

Profinite monads on $\mathcal{S}et$ from a Stone perspective

M. Zaïdi

19 July 2016

Contents

Introduction	5
1 Prerequisites	7
1.1 Categorical preliminaries	7
1.2 Stone duality for Boolean algebras	16
1.3 Profiniteness and duality	28
2 Constructing the profinite monad \bar{T}	33
2.1 Ideas and motivations	33
2.2 An algebraic point of view on Stone duality on $\text{Rec}(A)$	34
2.3 Categorical constructions of the profinite monad	40
3 Relating \bar{T} to a monad on the category of Stone spaces	45
3.1 Ideas and motivation	45
3.2 Kan extensions	46
3.3 \bar{T} with Kan extensions and \tilde{T}	54
Conclusion	57
Bibliography	59

Introduction

The starting point of this paper is the article *Recognisable Languages over Monads* [1].

Theories of regular languages and finite automata have strong connections with algebraic theories such as the theory of finite monoid: results such as "regular languages of finite words are precisely the languages recognized by finite monoids" are well known examples.

The categorical generalization of monoids being monads, it seems natural to think that studying them will lead to new possibilities in terms of recognition. In his work, Bojańczyk applies monads and their algebras to extend recognition to different structures such as words, chains and trees.

He also shows that some results about languages can be stated and proved on the level of monads: Myhill-Nerode theorem, Eilenberg's pseudovariety theorem... All the monads T considered by Bojańczyk are taken on the category of sets, and an important part to get to his results relies on the construction of a new monad: the profinite monad \overline{T} .

Since profinite constructions and the category of sets are involved, one could expect that Stone duality will be a powerful tool that could be used to understand the situation : we will see that this is indeed the case, since we can actually get the profinite monad \overline{T} from a monad on the category of Stone spaces, named \tilde{T} .

This article is divided in three parts: first of all, we will present a compendium of the techniques (categorical, topological and language theoretic) required to understand the construction of the profinite monad.

Then, we will quickly recall the construction of the profinite monad by Bojańczyk, taking the opportunity to develop the points that were not treated in full details in his article, and thus present it in a slightly different way.

Finally, we express the profinite monad \overline{T} as a composition of functors that involves a monad \tilde{T} on the category of Stone spaces, and Kan extensions.

We hope that this presentation of \overline{T} , more technical in terms of categorical tools, but more synthetic and natural, might constitute an useful characteriza-

tion of profinite monads in practice.

Chapter 1

Prerequisites

Contents

1.1	Categorical preliminaries	7
1.2	Stone duality for Boolean algebras	16
1.3	Profniteness and duality	28

1.1 Categorical preliminaries

In this section, we will introduce all the different concepts related to monads that will be useful in the following work.

The reader is supposed to be familiar with the most basic notions of category theory (functors, natural transformations, limits/colimits, adjunctions), and one does not need to know more to understand the paper.

The only exception to this statement is the notion of Kan extension, required in the third chapter, which will be introduced and developed by then.

Let's open this section with the definition of the concept motivating all of our work in this article: the notion of monad.

A reason to study these categorical objects is that some results about language theory can be stated, and proved, at the level of monads.

One can easily make the parallel with the strong relationship existing between language theory and the algebraic theory of monoids, for instance facts such as: "*a language is rational if and only if its syntactic monoid is finite*".

The idea motivating the definition of a monad is fairly simple to picture: we generalize, at the categorical level, the axioms involved in the definition of a monoid:

Definition 1.1.1.

Let \mathcal{C} be a category.

A monad on \mathcal{C} is given by (T, η, μ) , where:

- $T : \mathcal{C} \rightarrow \mathcal{C}$ is an endofunctor,
- $\mu : T \circ T \rightarrow T$ is a natural transformation,
- $\eta : Id_{\mathcal{C}} \rightarrow T$ is a natural transformation,

such that, for every object X in \mathcal{C} , the following diagrams commute:

$$\begin{array}{ccc}
 TTTX & \xrightarrow{T\mu_X} & TT X \\
 \downarrow \mu_{TX} & & \downarrow \mu_X \\
 TT X & \xrightarrow{\mu_X} & T X
 \end{array}
 \quad
 \begin{array}{ccc}
 T X & \xrightarrow{T\eta_X} & TT X \\
 \downarrow \eta_{TX} & \searrow Id_{TX} & \downarrow \mu_X \\
 TT X & \xrightarrow{\mu_X} & T X
 \end{array}$$

(We will often, abusively, refer to the monad as T .)

Example 1.1.2.

Let's denote by *Mon* the category of monoids, *Set* the category of sets.

The idea is the following: we want to take a monoid (Σ, \cdot) , forget everything about its monoid structure, and endow the remaining set with a new monoid structure: only then, we will consider the new carrier set we obtained.

With this idea in mind, let's define the following correspondence:

$$T : \left\{ \begin{array}{l} \text{Set} \longrightarrow \text{Set} \\ \Sigma \longmapsto \Sigma^* \\ (\Sigma \xrightarrow{f} \Gamma) \longmapsto (\Sigma^* \xrightarrow{T(f)} \Gamma^*) \end{array} \right.$$

where:

- Σ^* is the carrier set of the free monoid generated by Σ : the set of formal expressions that one can form by concatenating elements of Σ (or said in another way, the words on the alphabet Σ).

- Our correspondence is defined on arrows $f : \Sigma \rightarrow \Gamma$ in the following way:

$$T(f) : \left\{ \begin{array}{l} \Sigma^* \longrightarrow \Gamma^* \\ w = a_1 * \dots * a_n \longmapsto T(f)(w) = f(a_1) * \dots * f(a_n) \end{array} \right.$$

This expression is well defined thanks to the unicity of the expression of a word as a concatenation of letters.

This correspondence is actually a functor, it is very easy to check it by looking at what the action actually does on arrows:

- $T(Id_\Sigma)$ is the mapping

$$T(Id_\Sigma) : \left\{ \begin{array}{l} \Sigma^* \longrightarrow \Sigma^* \\ a_1 * \dots * a_n \longmapsto Id(a_1) * \dots * Id(a_n) = a_1 * \dots * a_n \end{array} \right.$$

therefore we have: $T(Id_\Sigma) = Id_{\Sigma^*}$

- Let's consider two mappings $f : \Sigma \rightarrow \Gamma$ and $g : \Gamma \rightarrow \Lambda$.
Then, by evaluation on any word $w \in \Sigma^*$, we see immediately that applying f then g on w or applying $g \circ f$ are exactly the same:
 $\forall w = a_1 * \dots * a_n \in \Sigma^*$,

$$T(g)T(f)(w) = T(g)(f(a_1) * \dots * f(a_n)) \quad (1.1)$$

$$= g \circ f(a_1) * \dots * g \circ f(a_n) \quad (1.2)$$

$$= T(g \circ f)(w) \quad (1.3)$$

so: $T(g \circ f) = T(g)T(f)$.

Furthermore, we can view this endofunctor as a monad by setting the natural transformations:

- $\eta : Id_{Set} \rightarrow T$ defined for every set Σ by:

$$\eta_\Sigma : \left\{ \begin{array}{l} \Sigma \longrightarrow \Sigma^* \\ a \longmapsto [a] \end{array} \right.$$

where $[a]$ is just the letter a of Σ^* , but seen as a word of length 1.

- $\mu : T^2 \rightarrow T$ defined for every set Σ by:

$$\mu_\Sigma : \left\{ \begin{array}{l} (\Sigma^*)^* \longrightarrow \Sigma^* \\ w_1 * \dots * w_n \longmapsto a_{1,1} * \dots * a_{1,i_1} * \dots * a_{n,1} * \dots * a_{n,i_n} \end{array} \right.$$

where, $\forall j \in \{1, \dots, n\}$, $w_j = a_{j,1} * \dots * a_{j,i_j}$ is a word on Σ with length i_j . This simply means that we see a word of word as a word, by flattening it.

Let's check that they are indeed natural transformations:

- We want to check that, for all sets Σ^* , Σ'^* , the following diagram commutes:

$$\begin{array}{ccc} \Sigma & \xrightarrow{f} & \Sigma' \\ \downarrow \eta_\Sigma & & \downarrow \eta_{\Sigma'} \\ \Sigma^* & \xrightarrow{T(f)} & \Sigma'^* \end{array}$$

which is clear by evaluation, for every $a \in \Sigma$:

$$\begin{aligned} (\eta_{\Sigma'} \circ f)(a) &= \eta_{\Sigma'}(f(a)) = [f(a)] \\ T(f)(\eta_\Sigma)(a) &= T(f)[a] = [f(a)] \end{aligned}$$

- We want to check that, for all sets Σ^* , Σ'^* , the following diagram commutes:

$$\begin{array}{ccc} (\Sigma^*)^* & \xrightarrow{TT(f)} & (\Sigma'^*)^* \\ \downarrow \mu_\Sigma & & \downarrow \mu_{\Sigma'} \\ \Sigma^* & \xrightarrow{T(f)} & \Sigma'^* \end{array}$$

which is clear by evaluation, for every $w_1 * \dots * w_n \in (\Sigma'^*)^*$:

$$\begin{aligned} (\mu_{\Sigma'} \circ TTf)(w_1 * \dots * w_n) &= \mu_{\Sigma'}(f(w_1) * \dots * f(w_n)) \\ &= f(a_{1,1}) * \dots * f(a_{1,i_1}) * \dots * f(a_{n,1}) * \dots * f(a_{n,i_n}) \\ &= T(f) \circ \mu_\Sigma(w_1 * \dots * w_n) \end{aligned}$$

What is left is the verification that we actually have a monad, by checking the two diagram properties of definition 1.1.1, which is not harder than what we did previously, we just need to understand what the functor effectively do:

- The first diagram in the definition of a monad is almost immediate: it only means that, taking an expression with 3 levels of brackets, it is the same to erase exterior brackets first and then intermediate brackets (which have become exterior after the first operation), as it is to erase intermediate bracket first and then exterior brackets.

- On one hand, $T\eta_\Sigma$ sends, for instance an expression of the form $[a]$, where $a \in \Sigma$, to $[[a]]$, and more generally any expression to the expression obtained by doubling the brackets around every element of Σ in the expression.
- On the other hand, $\eta_{T\Sigma}$ sends an expression, let's note it S , to $[S]$ defined by taking the bracket of the whole expression: this being said, the relations $\mu \circ T\eta = Id = \mu \circ \eta_T$ are clear.

Example 1.1.3.

More generally, we can get a monad from any adjunction.

Let \mathcal{C} and \mathcal{D} be categories, and $F : \mathcal{C} \rightarrow \mathcal{D}$, $G : \mathcal{D} \rightarrow \mathcal{C}$ be functors.

We suppose that F is left adjoint to G , with $\eta : Id_{\mathcal{C}} \rightarrow G \circ F$ the unit, and $\epsilon : F \circ G \rightarrow Id_{\mathcal{D}}$ the counit:

$$\begin{array}{ccc} & F & \\ \mathcal{C} & \begin{array}{c} \curvearrowright \\ \perp \\ \curvearrowleft \end{array} & \mathcal{D} \\ & G & \end{array}$$

Then, one can check that the data of $(G \circ F, \eta, G \circ \epsilon \circ F)$ satisfies the axioms defining a monad, a complete proof can be found in [9].

Definition 1.1.4.

Let \mathcal{C} be a category and T a monad on \mathcal{C} .

A T -algebra on \mathcal{C} is given by $(A, mult_A)$, where:

- A is an object of \mathcal{C} ,
- $mult_A : TA \rightarrow A$ is an arrow in \mathcal{C} ,

such that the following diagram commutes:

$$\begin{array}{ccc} TTA & \xrightarrow{\mu_A} & TA \\ T(mult_A) \downarrow & & \downarrow mult_A \\ TA & \xrightarrow{mult_A} & A \end{array} \quad \begin{array}{ccc} A & \xrightarrow{\eta_A} & TA \\ & \searrow Id_A & \downarrow mult_A \\ & & A \end{array}$$

Once again, we will often denote abusively a T -algebra $(A, mult_A)$ as A .

Example 1.1.5.

Let $T : \mathcal{C} \rightarrow \mathcal{C}$ be a monad over a category \mathcal{C} . Then, in particular, the axioms in the definition of a monad give us, for any object A in \mathcal{C} , the diagrams:

$$\begin{array}{ccc}
TTA & \xrightarrow{T\mu_A} & TTA & T A & \xrightarrow{\eta_{(TA)}} & TTA \\
\downarrow \mu_{TA} & & \downarrow \mu_A & \searrow Id_{TA} & & \downarrow \mu_A \\
TTA & \xrightarrow{\mu_A} & TA & & & TA
\end{array}$$

Thus for any object A in \mathcal{C} , (TA, μ_A) is a T -algebra.

Definition 1.1.6.

Let \mathcal{C} be a category and T a monad on \mathcal{C} .

Let A and B be T -algebras, then a T -morphism is an arrow $h : A \rightarrow B$ such that the following diagram commutes:

$$\begin{array}{ccc}
TA & \xrightarrow{Th} & TB \\
mult_A \downarrow & & \downarrow mult_B \\
A & \xrightarrow{h} & B
\end{array}$$

With the previous definitions in mind, it is only a trivial verification to check that taking for objects the T -algebras, and for arrows the T -morphisms gives us a category, we'll denote it as \mathcal{C}^T and call it the Eilenberg-Moore algebras category of the monad T (it is what we think about as "the category of algebras" over the monad T).

There are two canonical functors between \mathcal{C}^T and \mathcal{C} :

- $U : \left| \begin{array}{ccc} \mathcal{C}^T & \longrightarrow & \mathcal{C} \\ (X, mult_X) & \longmapsto & X \\ ((X, mult_X) \xrightarrow{f} (Y, mult_Y)) & \longmapsto & (X \xrightarrow{U(f)} Y) \end{array} \right.$
the forgetful functor which sends a T -algebra $(X, mult_X)$ to the object X , and a T -morphism $f : X \rightarrow Y$ to itself.

- $L : \left| \begin{array}{ccc} \mathcal{C} & \longrightarrow & \mathcal{C}^T \\ X & \longmapsto & (TX, \mu_X) \\ (X \xrightarrow{f} Y) & \longmapsto & ((TX, \mu_X) \xrightarrow{L(f)} (TY, \mu_Y)) \end{array} \right.$
the free algebra functor which sends an object X in \mathcal{C} to the T -algebra (TX, μ_X) , and sends an arrow $f : X \rightarrow Y$ to the T -morphism $T(f) : T(X) \rightarrow T(Y)$.

