* $\bigcup_{x \in S} x$.

For a space S, $(S, \emptyset)$ is the smallest σ-algebra and $(S, \mathcal{P}(S))$ the largest, where $\emptyset$ and $\mathcal{P}(S)$ denote the empty set and the powerset of S respectively.

Typically, $\sigma(S, \mathbb{G})$ is used to denote the smallest σ-algebra defined on a space S and containing a generating set of subsets $\mathbb{G}$. The function constructing this smallest σ-algebra can be defined in HOL as follows:

Formalisation 4 ($\sigma(S, \mathbb{G})$)**.**

$$\texttt{sigma}\ (\texttt{sp}, \texttt{sts}) = (\texttt{sp}, \texttt{BIGINTER}\ \{\texttt{s} \mid \texttt{sts}\ \texttt{SUBSET}\ \texttt{s} \wedge \texttt{sigma\_algebra}\ (\texttt{sp}, \texttt{s})\}),$$

where `BIGINTER s` *is the set of elements that are in all the sets in s. This reassuring property has been proved about* `sigma` *in HOL4:*

$$\forall\ \texttt{sp}\ \texttt{sts}.\ \texttt{subset\_class}\ \texttt{sp}\ \texttt{sts} \Rightarrow \texttt{sigma\_algebra}\ (\texttt{simga}\ (\texttt{sp}, \texttt{sts}))$$

i.e. `sigma` (`sp`, `sts`) *defines a* σ*-algebra assuming* `sts` *is a class of subsets of* `sp`*.*

3.2 Measure spaces

One of the most important developments presented in this section is the HOL definition of a *measure space*. Before the definition of a measure space can be formalised, it is first necessary to formalise the properties of *positivity* and *countable additivity*, which measure spaces must satisfy. We now review definitions for those properties and then examine their formalisations in higher-order logic.

Definition 4 (positivity)**.** *Let* $\mathbb{S}$ *be a class of subsets of space* S *containing the empty set and* λ *a function from* $\mathbb{S}$ *to* $\mathbb{R}$*. Then* λ *is* **positive** *with respect to* $(S, \mathbb{S})$ *iff* $\lambda(s) \geq 0$ *for any* $s \in \mathbb{S}$ *and* $\lambda(\emptyset) = 0$*.*

Formalisation 5 (positivity)**.**

$$\begin{aligned}&\texttt{positive}\ (\texttt{sp}, \texttt{sts}, \texttt{lambda}) = \\ &\quad (\texttt{lambda}\ \{\} = \texttt{0}) \wedge (\forall\ \texttt{s}.\ \texttt{s}\ \texttt{IN}\ \texttt{sts} \Rightarrow \texttt{0} \leq \texttt{lambda}\ \texttt{s}).\end{aligned}$$

Definition 5 (countable additivity)**.** *Let* $\mathbb{S}$ *be a class of subsets of space* S *and let* $\mathbb{S}$ *contain the empty set. Let* λ *be a function from* $\mathbb{S}$ *to* $\mathbb{R}$*. Then* λ *is* **countably additive** *with respect to* $(S, \mathbb{S})$ *iff*

$$\lim_{n \to \infty} \sum_{j=0}^{n} \lambda(s_j) \to \lambda(\bigcup_i s_i),$$

for any sequence $s_0, s_1, \ldots$ *of elements of* $\mathbb{S}$ *such that* $\bigcup_i s_i \in \mathbb{S}$ *and all* s_j *and* s_k *are* disjoint *when* $j \neq k$.

Formalisation 6 (countable additivity)**.**

```
countably_additive (sp, sts, lambda) =
  ∀ (f : num → sts). (∀ m n. m ≠ n ⇒ DISJOINT (f m) (f n)) ∧
      BIGUNION (IMAGE f (UNIV : num → bool)) IN sts ⇒
      (lambda ∘ f) sums
      (lambda (BIGUNION (IMAGE f (UNIV : num → bool)))),
```

where `IMAGE f s` *is the set of values taken by* `f` *when applied to the elements of* `s`, ∘ *is the function composition operation, and* `g sums c` *is equivalent to the mathematical notation* $\lim_{n\to\infty}(\sum_{i=0}^{n} \mathtt{g}(i)) \to \mathtt{c}$.

The correspondence between Definition 4 and Formalisation 5 is apparent. Definition 5 is more complicated, so Formalisation 6 requires careful examination. Note that, a sequence of sets s_i can be characterised by a function f from $\mathbb{N}$ to the elements of s_i, assuming that s_i is countable and its elements are disjoint; this function f maps the natural numbers to unique elements of s_i. This approach has been used to implicitly represent a countable sequence of sets in Formalisation 6.

At last we are ready to examine the HOL formalisation of a *measure space*; we begin by reviewing a textbook definition for measure spaces.

Definition 6 (measure space)**.** *Let* $\mathbb{S}$ *be a class of subsets of* S *such that* $(S, \mathbb{S})$ *defines a* σ*-algebra and let* λ *be a function from* $\mathbb{S}$ *to* $\mathbb{R}$*. Then* $(S, \mathbb{S}, \lambda)$ *defines a* **measure space** *iff* λ *is* positive *and* countably additive *with respect to* $(S, \mathbb{S})$.

Equivalently, one can state that $(S, \mathbb{S})$ *is* λ-measurable. $(S, \mathbb{S})$ *is considered to be* measurable *iff there exists some* λ *such that* $(S, \mathbb{S})$ *is* λ*-measurable i.e. iff* $(S, \mathbb{S})$ *defines a* σ*-algebra.*

S *is referred to as the* space *of the measure space,* $\mathbb{S}$ *as the* measurable sets *of the measure space, and* λ *as the* measure *of the measure space.*

Building on the formalisations developed above, it is straightforward to capture Definition 6 in a HOL formalisation.

Formalisation 7 (measure space)**.**

```
measure_space (sp, sts, lambda) = sigma_algebra (sp, sts) ∧
  positive (sp, sts, lambda) ∧ countably_additive (sp, sts, lambda).
```

3.3 Measurable functions

We complete our exploration of measure theory in HOL4 by studying definitions for *measurable functions* and *measure-preserving functions*. These properties are important for the definitions of Lebesgue integration (Section 4) and random variables (Section 5).

Definition 7 (measurable function)**.** *Let* $(S', \mathbb{S}')$ *and* $(S, \mathbb{S})$ *be measurable spaces. Let* $f[s]$ *denote the image of a function* f *on a set* s *and* $f^{-1}[s]$ *denote the inverse image of* f *on* s *i.e.* $\{y \mid \exists x.\ x \in s \wedge f(x) = y\}$ *and* $\{x \mid f(x) \in s\}$ *respectively. A function* f *from* S *into* S' *is* **measurable** *with respect to* $(S, \mathbb{S})$ *and* $(S', \mathbb{S}')$ *iff* $f^{-1}[s'] \in \mathbb{S}$ *for any* $s' \in \mathbb{S}'$*. Often,* f *is referred to as a measurable function from* $(S, \mathbb{S})$ *into* $(S', \mathbb{S}')$*.*

Formalisation 8 (measurable function)**.**

$$\begin{aligned}&\texttt{measurable}\ (\texttt{sp}, \texttt{sts})\ (\texttt{sp}', \texttt{sts}')\ \texttt{f}\ =\ \texttt{sigma\_algebra}\ (\texttt{sp}, \texttt{sts}) \wedge \\ &\quad \texttt{sigma\_algebra}\ (\texttt{sp}', \texttt{sts}')\ \wedge\ (\forall\ \texttt{x}.\ \texttt{x}\ \texttt{IN}\ \texttt{sp}\ \Rightarrow\ \texttt{f}\ \texttt{x}\ \texttt{IN}\ \texttt{sp}') \wedge \\ &\quad (\forall\ \texttt{s}'.\ \texttt{s}'\ \texttt{IN}\ \texttt{sts}'\ \Rightarrow\ ((\texttt{PREIMAGE}\ \texttt{f}\ \texttt{s}')\ \texttt{INTER}\ \texttt{sp})\ \texttt{IN}\ \texttt{sts}),\end{aligned}$$

where `INTER` *is the set-intersection operation and* `PREIMAGE f s` *denotes* $f^{-1}[s]$*.*

The HOL formalisation of Definition 7 is essentially a straightforward translation into higher-order logic. Recall from Definition 6 that a space is measurable if and only if it defines a σ-algebra. Thus, the first two conditions of Formalisation 8 correspond to the requirement in Definition 7 that both of the spaces involved are measurable. The next condition corresponds to the requirement that the function is from the one space into the other. The final condition is the property required of the inverse-image of the function by the definition.[5]

Definition 8 (measure-preserving)**.** *Let* $(S, \mathbb{S}, \lambda)$ *and* $(S', \mathbb{S}', \lambda')$ *be measure spaces and* f *a measurable function from* $(S, \mathbb{S})$ *into* $(S', \mathbb{S}')$*. Then* f *is* **measure preserving** *with respect to* $(S, \mathbb{S}, \lambda)$ *and* $(S', \mathbb{S}', \lambda')$ *iff* $\lambda(f^{-1}[s']) = \lambda'(s')$*, for any* $s' \in \mathbb{S}'$*.*

[5] In Formalisation 8, the inverse image of function `f` on set `s'` is intersected with space `sp`, but no such intersection occurs in Definition 7. This intersection is needed because HOL4 functions are total and `f` maps every value of the appropriate HOL type (including those outside of `sp`) to a value of the appropriate HOL type (which may or may not be in `sp'`). The third condition of the formalisation ensures that `f` maps values in `sp` into `sp'`, but there may still be values outside of `sp` which `f` maps into `sp'`. In the mathematical definition, a function from S to S' is only defined on S, so the intersection with S is unnecessary; in the formalisation, however, the inverse image of `f` must be intersected with the space `sp` in order to consider only values in `sp`.

Formalisation 9 (measure-preserving)**.** [6]

$$\begin{aligned}&\texttt{measure\_preserving}\ (\texttt{sp},\texttt{sts},\texttt{lambda})\ (\texttt{sp}',\texttt{sts}',\texttt{lambda}')\ \texttt{f}\ =\\&\quad\texttt{measurable}\ (\texttt{sp},\texttt{sts})\ (\texttt{sp}',\texttt{sts}')\ \texttt{f}\ \wedge\\&\quad(\forall\ \texttt{s}'.\ \texttt{s}'\ \texttt{IN}\ \texttt{sts}'\ \Rightarrow\\&\quad(\texttt{lambda}\ ((\texttt{PREIMAGE}\ \texttt{f}\ \texttt{s}')\ \texttt{INTER}\ \texttt{sp}) = \texttt{lambda}'\ (\texttt{s}'))).\end{aligned}$$

This concludes our study of measure theory formalised in higher-order logic. The majority of the effort in formalising the definitions above was devoted to proving that they satisfy appropriate properties. The proofs of those theorems are not included in this text, but the statement of each in HOL4 notation can be examined in Appendix B.

4 Lebesgue integration formalised in HOL4

We will now begin to focus on Lebesgue integration formalised in higher-order logic. Recall from Section 2.2, that this formalisation must be applicable to measurable functions from a space of an arbitrary type to a space that is a subset of the real numbers. Since the formalisation of measure theory in Section 3 defined measures to be real-valued (rather than using the extended-reals or hyper-reals), the integrals in this section are necessarily finite-valued; in the case of integrals diverging to positive or negative infinity, the integral is undefined.[7] This restriction to finite-valued measures was also adopted in the work of Hurd [7] and Richter [13] and still allows for a broad range of applications. In the formalisations below, the Lebesgue integral is defined using positive simple functions, as is done in standard textbooks [3, 10, 15] and in Richter's work.

4.1 Indicator functions and positive simple-functions

We begin our study of Lebesgue integration in HOL by reviewing definitions for *indicator functions* and *positive simple-functions*, which are needed to define the Lebesgue integral.

Definition 9 (indicator function)**.** *The* **indicator function** *of a set A, denoted by 1_A, is a real-valued function defined to be:*

$$1_A(a) = \begin{cases} 1 & a \in A \\ 0 & a \notin A \end{cases}$$

[6] The inverse image of `f` must be intersected with the space `sp` in Formalisation 9 for the same reason as in Formalisation 8.

[7] Strictly speaking, functions in HOL4 cannot be undefined. How undefinedness is handled in the system is explained in Appendix A.

The formalisation of Definition 9 in higher-order logic is straightforward as can be seen below.

Formalisation 10 (indicator function)**.**

$$\texttt{indicator\_fn A} \;=\; \lambda\texttt{a. if a IN A then 1 else 0}.$$

We now move on to review the definition of a *positive simple-function* and its integral with respect to a measure space; afterwards we will look at the formalisation of these definitions in higher-order logic.

Definition 10 (positive simple-function)**.** *Let $(S, \mathbb{S}, \lambda)$ be a measure space. A function f is a* **positive simple-function** *with respect to $(S, \mathbb{S}, \lambda)$ iff it can be defined as a linear combination of indicator functions of a finite number of disjoint elements of $\mathbb{S}$ with strictly positive coefficients i.e.*

$$f = \sum_{i=0}^{n} c_i(1_{a_i}),$$

for some finite *set of indices $i \in \{0, \ldots, n\}$, a set of coefficients c_i satisfying*

$$\forall i \in \{0, \ldots, n\}.\ 0 < c_i,$$

and a set of measurable sets of $(S, \mathbb{S}, \lambda)$, a_i, satisfying

$$(\forall i \in \{0, \ldots, n\}.\ a_i \in \mathbb{S}) \wedge (\forall i, j \in \{0, \ldots, n\}.\ i \neq j \Rightarrow a_i \cap a_j = \emptyset).$$

Equivalently, the coefficients c_i above can be required to be non-negative, rather than strictly positive, and the measurable sets a_i required to form a partition of S i.e. $\bigcup_{i=0}^{n} a_i = S$.

The integral of a positive simple-function f, defined by coefficients c_i and measurable sets a_i, on space S with respect to its measure λ is defined as

$$\int_S f\ d\lambda = \sum_{i=0}^{n} c_i(\lambda(a_i)).$$

The HOL definition of a positive simple-function below uses the second representation from Definition 10, allowing the coefficients to be non-negative and requiring the measurable sets to form a partition of the measure space. This form simplifies the proofs of some basic properties of positive simple-functions; Richter [13] used the same approach.

Formalisation 11 (positive simple-function)**.**

$$\begin{aligned}
&\texttt{pos_simple_fn (sp, sts, lambda) f s a c} \;=\; (\forall\ \texttt{x}.\ 0 \leq \texttt{f x}) \wedge \\
&\quad (\forall\ \texttt{x}.\ \texttt{x IN sp} \Rightarrow (\texttt{f x} = \texttt{SIGMA}\ (\lambda\ \texttt{i}.\ \texttt{c i} * (\texttt{indicator_fn (a i) x}))\ \texttt{s})) \wedge \\
&\quad (\forall\ \texttt{i}.\ \texttt{i IN s} \;\Rightarrow\; \texttt{a i IN sts}) \;\wedge\; (\forall\ \texttt{i}.\ 0 \leq \texttt{c i}) \wedge \\
&\quad \texttt{FINITE s} \;\wedge\; (\forall\ \texttt{i j}.\ \texttt{i IN s} \;\wedge\; \texttt{j IN s} \;\Rightarrow\; \texttt{DISJOINT (a i) (a j)}) \wedge \\
&\quad (\texttt{BIGUNION (IMAGE a s)} = \texttt{sp}),
\end{aligned}$$

where $(\texttt{sp}, \texttt{sts}, \texttt{lambda})$ *is a measure space,* `f` *is the function in question,* `s` *is a set of natural numbers representing the indices,* `a` *is a function from natural numbers to sets representing the measurable sets* a_i*, and* `c` *is a function from natural numbers to the real numbers representing the coefficients* c_i*.* `SIGMA f s` *is equivalent to* $\sum_{x \in s} f(x)$ *when* `s` *is finite and undefined otherwise.* `IMAGE f s` *denotes the image of function* `f` *on set* `s`*.*

The first condition in Formalisation 11 ensures that `f` takes only non-negative values.[8] This condition is implicit in the positivity of the measure λ and the non-negativity of the coefficients c_i in Definition 10. The second condition requires that the values taken by `f` on `sp` are defined by a linear combination of the coefficients and the indicator functions of the measurable sets.[9] The rest of the conditions are respectively: the `a i`'s are in `sts`, the coefficients are non-negative, the set of indices is *finite*, the `a i`'s are *disjoint* for different indices, and finally the `a i`'s form a partition of `sp` (more precisely that their union is `sp` which together with their disjointness makes them a partition).

Building on the formalised definition of a positive simple-function, the integral thereof has been defined in HOL as follows.

Formalisation 12 (integral of a positive simple-function)**.**

```
pos_simple_fn_integral (sp, sts, lambda) s a c =
  SIGMA (λ i. c i * lambda (a i)) s.
```

Note that there is no explicit reference to the function that is being integrated; instead it is represented through an index set, coefficients, and measurable sets. The representation of a positive simple function f by a set of coefficients and measurable sets is not necessarily distinct i.e. there may be many combinations of coefficients and measurable sets that are equivalent in their representation of f. Moreover, the integral of f is unique regardless of which representation is used to compute it. Thus, the integral of f can be referred to directly, without making explicit reference to sets and coefficients.

Following Richter's [13] lead, the *set of integrals* of a function f has been formalised, allowing the integral of f to be referred to directly. The set of integrals of f contains a single unique element when f is a positive simple-function and is empty otherwise. Below we will examine the HOL definitions

[8] Although only the values `f` takes on `sp` are of concern, `f` must be a total-function in HOL4 and might take negative values when applied outside of `sp`. For the sake of convenience `f` is required to be non-negative for all inputs; this is done without any loss of generality.

[9] Again, `f` is total and therefore well defined on values outside of `sp`, but only the values it takes on `sp` are of interest.

for the set of representations of a positive simple-function and the set of integrals of a particular positive simple-function.

Formalisation 13 (set of representations of a pos. simple-function)**.**

$$\begin{aligned}&\texttt{psfs}\ (\texttt{sp},\texttt{sts},\texttt{lambda})\ \texttt{f}\ = \\ &\quad \{(\texttt{s},\texttt{a},\texttt{c})\mid \texttt{pos\_simple\_fn}\ (\texttt{sp},\texttt{sts},\texttt{lambda})\ \texttt{f}\ \texttt{s}\ \texttt{a}\ \texttt{c}\}.\end{aligned}$$

Formalisation 14 (set of integrals of a positive simple-function)**.**

$$\begin{aligned}&\texttt{psfis}\ (\texttt{sp},\texttt{sts},\texttt{lambda})\ \texttt{f}\ = \\ &\quad \texttt{IMAGE}\ (\lambda\ (\texttt{s},\texttt{a},\texttt{c}).\ \texttt{pos\_simple\_fn\_integral}\ (\texttt{sp},\texttt{sts},\texttt{lambda})\ \texttt{s}\ \texttt{a}\ \texttt{c}) \\ &\qquad (\texttt{psfs}\ (\texttt{sp},\texttt{sts},\texttt{lambda})\ \texttt{f}).\end{aligned}$$

Before moving on to look at integration for positive measurable-functions, we will briefly review some useful and reassuring properties that have been proved about the formalisations developed above.

