A Formalization of the Ionescu-Tulcea Theorem in mathlib

Author : Etienne Marion

ENS de Lyon, 46, Allée d'Italie, 69007 Lyon, France
etienne.marion@ens-lyon.fr

Abstract

1 Introduction

Being able to talk about the joint distribution of an infinite family of random variables is crucial in probability theory. For example, one often requires a family of independent random variables. The existence of such a family relies on the existence of an infinite product measure. Indeed, given $(\Omega_i, \mathcal{F}_i, \mu_i)_{i\in\iota}$ a family of probability spaces, the existence of the product measure $\bigotimes_{i\in\iota}\mu_i$ yields a new probability space $(\prod_{i\in\iota}\Omega_i, \bigotimes_{i\in\iota}\mathcal{F}_i, \bigotimes_{i\in\iota}\mu_i)$, and the projections $X_i : \prod_{j\in\iota}\Omega_j \to \Omega_i$ give the desired family. For another example, consider discrete-time Markov chains. Given a measurable space $(E, \mathcal{A})$ and a Markov kernel $\kappa : E \to E$, one might want to build a sequence $(X_n)_{n\in\mathbb{N}}$ of random variables with values in E such that the conditional distribution of X_{n+1} given $X_0, ..., X_n$ is $\kappa(X_n, \cdot)$. Such objects are fundamental in probability theory: families of independent variables allow to build more complicated objects, such as Brownian motion, while discrete-time Markov chains form a huge class of stochastic processes which contains random walks for instance. It so happens that those objects always exist without any restrictions on the spaces we consider. This is a direct consequence of the Ionescu-Tulcea theorem.

The goal of this contribution is to provide a formalization of the proof of the aforementioned theorem using the proof assistant Lean and the associated library `mathlib`. It heavily relies on previous work done in [cite] which formalizes the Kolmogorov extension theorem. While this specific theorem is not used directly in the formalization presented here, we use the notion of measurable cylinders, projective families or additive contents which were introduced to prove the Kolmogorov extension theorem.

2 Statement of the theorem

The Ionescu-Tulcea theorem is about the existence of a certain Markov kernel. We therefore start with a reminder about those.

Definition 2.1. Let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be two measurable spaces. A *Markov kernel* from X to Y is a map $\kappa : X \times \mathcal{B} \to [0, 1]$ such that:

- for any $x \in X$, $\kappa(x, \cdot)$ is a probability measure;
- for any $B \in \mathcal{B}$, $\kappa(\cdot, B)$ is measurable.

One can therefore consider a Markov kernel as a measurable map which to any point in X associates a probability distribution over the space Y.

To ease the writing, let us give some notations.

Notation. If I is a set of indices and $(X_i, \mathcal{A}_i)_{i\in I}$ a family of measurable spaces, then:

- we write X for $\prod_{i\in I} X_i$ and $\mathcal{A}$ for $\bigotimes_{i\in I}\mathcal{A}_i$;
- if $J \subseteq I$, we write X^J for $\prod_{i\in J} X_i$ and $\mathcal{A}^J$ for $\bigotimes_{i\in J}\mathcal{A}_i$;

- if $I = \mathbb{N}$ and $n \in \mathbb{N}$ we write $X^{\geq n}$ for $X^{\{k \in \mathbb{N} | k \geq n\}}$ and $\mathcal{A}^{\geq n}$ for $\mathcal{A}^{\{k \in \mathbb{N} | k \geq n\}}$, with similar notations for $\leq, >, <$;
- if $J \subseteq I$, we write π_J for the canonical map from X to X^J;
- if $K \subseteq J \subseteq I$, we write $\pi_{J \to K}$ for the canonical map from X^J to X^K.

Let us now state the theorem.