These two functors are actually strongly related:

Property 1.1.7.

For every monad (T, η, μ) on \mathcal{C} , L is left adjoint to U , and furthermore:

$$T = U \circ L$$

Proof.

First of all, the fact that $T = U \circ L$ is obvious by applying directly the definition of U and L on objects and arrows of \mathcal{C} .

Now, to show the expected adjunction, we want to exhibit a natural bijection:

$$\theta : \text{Hom}_{\mathcal{C}^T}((T(X), \mu_X), (Y, k)) \rightarrow \text{Hom}_{\mathcal{C}}(X, Y)$$

Let $f : T(X) \rightarrow Y$ be a T -algebra morphism, we set:

$$\theta(f) = f \circ \eta_X \in \text{Hom}_{\mathcal{C}}(X, Y)$$

On the other hand, let $g : X \rightarrow Y$ be an arrow in \mathcal{C} , then we take:

$$\phi(g) = k \circ T(g)$$

We'll show that $\phi(g)$ is actually a T -morphism, and that ϕ the inverse of θ , making θ a bijection:

- $k \circ T(\phi(g)) = k \circ T(k) \circ T^2(g) = k \circ (\mu_Y) \circ T^2(g) = k \circ T(g) \circ \mu_X = \phi(g) \circ \mu_X$
So we have a T -algebra morphism.

- $\phi(\theta(f)) = k \circ T(f \circ \eta_X) = k \circ T(f) \circ T(\eta_X) = f \circ \mu_X \circ T(\eta_X) = f$
 $\theta(\phi(g)) = k \circ T(g) \circ \eta_X = k \circ \eta_Y \circ g = g$

So ϕ is the inverse of θ .

Finally we have to check that θ is natural in X and in (Y, k) , but for every $u : X' \rightarrow X$ and $v : (Y, k) \rightarrow (Y', k')$ we have:

$$\theta(f \circ T(u)) = f \circ T(u) \circ \eta_{X'} = f \circ \eta_X \circ u = \theta(f) \circ u, \text{ and}$$

$$\theta(v \circ f) = v \circ f \circ \eta_X = v \circ \theta(f), \text{ which allows us to conclude.}$$

□

Now that all of the monad theory background has been presented, we'll close this section by introducing recognition theory for a monad over $\mathcal{S}et$.

In this part, we will suppose in this part that the reader is familiar with the

basics of the theory of recognition by finite automata, an introduction to these methods can be found in [4] for instance.

To generalize recognition theory to the categorical level, the notions we know about finite languages have to be enlarged.

In [1], a general notion of recognition over a monad on any category \mathcal{C} is shortly introduced.

In particular, it is stated that to define a notion of recognition you need:

- A category \mathcal{C}
- A monad T on \mathcal{C}
- A notion of finite alphabet
- A notion of finite T -algebra

In our framework though, since we want to study monads over $\mathcal{S}et$, the ingredients one can take are easy to identify: if we consider $\mathcal{C} = \mathcal{S}et$, and look at a monad T on $\mathcal{S}et$, then the natural choice for a finite alphabet is a finite set, and for a finite T -algebra a T -algebra whose carrier set is finite.

Before going through our specific case, let's begin this section by introducing a monadic generalization of the notion of languages, T -languages:

Definition 1.1.8.

Let \mathcal{C} be a category, T a monad on \mathcal{C} , A a fixed T -algebra.

A T -colouring on A is the data of an object C in \mathcal{C} , and an arrow $\phi : A \rightarrow C$ in \mathcal{C} .

In particular, if X is an object of \mathcal{C} , we've already seen in the first part that TX is a T -algebra, thus we talk about a T -colouring on X when, according to the previous definition, we should say "a T -colouring on TX ".

Example 1.1.9.

Let's consider the example of $\mathcal{C} = \mathcal{S}et$, $C = \{0,1\}$, and T the monad of monoids introduced in the first part, where $T\Sigma = \Sigma^$, for every set Σ .*

In that case, a T -colouring on a set Σ is only a mapping $\phi : \Sigma^ \rightarrow \{0,1\}$.*

But we know that it is exactly the same as giving a subset $L \subseteq \Sigma^$, more precisely, the subset $L = \phi^{-1}(\{1\})$: we retrieved the usual definition of languages.*

Let's remark that the notion of T -colouring is indeed much more general than languages we're used to deal with, since another choice of C would give us another notion of language.

With the previous notations, if the monad T we're considering is on \mathcal{Set} , and we chose to take $C = \{0, 1\}$, instead of talking about T -colouring we'll talk about T -languages: so thanks to the previous example, a T -language on a set Σ is just a subset $L \subseteq T\Sigma$.

Now, since we generalised the notion of languages to monads, let's do the same with recognition:

Definition 1.1.10.

Let \mathcal{C} be a category, T a monad on \mathcal{C} , A a fixed T -algebra and $(C, \phi : A \rightarrow C)$ a T -colouring on A .

This T -colouring is T -recognisable if and only if there exists a finite T -algebra B , with an arrow $\phi' : B \rightarrow C$, and a T -morphism $h : A \rightarrow B$ such that the following diagram commutes:

$$\begin{array}{ccc} A & \xrightarrow{\phi} & C \\ h \downarrow & \nearrow \phi' & \\ B & & \end{array}$$

Example 1.1.11.

Let's consider the example of $\mathcal{C} = \mathcal{Set}$, $C = \{0, 1\}$, and T any monad on \mathcal{Set} . In that case, as we've already stated it, a T -language on a T -algebra A is just a subset $L \subseteq A$.

Now, by definition, P is T -recognizable if and only if there exists a finite T -algebra B , with a mapping $\phi' : B \rightarrow \{0, 1\}$ and a T -algebra morphism $h : A \rightarrow B$ such that: $\phi = \phi' \circ h$.

However, we know that considering a T -algebra B , with a mapping $\phi' : B \rightarrow \{0, 1\}$ is the same than considering a subset Q , and we also know that, by definition, $\phi(x) = 1$ if and only if $x \in L$, and $\phi' \circ h(x) = 1$ if and only if $h(x) \in Q$, so $x \in h^{-1}(Q)$: thus we have the equality $h^{-1}(Q) = P$.

This example confirms the idea we had that recognition over a T -algebra generalizes recognition over a monoid to the categorical level.

This extension of the notion of recognizable language isn't only for the sake of the theory, one could think about several concrete example where we would need a C different from $\{0, 1\}$, for instance $C = [0, 1]$ could correspond to a case where we can't tell if a word is recognized by an automaton but we can still tell the probability of it being recognized.

Remark 1.1.12.

In the case of a monad over \mathcal{Set} , the one of monoids for instance, the fact that the underlying set of the objects is finite will satisfy this idea of a finite T-algebra, so there is no problem to talk about recognition over a T-algebra. However, it is not often obvious to define it, and one should keep in mind that to generalize this theory to monads over any category, he should define properly a notion of finite T-algebra.

1.2 Stone duality for Boolean algebras

The main purpose of this section is to demonstrate the following statement which will be fundamentally important for the rest of the paper:

There is a categorical duality between the category of Boolean algebras with Boolean homomorphisms and the category of Stone spaces with continuous mappings.

This duality between Boolean algebras and Stone spaces is a very powerful tool. It transfers algebraic objects to topological ones, without losing any information about them during the process. Let's begin this section by explaining what we exactly mean by a categorical duality.

Definition 1.2.1.

Let \mathcal{C} and \mathcal{D} be categories, we say that there is a duality between them if we have the anti-equivalence of categories: $\mathcal{C} \simeq \mathcal{D}^{op}$

The definition says it all: it is just an equivalence of categories involving two contravariant functors.

Duality in mathematics usually refers to an equivalence between two things, but in an opposed way: in this incarnation of the concept of duality, the arrows are the things that are reversed.

Therefore, what we want to show is that, up to Boolean isomorphisms and homeomorphisms, there is a one-to-one correspondence between Boolean algebras and Stone spaces.

But first, let's give a definition of Boolean algebras, an algebraic structure which appears naturally in the theory of formal languages.

Definition 1.2.2.

A Boolean algebra is given by $(B, \vee, \wedge, \neg, 0, 1)$ where B is a set, \vee and \wedge are binary operations, \neg unary operation, and 0 and 1 are nullary operations such that, for all $a, b, c \in B$:

- $a \vee b = b \vee a$
- $a \wedge b = b \wedge a$
- $a \vee (b \vee c) = (a \vee b) \vee c$
- $a \wedge (b \wedge c) = (a \wedge b) \wedge c$
- $a \vee a = a$
- $a \wedge a = a$
- $a = a \vee (a \wedge b)$
- $a = a \wedge (a \vee b)$
- $a \vee (b \vee c) = (a \wedge b) \vee (a \wedge c)$
- $a \wedge (b \vee c) = (a \vee b) \wedge (a \vee c)$
- $a \wedge 0 = 0$
- $a \vee 1 = 1$
- $a \wedge (\neg a) = 0$
- $a \vee (\neg a) = 1$

Example 1.2.3.

Let X be a set, then one can easily check that $(\mathcal{P}(X), \cup, \cap, ^c, \emptyset, X)$ satisfies the previous axioms and therefore is an example of a Boolean algebra.

In particular, if \mathcal{A} is an alphabet, then $(\mathcal{P}(\mathcal{A}^), \cup, \cap, ^c, \emptyset, \mathcal{A}^*)$ the set of all languages over \mathcal{A} , considered with the usual operations, is a Boolean algebra.*

Definition 1.2.4.

A non-empty subset $A \subseteq B$ is a Boolean subalgebra of a Boolean algebra $(B, \wedge, \vee, \neg, 0, 1)$ if A is closed under \wedge, \vee and \neg .

Example 1.2.5.

Let \mathcal{A} be an alphabet, the set of recognizable languages $\text{Rec}(\mathcal{A})$ is closed under union, intersection and complement: therefore $\text{Rec}(\mathcal{A})$ is a subalgebra of $(\mathcal{P}(\mathcal{A}^*), \cup, \cap, ^c, \emptyset, \mathcal{A}^*)$.

Definition 1.2.6.

Let B and C be Boolean algebras.

A map $f : B \rightarrow C$ is a Boolean homomorphism if f preserves \vee , \wedge and \neg :

$\forall a, b \in B$,

- $f(a \vee b) = f(a) \vee f(b)$
- $f(a \wedge b) = f(a) \wedge f(b)$
- $f(\neg a) = \neg(f(a))$

If f is injective, it is called a Boolean embedding.

If f is a bijection, it is called a Boolean isomorphism.

Remark 1.2.7.

We can check that the condition $f(\neg a) = \neg(f(a))$ is equivalent to the condition $f(0) = 0$ and $f(1) = 1$.

We denote by $\mathcal{B}\text{ool}$ the category of Boolean algebras whose objects are Boolean algebras and mappings are the Boolean homomorphisms.

We just introduced Boolean algebras from a purely algebraic point of view, however Boolean algebras can also be thought of as special kinds of ordered sets.

Proposition 1.2.8.

Let $(B, \wedge, \vee, \neg, 0, 1)$ be a Boolean algebra.

We define a binary relation on $B : \forall a, b \in B, (a \leq b \iff (a \vee b = b))$

This relation is a partial order on B .

Proof.

- $\forall a \in B, a \vee a = a$ by definition of a Boolean algebra, so we have reflexivity.
- If we have $a \leq b$ and $b \leq a$, then by applying the axiom of commutativity in the definition of a Boolean algebra we have: $a \vee b = b \vee a$, so $a = b$ and thus we have antisymmetry.

- If we have $a \leq b$ and $b \leq c$, then:
 $a \vee c = a \vee (b \vee c) = (a \vee b) \vee c = b \vee c = c$ and finally: $a \leq c$, therefore it is transitive. □

It is well known that the supremum (or infimum) of a subset of an arbitrary poset may not exist.

However, for the posets arising from Boolean algebras, the supremum and infimum of all finite subsets exist and are actually related to the algebraic operation defining B in a very simple way:

Proposition 1.2.9.

Let $(B, \wedge, \vee, \neg, 0, 1)$ be a Boolean algebra and \leq the order on B previously defined.

Then, every finite subset $A \subseteq B$ has both a supremum and an infimum.

Moreover : $\forall a, b \in B, \sup\{a, b\} = a \vee b$ and $\inf\{a, b\} = a \wedge b$

Proof.

We'll only prove the results for the supremum, one can deduce the infimum in an analogous way.

- Let's assume that the second point of the proposition is true, we'll show the first point by induction on the cardinal of A.
 First of all, we know that $\sup(\emptyset) = 0$. If A has two elements, it is obvious by the second point, and if A has three elements or more, since A is finite, considering the upper bound for two elements in A, while keeping at each step the current upper bound will eventually lead to getting the supremum of A.
- First, let's see that $a \vee b$ is an upper bound of a and b:
 $a \vee (a \vee b) = (a \vee a) \vee b = a \vee b$, so $a \leq a \vee b$.
 In the same way: $b \vee (a \vee b) = b \vee (b \vee a) = b \vee a$, so $b \leq a \vee b$.
 Now, let's show that it is the supremum: if $y \in B$ is another upper bound of a and b, then: $a \vee y = y = b \vee y$.
 Thus: $y = a \vee b = a \vee (b \vee y) = (a \vee b) \vee y$, so $a \vee b \leq y$ which is what we wanted. □

Let's remark that the definition of the order over B is directly related to its algebraic structure as a Boolean algebra.

From now on we'll use the algebraic or the order point of view without mentioning it.

When A is a subset of a Boolean algebra, the supremum is often referred to as the join of A and the infimum of A is often called the meet of A .

Let's recap: for now we know a bit more about Boolean algebras, and our final goal is to find a correspondence between them and Stone spaces.

To do so, an important remark is the following: we have seen that powerset algebras were a particular case of Boolean algebras, but, to a weaker extent, the converse is more or less true: we will show that B can be embedded in a powerset algebra!

The reason we bother introducing this representation of Boolean algebra is because considering a topology over a powerset algebra is much more convenient, we will use it to construct the Stone space we need.

Thus we need to find a set Y such that $B \subseteq \mathcal{P}(Y)$.

Definition 1.2.10.

Let X be a set, and \mathcal{F} a non-empty set of subsets of X .

We say that \mathcal{F} is a filter over X if it satisfies the following properties, for every A and B included in X :

- $\emptyset \notin \mathcal{F}$
- $(A \subseteq B, A \in \mathcal{F}) \Rightarrow B \in \mathcal{F}$
- $(A \in \mathcal{F}, B \in \mathcal{F}) \Rightarrow (A \cap B \in \mathcal{F})$

A good way for oneself to represent the notion of filter is to consider that a filter over X contains the "big parts of X ": in that sense, the previous axioms we picked in the definition seem coherent.