HOL Theorem 1 (uniqueness)**.** *The integral of a positive simple-function is* unique *regardless of the choice of representation using coefficients and measurable sets.*

$$\forall\ \texttt{m}\ \texttt{f}\ \texttt{a}\ \texttt{b}.\ \texttt{measure\_space}\ \texttt{m}\ \wedge\ \texttt{a}\ \texttt{IN}\ \texttt{psfis}\ \texttt{m}\ \texttt{f}\ \wedge\ \texttt{b}\ \texttt{IN}\ \texttt{psfis}\ \texttt{m}\ \texttt{f}\ \Rightarrow\ (\texttt{a}=\texttt{b}).$$

HOL Theorem 2 (additivity)**.** *The integral of the function defined by adding two positive simple-functions is the addition of their integrals.*

$$\begin{aligned}\forall\ \texttt{m}\ \texttt{f}\ \texttt{g}\ \texttt{a}\ \texttt{b}.\ \texttt{measure\_space}\ \texttt{m}\ \wedge\ \texttt{a}\ \texttt{IN}\ \texttt{psfis}\ \texttt{m}\ \texttt{f}\ \wedge\ \texttt{b}\ \texttt{IN}\ \texttt{psfis}\ \texttt{m}\ \texttt{g} \\ \Rightarrow\ \texttt{a}+\texttt{b}\ \texttt{IN}\ \texttt{psfis}\ \texttt{m}\ (\lambda\ \texttt{x}.\ \texttt{f}\ \texttt{x}+\texttt{g}\ \texttt{x}).\end{aligned}$$

Note that this property can be generalised to any finite linear combination of positive simple-functions; this generalised additivity property has also been proved.

HOL Theorem 3 (multiplicativity)**.** *The integral of the function defined by multiplying a positive simple-function f by a non-negative constant c is the integral of f multiplied by c.*

$$\begin{aligned}\forall\ \texttt{m}\ \texttt{f}\ \texttt{a}\ \texttt{c}.\ \texttt{measure\_space}\ \texttt{m}\ \wedge\ \texttt{a}\ \texttt{IN}\ \texttt{psfis}\ \texttt{m}\ \texttt{f}\ \wedge\ 0\leq\texttt{c} \\ \Rightarrow\ \texttt{c}*\texttt{a}\ \texttt{IN}\ \texttt{psfis}\ \texttt{m}\ (\lambda\ \texttt{x}.\ \texttt{c}*(\texttt{f}\ \texttt{x})).\end{aligned}$$

HOL Theorem 4 (monotonicity)**.** *If f is a positive simple-function which is pointwise less than or equal to another positive simple-function g, then the integral of f is less than or equal to the integral of g.*

$$\forall\ \texttt{m f g a b}.\ \texttt{measure\_space m}\ \wedge\ \texttt{a IN psfis m f}\ \wedge\ \texttt{b IN psfis m g}\ \wedge$$
$$(\forall\ \texttt{x}.\ \texttt{f x} \leq \texttt{g x})\ \Rightarrow\ \texttt{a} \leq \texttt{b}.$$

HOL Theorem 5 (commonality)**.** *If f and g are positive simple-functions on the same measure space m, then there is a set of measurable sets of m which, paired with a set of coefficients for f and a set of coefficients for g, can be used to characterise both f and g. The statement of this theorem in HOL4 is omitted here for the sake of brevity, but can be found in Appendix C as* `psfis_present`*.*

Theorem 1 (convergence of integrals of positive simple-functions)**.** *Let $(S, \mathcal{P}(S), \lambda)$ be a measure space whose measurable sets are the powerset of its space. Let S' be some subset of the reals and f be a non-negative real-valued function which is measurable from $(S, \mathcal{P}(S))$ to $(S', \mathcal{P}(S'))$. Let $\{f_i\}$ be a pointwise monotone-increasing sequence of positive simple-functions such that*

$$\forall\ x \in S.\ \lim_{i \to \infty} f_i(x) \to f(x).$$

If $\lim_{i\to\infty} \int_S f_i\ d\lambda \to r$ for some r, then for any positive simple-function g such that $\forall x.\ g(x) \leq f(x)$:

$$\int_S g\ d\lambda \leq r.$$

In terms of subsequent developments in this chapter, Theorem 1 is the most important property that has been proved about integration for positive simple-functions. Since the statement of that theorem is intricate and the proof lengthy, it appears here only in mathematical notation; the statement of Theorem 1 in HOL notation can be found in Appendix C under the name `psfis_mono_conv_mono`.

4.2 Integration of positive measurable-functions

Having looked at integration for positive simple-functions in the previous section, the next step towards Lebesgue integration for measurable functions is integration of *positive* measurable-functions. We will now review a textbook definition for the integral of a positive measurable-function, followed by its formalisation in higher-order logic.

Definition 11 (integral of a positive measurable-function)**.** *Let $(S, \mathbb{S}, \lambda)$ be a measure space, $(S', \mathbb{S}')$ a measurable space, and f a non-negative real-valued function that is measurable from $(S, \mathbb{S})$ to $(S', \mathbb{S}')$. Define the integral of f on S with respect to λ as*

$$\int_S f\ d\lambda = \sup \left\{ \int_S g\ d\lambda \;\middle|\; \begin{array}{l} (\forall\ x.\ g(x) \leq f(x)) \wedge \\ \text{g is a positive simple function w.r.t. } (\mathrm{S}, \mathbb{S}, \lambda) \end{array} \right\}.$$

Formalisation 15 (integral of a positive measurable-function)**.**

```
pos_fn_integral (sp, sts, lambda) f =
  sup {r | ∃ g. r IN psfis (sp, sts, lambda) g ∧ ∀ x. g x ≤ f x}.
```

Recall that the formalisations in this chapter are defined on the real numbers rather than the extended reals. If the integrals of the positive simple-functions less than `f` *diverge to infinity, they have no (strictly) real-valued supremum. Thus, the supremum above, and thereby the integral of* `f`*, might not exist.*

Building on Theorem 1 above, a convergence theorem guaranteeing the uniqueness of the integral of a positive measurable-function has been proved. This theorem is of great importance for subsequent formalisations, most notably for the proof simplifying the definition of Lebesgue integration for discrete spaces. The statement of this theorem is presented below in mathematical notation for the sake of brevity and readability; see Appendix C for the statement in HOL notation as `pos_fn_integral_eq_mono_conv_limit`.

Theorem 2 (convergence to the integral of a pos. measurable-fn.)**.** *Let* $(S, \mathcal{P}(S), \lambda)$ *be a measure space whose measurable sets are the powerset of its space. Let* S' *be a subset of the reals and* f *be a non-negative real-valued function that is measurable from* $(S, \mathcal{P}(S))$ *to* $(S', \mathcal{P}(S'))$*. Let* $\{f_i\}$ *be a pointwise monotone-increasing sequence of positive simple-functions such that*

$$\forall\ x \in S.\ \lim_{i \to \infty} f_i(x) \to f(x)\ \wedge\ \forall\ i\ x.\ f_i(x) \leq f(x).$$

If $\lim_{i \to \infty} \int_S f_i\ d\lambda \to r$*, for some* r*, then*

$$\int_S f\ d\lambda = r.$$

The proof of this property follows from Theorem 1.

If *a particular* sequence of positive simple-functions converge to a function f and their integrals converge to a value r, then Theorem 2 guarantees that the integral of f is r; this eliminates the need to consider the integrals of *every* positive simple-function which is pointwise less than or equal to f.

4.3 Lebesgue integration of measurable functions

Having studied integration of positive simple-functions and positive measurable-functions above, we can now take the final step and examine the Lebesgue integral of a measurable function. We will now review a textbook definition of this integral.

Definition 12 (Lebesgue integral of a measurable function)**.** *Let $(S, \mathbb{S}, \lambda)$ be a measure space, $(S', \mathbb{S}')$ a measurable space, and f a real-valued measurable function from $(S, \mathbb{S})$ to $(S', \mathbb{S}')$. Let $f^+(x) = \max(f(x), 0)$ and $f^-(x) = \max(-f(x), 0)$ i.e. f^+ and f^- are the positive and negative portions of f respectively. Note that $f(x) = f^+(x) - f^-(x)$ and f^+ and f^- are both non-negative functions. Define the integral of f on S with respect to λ to be*

$$\int_S f \; d\lambda = \int_S f^+ \; d\lambda - \int_S f^- \; d\lambda.$$

Note that the integral of f is well defined iff f^+ and f^- are both measurable from $(S, \mathbb{S})$ to $(S', \mathbb{S}')$ and their integrals do not both diverge to infinity. This measurability condition is non-trivial and does not necessarily hold for any choice of spaces and measurable function.

Seeing as the integral of a measurable function is simply the difference of the integrals of its positive and negative portions, it can be defined easily in terms of Formalisation 15.

Formalisation 16 (Lebesgue integral of a measurable function)**.**

```
fn_integral (sp, sts, lambda) f =
  pos_fn_integral (sp, sts, lambda) (λ x. if 0 < f x then f x else 0) −
  pos_fn_integral (sp, sts, lambda) (λ x. if f x < 0 then − f x else 0).
```

Since *finite* real values are used for the formalisations in this text, Formalisation 16 is undefined if the integral of either the positive or the negative portion of the function is infinite. This is a stricter requirement than in Definition 12, where only one of the integrals needs to be finite. In contrast, Lester's formalisation [9] allows infinite values, but is restricted to the non-negative extended reals. His formalisation stops with something akin to Formalisation 15 and does not require a formalisation of Definition 12, because negative values are not allowed. His approach simplifies some of the requirements for well-definedness, but does not allow for functions taking both positive and negative values.

Definition 12 generalises simpler definitions for finite and countable spaces; Formalisation 16 has been proved equivalent to those simpler forms in the theorem prover. Before looking at those proofs, let's examine HOL definitions of those simplified forms of the integral.

Formalisation 17 (integral of a measurable-fn. on a countable space)**.**

```
countable_measurable_fn_integral (sp, sts, lambda) f =
  let e = enumerate (IMAGE f sp) in
  suminf (λ i. e i * lambda (PREIMAGE f {e i} INTER sp)),
```

where `enumerate s` *is a bijection from the natural numbers to the elements of set* `s`*;* `enumerate s` *is well-defined only when* `s` *is a countably-infinite set.* `suminf f` *is the infinite summation of a real-valued function* `f` *on the natural numbers and is equivalent to the mathematical notation* $\sum_{i=0}^{\infty} f(i)$ *when this sum converges and is undefined otherwise.*

Because `enumerate` in Formalisation 17 is only well-defined on countably-*infinite* sets, it is necessary to formalise a separate definition for *finite* sets using a simpler finite summation.

Formalisation 18 (integral of a measurable-fn. on a finite space)**.**

```
finite_measurable_fn_integral (sp, sts, lambda) f =
  SIGMA (λ r. r * lambda (PREIMAGE f {r} INTER sp)) (IMAGE f sp).
```

Clearly, it is simpler and more intuitive to define integration as a summation (as in Formalisations 17 and 18) than as a difference of limits of sums of approximating functions (as in Formalisation 16). Let's take a look at an outline of the HOL proof that allows the general defintion of Lebesgue integration to be reduced to its simpler form for finite spaces; that proof is a useful contribution of this section and allows many general definitions in Section 5 to be simplified when considering *discrete* probability spaces.

HOL Theorem 6 (equivalence of Forms. 16 and 18 for finite spaces)**.**

```
∀ (sp, sts, lambda) f s. measure_space (sp, sts, lambda) ∧
  f IN measurable (sp, sts) (s, POW s) ∧ FINTE sp ∧ (s ≠ {})
  ⇒ (fn_integral (sp, sts, lambda) f =
       finite_measureable_fn_integral (sp, sts, lambda) f),
```

where `POW s` *denotes the powerset of* `s`*.*

Proof. Since `sp` *is finite, there exists a bijection* `e` *from* $\{0, \ldots, n\}$ *to the image of* `f` *on* `sp`*, for some natural number* `n`*.*

By the definition of a positive simple-function, the positive portion of `f`*, the function* (λ `x. if 0 < f x then f x else 0`)*, is a positive simple-function; it can be represented using the index set* $\{0, \ldots, n\}$*, the measurable sets*

```
a i = PREIMAGE f {e i} INTER sp,
```

and the coefficients

```
c i = if 0 ≤ e i then e i else 0.
```

Thus, by the definition of supremum and the reflexivity of $\leq$*,*

$$\begin{aligned} & \texttt{pos\_fn\_integral}\ (\texttt{sp},\texttt{sts},\texttt{lambda})\ (\lambda\ \texttt{x}.\ \texttt{if}\ 0 < \texttt{f}\ \texttt{x}\ \texttt{then}\ \texttt{f}\ \texttt{x}\ \texttt{else}\ 0) \\ = \ & \texttt{pos\_simple\_fn\_integral}\ (\texttt{sp},\texttt{sts},\texttt{lambda})\ \{0,\ldots,\texttt{n}\}\ \texttt{a}\ \texttt{c}. \end{aligned}$$

The reduction of the integral of the negative portion of `f` *to*

$$\begin{aligned} & \texttt{pos\_simple\_fn\_integral}\ \ (\texttt{sp},\texttt{sts},\texttt{lambda})\ \{0,\ldots,\texttt{n}\} \\ & \quad (\lambda\ \texttt{i}.\texttt{PREIMAGE}\ \texttt{f}\ \{\texttt{e}\ \texttt{i}\}\ \texttt{INTER}\ \texttt{sp})\ (\lambda\ \texttt{i}.\ \texttt{if}\ \texttt{e}\ \texttt{i} \leq 0\ \texttt{then}\ \texttt{e}\ \texttt{i}\ \texttt{else}\ 0) \end{aligned}$$

is symmetric to the case for the positive portion of `f`*.*

The goal follows immediately from the basic definitions and the additive property of `SIGMA`*.* □

Now let's look at the case of countably-infinite spaces.

HOL Theorem 7 (equivalence of forms. 16 and 17 for countable sps.)**.**

$$\begin{aligned} & \forall\ (\texttt{sp},\texttt{sts},\texttt{lambda})\ \texttt{f}\ \texttt{s}\ \texttt{s}'\ \texttt{p}\ \texttt{p}'.\ \texttt{measure\_space}\ (\texttt{sp},\texttt{sts},\texttt{lambda}) \wedge \\ & \quad \texttt{f}\ \texttt{IN}\ \texttt{measurable}\ (\texttt{sp},\texttt{sts})\ (\texttt{s},\texttt{POW}\ \texttt{s}) \wedge \\ & \quad (\lambda\ \texttt{x}. -\texttt{f}\ \texttt{x})\ \texttt{IN}\ \texttt{measurable}\ (\texttt{sp},\texttt{sts})\ (\texttt{s}',\texttt{POW}\ \texttt{s}') \wedge \\ & \quad \texttt{f}\ \texttt{IN}\ \texttt{measurable}\ (\texttt{sp},\texttt{sts})\ (\texttt{s}',\texttt{POW}\ \texttt{s}')\ \wedge\ \texttt{countable}\ \texttt{sp} \wedge \\ & \quad \neg\ \texttt{FINITE}\ (\texttt{s}\ \texttt{INTER}\ \texttt{IMAGE}\ \texttt{f}\ \texttt{sp})\ \wedge\ \neg\ \texttt{FINITE}\ (\texttt{s}'\ \texttt{INTER}\ \texttt{IMAGE}\ \texttt{f}\ \texttt{sp}) \wedge \\ & \quad (\texttt{s} \neq \{\})\ \wedge\ (\texttt{s}' \neq \{\})\ \wedge\ 0\ \texttt{IN}\ \texttt{s}\ \wedge\ 0\ \texttt{IN}\ \texttt{s}' \wedge \\ & \quad (\lambda\ \texttt{r}.\ (\texttt{if}\ 0 < \texttt{r}\ \texttt{then}\ \texttt{r}\ \texttt{else}\ 0) * \texttt{lambda}\ (\texttt{PREIMAGE}\ \texttt{f}\ \{\texttt{r}\}\ \texttt{INTER}\ \texttt{sp})) \circ \\ & \qquad (\texttt{enumerate}\ (\texttt{IMAGE}\ \texttt{f}\ \texttt{sp}))\ \texttt{sums}\ \texttt{p} \wedge \\ & \quad (\lambda\ \texttt{r}.\ (\texttt{if}\ \texttt{r} < 0\ \texttt{then} -\texttt{r}\ \texttt{else}\ 0) * \texttt{lambda}\ (\texttt{PREIMAGE}\ \texttt{f}\ \{\texttt{r}\}\ \texttt{INTER}\ \texttt{sp})) \circ \\ & \qquad (\texttt{enumerate}\ (\texttt{IMAGE}\ \texttt{f}\ \texttt{sp}))\ \texttt{sums}\ \texttt{p}'\ \Rightarrow \\ & \quad (\texttt{fn\_integral}\ (\texttt{sp},\texttt{sts},\texttt{lambda})\ \texttt{f}\ = \\ & \quad\ \texttt{countable\_measurable\_fn\_integral}\ (\texttt{sp},\texttt{sts},\texttt{lambda})\ \texttt{f}), \end{aligned}$$

where `sums f r` *denotes that the infinite summation of* `f` *(as defined by* `suminf`*) converges to* `r` *and* $\circ$ *denotes the function composition operator. For generality, the positive and negative portions of* `f` *can be measurable on different subsets of the reals; if needed, a subset of the reals on which they are both measurable can always be found (in the worst case* $\mathbb{R}$ *itself). Also note that the subset of the reals on which the positive/negative portion of* `f` *is measurable must include* 0*. This is necessary in order for the splitting of the function to maintain measurability i.e. the preimage of* `f` *on* $\{0\}$ *must be in* `sts`*.*

Proof. By the countability of `sp` *and the infiniteness of* `IMAGE f sp INTER s`*,* `enumerate (IMAGE f s INTER sp)` *is a bijection from the natural numbers to*

IMAGE f s INTER sp. *Let* e *abbreviate this enumeration. Define:*

$$\texttt{fi i} = (\lambda\ \texttt{x}.\ \texttt{if x IN PREIMAGE f (IMAGE e } \{0, \ldots, \texttt{i}\}\texttt{) INTER sp then}$$
$$\texttt{(if 0} < \texttt{f x then f x else 0) else 0}).$$

The sequence of functions {fi} *is monotone-increasing and converges pointwise to* f*. Define:*

$$\texttt{a n} = (\lambda\ \texttt{i}.\ \texttt{if i} = \texttt{n} + 1\ \texttt{then sp DIFF PREIMAGE f (IMAGE e } \{0, \ldots, \texttt{i}\})$$
$$\texttt{else PREIMAGE f } \{\texttt{e i}\}\ \texttt{INTER sp}),$$

$$\texttt{c n} = (\lambda\ \texttt{i}.\ \texttt{if i} = \texttt{n} + 1\ \texttt{then 0 else (if 0} < \texttt{e i then e i else 0})).$$

For any n*,* fi n *is a positive simple-function represented by the index set* $\{0, \ldots, \texttt{n} + 1\}$*, the measurable sets* a n*, and the coefficients* c n*.*

By HOL theorem 2, if the integrals of the fi*'s converge to a value* x*, then the integral of the positive portion of* f *is* x*.*

By basic definitions, the countable_measurable_fn_integral *of the positive portion of* f *is the value to which the integrals of the* fi*'s converge to i.e.* p*.*

The case for the negative portion of f *is symmetric.The goal then follows directly from the additive property for infinite summations.* □

One final useful property of Lebesgue integration is that the integral of the indicator function of a measurable set is the measure of that set. The statement of that theorem in HOL4 and in mathematical notation can be found below. The proof is relatively straightforward and has been omitted for the sake of brevity.

