The Theory of Stabilisation Monoids and Regular Cost Functions*

Thomas Colcombet

LIAFA/CNRS/Université Paris 7, Denis Diderot, France

Abstract. We introduce the notion of regular cost functions: a quantitative extension to the standard theory of regular languages.

We provide equivalent characterisations of this notion by means of automata (extending the nested distance desert automata of Kirsten), of history-deterministic automata (history-determinism is a weakening of the standard notion of determinism, that replaces it in this context), and a suitable notion of recognisability by stabilisation monoids. We also provide closure and decidability results.

1 Introduction

When considering standard regular languages (say on finite words), some results appear as cornerstones on which the whole theory is constructed. The first such kind of results are the equivalences between many different formalisms: non-deterministic automata, deterministic automata, recognisability by monoids, regular expressions, etc. The second one consists in the numerous closure properties that regular languages enjoy: union, intersection, projection (mapping under a length-preserving morphism), complementation, etc. From these facts one can derive a third kind of results: the equivalence with logical formalisms such as monadic (second-order) logic. Finally, all these properties do not come at an unaffordable price: emptiness is decidable, and hence the satisfaction of the logic is also decidable.

In this paper, we present a quantitative extension to the standard notion of regularity in which those cornerstone results still hold. We consider a quantitative notion of regularity which allows to attach non-negative integer values to words, such as the number of occurrences of a pattern, the length of segments, etc. One also possess some freedom for combining those values, e.g., using minimum or maximum. One can for instance describe the maximum number of occurrences of letter a that are not separated by a letter b. Those integer values are considered modulo an equivalence which preserves the existence of bounds, but does not preserve exact values – as opposed to the usual way one considers quantitative forms of automata. This is the price to pay for keeping all equivalences and closure properties.

Originally, this work aimed at unifying and reinterpreting some recent results from the literature. Let us review them.

* Supported by the ANR project JADE: 'Jeux et Automates, Décidabilité et Extensions'.

S. Albers et al. (Eds.): ICALP 2009, Part II, LNCS 5556, pp. 139–150, 2009.

First, in [9] Kirsten gives a new proof to the decidability of the (restricted) star-height problem[1]. This problem is known to be decidable from Hashiguchi [7], but with a very difficult proof. The first part in Kirsten's proof consists in reducing the star-height problem to a problem of limitedness: decide the existence of a bound for some function defined by means of a nested distance desert automata, a new form of automata introduced for this purpose. The second part consists in proving the decidability of this limitedness problem. This is done by turning this automata-related question into an algebraic one: the automaton is translated into a monoid equipped with a stabilisation operator $\sharp$. The limitedness problem becomes easy to decide in this presentation. Kirsten's paper is itself the continuation of a long line of research concerning distance automata, tropical semiring, desert automata, etc. [6,8,11,12,13,14,15,16,17].

Second, the paper [3] provides a study of an extension of the monadic second-order logic over infinite words with new 'bound' quantifiers such as: 'there exists a set of arbitrary large size satisfying some property'. The goal being different, the presentation is also significantly different, and getting results comparable to the ones in the present paper requires a translation that we cannot detail here. However, two new forms of automata are introduced in [3] as intermediate objects in the proofs, namely B-automata and S-automata. The class of B-automata corresponds essentially to the non-nested variant of the nested desert distance automata, while the class of S-automata is a new dual variant. The decidability of limitedness can be derived from this work but with a bad complexity (non-elementary, as opposed to [9]). Independently, B-automata were also introduced in [1] under the name of R-automata, and the decidability of the limitedness problem established using another technique, yielding better complexity.

Other applications of the technique have also been described. Still in this framework, the restricted star-height problem for trees has been shown decidable [4], and the Mostowski hierarchy problem[2] has been reduced to the corresponding limitedness problem over infinite trees [5], which remains open. The existence of a bound on the number of iterations necessary for reaching the fixpoint of a monadic second-order formula over words has been also shown decidable using distance automata [2].

Contribution. Our contribution can be roughly described as 1) a unification of the ideas in [9] and [3], and 2) the development of a suitable mathematical background and the establishment of new results in order to make this theory a complete extension of the standard theory of regular languages. Let us be more precise.

The first contribution lies in the definition of a cost function: cost functions are mappings from words (or from any set in general) to $\omega + 1$ quotiented by a suitable equivalence that preserves the notion of bound ($\approx$ in the paper). In our framework, cost functions can be seen as a refinement of the notion of language (each language can be seen as a cost function, while the converse is not true).

[1] Problem: given a regular language L of words and an integer k, is it possible to describe L with a regular expression using at most k nesting of Kleene stars?.

[2] The hierarchy induced by the number of priorities used by a non-deterministic parity automaton running on infinite trees.

We then introduce B- and S-automata, automata that accept cost functions rather than languages. Those are slight extensions of the automata in [3]. We establish the equivalence of the two forms of automata, via an elementary construction, as well as the equivalence with their history-deterministic form. The new notion of history-determinism is a weakening of the classical notion of determinism (deterministic automata are strictly weaker in this framework). It is needed for the further extension of the theory to trees. Quiet naturally, we call regular the cost functions described by one of these formalisms.

