Varieties of Cost Functions

Laure Daviaud[1], Denis Kuperberg[2], and Jean-Éric Pin[3]

1   **Aix Marseille Université, CNRS, LIF UMR 7279 and LIP, ENS de Lyon, France**
    `laure.daviaud@ens-lyon.fr`
2   **ONERA/DTIM, IRIT, University of Toulouse, France**
    `denis.kuperberg@gmail.com`
3   **LIAFA, CNRS and Université Paris Diderot, France**
    `jean-eric.pin@liafa.univ-paris-diderot.fr`

―――― **Abstract** ――

Regular cost functions were introduced as a quantitative generalisation of regular languages, retaining many of their equivalent characterisations and decidability properties. For instance, stabilisation monoids play the same role for cost functions as monoids do for regular languages. The purpose of this article is to further extend this algebraic approach by generalising two results on regular languages to cost functions: Eilenberg's varieties theorem and profinite equational characterisations of lattices of regular languages. This opens interesting new perspectives, but the specificities of cost functions introduce difficulties that prevent these generalisations to be straightforward. In contrast, although syntactic algebras can be defined for formal power series over a commutative ring, no such notion is known for series over semirings and in particular over the tropical semiring.

1   Introduction

Quantitative extensions of regular languages have been studied for over 50 years. Most of them rely on the early work of Schützenberger [25, 26, 27], who extended Kleene's theorem to formal power series over a semiring. A very nice presentation of this theory can be found in the book of Berstel and Reutenauer [5]. In this setting, weighted automata play the role of automata and weighted logic was introduced as an attempt to generalise Büchi's characterisation of regular languages in monadic second order logic. See the handbook [12] for an overview and further references.

However, this theory also suffers some weaknesses. For instance, the equality problem for rational series with multiplicities in the tropical semiring is undecidable [15], a major difference with the equality problem for regular languages, which is decidable. To overcome this problem and other related questions, Colcombet introduced the notion of regular cost functions [9], an other quantitative generalisation of regular languages. Cost functions are formally defined as equivalence classes of power series with coefficients in the semiring $\mathbb{N} \cup \{\infty\}$. This equivalence does not retain the exact values of the coefficients of the series but measures boundedness in some precise way. Thus cost functions are less general than power series, but are still more general than languages, which can be viewed as cost functions associated with their characteristic functions.

SYMPOSIUM
ON THEORETICAL
ASPECTS
OF COMPUTER
SCIENCE

This approach proved to be very successful. It leads to simplified proofs of several major results related to boundedness (the limitedness problem of distance automata, Kirsten's proof of the star-height problem, etc.). Moreover, the equivalences on regular languages

regular languages ⇔ finite automata ⇔ finite monoids ⇔ monadic second order logic

admit the following nontrivial extension

regular cost functions ⇔ cost automata ⇔ stabilisation monoids ⇔ cost monadic logic .

Contributions

The aim of this paper is to show that the algebraic approach to regular languages also extends to the setting of cost functions. To this end, we change the recognising object of cost functions from *stabilisation monoid* [9] to a structure with better algebraic properties, called *stabilisation algebra*. This gives us a new way of interpreting cost functions, as particular sets of a free stabilisation algebra $F(A)$ on the alphabet A, generalising the set of words A^*. This allows us to extend the ordered version [19] of Eilenberg's varieties theorem [13], which gives a bijective correspondence between positive varieties of languages and varieties of finite ordered monoids (Theorem 5.5). We show that the profinite algebra $\widehat{F(A)}$ generalising profinite words is the dual of the lattice of regular cost functions. This leads to an extension of the duality results between profinite words and regular languages. In particular, we extend the equational approach to lattices of regular languages given in [14] (Theorem 7.5). Our approach not only subsumes the corresponding results on languages but it also gives a nice algebraic framework for the results of [11, 16]. A series of examples is given in Section 8.

Related work

Toruńczyk [30] also established a link between cost functions and profinite words, using a different approach. More precisely, Toruńczyk identifies a regular cost function with the set of profinite words that are limits of infinite sequences of words over which the function is bounded.

It is also interesting to compare these results to similar results on formal power series. Syntactic algebras of formal power series over a commutative ring were introduced by Reutenauer [22, 23], but no such notion is known for semirings. Reutenauer also extended Eilenberg's varieties theorem to power series over a commutative field. However, as shown in [24], equational theory only works for power series over finite fields.

Finally, let us mention two new promising approaches to recognisability, using respectively categories [1, 2] and monads [7, 8]. For the time being, these two approaches do not seem to apply to cost functions, but we hope our paper will serve as a test bench for future developments of this new point of view.

2   Regular Cost Functions and Stabilisation Monoids

In this section, we introduce the notions of cost functions and of stabilisation monoids. For a more complete and detailed presentation, the reader is referred to [10].

Let A be a finite alphabet and let $\mathcal{F}$ be the set of all functions from A^* to $\mathbb{N} \cup \{\infty\}$. Colcombet [9] introduced the following equivalence relation on $\mathcal{F}$: two elements f and g of $\mathcal{F}$ are *equivalent* (denoted $f \approx g$) if, for each subset S of A^*, f is bounded on S if and only if g is bounded on S. A *cost function* is a $\approx$-class. In practice, cost functions are always represented by one of their representatives in $\mathcal{F}$.

The equivalence relation $\approx$ behaves well with respect to the operations min and max, defined in the usual way. Indeed for all $f, g, h \in \mathcal{F}$, if $f \approx g$, then $\min(f, h) \approx \min(g, h)$ and $\max(f, h) \approx \max(g, h)$ [9]. It follows that the minimum and the maximum of two cost functions are well-defined notions.

