FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatik

Generic Construction of Probability Spaces for Paths of Stochastic Processes in Isabelle/HOL

Fabian Immler

FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatik

Generic Construction of Probability Spaces for Paths of Stochastic Processes in Isabelle/HOL

Allgemeine Konstruktion von Wahrscheinlichkeitsräumen für Pfade Stochastischer Prozesse in Isabelle/HOL

Author:	Fabian Immler
Supervisor:	Prof. Tobias Nipkow, Ph.D.
Advisor:	Johannes Hölzl
Submission Date:	October 15, 2012

I assure the single handed composition of this master's thesis only supported by declared resources

Munich, October 15, 2012 Fabian Immler

Acknowledgments

Tobias Nipkow introduced me to theoretical computer science, logics and theorem proving already at the beginning of my studies. I am very thankful for this and the fact that he gave me the opportunity to work in his group and participate in research already during my studies.

Johannes Hölzl was a very helpful and kind advisor. He managed to propose solutions in cases where I could not even express the problem. I want to thank him for reading draft versions of this thesis, however my feelings are ambivalent about the fact that he automated many of my interactive proofs when inspecting the source code ;)

Roland Immler and Henning Femmer helped to improve this thesis by commenting on draft versions, thank you!

Last but not least, I want to thank Hannelore and Franz-Josef Immler for their constant support and encouragement; simply for being such great parents.

Abstract

Stochastic processes are used in probability theory to describe the evolution of random systems over time. The principal mathematical problem is the construction of a probability space for the paths of stochastic processes. The Daniell-Kolmogorov theorem solves this problem: it shows how a family of finite-dimensional distributions defines the distribution of the stochastic process. The construction is generic, i.e., it works for discrete time as well as for continuous time.

Starting from the existing formalizations of measure theory and products of probability spaces in Isabelle/HOL, we provide a formal proof of the Daniell-Kolmogorov theorem in Isabelle/HOL. This requires us to formalize concepts from topology, namely polish spaces and regularity of measures on polish spaces.

These results can serve as a foundation to formalize for example discrete-time or continuous-time Markov chains, Markov decision processes, or physical phenomena like Brownian motion.

Contents

Acknowledgements **vii**

Abstract **ix**

1 Introduction **1**

1.1 Stochastic Processes 1

1.2 Paths of Stochastic Processes 2

1.3 Probability Spaces for Paths of Stochastic Processes 3

1.4 Formalization in Isabelle/HOL 3

1.5 Problem Statement 4

1.6 Contributions 4

1.7 Outline 4

2 Isabelle/HOL **7**

2.1 Source Code of the Formalization 8

2.2 Simplified Notations 8

2.3 Basic Terms and Types 8

2.4 Specifications 9

2.5 Sets 9

2.6 Functions 9

2.7 Sequences and Subsequences 10

2.8 Type Classes 10

2.9 Topology 10

2.10 Multivariate Analysis 11

2.11 Probability Theory 11

2.11.1 Measure Space and σ-Algebra 11

2.11.2 Dynkin Systems 12

2.11.3 Extension of a Content to a Measure 12

2.11.4 Measurability and Random Variables 13

2.11.5 Products of Measurable Spaces 13

2.11.6 Probability Spaces for Discrete-Time Markov Chains 14

3 Auxiliary Developments **15**

3.1 Countable Union of Finite Sets 15

3.2 Diagonal Subsequences 15

4 Topology 19

4.1 Characterization of Compact Sets with Total Boundedness 19

4.2 Characterization of Closed Sets with Infimum Distance 20

4.3 Topological Basis 20

4.4 Enumerable Basis 21

4.5 Polish Space 22

4.6 Regularity of Measures 23

5 Finite Map 25

5.1 Type Definition and Basic Properties 25

5.2 Metric Space 26

5.3 Product Set 26

5.4 Polish Finite Map 27

5.4.1 Completeness 27

5.4.2 Enumerable Basis 28

5.5 σ-Algebra of Products of Finite Maps 29

5.6 Measure Space Isomorphisms Involving Products of Finite Maps . . . 30

5.6.1 Functions and Finite Maps 31

5.6.2 Transferring the Domain of a Function 32

5.7 Discussion 33

6 Projective Limit 35

6.1 Projective Family 35

6.2 Generator and Content 36

6.3 Extension of Content to Measure 38

6.3.1 Probabilistic Argument 38

6.3.2 Construction of Diagonal Sequence 41

6.4 Summary: Formalization of the *Daniell-Kolmogorov* Theorem 42

6.5 Discussion 43

7 Conclusion 45

7.1 Formalization Efforts 45

7.2 Related Work 46

7.3 Future Work 47

7.3.1 Generalizations 47

7.3.2 Construction of Stochastic Processes 47

7.3.3 Applications of Stochastic Processes 48

Appendix **51**

8 Source Code **51**

Bibliography **53**

1 Introduction

Stochastic processes are a mathematical model to describe the evolution of the state of a random system over time. This work is about formalized analysis of stochastic processes. The only existing formalization of stochastic processes is limited to discrete time and space, we extend this by providing a formal proof of a well-known mathematical result which is necessary to analyze stochastic processes with continuous time and space.

1.1 Stochastic Processes

Science often constructs models of the real world in order to analyze them. If the state of a system evolves randomly in time, the system can be modeled as a stochastic process. Stochastic processes can be applied to diverse fields: For example, economists can model financial markets as stochastic processes, describing the evolution of prices. Stochastic processes can be used to describe the size of populations, when biologists model the birth and death of individuals probabilistically. Queuing theory has various applications in telecommunication and computer science and uses stochastic processes to describe the length of queues.

A concrete example of a stochastic process is given in figure 1.1 which depicts a very simple weather forecasting model (adapted from Kulkarni [15]): It consists of two states, warm and cold weather. Ignoring the past weather, there is a chance of 80% that the weather is warm tomorrow if it is warm today; if it is cold today, there is a chance of 30% that it stays cold. Switching from one day to the next, one follows transitions with the probability given by the labels on the arrows.

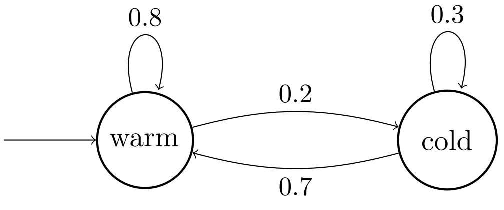

Figure 1.1: A stochastic process modeling weather forecast

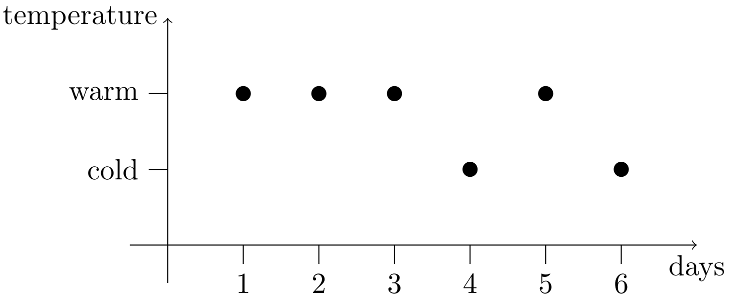

Figure 1.2: Example path of stochastic process with discrete time and state space

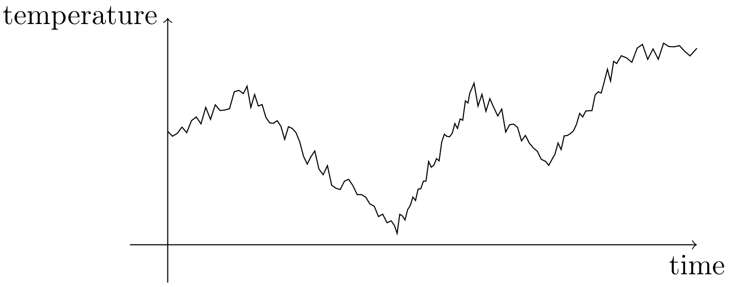

Figure 1.3: Example path of stochastic process with continuous time and state space

1.2 Paths of Stochastic Processes

In order to analyze stochastic processes, one investigates the probabilities of so-called paths. A path is the description of one possible behavior of the system, that means a path is a function that maps time to the state of the system at that time. The graph in figure 1.2 depicts an example of one possible path of the weather forecasting stochastic process. As time advances in discrete steps, time is represented by the natural numbers.

In our concrete example, it is easy to assign probabilities to paths: One multiplies the probabilities of the transitions needed to form the path, e.g., the probability for the path warm-warm-warm-cold-warm-cold is $0.8 \cdot 0.8 \cdot 0.2 \cdot 0.7 \cdot 0.2 = 1.79\%$.

Beware that our concrete example is a stochastic process with discrete time and state space, but stochastic processes may also – especially for modeling real-world phenomena – consist of continuous time and state space. Considering our weather forecasting example, it might be more appropriate or accurate to describe the evolution of temperature continuously. An example of a possible path of such a stochastic process is depicted in figure 1.3.

1.3 Probability Spaces for Paths of Stochastic Processes

The representation of stochastic processes is more involved in the continuous than in the discrete case. The same holds for the assignment of probabilities to paths. Nevertheless stochastic processes are well studied objects in mathematics, the analysis of stochastic processes is approached by constructing probability spaces for paths.

Probability spaces are a necessary concept in probability theory because assigning probabilities to outcomes of random experiments is a delicate task: It is difficult (or even impossible) to assign consistent probabilities to arbitrary sets of outcomes of random experiments. Probability spaces organize these outcomes (which are paths in the case of stochastic processes) in a way that makes it possible to assign consistent probabilities to them.

1.4 Formalization in Isabelle/HOL

The fact that it is necessary to introduce a nontrivial concept (probability spaces) in order to avoid inconsistencies shows that stochastic processes need to be analyzed with great care. It is therefore desirable to conduct the whole analysis of probability spaces for stochastic processes in a rigorous calculus. A rigorous calculus makes sure that every reasoning is based on purely syntactic transformations in a formal language. Performing mathematics in a rigorous calculus is called formalization of mathematics.

The fact that one works with purely syntactic transformations allows to formalize mathematics in computer systems. One computer system that is used for formalization of mathematics is the interactive theorem prover Isabelle/HOL. Interactive theorem provers are programs that allow the user to state theorems and guide the system through a proof. Proofs are mechanically checked, therefore an analysis carried out in an interactive theorem prover is highly reliable.

Formalized mathematics can be used to analyze models with infinite state space or to reason symbolically about parametric systems. This is not possible in other computer-assisted techniques like for example model checking which is usually restricted to fixed models with finite size.

The high reliability of formalized mathematics comes – due to the strictness that is necessary – at the cost of an higher effort. However, previous work by Hölzl [12] has shown that this approach is realistic in the sense that that stochastic processes with discrete time and state space can be analyzed in Isabelle/HOL.

1.5 Problem Statement

Unfortunately, Hölzl's construction of probability spaces [10] is limited to stochastic processes with discrete time and state space and does not generalize to the continuous case. In this work, we formalize a method to construct probability spaces for paths of arbitrary (i.e., discrete and continuous) stochastic processes. To be more specific, we formalize the (in probability theory) well-known *Daniell-Kolmogorov* theorem, which allows to construct probability spaces for arbitrary paths under the assumption of a family of probability spaces for paths with finite domains.

1.6 Contributions

The proof of the *Daniell-Kolmogorov* theorem exploits the fact that paths with finite domain can be equipped with a so-called *polish* topology. It then uses the fact that measures on polish spaces are regular. Neither polish spaces nor regularity of measures have been formalized in Isabelle/HOL yet. To the best of our knowledge, this is the first formalization of a construction of probability spaces for paths of arbitrary stochastic processes.

Our contributions are therefore the formalization of the following in Isabelle/HOL:

- polish spaces
- regularity of measures on polish spaces
- proof that functions with finite domain form a polish space
- the *Daniell-Kolmogorov* theorem, i.e., a generic construction of probability spaces for paths of stochastic processes

1.7 Outline

We give an introduction to Isabelle/HOL and our notations in chapter 2 and describe some general auxiliary developments in chapter 3.

We present our formalizations of polish spaces and regularity of measures in chapter 4. We explicitly formalize paths with finite domain and show that they are polish in chapter 5. Finally, the formalization of the *Daniell-Kolmogorov* theorem in Isabelle/HOL is described in chapter 6.

Chapter 7 outlines possible future developments, especially a means to construct concrete instances of stochastic processes via so called *kernels* and sketches possible applications of stochastic processes.

If you are not familiar with Isabelle/HOL, in order to get an overview about the formalization, you might want to jump directly from the introduction to Isabelle/HOL in chapter 2 to the formalization of the *Daniell-Kolmogorov* theorem in chapter 6 and look up the technical details when you feel the need.

2 Isabelle/HOL

Isabelle[1] is an interactive theorem proving framework. The way a user guides a proof in Isabelle is declarative and enforced by the (both human and machine readable) proof language Isabelle/Isar [25]: The user gives intermediate steps and lets the system fill in the missing details automatically. The intermediate steps can be relatively large because Isabelle/HOL provides powerful automatic tools: It can perform for example term rewriting, classical reasoning, use decision procedures for linear arithmetic and can invoke external automatic provers for first order logic. In order to guarantee trustworthiness, Isabelle follows the so-called LCF-approach: Theorems are abstract types that can only be constructed by the inference rules of the underlying logical system.

Isabelle can be instantiated with different kinds of object logics, however we only work with the instantiation of Isabelle with higher-order logic which is called Isabelle/HOL. In this chapter, we describe the existing formalizations in Isabelle/HOL that we use in our work and present our notational conventions.