Theorem 2.2 (Ionescu-Tulcea). *Let $(X_n, \mathcal{A}_n)_{n \in \mathbb{N}}$ be a family of measurable spaces. Let $(\kappa_n)_{n \in \mathbb{N}}$ be a family of Markov kernels such that for any n, κ_n is a kernel from $X_0 \times ... \times X_n$ to X_{n+1}. Then there exists a unique Markov kernel $\eta : X_0 \to \prod_{n \geq 1} X_n$ such that for any $n \geq 1$,*

$$\pi_{n*}\eta = \kappa_0 \otimes ... \otimes \kappa_{n-1}.$$

By taking each kernel to be constant we easily get the product of a countable family of probability measures, which can then be used to build the product of an arbitrary family of measures. Likewise if we take $X_n = X$ for all n and $\kappa_n = \kappa$ a kernel from $X \to X$ we get a Markov kernel from X to $X^{\mathbb{N}_+}$ which, if we compose it with a measure μ at the beginning, gives the measure associated to a Markov chain with initial distribution μ and transition kernel κ.

We will now give a sketch of the proof of the theorem to highlight the elements which are required for the formalization. The idea is to use the Carathéodory extension theorem which allows to extend a measure once we have defined it on a restricted family of measurable sets for which it is to give the expression of the measure. In what follows we consider $(X_n, \mathcal{A}_n)_{n \in \mathbb{N}}$ a family of measurable spaces and we set $\pi_n := \pi_{[\![1,n]\!]}$. we also consider of Markov kernels $(\kappa_n : X^{\leq n} \to_* X_{n+1})_{n \in \mathbb{N}}$ and we set $\eta_{k,n} := \kappa_k \otimes ... \otimes \kappa_{n-1} : X^{\leq k} \to_* X^{[\![k+1,n]\!]}$ and $\eta_n := \eta_{0,n}$.

Definition 2.3. Sets of the form $A \times X^{>n}$, where $n \geq 1$ and $A \in \mathcal{A}^{\leq n}$ are called cylinders.

By writing $A \times X^{>n} = \pi_n^{-1}(A)$ it is easy to check that the set of cylinders is closed by union and complement, and thus it is an algebra of sets which generates the product σ-algebra. We now have to define a measure over cylinders which we will then extend to the whole σ-algebra. We also do this for cylinders in $X^{\geq n}$ for $n \geq 1$ as we will need it for the extension afterwards.

Definition 2.4. Let $x_0 \in X_0$. If $A := A' \times X^{>n} \subseteq X^{\geq 1}$ is a cylinder we set

$$P_0(x_0, A) := \eta_n(x_0, A').$$

This map is well defined because if $n \leq m$ then

$$\begin{aligned}
\eta_m\left(x_0, A' \times X^{[\![n+1,m]\!]}\right) &= \eta_n \otimes \eta_{n+1,m}\left(x_0, A' \times X^{[\![n+1,m]\!]}\right) \\
&= \int_{A'} \eta_{n+1,m}\left(x_0, ..., x_n, X^{[\![n+1,m]\!]}\right) \eta_n(x_0, \mathrm{d}x_1 ... \mathrm{d}x_n) \\
&= \int \mathbb{1}_{A'} \eta_n(x_0, \mathrm{d}x_1 ... \mathrm{d}x_n) \\
&= \eta_n(x_0, A').
\end{aligned}$$

For $k \geq 1$ and $A := A' \times X^{>n} \subseteq X^{>k}$,we define likewise

$$P_k(x_0, ..., x_k, A) := \eta_{k,n}(x_0, ..., x_k, A').$$

Lemma 2.5. *For any $k \in \mathbb{N}$ and any $(x_0, ..., x_k) \in X^{\leq k}$, the map $P_k(x_0, ..., x_k, \cdot)$ is an additive content over the cylinders. Namely if A, B are two disjoint cylinders then $P_0(x_0, ..., x_k, A \cup B) = P_0(x_0, ..., x_k, A) + P_0(x_0, ..., x_k, B)$.*