The second aspect of the theory that we develop is the algebraic formalism. We introduce the notion of stabilisation monoids: finite monoids equipped with a stabilisation operator, inspired from [9]. We develop a mathematical framework – new to the knowledge of the author – in order to define the semantics of stabilisation monoids. The key result here is the existence of unique semantics (that we call compatible mappings) for each stabilisation monoid[3]. Building on these notions, we introduce the notion of recognisable cost functions. As we may expect, these happen to be exactly the regular cost functions.

While describing the above objects, we prove the closure of regular cost functions under operations which correspond to union, intersection, projection and dual of projection in the world of languages. We also provide decision procedures subsuming the limitedness results from [9].

Structure of the paper. We present in Section 2 cost functions and the automata part of the theory. We present in Section 3 the algebraic framework, and the equivalent notion of recognisability.

Some Notations

As usual, we denote by ω the set of non-negative integers and $\omega + 1$ the set $\omega \cup \{\omega\}$. Those are ordered by $0 < 1 < \cdots < \omega$. The identity mapping over ω is *id*. Given a set E, E^ω is the set of sequences of ω-length of elements in E. Such sequences will be denoted by bold letters ($\boldsymbol{a}$, $\boldsymbol{b}$,...). We fix a *finite alphabet* $\mathbb{A}$ consisting of *letters*. The set of words over $\mathbb{A}$ is $\mathbb{A}^*$. The empty word is ε. The concatenation of a word u and word v is uv. The length of word u is $|u|$. The number of occurrences of a letter a in u is $|u|_a$.

2 Regular Cost Functions

We introduce in Section 2.1 the notion of cost function. We present B and S-automata in Section 2.2, and their history-deterministic form in Section 2.3. The key duality result is the subject of Section 2.4.

2.1 Cost Functions

A *correction function* is a mapping from ω to ω. From now, the symbols $\alpha, \alpha', \ldots$ implicitly designate correction functions. Given x, y in $\omega + 1$, $x \preccurlyeq_\alpha y$ holds

[3] This result is reminiscent of the one for infinite words stating that each finite Wilke algebra can be uniquely extended into an ω-semigroup.

if $x \leq \overline{\alpha}(y)$ in which $\overline{\alpha}$ is the extension of α with $\overline{\alpha}(\omega) = \omega$. For every set E, $\preccurlyeq_\alpha$ is extended to $(\omega+1)^E$ in a natural way by $f \preccurlyeq_\alpha g$ if $f(x) \preccurlyeq_\alpha g(x)$ for all $x \in E$, or equivalently $f \leq \overline{\alpha} \circ g$. Intuitively, f is dominated by g after it has been 'stretched' by α. One also writes $f \approx_\alpha g$ if $f \preccurlyeq_\alpha g$ and $g \preccurlyeq_\alpha f$.

Some elementary properties of $\preccurlyeq_\alpha$ are:

Fact 1. *If $\alpha \leq \alpha'$ and $f \preccurlyeq_\alpha g$, then $f \preccurlyeq_{\alpha'} g$. If $f \preccurlyeq_\alpha g \preccurlyeq_\alpha h$, then $f \preccurlyeq_{\alpha\circ\alpha} h$.*

Example 1. Over $\omega \times \omega$, maximum and sum are equivalent for the doubling correction function (for short, $(\max) \approx_{\times 2} (+)$). *Proof:* for all $x, y \in \omega$, $\max(x,y) \leq x + y \leq 2 \times \max(x,y)$.

Our second example concerns mappings from sequence of words to ω. Given words $u_1, \ldots, u_n \in \{a,b\}^*$, we have $|u_1 \ldots u_n|_a \approx_\alpha \max(|K|, \max_{i=1\ldots n} |u_i|_a)$ in which K is the set of indices i such that $|u_i|_a \geq 1$ and $\alpha(\theta) = \theta^2$. *Proof:* $\max(|K|, \max_{i=1\ldots n} |u_i|_a) \leq |u|_a \leq \Sigma_{i\in K}|u_i|_a \leq (\max(|K|, \max_{i=1\ldots n} |u_i|_a))^2$.

One also defines $f \preccurlyeq g$ (resp. $f \approx g$) to hold if $f \preccurlyeq_\alpha g$ (resp. $f \approx_\alpha g$) for some α. A *cost function* (over a set E) is an equivalence class of $\approx$ (i.e., a set of mappings from E to $\omega + 1$). The relation $\preccurlyeq$ has other characterisations:

Proposition 1. *For all f, g from E to $\omega+1$, the following items are equivalent:*

- $f \preccurlyeq g$,
- $\forall n \in \omega . \exists m \in \omega . \forall x \in E . g(x) \leq n \to f(x) \leq m$, *and;*
- *for all $X \subseteq E$, $g|_X$ is bounded implies $f|_X$ is bounded.*

The last characterisation shows that the relation $\approx$ is an equivalence relation that preserves the existence of bounds. Indeed, all this theory can be seen as an automata theoretic method for proving the existence/non-existence of bounds.