Let B be a Boolean algebra, a filter \mathcal{F} of B is said to be proper if $\mathcal{F} \neq B$. It is equivalent to that $0 \notin \mathcal{F}$

Example 1.2.11.

Let X be a set, and $x \in X$ an element of X , we define the following set:

$$\mathcal{F}_x = \uparrow \{x\} = \{A \subseteq X / x \in A\}.$$

We can verify that this set of subsets of X is indeed a filter over X , which will be called the principal filter of x over X :

- \mathcal{F}_x is clearly non-empty since $X \in \mathcal{F}_x$, and also by definition of \mathcal{F}_x the empty set does not belong to \mathcal{F}_x .

- If A and B are included in X , with $A \subseteq B$ and $A \in \mathcal{F}_x$, then we have $x \in A \cap B$, so $x \in B$, and finally $B \in \mathcal{F}_x$.
- If A and B are in \mathcal{F}_x , then x is in A and also in B , thus x is in $A \cap B$ so $A \cap B \in \mathcal{F}_x$.

In this example \mathcal{F}_x simply means that we define "a big set" as a set containing x : the mere presence of x in X decides the size of the set.

Definition 1.2.12.

Let B be a Boolean algebra, and \mathcal{F} a non-empty subset of B .

We say that \mathcal{F} is a filter over B if it satisfies the following properties, for every a and b in B :

- $0 \notin \mathcal{F}$
- $(a \in \mathcal{F}, a \leq b) \Rightarrow b \in \mathcal{F}$
- $(a \in \mathcal{F}, b \in \mathcal{F}) \Rightarrow (a \wedge b \in \mathcal{F})$

Proposition 1.2.13.

Let B be a Boolean algebra, and \mathcal{F} a proper filter over B . Then we say that \mathcal{F} is an ultrafilter over B if it satisfies one of the equivalent conditions:

- $\forall a \in B, (a \in \mathcal{F} \iff \neg a \notin \mathcal{F})$
- $\forall a, b \in B, a \vee b \in \mathcal{F} \Rightarrow (a \in \mathcal{F} \text{ or } b \in \mathcal{F})$

Proof.

- Let's assume that \mathcal{F} is a filter on B satisfying the second condition. Let's take $a \vee b \in \mathcal{F}$ with $b \notin \mathcal{F}$, we want to show that $a \in \mathcal{F}$.
By hypothesis, we have $b \in \mathcal{F}$.
Now, we have: $(a \vee b) \wedge \neg b = (a \wedge \neg b) \vee (b \wedge \neg b) = (a \wedge \neg b)$ and $(a \wedge \neg b) \in \mathcal{F}$, so $(a \vee b) \wedge \neg b \in \mathcal{F}$.
But since $a \wedge \neg b \leq a$, we have $a \in \mathcal{F}$.
Now, since \mathcal{F} is an ultrafilter, there are $c, d \in \mathcal{F}$ such that:
 $c \wedge a = d \wedge b = 0$.
So: $0 = 0 \wedge 0 = ((c \wedge d) \wedge a) \vee ((c \wedge d) \wedge b) = ((c \wedge d) \wedge (a \vee b))$
Since \mathcal{F} is a proper filter containing $c \wedge d$ and $a \vee b$, we have a contradiction.

- Let's assume that \mathcal{F} is a filter on B satisfying the first condition. Then, for all $a \in \mathcal{F}$, $a \vee \neg a = 1$, but $1 \in \mathcal{F}$, thus by hypothesis we will have a or $\neg a$ in \mathcal{F} . However, if we had both of them in \mathcal{F} , then we would have: $a \wedge \neg a = 0 \in \mathcal{F}$, which is absurd since \mathcal{F} is proper. □

First, let's remark that the first condition is the actual definition of ultrafilter in the general case, the second is the definition of a prime filter. In the case of Boolean algebras, the two notions are equivalent, we'll use both of them without mentioning it and note $\text{Ult}(B)$ the set of ultrafilter on B a Boolean algebra.

Furthermore, it is useful to know that ultrafilters constitute maximal elements for the relation of inclusion between filters. For instance, principal filters are an example of an ultrafilter. One can show that every filter of a Boolean algebra is contained in an ultrafilter by using the Zorn lemma.

We are now ready to state the representation theorem about Boolean algebras and powerset algebras we mentioned before:

Proposition 1.2.14.

Let B be a Boolean algebra, then B is isomorphic to a subalgebra of $\mathcal{P}(\text{Ult}(B))$. To be precise, $\phi : \begin{cases} B & \longrightarrow \mathcal{P}(\text{Ult}(B)) \\ a & \longmapsto \{\mathcal{F} \in \mathcal{P}(\text{Ult}(B)) \mid a \in \mathcal{F}\} \end{cases}$ is a Boolean algebra embedding.

Proof.

First, let's prove that ϕ is indeed a Boolean morphism.

Let's consider $\mathcal{F} \in \text{Ult}(B)$, \mathcal{F} is a filter, therefore:

$$(a \wedge b) \in \mathcal{F} \Rightarrow (a \in \mathcal{F} \text{ and } b \in \mathcal{F}).$$

But since \mathcal{F} is also an ultrafilter, we have: $(a \vee b) \in \mathcal{F} \Rightarrow (a \in \mathcal{F} \text{ or } b \in \mathcal{F})$.

Thus,

$$(a \wedge b) = \{\mathcal{F} \in \text{Ult}(B) \mid a \wedge b \in \mathcal{F}\} \tag{1.4}$$

$$= \{\mathcal{F} \in \text{Ult}(B) \mid a \in \mathcal{F}, b \in \mathcal{F}\} \tag{1.5}$$

$$= \{\mathcal{F} \in \text{Ult}(B) \mid a \in \mathcal{F}\} \cap \{\mathcal{F} \in \text{Ult}(B) \mid b \in \mathcal{F}\} \tag{1.6}$$

$$= \phi(a) \cap \phi(b) \tag{1.7}$$

We can prove that $\phi(a \vee b) = \phi(a) \vee \phi(b)$ and $\phi(\neg a) = \neg\phi(a)$ in the exact same way.

All we have left to check is that it is indeed an embedding.

Let's take $a \neq b \in B$, we want to show that $\phi(a) \neq \phi(b)$.

Since $a \neq b$, let's assume that $a \not\leq b$, so we have $a \vee b \neq b$.

Now: $a \wedge \neg b = (a \wedge \neg b) \vee (b \wedge \neg b) = (a \vee b) \wedge b = b \wedge b \neq 0$

Now, \mathcal{F} , the principal filter of $a \wedge \neg b$ is a non-trivial filter containing a and $\neg b$.

As we saw, by maximality of ultrafilters, we can find an ultrafilter \mathcal{F}' such that $\mathcal{F} \subseteq \mathcal{F}'$.

Since \mathcal{F}' is an ultrafilter, $b \notin \mathcal{F}'$ and $b \notin \mathcal{F}$.

Thus, $\mathcal{F} \in \phi(a)$ and $\mathcal{F} \notin \phi(b)$, so finally $\phi(a) \neq \phi(b)$. \square

Before going on, let's prove two properties related to the Boolean embedding ϕ we just exhibited:

Proposition 1.2.15.

- Every element of $Ult(B)$ can be expressed as a join of meets of elements from $\phi(B)$.
- Every element of $Ult(B)$ can be expressed as a meet of joins of elements from $\phi(B)$.
- Let's take $S, T \subseteq B$, such that $\bigwedge \phi(S) \leq \bigvee \phi(T)$.
Then, there is $S' \subseteq S$, $T' \subseteq T$ such that: $\bigwedge S' \leq \bigvee T'$.
- $\phi(B)$ is exactly constituted by the clopen subsets of $Ult(B)$.

Proof.

- Let's consider U in $\mathcal{P}(Ult(B))$, first, we notice that $U = \bigvee_{\mathcal{F} \in \mathcal{P}(Ult(B))} \mathcal{F}$.
All we have left to show is that, for any $\mathcal{F} \in Ult(B)$, $\{\mathcal{F}\}$ is a meet of elements in $\phi(B)$.

Naturally we think about the $\phi(a)$ where $a \in \mathcal{F}$, so we'll show that:

$$\{\mathcal{F}\} = \bigwedge \{\phi(a) \mid a \in \mathcal{F}\}$$

The first inclusion is clear, and conversely, let's consider

$\mathcal{G} \in \bigwedge \{\phi(a) \mid a \in \mathcal{F}\}$. Since $\mathcal{F} \subseteq \mathcal{G}$, and we know that \mathcal{F}, \mathcal{G} are ultrafilters, we have $\mathcal{F} = \mathcal{G}$ by maximality of ultrafilters, which allows us to conclude.

- Reversing \vee and \wedge , the proof is exactly the same.
- Let's consider S, T such that $\bigwedge \phi(S) \leq \bigvee \phi(T)$.
The trick is to take $A = S \cup (\neg T)$, and then \mathcal{F} the smallest filter containing A , or said in another way:

$$\mathcal{F} = \{b \in B \mid a_1 \wedge \dots \wedge a_n \leq b, \text{ where all the } a_i \text{ are in } A\}$$

First, let's show that the filter we created is not a proper filter: if \mathcal{F} was, then we would also have an ultrafilter \mathcal{F}' containing \mathcal{F} , in particular $S \subseteq \mathcal{F}'$. So $\mathcal{F}' \in \{H \in \text{Ult}(B) \mid S \subseteq H\} = \bigcap \phi(S) = \bigwedge \phi(S)$. Since $T \subseteq \mathcal{F}'$, $\mathcal{F}' \notin \{H \in \text{Ult}(B) \mid \exists t \in T, t \in H\} = \bigcup \phi(S) = \bigvee \phi(S)$, thus since $\bigwedge \phi(S) = \bigvee \phi(T)$, it is absurd.

Therefore, \mathcal{F} is not a proper filter, so $0 \in \mathcal{F}$ and by definition:

$$\exists a_1, \dots, a_n \in A, a_1 \wedge \dots \wedge a_n = 0.$$

$$\text{Thus: } \exists s_1, \dots, s_i \in S, \exists t_1, \dots, t_j \in T, s_1 \wedge \dots \wedge s_i \wedge \neg t_1 \wedge \dots \wedge \neg t_j = 0$$

$$\text{Thus: } s_1 \wedge \dots \wedge s_i \wedge \neg(t_1 \wedge \dots \wedge t_j) = 0$$

$$\text{and eventually: } s_1 \wedge \dots \wedge s_i \leq t_1 \wedge \dots \wedge t_j.$$

Finally, by setting $S' = \{s_1, \dots, s_i\}$, $T' = \{t_1, \dots, t_j\}$, we can conclude.

- We can give a topology to $\text{Ult}(B)$ in a natural way: $\forall b \in B$, let's consider $D_b = \{U \in \text{Ult}(B) \mid b \in U\}$: the family of all the D_b for $b \in B$ is a base of a topology on $\text{Ult}(B)$.

Let's consider $U \in \phi(B)$, then $\exists a \in B, U = \phi(a)$.

By definition of the topology on $\text{Ult}(B)$, U is an open subset.

Furthermore, $\text{Ult}(B) \setminus U = \text{Ult}(B) \setminus \phi(a) = \neg \phi(a) = \phi(\neg a)$, so it is an open subset, and thus U is closed: U is a clopen subset.

Let's consider $U \in \text{Cl}(\text{Ult}(B))$, since U is open we have by definition of the topology on $\text{Ult}(B)$, $U = \bigcup \phi(T)$ for some $T \subseteq B$.

In the same way, $\text{Ult}(B) \setminus U = \bigcup \phi(S)$ for some $S \subseteq B$.

So: $U = \text{Ult}(B) \setminus \bigcup \phi(S) = \bigcap \phi(\neg S)$, thus $\bigcap \phi(\neg S) = \bigcup \phi(T)$.

But, thanks to what we saw previously: $\exists S' \subseteq S \exists T' \subseteq T, \bigwedge S' \leq \bigvee T'$.

Now, we have: $U = \bigcap \phi(\neg S) \subseteq \bigcap \phi(S') = \phi(\bigwedge S') \subseteq \phi(\bigvee T')$

$= \bigcup \phi(T') \subseteq \bigcup \phi(T) = U$, and therefore: $\bigvee \phi(T') = U$,

so $U = \phi(\bigvee T')$.

Finally, $U \in \phi(B)$.

□

Now that all of these notions have been introduced, all there is left to show is the correspondence between Boolean algebras and certain topological spaces which will be called Stone spaces.

Definition 1.2.16.

X is a Stone space if it is a compact, Hausdorff totally disconnected topological space.

(Let's recall that a topological space X is totally disconnected if the connected components in X are the one-point sets.)

We note Stone the category of Stone spaces whose objects are Stone spaces and mappings are the continuous homomorphisms.

Now that everything is well defined, we can state the duality:

Proposition 1.2.17.

There is a categorical duality between $\mathcal{B}ool$ and $\mathcal{S}tone$

Proof.

To show the equivalence of categories previously stated, we need to construct two functors $F : \mathcal{B}ool \rightarrow \mathcal{S}tone$, $G : \mathcal{S}tone \rightarrow \mathcal{B}ool$ and two natural transformations $\phi : Id_{\mathcal{S}tone} \rightarrow F \circ G$, $\psi : Id_{\mathcal{B}ool} \rightarrow G \circ F$ such that $G \circ F \simeq Id_{\mathcal{B}ool}$ and $F \circ G \simeq Id_{\mathcal{S}tone}$.

Defining a functor $G : \mathcal{S}tone \rightarrow \mathcal{B}ool$

Taking a Stone space Y , we want to get a Boolean algebra. Since we are given a topology over Y , our first idea would be to take the family of open subsets of Y , but by trying to check at all of the axioms required, we observe that we also need the complementary of open subsets to be in our family. Therefore, by taking $Cl(Y) = \{U \subseteq Y \mid U \text{ is open and closed in } Y\}$ (the clopens), we now check easily that it satisfies all of the required axioms: we have defined the functor on the objects.

Now, let's consider a continuous mapping $f : Y \rightarrow X$, we want to use it to define a Boolean homomorphism $G(f) : Cl(X) \rightarrow Cl(Y)$.

Taking $U \in Cl(X)$, the natural way to get an element in $Cl(Y)$ is by simply putting: $G(f)(U) = f^{-1}(U)$, the continuous inverse image of a clopen being a clopen (since the statement is true for both open sets and closed sets).