▶ **Example 2.1.** Let $A = \{a, b\}$. Given a word u, let $|u|$ denote the length of u and $|u|_a$ the number of occurrences of the letter a in u. Let us define three functions f, g and h from A^* to $\mathbb{N} \cup \{\infty\}$ by setting $f(u) = |u|$, $g(u) = |u|_a$ and $h(u) = 2|u|_a$. Then g is equivalent to h and they represent the same cost function, whereas g is not equivalent to f. Indeed g is bounded on b^* and f is not since for all n, $g(b^n) = 0$ and $f(b^n) = n$.

The *characteristic function* of a language L on A^* is the function $\chi_L : A^* \to \mathbb{N} \cup \{\infty\}$ defined by $\chi_L(u) = 0$ if $u \in L$ and ∞ otherwise. The crucial observation that $\chi_L \approx \chi_{L'}$ if and only if $L = L'$ allows one to identify a language with the cost function defined by its characteristic function.

Stabilisation monoids were introduced in [9] in order to extend the classical notion of monoids recognising a language to the setting of cost functions. Recall that an *ordered monoid* is a set equipped with an associative binary product, a neutral element and an order compatible with the product, i.e., the conditions $x_1 \leqslant x_2$ and $y_1 \leqslant y_2$ imply $x_1 y_1 \leqslant x_2 y_2$. We let $E(M)$ denote the set of idempotents of a monoid M.

Following [9], we define a *stabilisation monoid* as an ordered monoid M together with a *stabilisation operator* $\sharp : E(M) \to E(M)$ satisfying the following properties:

(S_1) for all $s, t \in M$ such that $st \in E(M)$ and $ts \in E(M)$, one has $(st)^\sharp s = s(ts)^\sharp$,

(S_2) for all $e \in E(M)$, one has $(e^\sharp)^\sharp = e^\sharp e = e e^\sharp = e^\sharp \leqslant e$,

(S_3) for all $e, f \in E(M)$, $e \leqslant f$ implies $e^\sharp \leqslant f^\sharp$,

(S_4) $1^\sharp = 1$.

Given two stabilisation monoids M and N, a *morphism* φ from M to N is a monoid morphism which is order-preserving and $\sharp$-preserving: if $e \in E(M)$, then $\varphi(e)^\sharp = \varphi(e^\sharp)$.

Just like finite (ordered) monoids recognise regular languages, finite stabilisation monoids recognise regular cost functions. However, the formal definition of recognition is more involved for cost functions than for languages and relies on the notion of factorisation trees. Let M be a stabilisation monoid and let $h : A \to M$ be a function, called the *labelling map*.

▶ **Definition 2.2.** Let $w = a_1 a_2 \cdots a_k$ be a word of A^* where each a_i is a letter. An h-*factorisation tree of threshold n* for w is a finite tree labelled by the elements of M and such that:

(T_1) the tree has exactly k leaves, labelled by $h(a_1), \ldots, h(a_k)$, respectively,

(T_2) each binary node is labelled by the product of its left child's label by its right child's label,

(T_3) if a node has arity > 2, then all its children are labelled by the same idempotent e. If the arity of the node is $\leqslant n$, then the node is labelled by e, otherwise it is labelled by $e^\sharp$.

▶ **Example 2.3.** Let S_1 be the stabilisation monoid containing three idempotent elements $\{1, s, 0\}$ with $0 < s < 1$, $0^\sharp = s^\sharp = 0$, 0 is a zero of the monoid and 1 the neutral element. Let $h(a) = s$ and $h(b) = 1$. See on page 4 a factorisation tree of threshold 5 for the word $abaabbbbaaaabbba$. A variant of Simon's factorisation theorem [9, 29] guarantees the existence of trees of bounded height to evaluate input words. More precisely, for each labelling function

$h : A \to M$, there is a positive integer K $(= 3|M|)$ such that for all words w and for all integers $n \geqslant 3$, there is an h-factorisation tree of threshold n for w with height at most K.
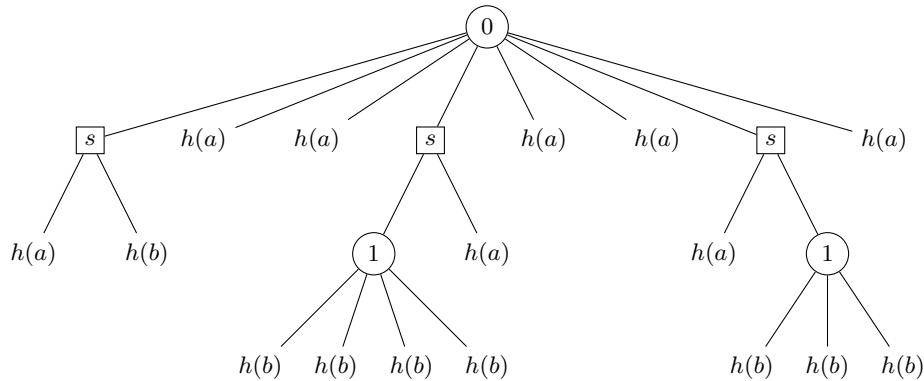
We can now give the formal definition of a regular cost function recognised by a finite stabilisation monoid. Recall that a subset D of a partially ordered set is a *downset* if the conditions $t \in D$ and $s \leqslant t$ imply $s \in D$.

▶ **Definition 2.4.** Let M be a finite stabilisation monoid, $h : A \to M$ a labelling map and I a downset of M. The cost function *recognised* by (M, h, I) is the equivalence class of the function that maps a word u to the maximal threshold n such that there exists an h-factorisation tree for u of threshold n, height at most $3|M|$ and root in I. Such an equivalence class of functions is called a *regular cost function*.

▶ **Example 2.5.** Consider the stabilisation monoid S_1 defined in Example 2.3 and let $I = \{0\}$. Let $h : \{a, b\} \to M$ be the labelling map defined by $h(a) = s$ and $h(b) = 1$. Then S_1 is a stabilisation monoid that can make a distinction between products with no s (that are 1), products containing "few" s (that are s) and products containing "a lot of" s (that are $0 = s^\sharp$). The cost function recognised by (S_1, h, I) is the equivalence class of the function $u \mapsto |u|_a$.