Cost functions over some set E ordered by $\preccurlyeq$ form a lattice. Given a subset $X \subseteq E$, one denotes by χ_X its *characteristic mapping* defined by $\chi_X(x) = 0$ if $x \in X$, ω otherwise. It is easy to see that for all $X, Y \subseteq E$, $\chi_X \preccurlyeq \chi_Y$ iff $Y \subseteq X$. To this respect, the lattice of cost functions is a refinement of the lattice of subsets of E equipped with the superset ordering. Keeping this in mind, the notion of regular cost function developed in the paper is an extension of the standard notion of regular language. This extension is strict as soon as E is infinite: there are cost functions that are not equivalent to any characteristic mapping. Consider for instance the size mapping over words, or the number of occurrences of some letter.

2.2 Automata

We present here the automata model we use. A *cost automaton* (that can be either a *B-automaton* or an *S-automaton*) is a tuple $\langle Q, \mathbb{A}, In, Fin, \Gamma, \Delta\rangle$ in which Q is a finite set of *states*, $\mathbb{A}$ is the alphabet, *In* and *Fin* are respectively the set of *initial* and *final* states, Γ is a finite set of *counters*, and $\Delta \subseteq Q \times A \times \{\epsilon, i, r, c\}^\Gamma \times Q$ is the set of transitions. The idea behind the letter in $\{\epsilon, i, r, c\}^\Gamma$ (called an *action*) is that each counter (the value of which ranges over ω) can either be left unchanged (ϵ), be incremented by one (i), be reset to 0 (r), or be checked (c). A *run* σ of an automaton over a word $a_1 \dots a_n$ is defined as a sequence $q_0, a_1, c_1, q_1, \dots, q_{n-1}, a_n, c_n, q_n$ such that q_0 is initial, q_n is final and for all $i = 1 \dots n$, $(q_{i-1}, a_i, c_i, q_i) \in \Delta$. Given a run σ, each counter $\iota \in \Gamma$ is initialized with value 0 and evolves from left to right according to $c_i(\iota)$: if $c_i(\iota)$ is ϵ or c, the value is left unchanged, if it is i, it is incremented by 1, if it is r, the counter is reset. The set $C(\sigma) \subseteq \omega$ is the set of values taken by the counters when checked (i.e., the value of counter ι when $c_i(\iota) = c$). The difference between B-automata and S-automata comes from their dual semantics, $[\![\cdot]\!]_B$ and $[\![\cdot]\!]_S$ respectively:

$$\begin{aligned} \text{for all } u \in \mathbb{A}^*, \quad [\![\mathcal{A}]\!]_B(u) &= \inf\{\sup C(\sigma) \ : \ \sigma \text{ run over } u\} \ , \\ \text{and,} \quad [\![\mathcal{A}]\!]_S(u) &= \sup\{\inf C(\sigma) \ : \ \sigma \text{ run over } u\} \ , \end{aligned}$$

in which we use the standard convention that $\inf \emptyset = \omega$ and $\sup \emptyset = 0$. Remark that if $\mathcal{A}$ is a non-deterministic finite automaton in the standard sense, accepting the language L, then it can be seen as a cost automaton without counters. Seen as a B-automaton, $[\![\mathcal{A}]\!]_B(u) = \chi_L$, while seen as an S-automaton $[\![\mathcal{A}]\!]_S(u) = \chi_{\mathbb{A}^* \setminus L}$.

Remark 1 (variants). The other similar automata known from the literature can essentially be seen as special instances of the above formalism. The B-automata and S-automata in [3] use only actions in $\{\epsilon, i, cr\}$ in which cr is an atomic operation that checks the counter and immediately resets it. The models are equivalent but the history-determinism (see below) cannot be achieved for S-automata in this restricted form. The *hierarchical automata* correspond to the case when $\Gamma = \{1, \dots, n\}$ and for all transitions (p, a, c, q), if for all $i \in \Gamma$, if $c(i) \neq \epsilon$ implies $c(j) = r$ for all $j < i$. The *nested distance desert automata* of Kirsten corresponds to hierarchical B-automata that use actions in $\{\epsilon, ic, r\}$ in which ic is an atomic operation which increments the counter and immediately checks it. The *R-automata* in [1] use also actions in $\{\epsilon, ic, r\}$, but without the hierarchical constraint. All those models are equivalent, up to $\approx$.

We conclude the section by showing some easy closure properties. Given a mapping f from $\mathbb{A}^*$ to $\omega + 1$ and a length-preserving morphism h from $\mathbb{A}^*$ to $\mathbb{B}^*$ ($\mathbb{B}$ is another alphabet) the *inf-projection* and *sup-projection* of f with respect to h are the mappings $f_{\inf,h}$ and $f_{\sup,h}$ from $\mathbb{B}^*$ to $\omega + 1$ defined for $v \in \mathbb{B}^*$ by:

$$f_{\inf,h}(v) = \inf\{f(u) \ : \ h(u) = v\} \quad \text{and} \quad f_{\sup,h}(v) = \sup\{f(u) \ : \ h(u) = v\}.$$

By simply adapting the standard constructions for intersection, union, and projection of non-deterministic automata, we get:

Proposition 2. *The mappings accepted by B-automata (resp. S-automata) are closed under* min *and* max*. The mappings accepted by B-automata (resp. S-automata) are closed under inf-projection (resp. sup-projection).*

2.3 History-Determinism

In general B and S-automata cannot be determinised (even modulo $\approx$). We consider here automata which possess a weaker property: history-determinism (note that this notion is meaningful even for other kinds of non-deterministic automata). Informally, a non-deterministic automaton is history-deterministic if it possible to choose deterministically the run while accepting an equivalent function. The subtlety comes from the fact that cost automata do not have a sufficient memory for 'implementing' this deterministic choice. History-determinism can be seen as a semantic notion of determinism as opposed to the standard notion that we can refer to as state-determinism[4]. This notion is required for the extension of the theory to trees.

