Regular Temporal Cost Functions

Thomas Colcombet, Denis Kuperberg, Sylvain Lombardy

Regular Temporal Cost Functions

Thomas Colcombet[1], Denis Kuperberg[1], Sylvain Lombardy[2]

[1] Liafa/CNRS/Université Paris 7, Denis Diderot, France
[2] Ligm - Université Paris-Est Marne-la-Vallée, France

Abstract. Regular cost functions have been introduced recently as an extension to the notion of regular languages with counting capabilities, which retains strong closure, equivalence, and decidability properties. The specificity of cost functions is that exact values are not considered, but only estimated.
In this paper, we study the strict subclass of *regular temporal cost functions*. In such cost functions, it is only allowed to count the number of occurrences of consecutive events. For this reason, this model intends to measure the length of intervals, i.e., a discrete notion of time. We provide various equivalent representations for functions in this class, using automata, and 'clock based' reduction to regular languages. We show that the conversions are much simpler to obtain, and much more efficient than in the general case of regular cost functions.
Our second aim in this paper is to use temporal cost function as a test-case for exploring the algebraic nature of regular cost functions. Following the seminal ideas of Schützenberger, this results in a decidable algebraic characterization of regular temporal cost functions inside the class of regular cost functions.

1 Introduction

Since the seminal works of Kleene [Kle56] and Rabin and Scott [RS59], the theory of regular languages is one of the cornerstones in computer science. Regular languages have many good properties, of closure, of equivalent characterizations, and of decidability, which makes them central in many situations.

Recently, the notion of regular cost function for words has been presented as a candidate for being a quantitative extension to the notion of regular languages [Col09c], while retaining most of the fundamental properties of the original theory such as the closure properties, the various equivalent characterizations, and the decidability. A cost function is an equivalence class of the functions from the domain (words in our case) to $\mathbb{N}_\infty$, modulo an equivalence relation $\approx$ which allows some distortion, but preserves the boundedness property over each subset of the domain. The model is an extension to the notion of languages in the following sense: one can identify a language with the function mapping each word inside the language to 0, and each word outside the language to ∞. It is a strict extension since regular cost functions have counting capabilities, e.g., counting the number of occurrences of letters, measuring the length of intervals, etc...

Related works and motivating examples

Regular cost functions are the continuation of a sequence of works that have intended to solve difficult questions in language theory. The prominent example is the star-height problem: given a regular language L and an integer k, decide whether L can be expressed using a regular expression using at most k-nesting of Kleene stars. It was raised by Eggan in 1963 [Egg63], but solved only 25 years later by Hashigughi using

a very intricate proof [Has88]. An improved and self-contained proof has been more recently proposed by Kirsten [Kir05]. The two proofs work along the same lines: show that the original problem can be reduced to the existence of a bound over some function from words to integers. This function can be represented using an automaton that have counting features (a distance automaton for Hashiguchi, and a nested distance desert automaton for Kirsten). The proof is concluded by showing that such boundedness problems are decidable.

Other decision problems can also be reduced to boundedness questions over words: in language theory the *finite power property* [Sim78,Has79] and the *finite substitution problem* [Bal04,Kir04], and in model theory the *boundedness problem* of monadic formulas over words [BOW09]. Distance automata are also used in the context of databases and image compression. Automata similar to the ones of Kirsten have also been introduced independently in the context of verification [AKY08].

Finally, using also ideas inspired from [BC06], the theory of those automata over words has been unified in [Col09c], in which cost functions are introduced, and suitable models of automata, algebra, and logic for defining them are presented and shown equivalent. Corresponding decidability results are provided. The resulting theory is a neat extension of the standard theory of regular languages to a quantitative setting. All the limitedness problems from the literature appear as special instances of those results, as well as all the central results known for regular languages.

Contributions

We introduce the subclass of regular temporal cost functions. Regular temporal cost functions are regular cost functions in which one can only count consecutive events: for instance, over the alphabet $\{a, b\}$, the maximal length of a sequence of consecutive letter a's is temporal, while the number of occurrences of letter a is not. This corresponds to the model of *desert automata* introduced by Kirsten [Kir04]. We believe that the notion of regular temporal cost function is of interest in the context of modelization of time.

We show that regular temporal cost functions admit various equivalent presentations. The first such representation is obtained as a syntactic restriction of B-automata and S-automata (the automata used for describing regular cost functions [Col09c]). Second, we provide an equivalent *clock-based* presentation, in which the regular temporal cost functions is represented as a regular language over words labeled with the ticks of a clock as an extra information. We show all the closure results for regular temporal cost functions (e.g., min, max, etc...) using this presentation. As opposed to the general theory of regular cost functions, all those results are obtained by a translation to the theory of regular languages. This results in constructions of better complexity, both in terms of number of states of automata, and in terms of technicality of the constructions themselves. Last but not least, while in the general theory of regular cost functions the error committed during the construction is bounded by a polynomial, it is linear for regular temporal cost functions.

Our second contribution is an algebraic characterization of this class. It is known from [Col09c] that regular cost functions are the one recognizable by stabilization monoids. This model of monoids extends the standard approach for languages. One of our objectives in studying regular temporal cost function was to validate the interest of this algebraic approach, and show that results similar to the famous Schützenberger theorem on star-free languages [Sch65] were possible. We believe that we succeeded in this direction, since we are able to algebraically characterize the class of regular temporal cost functions, and furthermore that this characterization is effective.

Organisation of the paper

After some notations, we present cost functions and cost automata in Section 2, and introduce the subclass of regular temporal cost functions. In Section 3 we propose a clock-based presentation to temporal cost functions, and advocate some of its advantages. In Section 4 we present the algebraic formalism and sketch our algebraic characterization for regular temporal cost functions. We finally conclude.

Notations

We will note $\mathbb{N}$ the set of non-negative integers and $\mathbb{N}_\infty$ the set $\mathbb{N} \cup \{\infty\}$, ordered by $0 < 1 < \cdots < \infty$. If E is a set, $E^{\mathbb{N}}$ is the set of infinite sequences of elements of E (we will not use here the notion of infinite words). Such sequences will be denoted by bold letters ($\boldsymbol{a}$, $\boldsymbol{b}$,...). We will work with a fixed finite alphabet $\mathbb{A}$. The set of words over $\mathbb{A}$ is $\mathbb{A}^*$. The empty word ϵ, and $\mathbb{A}^+ = \mathbb{A}^* \setminus \{\epsilon\}$. The concatenation of words u and v is uv.The length of u is $|u|$. The number of occurrences of letter a in u is $|u|_a$. We will note $\inf E$ and $\sup E$ the lower and upper bounds of a set $E \subseteq \mathbb{N}_\infty$, in particular $\inf \emptyset = \infty$ and $\sup \emptyset = 0$.

2 Regular cost functions

The theory of regular cost functions has been proposed in [Col09c]. In this section, we review some of the definitions useful for the present paper.

2.1 Basics on cost functions

The principle of cost functions is to consider functions modulo an equivalence relation $\approx$ allowing some distortions of the values. This distortion is controlled using a parameter $(\alpha, \alpha', \alpha_1 \dots)$ which is a mapping from $\mathbb{N}$ to $\mathbb{N}$ such that $\alpha(n) \geq n$ for all n, called the *correction function*. For $x, y \in \mathbb{N}_\infty$, $x \preccurlyeq_\alpha y$ means that either x and y are in $\mathbb{N}$ and $x \leq \alpha(y)$, or $y = \infty$. It is equivalent to write that $x \leq \alpha(y)$ in which we implicitly extend α to $\mathbb{N}_\infty$ by $\alpha(\infty) = \infty$. For all sets E, $\preccurlyeq_\alpha$ is naturally extended to mappings from E to $\mathbb{N}_\infty$ by $f \preccurlyeq_\alpha g$ if $f(x) \preccurlyeq_\alpha g(x)$ for all $x \in E$, or equivalently if $f \leq \alpha \circ g$ (using the same explicit extension of α). The intuition here is to consider that g dominates f up to a 'stretching factor' α. We note $f \approx_\alpha g$ if $f \preccurlyeq_\alpha g$ and $g \preccurlyeq_\alpha f$. Finally, we note $f \preccurlyeq g$ (resp. $f \approx g$) 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 $\approx$ among the set of functions from E to $\mathbb{N}_\infty$.

The relation $\preccurlyeq$ has other characterizations:

Proposition 1 *For all functions $f, g : E \to \mathbb{N}_\infty$, the following items are equivalent:*

- $f \preccurlyeq g$,
- *For all $X \subseteq E$, g bounded over X implies f bounded over X.*

The last characterization shows that $\approx$ preserves the existence of bounds.

To each subset $X \subseteq E$, one associates its characteristic mapping χ_X from E to $\mathbb{N}_\infty$ which to x associates 0 if $x \in X$, and ∞ otherwise. It is easy to see that $X \subseteq Y$ iff $\chi_X \succcurlyeq \chi_Y$. In this way, the notion of cost functions can be seen as an extension to the notion of language.

2.2 Cost-Automata

In this section, we will describe how some functions from $\mathbb{A}^*$ to $\mathbb{N}_\infty$ can be accepted by certain forms of automata using counters of value ranging in $\mathbb{N}$. We name such cost functions 'regular'.

A *cost automaton* is a tuple $\langle Q, \mathbb{A}, In, Fin, \Gamma, \Delta\rangle$ where Q is a finite set of *states*, $\mathbb{A}$ is a finite *alphabet*, In and Fin are the set of *initial* and *final* states respectively, Γ is a finite set of *counters*, and $\Delta \subseteq Q \times A \times (\{\mathrm{i},\mathrm{r},\mathrm{c}\}^*)^\Gamma \times Q$ is the set of *transitions*.

The value of each counter ranges over $\mathbb{N}$, and evolves according to *atomic actions* in $\{\mathrm{i},\mathrm{r},\mathrm{c}\}$: i *increments* the value by 1, r *resets* the value to 0, and c *checks* the value (but does not change it). Each action in $(\{\mathrm{i},\mathrm{r},\mathrm{c}\}^*)^\Gamma$ tells for each counter what sequence of atomic actions has to be performed. Hence, given a sequence of actions u, one can execute it as follows: at the begining, all counters share the value 0, and we read the word u letter by letter from left to right. For each letter, one applies the corresponding sequence of atomic actions on each counter. One sets the set $C(u)\subseteq \mathbb{N}$ to contain all values that are taken by a counter when checked (this set collects all the checked values indistinctly: there is no distinction concerning the counter the value originates from, or the moment of the check).

A *run* σ of a cost automaton over a word $a_1 \dots a_n$ is a sequence in Δ^* of the form $(q_0, a_1, t_1, q_1)(q_1, a_2, t_2, q_2)\dots(q_{n-1}, a_n, \sigma_n, q_n)$ such that q_0 is initial, q_n is final (the run ε is also valid iff there exists q_0, both initial and final). One sets $C(\sigma)= C(t_1 \dots t_n)$, i.e., to collect the set of values checked when executing the run over the counters.

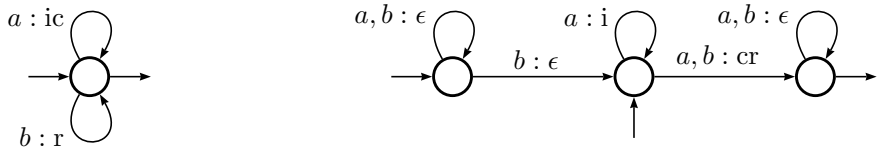
At this point, cost automata are instantiated in two versions, namely *B-automata* and *S-automata* that differs by their dual semantics, $[\![\cdot]\!]_B$ and $[\![\cdot]\!]_S$ respectively. These semantics are defined for all $u \in \mathbb{A}^*$ by:

$$[\![\mathcal{A}]\!]_B(u) = \inf\{\sup C(\sigma) \ : \ \sigma \text{ run over } u\} \ ,$$
$$\text{and} \quad [\![\mathcal{A}]\!]_S(u) = \sup\{\inf C(\sigma) \ : \ \sigma \text{ run over } u\} \ .$$

(Recall that $\sup\emptyset = 0$ and $\inf\emptyset = \infty$) One says that a B-automaton (resp. an S-automaton) *accepts* $[\![\mathcal{A}]\!]_B$ (resp. $[\![\mathcal{A}]\!]_S$).