We assume familiarity with logical notations, functional programming, and the λ-calculus. We only give a very rough overview over Isabelle/HOL, the interested reader shall consult for example the short overview about Isabelle/HOL by Nipkow [19] which is actually superseded by Nipkow's manual "Programming and Proving in Isabelle/HOL"[2] in the current distribution of Isabelle/HOL. A more comprehensive introduction is given by the "Isabelle/HOL Tutorial" [20], an updated version of which is also available in the distribution[3].

We can not give an extensive introduction into the different fields of mathematics that we treat in our formalization. We basically list the required concepts that are formalized in Isabelle/HOL. Please refer to the textbooks we chose to guide the principal part of our formalizations in topology [24], measure theory [5] or probability theory [4], since they provide suitable introductions to the respective fields. Other recommendable textbooks about measure and probability theory are from Ash [1] and Billingsley [6].

[1] http://isabelle.in.tum.de/

[2] http://isabelle.in.tum.de/dist/Isabelle2012/doc/prog-prove.pdf

[3] http://isabelle.in.tum.de/dist/Isabelle2012/doc/tutorial.pdf

2.1 Source Code of the Formalization

This thesis describes our formalization which we carry out in the Isabelle/HOL interactive theorem prover. You can browse and download the sources online[4] or find them on the CD. At some point in time, you might also find these developments in the Archive of Formal Proofs[5] or in the Isabelle distribution.

We give the name of corresponding definitions, lemmas or theorems in the source code of our formalization together with the definitions, lemmas and theorems we present in this document as in the following examples.

Definition 2.1. DEFINITION-IN-SOURCE:

Lemma 2.2. LEMMA-IN-SOURCE:

Theorem 2.3. THEOREM-IN-SOURCE:

2.2 Simplified Notations

In this thesis, we simplify notation where it is actually more complicated due to higher generality than necessary for our needs. Please refer to the source code if you need more details. We make use of notation that looks more like ordinary mathematics. But this is mostly just syntactic sugar in terms of infix or mixfix syntax that could as well be added to Isabelle/HOL which looks otherwise more like a functional programming language.

Apart from syntax, we try to stick close to the formalization in our descriptions, in particular we try not to simplify away the restrictions that come with the type system of Isabelle/HOL, which is more restrictive than ordinary set theory employed by mathematics.

2.3 Basic Terms and Types

The formal language of Isabelle/HOL follows the λ-calculus, i.e., function application is written $f\ t$ instead of $f(t)$. In order to stick closer to mathematical notation, we notate function abstraction as $(x \mapsto t)$ instead of $(\lambda x.\ t)$. Isabelle/HOL is a typed logic, the notation $t :: \tau$ means that the term t is of type τ. We also write $t_1, \ldots, t_n :: \tau$ to indicate that several terms are of type τ.

One basic type is $\mathbb{B}$, which denotes boolean truth values. Type $\mathbb{B}$ is used for logical formulas which are built from the usual operators for conjunction $\wedge$, disjunction$\vee$, implication $\longrightarrow$, and quantifiers $\exists, \forall$. Equality is written $=$, the negation thereof $\neq$.

[4] `http://home.in.tum.de/~immler/mastersthesis/`

[5] `http://afp.sf.net`

Types are built from further base types like $\mathbb{N}$ for natural numbers, $\mathbb{R}$ for real numbers, $\overline{\mathbb{R}}$ for real numbers extended with infinity ∞, $\mathbb{R}^n$ for vectors of real numbers with dimension n. We notate type variables with Greek letters α, β, and so on. The type of a function from α to β is written $\alpha \rightarrow \beta$, the type of a set of elements of type α is α *set*.

We write (a, b) for the pair of the terms $a :: \alpha$ and $b :: \beta$, we notate the type of the pair with $\alpha \times \beta$. More general, we write $(a_1, \ldots, a_n) :: \alpha_1 \times \cdots \times \alpha_n$ for tuples.

A special term is *undefined* $:: \alpha$, it denotes some fixed value of type α, which is not further specified.

2.4 Specifications

In order to define a constant *c* in terms of a variable x, we write *c* $x := \ldots$. In order to specify an arbitrary element that satisfies the property P, we write Hilbert choice as $(\varepsilon\ x.\ P\ x)$, i.e., $(\exists x.\ P\ x) \longrightarrow P(\varepsilon\ x.\ P\ x)$ holds.

2.5 Sets

We already mentioned the type of sets, namely *set*. $a \in A$ means that $a :: \alpha$ is an element of $A :: \alpha$ *set*. We write the negation with $a \notin A$. The set consisting of all elements of a specific type is called the type universe and we write *Univ* $:: \alpha$ *set* for the universe of type α. We build sets with set comprehension: $\{f\ x \mid P\ x\}$ denotes the set of all elements $f\ x$ depending on elements x satisfying a given predicate P. It is usually clear from the context, over which variables the predicate ranges.

We write union and intersection with $\cup$ and $\cap$, set difference for $A, B :: \alpha$ *set* as $A \setminus B = \{a \mid a \in A \wedge a \notin B\}$. We write $\sup A :: \alpha$ and $\inf A :: \alpha$ for the supremum and infimum of a set $A :: \alpha$ *set*. We write $\mathcal{P}(A) :: \alpha$ *set* for the power set of A, i.e., the set of all subsets of A.

2.6 Functions

For a function f, we write the preimage as $f^{-1}\ y = \{x \mid f\ x = y\}$. We write function composition with $\circ$, i.e., $(f \circ g)\ x = f\ (g\ x)$. We write the image of a set under a function with square brackets: $f[A] = \{f\ a \mid a \in A\}$ for $f :: \alpha \Rightarrow \beta, A :: \alpha$ *set*. We also notate the image of a type very sloppy like $f[\alpha]$, when we actually mean the image of a type universe $f[$*Univ* $:: \alpha]$.

HOL is a logic of total functions, functions $f :: \alpha \Rightarrow \beta$ are always defined on the whole type universe of α. One is, however, often interested in functions on only a subset of the type universe. This can be modeled by taking into account only functions that take the constant value *undefined* outside their domain. We write

$f|_I :: \alpha \Rightarrow \beta$ for the restriction of a function $f :: \alpha \Rightarrow \beta$ to the domain $I :: \alpha$ *set*. That means $f|_I\ i = f\ i$ if $i \in I$ and $f|_I\ i =$ *undefined* otherwise. Note that the composition of two restricted functions is not restricted in general. We therefore write restricted composition as $f \circ_I g$ such that $(f \circ_I g)\ x = f\ (g\ x)$ if $x \in I$ and *undefined* otherwise.

2.7 Sequences and Subsequences

Sequences are often used in analysis. A sequence in Isabelle/HOL is just a function $y :: \mathbb{N} \Rightarrow \alpha$. To make clear that we actually work with elements of the sequence, we write y_i for the ith element $y\ i$ and $(y_i)_{i \in \mathbb{N}}$ for the sequence as such. We call y' a subsequence of y, if there is a strictly monotonic function r such that $y'_i = y_{(r\ i)}$. In this case, we write $y' \preccurlyeq y$.

2.8 Type Classes

The formalization of topology is mostly based on the concept of type classes. A type class is (like in the functional programming language Haskell) used to provide a simple way of overloading. Type classes can also be seen like interfaces in object oriented languages. Isabelle/HOL extends this concept to axiomatic type classes, that means one can specify properties that the overloaded constant has to satisfy. An example of a type class that we use is *topological-space*, here a type α must provide a specification for the overloaded constant *open* :: α *set* $\Rightarrow \mathbb{B}$ which characterizes the open sets of a topology. If *open* satisfies the axioms of a topological space, i.e., if the empty set and the type universe are open, and if the union and finite intersection of open sets are open, then the type α is an instance of the type class *topological-space* which we denote by α :: *topological-space*. We also use the name of the type class as a predicate on types: *topological-space* (α) expresses that the type α is a topological space.

A further example of a type class denotes countable types. We write α :: *countable* for types α where the universe *Univ* :: α *set* is countable.

2.9 Topology

The formalization of topology in Isabelle/HOL is the foundation for the library for multivariate analysis which was ported from Harrison's Euclidean spaces for HOL-Light [9]. In addition to the type class *topological-space* with its parameter *open* for open sets in a topology, we use the type class *metric-space* for metric spaces, the overloaded constant *dist* gives the metric of the space. The open ball around x with radius e is denoted with *ball* $e\ x = \{y \mid$ *dist* $x\ y < e\}$.

If one can show that the metric of a metric space is complete, i.e., every Cauchy sequence (that is, a sequence of elements that get arbitrarily close according to the metric) converges, then the type represents a complete metric space, encoded in the type class *complete-space*. If a set X is complete, we write *complete* X. Closed sets are formalized as sets with open complement, compact sets are formalized as sets in which every sequence has a convergent subsequence. Predicates characterizing closed and compact sets are *closed* and *compact*.

2.10 Multivariate Analysis

The library for multivariate analysis is based on topological results on Euclidean spaces. Euclidean spaces are formalized in the type class *euclidean-space* as finite dimensional inner product spaces. $\mathbb{R}$ and $\mathbb{R}^n$ are instances of this type class.

2.11 Probability Theory

Let us give a brief introduction into the basic concepts of probability theory. Modern probability theory is centered around probability spaces. A probability space is a tuple $(\Omega, \mathcal{A}, P)$ consisting of a sample space Ω, a collection of events $\mathcal{A}$ and a probability distribution P which is defined on $\mathcal{A}$. The sample space is the set of elementary outcomes, an elementary outcome is the result of one run of the random experiment. An event is a subset of the sample space, $\mathcal{A}$ contains all events that can be assigned a consistent probability. Explicitly defining $\mathcal{A}$ is necessary, since (especially for uncountable, e.g., real-valued sample space) not every subset of the sample space can be assigned a consistent probability. Consistent means for example, that the probability of a (countable) union of disjoint events is the sum of the probabilities of the respective events.

2.11.1 Measure Space and σ-Algebra

Probability theory is based on measure theory, this also holds for the formalization in Isabelle/HOL. Measure theory has been formalized by Hölzl and Heller [11], for a detailed description of the current state of the formalization please refer to Hölzl [10].

Basic concepts of measure theory are formalized, a collection $\mathcal{A} :: \alpha$ *set set* of sets is a σ-algebra over $\Omega :: \alpha$ *set* if $\mathcal{A}$ contains Ω and the empty set $\emptyset$ and if $\mathcal{A}$ is closed under complement and countable union. A measure is a non-negative function from sets of a σ-algebra into the extended reals $\overline{\mathbb{R}}$ that assigns 0 to the empty set $\emptyset$ and that is additive on the union of a countable collection of pairwise disjoint elements of the σ-algebra. A function μ is additive, if $\mu(A \cup B) = \mu\ A + \mu\ B$.

A measure space in Isabelle/HOL is represented as a tuple consisting of a space Ω :: α *set*, a σ-algebra $\mathcal{A}$ :: α *set set* over Ω, and a measure μ :: α *set* $\Rightarrow \overline{\mathbb{R}}$ on $\mathcal{A}$. An important feature of the formalization is that measure spaces are collected in a particular type. We write $(\Omega, \mathcal{A}, \mu)$ :: α *measure-space* for measure spaces on type α. The fact that measure spaces form a type improves automation because every tuple in the type is a measure space by definition, therefore no hypotheses need to be discharged to apply results on measure spaces.

The type of measure spaces is also used to represent σ-algebras by taking as measure the function that is constant 0. We then speak of a measurable space $(\Omega, \mathcal{A})$. Isabelle/HOL provides the inductively defined σ operator, i.e., $\sigma(A)$ :: α *measure-space* is the smallest σ-algebra that contains the collection of sets A :: α *set set*. In this case, we say that A generates $\sigma(A)$, or that A is a generator.

Isabelle/HOL also provides the concept of Borel σ-algebras, i.e., the σ-algebra generated by the collection of all open sets. Open sets are determined by the type, we therefore write $\mathcal{B}$ :: α *set set* for the Borel σ-algebra of type α :: *topological-space* .

Probability spaces are measure spaces $(\Omega, \mathcal{A}, \mu)$ where the measure of the space equals one: $\mu\ \Omega = 1$. Then we call Ω the sample space, elements of $\mathcal{A}$ are called events and μ is called a probability measure.

2.11.2 Dynkin Systems

Dynkin systems can be used to inductively prove properties of σ-algebras – using a slightly different induction principle than the one given by the σ operator. If the generator is intersection-stable, Dynkin systems allow to weaken the inductive case for countable union to the countable union of *disjoint* sets.

2.11.3 Extension of a Content to a Measure

A content is an additive, non-negative function that is defined on a ring of sets (a ring is closed under intersection and union). A basic result from measure theory (Caratheodory's theorem) allows to extend this content to a measure on the σ-algebra generated by the ring. One way to do this is to show that the content is continuous at the empty set. That means, under the assumption of a decreasing sequence $(A_n)_{n \in \mathbb{N}}$ that converges to the empty set $\emptyset$, one needs to show that the content converges to 0.

Isabelle/HOL provides a useful helper function *extend-measure*. If you have a content μ on some generating set G over Ω, and if there exists an extension μ' such that $(\Omega, \sigma(G), \mu')$ is a measure space, then *extend-measure* $\Omega\ G\ \mu$ is a measure space that assigns to elements of the generator G the same value as the content μ.