HOL Theorem 8 (integral of the indicator-fn. of a measurable set)**.**

$$\forall\ (\texttt{sp}, \texttt{sts}, \texttt{lambda})\ \texttt{s}.\ \texttt{measure\_space}\ (\texttt{sp}, \texttt{sts}, \texttt{lambda})\ \wedge\ \texttt{s IN sts}\ \Rightarrow$$
$$(\texttt{fn\_integral}\ (\texttt{indicator\_fn s}) = \texttt{lambda s}).$$

Equivalently in mathematical notation: Let $(S, \mathbb{S}, \lambda)$ *be a measure space.*

$$\forall\ a \in \mathbb{S}.\ \int_S 1_a\ d\lambda = \lambda(a).$$

4.4 Radon-Nikodým derivatives

The Radon-Nikodým derivative of one measure with respect to another measure is needed to define some of the information-theoretic concepts that will be presented in Chapter **??**. Let's review a definition of the Radon-Nikodým derivative and then examine the formalisation of that definition in higher-order logic.

Definition 13 (Radon-Nikodým derivative)**.** *Let* $(S, \mathbb{S}, \mu)$ *and* $(S, \mathbb{S}, \nu)$ *be measure spaces. Notice that* μ *and* ν *are measures on the same space. Define the* **Radon-Nikodým derivative** *of* ν *with respect to* $(S, \mathbb{S}, \mu)$ *to be the function* f *such that*

$$\forall\ a \in \mathbb{S}.\ \nu(a) = \int_a f(a) d\mu,$$

where the integral above is a Lebesgue integral. The function f*, representing the Radon-Nikodým derivative of* ν *with respect to* μ*, is often denoted by* $\frac{d\nu}{d\mu}$*.*

The Radon-Nikodým theorem guarantees the existence of such a derivative and its uniqueness, up to μ*-null sets, for any absolutely continuous* ν*. More details about the Radon-Nikodým theorem can be found in standard textbooks addressing measure and integration [3, 10, 15].*

The formalisation of the Radon-Nikodým derivative in higher-order logic is surprisingly straightforward as can be seen below.

Formalisation 19 (Radon-Nikodým derivative)**.**

```
RN_deriv (sp,sts,mu) nu =
  @f. (∃ s. f IN measurable (sp,sts) (s,POW s)) ∧
      (∀ a. a IN sts ⇒
        (fn_integral (sp,sts,mu) (λ x. f x * indicator_fn a x) = nu a)),
```

where `@x. P x` *denotes the Hilbert's choice operator.* `@x. P x` *can be read informally as* "any `x` such that `P x`".

As noted above, the Radon-Nikodým theorem guarantees the existence and uniqueness (up to null sets) of the derivative. The proof of that theorem is non-trivial and best left until a larger body of integrability theorems have been proved. However, the reduction of Formalisation 19 needed for the developments in Chapter **??** does not rely on a general proof of the Radon-Nikodým theorem. In the case of finite, standard spaces, the R.-N. derivative of two measures of a non-null set reduces to a division of of the two measures. This reduction is stated more rigourously below.

HOL Theorem 9 (Radon-Nikodým derivative of finite, standard sps.)**.**

```
∀ sp mu nu. FINITE sp ∧ measure_space (sp,POW sp,mu) ∧
  measure_space (sp,POW sp,nu) ∧
  (∀ x. (mu {x} = 0) ⇒ (nu {x} = 0)) ⇒
  (∀ x. x IN sp ∧ (mu {x} ≠ 0) ⇒
    (RN_deriv (sp,POW sp,mu) nu x = nu {x}/mu {x})).
```

This reduction of the Radon-Nikodým derivative of ν *with respect to* μ *to* $\frac{\nu(x)}{\mu(x)}$ *is a nice justification of typical denotation as* $\frac{d\nu}{d\mu}(x)$*.*

4.5 Product measures

The concept of a *product measure* on a *product space* is needed for some definitions from information theory. This topic really belongs under the domain of measure theory (Section 3), but its presentation has been delayed until now because Lebesgue integration is required. A thorough formalisation of theorems relating to product measures is not in the scope of the present work and is left as a useful area for future work. We will now review a general definition of product measure spaces, followed by the HOL formalisation of that definition and theorems needed in subsequent chapters.

Definition 14 (product measure-space)**.** *Let* $(S_1, \mathbb{S}_1, \mu_1)$ *and* $(S_2, \mathbb{S}_2, \mu_2)$ *be measure spaces. Let* $S = S_1 \times S_2$ *be the space defined by the cross product of* S_1 *and* S_2 *and* $\mathbb{S}$ *be the class of subsets in the smallest* σ*-algebra generated by the cross products of sets in* $\mathbb{S}_1$ *and* $\mathbb{S}_2$ *respectively. If* μ *is a measure on* $(S, \mathbb{S})$ *such that*

$$\forall a_1 \in \mathbb{S}_1 \; a_2 \in \mathbb{S}_2.\; \mu(a_1 \times a_2) = \mu_1(a_1)\mu_2(a_2),$$

then $(S, \mathbb{S}, \mu)$ *is a* **product measure-space** *and* μ *is its* **product measure**. *A product measure* μ *for two measure spaces* $(S_1, \mathbb{S}_1, \mu_1)$ *and* $(S_2, \mathbb{S}_2, \mu_2)$ *can be constructed as*

$$\mu(a) = \int_{S_1} (\lambda x_1.\; \mu_2((\lambda x_2.\; (x_1, x_2))^{-1}[a])) = \int_{S_2} (\lambda x_2.\; \mu_1((\lambda x_1.\; (x_1, x_2))^{-1}[a])),$$

where $(\lambda\; x_2.\; (x_1, x_2))^{-1}[a]$ *denotes the inverse-image of* $(\lambda\; x_2.\; (x_1, x_2))$ *on* a*. The existence and uniqueness of the product measure are guaranteed by the Fubini-Lebesgue theorem. A detailed presentation of the Fubini-Lebesgue theorem can be found in standard textbooks covering measure theory and integration [3, 10, 15].*

The construction of the product measure in Definition 14 can defined in higher-order logic using the formalisation of Lebesgue integration developed above.

Formalisation 20 (product measure)**.**

```
prod_measure (sp1, sts1, mu1) (sp2, sts2, mu2) =
  (λ a. fn_integral (sp1, sts1, mu1)
        (λ x1. mu2 (PREIMAGE (λ x2. (x1, x2)) a))).
```

Building on this definition, product measure-spaces can be formalised in HOL as follows.

Formalisation 21 (product measure-space)**.**

```
prod_measure_space (sp1, sts1, mu1) (sp2, sts2, mu2) =
  (sp1 CROSS sp2,
   subsets(sigma((sp1 CROSS sp2) (prod_sets sts1 sts2))),
   prod_measure (sp1, sts1, mu1) (sp2, sts2, mu2)).
```

A general proof of the Fubini-Lebesgue theorem is outside the scope of the present work and will not be undertaken here. However, the equivalence of the product measure to the product of the respective measures has been proved for the case of finite, standard spaces. The statement of that theorem and that the product measure-space forms a valid measure space in the finite, standard case can be found in Appendix C as `finite_prod_measure_reduce` and `measure_space_finite_prod_measure` respectively.

5 Probability theory formalised in HOL4

5.1 A general formalisation of probability theory

Many of the foundational definitions in probability theory are simply specific instances of corresponding definitions in measure theory. By building on the formalisation of measure theory presented in Section 3, it is straightforward to generalise Hurd's formalisation of probability theory in HOL. This section examines such a formalisation. The specialisation of a measure space to define a probability space serves as a natural starting point. Thus, we will begin by reviewing a textbook definition for a *probability space* and then studying the HOL formalisation of that definition.

Definition 15 (probability space)**.** *Let* $(S, \mathbb{S}, \mu)$ *be a measure space.* $(S, \mathbb{S}, \mu)$ *is a* **probability space** *iff* $\mu(S) = 1$*. For a probability space* $(S, \mathbb{S}, \mu)$*, we refer to* S*,* $\mathbb{S}$*, and* μ *respectively as the* space*,* events*, and* probability *function of* $(S, \mathbb{S}, \mu)$*.*

Definition 15 can be formalised as extension of the HOL definition for measure spaces (Formalisation 7).

Formalisation 22 (probability space)**.**

```
probability_space (sp, sts, mu) =
  measurable_space (sp, sts, mu) ∧ (mu (sp) = 1).
```

The restriction on Hurd's formalisation of probability spaces [7] is precisely his restriction on measure spaces carried forward; that restriction was explained in detail in Sections 2 and 3 and will not be readdressed here.

One of the most important properties that can hold between two events in a probability space is *independence*; much of Hurd's formalisation work focused on independence results for events and functions on probability spaces. We will now look at a HOL definition for the independence of two events; first we review a textbook definition of this property.

Definition 16 (independent events)**.** *Let* $(S, \mathbb{S}, \mu)$ *be a probability space. Then* $s \in \mathbb{S}$ *and* $s' \in \mathbb{S}$ *are* **independent events** *of* $(S, \mathbb{S}, \mu)$ *iff*

$$\mu(s \cap s') = (mu(s))(mu(s')).$$

Note that $s \cap s'$ *is guaranteed to be an event of* $(S, \mathbb{S}, \mu)$*, when* s *and* s' *are both events of* $(S, \mathbb{S}, \mu)$*, because* $(S, \mathbb{S})$ *defines a* σ*-algebra and therefore* $\mathbb{S}$ *is closed under intersection.*

Formalisation 23 (independent events)**.**

$$\texttt{indep}\ (\texttt{sp}, \texttt{sts}, \texttt{mu})\ \texttt{s}\ \texttt{s}' =$$
$$\texttt{s IN sts}\ \wedge\ \texttt{s}'\ \texttt{IN sts}\ \wedge\ (\texttt{mu}\ (\texttt{s INTER s}') = \texttt{mu s} * \texttt{mu s}'),$$

assuming `probability_space` $(\texttt{sp}, \texttt{sts}, \texttt{mu})$ *holds.*

This concludes our look at HOL formalisations generalising Hurd's for probability theory. As was this case for measure theory, the majority of the effort required was not devoted to formalising the definitions, but to proving that they satisfy appropriate properties. That effort does not appear here; however, those theorems can be found in HOL notation in Appendix D.

5.2 Extensions to the formalisation of probability theory

Hurd's formalisations [7] did not include a number of basic definitions from probability theory that are needed for information theory; those definitions were not required for his applications. On the other hand, Hurd could not have formalised many of those definitions, without first formalising a number of other definitions. For example, he could not have developed a general measure-theoretic definition of expected value without first formalising Lebesgue integration. Having examined a generalisations of Hurd's constructions for probability theory, we will now move on to study extensions to those formalisations; those extensions include definitions from probability theory that are necessary for the development of information theory in Chapter **??**.

In the formalisations below, general measure-theoretic definitions have been used in order to maintain flexibility for future applications and a mathematically sound basis. Wherever practical, the general definitions have been proved

equivalent to simpler forms for discrete probability spaces i.e. those with a countable space. Although the general measure-theoretic definitions are sufficient by themselves, these proofs make the formalisation much easier to use by replacing more complex constructs (e.g. Lebesgue integration) with simpler ones (e.g. summation) wherever possible.

Discrete probability spaces are sufficient for the intended applications of this work, so focus has been placed on developments for those spaces. Users of the formalisation benefit from the simplifications that have been proved, without a loss of generality and applicability to continuous spaces and other more unusual spaces. Furthermore, probability spaces with a finite space are sufficient for the eventual applications of this work (Chapters **??** and **??**). As discussed in Section 4, technical details of the HOL4 formalisations require separate reductions for finite and countably-infinite probability spaces. This can be seen as a useful feature, allowing finite (rather than infinite) summations to be used in definitions for finite spaces. Considering the overall aims of this text, the presentation below will focus on developments for finite probability spaces in greater detail than for countably-infinite probability spaces; however, both formalisations will be explained thoroughly.

We begin our investigation of the extensions to the constructions of Section 5.1 by reviewing a textbook definition for a *random variable* followed by the formalisation of that definition in higher-order logic.

Definition 17 (random variable)**.** *Let $\mathcal{X}$ be a* measurable function *from $(S, \mathbb{S})$ to $(S', \mathbb{S}')$. If $(S, \mathbb{S})$ is equipped with a probability measure μ, i.e. $(S, \mathbb{S}, \mu)$ is a probability space, then $\mathcal{X}$ is called a* **random variable** *on $(S, \mathbb{S}, \mu)$.*

In the case that S' is countable, $\mathcal{X}$ is referred to as a discrete *random-variable. In the case that $(S', \mathbb{S}')$ is the Borel algebra and the probability of $\mathcal{X}$ taking a value r is 0 for any real-valued r, $\mathcal{X}$ is an* absolutely continuous *random-variable.*

Assuming the appropriate measurability requirements hold, the joint random-variable *of two random variables $\mathcal{X}$ and $\mathcal{Y}$ is the function mapping an input to the pair of values taken by $\mathcal{X}$ and $\mathcal{Y}$ for that input i.e.*

$$(\mathcal{X}, \mathcal{Y})(x) = (\mathcal{X}(x), \mathcal{Y}(x)).$$

Typically, $(\mathcal{X}, \mathcal{Y})$ is used to denote the joint random-variable of $\mathcal{X}$ and $\mathcal{Y}$, emphasising that it takes a pair of values.

Definition 17 is formalised simply by referring to a measurable function as a random variable in probability-theoretic contexts.

Formalisation 24 (random-variable)**.**

$$\texttt{random\_variable X (sp, sts, mu) (sp', sts')} \;=$$
$$\texttt{prob\_space (sp, sts, mu)} \;\wedge\; \texttt{X IN measurable (sp, sts) (sp', sts')}.$$

A specific formalisation for joint random-variables is unnecessary and only adds additional complication. The joint random-variable of `X` and `Y` can be define simply as (λ `x`. (`X x`, `Y x`)).

All random variables implicitly define a probability measure over the σ-algebra onto which they are measurable. This measure is known as the *probability mass function (PMF)* of the random variable and lies at the core of many of the definitions that follow. We will now review a general definition of the PMF of a random variable and and its relation to the typical presentations for discrete and continuous random variables as well as to the Lebesgue integral. We will then go on to study the HOL construction for the definition of the PMF of a random variable.

Definition 18 (probability mass function)**.** *Let $\mathcal{X}$ be a random variable from a probability space $(S, \mathbb{S}, \mu)$ to a σ-algebra $(S', \mathbb{S}')$. The* **probability mass function (PMF)** *of $\mathcal{X}$ is defined to be*

$$\mathrm{P}(\mathcal{X} \in A) = \mu(\mathcal{X}^{-1}[A]),$$

for any $A \in \mathbb{S}'$. Note that $(S', \mathbb{S}', (\lambda\ A.\ \mathrm{P}(\mathcal{X} \in A)))$ is a probability space. Thus, $\mathrm{P}(\mathcal{X} \in A)$ intuitively defines the probability of $\mathcal{X}$ taking a value in A.

In the case that $\mathcal{X}$ is a discrete *random-variable taking values $\{x_0, x_1, \ldots\}$, the term* **PMF** *is often overloaded to refer specifically to $\mathrm{P}(\mathcal{X} = x_i) = \mathrm{P}(\mathcal{X} \in \{x_i\})$. If $\mathcal{X}$ is a* continuous *random-variable, the term* **cumulative distribution function (CDF)** *is used to refer to $\mathrm{P}(\mathcal{X} \leq x) = \mathrm{P}(\mathcal{X} \in \{y \mid y \leq x\})$.*

Recall that for a probability space $(S, \mathbb{S}, \mu)$, $\forall\ A \in \mathbb{S}.\ \mu(A) = \int_S 1_A\ d\mu$. (HOL Theorem 8). Thus,

$$\mathrm{P}(\mathcal{X} \in A) = \int_S 1_{\mathcal{X}^{-1}[A]}\ d\mu = \int_{S'} 1_A\ d(\lambda\ A.\ \mathrm{P}(\mathcal{X} \in A)).$$

We will now look at the definition of the PMF of a random variable in HOL4.

Formalisation 25 (probability mass function)**.**

$$\texttt{pmf (sp, sts, mu) X} \;=\; (\lambda\ \texttt{A}.\ \texttt{mu (PREIMAGE X A INTER sp)}).$$

The use of this formalisation for the specialised cases of the cdf *and* discrete *pmf are straightforward, simply involving the restriction of the set* `A` *as outlined in Definition 18.*

The statement of HOL theorems showing that the PMF of a random variable defines a probability measure on its range space and the relationships of the PMF to Lebesgue integral outlined at the end of Definition 18 can be found in Appendix D as `pmf_prob_space`, `pmf_lebesgue_thm1`, and `pmf_lebesgue_thm2` respectively.

Two concepts closely related to the PMF of a random variable are the joint PMF of two random variables and the conditional PMF of two random variables. We will now review the definitions of those concepts; we will then study the HOL development of those definitions.

Definition 19 (joint probability mass function)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be random variables from a probability space $(S, \mathbb{S}, \mu)$ to σ-algebras $(S', \mathbb{S}')$ and $(S'', \mathbb{S}'')$ respectively. The* **joint probability mass function (joint PMF)** *of $\mathcal{X}$ and $\mathcal{Y}$ is defined to be*

$$\mathrm{P}(\mathcal{X} \in A; \mathcal{Y} \in B) = \mu(\mathcal{X}^{-1}[A] \cap \mathcal{Y}^{-1}[B]),$$

for any $A \in \mathbb{S}'$ and $B \in \mathbb{S}''$.

Formalisation 26 (joint probability mass function)**.**

```
joint_pmf (sp,sts,mu) X Y =
  (λ (A,B). mu ((PREIMAGE X A) INTER (PREIMAGE Y B) INTER sp)).
```

Definition 20 (conditional probability mass function)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be random variables from a probability space $(S, \mathbb{S}, \mu)$ to σ-algebras $(S', \mathbb{S}')$ and $(S'', \mathbb{S}'')$ respectively. The* **conditional probability mass function (conditional PMF)** *of $\mathcal{X}$ and $\mathcal{Y}$ is defined to be*

$$\mathrm{P}(\mathcal{X} \in A | \mathcal{Y} \in B) = \frac{\mathrm{P}(\mathcal{X} \in A; \mathcal{Y} \in B)}{\mathrm{P}(\mathcal{Y} \in B)},$$

for any $A \in \mathbb{S}'$ and $B \in \mathbb{S}''$.