For instance the tree from example 2.3 of height 4 and threshold 5 has root labelled by $s^\sharp = 0$ because it is a witness that there are more than 5 occurrences of a in the input word. Conversely, such a factorisation tree of threshold n and height k would have its root labelled by 1 if the input word contains no a, and labelled by s if there are at most n^k occurrences of a. Because k is a fixed constant, these trees can be used to recognise the cost function $u \mapsto |u|_a$, since it is equivalent to $u \mapsto (|u|_a)^k$.

Regular cost functions can also be recognised by generalised forms of nondeterministic finite automata, regular expressions or monadic second-order logical formulas. See [11] for a complete introduction. Moreover, every regular cost function f has a unique *syntactic stabilisation monoid M*, in the sense that:

(1) there is a unique pair (h, I) where $h : A \to M$ is a labelling function and I a downset of M such that f is recognised by (M, h, I),

(2) for any (M', h', I') recognising f, there is a surjective morphism $\varphi : M \to M'$ such that $h = \varphi \circ h'$ and $I' = \varphi^{-1}(I)$.

3   Stabilisation Algebras

The goal of the present work is to study algebraic properties of stabilisation monoids and cost functions. In particular, we would like to define regular cost functions as particular subsets

of a free stabilisation monoid. However, since in a stabilisation monoid, the $\sharp$-operator is only defined on idempotents, the notion of a free stabilisation monoid cannot be defined directly and requires the introduction of a new algebraic structure, in which idempotents are directly defined in the signature of the algebra: stabilisation algebras.

Given a countable set of variables X, let $T(X)$ be the free term algebra of signature $\{\cdot, \omega, \sharp, 1\}$ over X. An *identity over* $T(X)$ is an equation of the form $s \leqslant t$, where s and t are terms of $T(X)$.

A finite stabilisation monoid M *satisfies the identity* $s \leqslant t$ if the equation holds for any instantiation of the variables by elements of M, where 1 is interpreted as the neutral element of M, ω is interpreted as the idempotent power in M, and $\sharp$ is replaced with $\omega\sharp$ (to guarantee that $\sharp$ is only applied to idempotents). A finite stabilisation monoid M *satisfies the identity* $s = t$ if it satisfies the identities $s \leqslant t$ and $t \leqslant s$.

We can now define the structure of a stabilisation algebra in the following way.

▶ **Definition 3.1.** A *stabilisation algebra* is an ordered algebra M with signature $\langle 1, \leqslant, \cdot, \omega, \sharp \rangle$ satisfying the following axioms:

(**A**$_1$) all identities that are satisfied by all finite stabilisation monoids,

(**A**$_2$) a description of the behaviour of ω on idempotent elements: $x^2 = x$ implies $x^\omega = x$,

(**A**$_3$) the three properties expressing that the order $\leqslant$ is compatible with the operations $\cdot, \omega, \sharp$: $x_1 \leqslant x_2$ and $y_1 \leqslant y_2$ imply $x_1 y_1 \leqslant x_2 y_2$, and $x \leqslant y$ implies $x^\omega \leqslant y^\omega$ and $x^\sharp \leqslant y^\sharp$.

In particular, (**A**$_1$) implies that a stabilisation algebra is a monoid with neutral element 1. A *morphism* between two stabilisation algebras is a monoid morphism which is order-preserving, ω-preserving and $\sharp$-preserving. Let M and N be two stabilisation algebras. Then N is a *stabilisation subalgebra* of M if $N \subseteq M$, and N is a *quotient* of M if there exists a surjective morphism $M \to N$. The product of two stabilisation algebras M and N is defined on the set product $M \times N$, with operations defined componentwise.

Recall that in a finite monoid, every element x has a unique idempotent power, denoted x^ω. This fact allows one to identify finite stabilisation monoids and finite stabilisation algebras.

▶ **Proposition 3.2.** *Finite stabilisation algebras are in one-to-one correspondence with finite stabilisation monoids, by interpreting ω as the idempotent power.*

We now build a free stabilisation algebra $F(A)$ on each finite alphabet A. Recall that $T(A)$ is the set of terms of the free algebra of signature $\{\cdot, \omega, \sharp, 1\}$ over the alphabet A. Given s and t two elements of $T(A)$, we write $s \equiv t$ if and only if the identity $t = s$ holds in all finite stabilisation monoids. This defines an equivalence relation and we let $F(A)$ denote the set of $\equiv$-classes. Furthermore we let $\bar{t}$ denote the equivalence class of t in $F(A)$.

▶ **Proposition 3.3.** $F(A)$ *can be equipped with a structure of stabilisation algebra.*

This follows from a general result on ordered algebras mentioned without proof in [6].

A recent result [18] states that the equivalence of two $\sharp$-free terms of $T(A)$ is decidable. Actually, the result is more general and also covers the case of $\omega - 1$ powers. However, deciding the equivalence of arbitrary terms in $T(A)$ seems to still be an open problem.

The following theorem shows that $F(A)$ is a free object, by making explicit the corresponding universal property.

▶ **Theorem 3.4** (Universal Property). *For any stabilisation algebra M and any function $h : A \to M$, there exists a unique morphism of stabilisation algebra $\overline{h} : F(A) \to M$ extending h.*

▶ **Corollary 3.5.** *Every A-generated stabilisation algebra is a quotient of $F(A)$.*

4 Recognisability

We now define the notion of recognisable downsets in the free stabilisation algebra. We will later see how a regular cost function can be identified with a recognisable downset. This will allow us to generalise the classical notions of syntactic congruence and syntactic monoid. We identify terms $t \in T(A)$ and their class $\overline{t} \in F(A)$ for more readability.

Let I be a downset of $F(A)$, let M be a stabilisation algebra and let $h : F(A) \to M$ be a surjective morphism. We say that I is *recognised by h* if there exists a downset J of M such that $I = h^{-1}(J)$. A downset I of $F(A)$ is said to be *recognisable* if it is recognised by some morphism onto a finite stabilisation algebra.