2.11.4 Measurability and Random Variables

Given a measure space $(\Omega, \mathcal{A}, \mu)$ :: α *measure-space*, a subset $A \subseteq \Omega$ is called measurable if it is contained in the σ-algebra $\mathcal{A}$. Given another measure space $(\Omega', \mathcal{A}', \mu')$ :: β *measure-space*, a function $f :: \alpha \Rightarrow \beta$ is called $\mathcal{A}$-$\mathcal{A}'$-measurable, if preimages (restricted to Ω, which we omit in our notations) of the measurable sets $\mathcal{A}'$ under f are measurable with respect to $\mathcal{A}$. The function $A' \mapsto \mu\ (f^{-1}[A'])$ is a measure on $(\Omega', \mathcal{A}')$ and is called the push-forward measure, we also write *push* $f\ \mu\ A' = \mu\ (f^{-1}[A'])$

Random variable are measurable functions from a probability space into a σ-algebra.

2.11.5 Products of Measurable Spaces

Hölzl [10] formalized the construction of products of measure spaces, which requires the notion of *product sets* and *embeddings*. Our formalization also uses these concepts.

Product Sets

It is convenient (and also done in textbooks), to identify (indexed) products with functions in a suitable way: Given an *index* or *parameter set* $J :: \alpha$ ***set*** and a function $A :: \alpha \Rightarrow \beta$ ***set***, the *product (set)* of A over J is the dependent function space from J into A restricted to functions with the domain J. In products, for $j \in J$, we also write A_j instead of $A\ j$.

Definition 2.4. PiE-def':

$$\prod_{i \in I} A_i := \{f :: \alpha \Rightarrow \beta \mid (\forall i \in I.\ f\ i \in A_i) \wedge f = f|_I\}$$

We call a product set (or product) $\prod_{i \in I} A_i$ finite if the index set I is finite. Accordingly for infinite, countable or uncountable products.

Embeddings

Embeddings establish a connection from products with smaller index set to products with larger index set. For $J, I :: \iota$ ***set*** with $J \subseteq I$ and component spaces given by Ω_i for all $i \in I$, we define the embedding of $X \subseteq (\prod_{j \in J} \Omega_j)$ into $(\prod_{i \in I} \Omega_i)$ as follows. We omit Ω in the definition because it is usually clear from the context.

Definition 2.5. prod-emb-def':

$$\textsf{emb}\ I\ J\ X := \left\{ \omega \in (\prod_{i \in I} \Omega_i) \;\middle|\; \omega|_J \in X \right\}$$

σ-Algebra of Function Products

Literature defines the product $\bigotimes_{i\in I}\mathcal{A}_i$ of σ-algebras $(\Omega_i,\mathcal{A}_i)$ as the smallest σ-algebra such that the projection for every index $i\in I$ from $\prod_{i\in I}\Omega_i$ into Ω_i is $(\bigotimes_{i\in I}\mathcal{A}_i)$-$\mathcal{A}$-measurable, i.e., the embedding (being the preimage of projection) $\mathsf{emb}\ I\ \{i\}\ X$ needs to be contained in $\bigotimes_{i\in I}\mathcal{A}_i$ for every $X\in\mathcal{A}$. Note that all finite embeddings can be generated by singleton embeddings and vice versa. Therefore the product $\bigotimes_{i\in I}\mathcal{A}_i$ is generated by embeddings $\mathcal{G}$ of finite products of measurable sets, as it is done in Hölzl's [10] formalization.

Definition 2.6. PROJ-ALGEBRA-EQ', SETS-PIP':

$$\mathcal{G}_I\ \mathcal{A} := \left\{\mathsf{emb}\ I\ J\ (\prod_{j\in J} A_j)\middle|\ \mathit{finite}\ J\wedge\emptyset\neq J\wedge J\subseteq I\wedge(\forall j\in J.\ A_j\in\mathcal{A}_j)\right\}$$

$$\bigotimes_{i\in I}\mathcal{A}_i := \sigma(\mathcal{G}_I\ \mathcal{A})$$

2.11.6 Probability Spaces for Discrete-Time Markov Chains

Hölzl uses the product of measure spaces as described before to formalize a probability space for paths of discrete-time Markov chains. Discrete-time Markov chains are stochastic processes with discrete time and state space and the property that probabilities for transitions depend only on the current time and state, not on the history of states. For this section, we assume discrete time I (e.g., $I=\mathsf{Univ}::\mathbb{N}$) and finite state space S. The transitions of the Markov chain are given by a stochastic matrix $(\pi_{s,t})_{s,t\in S}$. That means, $\pi_{s,t}$ denotes the probability for the transition from state s to state t. Therefore π defines a probability space $(S,\mathcal{P}(S),\mu_s)$ for every state s.

The measurable space for (infinite) paths is the product measurable space given by $(\prod_{i\in I}S,\bigotimes_{i\in I}\mathcal{P}(S))$. Literature usually defines this measurable space as generated by cylinder sets, i.e., sets of infinite paths with a common prefix, which results in the same measurable space. In order to obtain a probability space on paths, Hölzl takes the product $(\prod_{(i,s)\in I\times S}S,\bigotimes_{(i,s)\in I\times S}\mathcal{P}(S),\mu_P)$ of the probability spaces for states. A formalized theorem about infinite products guarantees the existence of a probability measure μ_P on the product measurable space. Hölzl then casts this probability space into a probability space on paths by giving a measurable function path from the product of probability spaces to the measurable space of paths.

This construction can not be used to construct a probability space for arbitrary stochastic processes: path is not measurable for arbitrary products (especially for uncountable time or state space), therefore a different construction is necessary, which we provide with the *Daniell-Kolmogorov* theorem.

3 Auxiliary Developments

Some parts of the formalization turned out to be of a quite general nature. We therefore formalized them in a way that can be reused in more generic settings. The first is to show that a countable union of finite sets is countable. The second assumes a sequence of properties on sequences and constructs a (diagonal) subsequence which, at some point, satisfies an arbitrary property of the sequence.

3.1 Countable Union of Finite Sets

We show that a countable union of finite sets is countable by giving an injective function from the elements of the union to the natural numbers. We assume a sequence of sets $(J_n)_{n\in\mathbb{N}}$ such that every J_n is finite. Being finite, every J_n can be enumerated by some function enum_n. We associate to an element $j \in \bigcup_n J_n$ the pair consisting of the least index n such that $j \in J_n$ and the number m with $\mathit{enum}_n\ m = j$. This pair is then encoded as a natural number. We call the resulting function (depending on a sequence J of finite sets) $\mathit{to\text{-}nat}_J$ and obtain that it is injective:

Lemma 3.1. FINITE-SET-SEQUENCE.INJ-ON-UN-TO-NAT: *When J_n is finite for all $n \in \mathbb{N}$, then $\mathit{to\text{-}nat}_J$ is injective on $\bigcup_n J_n$.*

3.2 Diagonal Subsequences

In the course of our formalization it occurred twice to us, that starting from some sequence y, we had to iteratively construct a subsequence y' of y with some property, then a subsequence y'' of y' with some additional property and so on, iteratively taking subsequences $y^{(n+1)}$ from the subsequence $y^{(n)}$. Taking the diagonal $(y_n^{(n)})_{n\in\mathbb{N}}$ of this sequence of sequences, one gets a sequence that satisfies (apart from finitely many indices) every property (see figure 3.1).

We formalized this in the following setting. Assume a sequence of properties $(P_n)_{n\in\mathbb{N}}$ and a sequence $(y_i)_{i\in\mathbb{N}}$. Moreover assume that for every property P_n and subsequence y' of y, one can give a subsequence y'' of y' such that property $P_n\ y''$ holds:

$$\forall n.\ \forall y' \preccurlyeq y.\ \exists y'' \preccurlyeq y'.\ P_n\ y''$$

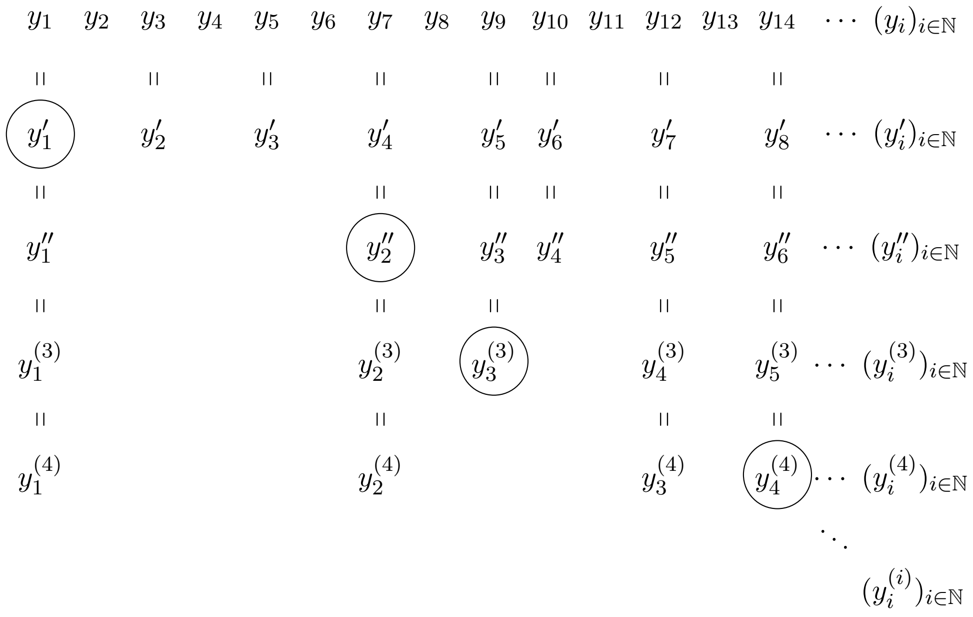
y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 y11 y12 y13 y14 ··· (yi)i∈ℕ
y'1 y'2 y'3 y'4 y'5 y'6 y'7 y'8 ··· (y'i)i∈ℕ
y''1 y''2 y''3 y''4 y''5 y''6 ··· (y''i)i∈ℕ
y(3)1 y(3)2 y(3)3 y(3)4 y(3)5 ··· (y(3)i)i∈ℕ
y(4)1 y(4)2 y(4)3 y(4)4 ··· (y(4)i)i∈ℕ
(y(i)i)i∈ℕ

Figure 3.1: Subsequences of y and (in circles) elements of the diagonal sequence

Exploiting this assumption, we (recursively) define subsequences $(y_i^{(n+1)})_{i\in\mathbb{N}}$ of $(y_i^{(n)})_{i\in\mathbb{N}}$ and obtain a sequence of sequences $(y^{(n)})_{n\in\mathbb{N}}$ such that for $y^{(n)}$ the property P_n holds. We use Isabelle/HOL's mechanism to define constants from primitive recursive specifications.

Definition 3.2. SUBSEQS.SEQSEQ.SIMPS:

$$\begin{aligned} y^{(0)} &:= y \\ y^{(n+1)} &:= (\varepsilon y''.\ y'' \preccurlyeq y^{(n)} \wedge P_n\ y'') \end{aligned}$$

If we take the assumption mentioned before into account, we can show that indeed, property P_n holds for $y^{(n+1)}$.

Lemma 3.3. SUBSEQS.SEQSEQ-EX:

$$P_n\ y^{(n+1)}$$

Our aim is to obtain a subsequence for which at some point, an arbitrary property holds. We therefore define the *diagonal subsequence* z as follows: Take as ith element the ith element of the ith subsequence $y^{(i)}$, as illustrated in figure 3.1.

Definition 3.4. SUBSEQS.DIAGSEQ:

$$z_i := y_i^{(i)}$$

We can prove that $(z_i)_{i\in\mathbb{N}}$ is a subsequence of y and for every n, apart from finitely many (actually n) indices in the beginning, z is a subsequence of $y^{(n)}$.

Lemma 3.5. SUBSEQS.SUBSEQ-DIAGSEQ:

$$z \preccurlyeq y$$

Lemma 3.6. SUBSEQS.DIAGSEQ-SEQSEQ:

$$(z_{i+n})_{i\in\mathbb{N}} \preccurlyeq (y_i^{(n)})_{i\in\mathbb{N}}$$

Now if the properties P_n are stable under the building of subsequences – i.e., if P_n holds for $y^{(n)}$ then it also holds for every subsequence of $y^{(n)}$ – and invariant for the prepending of finite sequences, the diagonal subsequence z satisfies arbitrary properties P_n if we take lemma 3.3 into account.

4 Topology

An important step in the proof of the *Daniell-Kolmogorov* theorem is the approximation of measures with measures of compact sets. This reasoning relies on *regularity* of measures, a notion that is based on results from topology. Measures are regular on *polish spaces*, which must have an enumerable *topological basis*. For the proof of regularity, we need an alternative characterization of compact sets and introduce a notion of distance between points and sets. These concepts were not yet formalized in Isabelle/HOL, so we describe our formalizations in the following.

4.1 Characterization of Compact Sets with Total Boundedness

We provide an alternative characterization of compact sets which is fairly standard in mathematics, but not yet formalized in Isabelle/HOL. The library already contains the implication that a compact set is totally bounded. This means that for every compact set C and every $e > 0$, there is a *finite* set of neighborhoods $\mathsf{ball}\ e$ which covers C.

We show the reverse implication: We assume that for every $e > 0$ there is a finite covering of C and show that the set is compact. We show this by going back to the definition, i.e., showing that every sequence in C has a convergent subsequence. We construct such a convergent subsequence as follows: For every $e = \frac{1}{n}$, we find some k and a subsequence in $\mathsf{ball}\ e\ k$ that contains infinitely many elements of the original sequence. From this property, we can build a diagonal sequence as introduced in section 3.2: The nth element of the diagonal sequence lies in a neighborhood $\mathsf{ball}\ \frac{1}{n}\ k_n$ (which is contained in $\mathsf{ball}\ \frac{1}{n-1}\ k_{n-1}$). Consequently, the diagonal sequence is a Cauchy sequence in C. If we assume that C is complete, the diagonal sequence converges.