Formally, let us fix ourselves a cost automaton (either B or S) with unique initial state $\mathcal{A} = \langle Q, \mathbb{A}, \{q_0\}, Fin, \Gamma, \Delta\rangle$. A *translation strategy*[5] for $\mathcal{A}$ is a mapping δ from $\mathbb{A}^* \times \mathbb{A}$ to Δ which tells deterministically how to construct a run of $\mathcal{A}$ over a word. One defines the *run of $\mathcal{A}$ over the word u driven by δ* inductively as follows: if $u = \varepsilon$, the run is q_0. If u is of the form va, the run is the run of $\mathcal{A}$ over v driven by δ prolonged with the transition $\delta(v, a)$ (if this procedure does not provide a valid run over u, then there is no run driven by δ over this entry). If $\mathcal{A}$ is a B-automaton the value $[\![\mathcal{A}]\!]_B^{\delta}(u)$ is $\sup C(\sigma)$ where σ is the run of $\mathcal{A}$ over u driven by δ, and ω if there is no such run. If $\mathcal{A}$ is an S-automaton the value $[\![\mathcal{A}]\!]_S^{\delta}(u)$ is $\inf C(\sigma)$ where σ is the run of $\mathcal{A}$ over u driven by δ, and 0 if there is no such run.

A B-automaton is *history-deterministic* if there exists α and for all $n \in \omega$ a translation strategy δ_n such that for all words u, $[\![\mathcal{A}]\!]_B(u) \leq n$ implies $[\![\mathcal{A}]\!]_B^{\delta_n}(u) \leq \alpha(n)$. An S-automaton is *history-deterministic* if there exists α and for all $n \in \omega$ a translation strategy δ_n such that for all word u, $[\![\mathcal{A}]\!]_S(u) \geq \alpha(n)$ implies $[\![\mathcal{A}]\!]_S^{\delta_n}(u) \geq n$. In other words, the automaton, when driven by δ, computes an $\approx_\alpha$-equivalent function.

2.4 Duality and Regularity

Duality relates all the above notions together. It is central in the theory.

Theorem 1 (duality). *A cost function over words is accepted by an [history-deterministic] [hierarchical] B-automaton, iff it is accepted by an [history-deterministic] [hierarchical] S-automaton. Those equivalences are effective and of elementary complexity. Such cost functions are called* regular.

In fact, the proof of Theorem 1 and Theorem 3 below are interdependent. Indeed, the way to transform a cost function accepted by a B-automaton into a cost function accepted by an S-automaton (and vice-versa) is to transform it first into a recognisable cost function, and only then to construct an S-automaton. The results have been separated in this abstract for being easier to present.

One can remark that in the absence of counters, translating a B-automaton into an S-automaton (and vice-versa) is easy to achieve by using any complementation construction for standard non-deterministic automata. Hence Theorem 1

[4] In state-determinism, given the current state and a letter, there is only one possible transition, while in history-determinism, given the prefix of word seen so far (the history), and a letter, it is possible to uniquely choose one transition.

[5] The name comes from the game theoretic part which is not developed here.

can be seen as a replacement for both the results of complementation and determinisation in the classical theory of regular languages.

Even if not explicitly stated, the equivalence between [hierarchical] B-automata and [hierarchical] S-automata, can be derived from the results in [3]. However, using the proof in [3] entails a long theoretical detour, and the constructions in [3] give a non-elementary blowup in the number of states. Furthermore, the notion of history-determinism has no equivalent in [3].

3 Stabilisation Monoids and Recognisable Cost Functions

In this section we describe our algebraic characterisation of regular cost functions. The core algebraic object is the stabilisation monoid that we describe in Section 3.1. In Sections 3.2, 3.3 and 3.4, we show how to attach semantics to stabilisation monoids. In Section 3.5 we introduce recognisability, state that it is equivalent to regularity and give a decidability result.

3.1 Stabilisation Monoids

A *monoid* $\mathbf{M} = \langle M, \cdot \rangle$ is a set M equipped with an associative operation $\cdot$ that has a *neutral element* 1, i.e., such that $1 \cdot x = x \cdot 1 = x$ for all $x \in M$. One extends the product to products of arbitrary length by defining π from M^* to M by $\pi(\varepsilon) = 1$ and $\pi(ua) = \pi(u) \cdot a$. An idempotent in $\mathbf{M}$ is an element $e \in M$ such that $e \cdot e = e$. One denotes by $E(\mathbf{M})$ the set of idempotents in $\mathbf{M}$. An *ordered monoid* $\langle M, \cdot, \leq \rangle$ is a monoid $\langle M, \cdot \rangle$ together with an order $\leq$ over M such that the product $\cdot$ is *compatible* with $\leq$; i.e., $a \leq a'$ and $b \leq b'$ implies $a \cdot b \leq a' \cdot b'$.