However, we still have to check it is actually a Boolean homomorphism.

For all $U, V \in X$,

- $G(f)(U \cup V) = G(f)(U) \cup G(f)(V)$
- $G(f)(U \cap V) = G(f)(U) \cap G(f)(V)$
- $G(f)(\emptyset) = \emptyset$, $G(f)(X) = Y$

So we have a Boolean homomorphism. Thus, we defined a correspondence:

This correspondence is indeed a contravariant functor:

- $G(Id_Y)$ is the mapping that sends every clopen U of Y on itself, thus it is the same that $Id_{Cl(Y)}$.
- $G(g \circ f) = G(f) \circ G(g)$ simply comes from the fact that $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$.

Defining a functor $F : \mathbf{Bool} \rightarrow \mathbf{Stone}$

Now, starting from a Boolean algebra B , we want to get a Stone space.

First we'll just try to associate a topological space to B , and check if it is Stone or not.

The whole point of showing the Boolean homomorphism between B and a subalgebra of $\mathcal{P}(\text{Ult}(B))$ is that it will be much more convenient to define a topology on a set of subsets than directly on the Boolean algebra.

Thus we want to define a topology on $\text{Ult}(B) \simeq \text{Im}(\phi) = \phi(B)$

We see that $\phi(0) = \emptyset$, $\phi(1) = \text{Ult}(B)$, and $\text{Im}(\phi)$ is closed by finite intersection, and union, which motivates the following choice:

$$\mathcal{T}_B = \{U \subseteq \text{Ult}(B) \mid U \text{ is a union of elements of } \phi(B)\}$$

Finally, we will send a Boolean algebra B to the topological space $(\text{Ult}(B), \mathcal{T}_B)$: we have defined the functor on the objects.

Before looking at the arrows, let's check that this is indeed a Stone space.

First, let's remark that the definition of totally disconnected space X is equivalent to the following: for any $x, y \in X$, there exists a clopen subset $U \subseteq X$ such that $x \in U$ and $y \notin U$.

Now, let's consider $\mathcal{F}, \mathcal{F}'$ in $\text{Ult}(B)$, $\mathcal{F} \neq \mathcal{F}'$, we can take $x \in B$ such that $x \in \mathcal{F}$ and $x \notin \mathcal{F}'$.

Therefore, $\mathcal{F} \in \phi(x)$ and $\mathcal{F}' \notin \phi(x)$ which allows us to conclude since $\phi(x)$ is clopen by definition.

Now, let's check that $\text{Ult}(B)$ is Hausdorff: if $\mathcal{F}, \mathcal{F}' \in \text{Ult}(B)$, since we know that $\text{Ult}(B)$ is totally disconnected, we can consider a clopen $U \in \text{Ult}(B)$ such that $\mathcal{F} \in U$ and $\mathcal{F}' \notin U$.

Since $\text{Ult}(B) \setminus U$ is open, and $\mathcal{F}' \in \text{Ult}(B) \setminus U$, \mathcal{F} and \mathcal{F}' are separated by two clearly disjoint open subset of $\text{Ult}(B)$.

Finally, let's check that $\text{Ult}(B)$ is compact: let's consider \mathcal{U} an open cover of $\text{Ult}(B)$, we want to show that it contains a finite sub-cover of $\text{Ult}(B)$.

By definition of the topology of $\text{Ult}(B)$, an open is an union of $\phi(B)$, thus we can assume that $\mathcal{U} \subseteq \phi(B)$, said in another way: $\mathcal{U} = \{\phi(a) \mid a \in A\}$ for a subset $A \subseteq B$.

Thus: $\text{Ult}(B) = \bigcup \phi(A)$. and we know that $\text{Ult}(B) = \phi(1) = \bigcap \phi(1)$.

Thus: $\bigcup \phi(A) = \bigcap \phi(1)$.

But by applying proposition 1.2.13, we have $A' \subseteq A$ such that $\text{Ult}(B) = \bigcap \phi(1) = \bigcup \phi(A')$, and we have a finite sub cover of $\text{Ult}(B)$: $\mathcal{V} = \{\phi(a) \mid a \in A'\}$.

Now, let's consider a Boolean homomorphism $h : A \rightarrow B$, we want to use it to define a continuous mapping $F(h) : \text{Ult}(B) \rightarrow \text{Ult}(A)$.

Taking $y \in \text{Ult}(B)$, the natural way to get an element in $\text{Ult}(A)$ is by simply taking: $F(h)(y) = h^{-1}(y)$.

However, we still have to check it is actually a continuous mapping, so by the

definition of the topology over $\text{Ult}(A)$, all we have to check is that $F(h)^{-1}(\phi(a))$ is open in $\text{Ult}(B)$, for all $a \in A$.

Let's consider $y \in F(h)^{-1}(\phi(a))$, then $a \in h^{-1}(y)$, so $h(a) \in y$, and finally $y \in (\phi(h(a)))$ which is open in $\text{Ult}(B)$ by the definition of the topology over $\text{Ult}(B)$.

Showing that $F \circ G \simeq Id_{\text{Stone}}$

We need to find a natural transformation ψ such that, for every Stone space X , ψ_X is an homeomorphism and also such that the following diagram commutes

$$\begin{array}{ccc} Y & \xrightarrow{f} & X \\ \downarrow \psi_Y & & \downarrow \psi_X \\ F \circ G(Y) & \xrightarrow{F \circ G(f)} & F \circ G(X) \end{array}$$

Taking Y a Stone space, let's consider the mapping $\psi : \begin{array}{l} Y \rightarrow \text{Ult}(Cl(Y)) \\ x \mapsto \{a \in Cl(Y) \mid x \in a\} \end{array}$.

This mapping is indeed well defined since $\psi(x)$ is an ultrafilter in B .

Furthermore, since Y is Hausdorff compact, all we have left to show that ψ is a homeomorphism is that it is also continuous and bijective.

Let's take $x, y \in Y$ distincts, since Y is totally disconnected, there is a clopen $a \in B$ which contains x but not y .

Thus $a \in \psi(x)$ and $a \notin \psi(y)$, so $\psi(x) \neq \psi(y)$, and thus ψ is injective.

Now let's show the continuity, let's consider an open U in the basis of $\text{Ult}(B)$, by definition of the topology of $\text{Ult}(B)$ we have to show that $\psi^{-1}(\phi(a))$ is open for all $a \in B$.

However: $\psi^{-1}(\phi(a)) = \{x \in Y \mid \psi(x) \in \phi(a)\} = \{x \in Y \mid a \in \psi(x)\} = \{a\}$, which is open so we can conclude.

Finally to prove the surjectivity: suppose there is $x \in \text{Ult}(B)$ such that $x \notin \psi(Y)$.

As $\text{Ult}(B)$ is totally disconnected, there exists for every $y \in \psi(Y)$ a clopen subset V_y of $\text{Ult}(B)$ such that $y \in V_y$ and $x \notin V_y$.

As $\psi(Y)$ is closed, and hence compact, there is an open subcover of $\psi(Y)$.

That is, there exist y_1, y_2, \dots, y_n such that $\psi(Y) \subseteq V = V_{y_1} \cup \dots \cup V_{y_n}$.

As V is a finite union of clopen sets it is again clopen. So there exists an a in B such that $V = \phi(a)$. By definition of V this implies $\psi(Y) \subseteq \phi(a)$ and $x \notin \phi(a)$.

So $Y = \psi^{-1}(\phi(a)) = a$.

But this is in contradiction with the fact that $x \notin \phi(a)$.

Now, for the "natural transformation" part, all we have to check is that:

$$G \circ F(f) \circ \phi(A) = \phi(B) \circ h$$

$$\text{Let's take } a \in A, x \in \text{Ult}(B), y \in G \circ F(h) \circ \phi_A(a) \iff F(h)(y) \in \phi_A(a) \iff h(a) \in y \iff y \in (\phi_B \circ h)(a)$$

Showing that $G \circ F \simeq Id_{\mathcal{B}ool}$

We need to find a natural transformation ϕ such that, for every Boolean algebra B , ϕ_B is a Boolean homomorphism and also such that the following diagram commutes:

$$\begin{array}{ccc} B & \xrightarrow{f} & A \\ \downarrow \phi_B & & \downarrow \phi_A \\ G \circ F(B) & \xrightarrow{G \circ F(f)} & G \circ F(A) \end{array}$$

Let's take ϕ the embedding we constructed in proposition 1.2.14.

We already showed that $\phi(B)$ is exactly constituted by the clopen subsets of $Ult(B)$.

But, since ϕ is a Boolean algebra embedding: $B \simeq \phi(B) \simeq Cl(Ult(B)) = G \circ F(B)$.

The computation to prove that we obtained a natural transformation is similar to the one we performed in the previous paragraph.

□

1.3 Profiniteness and duality

The main purpose of this section is twofold: first introducing the terminology related to pro-categories, second describing the category of Stone spaces introduced in the previous section as a pro-completion of the category of finite sets.

This will be useful because it will allow us to perform some calculations relative to Stone spaces by doing them on finite sets and then "taking the limit".

Definition 1.3.1.

Let I be a category, we say that I is cofiltered if:

- I is not empty
- For all objects i, i' in I , there exists an object i'' in I and two arrows $f : i'' \rightarrow i, f' : i'' \rightarrow i'$ in I
- For all arrows $u, v : i' \rightarrow i$, there exists an object i'' in I and an arrow $w : i'' \rightarrow i'$ such that $u \circ w = v \circ w$.

Definition 1.3.2.

Let I and \mathcal{C} be categories, I being a cofiltered category.
A cofiltered limit is a limit of a functor $F : I \rightarrow \mathcal{C}$.

Definition 1.3.3.

Let \mathcal{C} be a category, a pro-object of \mathcal{C} is a formal cofiltered limit of objects in \mathcal{C} .
If \mathcal{D} and \mathcal{E} are two small cofiltered categories, and $F : \mathcal{D} \rightarrow \mathcal{C}$, $G : \mathcal{E} \rightarrow \mathcal{C}$ two diagrams in \mathcal{C} , then we define the set of arrows as:

$$\text{Hom}_{\text{Pro-}\mathcal{C}}(F, G) = \varprojlim_{e \in \mathcal{E}} \varinjlim_{d \in \mathcal{D}} \text{Hom}_{\mathcal{C}}(F(d), G(e))$$

In a dual way, one defines an Ind-object of a category \mathcal{C} as a formal filtered colimit of objects in \mathcal{C} , and similarly we will have a category $\text{Ind-}\mathcal{C}$.

Before stepping into the main fact of this section, let's pass through a few digressions relative to algebraic theories:

Definition 1.3.4.

- We define a finitary operation over a set A as an application $op : A^k \rightarrow A$ for some $k \in \mathbb{N}$.
- Many of the concrete categories we have to deal with have in common that the "structure" given to the carrier sets consists of a number of finitary operations which are required to satisfy a number of equational conditions (commutative and distributive laws).
Such categories are called finitary algebraic categories.
- Let \mathcal{A} be a category, assuming that we have a free functor $F : \text{Set} \rightarrow \mathcal{A}$, giving a presentation of an object A in \mathcal{A} by generators and relations means to give a coequalizer diagram: $FR \rightrightarrows FX \longrightarrow A$ in \mathcal{A} , where X is the set of generators and R the set of relations.
We say that A is finitely-presented if we can find such a presentation with both X and R finite; and we write \mathcal{A}_{fp} for the full subcategory of finitely presented algebras in \mathcal{A} .
- We also write \mathcal{A}_{fin} for the full subcategory of finite algebras in \mathcal{A} .
If the theory of \mathcal{A} is finitely-generated (we mean that the definition of an object of \mathcal{A} involves only a finite number of finitary operations), then we clearly have $\mathcal{A}_{fin} \subseteq \mathcal{A}_{fp}$.
The converse inclusion $\mathcal{A}_{fp} \subseteq \mathcal{A}_{fin}$ holds if we know that the finitely generated free algebras in \mathcal{A} are finite.

Example 1.3.5.

If a Boolean algebra is generated (as a Boolean algebra) by a set X , then it is generated as a distributive lattice by the elements of X and their complements. It is shown in [10] that finitely-generated free Boolean algebras are finite: so by definition we have $\mathcal{B}ool_{fp} = \mathcal{B}ool_{fin}$.

One could believe that this extremely formal definition would only lead to some very abstract result, but we will see in the next proposition and the related examples that a few pro-completions can actually be seen as concrete categories. The interested reader can find the slightly technical proof of the following fact in [10] for instance:

Proposition 1.3.6.

Let \mathcal{A} be a finitary algebraic category.

Then we have the categorical equivalence: $Ind\text{-}\mathcal{A}_{fp} \simeq \mathcal{A}$

Example 1.3.7.

One can check in the definition of $\mathcal{B}ool$ the obvious fact that the structure on the underlying sets of the objects on $\mathcal{B}ool$ consists of a number of finitary operations satisfying equational conditions.

Thus it is a finitary algebraic category and therefore we have:

$$Ind\text{-}\mathcal{B}ool_{fp} \simeq \mathcal{B}ool.$$

Now, we have everything in hand to prove the main theorem of this section.

Proposition 1.3.8.

The category *Stone* of Stone spaces and continuous maps is equivalent to the pro-completion of the category Set_{fin} of finite sets.

Proof.

By Stone duality, we know that: $Stone \simeq \mathcal{B}ool^{op}$.

However, we could check in 1.3.7 that $\mathcal{B}ool \simeq Ind\text{-}\mathcal{B}ool_{fp}$.

As we checked in 1.3.5, $Ind\text{-}\mathcal{B}ool_{fp} \simeq Ind\text{-}\mathcal{B}ool_{fin}$, since finitely generated free Boolean algebras are finite, so: $Stone \simeq (Ind\text{-}\mathcal{B}ool_{fin})^{op}$

But, by definition of the Ind and Pro categories, we have:

$$(Ind\text{-}\mathcal{B}ool_{fin})^{op} \simeq Pro\text{-}(\mathcal{B}ool_{fin})^{op}$$

Finally, restricting Stone duality to finite Stone spaces and finite sets gives us $Set_{fin} \simeq (\mathcal{B}ool_{fin})^{op}$, and finally we proved that $Stone \simeq Pro\text{-}Set_{fin}$. \square

Remark 1.3.9.

To close this section, let's note that profinite methods have other applications, for instance one can use them as a way to extend the finite Stone duality on $\mathcal{S}et_{fin}$ to the one on $\mathcal{S}et$.

The idea to remember is that, since usually dualities are easier to establish for small categories with finite structure, a good way to generalize them through large categories is to consider profinite objects (if the large category is the pro-category of some suitable smaller category).