Having both implications (together with the fact that compact sets are complete), we obtain that compact sets are characterized as complete and totally bounded.

Lemma 4.1. COMPACT-EQ-TOTALLY-BOUNDED:

$$\mathsf{compact}\ C \longleftrightarrow \mathsf{complete}\ C \wedge \left(\forall e > 0.\ \exists K.\ \mathit{finite}\ K \wedge C \subseteq \bigcup_{k \in K} \mathsf{ball}\ e\ k\right)$$

4.2 Characterization of Closed Sets with Infimum Distance

We needed to provide a notion of distance between a point and a set on a metric space $\alpha :: \mathit{metric\text{-}space}$ (where we assume a distance *dist*). This helps to characterize the elements of closed sets. The standard way to define this distance is to take the infimum of the possible distances:

Definition 4.2. INFDIST-DEF:

$$\mathit{infdist}\ x\ A := \inf\ \{\mathit{dist}\ x\ a \mid a \in A\}$$

Note that *infdist* is underspecified for the case $A = \emptyset$. We can show some sort of triangle inequality for the distance to a nonempty set:

Lemma 4.3. INFDIST-TRIANGLE: *Assume* $A \neq \emptyset$ *Then:*

$$\mathit{infdist}\ x\ A \leq \mathit{dist}\ x\ y + \mathit{infdist}\ y\ A$$

The characterization mentioned before is that an element is contained in a closed set, if and only if the distance to this set is zero:

Lemma 4.4. IN-CLOSURE-IFF-INFDIST-ZERO: *Assume* $A \neq \emptyset$ *and* *closed* A *Then:*

$$(x \in A \longleftrightarrow \mathit{infdist}\ x\ A = 0)$$

4.3 Topological Basis

From now on we assume a topological space. Spoken in Isabelle/HOL, we fix a type $\alpha :: \mathit{topological\text{-}space}$ and assume a predicate $\mathit{open} :: \alpha\ \mathit{set} \Rightarrow \mathbb{B}$ on sets that specifies whether the set is open in the topology given by α or not.

A set of open sets $\mathcal{T} :: \alpha\ \mathit{set}\ \mathit{set}$ is called *topological basis* iff every open set is the union of some sets in $\mathcal{T}$.

Definition 4.5. TOPOLOGICAL-BASIS-DEF:

$$\mathit{topological\text{-}basis}\ \mathcal{T} := \Big(\forall T \in \mathcal{T}.\ \mathit{open}\ T\Big) \wedge \\ \Big(\forall X.\ \mathit{open}\ X \longrightarrow \exists \mathcal{T}' \subseteq \mathcal{T}.\ X = \bigcup_{T' \in \mathcal{T}'} T'\Big)$$

An alternative, often more convenient characterization is given by the following lemma which allows to do the proofs in a more streamlined way: Fix x and X, assume $x \in X$ and *open* X, then find an element T of the basis which contains x and is a subset of X.

Lemma 4.6. TOPOLOGICAL-BASIS-IFF: *$\mathcal{T}$ with $(\forall T \in \mathcal{T}.\ \mathsf{open}\ T)$ is a topological basis iff:*

$$\forall X.\ \mathsf{open}\ X \longrightarrow (\forall x \in X.\ \exists T \in \mathcal{T}.\ x \in T \wedge T \subseteq X)$$

4.4 Enumerable Basis

One can obtain interesting results about topological spaces if one restricts its basis to a countable set. As a concrete example, think of the real numbers: The set of open intervals with rational coordinates is a countable basis. A type α :: $\mathsf{topological\text{-}space}$ is an instance of the class $\mathsf{enumerable\text{-}basis}$ if there exists a function $f :: \mathbb{N} \Rightarrow \alpha\ \mathsf{set}$ that enumerates a topological basis.

Definition 4.7. EX-ENUM-BASIS:

$$\mathsf{enumerable\text{-}basis}(\alpha) := \mathsf{topological\text{-}space}\,(\alpha) \wedge (\exists f.\ \mathsf{topological\text{-}basis}\ \ (f[\mathbb{N}]))$$

For the rest of this chapter we are going to assume a type α :: $\mathsf{enumerable\text{-}basis}$. In order to have something to work with, we define $\mathsf{enum\text{-}basis}$ as one possible enumeration of one possible basis $\mathsf{enum\text{-}basis}$:

Definition 4.8. ENUM-BASIS'-DEF:

$$\mathsf{enum\text{-}basis} := (\varepsilon f.\ \mathsf{topological\text{-}basis}\ \ (f[\mathbb{N}]))$$

From this arbitrary enumeration, we enumerate an alternative basis which is closed under (finite) union. This closure property comes in handy in some proofs. We assume a function $\mathsf{from\text{-}nat} :: \mathbb{N} \Rightarrow \mathbb{N}\ \mathsf{set}$, which enumerates all finite sets of type $\mathbb{N}$.

Definition 4.9. ENUM-BASIS-DEF:

$$\widetilde{\mathsf{enum\text{-}basis}}\ n := \bigcup_{m \in \mathsf{from\text{-}nat}\ \ n} \mathsf{enum\text{-}basis}\ m$$

The definition allows to prove that $\widetilde{\mathsf{enum\text{-}basis}}$ actually also enumerates a topological basis and that this basis is closed under (finite) union.

Lemma 4.10. ENUMERABLE-BASIS:

$$\mathsf{topological\text{-}basis}\ \ (\widetilde{\mathsf{enum\text{-}basis}}[\mathbb{N}])$$

Lemma 4.11. UNION-BASISI:

$$\widetilde{\mathsf{enum\text{-}basis}}\ m \cup \widetilde{\mathsf{enum\text{-}basis}}\ n \in \widetilde{\mathsf{enum\text{-}basis}}[\mathbb{N}]$$

Closure under union allows us to prove that every open set is the union of an increasing sequence of elements of the basis – consider the (enumerable) elements of the basis which make up the open set and incrementally take the union of these elements to obtain an increasing sequence. So for every open set X, there exists an increasing sequence $(S_n)_{n\in\mathbb{N}}$ that unions up to X.

Lemma 4.12. OPEN-IMP-UNION-OF-INCSEQ:

$$\forall X :: \alpha\ \textit{set}.\ \exists S :: \mathbb{N} \Rightarrow \alpha\ \textit{set}.\ \textit{open}\ X \longrightarrow \left(X = \bigcup_{n\in\mathbb{N}} S_n \wedge (\forall n.\ S_n \subseteq S_{n+1}) \right)$$

In a topological space with enumerable basis, we can represent open sets as a countable union of elements of the basis, we therefore have that the enumeration of the basis is a generator of the Borel sets $\mathcal{B} :: \alpha\ \textit{set set}$.

Lemma 4.13. BOREL-EQ-SIGMA-ENUM-BASIS:

$$\mathcal{B} :: \alpha\ \textit{set set} = \sigma\,(\textit{enum-basis}[\mathbb{N}])$$

Another interesting property that follows rather immediately for spaces with an enumerable basis is the fact that they possess a countable *dense set*. A set lies dense in a topological space, if every (nonempty) open set contains (at least) one element of the dense set. Recall the example from above: The rationals lie dense in the reals. The existence of a countable dense set $D[\mathbb{N}]$ can be stated as follows.

Lemma 4.14. COUNTABLE-DENSE-SET:

$$\exists D :: \mathbb{N} \Rightarrow \alpha.\ \forall A.\ (\textit{open}\ A \wedge A \neq \emptyset) \longrightarrow (\exists n.\ D\ n \in A)$$

4.5 Polish Space

Standard textbooks (like the one of von Querenburg [24]) define *polish spaces* as *completely metrizable* topological spaces with an enumerable basis. The notion of completeness is tied to a metric: A space is complete with respect to some metric.

In Isabelle/HOL, the metric is fixed for a given type – the metric *dist* is a parameter of the type class *metric-space*. That means, once we choose a type α :: *metric-space*, we are given the metric *dist* and can either prove that the type α is an instance of *complete-space* or not. From a mathematical (or set theoretic) point of view, if the fixed metric does not render the space complete, there may be a different metric in which case one would speak of a completely metrizable space. If one encountered such a case in Isabelle/HOL, one would need to define a copy of the type with the complete metric associated to it.

As a consequence, we define polish spaces as the type class of complete metric spaces with an enumerable basis:

Definition 4.15. POLISH-SPACE-CLASS-DEF:

$$\textsf{\textit{polish-space}}(\alpha) := \textsf{\textit{complete-space}}(\alpha) \wedge \textsf{\textit{enumerable-basis}}(\alpha)$$

To support that our formalizations are sensible and that we can actually profit from them, we show that Euclidean spaces are polish: In fact, Euclidean spaces over $\mathbb{R}$ are complete and cubes with rational coordinates form a countable basis.

Lemma 4.16. CLASSREL-ORDERED-EUCLIDEAN-SPACE-POLISH-SPACE:

$$(\alpha :: \textsf{\textit{euclidean-space}}) \longrightarrow (\alpha :: \textsf{\textit{polish-space}})$$

Another easy task is to show that the natural numbers are polish: We equip the natural numbers with the discrete topology, i.e., every subset is an open set, and define the discrete metric, i.e., a metric that assigns 1 to distinct points. The resulting space is complete and the enumeration of the singleton sets forms a basis. It follows:

Lemma 4.17. ARITY-POLISH-SPACE-NAT:

$$(\mathbb{N} :: \textsf{\textit{polish-space}})$$

4.6 Regularity of Measures

The main reason why we introduce polish spaces is the fact that one can show *regularity* for measures on polish spaces. Regularity means that the measure of an arbitrary Borel set can be approximated by open or compact sets. The proof requires the characterization of compact sets by total boundedness (lemma 4.1), the existence of a countable dense set (lemma 4.14), and the notion *infdist* for the distance between a point and a set which characterizes closed sets (lemma 4.3). With these means, we construct a Dynkin system (see section 2.11.2) for which the approximation property holds and conclude that the Dynkin system generates the Borel sets.

Theorem 4.18. INNER-REGULAR, OUTER-REGULAR:
Assume a finite measure μ on the Borel σ-algebra $\mathcal{B}$ of type α and some $B \in \mathcal{B}$. Then the following holds:

- *Inner regularity:*

$$\mu\ B = \sup \{\mu\ K \mid K \subseteq B \wedge \textsf{compact}\ K\}$$

- *Outer regularity:*

$$\mu\ B = \inf \{\mu\ U \mid U \supseteq B \wedge \textsf{open}\ U\}$$

Apart from the missing topological fundamentals we described in this chapter, the formalization of this theorem (the textbook proof of Bauer [5] takes roughly three pages) went very smoothly and without technical difficulties. One possible explanation is the fact that Bauer bases his arguments (except the topological ones) on concepts he introduced earlier in his book – the very same book that served as the inspiration of the formalization of measure theory in Isabelle/HOL.

5 Finite Map

We introduced polish spaces as a type class, which is why we now introduce a new type in order to profit from our developments. The basic point where we need a polish space in the proof of the *Daniell-Kolmogorov* theorem is where we approximate a measure on functions with finite domain. We collect all these functions in the type of *finite maps* (we call it map because the type we define explicitly carries a domain) and show that this type is polish. Since (for the *Daniell-Kolmogorov* theorem) we actually want to obtain a measure on functions, we need to provide a means to transfer results between these types: We are mostly interested in measures of sets of the respective types, therefore we provide measurable functions that convert between functions and finite maps in order to establish an isomorphism between the respective measure spaces.

5.1 Type Definition and Basic Properties

One way to represent functions with a finite domain in Isabelle/HOL is to explicitly give a domain $I :: \iota\ \mathsf{set}$ and a function $f :: \iota \Rightarrow \alpha$ that takes the constant value *undefined* outside this domain. Consult the discussion that follows in section 5.7 for considerations of other representations. We notate the type constructor for finite maps with domain in ι and codomain in α as $\iota \Rightarrow_F \alpha$.

Definition 5.1. TYPE-DEFINITION-FINMAP:

$$(\iota \Rightarrow_F \alpha) := \{(I :: \iota\ \mathsf{set}, f :: \iota \Rightarrow \alpha) \mid \mathit{finite}\ I \wedge f = f|_I\}$$

We provide the constructor *finmap* to define a finite map. Moreover we define *dom* to get the domain of the finite map and the operator $(\cdot)_F$ to perform function application. To imitate the behavior of the representing elements of the base type, the application of elements outside the domain of the finite map yields *undefined*.

Lemma 5.2. DOMAIN-FINMAP-OF, PROJ-FINMAP-OF: *Assume finite I, then:*

$$\mathsf{dom}\ (\mathsf{finmap}\ I\ f) = I$$

$$(\mathsf{finmap}\ I\ f)_F = f|_I$$

Given two finite maps with the same domain, equality is therefore characterized component-wise.

Lemma 5.3. FINMAP-EQ-IFF:

$$f = g \longleftrightarrow (\mathsf{dom}\ f = \mathsf{dom}\ g \wedge (\forall i \in \mathsf{dom}\ f.\ (f)_F\ i = (g)_F\ i))$$

5.2 Metric Space

One of our goals is to show that the type $\Rightarrow_F$ is polish. For this, we first need to provide a topology on $\Rightarrow_F$ which we do by defining a metric on $\Rightarrow_F$. We therefore assume a type ι and α :: $\mathsf{metric\text{-}space}$. We denote by dist_α the metric given by the type α. For finite maps with the same domain, one possible choice is the *Manhattan*-metric, summing over the distances for each coordinate. Since our type $\Rightarrow_F$ contains finite maps of all possible domains, we compare the finite maps as if they had a larger domain and associate to each index outside the domain the value $\mathsf{undefined}$. To make sure that we actually have a metric, i.e., that we assign the distance 0 only to finite maps that are equal, we include the difference between the domains.