We are now ready to introduce the new notion of stabilisation monoid.

Definition 1. *A* stabilisation monoid $\langle M, \cdot, \leq, \sharp \rangle$ *is an ordered monoid* $\langle M, \cdot, \leq \rangle$ *together with an operator* $\sharp \colon E(\mathbf{M}) \to E(\mathbf{M})$ *(called* stabilisation*) such that:*

- *for all* $a, b \in M$ *with* $a \cdot b \in E(\mathbf{M})$ *and* $b \cdot a \in E(\mathbf{M})$, $(a \cdot b)^\sharp = a \cdot (b \cdot a)^\sharp \cdot b$;[6]
- *for all* $e \in E(\mathbf{M})$, $(e^\sharp)^\sharp = e^\sharp \leq e$;
- *for all* $e \leq f$ *in* $E(\mathbf{M})$, $e^\sharp \leq f^\sharp$;
- $1^\sharp = 1$.

From now, we consider that all stabilisation monoids are finite*.*

The intuition is that $e^\sharp$ represents what is the value of e^n when n becomes 'very large'. This idea – which is incompatible with the classical view on monoids – fits well with the following consequences of the definition:

$$\text{for all } e \in E(\mathbf{M}), \qquad e^\sharp = e \cdot e^\sharp = e^\sharp \cdot e = e^\sharp \cdot e^\sharp = (e^\sharp)^\sharp \ .$$

Most of the remaining of the section is devoted to the formalisation of this intuition. This requires the development of a suitable mathematical framework. This approach is then validated by Theorem 2 which associates unique semantics to stabilisation monoids.

[6] This equation states that $\sharp$ is a *consistent mapping* in the sense of [9,10].

3.2 Cost Sequences

In order to give quantitative semantics to stabilisation monoids, the basic object is not the element of the monoid, but sequences of such elements. New relations $\preceq_\alpha$ and $\sim_\alpha$ are used to relate such sequences together. Those are tightly connected to $\preccurlyeq_\alpha$ and $\approx_\alpha$ (see Section 3.3 for a formalisation of this link).

From now, θ and θ' implicitely range over ω. Given an ordered set $(E, \leq)$, a correction function α, and two sequences $\boldsymbol{a}, \boldsymbol{b} \in E^\omega$, define $\boldsymbol{a} \preceq_\alpha \boldsymbol{b}$ to hold when:

$$\forall\theta.\forall\theta'. \quad \alpha(\theta) \leq \theta' \rightarrow \boldsymbol{a}(\theta) \leq \boldsymbol{b}(\theta')\ .$$

We set $\sim_\alpha$ to be $\preceq_\alpha \cap \succeq_\alpha$. The following fact is easy (analogue to Fact 1):

Fact 2. *If $\alpha \leq \alpha'$ and $\boldsymbol{a} \preceq_\alpha \boldsymbol{b}$, then $\boldsymbol{a} \preceq_{\alpha'} \boldsymbol{b}$. If $\boldsymbol{a} \preceq_\alpha \boldsymbol{b} \preceq_\alpha \boldsymbol{c}$, then $\boldsymbol{a} \preceq_{\alpha\circ\alpha} \boldsymbol{c}$.*

The mapping α is used as a parameter of 'precision' for $\sim$ and $\preceq$. The above fact states that using one transitivity step costs precision. (In practice, when doing proofs, we omit the correction function subscript, and rather ensure that the proofs conform to a structural property – very natural at use – ensuring that the length of chains of transitivity steps are bounded.)

A sequence $\boldsymbol{a}$ is said *α-non-decreasing* if $\boldsymbol{a} \preceq_\alpha \boldsymbol{a}$. Fact 3 is for helping intuition: it shows that one can almost think of α-non-decreasing sequences as if those were non-decreasing functions, and simplify the relation $\preceq_\alpha$ in this case:

Fact 3. *Every α-non-decreasing sequence is $\sim_\alpha$-equivalent to a non-decreasing sequence. If $\boldsymbol{a}$ and $\boldsymbol{b}$ are non-decreasing, then $\boldsymbol{a} \preceq_\alpha \boldsymbol{b}$ iff $\boldsymbol{a} \leq \boldsymbol{b} \circ \alpha$.*

The above inequality $\boldsymbol{a} \leq \boldsymbol{b} \circ \alpha$ conveys the important intuition that $\boldsymbol{a}$ is 'dominated' by $\boldsymbol{b}$ after 'shrinking' its coordinates (by α). This has to be compared to the definition of $\preccurlyeq_\alpha$ in which the correction function is used for 'stretching'.

From now, we identify each element $a \in E$ with the sequence constant equal to a. According to Fact 3, the relation $\preceq_\alpha$ (whatever is α) coincide with $\leq$ over those sequences. Hence the α-non-decreasing sequences equipped with the relation $\preceq_\alpha$ can be seen as a refinement of $(E, \leq)$.