Chapter 2

Constructing the profinite monad \overline{T}

Contents

2.1	Ideas and motivations	33
2.2	An algebraic point of view on Stone duality on $\text{Rec}(A)$	34
2.3	Categorical constructions of the profinite monad	40

2.1 Ideas and motivations

Considering a monad T over Set , our purpose will be to construct a "profinite" monad, $\overline{T} : \text{Set} \rightarrow \text{Set}$, associated to T .

The terminology "profinite" arised from the fact that this construction appears naturally as the categorical generalization of the profinite monoid.

An important property of this monad \overline{T} will be that it "extends" T in the sense that it extends some of its properties.

This construction has already been discussed in [1], but we will be going through all the details that were not explicitly presented in this paper.

Furthermore, writing consciously the categorical setting necessary to define \overline{T} will allow us to see this functor as a Kan extension, which will be exploited in the last part of the paper.

2.2 An algebraic point of view on Stone duality on $\text{Rec}(A)$

In order to construct the profinite monad \bar{T} , we will need to resort to the tools introduced in the first chapter about Stone duality.

However, such an approach might look tricky at first sight : since, eventually, we expect to obtain a new monad structure, we would have expected to do something more "diagrammatic".

For this reason, we will look at Stone duality from a more algebraic point of view, using a new terminology (T-morphisms type instead of the Stone dual), equivalent to Stone duality in our case, and which will happen to be much more convenient for categorical construction.

Before going through the details, let's reformulate the definition of recognition on a monad on Set :

Definition 2.2.1.

Let $T : \text{Set} \rightarrow \text{Set}$ be a monad on Set , and A a T-algebra over T .

Then, according to the more general definition given in chapter 1, a subset $P \subseteq A$ is T-recognizable if and only if there is a finite T-algebra B , a T-morphism $\phi : A \rightarrow B$ and a subset $Q \subseteq B$ such that: $P = \phi^{-1}(Q)$.

Thus, we define the set of subsets recognizable in A :

$$\text{Rec}(A) = \{ P \subseteq A \mid \exists B \text{ a finite T-algebra, } \exists \phi : A \rightarrow B \text{ a T-algebra morphism, } \exists Q \subseteq B, \text{ such that: } P = \phi^{-1}(Q) \}$$

An important thing to notice is that, since we don't consider a monad over any category but specifically on Set , T-algebras are way easier to manipulate.

Thus, if we consider the usual operations over Set (union, intersection, complementary), we will be able to see $\text{Rec}(A)$ as a Boolean algebra.

Lemma 2.2.2.

Let $T : \text{Set} \rightarrow \text{Set}$ be a monad on Set , and A a T-algebra over T .

Then, $(\text{Rec}(A), \cap, \cup, {}^c)$ is a Boolean algebra.

Proof.

Since $\text{Rec}(A) \subseteq \mathcal{P}(A)$ and it is non-empty (since $\emptyset \in \text{Rec}(A)$), to show that $\text{Rec}(A)$ is a Boolean algebra, all we have to check is that it's closed under the Boolean operations.

Let's consider $P, P' \subseteq \text{Rec}(A)$, then by applying the definition of $\text{Rec}(A)$, we have N, N' finite T-algebras, $\phi : A \rightarrow N, \phi' : A \rightarrow N'$ T-algebra morphisms, and finally $Q \subseteq N, Q' \subseteq N'$ such that: $P = \phi^{-1}(Q)$ and $P' = \phi'^{-1}(Q')$.

Let's look at $\phi'' : A \rightarrow N \times N'$ ($N \times N'$ being a finite set) defined for all x in

2.2. AN ALGEBRAIC POINT OF VIEW ON STONEDUALITY ON $\text{Rec}(A)$ 35

A by : $\phi''(x) = (\phi(x), \phi'(x))$.

Then, we have: $\phi''^{-1}(Q \cap Q') = P \cap P'$, thus $P \cap P' \in \text{Rec}(A)$.

Now, let's remark that $\phi^{-1}(N \setminus Q) = M \setminus P = {}^c P$, thus ${}^c P \in \text{Rec}(A)$.

Finally, since $P \cup P' = A \setminus ((A \setminus P) \cap (A \setminus P'))$, we have $P \cup P' \in \text{Rec}(A)$, and we can conclude. \square

Therefore, we can now define the set of ultrafilters of $\text{Rec}(A)$, which will simply be called $\text{Ult}(\text{Rec}(A))$.

Definition 2.2.3.

Let $T : \text{Set} \rightarrow \text{Set}$ be a monad on Set and A a T -algebra over T .

- To keep more practical notations, we define the set :

$$H_{fin}(A) = \{h : A \rightarrow B \mid B \text{ is a finite } T\text{-algebra and } h \text{ is a surjective } T\text{-morphism}\}$$

- We now define the T -morphisms type we mentioned in the introduction :

A T -morphism type over A is a map τ that provides, for each

$(h : A \rightarrow B)$ in $H_{fin}(A)$ an element $\tau(h) \in B$ satisfying the following compatibility condition :

$\forall h \in H_{fin}(A), \forall g : B \rightarrow C$ surjective T -morphism between finite T -algebras : $\tau(g \circ h) = g(\tau(h))$

$$\begin{array}{ccc} A & \xrightarrow{h} & B \\ & \searrow & \downarrow g \\ & g \circ h & B' \end{array}$$

- Finally, we define the set $\overline{A} = \{\tau \mid \tau \text{ is a } T\text{-morphism type over } A\}$.

Now, we'll see how this algebraic setting can help to keep a more algebraic point of view upon the set of ultrafilters over $\text{Rec}(A)$:

Proposition 2.2.4.

Let $T : \text{Set} \rightarrow \text{Set}$ be a monad, and A be a T -algebra.

Then we can define topologies on \overline{A} and $\text{Ult}(\text{Rec}(A))$, such that the two topological spaces obtained afterwards are homeomorphic spaces.

We will often refer to \overline{A} as the compactification of A .

Proof.

Defining a mapping from \bar{A} to $\text{Ult}(\text{Rec}(A))$

Let's take an element $\tau \in \bar{A}$, we want to send it on an element of $\text{Ult}(\text{Rec}(A))$.
Let's define for every $h \in H_{fin}(A)$,

$$L_{h,\tau} = h^{-1}(\tau(h))$$

(Indeed, $L_{h,\tau} \in \text{Rec}(A)$ since the subset $P = \{\tau(h)\}$ in the finite T-algebra B is such that $L_{h,\tau} = h^{-1}(\tau(h))$.)

Now that we have an element of $\text{Rec}(A)$, a good candidate to be an ultrafilter of $\text{Rec}(A)$ appears to be :

$$\Upsilon_\tau = \uparrow \{L_{h,\tau} \mid h \in H_{fin}(A)\} \quad (2.1)$$

$$= \{K \subseteq A \mid \exists h \in H_{fin}(A), L_{h,\tau} \subseteq K\} \quad (2.2)$$

$$= \{K \subseteq A \mid \exists h \in H_{fin}(A), h^{-1}(\tau(h)) \subseteq K\} \quad (2.3)$$

$$\subseteq \mathcal{P}(\text{Rec}(A)) \quad (2.4)$$

Now, we only have to check that the previous object is actually an ultrafilter :

- $\emptyset \notin \Upsilon_\tau$: Let's assume that $\emptyset \in \Upsilon_\tau$, then : $\exists h \in H_{fin}(A), h^{-1}(\tau(h)) \subseteq \emptyset$, i.e $h^{-1}(\tau(h)) = \emptyset$, which is absurd since h is surjective.
- Let's take $K' \in \mathcal{P}(\text{Rec}(A))$, $K \in \Upsilon_\tau$ (then : $\exists h \in H_{fin}(A), L_{h,\tau} \subseteq K$) and let's assume that $K \subseteq K'$, then $L_{h,\tau} \subseteq K \subseteq K' \subseteq A$, thus $K' \in \Upsilon_\tau$.
- If $L \in \text{Rec}(A)$, we want to show that $\exists! Y \in \{L; \bar{L}\}$ such that $Y \in \Upsilon_\tau$. Since $L \in \text{Rec}(A)$, we have $h : A \rightarrow B$ and $P \subseteq B$ such that $L = h^{-1}(P)$. Let's assume that $\tau(h) \in P$, then we have $L_{h,\tau} \subseteq L$, and therefore by definition : $L \in \Upsilon_\tau$.
On the other hand, let's assume that $\tau(h) \notin P$, then $\tau(h) \in \bar{P}$, but we also have: $h^{-1}(B \setminus P) = A \setminus h^{-1}(P) = A \setminus L$, so $L_{h,\tau} \subseteq \bar{L}$, thus $\bar{L} \in \Upsilon_\tau$.
- Finally, we have to show the stability for intersection, and it will be sufficient to do it on $\{L_{h,\tau} \mid h \in H_{fin}(A)\}$ since we stated that $\Upsilon_\tau = \uparrow \{L_{h,\tau}\}$. What we want to show is that: $\forall h, h' \in H_{fin}(A), \exists h'' \in H_{fin}(A)$ such that: $L_{h,\tau} \cap L_{h',\tau} = L_{h'',\tau}$, ie : $h^{-1}(\tau(h)) \cap h'^{-1}(\tau(h')) = h''^{-1}(\tau(h''))$. Let's take $a \in h^{-1}(\tau(h)) \cap h'^{-1}(\tau(h'))$, then we have $h(a) = \tau(h)$ and $h'(a) = \tau(h')$, so in particular, we understand that to satisfy the equality of sets we wanted, the mapping $h'' : A \rightarrow B''$ has to satisfy the condition:

$$h''(a) = \tau(h'')$$

$$\text{Therefore, we want to take : } \left. \begin{array}{l} A \longrightarrow B'' = B \times B'' \\ a \longmapsto (h(a), h'(a)) = h''(a) \end{array} \right\}$$

(which is in $H_{fin}(A)$, because $B \times B''$ is finite)

2.2. AN ALGEBRAIC POINT OF VIEW ON STONEDUALITY ON $\text{REC}(A)$ 37

That way, $h''(a) = (h(a), h'(a)) = (\tau(h), \tau(h'))$ and all we need to show to conclude is that: $(\tau(h), \tau(h')) = \tau(h'')$.

But, since $B \times B'$ is finite, we have two compatibility diagrams satisfying the conditions in the definition of τ :

$$\begin{array}{ccc} A & \xrightarrow{h''} & B \times B' \\ & \searrow h & \downarrow \pi \\ & & B \end{array} \quad \begin{array}{ccc} A & \xrightarrow{h''} & B \times B' \\ & \searrow h & \downarrow \pi' \\ & & B' \end{array}$$

Thus $\pi \circ \tau(h'') = \tau(h)$ and $\pi \circ \tau(h'') = \tau(h')$: so $\tau(h'') = (\tau(h), \tau(h'))$.
We finally have shown that we have defined an ultrafilter on $\text{Rec}(A)$.

Defining a mapping from $\text{Ult}(\text{Rec}(A))$ to \overline{A}

Let's take an ultrafilter of $\text{Rec}(A)$, let's say U , we want to define an element in \overline{A} : $\Delta U = \tau_U : \left| \begin{array}{ccc} H_{\text{fin}(A)} & \longrightarrow & B \\ h : A \mapsto B & \longmapsto & \tau_U(h) \end{array} \right.$

In the first place, we will need several technical lemmas :

Lemma 2.2.5.

For at most one element $b \in B$, we have $h^{-1}\{b\} \in U$

Proof.

Let's assume that: $\exists b, b' \in B, b \neq b'$ such that: $h^{-1}\{b\} \in U$ and $h^{-1}\{b'\} \in U$. Then, their intersection $h^{-1}\{b\} \cap h^{-1}\{b'\}$ is in U .

But, since $b \neq b'$, this intersection is \emptyset , and therefore $\emptyset \in U$ which is a contradiction since U is a filter. \square

Lemma 2.2.6.

There exist $b \in B$ such that $h^{-1}\{b\} \in U$

Proof.

Let's assume that: $\forall b \in B, h^{-1}\{b\} \notin U$. Then, $\forall b \in B, {}^c h^{-1}\{b\} \in U$.

Since B is finite, $\bigcap_{b \in B} ({}^c h^{-1}\{b\}) \in U$. Therefore, ${}^c \bigcup_{b \in B} h^{-1}\{b\} \in U$.

However, since we have: $\bigcup_{b \in B} h^{-1}\{b\} = A$, we conclude that ${}^c A \in U$, and by taking the complementary: $\emptyset \in U$, which is a contradiction, since U is a filter. \square

Proposition 2.2.7.

$\exists! b \in B$ satisfying : $h^{-1}\{b\} \in U$ (of course, it will be our choice for $\tau_U(h)$).

Proof.

An obvious conjunction of the two previous lemmas. \square

We still have to show that this choice of an element in B makes $\tau_U(h)$ an element of \overline{A} , said in another way we want to check that:

$$\forall h \in H_{fin}(A), \forall g \in H_{fin}(B), \tau_U(g \circ h) = g(\tau_U(h))$$

$$\begin{array}{ccc} A & \xrightarrow{h} & B \\ & \searrow & \downarrow g \\ & & C \end{array}$$

By definition, $\tau_U(g \circ h)$ is the only element $c \in C$ such that $(g \circ h)^{-1}\{c\} \in U$, and the same way we have that $\tau_U(h)$ is the only $b \in B$ such that $h^{-1}\{b\} \in U$. Therefore, all we have left to show is that $g(b)$ is an element of C satisfying the property : $(g \circ h)^{-1}\{g(b)\} \in U$.

Hence, $(g \circ h)^{-1}g(b) = (h^{-1} \circ g^{-1})\{g(b)\} = h^{-1}(\{d \in B \mid g(d) = g(b)\})$

But since $b \in g^{-1}\{g(b)\}$, $\{d \in B \mid g(d) = g(b)\} \neq \emptyset$ and we have:

$$h^{-1}\{b\} \subseteq (h^{-1} \circ g^{-1})\{g(b)\}.$$

Finally, since $h^{-1}\{b\} \in U$, we have: $(h^{-1} \circ g^{-1})\{g(b)\} \in U$ by the second property in the definition of a filter : this concludes the proof.