Example 1. If $\mathcal{A}$ is a standard non-deterministic automaton accepting $L \subseteq \mathbb{A}^*$, it can be seen as a cost automaton without any counter. Seen as a B-automaton, we have $[\![\mathcal{A}]\!]_B(u) = \chi_L$, and seen as an S-automaton, $[\![\mathcal{A}]\!]_S(u) = \chi_{\mathbb{A}^*\setminus L}$.

Example 2. We describe the two one counter cost automata $\mathcal{A}$ and $\mathcal{A}'$ by drawings:



Circles represent states, and a transition (p, a, t, q) is denoted by an edge from p to q labeled $a : t$ (the notation $a, b : t$ abbreviates multiple transitions). Initial states are identified by unlabeled ingoing arrows, while final states use unlabeled outgoing arrows. One checks that $[\![\mathcal{A}]\!]_B \approx [\![\mathcal{A}']\!]_S \approx f_a$ where $f_a(u) = \max\{n \in \mathbb{N} \ / \ u = va^n w\}$.

A B-automaton is *simple* if it uses actions in $\{\epsilon, \mathrm{ic}, \mathrm{r}\}^\Gamma$. A S-automaton is *simple* if it uses actions in $\{\epsilon, \mathrm{i}, \mathrm{cr}\}^\Gamma$. The following theorem is central in the theory:

Theorem 2 (duality [Col09c,Col09a]). *It is equivalent for a function, up to $\approx$, to be accepted by a [simple] B-automaton or to be accepted by a [simple] S-automaton.*

Such cost functions are called *regular*. This comes with a decision procedure:

Theorem 3 ([Col09c]). *The relations $\preccurlyeq$ and $\approx$ are decidable for regular cost functions.*

2.3 Regular temporal cost functions

The subject of the paper is to study the regular temporal cost functions, a subclass of regular cost functions. We give here a first definition of this class.

A B-automaton (resp. S-automaton) is *temporal* if it uses only actions in $\{\mathrm{ic}, \mathrm{r}\}^\Gamma$ (resp. $\{\mathrm{i}, \mathrm{cr}\}^\Gamma$). Hence temporal automata are simple automata in which it is disallowed in an action to leave counters unchanged. Intuitively, such automata can only measure consecutive events. We define $temp_B$ (resp. $temp_S$) to map sequences in $\{\mathrm{ic}, \mathrm{r}\}^*$ to $\mathbb{N}$ (resp. $\{\mathrm{i}, \mathrm{cr}\}^*$ to $\mathbb{N}_\infty$) which to u associates $(\sup C(u))$ (resp. $(\inf C(u))$). Those functions are extended to sets of counters and runs as in the general case of cost automata. We say that a cost function is B-temporal (resp. S-temporal) if it is accepted by a temporal B-automaton (resp. a temporal S-automaton). We will see below that these two notions coincide, up to $\approx$ (Theorem 7).

Example 3. Over the alphabet $\{a, b\}$, the cost function f_a from Example 2 is B-temporal (as witnessed by the example automaton).

However, the function $u \mapsto |u|_a$ is not B-temporal, even modulo $\approx$. Indeed, for the sake of contradiction, assume that there exists a temporal B-automaton $\mathcal{A} = \langle Q, \mathbb{A}, In, Fin, \Gamma, \Delta \rangle$ accepting g, with $g \approx_\alpha |\cdot|_a$ for some α. Let $K = |Q| + 1$ and $N = \alpha(K) + 1$. Let σ be the run of $\mathcal{A}$ over $u = (b^N a)^K$ which minimizes $\sup C(\sigma)$ (it has to exist since $g(u) \approx_\alpha |u|_a < \infty$). Since $K > |Q| + 1$, one can decompose u as xvy such that $|v|_a \geq 1$, $|v| \geq N$, and the run σ assumes same state p after reading both x and xv. Let $\sigma_x \sigma_v \sigma_y$ be the corresponding decomposition of the run σ. Assume first that there exists a counter which is never reset during σ_v, then we get $g(u) \geq N > \alpha(|u|_a)$. This contradicts $g \approx_\alpha |\cdot|_a$. Hence all counters have to be reset somewhere in σ_v. Consider the word $u_m = xv^m y$. One easily checks that $|u_m|_a \geq m$ since $|u|_a \geq_1$. However, the run $\sigma_x \sigma_v^m \sigma_y$ witnesses that $g(u_m) \leq \max(g(u), |u|)$. Hence $|\cdot|_a$ is unbounded over the u_m's, while g is bounded over the same set. This is a contradiction according to Proposition 1.

3 Clock-form of temporal cost functions

In this section, we give another characterization to B-temporal and S-temporal regular cost functions. This presentation makes use of clocks (the notion of clock should not be confused with the notion of clock used for timed automata).

A *clock* c is a word over the alphabet $\{\_, \downarrow\}$. It should be seen as describing the *ticks* of a clock: the letter is $\_$ if there is no tick at this moment, and it is $\downarrow$ when there is a tick. A clock naturally determines a factorization of time into intervals (we say segments). Here, one factorizes c as:

$$c = (\_^{n_1 - 1} \downarrow)(\_^{n_2 - 1} \downarrow) \dots (\_^{n_k - 1} \downarrow)\_^{m-1} .$$

One sets $\max{-}\mathrm{seg}(c)$ to be $\max\{n_1, \dots, n_k, m\} \in \mathbb{N}$, and $\min{-}\mathrm{seg}$ to be $\inf\{n_1, \dots, n_k\} \in \mathbb{N}_\infty$ (remark the asymmetry). A clock c *has period* $P \in \mathbb{N}$ if $n_1 = n_2 = \dots = n_k = P$, and $m \leq P$. This is equivalent to stating[3] $\max{-}\mathrm{seg}(c) \leq P \leq \min{-}\mathrm{seg}(c)$. Remark

[3] Remark that as soon as $k \geq 1$, the inequalities become–as one may expect–equalities.

that given n and P, there exists one and only one clock of length n and period P. You can remark that $\max{-}\mathrm{seg}(c) = temp_B(h_B(c)) + 1$ in which h_B maps $\_$ to ic and $\downarrow$ to r. Similarly, $\min{-}\mathrm{seg}(c) = temp_S(h_S(c)) + 1$ in which h_B maps $\_$ to i and $\downarrow$ to cr.

A *clock on* $u \in \mathbb{A}^*$ is a clock c of length $|u|$, In this case, one denotes by $\langle u, c\rangle$ the word over $\mathbb{A} \times \{\_, \downarrow\}$ obtained by pairing the letters in u and in c of same index. For L a language in $(\mathbb{A} \times \{\_, \downarrow\})^*$, we define the following functions from $\mathbb{A}^*$ to $\mathbb{N}_\infty$:

$$\langle\!\langle L\rangle\!\rangle_B : u \mapsto \inf\{\max{-}\mathrm{seg}(c) \ : \ c \text{ clock on } u,\ \langle u, c\rangle \in L\}$$
$$\langle\!\langle L\rangle\!\rangle_S : u \mapsto \sup\{\min{-}\mathrm{seg}(c) \ : \ c \text{ clock on } u,\ \langle u, c\rangle \notin L\} + 1$$

Lemma 1. *For all languages* $L \subseteq (\mathbb{A} \times \{\_, \downarrow\})^*$, $\langle\!\langle L\rangle\!\rangle_B \leq \langle\!\langle L\rangle\!\rangle_S$.

Proof. Fix u. Consider the minimal P such that the clock c over u of period P is such that $\langle u, c\rangle \in L$ (if there is no such period, $\langle u, \_^{|u|}\rangle \notin L$, and $\langle\!\langle L\rangle\!\rangle_S(u) = \omega$). We clearly have $\langle\!\langle L\rangle\!\rangle_B(u) \leq P$. On the other hand, $\langle u, c'\rangle \notin L$, where c' is the clock over u of period $P - 1$. Hence $\langle\!\langle L\rangle\!\rangle_B(u) \leq P \leq \langle\!\langle L\rangle\!\rangle_S(u)$. □

The notations $\langle\!\langle \cdot\rangle\!\rangle_B$ and $\langle\!\langle \cdot\rangle\!\rangle_S$ are easily convertible into temporal cost automata as shown by Fact 4.

Fact 4 *If L is regular and L (resp. $\complement L$) is accepted by a non-deterministic automaton with n states, then $\langle\!\langle L\rangle\!\rangle_B - 1$ (resp. $\langle\!\langle L\rangle\!\rangle_S - 1$) is accepted by a temporal B-automaton (resp. a temporal S-automaton) with n states and one counter.*

Proof. We have seen that $\max{-}\mathrm{seg} = (temp_B \circ h_B) + 1$. Hence, if we replace in the automaton for L each transition of the form $(p, (a, c), q)$ by a transition $(p, a, h_B(c), q)$, we immediately get the desired temporal B-automaton. The construction for temporal S-automata is identical, starting from the complement automaton, and using h_S. □

The important definition is the following:

Definition 5 *An* α-clock-language *(or simply a* clock-language *if there exists such an α) is a language $L \subseteq (\mathbb{A} \times \{\_, \downarrow\})^*$ such that $\langle\!\langle L\rangle\!\rangle_B \approx_\alpha \langle\!\langle L\rangle\!\rangle_S$. A function f* has an α-clock-form *if there exists an α-clock-language L such that $\langle\!\langle L\rangle\!\rangle_S \leq f \preccurlyeq_\alpha \langle\!\langle L\rangle\!\rangle_B$. A cost function* has a clock-form *if it contains a function that has an α-clock-form for some α. We note $\mathcal{CF}$ the set of cost functions that have a clock-form.*

One can remark that it is sufficient to prove $\langle\!\langle L\rangle\!\rangle_S \preccurlyeq_\alpha \langle\!\langle L\rangle\!\rangle_B$ for proving that L is an α-clock-language: Lemma 1 provides indeed the other direction.

Example 4. For $L \subseteq \mathbb{A}^*$, $K = L \times \{\_, \downarrow\}^*$ is a clock-language, which witnesses that χ_L has an *identity*-clock-form.

Example 5. Consider again the function f_a of Example 2, computing the maximal number of consecutive a's. The language $M = ((a, \_) + (b, \downarrow))^*$ verifies $\langle\!\langle M\rangle\!\rangle_B \approx f_a$, but it is not a clock-language: for instance the word ba^m is such that $f_a(ba^m) = m$, meanwhile, $\langle\!\langle M\rangle\!\rangle_S(ba^m) = 0$. This contradicts $\langle\!\langle M\rangle\!\rangle_S \approx f_a$ according to Proposition 1. This comes from the fact that the clock witnessing $\langle\!\langle M\rangle\!\rangle_B \approx f_a$ is chosen given the word (the one ticking exactly over b-letters). This is in contradiction with the important intuition behind being in clock-form which is that the clock can be chosen independently from the word.

However, it is possible to construct a rational clock-language L for f_a. It checks that each segment of consecutive a's contains at most one tick of the clock, i.e.:

$$L = K[((b, \_) + (b, \downarrow))K]^* \qquad \text{in which} \quad K = (a, \_)^* + (a, \_)^*(a, \downarrow)(a, \_)^* \ .$$

Let u be a word, and c be a clock such that $\min{-}\mathrm{seg}(c) = n$ and $\langle u, c\rangle \notin L$. Since $\langle u, c\rangle \notin L$, there exists a factor of u of the form a^k in which there are two ticks of the clock. Hence, $k \geq n+1$. From which we obtain $\langle\!\langle L\rangle\!\rangle_S \leq f_a$. Conversely, let u be a word, and c be a clock such that $\max{-}\mathrm{seg}(c) = n$ and $\langle u, c\rangle \in L$. Let $k = f_a(u)$. This means that there is a factor of the form a^k in u. Since $\langle u, c\rangle \in L$, there is at most one tick of the clock in this factor a^k. Hence, $k \leq 2n-1$. We obtain that $f_a < 2\langle\!\langle L\rangle\!\rangle_B$. Hence, L is an α-clock-language for f_a, with $\alpha : n \mapsto 2n$.