Syntactic congruence and syntactic stabilisation algebra

A *context* on A is an element of $T(A \cup \{x\})$, where $x \notin A$. In other words, a context is a term $T(A)$ with possible occurrences of the free variable x. Given a context C on A and an element t of $T(A)$, we let $C[t]$ denote the element of $T(A)$ obtained by replacing all the occurrences of x by t in C, i.e., $C[t] = C[x \leftarrow t]$. Let $\mathrm{Ctx}(A)$ denote the set of contexts on A.

Given a downset I of $F(A)$ and two elements t and s of $F(A)$, we write that $s \sim_I t$ if for any context C, $C[s] \in I$ is equivalent to $C[t] \in I$. This equivalence relation $\sim_I$ is a congruence on $F(A)$, called the *syntactic congruence* of I and the the quotient algebra $F(A)/\sim_I$ is the *syntactic stabilisation algebra* of I. The quotient morphism $h : F(A) \to F(A)/\sim_I$ is the *syntactic morphism* of I and the triple $(F(A)/\sim_I, h, h(I))$ is called the *syntactic triple* of I.

Given a recognisable downset I of M, there is a natural preorder among the morphisms recognising I: given $h_1 : M \to N_1$ and $h_2 : M \to N_2$, we set $h_1 \leqslant h_2$ if there exists a surjective morphism $h : N_1 \to N_2$ such that $h_2 = h \circ h_1$. Then we can state:

▶ **Proposition 4.1.** *The syntactic morphism is a minimal element for this preorder.*

The analog of this property in the framework of stabilisation monoids is given in [16].

Regular Cost Functions Versus Recognisable Downsets

We have seen that regular cost functions are recognised by finite stabilisation monoids and that recognisable downsets are recognised by finite stabilisation algebras. Now, Proposition 3.2 shows that finite stabilisation algebras correspond exactly to finite stabilisation monoids. These results indicate that regular cost functions and recognisable downsets are closely related.

One can make this relation a bijection as follows. Let f be a regular cost function and let M be its syntactic stabilisation monoid. Let also (h, I) be the unique pair (where $h : A \to M$ is a labelling function and I is a downset of M) such that f is recognised by (M, h, I). Then one can view M as a stabilisation algebra and extend h to a morphism $h : F(A) \to M$. Then the recognisable downset associated with f is $h^{-1}(I) \subseteq F(A)$.

Conversely, for every recognisable downset $I \subseteq F(A)$, one can define the regular cost function f associated to I by considering the syntactic triple (M, h, J) of I, and see M as a stabilisation monoid.

It is interesting to consider the link with regular languages. Let L be a regular language and let $\sim_L$ be its syntactic congruence. Let I_L be the downset corresponding to the regular cost function χ_L, i.e., the downset representing the language L. Since downsets represents cost functions via their unbounded elements, I_L is actually the set of elements of $T(A)$ representing words **not** in L. Any element of $T(A)$ can be tested for membership in L by evaluating it in the finite stabilisation algebra $M = A^*/\sim_L$ where $\sharp = id$ and ω is the idempotent power. Remark now that for all $u, v \in A^*$, we have $u \sim_L v$ if and only if $u \sim_{I_L} v$ (where u and v are interpreted in $T(A)$). In this sense, $\sim_I$ is an extension of the classical syntactic congruence on languages.

▶ **Example 4.2.** Let $A = \{a, b\}$. Consider the downset of $T(A)$ which consists of all terms containing an occurrence of a under the scope of $\sharp$. This set describes words containing a large number of a's. It is of finite index, and it represents the cost function g given in Example 2.1 that counts the number of a's in a word. It is also recognised by the stabilisation monoid given in Example 2.3 with elements $1 \geqslant a \geqslant 0$, all idempotent, and with $a^\sharp = 0^\sharp = 0$.

▶ **Proposition 4.3.** *The lattice of regular cost functions under* min *and* max *is isomorphic to the lattice of recognisable downsets under union and intersection.*

5   Varieties

We now generalise the notion of varieties of regular languages and some proofs from [13, 19].

A *lattice of recognisable downsets* is a set of recognisable downsets containing $\emptyset$ and $F(A)$, and closed under finite union and finite intersection. A lattice $\mathcal{L}$ of regular downsets of $F(A)$ is *closed under contexts* if, for every $I \in F(A)$ and each context $C \in \text{Ctx}(A)$, the condition $I \in \mathcal{L}$ implies $C^{-1}[I] \in \mathcal{L}$, where $C^{-1}[I] = \{t \in F(A) \mid C[t] \in I\}$.

A *variety of recognisable downsets* associates with each finite alphabet A a lattice $\mathcal{V}(A)$ of recognisable downsets of $F(A)$ satisfying the following properties:

(**V**$_1$) For each alphabet A, $\mathcal{V}(A)$ is closed under contexts.

(**V**$_2$) For each morphism $\varphi : F(B) \to F(A)$, the condition $I \in \mathcal{V}(A)$ implies $\varphi^{-1}(I) \in \mathcal{V}(B)$.

Varieties of downsets generalise positive varieties of languages [19], as there is no complementation for downsets.

▶ **Example 5.1.** A recognisable downset I is *aperiodic* if for all $t \in F(A)$, the relation $t^\omega \sim_I t^\omega t$ holds. It is not too difficult to show that aperiodic downsets form a variety of recognisable downsets.

We now define varieties of stabilisation algebras.

▶ **Definition 5.2.** A *variety of finite stabilisation algebras* is a class of finite stabilisation algebras closed under taking stabilisation subalgebras, quotients and finite products.

Notice that this notion is often called *pseudovarieties* in the literature, as opposed to Birkhoff varieties which are also closed under arbitrary products.

▶ **Example 5.3.** A finite stabilisation algebra M is aperiodic if for all $x \in M$, we have $x^\omega = x^\omega x$. Aperiodic stabilisation algebras form a variety of finite stabilisation algebras.