Formalisation 27 (conditional probability mass function)**.**

```
conditional_pmf (sp,sts,mu) X Y =
  (λ (A,B). joint_pmf (sp,sts,mu) X Y (A,B) / pmf (sp,sts,mu) Y B).
```

Having examined the definition of a random variable and its probability mass function, we now move on to the notion of the *expected value* or *expectation* of a random variable. It is here that we will see the efforts of Section 4 come to fruition. Let us begin by reviewing the definition of the expected value of a random variable.

Definition 21 (expected value)**.** *The* **expected value** *or* **expectation** *of a real-valued random variable $\mathcal{X}$ on probability space $(S, \mathbb{S}, \mu)$ is the Lebesgue intregral of $\mathcal{X}$ on S with respect to μ:*

$$\mathbb{E}(\mathcal{X}) = \int_S \mathcal{X}\ d\mu.$$

If $\mathcal{X}$ is a discrete random-variable taking values $\{x_0, x_1, \ldots\}$, then the expected value of $\mathcal{X}$ is typically expressed in the simpler form

$$\mathbb{E}(\mathcal{X}) = \sum_{x_i} (x_i) \mathrm{P}(\mathcal{X} = x_i).$$

Intuitively, $\mathbb{E}(\mathcal{X})$ is the mean *of the values taken by $\mathcal{X}$ on S, weighted by their respective probabilities.*

The HOL formalisation of Definition 21 is simply a matter of referring to the Lebesgue integral as expectation in probability-theoretic contexts.

Formalisation 28 (expected value)**.**

$$\texttt{expectation} \ = \ \texttt{fn\_integral}.$$

The practice of proving simplifying equivalences for HOL constructions has been carried forward to the definition of expectation; this includes a proof that the formalisation captures the intuitive notion of expectation for discrete random variables.

HOL Theorem 10 (reduction of expecation of a r.v. on a finite space)**.**

$$\begin{array}{l} \forall\ (\texttt{sp}, \texttt{sts}, \texttt{mu})\ \texttt{s}'\ \texttt{X}.\ \texttt{random\_variable}\ \texttt{X}\ (\texttt{sp}, \texttt{sts}, \texttt{mu})\ (\texttt{s}', \texttt{POW}\ \texttt{s}') \wedge \\ \quad \texttt{FINITE}\ \texttt{sp} \ \wedge \ (\texttt{s} \neq \{\}) \ \Rightarrow \\ \quad (\texttt{expectation}\ (\texttt{sp}, \texttt{sts}, \texttt{mu})\ \texttt{X} = \\ \quad\ \texttt{SIGMA}\ (\lambda\ \texttt{r}.\ \texttt{r} \ * \ \texttt{pmf}\ (\texttt{sp}, \texttt{sts}, \texttt{mu})\ \texttt{X}\ \{\texttt{r}\})\ (\texttt{IMAGE}\ \texttt{X}\ \texttt{sp})). \end{array}$$

This theorem captures the simplification of expectation to the intuitive notion of expectation for a discrete *random variable presented at the end of Definition 21. The proof follows from HOL Theorem 6 and basic definitions.*

A similar reduction of the formalisation of expectation holds for a random variable on a countably-infinite space; the proof is straightforward and follows from HOL Theorem 7 and basic definitions. For the sake of brevity this theorem is not stated here, but can be found in Appendix D as `countable_expectation`.

The final extension to the formalisation of probability theory that we will examine is the *conditional expectation* of a random variable. This construction will not be used for information-theoretic definitions in Chapter **??**, so it is

not developed in as much detail; however there are many useful applications for conditional expectation, which motivate a brief presentation of its formalisation. The definition of conditional expectation also provides an excellent example of how a subtle error might be introduced into probability-theoretic definitions by following intuition without resorting to a measure-theoretic basis. We will begin by reviewing a definition of conditional expectation and then proceed to study its development in HOL.

Definition 22 (conditional expectation)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be real-valued random variables from a probability space $(S, \mathbb{S}, \mu)$ to σ-algebras $(S', \mathbb{S}')$ and $(S'', \mathbb{S}'')$ respectively. The* **conditional expectation** *of $\mathcal{X}$ given $\mathcal{Y}$ is a random variable on $(S'', \mathbb{S}'', \mathrm{P}_{\mathcal{Y}})$ and is defined to be any function $\mathbb{E}(\mathcal{X}|\mathcal{Y} \in A)$ satisfying the following properties:*

- *$\mathbb{E}(\mathcal{X}|\mathcal{Y} \in A)$ is a real-valued random variable on $(S'', \mathbb{S}'', \mathrm{P}_{\mathcal{Y}})$,*
- *$\int_A \mathbb{E}(\mathcal{X}|\mathcal{Y} \in A)\ d\mathrm{P}_{\mathcal{Y}} = \mathbb{E}(\mathcal{X} 1_{\mathcal{Y}^{-1}[A]}) = \int_{\mathcal{Y}^{-1}[A]} \mathcal{X}\ d\mu$,*

where $\mathrm{P}_{\mathcal{Y}}$ is the PMF of $\mathcal{Y}$. Thus there can be different versions *of $\mathbb{E}(\mathcal{X}|\mathcal{Y} \in A)$. Notice that this form is similar to the way the Radon-Nikodým derivative is defined. In fact, the Radon-Nikodým theorem guarantees that the conditional expectation exists and that the versions of it are equal up to μ-null sets (which include singletons in the case of continuous random variables); additional details can be found in standard texts on probability and integration [3, 15].*

Intuitively, $\mathbb{E}(\mathcal{X}|\mathcal{Y} \in A)$ is the expected value of $\mathcal{X}$ when $\mathcal{Y}$ takes a value in A. If $\mathcal{X}$ and $\mathcal{Y}$ are discrete random-variables taking values $\{x_0, x_1, \ldots\}$ and $\{y_0, y_1, \ldots\}$ respectively, then the conditional expectation of $\mathcal{X}$ and $\mathcal{Y}$ can be expressed in the simpler form

$$\mathbb{E}(\mathcal{X}|\mathcal{Y} = y_j) = \sum_{x_i} (x_i) \mathrm{P}(\mathcal{X} = x_i | \mathcal{Y} = y_j).$$

One might be tempted to define conditional expectation in the continuous case as

$$\mathbb{E}(\mathcal{X}|\mathcal{Y} = y) = \int_{x \in S'} (x) \mathrm{P}(\mathcal{X} = x | \mathcal{Y} = y),$$

but such a definition would be incorrect. Recall that if $\mathcal{Y}$ is continuous then $\mathrm{P}(\mathcal{Y} = y) = 0$ for all y. Therefore $\mathrm{P}(\mathcal{X} = x|\mathcal{Y} = y) = 0$ for any x and y, and the incorrect definition above simply defines the constant function that is everywhere zero. This illustrates why conditional expectation cannot be defined outright and must be characterised (up to null sets) as above.

More details on how conditional expectation of continuous random variables can be characterised using the cumulative density function can be found in standard texts [3, 15]; that approach is not substantially easier than the general definition, so it is not addressed here.

Definition 22 has been formalised in HOL4 as follows:
Formalisation 29 (conditional expectation)**.**

```
conditional_expectation (sp, sts, mu) X Y =
  @f. (∃ s. f IN measurable (IMAGE Y sp, POW (IMAGE Y sp)) (s, POW s)) ∧
      ∀ g. g SUBSET (IMAGE Y sp) ⇒
      (fn_integral (g, POW g, pmf (sp, sts, mu) Y) f =
       fn_integral (sp, sts, mu)
             (λ x. X x * indicator_fn (PREIMAGE Y g INTER sp) x)),
```

where @x. P x *denotes the Hilbert's choice operator.* @x. P x *can be read informally as* "any x such that P x"*. As noted above, the Radon-Nikodým theorem is typically used to prove the uniqueness and existence of conditional expectation. Since a proof of equivalence between the construction above and the simpler form for discrete spaces can be proved without using the Radon-Nikodým theorem, the HOL proof of R.-N. theorem is left as future work.*

We conclude by examining the HOL theorem relating Formalisation 29 to the simpler form for finite spaces.

HOL Theorem 11 (reduction of cond. expectation for a finite r.v.)**.**

```
∀ (sp, sts, mu) s X Y y. random_variable X (sp, sts, mu) (s, POW s) ∧
  random_variable Y (sp, sts, mu) (IMAGE Y sp, POW (IMAGE Y sp)) ∧
  FINITE sp ∧ 0 IN s ∧ y IN IMAGE Y sp ∧
  (pmf (sp, sts, mu) Y {y} ≠ 0) ⇒
  (conditional_expectation (sp, sts, mu) X Y y =
   SIGMA (λ x. x * conditional_pmf (sp, sts, mu) X Y {x} {y}) (IMAGE X sp)).
```

This reduction corresponds directly to the intuition in Definition 22 about the conditional expectation of a discrete *random-variable. The proof follows from the basic definitions and the simplifications proved above for the Lebesgue integral. A similar reduction of Formalisation 29 could be proved for countably-infinite spaces; however, the number of side-conditions necessary to ensure that the infinite-summations involved in each stage of the proof converge, would result in a theorem that was tedious to prove and nearly as tedious to apply. By extending the work above to include theorems about integrability, the statement*

and proof of such a theorem would become much more elegant and usable. Therefore, the proof simplifying conditional-expectation for countably-infinite spaces is left as future work.

6 Summary

This chapter has explained formalisations for measure theory, Lebesgue integration, and probability theory, within the framework of the HOL4 theorem-prover. This work surpasses previous related work in a number of respects. Firstly, the HOL constructions make explicit the space involved in various definitions; this is beneficial in that general measure-theoretic definitions can be formalised and then proved equivalent to simpler definitions for discrete or continuous spaces. The formalisation presented above also extends previous work by including additional definitions and theorems, such as the development of conditional expectation, product measures, and Radon-Nikodým derivatives.

A Glossary of HOL4 notation

HOL4 notation	Description	Mathematical notation
{}	*empty set*	$\emptyset$
[]	*nil/empty list*	
l1 ++ l2	*list concatenation*	
hd :: tl	*list construction*	
(f ∘ g) x	*function composition*	$f(g(x))$
BIGINTER S	*set intersection*	$\bigcap_{x \in S} x$
BIGUNION S	*set union*	$\bigcup_{x \in S} x$
COMPL S	*set complementation*	$\overline{S}$
R CROSS S	*Cartesian product of sets*	$R \times S$
R DIFF S	*set difference*	$R - S$ or $R \setminus S$
DISJOINT R S	*disjoint sets*	$R \cap S = \emptyset$
enumerate S	*enumeration of countable set*	
IMAGE f S	*function image*	$f[S] = \bigcup_{x \in S} f(x)$
x IN S	*set membership*	$x \in S$
x INSERT S	*set construction*	$\{x\} \cup S$
R INTER S	*set intersection*	$R \cap S$
logr b r	*logarithm*	$\log_b r$
POW S	*power set*	$\mathcal{P}(S)$ or 2^S
PREIMAGE f S	*inverse-function image*	$f^{-1}[S] = \{x \mid f(x) \in S\}$
set l	*defines a set based on list* l	
R SUBSET S	*subset*	$R \subseteq S$
SIGMA f S	*finite summation*	$\sum_{x \in S} f(x)$
SUC n	*successor of a natural number*	$n + 1$
suminf f	*infinite summation*	$\lim_{n \to \infty} \sum_{i=0}^{n} f(i)$
f sums r	*infinite summation*	$\lim_{n \to \infty} \sum_{i=0}^{n} f(i) \to r$
sup S	*supremum of set*	$\sup S$
R UNION S	*set union*	$R \cup S$
(UNIV : $\alpha \to$ bool)	*universal set*	U or $\{x : \alpha \mid \top\}$

B measureTheory

[additive_def] Definition

```
   |- !m.
        additive m =
        !s t.
          s IN measurable_sets m /\ t IN measurable_sets m /\
          DISJOINT s t ==>
          (measure m (s UNION t) = measure m s + measure m t)
```

```
[algebra_def]  Definition
```

```
   |- !a.
        algebra a =
        subset_class (space a) (subsets a) /\ {} IN subsets a /\
        (!s. s IN subsets a ==> space a DIFF s IN subsets a) /\
        !s t.
          s IN subsets a /\ t IN subsets a ==> s UNION t IN subsets a
```

```
[closed_cdi_def]  Definition
```

```
   |- !p.
        closed_cdi p =
        subset_class (space p) (subsets p) /\
        (!s. s IN subsets p ==> space p DIFF s IN subsets p) /\
        (!f.
           f IN (UNIV -> subsets p) /\ (f 0 = {}) /\
           (!n. f n SUBSET f (SUC n)) ==>
           BIGUNION (IMAGE f UNIV) IN subsets p) /\
        !f.
          f IN (UNIV -> subsets p) /\
          (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) ==>
          BIGUNION (IMAGE f UNIV) IN subsets p
```

```
[countably_additive_def]  Definition
```

```
   |- !m.
        countably_additive m =
        !f.
          f IN (UNIV -> measurable_sets m) /\
          (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) /\
```

```
          BIGUNION (IMAGE f UNIV) IN measurable_sets m ==>
          measure m o f sums measure m (BIGUNION (IMAGE f UNIV))
```

```
[countably_subadditive_def]  Definition
```

```
   |- !m.
        countably_subadditive m =
        !f.
          f IN (UNIV -> measurable_sets m) /\
          BIGUNION (IMAGE f UNIV) IN measurable_sets m /\
          summable (measure m o f) ==>
          measure m (BIGUNION (IMAGE f UNIV)) <= suminf (measure m o f)
```

```
[increasing_def]  Definition
```

```
   |- !m.
        increasing m =
        !s t.
          s IN measurable_sets m /\ t IN measurable_sets m /\
          s SUBSET t ==>
          measure m s <= measure m t
```

```
[inf_measure_def]  Definition
```

```
   |- !m s.
        inf_measure m s =
        inf
          {r |
           ?f.
             f IN (UNIV -> measurable_sets m) /\
             (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) /\
             s SUBSET BIGUNION (IMAGE f UNIV) /\ measure m o f sums r}
```

```
[lambda_system_def]  Definition
```

```
   |- !gen lam.
        lambda_system gen lam =
        {l |
```

```
         l IN subsets gen /\
         !s.
           s IN subsets gen ==>
           (lam (l INTER s) + lam ((space gen DIFF l) INTER s) = lam s)}

[m_space_def]  Definition

   |- !sp sts mu. m_space (sp,sts,mu) = sp

[measurable_def]  Definition

   |- !a b.
        measurable a b =
        {f |
         sigma_algebra a /\ sigma_algebra b /\
         f IN (space a -> space b) /\
         !s. s IN subsets b ==> PREIMAGE f s INTER space a IN subsets a}

[measurable_sets_def]  Definition

   |- !sp sts mu. measurable_sets (sp,sts,mu) = sts

[measure_def]  Definition

   |- !sp sts mu. measure (sp,sts,mu) = mu

[measure_preserving_def]  Definition

   |- !m1 m2.
        measure_preserving m1 m2 =
        {f |
         f IN
         measurable (m_space m1,measurable_sets m1)
           (m_space m2,measurable_sets m2) /\
         !s.
           s IN measurable_sets m2 ==>
           (measure m1 (PREIMAGE f s INTER m_space m1) = measure m2 s)}
```

```
[measure_space_def]  Definition

   |- !m.
        measure_space m =
        sigma_algebra (m_space m,measurable_sets m) /\ positive m /\
        countably_additive m

[outer_measure_space_def]  Definition

   |- !m.
        outer_measure_space m =
        positive m /\ increasing m /\ countably_subadditive m

[positive_def]  Definition

   |- !m.
        positive m =
        (measure m {} = 0) /\
        !s. s IN measurable_sets m ==> 0 <= measure m s

[sigma_algebra_def]  Definition

   |- !a.
        sigma_algebra a =
        algebra a /\
        !c.
          countable c /\ c SUBSET subsets a ==> BIGUNION c IN subsets a

[sigma_def]  Definition

   |- !sp st.
        sigma sp st =
        (sp,BIGINTER {s | st SUBSET s /\ sigma_algebra (sp,s)})

[smallest_closed_cdi_def]  Definition

   |- !a.
        smallest_closed_cdi a =
```

```
        (space a,
         BIGINTER {b | subsets a SUBSET b /\ closed_cdi (space a,b)})

[space_def]  Definition

   |- !x y. space (x,y) = x

[subadditive_def]  Definition

   |- !m.
        subadditive m =
        !s t.
          s IN measurable_sets m /\ t IN measurable_sets m ==>
          measure m (s UNION t) <= measure m s + measure m t

[subset_class_def]  Definition

   |- !sp sts. subset_class sp sts = !x. x IN sts ==> x SUBSET sp

[subsets_def]  Definition

   |- !x y. subsets (x,y) = y

[ADDITIVE]  Theorem

   |- !m s t u.
        additive m /\ s IN measurable_sets m /\
        t IN measurable_sets m /\ DISJOINT s t /\ (u = s UNION t) ==>
        (measure m u = measure m s + measure m t)

[ADDITIVE_INCREASING]  Theorem

   |- !m.
        algebra (m_space m,measurable_sets m) /\ positive m /\
        additive m ==>
        increasing m

[ADDITIVE_SUM]  Theorem
```

```
   |- !m f n.
        algebra (m_space m,measurable_sets m) /\ positive m /\
        additive m /\ f IN (UNIV -> measurable_sets m) /\
        (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) ==>
        (sum (0,n) (measure m o f) =
         measure m (BIGUNION (IMAGE f (count n))))

[ALGEBRA_ALT_INTER]  Theorem

   |- !a.
        algebra a =
        subset_class (space a) (subsets a) /\ {} IN subsets a /\
        (!s. s IN subsets a ==> space a DIFF s IN subsets a) /\
        !s t.
          s IN subsets a /\ t IN subsets a ==> s INTER t IN subsets a

[ALGEBRA_COMPL]  Theorem

   |- !a s. algebra a /\ s IN subsets a ==> space a DIFF s IN subsets a

[ALGEBRA_DIFF]  Theorem

   |- !a s t.
        algebra a /\ s IN subsets a /\ t IN subsets a ==>
        s DIFF t IN subsets a

[ALGEBRA_EMPTY]  Theorem

   |- !a. algebra a ==> {} IN subsets a

[ALGEBRA_FINITE_UNION]  Theorem

   |- !a c.
        algebra a /\ FINITE c /\ c SUBSET subsets a ==>
        BIGUNION c IN subsets a

[ALGEBRA_INTER]  Theorem
```

```
   |- !a s t.
        algebra a /\ s IN subsets a /\ t IN subsets a ==>
        s INTER t IN subsets a

[ALGEBRA_INTER_SPACE]  Theorem

   |- !a s.
        algebra a /\ s IN subsets a ==>
        (space a INTER s = s) /\ (s INTER space a = s)

[ALGEBRA_SPACE]  Theorem

   |- !a. algebra a ==> space a IN subsets a

[ALGEBRA_SUBSET_LAMBDA_SYSTEM]  Theorem

   |- !m.
        algebra (m_space m,measurable_sets m) /\ positive m /\
        increasing m /\ additive m ==>
        measurable_sets m SUBSET
        lambda_system (m_space m,POW (m_space m)) (inf_measure m)

[ALGEBRA_UNION]  Theorem

   |- !a s t.
        algebra a /\ s IN subsets a /\ t IN subsets a ==>
        s UNION t IN subsets a

[CARATHEODORY]  Theorem

   |- !m0.
        algebra (m_space m0,measurable_sets m0) /\ positive m0 /\
        countably_additive m0 ==>
        ?m.
          (!s.
             s IN measurable_sets m0 ==>
             (measure m s = measure m0 s)) /\
```

```
          ((m_space m,measurable_sets m) =
           sigma (m_space m0) (measurable_sets m0)) /\ measure_space m

[CARATHEODORY_LEMMA]  Theorem

   |- !gsig lam.
        sigma_algebra gsig /\
        outer_measure_space (space gsig,subsets gsig,lam) ==>
        measure_space (space gsig,lambda_system gsig lam,lam)

[CLOSED_CDI_COMPL]  Theorem

   |- !p s.
        closed_cdi p /\ s IN subsets p ==> space p DIFF s IN subsets p

[CLOSED_CDI_DISJOINT]  Theorem

   |- !p f.
        closed_cdi p /\ f IN (UNIV -> subsets p) /\
        (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) ==>
        BIGUNION (IMAGE f UNIV) IN subsets p

[CLOSED_CDI_DUNION]  Theorem

   |- !p s t.
        {} IN subsets p /\ closed_cdi p /\ s IN subsets p /\
        t IN subsets p /\ DISJOINT s t ==>
        s UNION t IN subsets p

[CLOSED_CDI_INCREASING]  Theorem

   |- !p f.
        closed_cdi p /\ f IN (UNIV -> subsets p) /\ (f 0 = {}) /\
        (!n. f n SUBSET f (SUC n)) ==>
        BIGUNION (IMAGE f UNIV) IN subsets p

[COUNTABLY_ADDITIVE]  Theorem
```

```
   |- !m s f.
        countably_additive m /\ f IN (UNIV -> measurable_sets m) /\
        (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) /\
        (s = BIGUNION (IMAGE f UNIV)) /\ s IN measurable_sets m ==>
        measure m o f sums measure m s
```