Let us turn ourselves to closure properties for languages in clock-form. Consider a mapping f from $\mathbb{A}^*$ to $\mathbb{N}_\infty$ and a mapping h from $\mathbb{A}$ to $\mathbb{B}$ ($\mathbb{B}$ being another alphabet) that we extend into a monoid morphism from $\mathbb{A}^*$ to $\mathbb{B}^*$, the inf-projection of f (resp. sup-projection) with respect to h is the mapping $f_{\inf,h}$ (resp. $f_{\sup,h}$) from $\mathbb{B}^*$ to $\mathbb{N}_\infty$ defined for all $v \in \mathbb{B}^*$ by:

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

The following theorem shows closure properties of cost functions in clock-form that are obtained by translation to a direct counterpart in language theory:

Theorem 6 *Given L, M α-clock-languages over $\mathbb{A}$, h from $\mathbb{A}$ to $\mathbb{B}$ and g from $\mathbb{B}$ to $\mathbb{A}$, we have:*

- *$L \cup M$ is an α-clock-language and $\langle\!\langle L \cup M\rangle\!\rangle_B = \min(\langle\!\langle L\rangle\!\rangle_B, \langle\!\langle M\rangle\!\rangle_B)$*
- *$L \cap M$ is an α-clock-language and $\langle\!\langle L \cap M\rangle\!\rangle_S = \max(\langle\!\langle L\rangle\!\rangle_S, \langle\!\langle M\rangle\!\rangle_S)$*
- *$L_{\circ g} = \{\langle u, c\rangle \ : \ \langle g(u), c\rangle \in L\}$ is an α-clock-language and $\langle\!\langle L_{\circ g}\rangle\!\rangle_B = \langle\!\langle L\rangle\!\rangle_B \circ g$*
- *$L_{\inf,h} = \{\langle h(u), c\rangle \ : \ \langle u, c\rangle \in L\}$ is an α-clock-language and $\langle\!\langle L_{\inf,h}\rangle\!\rangle_B = (\langle\!\langle L\rangle\!\rangle_B)_{\inf,h}$*
- *$L_{\sup,h} = \complement\{\langle h(u), c\rangle \ : \ \langle u, c\rangle \notin L\}$ is an α-clock-language and $\langle\!\langle L_{\sup,h}\rangle\!\rangle_S = (\langle\!\langle L\rangle\!\rangle_S)_{\sup,h}$*

Proof. The five items follow all the same proof principle. Let us treat the case of inf-projection. The equality is proved by the following sequence of equalities:

$$\begin{aligned}(\langle\!\langle L_{\inf,h}\rangle\!\rangle_B)(v) &= \inf\{\max{-}\mathrm{seg}(c) \ : \ \langle v, c\rangle \in L_{\inf,h}\} \\ &= \inf\{\max{-}\mathrm{seg}(c) \ : \ \langle u, c\rangle \in L, \ h(u) = v\} \\ &= \inf\{\inf\{\max{-}\mathrm{seg}(c) \ : \ \langle u, c\rangle \in L\} \ : \ h(u) = v\} \quad = (\langle\!\langle L\rangle\!\rangle_B)_{\inf,h}(v)\end{aligned}$$

Assume L is an α-clock-language, it remains to be shown that $L_{\inf,h}$ is also an α-clock-language. Let v be a word and c be the clock witnessing $\langle\!\langle L\rangle\!\rangle_B(v) = n$, i.e., such that $\langle v, c\rangle \in L_{\inf,h}$ and $\max{-}\mathrm{seg}(c) = n$. Let c' be a clock over v such that $\min{-}\mathrm{seg}(c') > \alpha(n)$, we have to show $\langle v, c'\rangle \in L_{\inf,h}$. Since $\langle v, c\rangle \in L_{\inf,h}$, there exists u such that $v = h(u)$ and $\langle u, c\rangle \in L$. Hence, since L is an α-clock-language, $\langle u, c'\rangle \in L$. It follows that $\langle v, c\rangle \in L_{\inf,h}$. □

Lemma 2. *$temp_B$ and $temp_S$ have $\times 2$-clock-forms with $\times 2(n) = 2n$.*

Proof. The proof for $temp_B$ is the same as in Example 5, in which one replaces the letter a by ic and the letter b by r. The $temp_S$ side is similar. *(See Appendix A.1)* □

Theorem 7 *If f is a regular cost function, the following assertions are equivalent :*

1. *f has a clock-form,*
2. *f is B-temporal,*
3. *f is computed by a temporal B-automaton with only one counter,*
4. *f is S-temporal,*
5. *f is computed by a temporal S-automaton with only one counter.*

Proof. (1)⇒(3) follows from Fact 4. (3)⇒(2) is trivial.
(2)⇒(1): Consider a temporal B-automaton $\mathcal{A} = \langle Q, \mathbb{A}, In, Fin, \Gamma, \Delta \rangle$ using counters $\Gamma = \{\gamma_1, \ldots, \gamma_k\}$. A run of $\mathcal{A}$ is a word on the alphabet $\mathbb{B} = Q \times \mathbb{A} \times \{\mathrm{ic}, \mathrm{r}\}^\Gamma \times Q$. It follows from the definition of $[\![\cdot]\!]_B$ that for all $u \in \mathbb{A}^*$:

$$[\![\mathcal{A}]\!]_B(u) = \inf_{\sigma \in \mathbb{B}^*} \{\max(\chi_R(\sigma), \mathit{temp}_B \circ \pi_1(\sigma), \cdots, \mathit{temp}_B \circ \pi_k(\sigma)) \ : \ \pi_{\mathbb{A}}(\sigma) = u\}$$

in which $R \subseteq \Delta^*$ is the (regular) set of valid runs; for all $i \in [\![1, k]\!]$, π_i projects each transition (p, a, t, q) to the its γ_i^{th} component of t (and is extended to words). Finally $\pi_{\mathbb{A}}$ projects each transition (p, a, t, q) to a (and is also extended to words). By Example 4, $\chi_R \in \mathcal{CF}$. By Lemma 2, $\mathit{temp}_B \in \mathcal{CF}$, and by Theorem 6, $\mathcal{CF}$ is stable under composition, max and inf-projection. Hence $[\![\mathcal{A}]\!]_B \in \mathcal{CF}$.

The equivalences (2)⇔(4)⇔(5) are proved in a similar way. □

Actually, Theorem 6 and Lemma 2 allow to state that if a function f is given by one of the five descriptions of Theorem 7, then for any other among these descriptions, there exists a function g which is $\approx_{\times 2}$-equivalent to f.

In the following, we will simply say that f is a *temporal* cost function instead of B-temporal or S-temporal.

Conclusion on clock-forms, and perspectives

Independently from the second part of the paper, we believe that some extra comments on the clock-form approach are interesting.

First of all, let us stress the remarkable property of the clock-form presentation of temporal cost functions: those can be seen either as defining a function as an infimum ($\langle\!\langle \cdot \rangle\!\rangle_B$) or as a supremum ($\langle\!\langle \cdot \rangle\!\rangle_S$). Hence, regular cost function in clock-forms can be seen either as B-automata or as S-automata. This presentation is in some sense 'self-dual'. Nothing similar is known for general regular cost functions.

Another difference with the general case is that all constructions are in fact reduction to constructions for languages: This is particularly obvious in the statement of Theorem 6. Furthermore, since everything is done at the level of languages, we do not require any specific presentation for the languages. Those can be described e.g. by any form of automata or algebra. For this reason, any specific optimised constructions for regular language should be reusable for regular temporal cost functions. However, since two different languages L, L' can be such that $\langle\!\langle L \rangle\!\rangle_B \approx \langle\!\langle L' \rangle\!\rangle_B$ (even $\langle\!\langle L \rangle\!\rangle_B = \langle\!\langle L' \rangle\!\rangle_B$), one must keep aware that optimal operations performed at the level of languages–such as minimization–will not be optimal anymore when used for describing temporal cost functions. It is a perspective of research to develop dedicated algorithmic for regular temporal cost functions.

A third difference is that the error committed, which is measured by the stretching factor α, is linear. This is much better than the general case of cost functions, in which, e.g., the equivalence between B-automata and S-automata requires a polynomial stretching factor. However, we do not take yet full advantage of this in the present paper since we do not try to use this precision, e.g., in new decision procedures. There are also here researches to be conducted.

In fact, the argument underlying temporal cost functions in clock-forms is interesting per se: it consists in approximating some quantitative notion, here the notion of length of intervals, using some extra unary information, here the ticks of the clock. Since unary information can be handled by automata, the approximation of the quantitative notion becomes also available to the automaton. This is a very robust principle

that clearly can be reused in several other ways. For instance, it would be no different to consider ticks of a clock over an infinite word (in fact the fact that words are finite is even entailing problems). It would be no different on trees (seen as a branching presentation of time), be they finite or infinite. Keeping on the same track, a clock is even not required to count the time, it could count some events already written on the input, such as the number of a's, etc. These examples show the versatility of the approach.

4 Algebraic approach

We first recall definitions of classic semigroups and stabilization semigroups for the general case of regular cost functions. We use them in a decidable algebraic characterization of temporal cost functions.

4.1 Standard semigroups

Definition An *ordered semigroup* $\mathbf{S} = \langle S, \cdot, \leq \rangle$ is a set S endowed with an associative product $\cdot : S \times S \to S$ and a partial order $\leq$ over S compatible with $\cdot$ (i.e. if $a \leq a'$ and $b \leq b'$, then $a \cdot b \leq a' \cdot b'$).

An *idempotent* element of $\mathbf{S}$ is an element $e \in S$ such that $e \cdot e = e$. We note $E(\mathbf{S})$ the set of idempotent elements of $\mathbf{S}$.

Recognizing languages In the standard theory, the recognition of a language by a finite semigroup is made through a morphism from words into the semigroup which can be decomposed into two steps: first, a length-preserving morphism $h : \mathbb{A}^+ \to S^+$, where S^+ is the set of words whose letters are in S, and second the function π: $S^+ \to S$ which maps every word on S onto the product of its letters. The language L recognized by the triple (S, h, P), where P is a subset of S, is $L = h^{-1}(\pi^{-1}(P))$, i.e. $u \in L$ iff $\pi(h(u)) \in P$.

It is standard that languages recognized by finite semigroups are exactly the regular languages. It is also by now well known that families of regular languages can be characterized by restrictions on the semigroups which recognize them. This is for instance the case in Eilenberg's variety theorem or in Schützenberger's theorem characterizing star-free languages as the one recognized by aperiodic semigroups [Sch65].

4.2 Stabilization semigroup

The notion of stabilization monoid has been introduced in [Col09c] as a quantitative extension of standard monoids, for the recognition of cost functions. Stabilization semigroup is a more convenient object in the present paper, since the empty word plays a special role (it has length 0). The relationship between stabilization monoids and stabilization semigroups is made explicit in [Col09b]. A side effect is that it is more easy to speak about regular cost functions over non-empty words. We do it from now for simplicity.

Definition 8 *A* stabilization semigroup $\langle S, \cdot, \leq, \sharp \rangle$ *is an ordered semigroup* $\langle S, \cdot, \leq \rangle$ *together with an operator* $\sharp$: $E(\mathbf{S}) \to E(\mathbf{S})$ *(called* stabilization*) such that:*

- *for all* $a, b \in S$ *with* $a \cdot b \in E(\mathbf{S})$ *and* $b \cdot a \in E(\mathbf{S})$, $(a \cdot b)^\sharp = a \cdot (b \cdot a)^\sharp \cdot b$;
- *for all* $e \in E(\mathbf{S})$, $(e^\sharp)^\sharp = e^\sharp \leq e$;
- *for all* $e \leq f$ *in* $E(\mathbf{S})$, $e^\sharp \leq f^\sharp$;

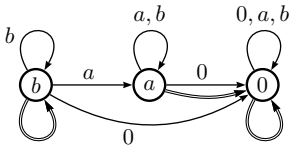
In this paper, we only consider finite stabilization semigroups. The intuition of the $\sharp$ operator is that $e^\sharp$ represents the value that gets e 'when repeated many times'. This may be different from e if one is interested in counting the number of occurrences of e.