Showing that both constructions are mutual inverses

Let's summarize :

- $\Upsilon_\tau = \{K \subseteq A \mid \exists h \in H_{fin}(A), h^{-1}(\tau(h)) \subseteq K\}$

- $\Delta U = \tau_U : \begin{array}{ccc} H_{fin}(A) & \longrightarrow & B \\ h : A \mapsto B & \longmapsto & \tau_U(h) \end{array}$

Let's see if the previous mappings are indeed inverses :

2.2. AN ALGEBRAIC POINT OF VIEW ON STONEDUALITY ON $REC(A)$ 39

- $\forall \tau \in \overline{A}, (\Delta \circ \Upsilon)(\tau) = \Delta(\{K \subseteq A \mid \exists h \in H_{fin}(A), h^{-1}(\tau(h)) \subseteq K\}) = \Delta(V)$ which is the mapping $\tau_V = \tau_U : \begin{array}{ccc} H_{fin}(A) & \longrightarrow & B \\ h : A \mapsto B & \longmapsto & b \end{array}$ where b is the only element such that $h^{-1}(\{b\}) \in V$, which means that $h^{-1}(\tau(h)) \subseteq h^{-1}(\{b\})$, and thus $\tau(h) = b$.
So we get the mapping we had in the first place, as desired.
- $\forall U \in Ult(Rec(A)),$
 $\Upsilon \circ \Delta(U) = \Upsilon_{\tau_U} = \{K \subseteq A \mid \exists h \in H_{fin}(A), h^{-1}(\tau_U(h)) \subseteq K\}$
 $= \{K \subseteq A \mid \exists h \in H_{fin}(A), h^{-1}(b) \subseteq K\}$
where b is the only element of B such that $h^{-1}(\{b\}) \in U$.
Therefore, by the same argument we used at the end of the previous part, this is the ultrafilter $U : \Upsilon_{\tau_U} = U$.

Giving a topology to both $Ult(Rec(A))$ and \overline{A}

- For $Ult(Rec(A))$, we can take the usual topology for ultrafilter spaces, given by the following basis of opens :
Let L be in $Rec(A)$, we take : $\overline{L} = \{U \mid U \text{ is an ultrafilter containing } L\}$.
Then, $(\overline{L})_{L \in Rec(A)}$ is a basis of open in $Rec(A)$.
- For \overline{A} , if $b \in B$, then the topology is given by the basis of opens :
 $\{\tau \in \overline{A} \mid \tau(h) = b\}$.

Showing that we have an homeomorphism

To conclude the proof, we need to show that the inverse image of an open of each space by each mappings previously defined is still an open in the other space.

- First, let's see where Υ sends an open in the basis of open of \overline{A} ,
 $\{\tau \in \overline{A} \mid \tau(h) = b\}$, for h and b fixed.
Let $\tau \in \overline{A}$ be such that $\tau(h) = b$, then :
 $\Upsilon_{\tau} = \{K \subseteq A \mid \exists h' : A \mapsto B, B \text{ finite, such that : } h'^{-1}\{\tau(h)\} \subseteq K\}$
 $= \{K \subseteq A \mid \exists h : A \mapsto B, B \text{ finite, such that : } h^{-1}\{b\} \subseteq K\}$
which is a set of ultrafilters containing $h^{-1}(\{b\})$, and therefore an open in the basis of opens of $Ult(Rec(A))$.
- Now, we see where Δ sends an open in the basis of open of $Ult(Rec(A))$.
Let's consider $\overline{L} = \{U \mid U \text{ is an ultrafilter containing } L\}$, then if $U \in \overline{L}$, we have:
 $\Delta(U) = \tau_U : \begin{array}{ccc} H_{fin}(A) & \longrightarrow & B \\ h : A \mapsto B & \longmapsto & b \end{array}$
where b is the only element in B such that $h^{-1}(B) \in U$ and we can conclude since it's the definition of an element in the open basis of \overline{A} we chose.

□

Let's close this section by defining mapping associated to the compactification of T-algebras we just defined :

Definition 2.2.8.

Let $h : A \rightarrow B$ be in $H_{fin}(A)$, then, the profinite extension of h is

$$\overline{h} : \begin{array}{l} \overline{A} \longrightarrow B \\ \tau \longmapsto \tau(h) \end{array}$$

2.3 Categorical constructions of the profinite monad

From now on, we will use the set of T-morphisms types instead of the Stone dual because it will happen to be more adapted to the construction of the profinite monad presented in [1].

Action of \overline{T} over the objects :

Let's consider a monad T on \mathbf{Set} . In [1], the profinite monad \overline{T} is a functor over \mathbf{Set} defined on the objects in the following way : if Σ is a set, then, we saw in the first chapter that $T\Sigma$ is a T-algebra. Thus, according to the previous section, we can consider its compactification $\overline{T\Sigma}$, and this will be the set we will chose : $\overline{T\Sigma} = \overline{T\Sigma}$

Before defining the functor on arrows, let's take a moment to look at the set we just defined : $\overline{T\Sigma} = \{\tau \mid \tau \text{ is a T-morphism type on } T\Sigma\}$. This set can actually be seen as a limit of a functor over \mathbf{Set} and we will see how right now.

Let's fix a T-algebra A , then we can define the category $H_{fin}(A)$ in the following way :

- An object $h : A \rightarrow B$ of $H_{fin}(A)$ is a surjective T-morphism from A into a finite T-algebra, let's call it B
- Taking two objects $h : A \rightarrow B$, $h' : A \rightarrow B'$ an arrow $\phi : h \rightarrow h'$ is just a

T-algebra morphism $\phi : B \rightarrow B'$ such that

$$\begin{array}{ccc} A & \xrightarrow{h} & B \\ & \searrow h' & \downarrow \phi \\ & & B' \end{array} \text{ commutes}$$

2.3. CATEGORICAL CONSTRUCTIONS OF THE PROFINITE MONAD 41

This category is essentially small, since the collection of objects is a set (to be exact, a set of mappings).

Before expressing the limit we talked about, let's introduce 3 functors that we will use :

- $G : \begin{array}{ccc} H_{fin}(A) & \longrightarrow & \mathcal{S}et_{fin}^T \\ h : A \rightarrow B & \longmapsto & B \end{array}$ if it's not
 $(h : A \rightarrow B \xrightarrow{f} h' : A \rightarrow B') \longmapsto (B \xrightarrow{G(f)} B')$
 already clear, it sends an arrow $f : h \rightarrow h'$, which can be written $f : B \rightarrow B'$ to itself.
- $U : \mathcal{S}et_{fin}^T \rightarrow \mathcal{S}et_{fin}$ the forgetful functor that was defined in chapter 1, taking $\mathcal{C} = \mathcal{S}et_{fin}$
- $\iota : \mathcal{S}et_{fin} \rightarrow \mathcal{S}et$ the "canonical injection" functor which is only seeing a finite set as a set.

Now, we can define the functor : $\iota \circ U \circ G : H_{fin}(A) \rightarrow \mathcal{S}et$.

More generally, if I is a small category and $F : I \rightarrow \mathcal{S}et$ a functor, we know that the limit of F can be expressed as :

$$\lim_I F = \{(x_i)_{i \in I} \in \prod_{i \in I} F(i) \mid \forall \phi : i \rightarrow j, F(\phi) : F(i) \rightarrow F(j), \text{ we have: } F(\phi)(x_i) = (x_j)\}$$

Thus, taking $I = H_{fin}(A)$ and $F = \iota \circ U \circ G$, we have:

$$\lim_{H_{fin}(A)} \iota \circ U \circ G = \{(\tau_h)_{h \in H_{fin}(A)} \in \prod_{h \in H_{fin}(A)} F(h) \mid \forall \phi : B \rightarrow B', \text{ we have: } \phi(\tau_h) = \tau'_h\}$$

which is exactly the definition of \overline{A} given in the previous section.

Now that we saw that this functor is a limit, let's look at what it does on arrows :

Action of \overline{T} over the arrows :

Let Σ, Σ' be two sets, and let's consider an arrow $f : \Sigma \rightarrow \Sigma'$.

By definition of the monad T , it induces an arrow $Tf : T\Sigma \rightarrow T\Sigma'$: we wish to use it to obtain the arrow required for $\overline{T}, \overline{T}f : \overline{T\Sigma} \rightarrow \overline{T\Sigma'}$.

Thanks to the previous section, we now can express $\overline{T\Sigma'}$ as a limit :

$$\overline{T\Sigma'} = \lim_{H_{fin}(T\Sigma')} \iota \circ U \circ G$$

The indexing category is $H_{fin}(T\Sigma')$, so the full data of the limit consists of the object $T\Sigma'$ and a family of morphisms $(p_h : \overline{T\Sigma'} \rightarrow \iota \circ U \circ G(h))_{h : T\Sigma' \rightarrow B}$ in $\mathcal{S}et$,

where B is a finite T -algebra and h a surjective T -morphism.

On the other hand, composing morphisms $h : T\Sigma' \rightarrow B$ with Tf gives us a family of morphisms $(h \circ Tf : T\Sigma \rightarrow B)_{h:T\Sigma' \rightarrow B}$.

But now, by using profinite extensions defined in 2.2.8, this family of morphisms gives another family of morphisms $(Tf \circ \bar{h} : \bar{T}\Sigma \rightarrow B)_{h:T\Sigma' \rightarrow B}$: thus, by applying the universal property of the limit $\bar{T}\Sigma'$, we get a unique morphism $\bar{T}\Sigma \rightarrow \bar{T}\Sigma'$ in Set : this will be our choice for $\bar{T}f$.

$$\begin{array}{ccc} \bar{T}\Sigma & \xrightarrow{\bar{T}f} & \bar{T}\Sigma' \\ & \searrow \bar{h} \circ Tf & \swarrow p_h \\ & & B \end{array}$$

μ and η to make it a monad :

Let's explain how you can get the natural transformations $\bar{\eta}$ and $\bar{\mu}$. (one can check that these natural transformations satisfies the property of a monad, as it was thoroughly performed in [1]).

- In the same way we did previously, we consider a set Σ and look at the limit $\bar{T}\Sigma$ and the family of morphisms $(p_h : \bar{T}\Sigma \rightarrow \iota \circ U \circ G(h))_{h:T\Sigma \rightarrow B}$ in Set .

On the other hand, since T is a monad, we have the morphism

$\eta_\Sigma : \Sigma \rightarrow T\Sigma$, which can be composed with $h : T\Sigma \rightarrow B$ in $H_{fin}(T\Sigma)$ to obtain a family of morphisms $(h \circ \eta_\Sigma : \Sigma \rightarrow B)_{h:T\Sigma \rightarrow B}$.

Now, by applying the universal property of the limit $\bar{T}\Sigma$, we obtain a unique morphism in Set : $\Sigma \rightarrow \bar{T}\Sigma$ which will be used as the definition of $\bar{\eta}_\Sigma$.

$$\begin{array}{ccc} \Sigma & \xrightarrow{\bar{\eta}_\Sigma} & \bar{T}\Sigma \\ \downarrow \eta_\Sigma & & \downarrow p_h \\ T\Sigma & \xrightarrow{h} & B \end{array}$$

- In the same way we did previously, we consider a set Σ and look at the limit $\bar{T}\Sigma$ and the family of morphisms $(p_h : \bar{T}\Sigma \rightarrow \iota \circ U \circ G(h))_{h:T\Sigma \rightarrow B}$ in Set .

On the other hand, since B is a T -algebra we have a morphism

$mult_B : TB \rightarrow B$, and by using the profinite extension defined in 2.2.8, we get a morphism $\overline{mult}_B : \bar{T}B \rightarrow B$.

Let's also remark that, applying the functor \bar{T} to the arrows p_h , we obtain a family of arrows $\bar{T}p_h : \bar{T}T\Sigma \rightarrow \bar{T}B$: composing it with \overline{mult}_B gives a family of morphisms $(\overline{mult}_B \circ \bar{T}p_h : \bar{T}T\Sigma \rightarrow B)$. Now, by applying the

2.3. CATEGORICAL CONSTRUCTIONS OF THE PROFINITE MONAD 43

universal property of the limit $\overline{T\Sigma}$, we obtain a unique morphism in \mathcal{Set} : $\overline{TT\Sigma} \rightarrow \overline{T\Sigma}$ which will be used as the definition of $\overline{\mu_\Sigma}$

$$\begin{array}{ccc} \overline{TT\Sigma} & \xrightarrow{\overline{\mu_\Sigma}} & \overline{T\Sigma} \\ \downarrow \overline{Tp_h} & & \downarrow \overline{h} \\ \overline{TB} & \xrightarrow{\overline{mult_B}} & B \end{array}$$

Remark 2.3.1.

As a transition between this chapter and the following, let's anticipate a bit and consider another way to see the limit defining \overline{T} .

We know that, for any set Σ :

$$\overline{T\Sigma} = \lim_{H_{fin}(T\Sigma)} \iota \circ U \circ G$$

But, looking at the object and arrows of $H_{fin}(T\Sigma)$, we already used the fact that it can be seen as a comma category:

$$\overline{T\Sigma} = \lim_{h:F\Sigma \rightarrow B} \iota \circ U \circ G$$

(where F is adjoint to $\iota \circ U$)

Now, since we can get any surjective morphism in the indexing category by restricting ourselves to good mappings, we can say it is equal to:

$$\lim_{h:F\Sigma \rightarrow B} i \circ U \circ G$$

Finally, by adjunction, this limit is equal to:

$$\lim_{\overline{h}:\Sigma \rightarrow (\iota \circ U)(B)} \iota \circ U \circ Q^B$$

(where Q^B will be defined properly in the next chapter)

That last expression will exactly be the expression, as a pointwise right Kan extension, of $Ran_{\iota \circ U}(\iota \circ U)$

Chapter 3

Relating \overline{T} to a monad on the category of Stone spaces

Contents

3.1	Ideas and motivation	45
3.2	Kan extensions	46
3.3	\overline{T} with Kan extensions and \tilde{T}	54

3.1 Ideas and motivation

We will propose an alternative point of view upon the profinite monad previously introduced by providing an equivalent expression, different from the one presented by Bojanczyk in [1].

This construction has the merit to be more natural as a categorical construction in the following sense: Bojanczyk's derives \overline{T} from T by endowing the objects, arrows, η and μ with Stone compactification.

It does indeed give the profinite monad, but one doesn't get much information about the functors on the categories involved.

On the other hand, our construction will eventually lead us to dress a diagram expressing \overline{T} as a composition of relatively well-known functors.

3.2 Kan extensions

To express \bar{T} the way we want, we need to construct several functors by using right Kan extensions; therefore we will present in this section a few results related to this notion.

First, let's remind the reader that you can compose natural transformations with functors in two ways (right or left), we'll quickly describe the one we need in our case (the other case is easy to deduce by constructing the diagrams, applying a functor and taking the "natural" arrows induced):

Definition 3.2.1.