[COUNTABLY_ADDITIVE_ADDITIVE] Theorem

```
   |- !m.
        algebra (m_space m,measurable_sets m) /\ positive m /\
        countably_additive m ==>
        additive m
```

[COUNTABLY_SUBADDITIVE] Theorem

```
   |- !m f s.
        countably_subadditive m /\ f IN (UNIV -> measurable_sets m) /\
        summable (measure m o f) /\ (s = BIGUNION (IMAGE f UNIV)) /\
        s IN measurable_sets m ==>
        measure m s <= suminf (measure m o f)
```

[COUNTABLY_SUBADDITIVE_SUBADDITIVE] Theorem

```
   |- !m.
        algebra (m_space m,measurable_sets m) /\ positive m /\
        countably_subadditive m ==>
        subadditive m
```

[INCREASING] Theorem

```
   |- !m s t.
        increasing m /\ s SUBSET t /\ s IN measurable_sets m /\
        t IN measurable_sets m ==>
        measure m s <= measure m t
```

[INCREASING_ADDITIVE_SUMMABLE] Theorem

```
   |- !m f.
```

```
        algebra (m_space m,measurable_sets m) /\ positive m /\
        increasing m /\ additive m /\
        f IN (UNIV -> measurable_sets m) /\
        (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) ==>
        summable (measure m o f)
```

[INF_MEASURE_AGREES] Theorem

```
   |- !m s.
        algebra (m_space m,measurable_sets m) /\ positive m /\
        countably_additive m /\ s IN measurable_sets m ==>
        (inf_measure m s = measure m s)
```

[INF_MEASURE_CLOSE] Theorem

```
   |- !m s e.
        algebra (m_space m,measurable_sets m) /\ positive m /\ 0 < e /\
        s SUBSET m_space m ==>
        ?f l.
          f IN (UNIV -> measurable_sets m) /\
          s SUBSET BIGUNION (IMAGE f UNIV) /\
          (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) /\
          measure m o f sums l /\ l <= inf_measure m s + e
```

[INF_MEASURE_COUNTABLY_SUBADDITIVE] Theorem

```
   |- !m.
        algebra (m_space m,measurable_sets m) /\ positive m /\
        increasing m ==>
        countably_subadditive (m_space m,POW (m_space m),inf_measure m)
```

[INF_MEASURE_EMPTY] Theorem

```
   |- !m.
        algebra (m_space m,measurable_sets m) /\ positive m ==>
        (inf_measure m {} = 0)
```

[INF_MEASURE_INCREASING] Theorem

```
   |- !m.
        algebra (m_space m,measurable_sets m) /\ positive m ==>
        increasing (m_space m,POW (m_space m),inf_measure m)
```

[INF_MEASURE_LE] Theorem

```
   |- !m s x.
        algebra (m_space m,measurable_sets m) /\ positive m /\
        increasing m /\
        x IN
        {r |
         ?f.
           f IN (UNIV -> measurable_sets m) /\
           s SUBSET BIGUNION (IMAGE f UNIV) /\ measure m o f sums r} ==>
        inf_measure m s <= x
```

[INF_MEASURE_NONEMPTY] Theorem

```
   |- !m g s.
        algebra (m_space m,measurable_sets m) /\ positive m /\
        s IN measurable_sets m /\ g SUBSET s ==>
        measure m s IN
        {r |
         ?f.
           f IN (UNIV -> measurable_sets m) /\
           (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) /\
           g SUBSET BIGUNION (IMAGE f UNIV) /\ measure m o f sums r}
```

[INF_MEASURE_OUTER] Theorem

```
   |- !m.
        algebra (m_space m,measurable_sets m) /\ positive m /\
        increasing m ==>
        outer_measure_space (m_space m,POW (m_space m),inf_measure m)
```

[INF_MEASURE_POS] Theorem

```
   |- !m g.
        algebra (m_space m,measurable_sets m) /\ positive m /\
        g SUBSET m_space m ==>
        0 <= inf_measure m g
```

[INF_MEASURE_POS0] Theorem

```
   |- !m g x.
        algebra (m_space m,measurable_sets m) /\ positive m /\
        x IN
        {r |
         ?f.
           f IN (UNIV -> measurable_sets m) /\
           (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) /\
           g SUBSET BIGUNION (IMAGE f UNIV) /\ measure m o f sums r} ==>
        0 <= x
```

[INF_MEASURE_POSITIVE] Theorem

```
   |- !m.
        algebra (m_space m,measurable_sets m) /\ positive m ==>
        positive (m_space m,POW (m_space m),inf_measure m)
```

[IN_MEASURABLE] Theorem

```
   |- !a b f.
        f IN measurable a b =
        sigma_algebra a /\ sigma_algebra b /\
        f IN (space a -> space b) /\
        !s. s IN subsets b ==> PREIMAGE f s INTER space a IN subsets a
```

[IN_MEASURE_PRESERVING] Theorem

```
   |- !m1 m2 f.
        f IN measure_preserving m1 m2 =
        f IN
        measurable (m_space m1,measurable_sets m1)
          (m_space m2,measurable_sets m2) /\
```

```
        !s.
          s IN measurable_sets m2 ==>
          (measure m1 (PREIMAGE f s INTER m_space m1) = measure m2 s)
```

[IN_SIGMA] Theorem

```
   |- !sp a x. x IN a ==> x IN subsets (sigma sp a)
```

[LAMBDA_SYSTEM_ADDITIVE] Theorem

```
   |- !g0 lam l1 l2.
        algebra g0 /\ positive (space g0,subsets g0,lam) ==>
        additive (space g0,lambda_system g0 lam,lam)
```

[LAMBDA_SYSTEM_ALGEBRA] Theorem

```
   |- !g0 lam.
        algebra g0 /\ positive (space g0,subsets g0,lam) ==>
        algebra (space g0,lambda_system g0 lam)
```

[LAMBDA_SYSTEM_CARATHEODORY] Theorem

```
   |- !gsig lam.
        sigma_algebra gsig /\
        outer_measure_space (space gsig,subsets gsig,lam) ==>
        !f.
          f IN (UNIV -> lambda_system gsig lam) /\
          (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) ==>
          BIGUNION (IMAGE f UNIV) IN lambda_system gsig lam /\
          lam o f sums lam (BIGUNION (IMAGE f UNIV))
```

[LAMBDA_SYSTEM_COMPL] Theorem

```
   |- !g0 lam l.
        algebra g0 /\ positive (space g0,subsets g0,lam) /\
        l IN lambda_system g0 lam ==>
        space g0 DIFF l IN lambda_system g0 lam
```

```
[LAMBDA_SYSTEM_EMPTY]  Theorem

   |- !g0 lam.
        algebra g0 /\ positive (space g0,subsets g0,lam) ==>
        {} IN lambda_system g0 lam

[LAMBDA_SYSTEM_INCREASING]  Theorem

   |- !g0 lam.
        increasing (space g0,subsets g0,lam) ==>
        increasing (space g0,lambda_system g0 lam,lam)

[LAMBDA_SYSTEM_INTER]  Theorem

   |- !g0 lam l1 l2.
        algebra g0 /\ positive (space g0,subsets g0,lam) /\
        l1 IN lambda_system g0 lam /\ l2 IN lambda_system g0 lam ==>
        l1 INTER l2 IN lambda_system g0 lam

[LAMBDA_SYSTEM_POSITIVE]  Theorem

   |- !g0 lam.
        positive (space g0,subsets g0,lam) ==>
        positive (space g0,lambda_system g0 lam,lam)

[LAMBDA_SYSTEM_STRONG_ADDITIVE]  Theorem

   |- !g0 lam g l1 l2.
        algebra g0 /\ positive (space g0,subsets g0,lam) /\
        g IN subsets g0 /\ DISJOINT l1 l2 /\
        l1 IN lambda_system g0 lam /\ l2 IN lambda_system g0 lam ==>
        (lam ((l1 UNION l2) INTER g) =
         lam (l1 INTER g) + lam (l2 INTER g))

[LAMBDA_SYSTEM_STRONG_SUM]  Theorem

   |- !g0 lam g f n.
        algebra g0 /\ positive (space g0,subsets g0,lam) /\
```

```
        g IN subsets g0 /\ f IN (UNIV -> lambda_system g0 lam) /\
        (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) ==>
        (sum (0,n) (lam o (\s. s INTER g) o f) =
         lam (BIGUNION (IMAGE f (count n)) INTER g))
```

[MEASUBABLE_BIGUNION_LEMMA] Theorem

```
   |- !a b f.
        sigma_algebra a /\ sigma_algebra b /\
        f IN (space a -> space b) /\
        (!s. s IN subsets b ==> PREIMAGE f s IN subsets a) ==>
        !c.
          countable c /\ c SUBSET IMAGE (PREIMAGE f) (subsets b) ==>
          BIGUNION c IN IMAGE (PREIMAGE f) (subsets b)
```

[MEASURABLE_BIGUNION_PROPERTY] Theorem

```
   |- !a b f.
        sigma_algebra a /\ sigma_algebra b /\
        f IN (space a -> space b) /\
        (!s. s IN subsets b ==> PREIMAGE f s IN subsets a) ==>
        !c.
          c SUBSET subsets b ==>
          (PREIMAGE f (BIGUNION c) = BIGUNION (IMAGE (PREIMAGE f) c))
```

[MEASURABLE_COMP] Theorem

```
   |- !f g a b c.
        f IN measurable a b /\ g IN measurable b c ==>
        g o f IN measurable a c
```

[MEASURABLE_COMP_STRONG] Theorem

```
   |- !f g a b c.
        f IN measurable a b /\ sigma_algebra c /\
        g IN (space b -> space c) /\
        (!x.
           x IN subsets c ==>
```

```
          PREIMAGE g x INTER IMAGE f (space a) IN subsets b) ==>
       g o f IN measurable a c
```

[MEASURABLE_COMP_STRONGER] Theorem

```
  |- !f g a b c t.
       f IN measurable a b /\ sigma_algebra c /\
       g IN (space b -> space c) /\ IMAGE f (space a) SUBSET t /\
       (!s. s IN subsets c ==> PREIMAGE g s INTER t IN subsets b) ==>
       g o f IN measurable a c
```

[MEASURABLE_DIFF_PROPERTY] Theorem

```
  |- !a b f.
       sigma_algebra a /\ sigma_algebra b /\
       f IN (space a -> space b) /\
       (!s. s IN subsets b ==> PREIMAGE f s IN subsets a) ==>
       !s.
         s IN subsets b ==>
         (PREIMAGE f (space b DIFF s) = space a DIFF PREIMAGE f s)
```

[MEASURABLE_I] Theorem

```
  |- !a. sigma_algebra a ==> I IN measurable a a
```

[MEASURABLE_LIFT] Theorem

```
  |- !f a b.
       f IN measurable a b ==>
       f IN measurable a (sigma (space b) (subsets b))
```

[MEASURABLE_POW_TO_POW] Theorem

```
  |- !m f.
       measure_space m /\ (measurable_sets m = POW (m_space m)) ==>
       f IN measurable (m_space m,measurable_sets m) (UNIV,POW UNIV)
```

[MEASURABLE_POW_TO_POW_IMAGE] Theorem

```
   |- !m f.
        measure_space m /\ (measurable_sets m = POW (m_space m)) ==>
        f IN
        measurable (m_space m,measurable_sets m)
          (IMAGE f (m_space m),POW (IMAGE f (m_space m)))
```

[MEASURABLE_PROD_SIGMA] Theorem

```
   |- !a a1 a2 f.
        sigma_algebra a /\ FST o f IN measurable a a1 /\
        SND o f IN measurable a a2 ==>
        f IN
        measurable a
          (sigma (space a1 CROSS space a2)
             (prod_sets (subsets a1) (subsets a2)))
```

[MEASURABLE_RANGE_REDUCE] Theorem

```
   |- !m f s.
        measure_space m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        ~(s = {}) ==>
        f IN
        measurable (m_space m,measurable_sets m)
          (s INTER IMAGE f (m_space m),
           POW (s INTER IMAGE f (m_space m)))
```

[MEASURABLE_SETS_SUBSET_SPACE] Theorem

```
   |- !m s.
        measure_space m /\ s IN measurable_sets m ==> s SUBSET m_space m
```

[MEASURABLE_SIGMA] Theorem

```
   |- !f a b sp.
        sigma_algebra a /\ subset_class sp b /\ f IN (space a -> sp) /\
        (!s. s IN b ==> PREIMAGE f s INTER space a IN subsets a) ==>
```

```
        f IN measurable a (sigma sp b)
```

[MEASURABLE_SIGMA_PREIMAGES] Theorem

```
   |- !a b f.
        sigma_algebra a /\ sigma_algebra b /\
        f IN (space a -> space b) /\
        (!s. s IN subsets b ==> PREIMAGE f s IN subsets a) ==>
        sigma_algebra (space a,IMAGE (PREIMAGE f) (subsets b))
```

[MEASURABLE_SUBSET] Theorem

```
   |- !a b.
        measurable a b SUBSET measurable a (sigma (space b) (subsets b))
```

[MEASURABLE_UP_LIFT] Theorem

```
   |- !sp a b c f.
        f IN measurable (sp,a) c /\ sigma_algebra (sp,b) /\
        a SUBSET b ==>
        f IN measurable (sp,b) c
```

[MEASURABLE_UP_SIGMA] Theorem

```
   |- !a b.
        measurable a b SUBSET measurable (sigma (space a) (subsets a)) b
```

[MEASURABLE_UP_SUBSET] Theorem

```
   |- !sp a b c.
        a SUBSET b /\ sigma_algebra (sp,b) ==>
        measurable (sp,a) c SUBSET measurable (sp,b) c
```

[MEASURE_ADDITIVE] Theorem

```
   |- !m s t u.
        measure_space m /\ s IN measurable_sets m /\
        t IN measurable_sets m /\ DISJOINT s t /\ (u = s UNION t) ==>
```

```
        (measure m u = measure m s + measure m t)
```

[MEASURE_COMPL] Theorem

```
   |- !m s.
        measure_space m /\ s IN measurable_sets m ==>
        (measure m (m_space m DIFF s) =
         measure m (m_space m) - measure m s)
```

[MEASURE_COUNTABLE_INCREASING] Theorem

```
   |- !m s f.
        measure_space m /\ f IN (UNIV -> measurable_sets m) /\
        (f 0 = {}) /\ (!n. f n SUBSET f (SUC n)) /\
        (s = BIGUNION (IMAGE f UNIV)) ==>
        measure m o f --> measure m s
```

[MEASURE_COUNTABLY_ADDITIVE] Theorem

```
   |- !m s f.
        measure_space m /\ f IN (UNIV -> measurable_sets m) /\
        (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) /\
        (s = BIGUNION (IMAGE f UNIV)) ==>
        measure m o f sums measure m s
```

[MEASURE_DOWN] Theorem

```
   |- !m0 m1.
        sigma_algebra (m_space m0,measurable_sets m0) /\
        measurable_sets m0 SUBSET measurable_sets m1 /\
        (measure m0 = measure m1) /\ measure_space m1 ==>
        measure_space m0
```

[MEASURE_EMPTY] Theorem

```
   |- !m. measure_space m ==> (measure m {} = 0)
```