4.3 Recognizing cost functions

The first step for recognizing cost function is to provide a 'quantitative semantic' to the stabilization semigroup $\mathbf{S} = \langle S, \cdot, \leq, \sharp \rangle$. This is done by a mapping ρ named a *compatible mapping*, which maps every word of S^+ to an infinite non-decreasing sequence of $S^{\mathbb{N}}$ (the original definition does not use non-decreasing sequences, but is equivalent, see e.g., [Col09c]). The principle is that the ith position in the sequence $\rho(u)$ tells the value of the word u for a threshold i separating what is considered as few and as lot. This is better seen on an example.

Example 6. Consider the following stabilization semigroup:

	b	a	0	$\sharp$
b	b	a	0	b
a	a	a	0	0
0	0	0	0	0



It is given both by its table of product augmented by a column for the stabilization and by its Cayley graph. In the Cayley graph there is an edge labelled by y linking element x to element $x \cdot y$. There is furthermore a double arrow going from each idempotent to its stabilized version.

The intention is to count the number of a's. Words with no a's correspond to element b. Words with at least one, but few a's correspond to element a. Finally, words that contain a lot of a's should have value 0: for instance, $a^\sharp = 0$ witnesses that iterating a lot of time a word with at least one a yields a word with a lot of a's.

A possible compatible mapping ρ for this stabilization semigroup attaches to each word over $\{b, a, 0\}^+$ an infinite sequence of values in $\{b, a, 0\}$ as follows: every word in b^+ is mapped to the constant sequence b; every word containing 0 is mapped to the constant sequence 0; every word $u \in b^*(ab^*)^+$ is mapped to 0 for indices up to $|u|_a - 1$ and a for indice $|u|_a$ and beyond. The idea is that for a threshold $i < |u|_a$, the word is considered as having a lot of a's in front of i (hence value 0), while it has few a's in front of i for $i \geq |u|_a$ (hence the value a). One can see that this sequence 'codes' the number of a's in the position in which it switches from value 0 to value a.

A formal definition of a compatible mapping requires to state the properties it has to satisfy, and which relate it to the stabilisation monoid. This would require much more material, and we have to stay at this informal level in this short paper *(See Appendix A.6)*. The important result here is that given a finite stabilization monoid, there exists a mapping compatible with it, and furthermore that it is unique up to an equivalence $\sim$ (which essentially corresponds to $\approx$) [Col09c,Col09b]. Hence, in the above example, the compatible mapping described is the unique possible (up to $\sim$).

Now that we know what the semantics of stabilization semigroups look like, one uses it for recognizing cost functions. Instead of computing the product of elements and checking whether it belongs to a subset P of the semigroup, the quantitative recognition consists in considering the infinite sequence obtained by the compatible mapping and observing the first moment it leaves a fixed ideal I of the semigroup (an ideal is a downward $\leq$-closed subset). Formally, the cost function f over $\mathbb{A}^+$ recognized by $(\mathbf{S}, h, I)$ is $f : u \mapsto \inf\{n \in \omega, \rho(h(u))(n) \notin I\}$, where $h : \mathbb{A}^+ \to S^+$ is a length-preserving morphism, and ρ is a mapping compatible with $\mathbf{S}$.

Typically, on the above example, the ideal is $\{0\}$, and h maps each letter in $\{a, b\}$ to the element of same name. For all words $u \in \{a + b\}^+$, the value computed is exactly $|u|_a$.

Theorem 9 *[Col09c] A cost function is regular iff it is recognized by a stabilization semigroup.*

Like for regular languages, this algebraic presentation can be minimized.

Theorem 10 *If f is a regular cost function, there exists effectively a (quotient-wise) minimal stabilization semigroup recognizing f.*

This minimal stabilization semigroup can be obtained from $(\mathbf{S}, h, I)$ by a Moore algorithm computing the coarsest congruence, compatible with the semigroup and stabilization operations, which separates elements of I from the other elements. This procedure is polynomial in the size of $\mathbf{S}$. *(See Appendix A.7)*

4.4 Temporal stabilization semigroups

Let us now characterize the regular temporal cost functions.

We say that an idempotent e is *stable* if $e^\sharp = e$. Otherwise it is *unstable*. The intuition is that stable idempotents are not counted by the stabilization semigroup (b in the example), while the iteration of unstable idempotents matters (a in the example).

Definition 11 *Let $\mathbf{S} = \langle S, \cdot, \leq, \sharp \rangle$ be a stabilization semigroup.* $\mathbf{S}$ *is* temporal *if for all idempotents s and $e = x \cdot s \cdot y$, if s is stable then e is also stable.*

For instance, the example stabilization semigroup is not temporal since b is stable but $a = a \cdot b \cdot a$ is unstable. This is related to temporal cost functions as follows:

Theorem 12 *Let f be a regular cost function, the following assertions are equivalent:*

- *f is temporal*
- *f is recognized by a temporal stabilization semigroup*
- *the minimal stabilization semigroup recognizing f is temporal*

We will briefly give an idea on how the definition of temporal semigroups is related to the intuition of consecutive events. Indeed, an unstable idempotent must be seen as an event we want to 'measure', whereas we are not interested in the number of occurrences of a stable idempotent. But if we have $e = x \cdot s \cdot y$ with e unstable and s stable, it means that we want to 'count' the number of occurrences of e without counting the number of s within e. In other words, we want to increment a counter when e is seen, but s can be repeated a lot inside a single occurrence of e. To accomplish this, we have no other choice but doing action ϵ on the counter measuring e while reading all the s's, however, this kind of behaviour is disallowed for temporal automata.

The two last assertions are equivalent, since temporality is preserved by quotient of stabilization semigroups. On our example, the stabilization semigroup is already the minimal one recognizing the number of occurrences of a, and hence, this cost function is not temporal. We gave a direct proof for this fact in Example 3.

Corollary 1. *The class of temporal cost functions is decidable.*

The corollary is obvious since the property can be decided on the minimal stabilization semigroup, which can be computed either from a cost automaton or a stabilization semigroup defining the cost function.

5 Conclusion

We defined a subclass of regular cost functions called the temporal class. Our first definition used cost automata. We then characterized regular temporal cost functions as the ones describable by clock-languages. This presentation allows to reuse all standard construction for regular languages taken from classic language theory. We then characterized the class in the algebraic framework of stabilization semigroups, the algebraic notion allowing to describe regular cost functions. This together with the construction of minimal stabilization semigroups gave us a decision procedure for the temporal class, and hopefully for more classes in future works.

The later decidable characterization result calls for continuations. Temporal cost functions correspond to desert automata of Kirsten [Kir04], but other subclasses of automata are present in the literature such as distance automata (which correspond to one-counter no-reset B-automata) or distance desert automata (a special case of two counters B-automata). Is there decidable characterizations for the regular cost functions described by those automata? More generally, what is the nature of the hierarchy of counters?

References

[AKY08] Parosh Aziz Abdulla, Pavel Krcal, and Wang Yi. R-automata. In Franck van Breugel and Marcha Chechik, editors, *Proceedings of CONCUR'08, Toronto, Canada.*, volume 5201 of *Lecture Notes in Computer Science*, pages 67–81. Springer-Verlag, 2008.

[Bal04] Sebastian Bala. Regular language matching and other decidable cases of the satisfiability problem for constraints between regular open terms. In *STACS*, volume 2996 of *Lecture Notes in Computer Science*, pages 596–607. Springer, 2004.

[BC06] Mikolaj Bojańczyk and Thomas Colcombet. Bounds in ω-regularity. In *LICS 06*, pages 285–296, August 2006.

[BOW09] Achim Blumensath, Martin Otto, and Mark Weyer. Boundedness of monadic second-order formulae over finite words. In *36th ICALP*, Lecture Notes in Computer Science, pages 67–78. Springer, July 2009.

[Col09a] Thomas Colcombet. Regular cost functions over words. Manuscript available online, 2009.

[Col09b] Thomas Colcombet. Regular cost functions, part i: logic and algebra over words. Submitted, 2009.

[Col09c] Thomas Colcombet. The theory of stabilization monoids and regular cost functions. *ICALP, Lecture Notes in Computer Science*, 2009.

[Egg63] L. C. Eggan. Transition graphs and the star-height of regular events. *Michigan Math. J.*, 10:385–397, 1963.

[Has79] Kosaburo Hashiguchi. A decision procedure for the order of regular events. *Theoretical Computer Science*, 8:69–72, 1979.

[Has88] K. Hashiguchi. Relative star height, star height and finite automata with distance functions. In *Formal Properties of Finite Automata and Applications*, pages 74–88, 1988.

[Has90] K. Hashiguchi. Improved limitedness theorems on finite automata with distance functions. *Theor. Comput. Sci.*, 72:27–38, 1990.

[Kir04] Daniel Kirsten. Desert automata and the finite substitution problem. In *STACS*, volume 2996 of *Lecture Notes in Computer Science*, pages 305–316. Springer, 2004.

[Kir05] Daniel Kirsten. Distance desert automata and the star height problem. *RAIRO*, 3(39):455–509, 2005.

[Kle56] Stephen C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 3–42. Princeton University Press, Princeton, New Jersey, 1956.

[RS59] Michael O. Rabin and Dana Scott. Finite automata and their decision problems. *IBM J. Res. and Develop.*, 3:114–125, April 1959.

[Sch65] M.-P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control 8*, pages 190–194, 1965.

[Sim78] Imre Simon. Limited subsets of a free monoid. In *FOCS*, pages 143–150, 1978.

A Appendix

A.1 Some proofs on S-automata

***temp*$_S$-side of Lemma 2** We will construct a rational language L $\times$2-clock-form of $temp_S$.

We will say that a clock c is *compatible* with a word u on $\{i, cr\}^+$ if u and c have the same length and there is at most one $\downarrow$ of the clock c in some block of i's ended by a cr in u. Let $L = \{\langle u, c\rangle, c \text{ compatible with } u\}$

$$L = K[((cr, \lrcorner) + (cr, \downarrow))K]^*((i, \lrcorner) + (i, \downarrow))^* \quad \text{in which} \quad K = (i, \lrcorner)^* + (i, \lrcorner)^*(i, \downarrow)(i, \lrcorner)^* \ .$$

We will show that L is a $\times$2-clock-form of $temp_S$. We just need to show that $temp_S \preccurlyeq \varphi_L^B$ and $\langle\!\langle L\rangle\!\rangle_S \preccurlyeq temp_S$.

Let $u \in \{a, b\}^+$, and c be a clock on u such that $\langle u, c\rangle \in L$ and max$-$seg(c) is minimal. Let $n = temp_S(u)$ be the size of the smallest block of i's in u followed by a cr. If max$-$seg$(c) \leq \lfloor n/2 \rfloor$, there is at least two $\downarrow$ of c in the smallest (therefore in any) block of i's followed by a cr in u, so c cannot be compatible with u. Hence we must have $\langle\!\langle L\rangle\!\rangle_B(u) = \max{-}\text{seg}(c) > \lfloor n/2 \rfloor = \lfloor temp_S(u)/2 \rfloor$. This is true for all u, so $temp_S \preccurlyeq_{\times 2} \langle\!\langle L\rangle\!\rangle_B$.

We now need to show that $\langle\!\langle L\rangle\!\rangle_S \preccurlyeq temp_S$.

Let $u \in \mathbb{A}^+$, and c be a clock on u such that $\langle u, c\rangle \notin L$ and min$-$seg(c) is maximal. $\langle u, c\rangle \notin L$ implies that there is two $\downarrow$ in c in any block of i's ended by a cr in u. This implies min$-$seg$(c) \leq temp_S(u)$. Hence by definition of c, $\langle\!\langle L\rangle\!\rangle_S(u) = \min{-}\text{seg}(c) \leq temp_S(u)$. It is true for all u so $\langle\!\langle L\rangle\!\rangle_S \leq temp_S$. □

***S*-side of Theorem 7** (4)⇒(1)

Consider a S-automaton $\mathcal{A} = \langle Q, \mathbb{A}, In, Fin, \Gamma, \Delta\rangle$ with $\Gamma = \{\gamma_1, \ldots, \gamma_k\}$.

A run of $\mathcal{A}$ is as a word on alphabet $\Delta \subseteq Q \times \mathbb{A} \times \{\mathrm{i}, \mathrm{cr}\}^\Gamma \times Q$.