We introduce now an important tool: α-monotonic mappings. This notion simplifies a lot the work with the $\preceq_\alpha$ and $\sim_\alpha$ relations. Given two ordered sets $(E, \leq)$ and $(F, \leq)$, a mapping f from E to F^ω is said *α-monotonic* if

$$\forall a, b \in E. \quad a \leq b \rightarrow f(a) \preceq_\alpha f(b)\ .$$

You can remark that in particular, for each $a \in E$, since $a \leq a$, we have $f(a) \preceq_\alpha f(a)$, and hence $f(a)$ is α-non-decreasing. Every α-monotonic f from E to F^ω can be turned into a mapping $\tilde{f}$ from E^ω to F^ω by setting:

$$\text{for all } \boldsymbol{a} \in E^\omega \text{ and all } \theta \in \omega, \qquad \tilde{f}(\boldsymbol{a})(\theta) = f(\boldsymbol{a}(\theta))(\theta)\ .$$

The following proposition discloses some key properties of α-monotonicity:

Proposition 3. *Let $f : E \rightarrow F^\omega$ be α-monotonic and $\boldsymbol{a}, \boldsymbol{b} \in E^\omega$, then:*

$$\boldsymbol{a} \preceq_\alpha \boldsymbol{b} \qquad \textit{implies} \qquad \tilde{f}(\boldsymbol{a}) \preceq_\alpha \tilde{f}(\boldsymbol{b})\ .$$

In particular, if $f : E \rightarrow F^\omega$ and $g : F \rightarrow G^\omega$ are α-monotonic, then $\tilde{g} \circ f$ is α-monotonic. Furthermore $\widetilde{(\tilde{g} \circ f)} = \tilde{g} \circ \tilde{f}$.

3.3 Relationship between $\preceq_\alpha$ and $\preccurlyeq_\alpha$

As mentioned above, $\preceq_\alpha$ and $\preccurlyeq_\alpha$ are tightly connected. We introduce in this section some useful notations and formalise this link in Proposition 4.

Given an ordered set $(E, \leq)$, an *ideal* is a subset $I \subseteq E$ such that for all $a \in I$ and $b \leq a$, $b \in I$. Its complement in E is $\overline{I}$. Given $a \in E$, the *ideal generated by* a is $I_a = \{b \in E \ : \ b \leq a\}$. Given a sequence $\boldsymbol{a} \in E^\omega$ and an ideal I, set $I[\boldsymbol{a}] = \sup\{\theta + 1 \ : \ \boldsymbol{a}(\theta) \in I\}$[7] and $\boldsymbol{a} \in E^\omega$, set $\overline{I}[\boldsymbol{a}] = \inf\{\theta \ : \ \boldsymbol{a}(\theta) \in \overline{I}\}$.

One goes back and forth between $\preceq_\alpha$ and $\preccurlyeq_\alpha$ using Proposition 4:

Proposition 4. *For all $\boldsymbol{a}, \boldsymbol{b} \in E^\omega$, $\boldsymbol{a} \preceq_\alpha \boldsymbol{b}$ iff $\overline{I}[\boldsymbol{a}] \succcurlyeq_\alpha I[\boldsymbol{b}]$ for all ideal $I \subseteq E$.*

3.4 Compatible Mappings

In this section, we capture the semantics of stabilisation monoids via the notion of compatible mappings (see definition below). We establish the existence and unicity of this semantics (Theorem 2).

Definition 2. *Given a stabilisation monoid $\mathbf{M} = \langle M, \cdot, \leq, \sharp \rangle$, a mapping ρ from M^* (words over M) to M^ω is* compatible with $\mathbf{M}$ *if for some α we have:*

Monotonicity. *ρ is α-monotonic,*
(M^* is ordered by $a_1 \dots a_m \leq b_1 \dots b_n$ if $m = n$ and $a_i \leq b_i$ for all i)
Letter. *for all $a \in M$, $\rho(a) \sim_\alpha a$, and $\rho(\varepsilon) \sim_\alpha 1$,*
(a and 1 denote the constant sequences equal to a and 1 respectively)
Product. *for all $a, b \in M$, $\rho(ab) \sim_\alpha a \cdot b$,*
($a \cdot b$ denotes the constant sequence equal to $a \cdot b$)
Stabilisation. *for all $e \in E(\mathbf{M})$, $m \in \omega$, $\rho(e^m) \sim_\alpha (e^\sharp|_m e)$,*
(e^m denotes the word consisting of m occurrences of the letter e)
(for $a \leq b$ in M, set $(a|_m b) \in M^\omega$ to map $[0, m)$ to a and $[m, \omega)$ to b)
Substitution. *for all $u_1, \dots, u_n \in M^*$, $n \in \omega$, $\rho(u_1 \dots u_n) \sim_\alpha \tilde{\rho}(\rho(u_1) \dots \rho(u_n))$.*
(in which $\rho(u_1) \dots \rho(u_n)$ is naturally seen as an α-non-decreasing sequence of words instead of a word over α-non-decreasing sequences)

Example 2. Consider the stabilisation monoid $\mathbf{M}$ with three elements $\bot \leq a \leq b$, for which the product is defined by $x \cdot y = \min_{\leq}(x, y)$ (hence $b = 1$), and the stabilisation by $b^\sharp = b$ and $a^\sharp = \bot^\sharp = \bot$. Given a word $u \in \{\bot, a, b\}^*$, one sets:

$$\rho(u) = \begin{cases} b & \text{if } u \in b^* \\ \bot|_{|u|_a} a & \text{if } u \in b^*(ab^*)^+ \\ \bot & \text{otherwise.} \end{cases}$$