Let \mathcal{B} , \mathcal{C} , \mathcal{D} be categories, $H : \mathcal{B} \rightarrow \mathcal{C}$, $F : \mathcal{C} \rightarrow \mathcal{D}$, $F' : \mathcal{C} \rightarrow \mathcal{D}$ be functors and $h : F \Rightarrow F'$ be a natural transformation.

$$\mathcal{B} \xrightarrow{H} \mathcal{C} \xrightarrow[\begin{array}{c} \Downarrow h \\ F' \end{array}]{F} \mathcal{D}$$

Then, we can define a natural transformation $hH : F \circ H \Rightarrow F' \circ H$ in the following way:

For every object A and B in \mathcal{C} , and for every arrow $f : A \rightarrow B$, since h is a natural transformation, we have the commutative diagram:

$$\begin{array}{ccc} F(A) & \xrightarrow{h_A} & F'(A) \\ \downarrow F(f) & & \downarrow F'(f) \\ F(B) & \xrightarrow{h_B} & F'(B) \end{array}$$

In particular, this diagram can be written for any arrow $H(f) : H(X) \rightarrow H(Y)$ in \mathcal{C} (where $f : X \rightarrow Y$ is an arrow in \mathcal{B}), so now we get:

$$\begin{array}{ccc} F \circ H(X) & \xrightarrow{(hH)_X} & F' \circ H(X) \\ \downarrow F \circ H(f) & & \downarrow F' \circ H(f) \\ F \circ H(Y) & \xrightarrow{(hH)_Y} & F' \circ H(Y) \end{array}$$

which clearly defines a natural transformation $hH : F \circ H \rightarrow F' \circ H$.

(Note that, in the rest of the text, we will often write FH instead of $F \circ H$ for the composition.)

Now, we can introduce Kan extensions properly, the motivation for this notions appears clearly in the diagram of the definition:

Definition 3.2.2.

Let $\mathcal{C}, \mathcal{D}, \mathcal{E}$ be categories, and $F : \mathcal{C} \rightarrow \mathcal{E}, K : \mathcal{C} \rightarrow \mathcal{D}$ functors.

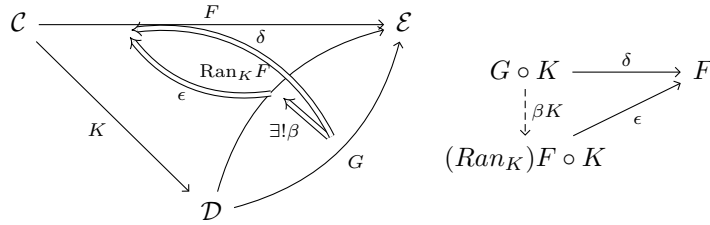
A right Kan extension of F along K is a functor $Ran_K F : \mathcal{D} \rightarrow \mathcal{E}$ together with a natural transformation $\epsilon : (Ran_K F) \circ K \rightarrow F$ satisfying the following universal property:

For every pair (G, δ) where $G : \mathcal{D} \rightarrow \mathcal{E}$ is a functor and $\delta : G \circ K \Rightarrow F$ is a natural transformation, there is a unique natural transformation

$$\beta : G \Rightarrow (Ran_K F)$$

which factorizes through ϵ , i.e.:

$$\delta = \epsilon \circ \beta K$$

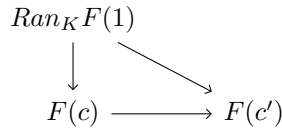


(Here we defined right Kan extensions, but we could define left Kan extension in a dual way.)

The notion of Kan extension is an universal construction which generalizes a lot of categorical notions in the sense that they can be seen as particular cases of Kan extensions.

Example 3.2.3.

Let's consider $\mathcal{D} = 1$ the category with only one object and one arrow. Then, there is obviously only one functor $K : \mathcal{C} \rightarrow 1$. In that case, a right Kan extension of F along K is a functor $Ran_K(F) : 1 \rightarrow \mathcal{E}$ with a natural transformation $\epsilon : Ran_K(F) \circ K \Rightarrow F$. But in this particular case, since K sends any object c in \mathcal{C} on 1 , we have for any objects c, c' in $\mathcal{C} : Ran_K(F) \circ K(c) = Ran_K(F) \circ K(c')$. Therefore, since ϵ is a natural transformation, we have for all objects c, c' in \mathcal{C} the commutative diagram :



Furthermore, $Ran_K(F) : 1 \rightarrow \mathcal{E}$ is an object e in \mathcal{E} . Thus, the data of $(Ran_K(F), \epsilon : Ran_K(F) \circ K \Rightarrow F)$ is the same than the data

of a cone of F .

Now, the universal property of right Kan extension applied in this case will translate as the universal property for cones of F , so $\text{Ran}_K F(1)$ is a limit of F . Of course, the exact same thing can be performed for colimits and left Kan extensions.

Thus, we can conclude that limits are just an example of right Kan extension.

We just stated that limits and colimits can be seen as Kan extension, but another fundamental aspect in the practical computation is that the converse is also valid for some Kan extensions.

This characterization is actually pretty important, because in the practice, most Kan extensions we encounter will satisfy this property : such Kan extensions will be called pointwise (this terminology comes from the fact that the expression of the functor is expressed as a limit, taken for any object).

First let's quickly recall the definition of comma categories:

Definition 3.2.4.

Let \mathcal{C}, \mathcal{D} be categories, $K : \mathcal{C} \rightarrow \mathcal{D}$ a functor and d an object in \mathcal{D} .

We define the comma category $(d \downarrow K)$ in the following way:

- *Objects:* (c, f) , where c is an object of \mathcal{C} and $f : c \rightarrow K(c)$ an arrow in \mathcal{D} .

- *Arrows:* Given two objects (c, f) and (c', f') , an arrow is the data of

$$h : c \rightarrow c' \text{ such that the following diagram commutes: } \begin{array}{ccc} d & & \\ & \searrow^{f'} & \\ & f & \\ K(c) & \xrightarrow{K(h)} & K(c') \end{array}$$

Let's remark that we also have a projection functor $Q^d : \begin{array}{l} (d \downarrow K) \longrightarrow \mathcal{C} \\ (c, f) \longmapsto c \end{array}$
(This is the one we said we would introduce at the end of chapter 2.)

Now we can define pointwise Kan extensions :

Proposition 3.2.5.

Given $K : \mathcal{C} \rightarrow \mathcal{D}$, let $F : \mathcal{C} \rightarrow \mathcal{E}$ be a functor such that the composite $(d \downarrow K) \rightarrow \mathcal{C} \rightarrow \mathcal{E}$ admits, for each $d \in \mathcal{D}$ a limit in \mathcal{E} , with limiting cone λ , written $R_d = \text{Lim}((d \downarrow K) \rightarrow \mathcal{C} \rightarrow \mathcal{E})$

Then, each $g : d \rightarrow d'$ induces an unique arrow :

$$R_g : \text{lim}(F \circ Q^d) \longrightarrow \text{Lim}(F \circ Q^{d'})$$

which means that R is a functor.

Moreover, for every $c \in \mathcal{C}$, the components $\lambda_{1_{K(c)}} = \epsilon_c$ define a natural transformation, and (R, ϵ) is the right Kan extension of F along K .

Proof.

See [9] theorem X.3.1 □

This is important according to two aspects : first, now in certain cases we have a way to explicitly calculate Kan extensions.

Thus, one can describe Kan extension in terms of limits/colimits and so, in some cases, have a more explicit construction.

On the other hand, if the Kan extensions are too painful to compute despite this formulation, we can at least deduce their existence thanks to completion properties, as the following statement shows :

Corollary 3.2.6.

Let \mathcal{C} , \mathcal{D} , \mathcal{E} be categories and $F : \mathcal{C} \rightarrow \mathcal{E}$, $K : \mathcal{C} \rightarrow \mathcal{D}$ functors.

If \mathcal{C} is small, then the right Kan extension of F along K exists when \mathcal{E} is complete, and the left Kan extension exists when \mathcal{E} is cocomplete.

Proof.

If \mathcal{C} is small and \mathcal{E} complete, then all the proper limits of $(d \downarrow K) \rightarrow \mathcal{C} \rightarrow \mathcal{E}$ exist.

Thus the right Kan extension exists, since it can be computed pointwise through these limits. We conclude in a dual way for the other case. □

Example 3.2.7.

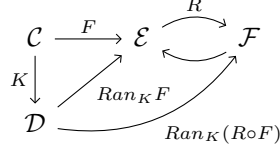
Applying this, let \mathcal{D} be a category: since Set is complete, if \mathcal{C} is a small category, then all functors $F : \mathcal{C} \rightarrow Set$ have both right and left Kan extensions along all functors $K : \mathcal{C} \rightarrow \mathcal{D}$.

Another important lemma for the proof we'll perform in the next section is that right adjoints will preserve right Kan extensions.

Lemma 3.2.8.

Let $\mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}$ be categories, and $F : \mathcal{C} \rightarrow \mathcal{E}, K : \mathcal{C} \rightarrow \mathcal{D}, R : \mathcal{E} \rightarrow \mathcal{F}$ functors. Let's assume that R is a right adjoint, then we have

$$\text{Ran}_K(RF) = R \circ \text{Ran}_K(F)$$



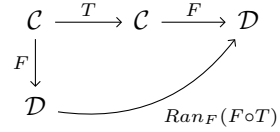
Proof.

This fact has also already been shown in [9]. □

Let's close this section with an important theorem : looking at the definition of $\text{Ran}_K F$, it is clear that taking $K=F$ makes the Kan extension an endofunctor. Therefore, one can wonder if it is also a monad : it happens to be the case as we demonstrate below (it will be a corollary of the next proposition). We call such a Kan extension the density comonad of the right Kan extension (respectively codensity monad of the right Kan extension).

Proposition 3.2.9.

Let \mathcal{C}, \mathcal{D} be categories, $T : \mathcal{C} \rightarrow \mathcal{C}$ a monad over \mathcal{C} and $F : \mathcal{C} \rightarrow \mathcal{D}$ a functor. Then, assuming $\text{Ran}_F(F \circ T) : \mathcal{D} \rightarrow \mathcal{D}$ exists pointwise, it is a monad.



Proof.

We will only construct the natural transformation μ and show that it satisfies the diagram in the definition of a monad, since proving the same for η uses pretty much the same idea and is as lengthy.

Let's denote as μ the natural transformation relative to T and μ' the natural transformation relative to $\text{Ran}_F(F \circ T) = S$ we want to create. Since S exists pointwise, $S(d) = \lim_{d \rightarrow F(c)} F(T(c))$.

We want to create an arrow $\mu' : S \circ S \rightarrow S$
 First of all, for all $i : d \rightarrow F(c)$ we have an arrow $p(i) : S(d) \rightarrow F(T(c))$.

Iterating the process, we also have $p(p(i)) : S(S(d)) \rightarrow F(T(T(c)))$ and $p(p(p(i))) : S(S(S(d))) \rightarrow F(T(T(T(c))))$. Thus we want to define $\mu'_d : S(S(d)) \rightarrow S(d)$ such that, for all $i : d \rightarrow F(c)$, the following diagram commutes :

$$\begin{array}{ccc} S(S(d)) & \xrightarrow{\bar{\mu}_d} & S(d) \\ p(p(i)) \downarrow & & \downarrow p(i) \\ F(T(T(c))) & \xrightarrow{F(\mu_c)} & F(T(c)) \end{array}$$

we can see that such an arrow μ'_d exists and is unique by applying the universal property of the limit defining $S(d)$.

Now, we have a natural transformation $\mu : S \circ S \rightarrow S$.

All we have to do to conclude is to show the commutativity of :

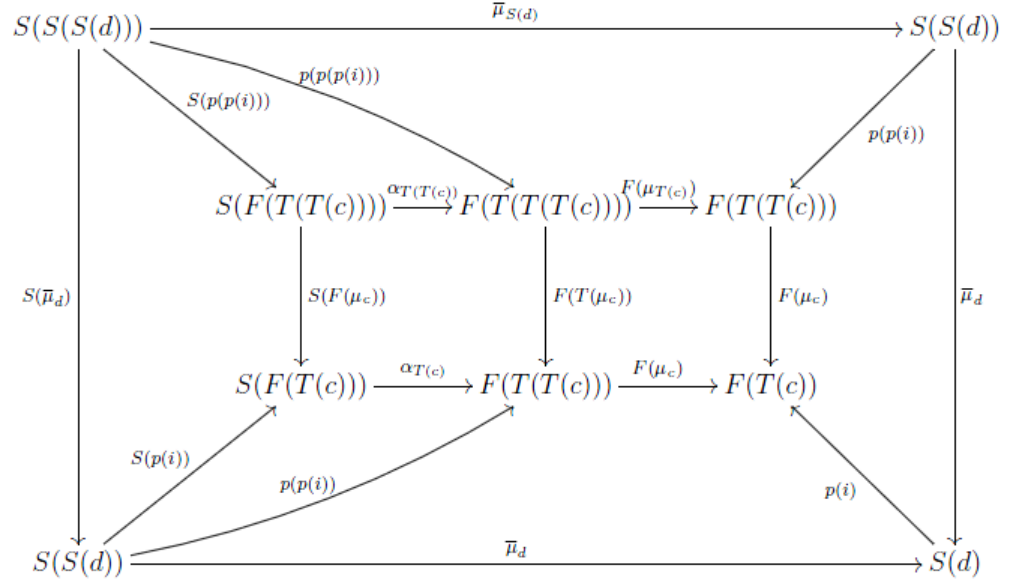
$$\begin{array}{ccc} S(S(S(d))) & \xrightarrow{\mu'_{S(d)}} & S(S(d)) \\ \downarrow S\mu'_d & & \downarrow \mu'_d \\ S(S(d)) & \xrightarrow{\mu'_d} & S(d) \end{array}$$

To do so, we need to use the information about the $p(i) : S(d) \rightarrow F(T(c))$ we introduced at the beginning : if we take $p(Id_{T(c)}) = \alpha_c$ we define an arrow $\alpha : SF \rightarrow FT$ such that the following diagram commutes :

$$\begin{array}{ccc} S(d) & \xrightarrow{S(i)} & S(F(c)) \\ & \searrow p(i) & \swarrow \alpha_c \\ & F(T(c)) & \end{array} \quad (\star)$$

Now, we can insert this information in the first diagram, let's take a look at

the physically painful diagram :



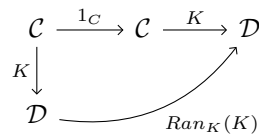
Since $(S(d) \xrightarrow{p(i)} F(T(c)))_i$ is a limit cone, the universal property of limit would allow us to conclude that the outer rectangle commutes, but only if we state that the rest of the diagram commutes; which is the case because :

- The first square commutes, since α is natural.
- The second square commutes thanks to the diagram property related to μ_c involved in the definition of the monad T .
- The triangles commute by applying \star
- The four remaining diagrams commute by definition of μ' .