It follows from the definition of $[\![\cdot]\!]_S$ that for all $u \in \mathbb{A}^*$:

$$[\![\mathcal{A}]\!]_S(u) = \sup_{\sigma \in \Delta^*} \{\min(\chi_{\complement R}(\sigma), temp_S \circ \pi_1(\sigma), \cdots, temp_S \circ \pi_k(\sigma)) \ : \ \pi_{\mathbb{A}}(\sigma) = u\}$$

in which $R \subseteq \Delta^*$ is the (regular) set of valid runs; for all $i \in [\![1, k]\!]$, π_i projects each transition (p, a, t, q) to the γ_i^{th} component of t (and is extended to words). Finally $\pi_{\mathbb{A}}$ projects each transition (p, a, t, q) to a (and is also extended to words). By Example 4, $\chi_{\complement R} \in \mathcal{CF}$. By Lemma 2, $temp_S \in \mathcal{CF}$, and by Theorem 6, $\mathcal{CF}$ is stable under composition, min and sup-projection. Hence $[\![\mathcal{A}]\!]_S \in \mathcal{CF}$. □

A.2 Cost sequences

The aim is to give a semantic to stabilization semigroups. Some mathematical preliminaries are required.

Let $(E, \leq)$ be an ordered set, α a function from $\mathbb{N}$ to $\mathbb{N}$, and $\boldsymbol{a}, \boldsymbol{b} \in E^{\mathbb{N}}$ two sequences. We define the relation $\preceq_\alpha$ by $\boldsymbol{a} \preceq_\alpha \boldsymbol{b}$ if :

$$\forall n. \forall m. \quad \alpha(n) \leq m \rightarrow \boldsymbol{a}(n) \leq \boldsymbol{b}(m) \ .$$

A sequence $\boldsymbol{a}$ is said *α-non-decreasing* if $\boldsymbol{a} \preceq_\alpha \boldsymbol{a}$. We define $\sim_\alpha$ as $\preceq_\alpha \cap \succeq_\alpha$, and $\boldsymbol{a} \preceq \boldsymbol{b}$ (resp. $\boldsymbol{a} \sim \boldsymbol{b}$) if $\boldsymbol{a} \preceq_\alpha \boldsymbol{b}$ (resp. $\boldsymbol{a} \sim_\alpha \boldsymbol{b}$) for some α.

Remarks:

- if $\alpha \leq \alpha'$ then $\boldsymbol{a} \preceq_\alpha \boldsymbol{b}$ implies $\boldsymbol{a} \preceq_{\alpha'} \boldsymbol{b}$,
- if $\boldsymbol{a}$ is α-non-decreasing, then it is α-equivalent to a non-decreasing sequence,
- $\boldsymbol{a}$ is id-non-decreasing iff it is non-decreasing,
- let $\boldsymbol{a}, \boldsymbol{b} \in E^{\mathbb{N}}$ be two non-decreasing sequences, then $\boldsymbol{a} \preceq_\alpha \boldsymbol{b}$ iff $\boldsymbol{a} \circ \alpha \leq \boldsymbol{b}$.

The α-non-decreasing sequences ordered by $\preceq_\alpha$ can be seen as a weakening of the $\alpha = id$ case. We will identify the elements $a \in E$ with the constant sequence of value a.

The relations $\preceq_\alpha$ and $\sim_\alpha$ are not transitives, but the following property guarantees a certain kind of transitivity.

Fact 13 *$\boldsymbol{a} \preceq_\alpha \boldsymbol{b} \preceq_\alpha \boldsymbol{c}$ implies $\boldsymbol{a} \preceq_{\alpha\circ\alpha} \boldsymbol{c}$ and $\boldsymbol{a} \sim_\alpha \boldsymbol{b} \sim_\alpha \boldsymbol{c}$ implies $\boldsymbol{a} \sim_{\alpha\circ\alpha} \boldsymbol{c}$.*

The function α is used as a "precision" parameter for $\sim$ and $\preceq$. Fact 13 shows that a transitivity step cost some precision. For any α, the relation $\preceq_\alpha$ coincide over constant sequences with order $\leq$ (up to identification of constant sequence with their constant value). In consequence, the sequence in $E^{\mathbb{N}}$ ordered by $\preceq_\alpha$ form an extension of $(E, \leq)$.

In the following, while using relations $\preceq_\alpha$ and $\sim_\alpha$, we may forget the subscript α and verify instead that the proof has a bounded number of transitivity steps.

Let$(E, \leq)$ and $(F, \leq)$ two ordered sets, a function $E \to F^{\mathbb{N}}$ is *α-monotone* if

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

In particular, for each $a \in E$, we have $a \leq a$, so $f(a) \preceq_\alpha f(a)$, hence $f(a)$ is α-non-decreasing. To each α-monotone function $f : E \to F^{\mathbb{N}}$ we associate $\tilde{f} : E^{\mathbb{N}} \to F^{\mathbb{N}}$ defined in the following way:

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

Proposition 14 *Let $f : E \to F^{\mathbb{N}}$ be a α-monotone function and $\boldsymbol{a}, \boldsymbol{b} \in E^{\mathbb{N}}$, then:*

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

In particular, if $f : E \to F^{\mathbb{N}}$ and $g : F \to G^{\mathbb{N}}$ are α-monotone, then $\tilde{g} \circ f$ is α-monotone. Moreover, $\widetilde{(\tilde{g} \circ f)} = \tilde{g} \circ \tilde{f}$.

Definition 15 *If f and g are functions $E \to F^{\mathbb{N}}$, we will say that $f \sim_\alpha g$ if for all $u \in E$, $f(u) \sim_\alpha g(u)$. As usual, $f \sim g$ if there exists α such that $f \sim_\alpha g$.*

We will also use this notions with the product order : if $(E, \leq)$ is an ordered set, the set of words in $u \in E^*$ is canonically ordered by $a_1 \dots a_n \leq b_1 \dots b_m$ iff $m = n$ and $a_i \leq b_i$ for $i = 1 \dots n$. We identify the elements of $(E^{\mathbb{N}})^*$ (words of sequences) with some elements of $(E^*)^{\mathbb{N}}$ (sequences of words of the same length). Notice that for any sequences $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n, \boldsymbol{b}_1, \dots, \boldsymbol{b}_n \in E^{\mathbb{N}}$, $\boldsymbol{a}_1 \dots \boldsymbol{a}_n \preceq_\alpha \boldsymbol{b}_1 \dots \boldsymbol{b}_n$ iff $\boldsymbol{a}_i \preceq_\alpha \boldsymbol{b}_i$ for $i = 1 \dots n$.

A.3 Ideals of an ordered set

This notion will be essential to define the cost function recognized by a stabilization semigroup.

Let $(E, \leq)$ be an ordered set, an *ideal* is a $\leq -$*closed* subset $I \subseteq E$, i.e. if $a \in I$ and $b \leq a$, then $b \in I$. let $a \in E$, the'*ideal generated by* a is $I_a = \{b \in E \ : \ b \leq a\}$. Let $\boldsymbol{a} \in E^{\mathbb{N}}$ and I be an ideal, we define $I[\boldsymbol{a}] = \sup\{n+1 \ : \ \boldsymbol{a}(n) \in I\}$.[4] Let I be an ideal, its complement in E is denoted by $\overline{I}$. Let $\boldsymbol{a} \in E^{\mathbb{N}}$, we define $\overline{I}[\boldsymbol{a}] = \inf\{n \ : \ \boldsymbol{a}(n) \in \overline{I}\}$.

[4] The $+1$ makes the calculus smoother in the following.

Proposition 16 *Let f and g be functions $E \to S^{\mathbb{N}}$ such that $f \sim_\alpha g$ and for any $u \in E$, $f(u)$ and $g(u)$ are non-decreasing. Then for any ideal I of $\mathbf{S}$, the cost functions $u \mapsto I[f(u)]$ and $u \mapsto I[g(u)]$ are $\approx_\alpha$ equivalent.*

Indeed, let $u \in E$, and $n = I[f(u)]$. Then $g(u)(\alpha(n)) \geq f(u)(n) \notin I$. I is an ideal so we get $g(u)(\alpha(n)) \notin I$. $g(u)$ is non-decreasing so $I[g(u)] \leq \alpha(n)$. By symmetry of f and g we finally get $u \mapsto I[f(u)] \approx_\alpha u \mapsto I[g(u)]$.

Definition 17 *Let $a, b \in E$ and $n \in \mathbb{N}$, we define the sequence $a|_n b$ by:*

$$\text{for all } k \in \mathbb{N}, \qquad (a|_n b)(k) = \begin{cases} a & \text{if } k < n, \\ b & \text{otherwise.} \end{cases}$$

A.4 Compatible functions

We now define the semantic of a stabilization semigroup with the notion of compatible function. The idea is to generalize the notion of product, by associating to each word of S^+, no longer an element of S, but a cost sequence in $S^{\mathbb{N}}$. this will allow us to express stabilization in a quantitative way. Intuitively, when n is fixed in the cost sequence, we can interpret the semantic as an automaton with limited resources. To avoid ambiguities, we will write uv the concatenation of u and v as words in S^+ and $a \cdot b$ the product of a and b as elements of $\mathbf{S}$.

$\langle S^+, \ , \leq \rangle$ forms a semigroup, partially ordered by the product ordered between words of same length described above.

Definition 18 *Let $\mathbf{S} = \langle S, \cdot, \leq, \sharp \rangle$ be stabilization semigroup. A function ρ from S^+ to $S^{\mathbb{N}}$ is said* compatible with $\mathbf{S}$ *if there exists α such that :*

Monotonicity. *ρ is α-monotone,*
Letter. *for all $a \in S$, $\rho(a) \sim_\alpha a$,*
Product. *for all $a, b \in S$, $\rho(ab) \sim_\alpha a \cdot b$,*
Stabilization. *for all $e \in E(\mathbf{S})$, $m \in \mathbb{N}$, $\rho(e^m) \sim_\alpha (e^\sharp|_m e)$,*
Substitution. *for all $u_1, \ldots, u_n \in S^+$, $n \in \mathbb{N}$, $\rho(u_1 \ldots u_n) \sim_\alpha \tilde{\rho}(\rho(u_1) \ldots \rho(u_n))$ (remind : we identify sequence of words and word of sequences, see section A.2)*

Example 7. Let $\mathbf{S}$ be the stabilization semigroup with 3 elements $\bot \leq a \leq b$, with product defined by : $x \cdot y = \min_\leq(x, y)$ (b neutral element), and stabilization by $b^\sharp = b$ and $a^\sharp = \bot^\sharp = \bot$. Lett $u \in \{\bot, a, b\}^+$, we define ρ by:

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

Then ρ is compatible with $\mathbf{S}$.

Remark 19 *When $^\sharp$ is the identity function, $\mathbf{S}$ becomes a standard ordered semigroup, and the classical extended product π is compatible with $\mathbf{S}$.*

Theorem 20 ([Col09c]) *For any stabilization semigroup $\mathbf{S}$, there exists a function ρ compatible with $\mathbf{S}$. Moreover, ρ is unique up to $\sim$.*

This theorem is fundamental, since it associates a unique (up to $\sim$) semantic to any stabilization semigroup.

Lemma 3. *Let ρ compatible with a semigroup $\mathbf{S}$. There exists γ such that for any $n \in \mathbb{N}$ and $u \in S^+$, if $|u| \leq n$ then for all $k \geq \gamma(n), \rho(u)(k) = \pi(u)$*

Proof. We show this result by induction on n. It is true for $n = 1$ by taking $\gamma(1) = 1$. We assume $\gamma(k)$ constructed for $k < n$, and we want to show the result for n. Let $u \in S^+$ of length n, $u = va$ with $|v| = n-1$ and $a \in S$. Letα a witness of ρ compatible with $\mathbf{S}$. The substitution property tells us that $\rho(u) \sim_\alpha \tilde{\rho}(\rho(v)a)$. but by induction hypothesis, for all $k \geq \gamma(n-1)$, $\tilde{\rho}(\rho(v)a)(k) = \rho(\rho(v)(k)a)(k) = \rho(\pi(v)a)(k)$. Moreover, $\rho(\pi(v)a) \sim_\alpha \pi(v) \cdot a = \pi(u)$. Hence we have for all $k \geq \alpha(\gamma(\alpha(n-1))), \rho(u)(k) = \pi(u)$.We get the result with $\gamma(n) = \alpha(\gamma(\alpha(n-1)))$. □

A.5 Generalities about stabilization semigroups

Structure An *idempotent* element of $\mathbf{S}$ is an element $e \in S$ such that $e \cdot e = e$. We note $E(\mathbf{S})$ the set of idempotent elements of $\mathbf{S}$.