The mapping ρ is compatible with $\mathbf{M}$:

Monotonicity. We prove *id*-monotonicity. Let $u \leq v$. If $v \in b^+$ then $\rho(u) \leq b = \rho(v)$ (since $\rho(v) = b$ is maximal), i.e., $\rho(u) \preceq_{id} \rho(v)$. If u contains the letter $\bot$ then $\rho(u) = \bot \leq \rho(v)$. Otherwise, since $u \leq v$, u and v contain at least one occurrence of a, and no occurrence of $\bot$. Hence $\rho(u) = (\bot|_{|u|_a} a)$ and $\rho(v) = (\bot|_{|v|_a} a)$. But since $u \leq v$, $|u|_a \geq |v|_a$. We get that $\rho(u) \preceq_{id} \rho(v)$.

[7] The $+1$ makes the theory more smooth, e.g., for Proposition 4.

Letter. We have $\rho(b) = b$, $\rho(\bot) = \bot$ and $\rho(a) = \bot|_1 a$. This implies $\rho(a) \sim_\alpha a$ for $\alpha(\theta) = \max(1, \theta)$.

Product. The only non trivial case is $\rho(aa) = (\bot|_2 a) \sim_\alpha (\bot|_1 a) = a \cdot a$ which holds for $\alpha(\theta) = \max(2, \theta)$.

Stabilisation. Every element is an idempotent. Let $m \geq 1$. For all $x \in M$, $\rho(x^m) = x^\sharp|_m x$ by definition.

Substitution. Let $u_1, \ldots, u_n \in M^*$ and $u = u_1 \ldots u_n$. If for all i, $u_i \in b^*$, then $\rho(u) = b = \tilde{\rho}(\rho(u_1) \ldots \rho(u_n))$. If the letter $\bot$ occurs in some u_i, then $\rho(u) = \bot = \tilde{\rho}(\rho(u_1) \ldots \rho(u_n))$. Otherwise u contains no $\bot$, and at least one occurrence of a. Let K be the set of indices i such that u_i contains an occurrence of a. By applying the definition of ρ and $\tilde{\rho}$ we claim that

$$\tilde{\rho}(\rho(u_1) \ldots \rho(u_n)) = \bot|_{\max(|K|, \max_{i \in K} |u_i|_a)} a \ ,$$

indeed if $\theta < |u_i|_a$ for some i, then $\rho(u_i)(\theta) = \bot$, and as a consequence $\tilde{\rho}(\rho(u_1) \ldots \rho(u_n))(\theta) = \bot$. And otherwise, if $\theta < |K|$, $(\rho(u_1) \ldots \rho(u_n))(\theta)$ contains no occurrences of $\bot$, but $|K|$-many occurrences of a, and hence once more $\tilde{\rho}(\rho(u_1) \ldots \rho(u_n))(\theta) = \bot$. Finally, if $\theta \geq \max(|K|, \max_{i \in K} |u_i|_a)$, then $(\rho(u_1) \ldots \rho(u_n))(\theta)$ contains no occurrences of $\bot$ and at most θ occurrences of a. We obtain $\tilde{\rho}(\rho(u_1) \ldots \rho(u_n))(\theta) = a$. Using Example 1 one has $\max(|K|, \max_{i \in K} |u_i|_a) \approx_\alpha |u|_a$ (for $\alpha(\theta) = \theta^2$), and consequently using Proposition 4, $\rho(u) \sim_\alpha \tilde{\rho}(\rho(u_1) \ldots \rho(u_n))$.

Remark 2. Let us state the link with the standard monoids. Consider a monoid $\mathbf{M} = \langle M, \cdot \rangle$. It can be turned into a stabilisation monoid $\langle M, \cdot, \leq, \sharp \rangle$ by a) setting $\leq$ to be the equality, and b) setting $e^\sharp = e$ for all idempotents e. In this case, it is easy to see that defining for all words $u \in M^*$, $\rho(u)$ to be the sequence constant equal to $\pi(u)$ provides a compatible mapping (for $\alpha = id$, i.e., $\sim_\alpha$ is the equality). According to Theorem 2 below, this is the only possible mapping compatible with $\langle M, \cdot, \leq, \sharp \rangle$.

Theorem 2 states that stabilisation monoids have unique semantics. It is reminiscent of the existence of unique extensions of finite Wilke algebras into ω-semigroups in the theory of regular languages of infinite words.

Theorem 2. *For every stabilisation monoid, there exists a mapping ρ compatible with it. Furthermore it is unique up to $\sim$.*

3.5 Recognisability

We now define the notion of recognisability for cost-functions.

(We use the definitions of Section 3.3.) Given a stabilisation monoid $\mathbf{M} = \langle M, \cdot, \leq, \sharp \rangle$, a length-preserving morphism h from $\mathbb{A}^*$ to M^*, and an ideal $I \subseteq M$, the triple $\mathbf{M}, h, I$ *recognises* the mapping $f : \mathbb{A}^* \to \omega + 1$ if there exists α such that for all $u \in \mathbb{A}^*$, $f(u) \approx_\alpha I[\rho(h(u))]$ in which ρ is a mapping compatible with M. A cost function from $\mathbb{A}^*$ to $\omega + 1$ is *recognisable* if some (equivalently all) function in the class are recognised by some $\mathbf{M}, h, I$.