Definition 5.4. DIST-FINMAP-DEF: *(Metric on $\Rightarrow_F$)*

$$\mathsf{dist}\ f\ g := \sum_{i \in \mathsf{dom}\ f \cup \mathsf{dom}\ g} \mathsf{dist}_\alpha\ ((f)_F\ i)\ ((g)_F\ i) + \\ |(\mathsf{dom}\ f \setminus \mathsf{dom}\ g) \cup (\mathsf{dom}\ g \setminus \mathsf{dom}\ f)|$$

We establish that $\Rightarrow_F$ associated with this metric forms a metric space. In this topological space, we have that the projection to a single index is continuous.

Lemma 5.5. ARITY-METRIC-SPACE-FINMAP:

$$(\iota \Rightarrow_F \alpha) :: \mathsf{metric\text{-}space}$$

Lemma 5.6. CONTINUOUS-PROJ: *The projection $(x \mapsto (x)_F\ i)$ is a continuous mapping.*

5.3 Product Set

Similar to definition 2.4 of the product set of functions, we define a product set of finite maps: For $I :: \iota$ and $A :: \iota \Rightarrow \alpha\ \mathsf{set}$, we define $\prod^F_{i \in I} A_i :: (\iota \Rightarrow_F \alpha)\ \mathsf{set}$ as the set of all finite maps that range at index i in the set given by A_i. Note that (compared to $\prod$) A is still just a regular function and that the restriction to the domain can now be declared explicitly. Moreover, when using $(\cdot)_F$ to interpret the elements as functions, they are restricted to I, just like for the product of functions.

Definition 5.7. Pi'-def:

$$\prod_{i\in I}^{F} A_i = \{f \mid \mathsf{dom}\ f = I \wedge (\forall i \in I.\ (f)_F\ i \in A_i)\}$$

The connection between products of finite maps and products of functions is given by the following lemma: Projecting all finite maps of a product yields the according product on functions.

Lemma 5.8. Pi-Pi':

$$\prod_{i\in I} A_i = \left\{ (f)_F \;\middle|\; f \in \prod_{i\in I}^{F} A_i \right\}$$

Consider $\iota \Rightarrow_F \alpha$, with $\alpha :: \mathsf{metric\text{-}space}$. The metric induces a useful topology and the notion of product is appropriate: A product of open sets in α is an open set of finite maps:

Lemma 5.9. open-Pi'I:

$$\big(\forall i \in I.\ \mathsf{open}_\alpha\ (A_i)\big) \longrightarrow \mathsf{open}\ \left(\prod_{i\in I}^{F} A_i\right)$$

5.4 Polish Finite Map

We introduced finite maps in order to use regularity of measures, for this we need to show that finite maps are polish. We are going to show $(\iota \Rightarrow_F \alpha) :: \mathsf{polish\text{-}space}$ for $\iota :: \mathsf{countable}$ and $\alpha :: \mathsf{polish\text{-}space}$. To this end, we are going to show that $\iota \Rightarrow_F \alpha$ is complete and possesses an enumerable basis.

5.4.1 Completeness

We show the completeness of finite maps $\iota \Rightarrow_F \alpha$ with $\alpha :: \mathsf{complete\text{-}space}$ in a standard way: Every Cauchy sequence converges. We establish that every Cauchy sequence of finite maps $(f_n)_{n\in\mathbb{N}}$ stabilizes at a certain domain: Since the elements of f get arbitrarily close, there exists an N with $\forall n \geq N.\ \mathsf{dist}\ (f_n)\ (f_N) < 1$ from which we can conclude that all elements after f_N have the same domain:

$$\forall n \geq N.\ \mathsf{dom}\ f_n = \mathsf{dom}\ f_N$$

The remainder of the argument is fairly standard: For every component i, the sequence $((f_n)_F\ i)_{n\in\mathbb{N}}$ converges to some g_i because $(f_n)_F\ i$ is of type α, which is complete. Since equality is defined component-wise (lemma 5.3), we can conclude that the sequence $(f_n)_{n\in\mathbb{N}}$ converges to the finite map given by $\mathsf{finmap}\ (\mathsf{dom}\ f_N)\ (i \mapsto g_i)$. We therefore have:

Lemma 5.10. ARITY-COMPLETE-SPACE-FINMAP:

$$\textsf{\textit{complete-space}}(\alpha) \longrightarrow \textsf{\textit{complete-space}}(\iota \Rightarrow_F \alpha)$$

5.4.2 Enumerable Basis

As a first step, we show that finite maps with a countable domain and countable codomain are countable, too: We can represent a finite map f with $\textsf{\textit{dom}}\ f = \{i_1, \ldots, i_n\}$ as a list of pairs of elements of the domain together with its associated value $((i_1, (f)_F\ i_1), \ldots, (i_n, (f)_F\ i_n))$. The pairs are countable since the components range over countable domains, moreover lists of countable elements are countable.

Lemma 5.11. ARITY-COUNTABLE-FINMAP:

$$(\textsf{\textit{countable}}(\iota) \wedge \textsf{\textit{countable}}(\alpha)) \longrightarrow \textsf{\textit{countable}}(\iota \Rightarrow_F \alpha)$$

If we assume $\alpha :: \textsf{\textit{enumerable-basis}}$, we may assume an enumeration $\textsf{\textit{enum-basis}}_\alpha$ of a topological basis of α. From this we construct an enumeration of a topological basis for $(\iota :: \textsf{\textit{countable}}) \Rightarrow_F \alpha$:

We can enumerate all finite maps of type $\iota \Rightarrow_F \mathbb{N}$ in a sequence $(f_n)_{n\in\mathbb{N}}$ according to lemma 5.11. If we map the codomain of f_n into elements of the basis of α, we basically enumerate all functions into all combinations of elements of the basis. We take the product set thereof and declare this as a candidate for the enumeration of a basis of $\iota \Rightarrow_F \alpha$:

Definition 5.12. ENUM-BASIS-FINMAP-DEF:

$$\textsf{\textit{enum-basis}}\ n := \prod_{i\in \textsf{\textit{dom}}\ f_n}^{F} \textsf{\textit{enum-basis}}_\alpha\ ((f_n)_F\ i), \qquad \textsf{\textit{enum-basis}} :: \mathbb{N} \Rightarrow (\iota \Rightarrow_F \alpha)\ \textsf{\textit{set}}$$

By definition, the range of $\textsf{\textit{enum-basis}}$ is the set of all products of all elements of a basis of α:

Lemma 5.13. RANGE-ENUM-BASIS-EQ:

$$\textsf{\textit{enum-basis}}[\mathbb{N}] = \left\{ \prod_{j\in J}^{F} A_j \;\middle|\; \textit{finite}\ J \wedge (\forall j \in J.\ A_j \in \textsf{\textit{enum-basis}}_\alpha[\mathbb{N}]) \right\}$$

In order to show that we actually enumerate a basis, we exploit the characterization of a topological basis given by lemma 4.6. We have that for arbitrary $x :: \iota \Rightarrow_F \alpha$ in an open set A, there is an open ball $\textsf{\textit{ball}}\ x\ e$ around x that is a subset of A. By the choice of our metric on finite maps, we have that a product of open balls $\prod_{i\in \textsf{\textit{dom}}\ x}^{F} \textsf{\textit{ball}}_\alpha\ ((x)_F\ i)\ e'$ with $e' = \frac{e}{|\textsf{\textit{dom}}\ x|}$ is contained in $\textsf{\textit{ball}}\ x\ e$. A use of lemma 4.6 for type α gives us elements B_i of the basis of α that are subsets of $\mathit{ball}_\alpha\ ((x)_F\ i)\ e'$. Consequently, the product $\prod^F_{i\in \mathit{dom}\ x} B_i$ is an open set with the desired properties, i.e., it is a subset of A and in the range of *enum-basis*. We can conclude that *enum-basis* enumerates a topological basis of type $\iota \Rightarrow_F \alpha$.

$$(\mathit{countable}(\iota) \wedge \mathit{enumerable\text{-}basis}(\alpha)) \longrightarrow \mathit{enumerable\text{-}basis}(\iota \Rightarrow_F \alpha)$$

Together with the results about completeness of finite maps and the definition of polish spaces, we can show that $\iota \Rightarrow_F \alpha$ is polish. As a consequence, we can now approximate measures on $\iota \Rightarrow_F \alpha$ according to regularity from theorem 4.18.

Theorem 5.14. ARITY-POLISH-SPACE-FINMAP:

$$(\mathit{countable}(\iota) \wedge \mathit{polish\text{-}space}(\alpha)) \longrightarrow \mathit{polish\text{-}space}(\iota \Rightarrow_F \alpha)$$

5.5 σ-Algebra of Products of Finite Maps

We introduced the type $\Rightarrow_F$ of finite maps and showed in the previous section that it is polish basically for one reason: We want to profit from theorem 4.18 which allows us to approximate Borel measures of polish measure spaces. Consequently, we provide a σ-algebra for finite maps similar to the one for function products. Then we show that this σ-algebra actually is the Borel σ-algebra of finite maps.

Similar to (function) product sets $\prod$ (definition 2.4), we introduced product sets $\prod^F$ also for finite maps (definition 5.7). Nevertheless, we define the product σ-algebra slightly different: The product $\bigotimes_{j\in J} \mathcal{A}_j$ contains only functions with the same domain J. For the finite maps, we allow several domains. The product σ-algebra of finite maps does therefore depend not only on a set $J :: \iota\ \mathit{set}$ but on a collection of sets $\mathcal{J} :: \iota\ \mathit{set}\ \mathit{set}$ that contains the allowed domains.

For measurable spaces $(\Omega_j :: \alpha\ \mathit{set}, \mathcal{A}_j :: \alpha\ \mathit{set}\ \mathit{set})$ for $j \in \bigcup_{J\in\mathcal{J}} J$, we define the product measurable space $(\Omega_{\mathcal{J}}, \bigotimes^F_{j\in J\in\mathcal{J}} \mathcal{A}_j)$ as follows:

Definition 5.15. PIF-DEF:

$$\Omega_{\mathcal{J}} := \bigcup_{J\in\mathcal{J}} \prod_{j\in J}^F \Omega_j$$

$$\bigotimes_{j\in J\in\mathcal{J}}^F \mathcal{A}_j := \sigma\left(\left\{ \prod_{j\in J}^F X_j \;\middle|\; J \in \mathcal{J} \wedge X \in \prod_{j\in J} \mathcal{A}_j \right\}\right)$$

Note that in the notation $j \in J \in \mathcal{J}$, J is something like an auxiliary binder for the product syntax. What we actually mean is $j \in \bigcup_{J\in\mathcal{J}} J$. If $\mathcal{J}$ is a singleton $\{J\}$, we just write $\bigotimes^F_{j\in J} \mathcal{A}_j$.

Having the ability to talk about a collection of domains J in $\mathcal{J}$ provides us with a good amount of flexibility: To measure functions from $\prod_{j\in J}\mathcal{A}_j$, we can choose the singleton $\mathcal{J} = \{J\}$. When we talk about Borel sets (which are defined on the whole type universe), we can choose $\mathcal{J} = \{J \mid \text{finite } J\}$ and actually include every possible finite map.

For countable domain, we have the following helpful lemma to reduce measurability on a set of domains to measurability for just one (arbitrary, but fixed) domain.

Lemma 5.16. FINITE-MEASURABLE-SINGLETONI:
For $\mathcal{J} :: (\iota :: \mathsf{countable})\ \mathsf{set\ set}$ *and* $\mathcal{C} :: \beta\ \mathsf{measure\text{-}space}$
Assume:

$$\forall J \in \mathcal{J}.\ \mathit{finite}\ \ J \longrightarrow f\ \ \mathit{is}\ \left(\bigotimes_{j\in J}^{F} \mathcal{A}_j\right)\text{-}\mathcal{C}\text{-}\mathit{measurable}$$

Then:

$$f\ \ \mathit{is}\ \left(\bigotimes_{j\in J\in\mathcal{J}}^{F} \mathcal{A}_j\right)\text{-}\mathcal{C}\text{-}\mathit{measurable}$$

We can show by adapting a similar proof for functions that the product σ-algebra $\bigotimes_{j\in J}^{F}\mathcal{A}_j$ of finite maps is generated by the product of the respective generators $\mathcal{G}_j$ with $\sigma(\mathcal{G}_j) = \mathcal{A}_j$. For this, we need to assume that there exists an increasing sequence of elements that covers the whole space. But here we can use lemma 4.12 since the whole space is an open set.

Regularity of measures is declared for Borel σ-algebras of a polish type. In fact, we can use the σ-algebra given by $\bigotimes^{F}$: If we include all (finite) domains, we can show that we have exactly the Borel σ-algebra of finite maps:

Theorem 5.17. BOREL-EQ-PIF-BOREL: *Assume* $\mathcal{J} = \{J \mid \mathit{finite}\ J\}$.

$$\mathcal{B} :: (\iota \Rightarrow_F \alpha)\ \mathsf{set\ set} = \left(\bigotimes_{j\in J\in\mathcal{J}}^{F} (\mathcal{B} :: \alpha\ \mathsf{set\ set})\right)$$

5.6 Measure Space Isomorphisms Involving Products of Finite Maps

The *Daniell-Kolmogorov* theorem specifies the existence of a measure on functions with arbitrary domain. We need to use $\Rightarrow_F$ in the proof because then we can exploit regularity of measures on polish spaces. But we can (by definition) represent only functions with finite domain with the type $\Rightarrow_F$, therefore the *Daniell-Kolmogorov* theorem needs to be stated in terms of arbitrary functions.