[MEASURE_PRESERVING_LIFT] Theorem

```
   |- !m1 m2 a f.
        measure_space m1 /\ measure_space m2 /\
        (measurable_sets m2 = subsets (sigma (m_space m2) a)) /\
        f IN measure_preserving m1 (m_space m2,a,measure m2) ==>
        f IN measure_preserving m1 m2

[MEASURE_PRESERVING_SUBSET]  Theorem

   |- !m1 m2 a.
        measure_space m1 /\ measure_space m2 /\
        (measurable_sets m2 = subsets (sigma (m_space m2) a)) ==>
        measure_preserving m1 (m_space m2,a,measure m2) SUBSET
        measure_preserving m1 m2

[MEASURE_PRESERVING_UP_LIFT]  Theorem

   |- !m1 m2 f.
        f IN measure_preserving (m_space m1,a,measure m1) m2 /\
        sigma_algebra (m_space m1,measurable_sets m1) /\
        a SUBSET measurable_sets m1 ==>
        f IN measure_preserving m1 m2

[MEASURE_PRESERVING_UP_SIGMA]  Theorem

   |- !m1 m2 a.
        (measurable_sets m1 = subsets (sigma (m_space m1) a)) ==>
        measure_preserving (m_space m1,a,measure m1) m2 SUBSET
        measure_preserving m1 m2

[MEASURE_PRESERVING_UP_SUBSET]  Theorem

   |- !m1 m2.
        a SUBSET measurable_sets m1 /\
        sigma_algebra (m_space m1,measurable_sets m1) ==>
        measure_preserving (m_space m1,a,measure m1) m2 SUBSET
        measure_preserving m1 m2
```

```
[MEASURE_REAL_SUM_IMAGE]  Theorem

   |- !m s.
        measure_space m /\ s IN measurable_sets m /\
        (!x. x IN s ==> {x} IN measurable_sets m) /\ FINITE s ==>
        (measure m s = SIGMA (\x. measure m {x}) s)

[MEASURE_SPACE_ADDITIVE]  Theorem

   |- !m. measure_space m ==> additive m

[MEASURE_SPACE_REDUCE]  Theorem

   |- !m. (m_space m,measurable_sets m,measure m) = m

[MEASURE_SPACE_SUBSET]  Theorem

   |- !s s' m.
        s' SUBSET s /\ measure_space (s,POW s,m) ==>
        measure_space (s',POW s',m)

[MONOTONE_CONVERGENCE]  Theorem

   |- !m s f.
        measure_space m /\ f IN (UNIV -> measurable_sets m) /\
        (!n. f n SUBSET f (SUC n)) /\ (s = BIGUNION (IMAGE f UNIV)) ==>
        measure m o f --> measure m s

[OUTER_MEASURE_SPACE_POSITIVE]  Theorem

   |- !m. outer_measure_space m ==> positive m

[POW_ALGEBRA]  Theorem

   |- algebra (sp,POW sp)

[POW_SIGMA_ALGEBRA]  Theorem
```

```
   |- sigma_algebra (sp,POW sp)
```

[SIGMA_ALGEBRA] Theorem

```
   |- !p.
        sigma_algebra p =
        subset_class (space p) (subsets p) /\ {} IN subsets p /\
        (!s. s IN subsets p ==> space p DIFF s IN subsets p) /\
        !c.
          countable c /\ c SUBSET subsets p ==> BIGUNION c IN subsets p
```

[SIGMA_ALGEBRA_ALGEBRA] Theorem

```
   |- !a. sigma_algebra a ==> algebra a
```

[SIGMA_ALGEBRA_ALT] Theorem

```
   |- !a.
        sigma_algebra a =
        algebra a /\
        !f.
          f IN (UNIV -> subsets a) ==>
          BIGUNION (IMAGE f UNIV) IN subsets a
```

[SIGMA_ALGEBRA_ALT_DISJOINT] Theorem

```
   |- !a.
        sigma_algebra a =
        algebra a /\
        !f.
          f IN (UNIV -> subsets a) /\
          (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) ==>
          BIGUNION (IMAGE f UNIV) IN subsets a
```

[SIGMA_ALGEBRA_ALT_MONO] Theorem

```
   |- !a.
        sigma_algebra a =
```

```
        algebra a /\
        !f.
          f IN (UNIV -> subsets a) /\ (f 0 = {}) /\
          (!n. f n SUBSET f (SUC n)) ==>
          BIGUNION (IMAGE f UNIV) IN subsets a
```

[SIGMA_ALGEBRA_COUNTABLE_UNION] Theorem

```
   |- !a c.
        sigma_algebra a /\ countable c /\ c SUBSET subsets a ==>
        BIGUNION c IN subsets a
```

[SIGMA_ALGEBRA_ENUM] Theorem

```
   |- !a f.
        sigma_algebra a /\ f IN (UNIV -> subsets a) ==>
        BIGUNION (IMAGE f UNIV) IN subsets a
```

[SIGMA_ALGEBRA_FN] Theorem

```
   |- !a.
        sigma_algebra a =
        subset_class (space a) (subsets a) /\ {} IN subsets a /\
        (!s. s IN subsets a ==> space a DIFF s IN subsets a) /\
        !f.
          f IN (UNIV -> subsets a) ==>
          BIGUNION (IMAGE f UNIV) IN subsets a
```

[SIGMA_ALGEBRA_FN_DISJOINT] Theorem

```
   |- !a.
        sigma_algebra a =
        subset_class (space a) (subsets a) /\ {} IN subsets a /\
        (!s. s IN subsets a ==> space a DIFF s IN subsets a) /\
        (!s t.
           s IN subsets a /\ t IN subsets a ==>
           s UNION t IN subsets a) /\
        !f.
```

```
          f IN (UNIV -> subsets a) /\
          (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) ==>
          BIGUNION (IMAGE f UNIV) IN subsets a
```

[SIGMA_ALGEBRA_SIGMA] Theorem

```
   |- !sp sts. subset_class sp sts ==> sigma_algebra (sigma sp sts)
```

[SIGMA_POW] Theorem

```
   |- !s. sigma s (POW s) = (s,POW s)
```

[SIGMA_PROPERTY] Theorem

```
   |- !sp p a.
        subset_class sp p /\ {} IN p /\ a SUBSET p /\
        (!s. s IN p INTER subsets (sigma sp a) ==> sp DIFF s IN p) /\
        (!c.
           countable c /\ c SUBSET p INTER subsets (sigma sp a) ==>
           BIGUNION c IN p) ==>
        subsets (sigma sp a) SUBSET p
```

[SIGMA_PROPERTY_ALT] Theorem

```
   |- !sp p a.
        subset_class sp p /\ {} IN p /\ a SUBSET p /\
        (!s. s IN p INTER subsets (sigma sp a) ==> sp DIFF s IN p) /\
        (!f.
           f IN (UNIV -> p INTER subsets (sigma sp a)) ==>
           BIGUNION (IMAGE f UNIV) IN p) ==>
        subsets (sigma sp a) SUBSET p
```

[SIGMA_PROPERTY_DISJOINT] Theorem

```
   |- !sp p a.
        algebra (sp,a) /\ a SUBSET p /\
        (!s. s IN p INTER subsets (sigma sp a) ==> sp DIFF s IN p) /\
        (!f.
```

```
          f IN (UNIV -> p INTER subsets (sigma sp a)) /\ (f 0 = {}) /\
          (!n. f n SUBSET f (SUC n)) ==>
          BIGUNION (IMAGE f UNIV) IN p) /\
       (!f.
          f IN (UNIV -> p INTER subsets (sigma sp a)) /\
          (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) ==>
          BIGUNION (IMAGE f UNIV) IN p) ==>
       subsets (sigma sp a) SUBSET p

[SIGMA_PROPERTY_DISJOINT_LEMMA]  Theorem

  |- !sp a p.
       algebra (sp,a) /\ a SUBSET p /\ closed_cdi (sp,p) ==>
       subsets (sigma sp a) SUBSET p

[SIGMA_PROPERTY_DISJOINT_LEMMA1]  Theorem

  |- !a.
       algebra a ==>
       !s t.
         s IN subsets a /\ t IN subsets (smallest_closed_cdi a) ==>
         s INTER t IN subsets (smallest_closed_cdi a)

[SIGMA_PROPERTY_DISJOINT_LEMMA2]  Theorem

  |- !a.
       algebra a ==>
       !s t.
         s IN subsets (smallest_closed_cdi a) /\
         t IN subsets (smallest_closed_cdi a) ==>
         s INTER t IN subsets (smallest_closed_cdi a)

[SIGMA_PROPERTY_DISJOINT_WEAK]  Theorem

  |- !sp p a.
       algebra (sp,a) /\ a SUBSET p /\ subset_class sp p /\
       (!s. s IN p ==> sp DIFF s IN p) /\
       (!f.
```

```
           f IN (UNIV -> p) /\ (f 0 = {}) /\
           (!n. f n SUBSET f (SUC n)) ==>
           BIGUNION (IMAGE f UNIV) IN p) /\
        (!f.
           f IN (UNIV -> p) /\
           (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) ==>
           BIGUNION (IMAGE f UNIV) IN p) ==>
        subsets (sigma sp a) SUBSET p
```

[SIGMA_REDUCE] Theorem

```
   |- !sp a. (sp,subsets (sigma sp a)) = sigma sp a
```

[SIGMA_SUBSET] Theorem

```
   |- !a b.
        sigma_algebra b /\ a SUBSET subsets b ==>
        subsets (sigma (space b) a) SUBSET subsets b
```

[SIGMA_SUBSET_MEASURABLE_SETS] Theorem

```
   |- !a m.
        measure_space m /\ a SUBSET measurable_sets m ==>
        subsets (sigma (m_space m) a) SUBSET measurable_sets m
```

[SIGMA_SUBSET_SUBSETS] Theorem

```
   |- !sp a. a SUBSET subsets (sigma sp a)
```

[SMALLEST_CLOSED_CDI] Theorem

```
   |- !a.
        algebra a ==>
        subsets a SUBSET subsets (smallest_closed_cdi a) /\
        closed_cdi (smallest_closed_cdi a) /\
        subset_class (space a) (subsets (smallest_closed_cdi a))
```

[SPACE] Theorem

```
   |- !a. (space a,subsets a) = a
```

[SPACE_SIGMA] Theorem

```
   |- !sp a. space (sigma sp a) = sp
```

[SPACE_SMALLEST_CLOSED_CDI] Theorem

```
   |- !a. space (smallest_closed_cdi a) = space a
```

[STRONG_MEASURE_SPACE_SUBSET] Theorem

```
   |- !s s'.
        s' SUBSET m_space s /\ measure_space s /\
        POW s' SUBSET measurable_sets s ==>
        measure_space (s',POW s',measure s)
```

[SUBADDITIVE] Theorem

```
   |- !m s t u.
        subadditive m /\ s IN measurable_sets m /\
        t IN measurable_sets m /\ (u = s UNION t) ==>
        measure m u <= measure m s + measure m t
```

[UNIV_SIGMA_ALGEBRA] Theorem

```
   |- sigma_algebra (UNIV,UNIV)
```

[finite_additivity_sufficient_for_finite_spaces] Theorem

```
   |- !s m.
        sigma_algebra s /\ FINITE (space s) /\
        positive (space s,subsets s,m) /\
        additive (space s,subsets s,m) ==>
        measure_space (space s,subsets s,m)
```

[finite_additivity_sufficient_for_finite_spaces2] Theorem

```
|- !m.
     sigma_algebra (m_space m,measurable_sets m) /\
     FINITE (m_space m) /\ positive m /\ additive m ==>
     measure_space m
```

C lebesgueTheory

```
[RN_deriv_def]  Definition

|- !m v.
     RN_deriv m v =
     @f.
       (?s.
          f IN measurable (m_space m,measurable_sets m) (s,POW s)) /\
       !a.
         a IN measurable_sets m ==>
         (fn_integral m (\x. f x * indicator_fn a x) = v a)

[countable_measurable_fn_integral_def]  Definition

|- !m f.
     countable_measurable_fn_integral m f =
     (let e = enumerate (IMAGE f (m_space m)) in
        suminf
          ((\r. r * measure m (PREIMAGE f {r} INTER m_space m)) o e))

[finite_measurable_fn_integral_def]  Definition

|- !m f.
     finite_measurable_fn_integral m f =
     SIGMA (\r. r * measure m (PREIMAGE f {r} INTER m_space m))
       (IMAGE f (m_space m))

[fn_integral_def]  Definition
```

```
   |- !m f.
        fn_integral m f =
        pos_fn_integral m (\x. (if 0 < f x then f x else 0)) -
        pos_fn_integral m (\x. (if f x < 0 then ~f x else 0))

[indicator_fn_def]  Definition

   |- !s. indicator_fn s = (\x. (if x IN s then 1 else 0))

[mono_increasing_def]  Definition

   |- !f. mono_increasing f = !m n. m <= n ==> f m <= f n

[pos_fn_integral_def]  Definition

   |- !m f.
        pos_fn_integral m f =
        sup {r | ?g. r IN psfis m g /\ !x. g x <= f x}

[pos_simple_fn_def]  Definition

   |- !m f s a x.
        pos_simple_fn m f s a x =
        (!t. 0 <= f t) /\
        (!t.
           t IN m_space m ==>
           (f t = SIGMA (\i. x i * indicator_fn (a i) t) s)) /\
        (!i. i IN s ==> a i IN measurable_sets m) /\ (!i. 0 <= x i) /\
        FINITE s /\
        (!i j. i IN s /\ j IN s /\ ~(i = j) ==> DISJOINT (a i) (a j)) /\
        (BIGUNION (IMAGE a s) = m_space m)

[pos_simple_fn_integral_def]  Definition

   |- !m s a x.
        pos_simple_fn_integral m s a x =
        SIGMA (\i. x i * measure m (a i)) s
```

```
[prod_measure_def]  Definition

   |- !m0 m1.
        prod_measure m0 m1 =
        (\a.
           fn_integral m0 (\s0. measure m1 (PREIMAGE (\s1. (s0,s1)) a)))

[prod_measure_space_def]  Definition

   |- !m0 m1.
        prod_measure_space m0 m1 =
        (m_space m0 CROSS m_space m1,
         subsets
           (sigma (m_space m0 CROSS m_space m1)
              (prod_sets (measurable_sets m0) (measurable_sets m1))),
         prod_measure m0 m1)

[psfis_def]  Definition

   |- !m f.
        psfis m f =
        IMAGE (\(s,a,x). pos_simple_fn_integral m s a x) (psfs m f)

[psfs_def]  Definition

   |- !m f. psfs m f = {(s,a,x) | pos_simple_fn m f s a x}

[IN_psfis]  Theorem

   |- !m r f.
        r IN psfis m f ==>
        ?s a x.
          pos_simple_fn m f s a x /\
          (r = pos_simple_fn_integral m s a x)

[countable_neg_range_measurable_fn_integral_reduce]  Theorem

   |- !m f s p.
```

```
        measure_space m /\ (POW (m_space m) = measurable_sets m) /\
        positive m /\
        (\x. ~f x) IN
        measurable (m_space m,measurable_sets m) (s,POW s) /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        countable (m_space m) /\
        ~FINITE (s INTER IMAGE f (m_space m)) /\ ~(s = {}) /\ 0 IN s /\
        (\r.
           (if r < 0 then ~r else 0) *
           measure m (PREIMAGE f {r} INTER m_space m)) o
        enumerate (IMAGE f (m_space m)) sums p ==>
        (pos_fn_integral m (\x. (if f x < 0 then ~f x else 0)) = p)

[countable_pos_range_measurable_fn_integral_reduce]  Theorem

   |- !m f s p.
        measure_space m /\ (POW (m_space m) = measurable_sets m) /\
        positive m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        countable (m_space m) /\
        ~FINITE (s INTER IMAGE f (m_space m)) /\ ~(s = {}) /\ 0 IN s /\
        (\r.
           (if 0 < r then r else 0) *
           measure m (PREIMAGE f {r} INTER m_space m)) o
        enumerate (IMAGE f (m_space m)) sums p ==>
        (pos_fn_integral m (\x. (if 0 < f x then f x else 0)) = p)

[countable_range_measurable_fn_integral_reduce]  Theorem

   |- !m f s s' p p'.
        measure_space m /\ (POW (m_space m) = measurable_sets m) /\
        positive m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        (\x. ~f x) IN
        measurable (m_space m,measurable_sets m) (s',POW s') /\
        f IN measurable (m_space m,measurable_sets m) (s',POW s') /\
        countable (m_space m) /\
        ~FINITE (s INTER IMAGE f (m_space m)) /\
```

```
        ~FINITE (s' INTER IMAGE f (m_space m)) /\ ~(s = {}) /\
        ~(s' = {}) /\ 0 IN s /\ 0 IN s' /\
        (\r.
           (if 0 < r then r else 0) *
           measure m (PREIMAGE f {r} INTER m_space m)) o
        enumerate (IMAGE f (m_space m)) sums p /\
        (\r.
           (if r < 0 then ~r else 0) *
           measure m (PREIMAGE f {r} INTER m_space m)) o
        enumerate (IMAGE f (m_space m)) sums p' ==>
        (fn_integral m f = p - p')

[countable_range_measurable_fn_integral_reduce2]  Theorem

   |- !m f s s' p p'.
        measure_space m /\ (POW (m_space m) = measurable_sets m) /\
        positive m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        (\x. ~f x) IN
        measurable (m_space m,measurable_sets m) (s',POW s') /\
        f IN measurable (m_space m,measurable_sets m) (s',POW s') /\
        countable (m_space m) /\
        ~FINITE (s INTER IMAGE f (m_space m)) /\
        ~FINITE (s' INTER IMAGE f (m_space m)) /\ ~(s = {}) /\
        ~(s' = {}) /\ 0 IN s /\ 0 IN s' /\
        (\r.
           (if 0 < r then r else 0) *
           measure m (PREIMAGE f {r} INTER m_space m)) o
        enumerate (IMAGE f (m_space m)) sums p /\
        (\r.
           (if r < 0 then ~r else 0) *
           measure m (PREIMAGE f {r} INTER m_space m)) o
        enumerate (IMAGE f (m_space m)) sums p' ==>
        (fn_integral m f =
         suminf
           ((\r. r * measure m (PREIMAGE f {r} INTER m_space m)) o
            enumerate (IMAGE f (m_space m))))
```

```
[exists_mono_increasing_simple_approximation_of_countable_fn]  Theorem

   |- !m f s.
        measure_space m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        countable (m_space m) /\ ~(s = {}) ==>
        ?gp gn.
          (!i. ?s a x. pos_simple_fn m (gp i) s a x) /\
          (!i. ?s a x. pos_simple_fn m (gn i) s a x) /\
          (!x m n. m <= n ==> gp m x <= gp n x) /\
          (!x m n. m <= n ==> gn m x <= gn n x) /\
          (!x.
             x IN m_space m ==>
             (\i. gp i x) --> (\x. (if 0 < f x then f x else 0)) x) /\
          !x.
            x IN m_space m ==>
            (\i. gn i x) --> (\x. (if f x < 0 then ~f x else 0)) x

[finite_RN_deriv_reduce]  Theorem

   |- !m v.
        measure_space m /\ FINITE (m_space m) /\
        measure_space (m_space m,measurable_sets m,v) /\
        (POW (m_space m) = measurable_sets m) /\
        (!x. (measure m {x} = 0) ==> (v {x} = 0)) ==>
        !x.
          x IN m_space m /\ ~(measure m {x} = 0) ==>
          (RN_deriv m v x = v {x} / measure m {x})

[finite_integral_on_set]  Theorem

   |- !m s X g.
        measure_space m /\ FINITE (m_space m) /\
        X IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        ~(s = {}) /\ g IN measurable_sets m /\ 0 IN s ==>
        (fn_integral m (\x. X x * indicator_fn g x) =
         SIGMA (\r. r * measure m (PREIMAGE X {r} INTER g)) (IMAGE X g))
```

```
[finite_inter_neg_range_measurable_fn_integral_reduce]  Theorem

   |- !m f s.
        measure_space m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        FINITE (s INTER IMAGE f (m_space m) INTER {x | x <= 0}) /\
        countable (m_space m) /\ ~(s = {}) ==>
        (pos_fn_integral m (\x. (if f x < 0 then ~f x else 0)) =
         ~SIGMA (\r. r * measure m (PREIMAGE f {r} INTER m_space m))
            (IMAGE f (m_space m) INTER {x | x <= 0}))

[finite_inter_pos_range_measurable_fn_integral_reduce]  Theorem

   |- !m f s.
        measure_space m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        FINITE (s INTER IMAGE f (m_space m) INTER {x | 0 <= x}) /\
        countable (m_space m) /\ ~(s = {}) ==>
        (pos_fn_integral m (\x. (if 0 < f x then f x else 0)) =
         SIGMA (\r. r * measure m (PREIMAGE f {r} INTER m_space m))
           (IMAGE f (m_space m) INTER {x | 0 <= x}))

[finite_measurable_fn_integral_reduce]  Theorem

   |- !m f s.
        measure_space m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        FINITE (m_space m) /\ ~(s = {}) ==>
        (fn_integral m f = finite_measurable_fn_integral m f)

[finite_measurable_fn_integral_reduce2]  Theorem

   |- !m f s.
        measure_space m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        (POW (m_space m) = measurable_sets m) /\ FINITE (m_space m) /\
        ~(s = {}) ==>
        (fn_integral m f = SIGMA (\x. f x * measure m {x}) (m_space m))
```

```
[finite_measurable_fn_integral_reduce3]  Theorem

   |- !m f.
        measure_space m /\ (POW (m_space m) = measurable_sets m) /\
        FINITE (m_space m) ==>
        (fn_integral m f = SIGMA (\x. f x * measure m {x}) (m_space m))

[finite_measurable_fn_sum_of_pos_simple_fn]  Theorem

   |- !m f s.
        measure_space m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        ~(s = {}) /\ FINITE (m_space m) ==>
        ?s s' a a' x x'.
          pos_simple_fn m (\x. (if 0 < f x then f x else 0)) s a x /\
          pos_simple_fn m (\x. (if f x < 0 then ~f x else 0)) s' a' x'

[finite_neg_range_measurable_fn_integral_reduce]  Theorem

   |- !m f s.
        measure_space m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        FINITE (s INTER {x | x <= 0}) /\ ~(s = {}) ==>
        (pos_fn_integral m (\x. (if f x < 0 then ~f x else 0)) =
         ~SIGMA (\r. r * measure m (PREIMAGE f {r} INTER m_space m))
            (IMAGE f (m_space m) INTER {x | x <= 0}))

[finite_pos_range_measurable_fn_integral_reduce]  Theorem

   |- !m f s.
        measure_space m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        FINITE (s INTER {x | 0 <= x}) /\ ~(s = {}) ==>
        (pos_fn_integral m (\x. (if 0 < f x then f x else 0)) =
         SIGMA (\r. r * measure m (PREIMAGE f {r} INTER m_space m))
           (IMAGE f (m_space m) INTER {x | 0 <= x}))
```

```
[finite_prod_measure_reduce]  Theorem

   |- !m0 m1.
        measure_space m0 /\ measure_space m1 /\ FINITE (m_space m0) /\
        FINITE (m_space m1) /\
        (POW (m_space m0) = measurable_sets m0) /\
        (POW (m_space m1) = measurable_sets m1) ==>
        !a0 a1.
          a0 IN measurable_sets m0 /\ a1 IN measurable_sets m1 ==>
          (prod_measure m0 m1 (a0 CROSS a1) =
           measure m0 a0 * measure m1 a1)

[finite_range_measurable_fn_integral_reduce]  Theorem

   |- !m f s.
        measure_space m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        FINITE s /\ ~(s = {}) ==>
        (fn_integral m f = finite_measurable_fn_integral m f)

[finite_range_measurable_fn_sum_of_pos_simple_fn]  Theorem

   |- !m f s.
        measure_space m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        ~(s = {}) /\ FINITE s ==>
        ?s s' a a' x x'.
          pos_simple_fn m (\x. (if 0 < f x then f x else 0)) s a x /\
          pos_simple_fn m (\x. (if f x < 0 then ~f x else 0)) s' a' x'

[fn_integral_zero]  Theorem

   |- !m. fn_integral m (\x. 0) = 0

[indicator_fn_split]  Theorem

   |- !r s b.
        FINITE r /\ (BIGUNION (IMAGE b r) = s) /\
```

```
        (!i j.
           i IN r /\ j IN r /\ ~(i = j) ==> DISJOINT (b i) (b j)) ==>
        !a.
          a SUBSET s ==>
          (indicator_fn a =
           (\x. SIGMA (\i. indicator_fn (a INTER b i) x) r))
```