Proof. We only give the proof in the case $k = 0$ as the other cases are identical. Let $A := A' \times X^{>m}$ and $B := B' \times X^{>n}$ be two disjoint cylinders and assume without loss of generality that $m \leq n$. We then set $A'' := A' \times X^{[\![m+1,n]\!]}$. As $A \cap B = \varnothing$ this implies $A'' \cap B' = \varnothing$. Moreover, $A \cup B = (A'' \cup B') \times X^{>n}$. Consequently :

$$P_0(x_0, A \cup B) = \eta_n(x_0, A'' \cup B') = \eta_n(x_0, A'') + \eta_n(x_0, B') = P_0(x_0, A) + P_0(x_0, B). \qquad \square$$

To apply Carathéodory theorem it remains to prove that $P_0(x_0, \cdot)$ is σ-additive over cylinders. Because it is an additive content and cylinders are an algebra of sets it is enough to show the following result:

Lemma 2.6. *Let $x_0 \in X_0$. If $(A_n)_{n \in \mathbb{N}}$ is a non-increasing sequence of cylinders with emty intersection then $P_0(x_0, A_n) \longrightarrow_{n\to\infty} 0$.*

Proof. For any n we write $A_n := B_n \times X^{>a_n}$. In this proof if $A \subseteq X^{\geq 1}$ et $(x_1, ..., x_n) \in X^{[\![1,n]\!]}$ we will denote by $A(x_1, ..., x_n) := \{y \in X^{>n} : (x_1, ..., x_n, y) \in A\}$ the associated section of A.

We use contraposition. Assume there exists $\epsilon > 0$ such that for any $n \in \mathbb{N}$, $P_0(x_0, A_n) \geq \epsilon$. We wish to prove that the intersection of A_n is nonempty. We define the following sequence of functions:

$$\begin{aligned} f_n : X_1 &\to [0,1] \\ x_1 &\mapsto P_1(x_0, x_1, A_n(x_1)). \end{aligned}$$

It allows us to write

$$\begin{aligned} P_0(x_0, A_n) &= \eta_{a_n}(x_0, B_n) \\ &= \kappa_0 \otimes \eta_{1,a_n}(B_n) \\ &= \int_{X_1} \eta_{1,a_n}(x_0, x_1, B_n(x_1))\kappa_0(x_0, \mathrm{d}x_1) \\ &= \int_{X_1} P_1(x_0, x_1, A_n(x_1))\kappa_0(x_0, \mathrm{d}x_1) \\ &= \int_{X_1} f_n(x_1)\kappa_0(x_0, \mathrm{d}x_1). \end{aligned}$$

Because (A_n) is non-increasing it is also the case of (f_n). We set $g := \lim_{n\to\infty} f_n$. By dominated convergence theorem we get $\epsilon \leq P_0(x_0, A_n) \longrightarrow_{n\to\infty} \int_{X_1} g(x_1)\kappa_0(x_0, \mathrm{d}x_1)$. As $\kappa_0(x_0, \cdot)$ is a probability measure there exists $x_1 \in X_1$ such that $g(x_1) \geq \epsilon$, and so for any $n \in \mathbb{N}$, $f_n(x_1) \geq \epsilon$ which implies that $P_1(x_0, x_1, A_n(x_1)) \geq \epsilon$.

We can thus repeat the same step replacing $P_0(x_0, \cdot)$ by $P_1(x_0, x_1, \cdot)$ and A_n by $A_n(x_1)$. We get a sequence $x := (x_k)_{k\geq 1}$ such that for any $n \in \mathbb{N}$ and $k \geq 1$ we have $P_k(x_0, ..., x_k, A_n(x_1, ..., x_k)) \geq \epsilon$. It remains to show that $x \in \bigcap_{n\in\mathbb{N}} A_n$.

Let $n \in \mathbb{N}$. As $A_n = B_n \times X^{>a_n}$ we in particular have that $A_n(x_1, ..., x_{a_n})$is either $X^{>a_n}$ or empty according to wether $(x_1, ..., x_{a_n})$ belongs to B_n or not. But as $P_{a_n}(x_0, ..., x_{a_n}, A_n(x_1, ..., x_{a_n})) \geq \epsilon > 0$ we have $A_n(x_1, ..., x_{a_n}) = X^{>a_n}$, which proves that $x \in A_n$. $\square$

This concludes the main argument of the proof. The measurability is easily provided by applying the π-λ theorem. We will now focus on those aspects of the proof which proved difficult to formalize and how they were overcome.