As a consequence, we need to somehow transfer our results between these types. What we are mostly interested in is measures of sets of the respective types. Therefore we provide an isomorphism between measure spaces on the σ-algebra generated by (function) product sets and the σ-algebra of (finite map) products.

A (point) isomorphism (see Rösler [23], Bogachev [7]) between measure spaces $(\Omega, \mathcal{A}, \mu)$ and $(\Omega', \mathcal{A}', \mu')$ is a bijective function from Ω to Ω' that is *measure preserving*. Measure preserving means that the push-forward measure of μ equals μ'; for a bijective function f:

$$\forall A' \in \mathcal{A}'.\ \mu'\ A' = \mu\ (f^{-1}[A']) = \textsf{push}\ f\ \mu\ A'$$

In this section, we provide two isomorphisms: First, we establish an isomorphism between a measure space $(\prod_{j\in J} \Omega_j, \bigotimes_{j\in J} \mathcal{A}_j, \mu)$ on functions and a measure space $(\prod^F_{j\in J} \mathcal{A}_j, \bigotimes^F_{j\in J} \mathcal{A}_j, \mu_F)$ on finite maps. Second, we show that the composition with a bijective function is an isomorphism. We will use both to establish an isomorphism between products of functions with a countable index set and products of finite maps with the natural numbers as index set. Note that we did not explicitly formalize the notion of isomorphisms in Isabelle/HOL, we only provide measure preserving bijections.

5.6.1 Functions and Finite Maps

The operator $(\cdot)_F$ and the constructor *finmap* provide means to transfer between functions and finite maps while preserving their "structure", i.e., function application and application $(\cdot)_F$ of finite maps yield the same results. We can show that the constructor *finmap* J is measurable.

Lemma 5.18. MEASURABLE-PIM-FINMAP-OF:

$$\textsf{finmap}\ J\ \textit{is}\ (\bigotimes_{j\in J} \mathcal{A}_j)\text{-}(\bigotimes_{j\in J}^{F} \mathcal{A}_j)\text{-}\textit{measurable}$$

For the inverse direction, we first show that the projection $x \mapsto (x)_F\ j$ of finite maps to a given single index j is $(\bigotimes^F_{j\in J} \mathcal{A}_j)$-$\mathcal{A}_j$-measurable. This serves as a foundation to show that $(\cdot)_F$ is a measurable function from finite maps to functions.

Lemma 5.19. MEASURABLE-PIM-PROJ:

$$(\cdot)_F\ \textit{is}\ (\bigotimes_{j\in J}^{F} \mathcal{A}_j)\text{-}(\bigotimes_{j\in J} \mathcal{A}_j)\text{-}\textit{measurable}$$

We also have that for finite J, $(\cdot)_F$ and *finmap* J provide a bijection between $\prod_{j\in J} A_j$ and $\prod^F_{j\in J} A_j$. For a given measure μ on function products, we have that the push-forward measure of μ under *finmap* J is a measure of an isomorphic measure space on finite maps (*finmap* J is a measure preserving bijection):

Lemma 5.20.

$$\Big(\prod_{j\in J} \mathcal{A}_j, \bigotimes_{j\in J} \mathcal{A}_j, \mu\Big) :: (\iota \Rightarrow \alpha)\ \textsf{\textit{measure-space}}$$

is isomorphic to

$$\Big(\prod_{j\in J}^{F} \mathcal{A}_j, \bigotimes_{j\in J}^{F} \mathcal{A}_j, \textsf{\textit{push}}\ (\textsf{\textit{finmap}}\ J)\ \mu\Big) :: (\iota \Rightarrow_F \alpha)\ \textsf{\textit{measure-space}}$$

5.6.2 Transferring the Domain of a Function

We encounter the issue that the domains of the functions we want to reason about range over a countable subset of an arbitrary type. To profit from the results about finite maps, we need the whole type universe to be countable. We therefore provide an isomorphism between measure spaces on products of functions where we transfer the domain of the functions.

For this, we assume an injective function $f :: \iota \Rightarrow \kappa$ on an index set J which is consequently a bijection between J and $f[J]$. The domain of a function is transferred by composing the function with f. If we take the restricted composition, i.e., the composition restricted to J, we have a measurable function to transfer the domain of function products:

Lemma 5.21. MEASURABLE-COMPOSE:

$$(m \mapsto m \circ_J f)\ \textit{is}\ (\bigotimes_{k\in f[J]} \mathcal{A}_k)\text{-}(\bigotimes_{j\in J} \mathcal{A}_j)\textit{-measurable}$$

In addition, the restricted composition with the inverse f^{-1} is also measurable. We consequently have an isomorphism between a measure space of products and a measure space for the transferred domain.

Lemma 5.22. *Assume an injective function $f :: \iota \Rightarrow \kappa$ on J. Then:*

$$\Big(\prod_{j\in J} \mathcal{A}_j, \bigotimes_{j\in J} \mathcal{A}_j, \mu\Big) :: (\iota \Rightarrow \alpha)\ \textsf{\textit{measure-space}}$$

is isomorphic to

$$\left(\prod_{k\in f[J]} \mathcal{A}_k, \bigotimes_{k\in f[J]} \mathcal{A}_k, \textsf{\textit{push}}\ (m \mapsto m \circ_J f)\ \mu\right) :: (\kappa \Rightarrow \alpha)\ \textsf{\textit{measure-space}}$$

5.7 Discussion

We built the type of finite maps as tuples of a domain and a function restricted to that domain. If you were to solely design a library of finite maps, you would probably start with the type *map* of Isabelle/HOL and restrict the elements of this type to finite domains. You could then profit from the developments about maps in general. We chose our formalization, because of the similar representation that is also used for products $\prod$ of functions. The results that we were interested in were mostly formalized for functions, not for maps. So basically, we were interested in functions, but we needed to add the restriction of the domain to the type to be able to obtain the topological properties of functions with finite domain.

Concerning the measure spaces on finite maps, the product construction explicitly carries the domain, so in this case there would have been no need to add the domain to the type. For the topological results, is seems like it is not really necessary that the domain is included in the type, so perhaps an extension of finite functions (formalized by Lochbihler [18]), i.e., functions with a finite domain, would have sufficed and would have been probably more appropriate and one would have saved some duplicate efforts.

However, we did not follow these considerations too much: The main reason why we had to introduce this new type is that almost all topological properties are formalized in terms of type classes, i.e., all assumptions have to hold on the whole type universe. It feels like a cleaner approach to relax all necessary topological definitions and results from types to sets because other applications might profit from that, too. This is a task we might tackle in the future.

6 Projective Limit

Our goal for this chapter is to provide a probability space to measure paths, i.e., functions from time into some state space. For the rest of the chapter we will write I to denote the time, i.e., the domain of the paths. We assume for the state space the Borel σ-algebra of a type α, namely $\mathcal{B} :: \alpha$ *set set*. The measurable space on paths will then be the product measurable space $\bigotimes_{i \in I} \mathcal{B}$ as introduced in definition 2.6.

Let us emphasize, that (for the rest of this chapter) we do not make any assumptions on the domain $I :: \iota$ or the type ι. However, we need to assume that the state space is polish (see section 4.5), this means for the rest of the chapter $\alpha ::$ *polish-space*.

The *Daniell-Kolmogorov* theorem which we approach in this chapter constructs a probability measure on the arbitrary product space depending on measures on finite-dimensional product spaces. For convenience, we abbreviate the measurable spaces with $(\Omega_J, \mathcal{B}_J)$ where J gives the domain of the functions or the dimension of the product space. In the following we are only interested in the product spaces with finite $J \subseteq I$, the only exception is the whole product space $(\Omega_I, \mathcal{B}_I)$.

Definition 6.1. PIP-DEF: *The measurable product space $(\Omega_J, \mathcal{B}_J)$ for domain J*

$$\Omega_J := \prod_{j \in J} \mathit{Univ}$$

$$\mathcal{B}_J := \bigotimes_{j \in J} \mathcal{B}$$

6.1 Projective Family

The *Daniell-Kolmogorov* theorem constructs a distribution P_I on the product space $(\Omega_I, \mathcal{B}_I)$ in terms of a family of finite dimensional distributions P_J on product spaces Ω_J, i.e., $J \subseteq I$ with J finite.

Let us give a short intuition to this: Assume P_I is given. Then for a finite J, P_J shall be the probability distribution of paths projected to the finite domain J. $P_J(\prod_{j \in J} B_j)$ is to be interpreted as the probability for a random path with domain J to pass at every index $j \in J$ through the set B_j (see for example figure 6.1).

To relate the different product spaces to each other, the product over J is embedded into the product space over I with *emb* I J (definition 2.5). A necessary

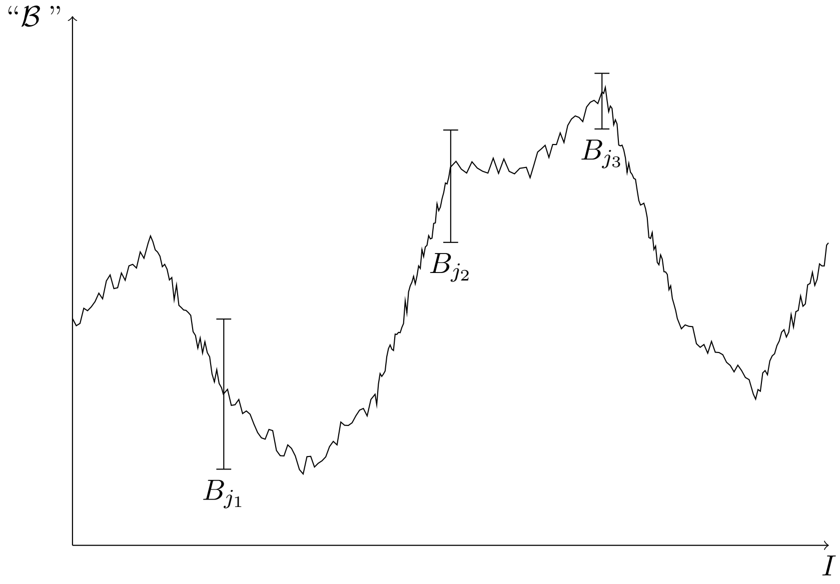

Figure 6.1: One possible path through B_j for $j \in \{j_1, j_2, j_3\}$

condition for the interpretation of P_J as the probability of the projection to be correct is $P_J(\prod_{j \in J} B_j) = P_I(\textsf{emb}\ I\ J\ (\prod_{j \in J} B_j))$.

The *Daniell-Kolmogorov* theorem now takes the opposite direction: It assumes a family P_J of finite-dimensional distributions and constructs a probability measure P_I on the product space $(\Omega_I, \mathcal{B}_I)$ that is consistent with the finite-dimensional distributions. Consistent means in this case that if we embed a product set B (over J) into a space with a larger index set H, then the measures should be the same in both product spaces. Consistency of the finite-dimensional family of distributions is captured in the notion of a *projective family*:

Definition 6.2. PROJECTIVE-FAMILY-DEF: $P :: \iota\ \textsf{set} \Rightarrow (\iota \Rightarrow \alpha)\ \textsf{measure-space}$ *is called a* projective family *over* $\mathcal{B}_I$*, iff the following holds:*

- *P_J is a probability measure on $(\Omega_J, \mathcal{B}_J)$ ($\forall J \subseteq I.$ finite J)*
- $\forall H\ J\ B.\ (J \subseteq H \wedge \textit{finite}\ H \wedge H \subseteq I) \longrightarrow P_H\ (\textsf{emb}\ H\ J\ B) = P_J\ B$

6.2 Generator and Content

The *Daniell-Kolmogorov* theorem states that the probability distribution on $\mathcal{B}_I$ is completely determined by a consistent (i.e., projective) family of finite-dimensional distributions on $\mathcal{B}_J$. We therefore assume from now on a *projective family* $P :: \iota\ \mathsf{set} \Rightarrow (\iota \Rightarrow \alpha)\ \mathsf{measure\text{-}space}$.

In order to construct the probability distribution P_I for the measurable space $(\Omega_I, \mathcal{B}_I)$, we define a generator $\mathcal{Z}$ of $\mathcal{B}_I$ and a content, i.e., an additive and non-negative function P_0 on $\mathcal{Z}$, and extend this content to a (probability) measure P_I on the σ-closure of $\mathcal{Z}$, i.e., on $\mathcal{B}_I$.

The generator is the union of all σ-algebras of finite embeddings:

Definition 6.3. PRODUCT-PROB-SPACE.GENERATOR-DEF:

$$\mathcal{Z} := \left(\bigcup_{J \neq \emptyset \wedge \mathit{finite}\ J \wedge J \subseteq I} (\mathsf{emb}\ I\ J)[\mathcal{B}_J] \right)$$

Lemma 6.4. SETS-PIM-GENERATOR:

$$\sigma(\mathcal{Z}) = \mathcal{B}_I$$

P_0 needs to be defined on sets $Z \in \mathcal{Z}$, i.e., embeddings of the form $Z = \mathsf{emb}\ I\ J\ B$ with $B \in \mathcal{B}_J$. We want to associate to an embedding of a product over J its probability given by the finite-dimensional distribution.

$$P_0\ (\mathsf{emb}\ I\ J\ B) = P_J\ B$$

However, for a given $Z \in \mathcal{Z}$, there may be several representations, i.e., $Z = \mathsf{emb}\ I\ J\ B = \mathsf{emb}\ I\ K\ C$. Since P is projective, one can show $P_J\ B = P_K\ C$ for finite $J, K \subseteq I$ (in the same way as it was done by Hölzl [10]). This means that P_0 is well-defined and can be defined using Hilbert choice.