[integral_cmul_indicator] Theorem

```
   |- !m s c.
        measure_space m /\ s IN measurable_sets m /\ 0 <= c ==>
        (fn_integral m (\x. c * indicator_fn s x) = c * measure m s)
```

[integral_indicator] Theorem

```
   |- !m s.
        measure_space m /\ s IN measurable_sets m ==>
        (fn_integral m (indicator_fn s) = measure m s)
```

[integral_mul_indicator] Theorem

```
   |- !m s c.
        measure_space m /\ s IN measurable_sets m ==>
        (fn_integral m (\x. c * indicator_fn s x) = c * measure m s)
```

[integral_neg_cmul_indicator] Theorem

```
   |- !m s c.
        measure_space m /\ s IN measurable_sets m /\ c <= 0 ==>
        (fn_integral m (\x. c * indicator_fn s x) = c * measure m s)
```

[measure_space_finite_prod_measure] Theorem

```
   |- !m0 m1.
        measure_space m0 /\ measure_space m1 /\ FINITE (m_space m0) /\
        FINITE (m_space m1) /\
        (POW (m_space m0) = measurable_sets m0) /\
        (POW (m_space m1) = measurable_sets m1) ==>
```

```
        measure_space (prod_measure_space m0 m1)

[measure_space_finite_prod_measure_POW]  Theorem

   |- !m0 m1.
        measure_space m0 /\ measure_space m1 /\ FINITE (m_space m0) /\
        FINITE (m_space m1) /\
        (POW (m_space m0) = measurable_sets m0) /\
        (POW (m_space m1) = measurable_sets m1) ==>
        measure_space
          (m_space m0 CROSS m_space m1,
           POW (m_space m0 CROSS m_space m1),prod_measure m0 m1)

[measure_split]  Theorem

   |- !r b m.
        measure_space m /\ FINITE r /\
        (BIGUNION (IMAGE b r) = m_space m) /\
        (!i j. i IN r /\ j IN r /\ ~(i = j) ==> DISJOINT (b i) (b j)) /\
        (!i. i IN r ==> b i IN measurable_sets m) ==>
        !a.
          a IN measurable_sets m ==>
          (measure m a = SIGMA (\i. measure m (a INTER b i)) r)

[mono_increasing_converges_to_sup]  Theorem

   |- !f r.
        (!i. 0 <= f i) /\ mono_increasing f /\ f --> r ==>
        (r = sup (IMAGE f UNIV))

[pos_fn_integral_eq_mono_conv_limit]  Theorem

   |- !m f p fi ri r.
        measure_space m /\ (POW (m_space m) = measurable_sets m) /\
        positive m /\
        f IN measurable (m_space m,measurable_sets m) (p,POW p) /\
        (!x. mono_increasing (\i. fi i x)) /\
        (!x. x IN m_space m ==> (\i. fi i x) --> f x) /\
```

```
        (!i. ri i IN psfis m (fi i)) /\ ri --> r /\
        (!i x. fi i x <= f x) ==>
        (pos_fn_integral m f = r)

[pos_fn_integral_zero]  Theorem

   |- !m. measure_space m ==> (pos_fn_integral m (\x. 0) = 0)

[pos_psfis]  Theorem

   |- !r m f. measure_space m /\ positive m /\ r IN psfis m f ==> 0 <= r

[pos_simple_fn_integral_REAL_SUM_IMAGE]  Theorem

   |- !m f s a x P.
        measure_space m /\
        (!i. i IN P ==> pos_simple_fn m (f i) (s i) (a i) (x i)) /\
        FINITE P ==>
        ?s' a' x'.
          pos_simple_fn m (\t. SIGMA (\i. f i t) P) s' a' x' /\
          (pos_simple_fn_integral m s' a' x' =
           SIGMA (\i. pos_simple_fn_integral m (s i) (a i) (x i)) P)

[pos_simple_fn_integral_add]  Theorem

   |- !m f s a x g s' b y.
        measure_space m /\ pos_simple_fn m f s a x /\
        pos_simple_fn m g s' b y ==>
        ?s'' c z.
          pos_simple_fn m (\x. f x + g x) s'' c z /\
          (pos_simple_fn_integral m s a x +
           pos_simple_fn_integral m s' b y =
           pos_simple_fn_integral m s'' c z)

[pos_simple_fn_integral_indicator]  Theorem

   |- !m A.
        measure_space m /\ A IN measurable_sets m ==>
```

```
        ?s a x.
          pos_simple_fn m (indicator_fn A) s a x /\
          (pos_simple_fn_integral m s a x = measure m A)
```

[pos_simple_fn_integral_mono] Theorem

```
   |- !m f s a x g s' b y.
        measure_space m /\ pos_simple_fn m f s a x /\
        pos_simple_fn m g s' b y /\ (!x. f x <= g x) ==>
        pos_simple_fn_integral m s a x <=
        pos_simple_fn_integral m s' b y
```

[pos_simple_fn_integral_mono_on_mspace] Theorem

```
   |- !m f s a x g s' b y.
        measure_space m /\ pos_simple_fn m f s a x /\
        pos_simple_fn m g s' b y /\
        (!x. x IN m_space m ==> f x <= g x) ==>
        pos_simple_fn_integral m s a x <=
        pos_simple_fn_integral m s' b y
```

[pos_simple_fn_integral_mult] Theorem

```
   |- !m f s a x.
        measure_space m /\ pos_simple_fn m f s a x ==>
        !z.
          0 <= z ==>
          ?s' b y.
            pos_simple_fn m (\x. z * f x) s' b y /\
            (pos_simple_fn_integral m s' b y =
             z * pos_simple_fn_integral m s a x)
```

[pos_simple_fn_integral_present] Theorem

```
   |- !m f s a x g s' b y.
        measure_space m /\ pos_simple_fn m f s a x /\
        pos_simple_fn m g s' b y ==>
        ?z z' c k.
```

```
          (!t.
             t IN m_space m ==>
             (f t = SIGMA (\i. z i * indicator_fn (c i) t) k)) /\
          (!t.
             t IN m_space m ==>
             (g t = SIGMA (\i. z' i * indicator_fn (c i) t) k)) /\
          (pos_simple_fn_integral m s a x =
           pos_simple_fn_integral m k c z) /\
          (pos_simple_fn_integral m s' b y =
           pos_simple_fn_integral m k c z') /\ FINITE k /\
          (!i j.
             i IN k /\ j IN k /\ ~(i = j) ==> DISJOINT (c i) (c j)) /\
          (!i. i IN k ==> c i IN measurable_sets m) /\
          (BIGUNION (IMAGE c k) = m_space m) /\ (!i. 0 <= z i) /\
          !i. 0 <= z' i
```

[pos_simple_fn_integral_unique] Theorem

```
   |- !m f s a x s' b y.
        measure_space m /\ pos_simple_fn m f s a x /\
        pos_simple_fn m f s' b y ==>
        (pos_simple_fn_integral m s a x =
         pos_simple_fn_integral m s' b y)
```

[psfis_REAL_SUM_IMAGE] Theorem

```
   |- !m f a P.
        measure_space m /\ (!i. i IN P ==> a i IN psfis m (f i)) /\
        FINITE P ==>
        SIGMA a P IN psfis m (\t. SIGMA (\i. f i t) P)
```

[psfis_add] Theorem

```
   |- !m f g a b.
        measure_space m /\ a IN psfis m f /\ b IN psfis m g ==>
        a + b IN psfis m (\x. f x + g x)
```

[psfis_indicator] Theorem

```
   |- !m A.
        measure_space m /\ A IN measurable_sets m ==>
        measure m A IN psfis m (indicator_fn A)
```

[psfis_intro] Theorem

```
   |- !m a x P.
        measure_space m /\ (!i. i IN P ==> a i IN measurable_sets m) /\
        (!i. 0 <= x i) /\ FINITE P ==>
        SIGMA (\i. x i * measure m (a i)) P IN
        psfis m (\t. SIGMA (\i. x i * indicator_fn (a i) t) P)
```

[psfis_mono] Theorem

```
   |- !m f g a b.
        measure_space m /\ a IN psfis m f /\ b IN psfis m g /\
        (!x. f x <= g x) ==>
        a <= b
```

[psfis_mono_conv_mono] Theorem

```
   |- !m f p fi ri r g r'.
        measure_space m /\ (POW (m_space m) = measurable_sets m) /\
        positive m /\
        f IN measurable (m_space m,measurable_sets m) (p,POW p) /\
        (!x. mono_increasing (\i. fi i x)) /\
        (!x. x IN m_space m ==> (\i. fi i x) --> f x) /\
        (!i. ri i IN psfis m (fi i)) /\ ri --> r /\ r' IN psfis m g /\
        (!x. g x <= f x) ==>
        r' <= r
```

[psfis_mult] Theorem

```
   |- !m f a.
        measure_space m /\ a IN psfis m f ==>
        !z. 0 <= z ==> z * a IN psfis m (\x. z * f x)
```

[psfis_pos] Theorem

```
   |- !m f a. a IN psfis m f ==> !x. 0 <= f x
```

[psfis_present] Theorem

```
   |- !m f g a b.
        measure_space m /\ a IN psfis m f /\ b IN psfis m g ==>
        ?z z' c k.
          (!t.
             t IN m_space m ==>
             (f t = SIGMA (\i. z i * indicator_fn (c i) t) k)) /\
          (!t.
             t IN m_space m ==>
             (g t = SIGMA (\i. z' i * indicator_fn (c i) t) k)) /\
          (a = pos_simple_fn_integral m k c z) /\
          (b = pos_simple_fn_integral m k c z') /\ FINITE k /\
          (!i j.
             i IN k /\ j IN k /\ ~(i = j) ==> DISJOINT (c i) (c j)) /\
          (!i. i IN k ==> c i IN measurable_sets m) /\
          (BIGUNION (IMAGE c k) = m_space m) /\ (!i. 0 <= z i) /\
          !i. 0 <= z' i
```

[psfis_unique] Theorem

```
   |- !m f a b.
        measure_space m /\ a IN psfis m f /\ b IN psfis m f ==> (a = b)
```

[witness_mono_increasing_simple_approximation_of_countable_fn] Theorem

```
   |- !m f s.
        measure_space m /\
        f IN measurable (m_space m,measurable_sets m) (s,POW s) /\
        countable (m_space m) /\
        ~FINITE (s INTER IMAGE f (m_space m)) /\ ~(s = {}) ==>
        (let e = enumerate (s INTER IMAGE f (m_space m)) in
           (!x i j.
              i <= j ==>
```

```
   (\i x.
      (if
         x IN PREIMAGE f (IMAGE e (count i)) INTER m_space m
       then
         (if 0 < f x then f x else 0)
       else
         0)) i x <=
   (\i x.
      (if
         x IN PREIMAGE f (IMAGE e (count i)) INTER m_space m
       then
         (if 0 < f x then f x else 0)
       else
         0)) j x) /\
(!x i j.
   i <= j ==>
   (\i x.
      (if
         x IN PREIMAGE f (IMAGE e (count i)) INTER m_space m
       then
         (if f x < 0 then ~f x else 0)
       else
         0)) i x <=
   (\i x.
      (if
         x IN PREIMAGE f (IMAGE e (count i)) INTER m_space m
       then
         (if f x < 0 then ~f x else 0)
       else
         0)) j x) /\
(!x.
   x IN m_space m ==>
   (\i.
      (\i x.
         (if
            x IN
            PREIMAGE f (IMAGE e (count i)) INTER m_space m
          then
```

```
           (if 0 < f x then f x else 0)
         else
           0)) i x) -->
   (\x. (if 0 < f x then f x else 0)) x) /\
(!x.
   x IN m_space m ==>
   (\i.
      (\i x.
         (if
            x IN
            PREIMAGE f (IMAGE e (count i)) INTER m_space m
          then
            (if f x < 0 then ~f x else 0)
          else
            0)) i x) -->
   (\x. (if f x < 0 then ~f x else 0)) x) /\
(!i.
   pos_simple_fn m
     ((\i x.
         (if
            x IN
            PREIMAGE f (IMAGE e (count i)) INTER m_space m
          then
            (if 0 < f x then f x else 0)
          else
            0)) i) (count (SUC i))
     (\j.
        (if j = i then
           m_space m DIFF PREIMAGE f (IMAGE e (count i))
         else
           PREIMAGE f {e j} INTER m_space m))
     (\j.
        (if j = i then
           0
         else
           (if 0 < e j then e j else 0)))) /\
!i.
  pos_simple_fn m
```

```
                ((\i x.
                    (if
                       x IN
                       PREIMAGE f (IMAGE e (count i)) INTER m_space m
                     then
                       (if f x < 0 then ~f x else 0)
                     else
                       0)) i) (count (SUC i))
                (\j.
                   (if j = i then
                      m_space m DIFF PREIMAGE f (IMAGE e (count i))
                    else
                      PREIMAGE f {e j} INTER m_space m))
                (\j.
                   (if j = i then 0 else (if e j < 0 then ~e j else 0))))
```