In the sequel, we use a classic Green's relation. Let a and b be in S; we denote $a \leq_\mathcal{J} b$ if there exists x and y in $S \cup \{1\}$ such that $a = xby$. The relation $\leq_\mathcal{J}$ is a preorder. If $a \leq_\mathcal{J} b$ and $b \leq_\mathcal{J} a$, then a and b are in the same $\mathcal{J}$-class, and we denote $a\mathcal{J}b$. Obviously, $\leq_\mathcal{J}$ induces an order over $\mathcal{J}$-classes, also noted $\leq_\mathcal{J}$.

A *regular element* of $\mathbf{S}$ is an element a such that there exists $e \in E(\mathbf{S})$ with $a\mathcal{J}e$. Consequently, either all the elements of a $\mathcal{J}$-class are regular (we say that the $\mathcal{J}$-class is regular), either no element is (the $\mathcal{J}$-class is *irregular*).

We can extend $\sharp$ to all regular elements of S. If a is a regular element, there exists $e \in E(\mathbf{S})$ and $x, y \in S \cup \{1\}$ such that $x \cdot e \cdot y = a$. We define then $a^\sharp = x \cdot e^\sharp \cdot y$, which does not depend on the choice of the decomposition (*cf.* [Kir05]).

Definition 21 *A regular $\mathcal{J}$-class J is* stable *if there exists an idempotent a in J such that $a^\sharp \in J$, otherwise J is* unstable. *If J is stable, then for all idempotent a in J, $a^\sharp = a$.*

Definition 22 (Product of stabilization semigroups) *Let $\mathbf{S}_1 = \langle S_1, \cdot_1, \leq_1, \sharp_1 \rangle$ and $\mathbf{S}_2 = \langle S_2, \cdot_2, \leq_2, \sharp_2 \rangle$ be stabilization semigroups, then their product $\mathbf{S}_1 \times \mathbf{S}_2$ is the tuple $\langle S_1 \times S_2, \cdot, \leq, \sharp \rangle$ such that $(a_1, a_2) \cdot (b_1, b_2) = (a_1 \cdot b_1, a_2 \cdot b_2)$, $(a_1, a_2) \leq (b_1, b_2)$ if $a_1 \leq b_1$ and $a_2 \leq b_2$, and $(e_1, e_2)^\sharp = (e_1^{\sharp_1}, e_2^{\sharp_2})$.*

Proposition 23 *If $\mathbf{S}_1$ and $\mathbf{S}_2$ are stabilization semigroups, then $\mathbf{S}_1 \times \mathbf{S}_2$ is one too. Moreover, if ρ_1 is compatible with $\mathbf{S}_1$ and ρ_2 with $\mathbf{S}_2$ then ρ defined for all $u = (u_1, u_2) \in (S_1 \times S_2)^+$ and $k \in \mathbb{N}$ by $\rho(u)(k) = (\rho_1(u_1)(k), \rho_2(u_2)(k))$, is compatible with $\mathbf{S}_1 \times \mathbf{S}_2$.*

Definition 24 *A function ϕ from $\mathbf{S}$ to $\mathbf{S}'$ is a morphism of stabilization semigroups if*

- *for all u, v in $\mathbf{S}$, $\phi(u \cdot v) = \phi(u) \cdot \phi(v)$,*
- *For all $u \in E(\mathbf{S})$, $\phi(u) \in E(\mathbf{S}')$ and $\phi(u^\sharp) = \phi(u)^\sharp$.*

Lemma 4. *Let $\mathbf{S}, \mathbf{S}'$ be stabilization semigroups, ρ and ρ' compatible with $\mathbf{S}$ and $\mathbf{S}'$. We assume there exists a morphism of stabilization semigroups τ from $\mathbf{S}$ to $\mathbf{S}'$. Let $\tau^+ : \mathbf{S}^+ \to \mathbf{S}'^+$ and $\tau^\mathbb{N} : \mathbf{S}^\mathbb{N} \to \mathbf{S}'^\mathbb{N}$ the natural extensions of τ to finite and infinite sequences. Then $\tau^\mathbb{N} \circ \rho \sim \rho' \circ \tau^+$.*

Proof. Let $K = \{(a, \tau(a)), a \in S\}$. We can provide K with a structure of stabilization semigroup, as a sub-stabilization semigroup of $\mathbf{S} \times \mathbf{S}'$.

Let $\phi : K^+ \to K^\mathbb{N}$ defined by $\phi(u, \tau^+(u)) = (\rho(u), \tau^\mathbb{N}(\rho(u))$. Let α be a witness of ρ compatible with $\mathbf{S}$. We show that ϕ is compatible with K.

- *Monotonicity.* ρ is α-monotone, so ϕ is too
- *Letter.* Let $a \in K$,$a = (b, \tau(b))$ with $b \in S$, $\phi(a) = (\rho(b), \tau(\rho(b)) \sim_\alpha (b, \tau(b)) = a$
- *Product.* Let $a, b \in K$, $a = (a', \tau(a')), b = (b', \tau(b')), \phi(ab) = (\rho(a'b'), \tau(\rho(a'b'))) \sim_\alpha (a' \cdot b', \tau(a' \cdot b')) = a \cdot b$,
- *Stabilization.* Let $e \in E(K)$, $m \in \mathbb{N}$, $e = (a, \tau(a))$ with $a \in E(\mathbf{S})$, $\phi(e^m) = (\rho(a^m), \tau^{\mathbb{N}}(\rho(a^m))) \sim_\alpha (a^\sharp|_m a, \tau^{\mathbb{N}}(a^\sharp|_m a)) = e^\sharp|_m e$,
- *Substitution.* Let $u_1, \ldots, u_n \in S^+, n \in \mathbb{N}, \forall i, \exists v_i \in S^+, u_i = (v_i, \tau^+(v_i)), \phi(u_1 \ldots u_n) = (\rho(v_1 \ldots v_n), \tau^{\mathbb{N}}(\rho(v_1 \ldots v_n))) \sim_\alpha (\tilde{\rho}(\rho(v_1) \ldots \rho(v_n)), \tau^{\mathbb{N}}(\tilde{\rho}(\rho(v_1) \ldots \rho(v_n)))$. We get $\phi \sim_\alpha g$ with $g(u_1 \ldots u_n)(k) = (\rho(\rho(v_1)(k) \ldots \rho(v_n)(k))(k), \tau^{\mathbb{N}}(\rho(\rho(v_1)(k) \ldots \rho(v_n)(k))(k)))$. To conclude,

$$\begin{aligned}
\tilde{\phi}(\phi(u_1) \ldots \phi(u_n))(k) &= \phi((\rho(v_1), \tau^{\mathbb{N}}(\rho(v_1))(k) \ldots (\rho(v_n), \tau^{\mathbb{N}}(\rho(v_n))(k))(k) \\
&= \phi(\rho(v_1)(k) \ldots \rho(v_n)(k), \tau^+(\rho(v_1)(k) \ldots \rho(v_n)(k))(k) \\
&= (\rho(\rho(v_1)(k) \ldots \rho(v_n)(k))(k), \tau^+(\rho(\rho(v_1)(k) \ldots \rho(v_n)(k))(k))) \\
&\sim_\alpha \phi(u_1 \ldots u_n)(k)
\end{aligned}$$

But (ρ, ρ') is also compatible with K, since K is a sub-stabilization semigroup of $S \times S'$ (Proposition 23). The uniqueness (up to $\sim$) of the compatible function (Theorem 20) gives us $\phi \sim (\rho, \rho')$, hence by projection on the second component, we get $\tau^{\mathbb{N}} \circ \rho \sim \rho' \circ \tau^+$. □

A.6 Recognized cost functions

We now have all the mathematical tools to define how stabilization semigroups can recognize cost functions.

Let $\mathbf{S} = \langle S, \cdot, \leq, \sharp \rangle$ be a stabilization semigroup. Let $h : \mathbb{A} \to S$ be a morphism, canonically extended to $h : \mathbb{A}^+ \to S^+$, and $I \subseteq S$ an ideal. Then the triplet $\mathbf{S}, h, I$ *recognizes* the function $f : \mathbb{A}^+ \to \mathbb{N}_\infty$ defined by $f(u) = I[\rho(h(u))]$ where ρ is compatible with $\mathbf{S}$. A cost function from $\mathbb{A}^+$ to $\mathbb{N}_\infty$ is said recognizable if it is $\approx$-equivalent to a function recognized by some $\mathbf{S}, h, I$. By Proposition 16, the recognized cost function does not depend on the choice of ρ.

Example 8. Let $\mathbb{A} = \{a, b\}$, the cost function $|\cdot|_a$ is recognizable. We take the stabilization semigroup from Example 7, h defined by $h(a) = a, h(b) = b$, and $I = \{\perp\}$. We have then $|u|_a = I[\rho(h(u))]$ for all $u \in \mathbb{A}^+$.

The following result shows the analogy with regular languages, and justify the name of "regular" cost functions :

Theorem 25 ([Col09c]) *For a cost function, the following properties are equivalent :*

- *being recognizable by stabilization semigroup,*
- *being computable by B-automaton,*
- *being computable by S-automaton.*

We can report [Col09c] for more details about stabilization semigroups, in particular for interesting decidability results about applications to regular language theory.

A.7 Minimization

We will show that for any given $\mathbf{S}, h, I$ recognizing a regular cost function f, we can build a (quotient-wise) minimal stabilization semigroup recognizing f.

If $X \subseteq S$, we note $\langle X \rangle^\sharp$ the closure of X in S by product and stabilization. We can assume that $\mathbf{S}$ only has "useful" elements i.e. $S = \langle h(\mathbb{A}) \rangle^\sharp$.

Let $\equiv$ be the coarsest equivalence relation on $\mathbf{S}$ such that : $\forall x, y, a \in S$,

$$\begin{cases} x \equiv y \Rightarrow (x \in I \Leftrightarrow y \in I) \\ x \equiv y \Rightarrow a \cdot x \equiv a \cdot y \\ x \equiv y \Rightarrow x \cdot a \equiv y \cdot a \\ x \equiv y \Rightarrow x^\sharp \equiv y^\sharp \end{cases}$$

in other words $\equiv$ is the coarsest equivalence relation saturating I (in particular $\mathbf{S}/\equiv$ is a stabilization semigroup). This relation can be computed effectively, starting from whole $S \times S$ then iteratively removing couples which don't verify the above conditions. This is a kind of Moore algorithm, and its complexity is polynomial in $|S|$.

Theorem 26 *$\mathbf{S}/\equiv$ recognizes f.*

Proof. Let τ be the canonical projection $S \longrightarrow S/\equiv$ naturally extended to $\tau^+ : S^+ \longrightarrow (S/\equiv)^+$, and also to $\tau^{\mathbb{N}} : S^{\mathbb{N}} \longrightarrow (S/\equiv)^{\mathbb{N}}$.

We define $I' = \tau(I)$, $h' = \tau^+ \circ h$, and we want to show that $\mathbf{S}/\equiv, h', I'$ recognizes f.

Let ρ' compatible with $\mathbf{S}/\equiv$. By Lemma 4, there exists α such that

$$\forall u \in \mathbf{S}^+, \tau^{\mathbb{N}}(\rho(u)) \sim_\alpha \rho'(\tau^+(u))$$

If $u \in \mathbb{A}^+$, we have $I[\rho(h(u))] = I'[\tau^{\mathbb{N}}(\rho(h(u)))]$ and $I'[\rho'(h'(u))] = I'[\rho'(\tau^+(h(u)))]$. By Proposition 16, the functions $u \mapsto I'[\tau^{\mathbb{N}}(\rho(h(u)))]$ and $u \mapsto I'[\rho'(\tau^+(h(u)))]$ are thus $\approx_\alpha$-equivalent. We conclude that $\mathbf{S}/\equiv, h', I'$ recognizes the cost function f. □

In order to show that $\mathbf{S}/\equiv$ is minimal for the quotient relation, we have to build set of words which we will use as counter-examples. We need for that a tool introduced by Hashigushi, called $\sharp$-expression.