Definition 6.5. μG'-DEF:

$$P_0\ Z := (\varepsilon x.\ \forall J \subseteq I.\ \mathit{finite}\ J \longrightarrow (\forall B \in \mathcal{B}_I.\ Z = \mathsf{emb}\ I\ J\ B \longrightarrow x = P_J\ B))$$

P_0 is specified for every element of $\mathcal{Z}$. These elements are finite embeddings in I. We therefore have the desired projectivity of P_0 on $\mathcal{Z}$.

Lemma 6.6. GENERATORE': *Assuming $Z \in \mathcal{Z}$, there exists a finite $J \subseteq I$ and $B \in \mathcal{B}_J$ such that*

- $Z = \mathsf{emb}\ I\ J\ B$
- $P_0\ Z = P_J\ B$

It is easy to see that P_0 is non-negative and additive, hence a *content* on the generator $\mathcal{Z}$.

Lemma 6.7. POSITIVE-μG': *P_0 is non-negative on $\mathcal{Z}$.*

Lemma 6.8. ADDITIVE-μG': *P_0 is additive on $\mathcal{Z}$.*

One observation that is actually not surprising is the considerable amount of redundancy with respect to the formalization of products of probability spaces by Hölzl [10]. We profited from the developments about products and basically copied proofs about the pre-measure operating on the generator of the product σ-algebra. This is because the proof for products of measure spaces is based on a particular projective family, without introducing the notion of projectivity and thus working only in this particular setting. It would be worth generalizing the two developments.

6.3 Extension of Content to Measure

In the previous section, we showed that P_0 is a content on $\mathcal{Z}$. Moreover Hölzl [10] already formalized that $\mathcal{Z}$ is a ring of sets. The standard way in measure theory to prove the existence of a (probability) measure P_I that extends P_0 on the σ-closure of $\mathcal{Z}$ is described in section 2.11.3: One needs to show that P_0 is continuous at the empty set $\emptyset$. That is, one assumes a decreasing sequence $(Z_n)_{n\in\mathbb{N}}$ with $Z_n \in \mathcal{Z}$ that converges to $\emptyset$ and shows that the content $P_0\ Z_n$ converges to 0. We are going to show the contrapositive proposition: We assume that $P_0\ Z_n$ converges to $a > 0$ and construct an element $z \in \bigcap_n Z_n$, i.e., we show that Z_n does not converge to $\emptyset$. Note that since the sequence is decreasing, we may assume $P_0\ Z_n \geq a$ for all n.

The structure of the proof can be split into two parts: First, a sequence of functions $y_n :: \iota \Rightarrow \alpha$ is constructed in a way such that infinitely many elements of the sequence lie in compact subsets of Z_n. We show the existence of these elements by proving that these compact subsets have a (probability) measure greater than zero. Second, for every index $t \in I$, compactness guarantees that there exists a subsequence $(y'_n)_{n\in\mathbb{N}}$ of $(y_n)_{n\in\mathbb{N}}$ that converges for this index t, i.e., $(y_n\ t)_{n\in\mathbb{N}}$ converges. One then constructs a diagonal sequence that converges for every index (note that for every n, y_n stems from an embedding of a finite product such that there is only a countable number of indices t involved).

We can define the extension of the content P_0 to a measure P_I on $(\Omega_I, \mathcal{B}_I)$ if we take lemma 6.4 into account:

Definition 6.9. PIP-DEF:

$$P_I := \textsf{extend-measure}\ \mathcal{Z}\ P_0$$

6.3.1 Probabilistic Argument

In the following, you will find a description of how the proof is done in a more detailed fashion, concentrating on the technical difficulties of the formalization.

From Embeddings to Finite Products

For every n the element Z_n of the decreasing sequence $Z_n \in \mathcal{Z}$ is an embedding of some finite product $B_n \in \mathcal{B}_{J_n}$ for some domain J_n.

$$Z_n = \textsf{\textit{emb}}\ I\ J_n\ B_n$$

Note that one can assume (by embedding into the union of the index sets) that the sequence J_n of the index sets is increasing:

$$n \leq m \longrightarrow J_n \subseteq J_m$$

Establish Isomorphism to a Polish Type

The key property that is used is the fact that B_n lies in the space $\mathcal{B}_{J_n}$, which is (from a mathematical point of view) polish. We have formalized polish spaces only as a type class, that means we need to move to a suitable type, namely the finite maps $\Rightarrow_F$. Recall that $\Rightarrow_F$ is only polish if the type of the index set is countable (recall theorem 5.17). We therefore provide a suitable bijection $\textsf{\textit{fm}}$ between functions $\iota \Rightarrow \alpha$ and finite maps $\mathbb{N} \Rightarrow_F \alpha$:

Definition 6.10. FUNCTION-TO-FINMAP.FM-DEF:

$$\textsf{\textit{fm}}_n\ m := (\textsf{\textit{finmap}}\ J_n) \circ (m \circ_{J_n} \textsf{\textit{to-nat}}_J)$$

We established that $(\textsf{\textit{finmap}}\ J_n)$ provides an isomorphism in lemma 5.20. Moreover $\textsf{\textit{to-nat}}_J$ is injective on J_n according to lemma 3.1, therefore $(m \mapsto m \circ_{J_n} \textsf{\textit{to-nat}}_J)$ provides an isomorphism, too. The composition gives us an isomorphism to a polish measure space: For better readability, let us abbreviate the family of transferred domains as $N_n := \textsf{\textit{to-nat}}_J(J_n)$ and denote with $(\Omega^F_{N_n}, \mathcal{B}^F_{N_n})$ the transferred measurable spaces $(\prod^F_{i \in N_n} \textsf{\textit{Univ}}, \bigotimes^F_{i \in N_n} \mathcal{B})$, and write $P^F_{N_n}$ for the push-forward measure of P_{J_n} under $\textsf{\textit{fm}}_n$.

Lemma 6.11. FUNCTION-TO-FINMAP.MAPMEASURE-PIM:

$$(\Omega_{J_n}, \mathcal{B}_{J_n}, P_{J_n}) :: (\iota \Rightarrow \alpha)\ \textsf{\textit{measure-space}}$$

is isomorphic to

$$(\Omega^F_{N_n}, \mathcal{B}^F_{N_n}, P^F_{N_n}) :: (\mathbb{N} \Rightarrow_F \alpha)\ \textsf{\textit{measure-space}} :: \textsf{\textit{polish-space}}$$

$$\begin{tikzcd}
\mathcal{Z}, P_0 \arrow[r, dash, "\text{projectivity}"] & (\Omega_{J_n}, \mathcal{B}_{J_n}, P_{J_n}) \arrow[r, dash, "\cong"] & (\Omega^F_{N_n}, \mathcal{B}^F_{N_n}, P^F_{N_n}) \\
Z_n \arrow[r, "(\mathsf{emb}\ I\ J_n)^{-1}"] & B_n \arrow[r, "\mathsf{fm}_n"] & B^F_n \arrow[d, dash, "\approx"'] \\
Z'_n & K_n \arrow[l, "\mathsf{emb}\ I\ J_n"'] & K^F_n \arrow[l, "\mathsf{fm}_n^{-1}"']
\end{tikzcd}$$

Figure 6.2: The objects occurring in the probabilistic argument

Exploit Inner Regularity

Let us denote the transferred version of B_n with $B^F_n := \mathsf{fm}_n[B_n]$. Because the probability space $(\Omega^F_{N_n}, \mathcal{B}^F_{N_n}, P^F_{N_n}) :: (\mathbb{N} \Rightarrow_F \alpha)$ *measure-space* is polish, inner regularity (theorem 4.18) allows us to approximate the measure $P^F_{N_n}\ B^F_n$ with the measure of some compact subset $K^F_n \subseteq B^F_n$:

$$P^F_{N_n}\ K^F_n \approx P^F_{N_n}\ B^F_n$$

Over Products Back to Embeddings

We can take the isomorphism in the inverse direction and obtain from K^F_n a finite product $K_n := \mathsf{fm}_n^{-1}[K^F_n]$ such that $P_{J_n}\ B_n$ approximates $P_{J_n}\ K_n$ – here we exploit measure preservation of the isomorphism. Moving back to the content P_0 on embeddings $\mathcal{Z}$, it follows from the projectivity of the family P_J that the content P_0 of the embedding Z'_n of K_n approximates Z_n.

$$\begin{aligned}
Z'_n :=& \mathsf{emb}\ I\ J_n\ K_n \\
P_{J_n}\ B_n \approx& P_{J_n}\ K_n \\
P_0\ Z_n \approx& P_0\ Z'_n
\end{aligned}$$

Consult figure 6.2 for a summary of the different objects that occurred in the proof up to now: We have Z_n and Z'_n in the generator $\mathcal{Z}$ of the arbitrary dimensional measurable space $(\Omega_I, \mathcal{B}_I)$. The assumption of a projective family and the way we defined the content P_0 guarantees that we preserve the measure of the finite products B_n and K_n in the measure space $(\Omega_{J_n}, \mathcal{B}_{J_n}, P_{J_n})$. The isomorphism between this measure space and $(\Omega^F_{N_n}, \mathcal{B}^F_{N_n}, P^F_{N_n})$ given by fm_n allows us to identify corresponding products of finite maps where the measure of B^F_n approximates the measure of K^F_n due to regularity of measures of finite maps.

The Probabilistic Argument

The approximations we mentioned before are chosen with a precision that suffices to show that the content P_0 of the difference between Z_n and the intersection of all Z'_k with $k \le n$ is smaller than a. Together with the initial assumption $P_0\ Z_n \ge a$ we have

$$P_0\ (\bigcap_{k \le n} Z'_k) > 0$$

from which we can conclude that for every n, we can choose an element y_n:

$$y_n \in \bigcap_{k \le n} Z'_k$$

For a fixed n and $m \ge n$, the definitions ensure that the restriction of y_m to the domain J_n (recall that J_n is increasing) lies in K_n.

$$\forall m \ge n.\ \ y_m|_{J_n} \in K_n$$

6.3.2 Construction of Diagonal Sequence

Consider now an arbitrary index n and the (transferred) sequence of finite maps $(\mathit{fm}_n\ y_i)_{i \in \mathbb{N}}$. For $i \ge n$, this sequence lies in the compact set K^F_n according to the previous section and the isomorphism fm. The projection of this sequence to the index n is contained in the projection of $\mathit{fm}_n(K_n)$ to n. Projection on finite maps is continuous according to lemma 5.6, therefore the projection is compact, too. The definition of compactness yields a convergent subsequence of projections. This argument also applies to arbitrary subsequences y' of y:

Lemma 6.12. FINMAP-SEQS-INTO-COMPACT.DIAGONAL-TENDSTO: *Existence of subsequence converging for index n:*

$$\forall y' \preccurlyeq y.\ \exists y'' \preccurlyeq y'.\ ((\mathit{fm}_n\ y''_i)_F\ n)_{i \in \mathbb{N}} \textit{ converges}$$

From this, we can construct a diagonal sequence as defined in section 3.2: Starting with a subsequence y' of y that converges for $n = 1$, we take a subsequence of y'' that converges for $n = 2$ and so on. The diagonal sequence $((\mathit{fm}_n\ y^{(i)}_i)_F\ n)_{i \in \mathbb{N}}$ therefore converges for every index n. Since for every $j \in \bigcup_i J_i$ there is $n = \textit{to-nat}_J\ j$, the subsequence $(y^{(i)})_{i \in \mathbb{N}}$ converges for every index $j \in \bigcup_i J_i$. We define the resulting limit point of $(y^{(i)}\ j)_{i \in \mathbb{N}}$ (depending on the index j) z_j. Moreover we call z the function $(j \mapsto z_j)$.

Now consider an arbitrary n: The restriction $z|_{J_n}$ of z to J_n is the limit point z_* of a sequence of restrictions with elements in K_n. Its corresponding set K_n^F is compact and therefore closed. It follows that the corresponding z_*^F lies in K_n^F. Therefore $z|_{J_n}$ is contained in K_n. One can conclude that the embedding $\mathsf{emb}\ I\ J_n\ \{z|_{J_n}\}$ lies in Z'_n. Z'_n approximates Z_n, therefore $z|_{J_n} \in Z'_n \subseteq Z_n$ which means that Z_n can not be empty.

Since n was chosen arbitrarily, $\bigcap_i Z_i$ can not be empty, i.e., does not converge to the empty set $\emptyset$, which concludes the proof.

6.4 Summary: Formalization of the *Daniell-Kolmogorov* Theorem

Let us summarize the contents of this chapter in order to give a concise overview of what has actually been formalized:

- We started out by using the existing formalizations about products and embeddings to declare the product space Ω_I and its associated product σ-algebra $\mathcal{B}_I$.
- We defined the notion of a *projective family*, the family of finite-dimensional distributions of a stochastic process.
- One generator of the product σ-algebra is the set of embeddings of finite-dimensional products. We assume a projective family to define a content on this generator.
- We prove the existence of an extension of the content on this generator to the generated σ-algebra $\mathcal{B}_I$ with standard means from measure theory: We show that the content is continuous at the empty set. This argument is then rather involved:
  - We construct a sequence of approximations: For this we need results about polish spaces. In order to profit from them, we have to switch to the type $\Rightarrow_F$ of finite maps
  - In order to show that one can choose elements out of these approximations, we use a probabilistic argument. We therefore switch back to the type of functions.
  - We transfer the obtained sequence of functions to a sequence of finite maps. Here we can use topological arguments to construct a convergent diagonal sequence.