D probabilityTheory

```
[conditional_expectation_def]  Definition

   |- !p X Y.
        conditional_expectation p X Y =
        @f.
          (?s.
             f IN
             measurable (IMAGE Y (p_space p),POW (IMAGE Y (p_space p)))
               (s,POW s)) /\
          !g.
            g SUBSET IMAGE Y (p_space p) ==>
            (fn_integral (g,POW g,pmf p Y) f =
             fn_integral p
               (\x.
                  X x * indicator_fn (PREIMAGE Y g INTER p_space p) x))

[conditional_pmf_def]  Definition
```

```
   |- !p X Y.
        conditional_pmf p X Y =
        (\(A,B). joint_pmf p X Y (A,B) / pmf p Y B)

[conditional_prob_def]  Definition

   |- !p e1 e2.
        conditional_prob p e1 e2 = prob p (e1 INTER e2) / prob p e2

[events_def]  Definition

   |- events = measurable_sets

[expectation_def]  Definition

   |- expectation = fn_integral

[indep_def]  Definition

   |- !p a b.
        indep p a b =
        a IN events p /\ b IN events p /\
        (prob p (a INTER b) = prob p a * prob p b)

[indep_families_def]  Definition

   |- !p q r.
        indep_families p q r = !s t. s IN q /\ t IN r ==> indep p s t

[joint_pmf_def]  Definition

   |- !p X Y.
        joint_pmf p X Y =
        (\(A,B).
           prob p (PREIMAGE X A INTER PREIMAGE Y B INTER p_space p))

[p_space_def]  Definition
```

```
   |- p_space = m_space

[pmf_def]  Definition

   |- !p X. pmf p X = (\A. prob p (PREIMAGE X A INTER p_space p))

[possibly_def]  Definition

   |- !p e. possibly p e = e IN events p /\ ~(prob p e = 0)

[prob_def]  Definition

   |- prob = measure

[prob_preserving_def]  Definition

   |- prob_preserving = measure_preserving

[prob_space_def]  Definition

   |- !p. prob_space p = measure_space p /\ (measure p (p_space p) = 1)

[probably_def]  Definition

   |- !p e. probably p e = e IN events p /\ (prob p e = 1)

[random_variable_def]  Definition

   |- !X p s.
        random_variable X p s =
        prob_space p /\ X IN measurable (p_space p,events p) s

[ABS_1_MINUS_PROB]  Theorem

   |- !p s.
        prob_space p /\ s IN events p /\ ~(prob p s = 0) ==>
        abs (1 - prob p s) < 1
```

```
[ABS_PROB]  Theorem

   |- !p s.
        prob_space p /\ s IN events p ==> (abs (prob p s) = prob p s)

[ADDITIVE_PROB]  Theorem

   |- !p.
        additive p =
        !s t.
          s IN events p /\ t IN events p /\ DISJOINT s t ==>
          (prob p (s UNION t) = prob p s + prob p t)

[COUNTABLY_ADDITIVE_PROB]  Theorem

   |- !p.
        countably_additive p =
        !f.
          f IN (UNIV -> events p) /\
          (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) /\
          BIGUNION (IMAGE f UNIV) IN events p ==>
          prob p o f sums prob p (BIGUNION (IMAGE f UNIV))

[EVENTS]  Theorem

   |- !a b c. events (a,b,c) = b

[EVENTS_ALGEBRA]  Theorem

   |- !p. prob_space p ==> algebra (p_space p,events p)

[EVENTS_COMPL]  Theorem

   |- !p s.
        prob_space p /\ s IN events p ==> p_space p DIFF s IN events p

[EVENTS_COUNTABLE_INTER]  Theorem
```

```
   |- !p c.
        prob_space p /\ c SUBSET events p /\ countable c /\
        ~(c = {}) ==>
        BIGINTER c IN events p

[EVENTS_COUNTABLE_UNION]  Theorem

   |- !p c.
        prob_space p /\ c SUBSET events p /\ countable c ==>
        BIGUNION c IN events p

[EVENTS_DIFF]  Theorem

   |- !p s t.
        prob_space p /\ s IN events p /\ t IN events p ==>
        s DIFF t IN events p

[EVENTS_EMPTY]  Theorem

   |- !p. prob_space p ==> {} IN events p

[EVENTS_INTER]  Theorem

   |- !p s t.
        prob_space p /\ s IN events p /\ t IN events p ==>
        s INTER t IN events p

[EVENTS_SIGMA_ALGEBRA]  Theorem

   |- !p. prob_space p ==> sigma_algebra (p_space p,events p)

[EVENTS_SPACE]  Theorem

   |- !p. prob_space p ==> p_space p IN events p

[EVENTS_UNION]  Theorem

   |- !p s t.
```

```
        prob_space p /\ s IN events p /\ t IN events p ==>
        s UNION t IN events p

[INCREASING_PROB]  Theorem

   |- !p.
        increasing p =
        !s t.
          s IN events p /\ t IN events p /\ s SUBSET t ==>
          prob p s <= prob p t

[INDEP_EMPTY]  Theorem

   |- !p s. prob_space p /\ s IN events p ==> indep p {} s

[INDEP_REFL]  Theorem

   |- !p a.
        prob_space p /\ a IN events p ==>
        (indep p a a = (prob p a = 0) \/ (prob p a = 1))

[INDEP_SPACE]  Theorem

   |- !p s. prob_space p /\ s IN events p ==> indep p (p_space p) s

[INDEP_SYM]  Theorem

   |- !p a b. prob_space p /\ indep p a b ==> indep p b a

[INTER_PSPACE]  Theorem

   |- !p s. prob_space p /\ s IN events p ==> (p_space p INTER s = s)

[POSITIVE_PROB]  Theorem

   |- !p.
        positive p =
        (prob p {} = 0) /\ !s. s IN events p ==> 0 <= prob p s
```

```
[PROB]  Theorem

   |- !a b c. prob (a,b,c) = c

[PROB_ADDITIVE]  Theorem

   |- !p s t u.
        prob_space p /\ s IN events p /\ t IN events p /\
        DISJOINT s t /\ (u = s UNION t) ==>
        (prob p u = prob p s + prob p t)

[PROB_COMPL]  Theorem

   |- !p s.
        prob_space p /\ s IN events p ==>
        (prob p (p_space p DIFF s) = 1 - prob p s)

[PROB_COMPL_LE1]  Theorem

   |- !p s r.
        prob_space p /\ s IN events p ==>
        (prob p (p_space p DIFF s) <= r = 1 - r <= prob p s)

[PROB_COUNTABLY_ADDITIVE]  Theorem

   |- !p s f.
        prob_space p /\ f IN (UNIV -> events p) /\
        (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) /\
        (s = BIGUNION (IMAGE f UNIV)) ==>
        prob p o f sums prob p s

[PROB_COUNTABLY_SUBADDITIVE]  Theorem

   |- !p f.
        prob_space p /\ IMAGE f UNIV SUBSET events p /\
        summable (prob p o f) ==>
        prob p (BIGUNION (IMAGE f UNIV)) <= suminf (prob p o f)
```

```
[PROB_COUNTABLY_ZERO]  Theorem

   |- !p c.
        prob_space p /\ countable c /\ c SUBSET events p /\
        (!x. x IN c ==> (prob p x = 0)) ==>
        (prob p (BIGUNION c) = 0)

[PROB_EMPTY]  Theorem

   |- !p. prob_space p ==> (prob p {} = 0)

[PROB_EQUIPROBABLE_FINITE_UNIONS]  Theorem

   |- !p s.
        prob_space p /\ s IN events p /\
        (!x. x IN s ==> {x} IN events p) /\ FINITE s /\
        (!x y. x IN s /\ y IN s ==> (prob p {x} = prob p {y})) ==>
        (prob p s = & (CARD s) * prob p {CHOICE s})

[PROB_EQ_BIGUNION_IMAGE]  Theorem

   |- !p.
        prob_space p /\ f IN (UNIV -> events p) /\
        g IN (UNIV -> events p) /\
        (!m n. ~(m = n) ==> DISJOINT (f m) (f n)) /\
        (!m n. ~(m = n) ==> DISJOINT (g m) (g n)) /\
        (!n. prob p (f n) = prob p (g n)) ==>
        (prob p (BIGUNION (IMAGE f UNIV)) =
         prob p (BIGUNION (IMAGE g UNIV)))

[PROB_EQ_COMPL]  Theorem

   |- !p s t.
        prob_space p /\ s IN events p /\ t IN events p /\
        (prob p (p_space p DIFF s) = prob p (p_space p DIFF t)) ==>
        (prob p s = prob p t)
```

```
[PROB_FINITELY_ADDITIVE]  Theorem

   |- !p s f n.
        prob_space p /\ f IN (count n -> events p) /\
        (!a b. a < n /\ b < n /\ ~(a = b) ==> DISJOINT (f a) (f b)) /\
        (s = BIGUNION (IMAGE f (count n))) ==>
        (sum (0,n) (prob p o f) = prob p s)

[PROB_INCREASING]  Theorem

   |- !p s t.
        prob_space p /\ s IN events p /\ t IN events p /\ s SUBSET t ==>
        prob p s <= prob p t

[PROB_INCREASING_UNION]  Theorem

   |- !p s f.
        prob_space p /\ f IN (UNIV -> events p) /\
        (!n. f n SUBSET f (SUC n)) /\ (s = BIGUNION (IMAGE f UNIV)) ==>
        prob p o f --> prob p s

[PROB_INDEP]  Theorem

   |- !p s t u.
        indep p s t /\ (u = s INTER t) ==>
        (prob p u = prob p s * prob p t)

[PROB_LE_1]  Theorem

   |- !p s. prob_space p /\ s IN events p ==> prob p s <= 1

[PROB_ONE_INTER]  Theorem

   |- !p s t.
        prob_space p /\ s IN events p /\ t IN events p /\
        (prob p t = 1) ==>
        (prob p (s INTER t) = prob p s)
```

```
[PROB_POSITIVE]  Theorem

   |- !p s. prob_space p /\ s IN events p ==> 0 <= prob p s

[PROB_PRESERVING]  Theorem

   |- !p1 p2.
        prob_preserving p1 p2 =
        {f |
         f IN
         measurable (p_space p1,events p1) (p_space p2,events p2) /\
         !s.
           s IN events p2 ==>
           (prob p1 (PREIMAGE f s INTER p_space p1) = prob p2 s)}

[PROB_PRESERVING_LIFT]  Theorem

   |- !p1 p2 a f.
        prob_space p1 /\ prob_space p2 /\
        (events p2 = subsets (sigma (m_space p2) a)) /\
        f IN prob_preserving p1 (m_space p2,a,prob p2) ==>
        f IN prob_preserving p1 p2

[PROB_PRESERVING_SUBSET]  Theorem

   |- !p1 p2 a.
        prob_space p1 /\ prob_space p2 /\
        (events p2 = subsets (sigma (p_space p2) a)) ==>
        prob_preserving p1 (p_space p2,a,prob p2) SUBSET
        prob_preserving p1 p2

[PROB_PRESERVING_UP_LIFT]  Theorem

   |- !p1 p2 f.
        f IN prob_preserving (p_space p1,a,prob p1) p2 /\
        sigma_algebra (p_space p1,events p1) /\ a SUBSET events p1 ==>
        f IN prob_preserving p1 p2
```

```
[PROB_PRESERVING_UP_SIGMA]  Theorem

   |- !p1 p2 a.
        (events p1 = subsets (sigma (p_space p1) a)) ==>
        prob_preserving (p_space p1,a,prob p1) p2 SUBSET
        prob_preserving p1 p2

[PROB_PRESERVING_UP_SUBSET]  Theorem

   |- !p1 p2.
        a SUBSET events p1 /\ sigma_algebra (p_space p1,events p1) ==>
        prob_preserving (p_space p1,a,prob p1) p2 SUBSET
        prob_preserving p1 p2

[PROB_REAL_SUM_IMAGE]  Theorem

   |- !p s.
        prob_space p /\ s IN events p /\
        (!x. x IN s ==> {x} IN events p) /\ FINITE s ==>
        (prob p s = SIGMA (\x. prob p {x}) s)

[PROB_REAL_SUM_IMAGE_FN]  Theorem

   |- !p f e s.
        prob_space p /\ e IN events p /\
        (!x. x IN s ==> e INTER f x IN events p) /\ FINITE s /\
        (!x y. x IN s /\ y IN s /\ ~(x = y) ==> DISJOINT (f x) (f y)) /\
        (BIGUNION (IMAGE f s) INTER p_space p = p_space p) ==>
        (prob p e = SIGMA (\x. prob p (e INTER f x)) s)

[PROB_SPACE]  Theorem

   |- !p.
        prob_space p =
        sigma_algebra (p_space p,events p) /\ positive p /\
        countably_additive p /\ (prob p (p_space p) = 1)

[PROB_SPACE_ADDITIVE]  Theorem
```

```
   |- !p. prob_space p ==> additive p

[PROB_SPACE_COUNTABLY_ADDITIVE]  Theorem

   |- !p. prob_space p ==> countably_additive p

[PROB_SPACE_INCREASING]  Theorem

   |- !p. prob_space p ==> increasing p

[PROB_SPACE_POSITIVE]  Theorem

   |- !p. prob_space p ==> positive p

[PROB_SUBADDITIVE]  Theorem

   |- !p s t u.
        prob_space p /\ t IN events p /\ u IN events p /\
        (s = t UNION u) ==>
        prob p s <= prob p t + prob p u

[PROB_UNIV]  Theorem

   |- !p. prob_space p ==> (prob p (p_space p) = 1)

[PROB_ZERO_UNION]  Theorem

   |- !p s t.
        prob_space p /\ s IN events p /\ t IN events p /\
        (prob p t = 0) ==>
        (prob p (s UNION t) = prob p s)

[PSPACE]  Theorem

   |- !a b c. p_space (a,b,c) = a

[countable_expectation]  Theorem
```

```
   |- !p X s s'.
        (POW (p_space p) = events p) /\ random_variable X p (s,POW s) /\
        random_variable (\x. ~X x) p (s',POW s') /\
        random_variable X p (s',POW s') /\ countable (p_space p) /\
        ~FINITE (s INTER IMAGE X (p_space p)) /\
        ~FINITE (s' INTER IMAGE X (p_space p)) /\ ~(s = {}) /\
        ~(s' = {}) /\ 0 IN s /\ 0 IN s' /\
        summable
          ((\x.
              (if 0 < x then x else 0) *
              prob p (PREIMAGE X {x} INTER p_space p)) o
           enumerate (IMAGE X (p_space p))) /\
        summable
          ((\x.
              (if x < 0 then ~x else 0) *
              prob p (PREIMAGE X {x} INTER p_space p)) o
           enumerate (IMAGE X (p_space p))) ==>
        (expectation p X =
         suminf
           ((\x. x * pmf p X {x}) o enumerate (IMAGE X (p_space p))))

[countable_expectation1]  Theorem

   |- !p X s s'.
        (POW (p_space p) = events p) /\ random_variable X p (s,POW s) /\
        random_variable (\x. ~X x) p (s',POW s') /\
        random_variable X p (s',POW s') /\ countable (p_space p) /\
        ~FINITE (s INTER IMAGE X (p_space p)) /\
        ~FINITE (s' INTER IMAGE X (p_space p)) /\ ~(s = {}) /\
        ~(s' = {}) /\ 0 IN s /\ 0 IN s' /\
        summable
          ((\x.
              (if 0 < x then x else 0) *
              prob p (PREIMAGE X {x} INTER p_space p)) o
           enumerate (IMAGE X (p_space p))) /\
        summable
          ((\x.
```

```
              (if x < 0 then ~x else 0) *
              prob p (PREIMAGE X {x} INTER p_space p)) o
            enumerate (IMAGE X (p_space p))) ==>
         (expectation p X =
          suminf
            ((\x. x * prob p (PREIMAGE X {x} INTER p_space p)) o
             enumerate (IMAGE X (p_space p))))
```

[finite_conditional_expectation] Theorem

```
   |- !p s X Y y.
        FINITE (p_space p) /\ random_variable X p (s,POW s) /\ 0 IN s /\
        random_variable Y p
          (IMAGE Y (p_space p),POW (IMAGE Y (p_space p))) /\
        y IN IMAGE Y (p_space p) /\ ~(pmf p Y {y} = 0) ==>
        (conditional_expectation p X Y y =
         SIGMA (\x. x * conditional_pmf p X Y ({x},{y}))
           (IMAGE X (p_space p)))
```

[finite_expectation] Theorem

```
   |- !p s X.
        FINITE (p_space p) /\ random_variable X p (s,POW s) /\
        ~(s = {}) ==>
        (expectation p X =
         SIGMA (\r. r * pmf p X {r}) (IMAGE X (p_space p)))
```

[finite_expectation1] Theorem

```
   |- !p s X.
        FINITE (p_space p) /\ random_variable X p (s,POW s) /\
        ~(s = {}) ==>
        (expectation p X =
         SIGMA (\r. r * prob p (PREIMAGE X {r} INTER p_space p))
           (IMAGE X (p_space p)))
```

[finite_marginal_product_space] Theorem

```
   |- !p X Y.
        (POW (p_space p) = events p) /\
        random_variable X p
          (IMAGE X (p_space p),POW (IMAGE X (p_space p))) /\
        random_variable Y p
          (IMAGE Y (p_space p),POW (IMAGE Y (p_space p))) /\
        FINITE (p_space p) ==>
        measure_space
          (IMAGE X (p_space p) CROSS IMAGE Y (p_space p),
           POW (IMAGE X (p_space p) CROSS IMAGE Y (p_space p)),
           (\a. prob p (PREIMAGE (\x. (X x,Y x)) a INTER p_space p)))
```

[pmf_lebesgue_thm1] Theorem

```
   |- !X p s A.
        random_variable X p s /\ A IN subsets s ==>
        (pmf p X A =
         fn_integral p (indicator_fn (PREIMAGE X A INTER p_space p)))
```

[pmf_lebesgue_thm2] Theorem

```
   |- !X p s A.
        random_variable X p s /\ A IN subsets s ==>
        (pmf p X A =
         fn_integral (space s,subsets s,pmf p X) (indicator_fn A))
```

[pmf_prob_space] Theorem

```
   |- !p X s.
        random_variable X p s ==> prob_space (space s,subsets s,pmf p X)
```

[pmf_x_eq_1_imp_pmf_y_eq_0] Theorem

```
   |- !X p s x.
        random_variable X p (s,POW s) /\ x IN s /\ (pmf p X {x} = 1) ==>
        !y. y IN s /\ ~(y = x) ==> (pmf p X {y} = 0)
```

[prob_x_eq_1_imp_prob_y_eq_0] Theorem

```
|- !p x.
     prob_space p /\ {x} IN events p /\ (prob p {x} = 1) ==>
     !y. {y} IN events p /\ ~(y = x) ==> (prob p {y} = 0)
```

References

[1] Józef Białas. The σ-additive measure theory. *Journal of Formalized Mathematics*, 2(2):263–270, 1991.

[2] Józef Białas. Properties of caratheodor's measure. *Journal of Formalized Mathematics*, 3(1):67–70, 1992.

[3] J. L. Doob. *Measure Theory*. Number 143 in Graduate Texts in Mathematics. Springer, 1991.

[4] John Harrison. *Theorem Proving with the Real Numbers*. Springer-Verlag, 1998.

[5] O. Hasan and S. Tahar. Verification of expectation properties for discrete random variables in hol. *Theorem Proving in Higher-Order Logics*, (4732):119–134, September 2007.

[6] Osman Hasan and Sofiène Tahar. Formalization of continuous probability distributions. In *CADE* [6], pages 3–18.

[7] Joe Hurd. *Formal Verification of Probabilistic Algorithms*. PhD thesis, University of Cambridge, 2002.

[8] Andrei Nikolaevich Kolmogorov. *Foundations of the Theory of Probability*. Chelsea Publishing Company, second edition, 1956.

[9] David R Lester. Topology in pvs: continuous mathematics with applications. In *AFM '07: Proceedings of the second workshop on Automated formal methods*, pages 11–20, New York, NY, USA, 2007. ACM.

[10] Paul Malliavin. *Integration and Probability*. Number 157 in Graduate Texts in Mathematics. Springer-Verlag, 1995.

[11] Andrzej Nędzusiak. σ-fields and probability. *Journal of Formalized Mathematics*, 1989.

[12] Andrzej Nędzusiak. Probability. *Journal of Formalized Mathematics*, 1(4):745–749, 1990.

[13] Stefan Richter. Formalizing integration theory, with an application to probabilistic algorithms. Master's thesis, Technische Universität München, 2003.

[14] Stan Wagon. *The Banach-Tarski Paradox*. Cambridge University Press, 1993.

[15] David Williams. *Probability with Martingales*. Cambridge Mathematical Textbooks. Cambridge University Press, 1991.