Let M, N be two stabilisation algebras. We say that M *divides* N if M is a quotient of a stabilisation subalgebra of N. It follows from the definition that varieties of stabilisation algebras are closed under division. If S is a (possibly infinite) set of finite stabilisation algebras, the *variety generated by S* is the smallest variety of finite stabilisation algebras containing the elements of S.

▶ **Lemma 5.4.** *Let* **V** *be a variety generated by a set S of finite stabilisation algebras, and M be a finite stabilisation algebra. Then $M \in \mathbf{V}$ if and only if M divides a finite product of elements of S.*

Given a variety **V** of finite stabilisation algebras, let $\mathcal{V}(A)$ denote the set of recognisable downsets over A whose syntactic stabilisation algebra belongs to **V**. The correspondence $\mathbf{V} \to \mathcal{V}$ associates with each variety of finite stabilisation algebras a class of recognisable downsets.

Thus, each variety of recognisable downsets $\mathcal{V}$ is associated to the variety of finite stabilisation algebras **V** generated by the syntactic stabilisation algebras of downsets in $\mathcal{V}$. This defines a correspondence $\mathcal{V} \to \mathbf{V}$. The analog of the ordered version of Eilenberg's theorem can now be stated as follows:

▶ **Theorem 5.5.** *The correspondences $\mathcal{V} \to \mathbf{V}$ and $\mathbf{V} \to \mathcal{V}$ define mutually inverse bijective correspondences between varieties of finite stabilisation algebras and varieties of recognisable downsets.*

6 Profinite Stabilisation Algebra

The free profinite monoid on A, denoted $\widehat{A^*}$, can be defined as the completion of A^* for the *profinite metric*. See [4, 20, 21] for more information on this space.

We now prove the existence of free profinite stabilisation algebras. Taking the construction of free profinite monoids as a model, we define it as the completion $\widehat{F(A)}$ of $F(A)$ for an appropriate metric.

▶ **Definition 6.1.** A stabilisation algebra M *separates* two elements s and t of $F(A)$ if there is a morphism $\varphi : F(A) \to M$ such that $\varphi(s) \neq \varphi(t)$. For $s, t \in F(A)$, define

$$d(s,t) = \begin{cases} 0 & \text{if } s = t \\ 2^{-n(s,t)} & \text{otherwise} \end{cases}$$

where $n(s,t)$ is the minimum size of a finite stabilisation algebra separating s and t.

Note that d is well defined, since if $s \neq t \in F(A)$, then there is by $(\mathbf{A}_1)$ a finite stabilisation monoid in which the identity $s = t$ fails. Such a monoid can be viewed as a finite stabilisation algebra separating s and t. The following proposition gathers the properties of d and $\widehat{F(A)}$:

▶ **Proposition 6.2.**
(1) *d is an ultrametric distance.*
(2) *The operations on $F(A)$ are uniformly continuous and thus extend by continuity to $\widehat{F(A)}$.*
(3) *The resulting stabilisation algebra $\widehat{F(A)}$ is compact.*

The idempotent power. If M is a finite monoid, then for any $m \in M$ and $n \geqslant |M|$, we have $m^\omega = m^{n!}$ (where ω is the idempotent power). Since finite stabilisation algebras are in particular monoids where ω is the idempotent power, we obtain that for any $u \in F(A)$ and $n > 0$, $d(u^{n!}, u^\omega) \leqslant 2^{-n}$. Therefore, for any element $u \in F(A)$, the sequence $(u^{n!})_{n \in \mathbb{N}}$ converges in $\widehat{F(A)}$, to u^ω.

▶ **Proposition 6.3.** *The morphism* $\varphi : F(A) \to \widehat{A^*}$ *defined by* $\varphi(a) = a$, $\varphi(t^\omega) = t^\omega$ *and* $\varphi(t^\sharp) = t^\omega$ *is uniformly continuous, and therefore can be uniquely extended into a continuous morphism of stabilisation algebras* $\widehat{\varphi} : \widehat{F(A)} \to \widehat{A^*}$.

Notice however that this morphism does not coincide with the interpretation of regular cost functions as subsets of $\widehat{A^*}$ as done in [30].

The profinite metric can be relativised to any variety of stabilisation algebras to obtain the so-called pro-**V** metric. For $s, t \in F(A)$ and **V** a variety of stabilisation algebras, define $d_{\mathbf{V}}(s, t) = 2^{-|M|}$ where M is one of the smallest stabilisation algebras from **V** separating s and t and $d_{\mathbf{V}}(s, t) = 0$ if there is no such M. Remark that the metric d of Definition 6.1 corresponds to $d_{\mathbf{V}}$ where **V** is the variety of all finite stabilisation algebras. We also define an equivalence $s \sim_{\mathbf{V}} t$ by $d_{\mathbf{V}}(s, t) = 0$.

▶ **Proposition 6.4.** *For any variety* **V**, $\sim_{\mathbf{V}}$ *is a congruence on* $F(A)$ *and* $d_{\mathbf{V}}$ *is an ultrametric distance on* $F(A)/\sim_{\mathbf{V}}$.

We now define the pro-**V** stabilisation algebra $\widehat{F_{\mathbf{V}}(A)}$ as the completion of $F(A)/\sim_{\mathbf{V}}$ with respect to $d_{\mathbf{V}}$. As before, we can show that $\widehat{F_{\mathbf{V}}(A)}$ is compact and can be equipped with a structure of stabilisation algebra. The following result now follows from general results on profinite algebras.

▶ **Theorem 6.5.** *A finite A-generated stabilisation algebra belongs to* **V** *if and only if it is a continuous quotient of* $\widehat{F_{\mathbf{V}}(A)}$.

7   Duality, Equations and Identities

Stone duality tells us that every bounded distributive lattice $\mathcal{L}$ has an associated compact Hausdorff space, called its *dual space*. The dual space of the Boolean algebra of all *regular languages* of A^* [3] is the free profinite monoid on A.