- We switch back to functions to complete the argument.

The *Daniell-Kolmogorov* theorem can then be stated as follows:

Theorem 6.13. POLISH-PROJECTIVE.MEASURE-PIB-EMB:
Assume

- *an arbitrary type* ι
- *an index set* $I :: \iota$
- *a polish type* α
- *the Borel* σ*-algebra* $\mathcal{B} :: \alpha$ **set set**
- *a projective family* P *over* $\mathcal{B}$

Then there exists a measure P_I *that assigns to the embedding of every product of Borel sets over finite* J *the value of the according finite-dimensional measure* P_J*:*

$$\forall J \subseteq I.\ (\mathit{finite}\ J \wedge X \in \mathcal{B}_J) \longrightarrow P_I\ (\mathsf{emb}\ I\ J\ X) = P_J\ X$$

Moreover P_I *is a probability measure.*

6.5 Discussion

To summarize the experiences of the formalization of the *Daniell-Kolmogorov* theorem, the main complication is that we have to switch between different types: On the one hand, we want to construct a probability space on arbitrary-dimensional products, we therefore need to use the function type. On the other hand, the generator of the probability space involves only finite-dimensional products, because they have nice properties which we exploit in the proof. The topological properties we use are only defined for type classes, i.e., they have to hold on the universe of a type, which forces us to introduce a new type of finite maps. This forces us to provide suitable isomorphisms between the different types. Transferring between the different types and providing suitable isomorphisms turned out to be quite tedious.

7 Conclusion

The existing formalization of probability theory in Isabelle/HOL allows to construct probability spaces for paths of particular stochastic processes with discrete time and state space (discrete-time Markov chains). This work provides a generic construction, i.e., a construction that also works for continuous time and state space. We formalized the *Daniell Kolmogorov* theorem, it allows to construct probability spaces for arbitrary paths from probability spaces for paths with finite domain. In order to formalize this theorem, we provided topological foundations: We formalized polish spaces as type classes and we formalized regularity of measures on polish spaces. The fact that topological notions are formalized for type classes produced technical complications in our proofs: We needed to define a separate type of functions with finite domain and provide means to transfer between this type and the ordinary type of functions.

7.1 Formalization Efforts

One can estimate the efforts for the formalization by the number of lines in our sources: The auxiliary developments take about 450 lines. Results about topology and polish spaces (including regularity of measures) take roughly 1000 lines. The developments about finite maps take 1500 lines. About half of this, however, is redundant with respect to the existing developments about functions (especially product sets and the product σ-algebra). The construction of the projective limit consists of 800 lines of formalized proof.

Let us compare our proofs with the textbook proofs we took as inspiration. Our proofs about topology are less general than the ones presented by von Querenburg [24]. His proofs are based on uniform spaces, but there is no formalization of uniform spaces in Isabelle/HOL. We therefore formalize our theorems in the more specific setting of metric spaces.

The textbook proofs for regularity of measures [5] and the existence of the projective limit [4], both written by Bauer at a similar level of detail take about three pages each in the textbooks. Also the sizes of the formalized proofs are about equal (regularity 300 lines, projective limit 400 lines). Note that it is hard to be accurate about these numbers, because one can always exclude parts of the reasoning in separate lemmas. We believe, however, that the main parts of our formalizations expose roughly the same amount of detail.

Despite the similar size of the core proofs, formalizing the projective limit turned out to be much more tedious: Several topological foundations were missing and we needed to introduce the separate type of finite maps – because most of topology in Isabelle/HOL is formalized on type classes – and transfer between measures on the types of functions and finite maps. We identified measure space isomorphisms as a suitable concept to structure these technical parts of the proof. We also needed to provide a formal notion of diagonal sequences instead of the intuitive description given in Bauer's proof.

The formalization of regularity was much more straightforward. No technical difficulties arose, we could therefore basically translate the textbook proof into Isabelle/HOL. It was surely an advantage that we took the proof about regularity from the same textbook that was used to inspire the formalization of measure theory in Isabelle/HOL.

7.2 Related Work

Our work builds on the probability theory (based on measure theory) formalized by Hölzl and Heller [10, 11]. Their work is based on the library for multivariate analysis which was ported to Isabelle/HOL from Harrison's Euclidean spaces [9].

The first formalization of probability theory in HOL was given by Hurd [14], who formalizes (in hol98) a probability space for a random bit generator, i.e., an infinite sequence of random bits. Based on this probability space, he analyzes some probabilistic algorithms.

Lester [17] formalizes topology and on top of that measure and probability theory in PVS. Daumas and Lester [8] analyze accumulations of rounding errors with the help of Doob's inequality for martingales with finite index set. This is related to our work because every martingale is a stochastic process, moreover martingales with finite index set can be used to construct martingales with continuous index set – and therefore particular stochastic processes with continuous time (see e.g., Bauer [4]).

To the best of our knowledge, there are no other formalizations of stochastic processes or formalizations of the *Daniell-Kolmogorov* theorem.

A key step in the proof of the *Daniell-Kolmogorov* theorem is a probabilistic argument, i.e., the existence of elements with a given property is established by giving a positive probability to the set of elements satisfying this property. Noschinski [21] formalizes a proof that is based on a probabilistic argument.

7.3 Future Work

Future work on this formalization might include generalizing the existing developments to reduce redundancy, formalize important methods to construct stochastic processes based on the *Daniell-Kolmogorov* theorem, and actually apply these concepts by formally analyzing stochastic processes.

7.3.1 Generalizations

There is considerable amount of redundancy because proofs of the formalization of products of probability spaces by Hölzl [10] exploit the (projective) properties of a particular projective family. It would be worth generalizing everything that relies exclusively on projectivity in Hölzl's developments and use the more general proofs we gave in sections 6.1 and 6.2.

The introduction of a separate type of finite maps also introduced redundancy, because many constants operating on and facts about functions needed to be duplicated for finite maps (especially for product σ-algebras). We already discussed in section 5.7 how a relaxation of topological notions from types to sets would render the separate type superfluous.

7.3.2 Construction of Stochastic Processes

The *Daniell-Kolmogorov* theorem we formalized allows to construct a probability space for stochastic processes from a projective family. But usually one does not directly give a projective family to construct a stochastic process. Discrete-time Markov chains – particular stochastic process with discrete time and state space, see section 2.11.6 – can be described by a stochastic matrix which gives probabilities for the transitions between states. In order to describe arbitrary stochastic processes, one can generalize these stochastic matrices to so-called Markov kernels. A Markov kernel P yields for every point $x \in \Omega$ (of a measurable space $(\Omega, \mathcal{A})$) a probability space $(\Omega', \mathcal{A}', P\ x)$. $P\ x$ gives the probability distribution when the Markov chain is in state x.

For continuous time, Markov kernels can be indexed with time, such that $P_t\ x\ A'$ gives the probability that a process in state x finds itself in a state in A' after t units of time. One can provide a method to combine Markov kernels, i.e., to give the probability to reach a state after $t + s$ steps by combining the kernels P_t and P_s. One can show that semi-groups of kernels, i.e., families of kernels where P_{t+s} is the same as the combination of P_t and P_s, define a projective family. The projective family can then be used to construct a stochastic process. The formalization of Markov kernels and operations on them would be useful to construct a variety of stochastic processes in Isabelle/HOL.

A restriction of the *Daniell-Kolmogorov* theorem is the fact that the constructed probability spaces consist of all paths from time into the state space. But one might be interested only in particular, e.g., continuous paths. Results from probability theory that show the existence of continuous modifications, i.e., stochastic processes with continuous paths and the same probability distribution, could be formalized to overcome this restriction.

7.3.3 Applications of Stochastic Processes

With stochastic processes formalized in Isabelle/HOL, one could formally analyze any application of stochastic processes. A standard example in probability theory is the so-called Wiener process, which describes Brownian motion, the random movement of a particle (also called random walk). This would require continuous modifications. Discrete-time and continuous-time Markov chains can be applied to queuing theory to model for example population sizes in birth-death processes or queues in telecommunication or computer systems [15]. Also financial markets can be modeled as stochastic processes.

Moreover, stochastic processes have important applications in computer science, in the verification of probabilistic systems: After formalizing probability spaces for continuous-time Markov chains, one could – similar to the formalization of pCTL model checking by Hölzl [13] – formalize model checking over continuous stochastic logic (CSL) formulas which is described for example by Kwiatkowska [16] or Baier [2].

Another interesting topic in computer science is the analysis of continuous-time Markov decision processes as given by Puterman [22]: Markov decision processes (see also Baier [3] for discrete-time Markov decision processes) include both nondeterministic actions and probabilistic behavior and can therefore be used to model for example concurrent probabilistic systems. Actions are taken at real-valued times, therefore a history of such a Markov decision process is a sequence of triples consisting of the time, the state at that time, and the decision taken at that time. Schedulers are used to eliminate nondeterminism, they are based on the history and therefore possess (for continuous time) an uncountable state space, which makes the construction of a probability space using the *Daniell-Kolmogorov* theorem necessary.

Appendix

8 Source Code

Attached is a CD with the complete source code. You can also browse and download the sources online[1].

[1] `http://home.in.tum.de/~immler/mastersthesis/`

Bibliography

[1] Robert B. Ash. *Probability and measure theory.* Harcourt Academic Press, 2000.

[2] Christel Baier. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29:524–541, 2003.

[3] Christel Baier and Joost-Pieter Katoen. *Principles of model checking.* MIT Press, 2008.

[4] Heinz Bauer. *Probability theory.* de Gruyter, 1996.

[5] Heinz Bauer. *Measure and integration theory.* de Gruyter, Berlin, 2001.

[6] Patrick Billingsley. *Probability and measure.* Wiley, 1985.

[7] Vladimir I. Bogachev. *Measure theory volume II.* Springer, 2007.

[8] Marc Daumas and David Lester. Stochastic formal methods: An application to accuracy of numeric software. In *Hawaii International Conference on Systems Science (HICSS 2007)*, pages 262–269. IEEE Computer Society, 2007.

[9] John Harrison. A HOL theory of Euclidean space. In Joe Hurd and Tom Melham, editors, *Theorem Proving in Higher Order Logics*, volume 3603 of *Lecture Notes in Computer Science*, pages 114–129. Springer, 2005.

[10] Johannes Hölzl. *Construction and stochastic applications of measure spaces in higher-order logic.* Ph.D. thesis (submitted), Technische Universität München, 2012.

[11] Johannes Hölzl and Armin Heller. Three chapters of measure theory in Isabelle/HOL. In Marko C J D van Eekelen, Herman Geuvers, Julien Schmaltz, and Freek Wiedijk, editors, *Interactive Theorem Proving (ITP 2011)*, volume 6898 of *LNCS*, pages 135–151, 2011.

[12] Johannes Hölzl and Tobias Nipkow. Interactive verification of Markov chains: Two distributed protocol case studies. In *Quantities in Formal Methods*, volume 1480, 2012.

[13] Johannes Hölzl and Tobias Nipkow. Verifying pCTL model checking. In C Flanagan and B König, editors, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2012)*, volume 7214 of *LNCS*, pages 347–361, 2012.

[14] Joe Hurd. *Formal verification of probabilistic algorithms.* Ph.D. thesis, University of Cambridge, 2002.

[15] Vidyadhar Kulkarni. *Modeling and analysis of stochastic systems.* Chapman and Hall, 1995.

[16] Marta Kwiatkowska, Gethin Norman, and David Parker. Stochastic model checking. In Marco Bernardo and Jane Hillston, editors, *Formal Methods for Performance Evaluation*, volume 4486 of *LNCS*, pages 220–270. Springer, 2007.

[17] David R Lester. Topology in PVS: Continuous mathematics with applications. In Lee Pike, Andrew Ireland, Paul Jackson, Bill James Ellis, and Kathleen Sharp, editors, *AFM ' 07 : Second Workshop on Automated Formal Methods*, pages 11–20, 2007.

[18] Andreas Lochbihler. Code generation for functions as data. *Archive of Formal Proofs*, 2009.

[19] Tobias Nipkow. Interactive proof: Introduction to Isabelle/HOL. In O. Grumberg, T. Nipkow, and B. Hauptmann, editors, *Software Safety and Security*, pages 254–285. MOD2011, IOS Press, 2012.

[20] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL: A proof assistant for higher-order logic.* Number 2283 in LNCS. Springer, 2002.

[21] Lars Noschinski. Proof Pearl: A probabilistic proof for the girth-chromatic number theorem. In Lennart Beringer and Amy Felty, editors, *Interactive Theorem Proving*, volume 7406 of *LNCS*, pages 393–404. Springer, 2012.

[22] Marrin L. Puterman. *Markov decision processes.* Wiley, 1994.

[23] Uwe Rösler. Maßtheorie. `http://www.math.uni-kiel.de/stochastik/roesler/vorlesung/mass/Mass.pdf`, 2011.

[24] Boto von Querenburg. *Mengentheoretische Topologie.* Springer Berlin Heidelberg, 1976.

[25] Markus Wenzel. *Isabelle / Isar — a versatile environment for human-readable formal proof documents.* Ph.D. thesis, Technische Universität München, 2002.