Definition 27 ($\sharp$-expression) *[Has90] We define the* $\sharp$-expressions *by induction. Every letter $a \in \mathbb{A}$ is a $\sharp$-expression, if e and e' are $\sharp$-expressions, ee' and $e^\sharp$ are $\sharp$-expressions.*

If e is a $\sharp$-expression and $k \in \mathbb{N}$, we define the word $e(k)$ by induction in the following way : if e is a letter then $e(k) = e$, and if e and e' are $\sharp$-expressions, $ee'(k) = e(k)e'(k)$ and $e^\sharp(k) = e(k)^k$.

We also define an operation eval (depending on the semigroup and the morphism h)to associate a value to any $\sharp$-expression by induction : if e is a letter then $eval(e) = h(e)$, and if e and e' are $\sharp$-expressions, $eval(ee') = eval(e) \cdot eval(e')$ and $eval(e^\sharp) = eval(e)^\sharp$ ($eval(e)$ has to be an idempotent). A $\sharp$-expression is well-formed if $eval(e)$ exists. For all $k \in \mathbb{N}$, we define $\lim_k$ in the same way that eval except for $\lim_k(e^\sharp) = \lim_k(e)^k$ (it is a product, not a concatenation).

Definition 28 (Context) *A* context *$C[]$ is a $\sharp$-expression with a possible occurrence of a free variable x. If e is a $\sharp$-expression, $C[e]$ is the $\sharp$-expression obtained by replacing x by e in $C[]$.*

Example 9. If $\mathbb{A} = \{a, b, c\}$, $e = ab(bc^\sharp b)^\sharp a^\sharp bb$ is a $\sharp$-expression,
$e(3) = abbcccbbcccbbcccbaaabb$ and $eval(e) = h(a)h(b)(h(b)h(c)^\sharp h(b))^\sharp h(a)^\sharp h(b)h(b)$.
An example of context is $C[] = ab(ax^\sharp)^\sharp$, we have $C[b] = ab(ab^\sharp)^\sharp$

The following lemma shows how $\sharp$-expression behave relatively to compatible functions.

Lemma 5. *For all $\sharp$-expression e, there exists a α_e such as for all $k \in \mathbb{N}$, $\rho(h(e(k))) \sim_{\alpha_e} eval(e)|_k \lim_k(e)$.*

Proof. Let β be a witness of ρ compatible with $\mathbf{S}$. We proceed by induction on e:

- if e is a letter, then for all $k \in \mathbb{N}, \rho(h(e(k))) \sim_\beta h(e) = eval(e)|_k \lim_k(e)$.
- if $e = rs$, then for all $k \in \mathbb{N}, \rho(h(e(k))) = \rho(h(r(k))h(s(k))) \sim_\beta \tilde{\rho}(\rho(h(r(k)))\rho(h(s(k)))$, but by induction hypothesis, there exists α_r and α_s such as $\rho(h(r(k))) \sim_{\alpha_r} eval(r)|_k \lim_k(r)$ and $\rho(h(s(k))) \sim_{\alpha_s} eval(s)|_k \lim_k(s)$, so by choosing $\alpha_e = \beta \circ \max(\alpha_r, \alpha_s)$, we get the result.
- if $e = r^\sharp$ with $eval(r)$ idempotent, then for all $k \in \mathbb{N}$,

$$\begin{aligned}\rho(h(e(k))) &\sim_\beta \tilde{\rho}(\rho(h(r(k)))^k) \\ &\sim_{\alpha_r} \tilde{\rho}((eval(r)|_k \lim_k(r))^k) \\ &\sim_{\beta\circ\gamma} eval(r)^\sharp|_k \lim_k(r) \\ &= eval(e)|_k \lim_k(e)\end{aligned}$$

We get the result with $\alpha_e = \beta \circ \alpha_r \circ \beta \circ \gamma$, where γ comes from Lemma 3.

□

Lemma 6. *If $\mathbf{S}, h, I$ and $\mathbf{S}', h', I'$ recognize the same cost function, then $\mathbf{S}/\equiv$ and $\mathbf{S}'/\equiv'$ are isomorphic.*

Proof. We will consider here that all elements of the semigroups are accessible by product and stabilization from $h(\mathbb{A})$.

$\mathbf{S}, h, I$ and $\mathbf{S}'/\equiv', h', I'$ recognize f. We will show that there is a surjective morphism ϕ from $\mathbf{S}$ to $\mathbf{S}'/\equiv'$.

Let $eval$ and $eval'$ be the evaluations relatively to $\mathbf{S}, h$ and $\mathbf{S}'/\equiv', h'$.

For all $\sharp$-expression e, let $\phi(eval(e)) = eval'(e)$. We show that it is indeed a surjective morphism.

Let assume that there exist e_1, e_2 some $\sharp$-expressions such that $eval(e_1) = eval(e_2)$ but $eval'(e_1) \neq eval'(e_2)$. By the definition of $\equiv'$, $eval'(e_1)$ and $eval'(e_2)$ can be distinguished by I', so there is a context $C[]$ such that $eval'(C[e_1]) \in I'$ and $eval'(C[e_2]) \notin I'$ (up to reversing e_1 and e_2). But we have $eval(C[e_1]) = eval[C(e_2)]$.

If $eval(C[e_1]) \in I$, let $u_k = C[e_2](k)$ for all $k \in \mathbb{N}$.

Let ρ, ρ' be compatible with $\mathbf{S}, \mathbf{S}'$. By Lemma 5, there exists α such that $\rho(h(C[e_2](k))) \sim_\alpha eval(C[e_2])|_k \lim_k(C[e_2])$ and $\rho'(h'(C[e_2](k))) \sim_\alpha eval'(C[e_2])|_k \lim'_k(C[e_2])$ We have $eval(C[e_2]) \in I$ so $I[\rho(h(u_k))] \geq_\alpha k$, but $eval'(C[e_2]) \notin I$, so $I'[\rho'(h'(u_k))] = 0$ for k large enough. However, $\mathbf{S}, h, I$ and $\mathbf{S}', h', I'$ recognize the same cost function. Hence we have a contradiction.

In the case where $eval(C[e_1]) \notin I$, we can do the symmetrical reasoning and take $u_k = C[e_1](k)$, we also get a contradiction. In conclusion, such a couple e_1, e_2 cannot exist, hence ϕ is well defined. Moreover, ϕ is a surjection because we limited the semigroups to $\langle h(\mathbb{A})\rangle^\sharp$ to build $S'/\equiv'$.

The only thing left to check is that ϕ is a morphism of stabilization semigroups. Let $a, b \in \mathbf{S}$. By hypothesis on $\mathbf{S}$, there exists e_a, e_b such that $a = eval(e_a)$ and $b = eval(e_b)$. We have $\phi(a \cdot b) = \phi(eval(e_a) \cdot eval(e_b)) = \phi(eval(e_a e_b)) = eval'(e_a e_b) =$

$eval'(e_a) \cdot eval'(e_b) = \phi(a) \cdot \phi(b)$, and $\phi(a^\sharp) = \phi(eval(e_a)^\sharp) = \phi(eval(e_a^\sharp)) = eval'(e_a^\sharp) = eval'(e_a)^\sharp = \phi(a)^\sharp$.

ϕ is a surjective morphism of stabilization semigroups from $\mathbf{S}$ to $\mathbf{S}'/\equiv'$. By reversing the roles of $\mathbf{S}$ and $\mathbf{S}'$, we get that $\mathbf{S}/\equiv$ and $\mathbf{S}'/\equiv'$ are isomorphic. □

A.8 Temporal fragment

Proposition 29 *Let $\mathbf{S} = \langle S, \cdot, \leq, \sharp \rangle$ be a stabilization semigroup. $\mathbf{S}$ is a temporal semigroup if the following condition holds. Let J be a stable $\mathcal{J}$-class, for every $\mathcal{J}$-class J', if $J \geq_{\mathcal{J}} J'$, then J' is stable.*

Proof of Theorem 12 The aim is to associate a temporal semigroup to any temporal B-automaton, in a way that both objects recognize the same cost function. We first associate a temporal semigroup. We start by building the temporal semigroup representing the operations on one counter in a temporal B-automaton.

Let $\mathbf{S}_\gamma = \langle S, \cdot, \leq, \sharp \rangle$ with $S = \{ic, r, \bot\}$. All elements are idempotent, $\bot$ is a zero, $ic \cdot r = r \cdot ic = r = r^\sharp$, $ic^\sharp = \bot^\sharp = \bot$, and $\bot \leq ic \leq r$. The semigroup $\mathbf{S}_\gamma$ describes the semantic of the operations on one counter.

Let $\mathcal{A} = \langle Q, \mathbb{A}, In, Fin, \gamma, \Delta \rangle$ be a temporal B-automaton (we can take it with one counter by Theorem 7). We associate to it a stabilization semigroup $\mathbf{S}_\mathcal{A}$ in the following way:

Let $S_\mathcal{A} = S_\gamma^{Q \times Q}$.

If $E, F \in S_\mathcal{A}$, we define their product by :

$$\forall p, t \in Q, E \cdot F(p,t) = \max\{E(p,q) \cdot F(q,t), q \in Q\},$$

and the order by $E \leq F$ iff for all $p, q \in Q, E(p,q) \leq F(p,q)$. Finally, if E is an idempotent, we define $E^\sharp$ by:

$$\forall p, q \in Q, E^\sharp(p,q) = \max\{E(p,t) \cdot E(t,t)^\sharp \cdot E(t,q) \ / \ t \in Q\}.$$

Theorem 30 *[Col09c] $\mathbf{S}_\mathcal{A} = \langle S_\mathcal{A}, \cdot, \leq, \sharp \rangle$ is a stabilization semigroup, and by taking $I = \{E / \forall (p,q) \in In \times Fin, E(p,q) = \bot\}$ and $h(a)(p,q) = \max\{\sigma / (p, a, \sigma, q) \in \Delta\}$ for all $a \in \mathbb{A}$ and $p, q \in Q$, $\mathbf{S}_\mathcal{A}, h, I$ recognizes $[\![\mathcal{A}]\!]_B$.*

We still have to show that $\mathbf{S}_\mathcal{A}$ is a temporal stabilization semigroup.

Lemma 7. *Let E be an idempotent and $p, q \in Q$, then there exists $t \in Q$ such that $E(p,q) \leq E(p,t) \cdot E(t,t) \cdot E(t,q)$.*

Proof. We can write $E = E \cdots E$, product of length k, with $k > |Q|$. There exists a sequence $p_0, a_1, p_1, \ldots, p_k$ such that $p_0 = p$, $p_k = q$ and $E(p,q) = a_1 \cdots a_k$. (it is the sequence realizing the max in the definition of the product of $\mathbf{S}_\mathcal{A}$). But $k > |Q|$ so $\exists (j,l), 1 < j < l < k$ and $p_j = p_l = t$. We have $E(p,t) \geq a_1 \cdots a_j$, $E(t,t) \geq a_j + 1 \cdots a_l$ and $E(t,q) \geq a_{l+1} \cdots a_k$, which shows the result. □

Lemma 8. *Let J be $\mathcal{J}$-class of $\mathbf{S}_\mathcal{A}$, then*

J unstable iff there exists $E \in J$ idempotent and $p, q \in Q$ such that $E(p,q) = ic$.

Proof. Let J be an unstable $\mathcal{J}$-class, there exists $E \in J$ idempotent with $E^\sharp \neq E$. But $E^\sharp \leq E$, so there exists $p, q \in Q$ such that $E^\sharp(p,q) < E(p,q)$

By Lemma 7, there is $t \in Q$ such that $E(p,q) \leq E(p,t) \cdot E(t,t) \cdot E(t,q)$. But $E^\sharp(p,q) \geq E(p,t) \cdot E(t,t)^\sharp \cdot E(t,q)$, so we have $E^\sharp(p,q) < E(p,q)$ then $E(t,t)^\sharp \neq E(t,t)$, which implies $E(t,t) = ic$. We have shown the first implication.