A similar result holds for the lattice of regular cost functions, which, by Proposition 4.3, is isomorphic to the lattice of recognisable downsets under union and intersection.

▶ **Theorem 7.1.** *The dual space of the lattice of recognisable downsets of $F(A)$ is the space* $\widehat{F(A)}$.

7.1   Equations of Lattices

It is shown in [14] that any lattice of regular languages can be defined by a set of equations of the form $u \to v$, where u and v are profinite words. This result can also be extended to recognisable downsets.

Let $u, v \in \widehat{F(A)}$. We say that a recognisable downset I of $F(A)$ *satisfies the equation* $u \to v$ if $u \in \overline{I}$ implies $v \in \overline{I}$, where $\overline{I}$ denotes the topological closure of I.

A set $\mathcal{L}$ of recognisable downsets is *defined by a set E of equations* if the following property holds: a recognisable downset belongs to $\mathcal{L}$ if and only if it satisfies all the equations of E. We can now state our second main result.

▶ **Theorem 7.2.** *A set of recognisable downsets of $F(A)$ is a lattice of recognisable downsets if and only if it is defined by a set of equations of the form $u \to v$.*

The case of lattices of languages closed under quotients was also considered in [14]. The corresponding notion for lattices of downsets is to be closed under contexts.

A *profinite context* C on the finite alphabet A is an element of $F(\widehat{A \cup \{x\}})$ where $x \notin A$. If u is an element of $\widehat{F(A)}$, then $C[u]$ is also an element of $\widehat{F(A)}$, defined by replacing x by u and evaluating the operations ω and $\sharp$ in the stabilisation algebra $\widehat{F(A)}$.

▶ **Definition 7.3.** A recognisable downset of $F(A)$ *satisfies the equation* $u \leqslant v$ if, for all profinite contexts C, it satisfies the equation $C[u] \to C[v]$.

Equivalently, a stabilisation algebra *satisfies the equation* $u \leqslant v$ if, for all downsets J of M and for all contexts C, $C[v] \in J$ implies $C[u] \in J$. By density of $F(A)$ in $\widehat{F(A)}$, it is enough to consider contexts in Ctx_A for this definition. The notation $u = v$ is used as a shortcut for $u \leqslant v$ and $v \leqslant u$. We can now state:

▶ **Theorem 7.4.** *A set of recognisable downsets of $F(A)$ is a lattice of recognisable downsets closed under contexts if and only if it is defined by a set of equations of the form $u \leqslant v$.*

7.2   Identities of Varieties

Condition (V_2) of the definition of a variety allows one to use identities instead of equations.

Let B be an alphabet and let u and v be two elements of $\widehat{F(B)}$. We say that a recognisable downset I of $F(A)$ *satisfies the profinite identity* $u \leqslant v$ if, for each morphism $\gamma : F(B) \to F(A)$, I satisfies the equation $\widehat{\gamma}(u) \leqslant \widehat{\gamma}(v)$.

We use the term *identity* because, in this case, each letter of B can be replaced (through the morphism γ) by *any* element of $F(A)$.

In practice, it is more convenient to use the following characterisation. Let I be a recognisable downset and let M be its syntactic stabilisation algebra. Then I satisfies the identity $u \leqslant v$ if and only if for every continuous morphism $h : F(B) \to M$, one has $\widehat{h}(u) \leqslant_M \widehat{h}(v)$, where $\leqslant_M$ is the order of M.

▶ **Theorem 7.5.** *A class of finite stabilisation algebras (resp., recognisable downsets) is a variety if and only if it is defined by a set of identities of the form $u \leqslant v$.*

8   Examples of Equational Descriptions of Varieties and Lattices

In this section, we gather examples of varieties of regular cost functions and of sets of equations. First, it is interesting to see how the identification of regular languages with cost functions extends to varieties.

▶ **Proposition 8.1.** *If a positive variety of regular languages is defined by a set of identities E, then it is a variety of regular cost functions, defined by the set of identities $E \cup \{x^{\sharp} = x^{\omega}\}$. Conversely, if a variety of regular cost functions is defined by a set of identities E, then the variety of regular cost functions defined by $E \cup \{x^{\sharp} = x^{\omega}\}$ can be identified with a positive variety of languages.*

For instance, the variety of regular cost functions defined by $x^{\omega} = x^{\omega+1}$ and $x^{\sharp} = x^{\omega}$ contains only the characteristic functions of star-free languages [28].

Aperiodic Cost Functions

The variety of *aperiodic cost functions* is defined by the identity $x^{\omega} = x^{\omega+1}$. It contains recognisable downsets that are not languages, like $u \mapsto |u|_a$. This variety has a nice connection with the logics CFO and CLTL, first introduced in [16, 17] as a generalisation to cost functions of the logics FO and LTL on words. Indeed, the results of [16, 17] can be reformulated as follows:

▶ **Theorem 8.2.** *The variety of aperiodic cost functions coincides with the variety of CFO-definable cost functions and with the variety of CLTL-definable cost functions.*

Note that given a finite stabilisation algebra M, one can effectively test whether it verifies equations like $x^\omega = x^{\omega+1}$ or $x^\sharp = x^\omega$: it suffices to check that it stands for each x in M. It follows that one can effectively decide whether a regular cost function is CFO-definable (respectively CLTL-definable).