3 Dealing with composition of kernels

One of the key tools for the Ionescu-Tulcea theorem is to be able to speak about the composition of Markov kernels. Indeed that is necessary even to be able to state the theorem. However formalizing this in a way which allows us to smoothly write products proved to be quite challenging.

One common rule in formalization is that if we use "..." in a proof it will likely be hard to formalize. In our case the statement of Theorem 2.2 itself makes use of these when we write the composition of kernels, and they hide a real problem.

Transition kernels and their composition are already defined in Lean. However we need to take the composition of a finite family of kernels which does not exist in Mathlib, and here is why. By hand, if we have three Markov kernels $\kappa : X \to_* Y$, $\eta : X \times Y \to_* Z$ and $\xi : X \times Y \times Z \to_* T$ we can define $\kappa \otimes \eta \otimes \xi := (\kappa \otimes \eta) \otimes \xi$. Then we easily get $\kappa \otimes \eta \otimes \xi = \kappa \otimes (\eta \otimes \xi)$. But here we forget a detail: $X \times Y \times Z$ does not mean anything a priori. We can write $(X \times Y) \times Z$ or $X \times (Y \times Z)$ but those are two different types. In paper maths they are canonically isomorphic and we often act as though they were the same but it Lean this isomorphism has to be made explicit.

Coming back to kernels I can write that ξ is a kernel from $X \times (Y \times Z)$ to T. Then $(\kappa \otimes \eta) \otimes \xi$ is a kernel from X to $(Y \times Z) \times T$. But in that case $\kappa \otimes (\eta \otimes \xi)$ is not even well defined because ξ should start from $(X \times Y) \times Z$. If we were to add an isomorphism to define this object anyway the equality $(\kappa \otimes \eta) \otimes \xi = \kappa \otimes (\eta \otimes \xi)$ would still make no sense because on the left we have a kernel from X to $(Y \times Z) \times T$ while on the right it is a kernel from X to $Y \times (Z \times T)$, so they do not have the same type.

A similar issue arises when formalizing Theorem 2.2: if we have two kernels $\kappa : X_0 \to_* X^{[\![1,n]\!]}$ and $\eta : X^{\leq n} \to_* X^{[\![n+1,m]\!]}$, their composition does not make sense, we need isomorphisms. First to see η as a kernel from $X_0 \times X^{[\![1,n]\!]}$ to $X^{[\![n+1,m]\!]}$ to get $\kappa \otimes \eta : X_0 \to_* X^{[\![1,n]\!]} \times X^{[\![n+1,m]\!]}$, then to see $\kappa \otimes \eta$ as a kernel from X_0 to $X^{[\![1,m]\!]}$. To solve this problem we need to make these isomorphisms explicit and then hide them inside a black box to be able to use it without worrying about the implementation.

This work was initiated by Degenne. He introduced a compositon of kernels indexed by integer intervals which makes the equivalence invisible.

```
def compProdNat {i j k : ℕ} (κ : Kernel ((l : Iic i) → X l)
    ((l : Ioc i j) → X l))
    (η : Kernel ((l : Iic j) → X l) ((l : Ioc j k) → X l)) :
    Kernel ((l : Iic i) → X l) ((l : Ioc i k) → X l)
```

This takes two kernels $\kappa : X^{\leq i} \to_* X^{[\![i+1,j]\!]}$ and $\eta : X^{\leq j} \to_* X^{[\![j+1,k]\!]}$ and returns a kernel from $X^{\leq i}$ to $X^{[\![i+1,k]\!]}$. Degenne then proves that this composition is associative, and uses it to define

```
def kerNat (κ : (k : ℕ) → Kernel ((l : Iic k) → X l) (X (k + 1))) (i j : ℕ) :
    Kernel ((l : Iic i) → X l) ((l : Ioc i j) → X l).
```