Example 3. For $\mathbb{A} = \{a, b\}$, the function $|\cdot|_a$ which counts the number of occurrences of a in a word is recognisable. For this, consider the monoid of Example 2, the morphism defined by $h(a) = a, h(b) = b$, and the ideal $I = \{\perp\}$. Then we have $|u|_a = I[\rho(u)]$ for all $u \in \mathbb{A}^*$. This means that $|\cdot|_a$ is recognisable.

As one can expect, recognisability and regularity coincide:

Theorem 3. *A cost function over words is regular iff it is recognisable.*

As mentioned above, this theorem and Theorem 1 are proved at the same time. This proof is much more involved than the equivalent one for regular languages.

We conclude our description by a decidability result.

Theorem 4. *The relation $\preccurlyeq$ is decidable over recognisable cost functions.*

This decidability result extends previous results. For instance, the *boundedness problem* (deciding the existence of $n \in \omega$ such that $f(u) \leq n$ for all words u) corresponds to $f \preccurlyeq 0$. The standard *limitedness problem* (the boundedness over the support of the function) corresponds to $f \preccurlyeq \chi_L$ where L is $\{u : f(u) < \omega\}$.

4 Conclusion

We have introduced the notion of regular cost functions over words: equivalence classes of functions from words to $\omega + 1$. We have shown that those cost functions enjoy many equivalent representations: algebraic and automata theoretic. This paper is mainly oriented toward the algebraic part, in particular with Theorem 2 which shows that stabilisation monoids have a semantics independent from the automata counterpart. From those equivalences we obtain that the class of regular cost functions enjoy closure under min, max, inf-projection and sup-projection. From those closure properties, it is possible to derive an equivalence with a suitable extension of monadic second-order logic (not presented in the paper). We also provide a decision procedures for the $\preccurlyeq$ relation, and as a consequence the equivalence of cost functions. This result generalises the decidability of the limitedness problem in [9] and [1].

The results were carefully stated so that the extension of the theory to trees is possible (subject of a following paper). In particular we have introduced the notion of history-determinism, a semantic notion which replaces the classical notion of determinism in this framework. Let us finally remark that this whole framework can be extended without any problem to infinite words.

Acknowledgment

I am deeply grateful to Achim Blumensath, Olivier Carton, Pierre-Louis Curien, Denis Kuperberg, Christof Löding, Sylvain Lombardy and Konrad Zdanowski for their comments and helpful discussions on this work.

References

1. Abdulla, P.A., Krcál, P., Yi, W.: R-automata. In: van Breugel, F., Chechik, M. (eds.) CONCUR 2008. LNCS, vol. 5201, pp. 67–81. Springer, Heidelberg (2008)
2. Blumensath, A., Otto, M., Weyer, M.: Boundedness of monadic second-order formulae over finite words. In: ICALP 2009 (2009)
3. Bojańczyk, M., Colcombet, T.: Bounds in omega-regularity. In: LICS 2006, pp. 285–296 (2006)
4. Colcombet, T., Löding, C.: The nesting-depth of disjunctive mu-calculus for tree languages and the limitedness problem. In: Kaminski, M., Martini, S. (eds.) CSL 2008. LNCS, vol. 5213, pp. 416–430. Springer, Heidelberg (2008)
5. Colcombet, T., Löding, C.: The non-deterministic mostowski hierarchy and distance-parity automata. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 398–409. Springer, Heidelberg (2008)
6. Hashiguchi, K.: Limitedness theorem on finite automata with distance functions. J. Comput. Syst. Sci. 24(2), 233–244 (1982)
7. Hashiguchi, K.: Algorithms for determining relative star height and star height. Inf. Comput. 78(2), 124–169 (1988)
8. Hashiguchi, K.: Improved limitedness theorems on finite automata with distance functions. Theor. Comput. Sci. 72(1), 27–38 (1990)
9. Kirsten, D.: Distance desert automata and the star height problem. RAIRO 3(39), 455–509 (2005)
10. Kirsten, D.: Distance desert automata and star height substitutions. Habilitation, Universität Leipzig, Fakultät für Mathematik und Informatik (2006)
11. Leung, H.: Limitedness theorem on finite automata with distance functions: An algebraic proof. Theor. Comput. Sci. 81(1), 137–145 (1991)
12. Leung, H.: The limitedness problem on distance automata: Hashiguchi's method revisited. Theoretical Computer Science 310(1-3), 147 (2004)
13. Simon, I.: Limited subsets of a free monoid. In: FOCS, pp. 143–150 (1978)
14. Simon, I.: Recognizable sets with multiplicities in the tropical semiring. In: Koubek, V., Janiga, L., Chytil, M.P. (eds.) MFCS 1988, vol. 324, pp. 107–120. Springer, Heidelberg (1988)
15. Simon, I.: On semigroups of matrices over the tropical semiring. ITA 28(3-4), 277–294 (1994)
16. Weber, A.: Distance automata having large finite distance or finite ambiguity. Mathematical Systems Theory 26(2), 169–185 (1993)
17. Weber, A.: Finite-valued distance automata. Theor. Comput. Sci. 134(1), 225–251 (1994)