Temporal Cost Functions

Another interesting example is the class of temporal cost functions, first introduced in [11]. These functions allow one to count the number of occurrences of consecutive events. Many equivalent characterizations of these functions are known. In [11], the algebraic characterization is expressed in terms of the interplay between Green relations and stabilisation in the syntactic monoid, but it can be formulated in terms of equations as follows:

▶ **Theorem 8.3.** *Temporal cost functions over A form a lattice of regular cost functions, defined by the equations $(xy^\sharp z)^\sharp = (xy^\sharp z)^\omega$, for all $x, z \in F(A)$ and all $y \in F(A) - \{1\}$.*

Proof. Let M be the syntactic stabilisation monoid of a regular cost function f. An idempotent e is called *stable* if $e^\sharp = e$. The algebraic characterization from [11] states that f is temporal if and only if an idempotent $\mathcal{J}$-below a stable idempotent different from 1 is itself stable. Recall that the $\mathcal{J}$-order is defined by $e \leqslant_{\mathcal{J}} s$ if there exist $x, y \in M$ such that $e = xsy$. To show that our set of equations is equivalent to this characterization, it suffices to observe that an element is a stable idempotent if and only if it is of the form $s^\sharp$ for some s. This means that the characterization from [11] specifies that the idempotents of the form $(xs^\sharp z)^\omega$, with $s \neq 1$, are stable. Using Corollary 3.5, one can now lift these properties to $F(A)$, yielding the equations of the statement.   ◀

Commutative Cost Functions

The description of the variety of languages corresponding to commutative monoids is one of the first known examples of Eilenberg's correspondence between varieties of languages and varieties of monoids [13]. We prove below a similar result for cost functions.

Let us say that a finite stabilisation algebra M is *commutative* if for all $x, y \in M$, we have $xy = yx$. We will say that M is *$\sharp$-commutative* if it is commutative and for all $x, y \in M$, $x^\sharp y^\sharp = (xy)^\sharp$. A cost function is called *commutative* (resp., *$\sharp$-commutative*) if its syntactic stabilisation algebra is commutative (resp., $\sharp$-commutative).

▶ **Example 8.4.** The cost function $\mathrm{maxblock}_a$ on the alphabet $\{a, b\}$ defined by:

$$\mathrm{maxblock}_a(a^{n_1} b a^{n_2} b \cdots b a^{n_k}) = \max(n_1, n_2, \ldots, n_k).$$

is commutative but not $\sharp$-commutative. The downset representing this regular cost function consists in $\omega\sharp$-expressions on $\{a, b\}$ containing a subexpression of the form $u^\sharp$, where u is an $\omega\sharp$-expression on the alphabet $\{a\}$ with at least one occurrence of a. Its syntactic stabilisation algebra M has four elements: $0 \leqslant a \leqslant 1 \leqslant b$, all elements are idempotent and commute, and we have $a^\sharp = 0$, $x^\sharp = x$ for $x \neq a$, and $ab = b = ba$. It is not $\sharp$-commutative because $a^\sharp b^\sharp = 0b = 0$ and $(ab)^\sharp = b^\sharp = b$.

A stabilisation algebra is said to be *monogenic* if it can be generated by a single element. We will use freely the following useful lemma.

▶ **Lemma 8.5.** *Any finite ♯-commutative stabilisation algebra divides the product of its monogenic stabilisation subalgebras.*

The stabilisation monoid S_1 defined in Example 2.3 has three idempotent elements $0 < a < 1$ such that $0^♯ = a^♯ = 0$. It is also the syntactic stabilisation algebra of the function $f = u \mapsto |u|_a$. Let U_1^+ denote the stabilisation monoid with two idempotent elements $0 \leqslant 1$ such that $0^♯ = 0$.

▶ **Proposition 8.6.** *Let $\mathbf{J}_1^+$ be the variety of finite stabilisation algebras defined by the equations $x \leqslant 1$, $x^2 = x$, $xy = yx$ and $x^♯ y^♯ = (xy)^♯$. Then the corresponding variety of cost functions is generated by the functions $u \mapsto |u|_a$ for all letters a.*

Proof. Since S_1 is the syntactic stabilisation algebra of the function $u \mapsto |u|_a$, it is equivalent to prove that the variety of finite stabilisation algebras $\mathbf{V}$ generated by S_1 is equal to $\mathbf{J}_1^+$. Since S_1 satisfies all the equations of $\mathbf{J}_1^+$, the relation $\mathbf{V} \subseteq \mathbf{J}_1^+$ holds. To prove the opposite inclusion, consider a finite stabilisation algebra M of $\mathbf{J}_1^+$. By Lemma 8.5, M divides the product of its monogenic stabilisation subalgebras. But if $m \in M$, the stabilisation algebra generated by m is $\{1, m, m^♯\}$: indeed, the equations $m^2 = m$, $(m^♯)^2 = m^♯$ and the properties of a stabilisation monoid imply that $mm^♯ = m^♯ m = m^♯$. Thus, this stabilisation algebra is either $\{1\}, U_1^+$ or S_1. Since U_1^+ is a quotient of S_1, M actually divides a product of copies of S_1, and therefore $M \in \mathbf{V}$. Thus $\mathbf{V} = \mathbf{J}_1^+$. ◀

As stated earlier, if we add the equation $x^♯ = x^\omega$, we obtain the positive variety of regular languages corresponding to the variety of ordered monoids generated by the ordered monoid U_1^+ (see [19]).

▶ **Proposition 8.7.** *Let $\mathbf{Acom}$ be the variety of finite stabilisation algebras defined by the equations $x^\omega = x^{\omega+1}$, $xy = yx$ and $x^♯ y^♯ = (xy)^♯$. Then the corresponding variety of cost functions is generated by the functions $u \mapsto |u|_a$ and $\chi_{L_{a,k}}$ where $L_{a,k} = \{u \mid |u|_a = k\}$ for each $k \geqslant 0$ and each letter a.*

▶ **Proposition 8.8.** *Let $\mathbf{Com}$ be the variety of finite stabilisation algebras defined by the equations $xy = yx$ and $x^♯ y^♯ = (xy)^♯$. Then the corresponding variety of cost functions is generated by the functions $u \mapsto |u|_a$, $\chi_{L_{a,k}}$ and $\chi_{L_{k,n}}$, where $L_{a,k,n} = \{u \mid |u|_a \equiv k \bmod n\}$.*

9 Conclusion

We provide a new representation of regular cost functions as downsets of a free stabilisation algebra, an ordered algebraic structure. This new representation allows us to extend Eilenberg's variety theory, in its ordered version: varieties of regular cost functions correspond to varieties of finite stabilisation algebras and are characterised by profinite identities. Furthermore, we also extend the duality approach of [14] to this new setting, leading to profinite equational descriptions of lattices of regular cost functions. Finally, we give several examples of equational characterisations of classes of cost functions related to logic. We also investigate the extensions of commutative languages to regular cost functions. We uncover the role of a new identity, $x^♯ y^♯ = (xy)^♯$, in the study of these extensions.