This takes as an argument a family of kernels $(\kappa_k)_{k \in \mathbb{N}}$ with $\kappa_k : X^{\leq k} \to X_{k+1}$ just as in Theorem 2.2. Then given two integers i and j it returns a kernel `kerNat κ i j` which is the composition

$$\kappa_i \otimes \kappa_{i+1} \otimes \cdots \otimes \kappa_{j-1}.$$

We are therefore now able to speak about those compositions which are necessary to state the theorem. However `kerNat` proved to be not enough for later purposes. Indeed after proving the theorem and trying to write some API we thought that it would be best to to have a kernel which does not necesarrily starts at time 0 but at any instant. This would prove nicer for example when it comes to write formulas for the conditional expectation against the Ionescu-Tulcea kernel (see Section). However we bumped into an issue. Having different kernels starting at different times was proving hard to manipulate even with the compositions that were implemented. That is why I chose to add another layer on top of `kerNat` which proved to work out smoothly.

```
def partialKernel (a b : ℕ) :
    Kernel ((i : Iic a) → X i) ((i : Iic b) → X i)
```

Given integer a and b this function will output a kernel from $X^{\leq a}$ to $X^{\leq b}$. The idea is that the input of this kernel is the full trajectory up to time a and the output is the distribution of the full trajectory up to time b. Of course this distribution is deterministic up to time a and then we compose with `kerNat κ a b`. The advantage of this construction is that in the end we always manipulate kernels with domains and codomains of the form $X^{\leq n}$ for some integer n which makes all the composition much easier. For example if we take a, b and c three integers satisfying $a \wedge c \leq b$ we get

```
(partialKernel κ b c) ∘ₖ (partialKernel κ a b) = partialKernel κ a c
```

where $\circ_{\mathrm{k}}$ is the composition of kernels. This nice property has the advantage to be expressed with a regular operation on kernels rather than one which would be made up for the occasion. With this tool we get a satisfactory way of expressing the composition of multiple kernels.

4 Sections of sets

We see that the proof of Lemma 2.6 heavily relies on the use of sections of sets. This however is not well suited for formalization purposes because of the following. The set $\prod_{n\in\mathbb{N}} X_n$ is represented in lean as the type of functions f with domain $\mathbb{N}$ and such that for any $n \in \mathbb{N}$ $f(n)$ has type X_n. We encounter the same issue as before: $\prod_{n\in\mathbb{N}} X_n$ is different from $X_0 \times X^{\geq 1}$ so we have to use isomorphisms. Moreover, if $A \subseteq \prod_{n\in\mathbb{N}} X_n$ and $x \in X_0$then $A_x \subseteq X^{\geq 1}$. So A_x is a set of sequences indexed by $\mathbb{N}_+$ which is not a subset of $\mathbb{N}$ but rather a subtype in Lean. This makes things hard to manipulate because we want to deal with an element of $\mathbb{N}_+$ we have to provide a proof that it is positive. This would have to be done not only for non-zero integers but also for integers greater than 2, 3 and so on with the same troubles each time. Therefore Sébastien Gouëzel suggested another approach.

The idea relies on the notion of marginals exposed in Section 4 of [Van2024]. I generalized their idea as follows: given a and b integers, $f : X \to [0, +\infty]$ and $x \in X$ I defined `lamrginalPartialKernel κ a b f x` as

$$\int_{X^{\leq b}} f(y \rightsquigarrow_b x) \mathrm{d}\eta_{a,b}(x_0, ..., x_a).$$

This is great because it allows to write

`lamrginalPartialKernel κ b c (lamrginalPartialKernel κ a b f x) x`,

in other words we can first integrate between a and b and then between b and c. This allows us to peel off the measure of a cylinder as the integral of its indicator:

$$P_0(x_0, A_n) = \int_{X_1} P_1(x_0, x_1, A_n(x_1))\kappa_0(x_0, \mathrm{d}x_1).$$