□

Corollary 3.2.10.

Let \mathcal{C}, \mathcal{D} be categories, $K : \mathcal{C} \rightarrow \mathcal{D}$ a functor. Then, the right Kan extension $Ran_K(K) : \mathcal{D} \rightarrow \mathcal{D}$ is a monad over \mathcal{D} , we'll refer it as the codensity monad.



Proof.

It's only a special case of the previous theorem for $T = 1_C$. \square

Remark 3.2.11.

One can check that, in this particular case, we can get an actual description of η and μ .

- Indeed, $Id_{\mathcal{D}}$ is an endofunctor on \mathcal{D} , and we can take the natural transformation $Id_K : Id_{\mathcal{D}} \circ K \Longrightarrow K$.
Now, by applying the universal property of right Kan extensions to $(Id_{\mathcal{D}}, Id_K)$, we get a natural transformation $\eta : Id_{\mathcal{D}} \Longrightarrow Ran_K(K)$, and one can show that it satisfies the diagram property required in the definition of a monad.
- Let's consider $Ran_K(K) \circ Ran_K(K)$. By definition of the right Kan extension, we have a natural transformation $\epsilon : Ran_K(K) \circ K \Longrightarrow K$.
Thus, we have $Ran_K(K)\epsilon : Ran_K(K) \circ Ran_K(K) \circ K \Longrightarrow Ran_K(K) \circ K$, and eventually: $\epsilon \circ Ran_K(K)\epsilon : Ran_K(K) \circ Ran_K(K) \circ K \Longrightarrow K$.
Now, by applying the universal property of right Kan extensions to $(Ran_K(K) \circ Ran_K(K), \epsilon \circ Ran_K(K)\epsilon)$, we get a natural transformation $\mu : Ran_K(K) \circ Ran_K(K) \Longrightarrow Ran_K(K)$, and one can show that it satisfies the diagram property required in the definition of a monad.

Example 3.2.12.

Let's consider the following diagram (the functors used here will be described just after) :

$$\begin{array}{ccccc}
 Set_{fin}^T & \xrightarrow{U} & Set_{fin} & \xrightarrow{\beta_{fin}} & Stone \\
 U \downarrow & & & & \nearrow \\
 Set_{fin} & & & & \\
 \beta_{fin} \downarrow & & & & \\
 Stone & \xrightarrow{\tilde{T} = Ran_{\beta_{fin} \circ U}(\beta_{fin} \circ U)} & & &
 \end{array}$$

By the previous corollary, \tilde{T} is the codensity monad on the functor $Stone$ of $\beta_{fin} \circ U$.

Example 3.2.13.

In exactly the same way, we have the diagram :

$$\begin{array}{ccccc}
 \mathit{Set}_{fin}^T & \xrightarrow{U} & \mathit{Set}_{fin} & \xrightarrow{\iota} & \mathit{Set} \\
 U \downarrow & & & & \nearrow \\
 \mathit{Set}_{fin} & & & & \\
 \downarrow \iota & & & & \\
 \mathit{Set} & \xrightarrow{\quad \bar{T} = \mathit{Ran}_{\iota \circ U}(\iota \circ U) \quad} & & &
 \end{array}$$

So \bar{T} is the codensity monad on Set of the functor $\iota \circ U$.

3.3 \bar{T} with Kan extensions and \tilde{T}

This computation heavily relies on the following diagram :

$$\begin{array}{ccccc}
 \mathit{Set}_{fin}^T & \xrightarrow{U} & \mathit{Set}_{fin} & \xrightarrow{\beta_{fin}} & \mathit{Stone} & \xrightarrow{U'} & \mathit{Set} \\
 U \downarrow & & \nearrow \mathit{Ran}_U U & & \nearrow \tilde{T} & & \nearrow \beta \\
 \mathit{Set}_{fin} & & & & \mathit{Stone} & \xleftarrow{\beta} & \mathit{Set} \\
 \downarrow \beta_{fin} & & & & \downarrow \iota & & \\
 \mathit{Stone} & & & & \mathit{Set} & \xleftarrow{U'} & \mathit{Set} \\
 \downarrow U' & & & & \downarrow \beta & & \\
 \mathit{Set} & \xrightarrow{\quad \bar{T} \quad} & & & \mathit{Set} & \xrightarrow{\quad \bar{T} \quad} & \mathit{Set}
 \end{array}$$

where :

- U is the forgetful functor defined for T -algebras in the prerequisites.
- U' is the forgetful functor from Stone to Set .
- β is the Stone-Čech compactification (the construction of this functor isn't required to understand what we are going to do, but for the interested reader let's note that it uses ultrafilters and is explained in details in [10]). There's an adjunction between the two previous functors (once again, this classic proof can be found in [10]).
- ι is the "injection" functor that only consists of looking at a finite set as a set.

- β_{fin} is the restriction of β , defined earlier, to finite sets.
In this case, it's easy to understand how it works : given a finite set, we look at this same set, but endowed with the discrete topology, which is indeed an object in \mathcal{Stone} .
- $\overline{\overline{T}} = \text{Ran}_{U'}(U' \circ \widetilde{T})$
- $\widetilde{T} = \text{Ran}_{(\beta_{fin} \circ U)}(\beta_{fin} \circ U)$

Let's remark the two last functors' existence come from applying the corollary 3.2.8, since \mathcal{Stone} and \mathcal{Set} are known to be complete.

We now have another endofunctor in \mathcal{Set} , namely $\overline{\overline{T}}$, and it is expressed as a right Kan extension: truth is it is exactly the same functor than the one we saw in chapter 2: $\overline{\overline{T}}$!

Proposition 3.3.1.

$\overline{\overline{T}}$ define $\overline{\overline{T}}$ the same monads.

Proof.

In the chapter 2, we saw that the definition of $\overline{\overline{T}}$ in terms of limits can be formulated, thanks to the definition of pointwise right Kan extension, as :

$$\overline{\overline{T}} = \text{Ran}_{\iota \circ U}(\iota \circ U)$$

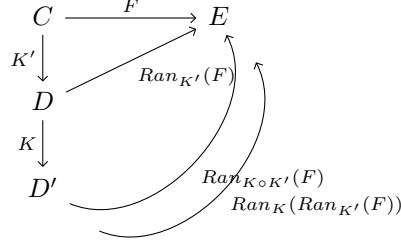
First of all, let's note that we can immediately check that $\overline{\overline{T}}$ is a monad by applying proposition 3.2.7.

Another important observation is that, $\iota = U' \circ \beta_{fin}$: indeed, if we take a finite set and look at it as a set (without any assumptions upon finiteness), it's exactly the same thing that endowing this finite set with the discrete topology, and only looking at the carrier set afterwards. Therefore, these two functors are the same (the verification on arrows is of the same level of triviality).

Thus : $\overline{\overline{T}} = \text{Ran}_{(\iota \circ U)}(\iota \circ U) = \text{Ran}_{(U' \circ \beta_{fin} \circ U)}(U' \circ \beta_{fin} \circ U)$

But, in general, we know that for any functors $K' : \mathcal{C} \rightarrow \mathcal{D}$, $K : \mathcal{D} \rightarrow \mathcal{D}'$, $F : \mathcal{C} \rightarrow \mathcal{E}$, we have: $\text{Ran}_{K \circ K'}(F) = \text{Ran}_K(\text{Ran}_{K'}F)$.

[[Indeed, we want to show that $\text{Ran}_{K \circ K'}(F)$ satisfies the universal property of $\text{Ran}_K(\text{Ran}_{K'}F)$, thus considering $(G : \mathcal{D}' \rightarrow \mathcal{E}, \delta : G \circ K \Rightarrow \text{Ran}_{K'}F)$ we're looking for a (unique) functor $\beta : G \Rightarrow \text{Ran}_K(\text{Ran}_{K'}F)$ satisfying the associativity of the universal property. But, since $\text{Ran}_K(\text{Ran}_{K'}F)$ is a right Kan extension, it already gives us a candidate, and this functor satisfies the required properties).]]



Therefore :

$$Ran_{(U' \circ \beta_{fin} \circ U)}(U' \circ \beta_{fin} \circ U) = Ran_{U'}(Ran_{\beta_{fin} \circ U}(U' \circ \beta_{fin} \circ U))$$

Now, since U' is a right adjoint to β , we can apply lemma 3.2.7 and we have:

$$Ran_{U'}(U' \circ Ran_{(\beta_{fin} \circ U)}(\beta_{fin} \circ U))$$

But this is exactly $Ran_{U'}(U' \circ \tilde{T})$, which is exactly the definition of \bar{T} . \square

Now that we identified Bojańczyk's monad to an expression where the monad on Stone \tilde{T} directly intervenes, we can finally conclude and express it more naturally.

Proposition 3.3.2.

We have: $\bar{T} = U' \circ \tilde{T} \circ \beta$

Proof.

We just showed that $\bar{T} = Ran_{U'}(U' \circ \tilde{T})$, but since U' is a right adjoint we have already stated that, in that case, $Ran_{U'}(U' \circ \tilde{T}) = U' \circ Ran_{U'}(\tilde{T})$.

Therefore all we have left to show is that $Ran_{U'}(\tilde{T}) = \tilde{T} \circ \beta$.

We will show that this statement is true in a more general context : let $\mathcal{C}, \mathcal{D}, \mathcal{E}$ be categories, $\beta : \mathcal{D} \rightarrow \mathcal{C}$, $U' : \mathcal{C} \rightarrow \mathcal{D}$, $G : \mathcal{C} \rightarrow \mathcal{E}$, assuming that $\beta \dashv U'$.

We want to show that $Ran_{U'}G = G \circ \beta$, to do so all we need to show is that $G \circ \beta$ satisfies the universal property describing $Ran_{U'}G$.

Let's consider a natural transformation $K \circ U'G$, composing with β we get a natural transformation $K \circ U' \circ \beta \implies G \circ \beta$.

On the other hand, since we have the adjunction $\beta \dashv U'$, the unit gives us a natural transformation $\eta : 1_{\mathcal{D}} \implies U' \circ \beta$.

Composing it with K , we get $K \implies K \circ U' \circ \beta$: composing with the transformation we obtained previously, we get a natural transformation $K \implies G \circ \beta$, which satisfies the associativity of the universal property of left Kan extension : therefore $Ran_{U'}G = G \circ \beta$ and in particular for $G = \tilde{T}$.

\square

Conclusion and thanks

I hope this article will be helpful in at least two ways I can think of for now. First, for the layman, I tried to make it a descent introduction to the most basic tools and results one needs when he has to cope with Stone duality and the related categorical constructions. Second, for the advanced player, the idea that the behavior of the monad on $\mathcal{S}et$ is strongly related to a monad on $\mathcal{S}tone$, and the functorial interactions between $\mathcal{S}et$ and it's "surrounding categories" is certainly a fundamental mechanism to try to implement in order to make these ideas reach the next level.

Indeed, to generalize to the fullest, it seems clear that the next step would be to study monads and T-algebras over any category; however one will quickly point out that most of the steps we took strongly relied on the good properties of the category $\mathcal{S}et$: keeping them seems like the good option at first. On the other hand, since languages and automaton are deeply related, maybe it would be a good idea to take a look at the categorical generalization of automaton (monoidal categories, since a product is required) : if things happen to behave similarly to the way they do for monoids, it would give a good intuition to attain the previous goal.

First of all, I would like to make this article slightly longer by thanking my mother who doesn't speak a word of English nor math.

Je voudrais également remercier ma soeur d'avoir pris la peine de venir à ma soutenance comme assist, même si au final je l'ai juste fait attendre 2 heures dans mon bureau, et qu'elle a fini par s'endormir.

More importantly, I want to thank my advisor, Mrs Mai Gehrke, for accepting to take me with her for these two months, and also for bothering finding this subject for me, which was really interesting in my opinion.

I really appreciated working on it, even though I sincerely wished I had more time to develop more solid content, I feel like I had to rush in at some moments.

Very important, I want to thank Daniela Petrisan for her regular help during these two months : thanks for answering my questions when I was stuck, and also for helping me to catch up with all the things I was not familiar with. Really, thank you Daniela, I can't use too much punctuation in an article cause it's probably illegal, but I mean it.

I'm not very keen to ask for help in general, but I think she helped me to learn the importance of doing it when you're seriously working in research, and I feel like it's worth remembering.

I also want to thank Mrs Olivia Caramello, for presenting me Mai, for replying to me very fast whenever I have a question on something, and for being an inspiring figure, something like Marie Curie but less green glowing.

Finally I'll gladly thank IRIF for the amazing working conditions they put me in when I arrived, especially because I think it's important to salute when people actually do a good job in that department : it's so rare.

I will also give a shout-out to all the PhD students I could speak with (especially Florent and Clément, for being so welcoming, and also for teaching me how to use Google maps), not because it's polite to do so, but because there was not one person who wasn't friendly, there's a really good atmosphere here, I enjoyed staying.

Last but not least, I would like to thank Alexis Roquefeuil and Léo Guetta for correcting my Latex dyslexia, and also Sacha Ikonnikov for taking a few hours in order to give me an intuition over monads.

Bibliography

- [1] Mikołaj Bojańczyk. Recognisable languages over monads. In *DLT*, volume 9168 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2015.
- [2] Henning Urbat, Jirí Adámek, Liang-Ting Chen, and Stefan Milius. One eilenberg theorem to rule them all. *CoRR*, abs/1602.05831, 2016.
- [3] Liang-Ting Chen, Jirí Adámek, Stefan Milius, and Henning Urbat. Profinite monads, profinite equations and reiterman’s theorem. *CoRR*, abs/1511.02147, 2015.
- [4] Howard Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhauser, 1994.
- [5] Mirte Dekkers. *Stone Duality, an application in the theory of formal languages*. 2008.
- [6] Jean-Eric Pin. *Automates finis*. 2002.
- [7] Alain Prouté. *Introduction à la logique catégorique*. 2010.
- [8] Steven Givant and Paul Halmos. *Introduction to Boolean Algebras*. Springer, 1974.
- [9] S.M. Lane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics. Springer New York, 1998.
- [10] Peter Johnstone. *Stone spaces*. Cambridge university press, 1982.