These results confirm the pertinence and the usefulness of the theory of regular cost functions as a well-behaved quantitative generalisation of regular languages. They also open new perspective for the study of cost functions.

For instance, it would be interesting to extend other known characterisations of varieties of languages to the setting of cost functions. An emblematic example would be Simon's characterisation of piecewise testable languages.

────── **References** ──────

1 Jirí Adámek, Stefan Milius, Robert S. R. Myers, and Henning Urbat. Generalized eilenberg theorem I: local varieties of languages. In Anca Muscholl, editor, *FOSSACS 2014*, volume 8412 of *Lecture Notes in Comput. Sci.*, pages 366–380. Springer, 2014.

2 Jirí Adámek, Robert S. R. Myers, Henning Urbat, and Stefan Milius. Varieties of languages in a category. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 414–425. IEEE, 2015.

3 Jorge Almeida. Residually finite congruences and quasi-regular subsets in uniform algebras. *Portugaliæ Mathematica*, 46:313–328, 1989.

4 Jorge Almeida. *Finite semigroups and universal algebra*. World Scientific Publishing Co. Inc., River Edge, NJ, 1994. Translated from the 1992 Portuguese original and revised by the author.

5 Jean Berstel and Christophe Reutenauer. *Noncommutative rational series with applications*, volume 137 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2011.

6 Stephen L. Bloom. Varieties of ordered algebras. *J. Comput. System Sci.*, 13(2):200–212, 1976.

7 Mikolaj Bojanczyk. Recognisable languages over monads. In Igor Potapov, editor, *Developments in Language Theory – 19th International Conference, DLT 2015*, volume 9168 of *Lecture Notes in Comput. Sci.*, pages 1–13. Springer, 2015.

8 Mikolaj Bojanczyk. Recognisable languages over monads. *CoRR*, abs/1502.04898, 2015.

9 Thomas Colcombet. The theory of stabilisation monoids and regular cost functions. In *Automata, languages and programming. Part II*, volume 5556 of *Lecture Notes in Comput. Sci.*, pages 139–150, Berlin, 2009. Springer.

10 Thomas Colcombet. Regular cost functions, Part I: Logic and algebra over words. *Log. Methods Comput. Sci.*, 9(3):3:3, 47, 2013.

11 Thomas Colcombet, Denis Kuperberg, and Sylvain Lombardy. Regular temporal cost functions. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Automata, languages and programming, ICALP 2010, Part II*, volume 6199 of *Lecture Notes in Comput. Sci.*, pages 563–574. Springer, Berlin, 2010.

12 Manfred Droste, Werner Kuich, and Heiko Vogler, editors. *Handbook of weighted automata*. Monographs in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2009.

13 Samuel Eilenberg. *Automata, languages, and machines. Vol. B.* Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1976.

14 Mai Gehrke, Serge Grigorieff, and Jean-Éric Pin. Duality and equational theory of regular languages. In L. Aceto and al., editors, *ICALP 2008, Part II*, volume 5126 of *Lecture Notes in Comput. Sci.*, pages 246–257, Berlin, 2008. Springer.

15 Daniel Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *Internat. J. Algebra Comput.*, 4(3):405–425, 1994.

16 Denis Kuperberg. Linear temporal logic for regular cost functions. In *28th International Symposium on Theoretical Aspects of Computer Science*, volume 9 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 627–636. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2011.

17 Denis Kuperberg and Michael Vanden Boom. On the expressive power of cost logics over infinite words. In *Automata, languages, and programming. Part II*, volume 7392 of *Lecture Notes in Comput. Sci.*, pages 287–298. Springer, Heidelberg, 2012.

18   I. Mikhailova. A proof of Zhil'tsov's theorem on decidability of equational theory of epigroups, 2013. URL: `http://arxiv.org/abs/1310.5023`.

19   Jean-Éric Pin. A variety theorem without complementation. *Russian Mathematics (Izvestija vuzov.Matematika)*, 39:80–90, 1995.

20   Jean-Éric Pin. Profinite methods in automata theory. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009)*, pages 31–50. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2009.

21   Jean-Éric Pin. Equational descriptions of languages. *Int. J. Found. Comput. S.*, 23:1227–1240, 2012.

22   Christophe Reutenauer. Variétés d'algèbres et de séries rationnelles. In *1er Congrès Math. Appl. AFCET-SMF*, volume 2, pages 93–102. AFCET, 1978.

23   Christophe Reutenauer. Séries formelles et algèbres syntactiques. *J. Algebra*, 66(2):448–483, 1980.

24   Christophe Reutenauer. Séries rationnelles et algèbres syntactiques. Thèse de Doctorat d'État, Université Paris 6, 1980.

25   Marcel-Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4:245–270, 1961.

26   Marcel-Paul Schützenberger. Finite counting automata. *Information and Control*, 5:91–107, 1962.

27   Marcel-Paul Schützenberger. On a theorem of R. Jungen. *Proc. Amer. Math. Soc.*, 13:885–890, 1962.

28   Marcel-Paul Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8:190–194, 1965.

29   Imre Simon. Factorization forests of finite height. *Theoret. Comput. Sci.*, 72(1):65–94, 1990.

30   Szymon Toruńczyk. *Languages of profinite words and the limitedness problem.* Phd thesis, Computer Science Department, University of Warsaw, 2011.