5 Dependence in coordinates

This trick of considering some function as defined over X while it really only depends on the coordinates $\geq n$ brings the necessity of introducing a simple way to indicate which variables a given function actually depends on. We therefore introduced the following predicate.

```
def DependsOn (f : ((i : ι) → α i) → β) (s : Set ι) : Prop :=
∀ ⦃x y⦄, (∀ i ∈ s, x i = y i) → f x = f y
```

The idea is that if $f : \prod_{i\in\iota} \alpha_i \to \beta$ and $s \subset \iota$, `DependsOn f s` means that to check $f(x) = f(y)$ it is enough to show that for any $i \in s$, $x_i = y_i$. In other words f only depends on coordinates indexed by s. For example, if f only depends on coordinates $\leq b$ and we integrate over its coordinates between a and b then the output only depends on coordinates $\leq a$. This is formalized as follows.

```
theorem dependsOn_lmarginalPartialKernel (a : ℕ) {b : ℕ}
  {f : ((n : ℕ) → X n) → ℝ≥0∞}
  (hf : DependsOn f (Iic b)) (mf : Measurable f) :
  DependsOn (lmarginalPartialKernel κ a b f) (Iic a)
```

6 Writing an API

When formalizing a new mathematical object one usually does not stop at writing the definition. Indeed the objected is intended to be usable by all the users without having to know how the object was implemented. That is why it is important to accompany the object with many results which allow other users to use the object without knowing the implementation details.

We now present some of these results. The API is still very basic and will be developed as other users pick up the object to formalize more objects. The best way to find new lemmas to formalize is to choose a target statement and try and formalize it. In our case we aim to prove the following result.

Notation. For $n \in \mathbb{N}$ we set $\mathcal{F}_n := \pi_{[\![0,n]\!]}^{-1}\left(\mathcal{A}^{\leq n}\right)$.

Theorem 6.1. *Let $a, b \in \mathbb{N}$ with $a \leq b$. Let $u \in X^{\leq a}$. Let $f : X \to \mathbb{R}$ be an integrable function with respect to $\lambda_a(u)$. Then for $\lambda_a(u)$-almost every $x := (x_n)_{n\in\mathbb{N}}$, f is integrable with respect to $\lambda_b(x_0, ..., x_b)$ and*

$$\mathbb{E}_{\lambda_a(u)}[f \mid \mathcal{F}_n](x) = \int_X f(y)\lambda_b(x_0, ..., x_b, \mathrm{d}y).$$

Let us give the proof to highlight the necessary intermediate results. Take $A \in \mathcal{A}^{\leq b}$ and set $\pi := \pi_{[\![0,b]\!]}$. Our goal is to prove that

$$\int_{\pi^{-1}(A)} \int_X f(y)\lambda_b(\pi(x), \mathrm{d}y)\lambda_a(u, \mathrm{d}x) = \int_{\pi^{-1}(A)} f \mathrm{d}\lambda_a.$$

We start by rewriting the outer integral using the push-forward measure:

$$\int_{\pi^{-1}(A)} \int_X f(y)\lambda_b(\pi(x), \mathrm{d}y)\lambda_a(u, \mathrm{d}x) = \int_A \int_X f(y)\lambda_b(\pi(x), \mathrm{d}y)(\pi_*\lambda_a)(u, \mathrm{d}x).$$

The first we will need is that $\pi_*\lambda_a = \eta_{a,b}$. This is quite obvious: as λ_a gives the whole trajectory, keeping only the cooordinates up to time b yields the trajectory up to time b. This is formalized as below.

```
theorem ionescuTulceaKernel_proj (a b : ℕ) :
(ionescuTulceaKernel κ a).map (frestrictLe b) = partialKernel κ a b
```

So we get

$$\int_{\pi^{-1}(A)} \int_X f(y)\lambda_b(\pi(x), \mathrm{d}y)\lambda_a(u, \mathrm{d}x) = \int_A \int_X f(y)\lambda_b(\pi(x), \mathrm{d}y)\eta_{a,b}(u, \mathrm{d}x).$$