Conversely, let $E \in J$ idempotent and $p,q \in Q$ such that $E(p,q) = ic$ If we assume J stable, we get $E^\sharp = E$ so $E^\sharp(p,q) = ic$. By the definition of $E^\sharp$, there exists $t \in Q$ such that $ic = E(p,t) \cdot E(t,t)^\sharp \cdot E(t,q)$. By the definition of the $\cdot$ operation in semigroup S_γ, that implies $E(p,t) = E(t,t)^\sharp = E(t,q) = ic$, and in particular $E(t,t)^\sharp = ic$ is absurd.

Hence J is unstable, this completes the second implication. □

Theorem 31 *$\mathcal{A}$ being a temporal B-automaton, $\mathbf{S}_\mathcal{A}$ is a temporal semigroup.*

Proof. Let J and J' bet two regular $\mathcal{J}$-classes of $\mathbf{S}_\mathcal{A}$ with $J \geq_\mathcal{J} J'$ and J stable. By Lemma 8, $\forall E \in J$ idempotent, $\forall s,t \in Q, E(s,t) \neq ic$.
Let $E' \in J'$, $J \geq_\mathcal{J} J'$ then $\exists E \in J, A, B \in \mathbf{S}_\mathcal{A}, E' = A \cdot E \cdot B$ let us assume there exists $p,q \in Q, E'(p,q) = ic$, then there exists $s,t \in Q, ic = A(p,s) \cdot E(s,t) \cdot B(t,q)$. We must have $E(s,t) = ic$, this is absurd. So by Lemma 8, J' is a stable class. $\mathbf{S}_\mathcal{A}$ is a temporal semigroup. □

We will now do the converse : associate a B-temporal automaton to any temporal semigroup.

Definition 32 *If $\mathbf{S}$ is a temporal semigroup, we define*

$$\mathit{Unstab} = \{x \in \mathbf{S} / \exists e \in E(\mathbf{S}), e^\sharp \neq e \text{ and } e \leq_\mathcal{J} x\}.$$

Unstab is therefore a union of unstable or irregular classes. We also define Stab as its complement (stable or irregular classes).

Lemma 9. *Let $\mathbf{S}$ be a temporal semigroup, there exists η such that*

$$\forall u \in \mathit{Stab}^+,\ \rho(u) \sim_\eta \pi(u).$$

This lemma expresses the fact that in the stable part, the stabilization semigroup is indeed a classic semigroup, and therefore its compatible function is equivalent to the product.

Theorem 33 *If f is a cost function recognized by $\mathbf{S}, h, I$ with $\mathbf{S}$ a temporal semigroup, then f is temporal.*

Proof. Let f recognized by $\mathbf{S}, h, I$ with $\mathbf{S}$ a temporal semigroup.

Let ρ be a function compatible with $\mathbf{S}$, with α_ρ as a witness.

we build a temporal B-automaton $\mathcal{A} = \langle Q, \mathbb{A}, \mathit{In}, \mathit{Fin}, \{\gamma\}, \Delta\rangle$ which will compute f. If u is the word given to the automaton, we want to find the unstable factors u which are "too long" and idempotent, in order to stabilize them as it happens in ρ. The idea is to non-deterministically guess an unstable idempotent factor of u, to which we can apply the $\sharp$ operator if it becomes too lon.

We take $Q = (\{1\} \cup \mathit{Stab}) \times (\{1\} \cup \mathit{Unstab}) \times (\{1\} \cup \mathit{Unstab})$. The first component keeps track of the current stable factor, the second one is the unstable factor we read before the idempotent factor, and the third one is the unstable idempotent factor we are currently reading. We therefore define :

$$\begin{aligned}
\Delta = {} & \{((s,a,1),l,ic,(s,a \cdot h(l),1))/a \cdot h(l) \in \mathit{Unstab}\} && (1)\\
& \cup \{((s,a,b),l,ic,(s,a,b \cdot h(l)))/a \cdot b \cdot h(l) \in \mathit{Unstab}\} && (2)\\
& \cup \{((s,a,b),l,r,(s \cdot a \cdot (b \cdot h(l))^\sharp,1,1))/a \cdot b \cdot h(l) \in \mathit{Instab}, b \cdot h(l) \text{ idempotent}\} && (3)\\
& \cup \{((s,a,b),l,r,(s \cdot a \cdot b \cdot h(l),1,1))/a \cdot b \cdot h(l) \in \mathit{Stab}\} && (4)
\end{aligned}$$

We finally choose $In = \{(1,1,1)\}$ and $Fin = \{(s,a,b), s \cdot a \cdot b \notin I\}$.

We start by showing $f \preccurlyeq_\alpha [\![\mathcal{A}]\!]_B$ for some α. Let $u \in \mathbb{A}^+$, and σ a valid run of $\mathcal{A}$ over u finishing in state (s,a,b). Let $n = \sup C(\sigma)$, $q = s \cdot a \cdot b$, and $w = h(u) \in S^+$. By definition of $\mathcal{A}$, w can be split in $x_1 y_1 z_1 \dots x_k y_k z_k$ with $q = \pi(x_1 y_1) \cdot \pi(z_1)^\sharp \cdots \pi(x_k y_k) \cdot \pi(z_k)^\sharp$, and for all $j \in [\![1,k]\!], \pi(x_j) \in \mathit{Stab} \cup \{1\}, \pi(y_j) \in \mathit{Unstab} \cup \{1\}$, and $\pi(z_j)$ unstable idempotent with $|y_j z_j| \leq n$ (we assume here without loss of generality that the last transition is of type (3)). The transitions used during lecture of the x_j's are of types (1), (2) and (4); those corresponding to y_j's are of type (1), and finally those corresponding to z_j's are of type (2) with one of type (3) at the end. Let γ be the function of Lemma 3, for all j and all $m \geq \gamma(n), \rho(y_j z_j)(m) = \pi(y_j z_j)$. Let η be the function of Lemma 9, by combining the two lemmas and using the fact that any unstable factor of x_j has length at most n, we get that for $m \geq \max(\eta(n), \gamma(n)),$, pour tout $j \in [\![1,k]\!]$, $\rho(x_j)(m) = \pi(x_j)$.

Hence we have, for all $m \geq \max(\eta(n), \gamma(n))$:

$$\begin{aligned}
\rho(w)(\alpha_\rho(m)) &\geq \rho(\rho(x_1)(m)\rho(y_1 z_1)(m) \dots \rho(x_k)(m)\rho(y_k z_k)(m))(m) \\
&\geq \rho(\rho(x_1)(m)\pi(y_1 z_1) \dots \rho(x_k)(m)\pi(y_k z_k))(m) \\
&\geq \rho(\pi(x_1)(\pi(y_1) \cdot \pi(z_1)^\sharp) \dots \pi(x_k)(\pi(y_k) \cdot \pi(z_k)^\sharp))(m) \\
&\geq q \qquad \text{(Lemma 9)}
\end{aligned}$$

In conclusion, for all run σ of value at most n over u and finishing in (s,a,b) with $q = s \cdot a \cdot b$, we get $q \leq \rho(h(u))(\alpha(n))$ with $\alpha = \alpha_\rho \circ \max(\eta, \gamma)$, but $q \notin I$ is the condition for (s,a,b) to be an accepting state, so $f(u) \leq \alpha(n)$. We can conclude $f \preccurlyeq_\alpha [\![\mathcal{A}]\!]_B$.

Conversely, let us show that $[\![\mathcal{A}]\!]_B \preccurlyeq_\beta f$ for some β. For all $u \in \mathbb{A}^+$ and $n \in \mathbb{N}$, we build a run σ of $\mathcal{A}$ over u, such that $\sup C(\sigma) \leq \beta(n)$, and such that by taking $q = s \cdot a \cdot b$ with (s,a,b)last state of σ, we have $\rho(h(u))(n)) \leq q$. In this way, if $f(u) \leq n$, then $\rho(h(u))(n) \notin I$ hence $q \notin I$ and the run is valid, which implies $[\![\mathcal{A}]\!]_B(u) \leq \beta(n)$. We therefore get the result $[\![\mathcal{A}]\!]_B \preccurlyeq_\beta f$.

We remind the Ramsey theorem : there exists α_R such that for all $t \in \mathbb{N}, w \in S^+$, there exists a decomposition of w into $x v_1 \dots v_n y$ with $\pi(v_1) = \cdots = \pi(v_t) = e$ idempotent, and $\alpha_R(t) \geq |w|$.

Let $u \in \mathbb{A}^+$ and $n \in \mathbb{N}$. Let $t = \alpha_\rho(\alpha_\rho(n+1))$. If there is in $h(u)$ a factor w with $\pi(w) \in \mathit{Instab}$ and $|w| = \alpha_R(t)$, we apply Ramsey theorem to get $w = x v_1 \dots v_t, y$ with $\pi(v_1) = \cdots = \pi(v_t) = e$ idempotent. This decomposition gives us a rune of $\mathcal{A}$ over u : on factor w, we do transitions of type (1) over x, type (2) over the v_j's, and type (3) at the end of v_n, the rest of the run is then constructed in the same way with y. This gives us a decomposition $h(u) = u_1 w_1 \dots u_k w_k u_{k+1}$, with $w_j = x_j v_1^j \dots v_t^j$ and $\pi(v_1^j) = \cdots = \pi(v_t^j) = e_j$ idempotent, for all $j \in [\![1,k]\!]$. We get a run σ verifying $\sup C(e) \leq \alpha_R(t)$, and ending in state (s,a,b) with $q = s \cdot a \cdot b = \pi(u_1 x_1) \cdot e_1^\sharp \cdots \pi(u_k x_k) \cdot e_k^\sharp \cdot \pi(u_{k+1})$. But we have :

$$\begin{aligned}
\rho(h(u))(n) &\leq \tilde{\rho}(\rho(u_1 x_1)\rho(v_1^1) \dots \rho(v_t^1) \dots \rho(u_k x_k)\rho(v_1^k) \dots \rho(v_t^k)\rho(u_{k+1}))(\alpha_\rho(n)) \\
&\leq \tilde{\rho}(\pi(u_1 x_1)\pi(v_1^1) \dots \pi(v_t^1) \dots \pi(u_k x_k)\pi(v_1^k) \dots \pi(v_t^k)\pi(u_{k+1}))(\alpha_\rho(n)) \\
&\leq \rho(\pi(u_1 x_1) e_1^t \dots \pi(u_k x_k) e_k^t \pi(u_{k+1}))(\alpha_\rho(n)) \\
&\leq (\pi(u_1 x_1) \cdot (e_1^\sharp|_t e_1) \dots \pi(u_k x_k) \cdot (e_k^\sharp|_t e_k) \cdot \pi(u_{k+1}))(\alpha_\rho(\alpha_\rho(n))) \\
&= q
\end{aligned}$$

We get the wanted result, with $\beta(n) = \alpha_R(\alpha_\rho(\alpha_\rho(n+1)))$.

Finally, $f \approx_{\max(\alpha,\beta)} [\![\mathcal{A}]\!]_B$, $\mathcal{A}$ recognizes the cost function f. □

In order to show the last item of Theorem 12 we still have to show that minimization preserves the temporal property of a stabilization semigroup :

Theorem 34 *If* $\mathbf{S}$ *is temporal, then* $\mathbf{S}/\equiv$ *is temporal.*

Proof. Let us assume that $\mathbf{S}/\equiv$ is not temporal. It means there is $x, y \in E(\mathbf{S}/\equiv)$ with $x^\sharp \neq x$, $y^\sharp = y$, and $x <_\mathcal{J} y$. Let τ be the canonical projection of $\mathbf{S}$ over $\mathbf{S}/\equiv$, there is $a, b \in \mathbf{S}$ such that $\tau(a) = x$ and $\tau(b) = y$. We have $a^\sharp \not\equiv a$ so a is unstable, $b^\sharp \equiv b$ and $a <_\mathcal{J} b^\sharp = (b^\sharp)^\sharp$, which shows that $\mathbf{S}$ is not temporal. □