Linear temporal logic for regular cost functions

Denis Kuperberg

LIAFA/CNRS/Université Paris 7, Denis Diderot, 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 define an extension of Linear Temporal Logic (LTL) over finite words to describe cost functions. We give an explicit translation from this new logic to automata. We then algebraically characterize the expressive power of this logic, using a new syntactic congruence for cost functions introduced in this paper.

1 Introduction

Since the seminal works of Kleene and Rabin and Scott, 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, while retaining most of the fundamental properties of the original theory such as the closure properties, the various equivalent characterizations, and the decidability [Col09]. A cost function is an equivalence class of the functions from the domain (words in our case) to $\mathbb{N} \cup \{\infty\}$, 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...

Linear Temporal Logic (LTL), which is a natural way to describe logical constraints over a linear structure, have also been a fertile subject of study, particularly in the context of regular languages and automata [VW86]. Moreover quantitative extensions of LTL have recently been successfully introduced. For instance the model Prompt-LTL introduced in [KPV09] is interested in bounding the waiting time of all requests of a formula, and in this sense is quite close to the aim of cost functions.

In this paper, we extend LTL (over finite words) into a new logic with quantitative features (LTL$^{\leq}$), in order to describe cost functions over finite words with logical formulae. We do this by adding a new operator $U^{\leq N}$: a formula $\phi U^{\leq N}\psi$ means that ψ holds somewhere in the future, and ϕ has to hold until that point, except at most N times (we allow at most N "mistakes" of the until formula).

Related works and motivating examples

Regular cost functions are the continuation of a sequence of works that intend to solve difficult questions in language theory. Among several other decision problems, the most 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. The problem was resolved by Hashigushi [Has88] using a very intricate proof, and later by Kirsten [Kir05] using an automaton that has counting features.

Finally, also using ideas inspired from [BC06], the theory of those automata over words has been unified in [Col09], 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.

On the logic side, Prompt-LTL, introduced in [KPV09], is an interesting way to extend LTL in order to look at boundedness issues, and already gave interesting decidability and complexity results. Prompt-LTL would correspond in the framework of regular cost functions to a subclass of temporal cost functions introduced in [CKL10]; in particular it is weaker than LTL$^{\leq}$ introduced here.

Contributions

It is known from [Col09] that regular cost functions are the ones recognizable by stabilization semigroups (or in an equivalent way, stabilization monoids), and from [CKL10] than there is an effective quotient-wise minimal stabilization semigroup for each regular cost function. This model of semigroups extends the standard approach for languages.

We introduce a quantitative version of LTL in order to describe cost functions by means of logical formulas. The idea of this new logic is to bound the number of "mistakes" of Until operators, by adding a new operator $U^{\leq N}$. The first contribution of this paper is to give a direct translation from LTL$^{\leq}$-formulas to B-automata, which is an extension of the classic translation from LTL to Büchi automaton for languages. This translation preserves exact values (i.e. not only cost functions equivalence), which could be interesting in terms of future applications. We then show that regular cost functions described by LTL formulae are the same as the ones computed by aperiodic stabilization semigroups, and this characterization is effective. The proof uses a syntactic congruence for cost functions, introduced in this paper.

This work validates the algebraic approach for studying cost functions, since the analogy extends to syntactic congruence. It also allows a more user-friendly way to describe cost functions, since LTL can be more intuitive than automata or stabilization semigroups to describe a given cost function.

As it was done in [CKL10] for temporal cost functions, the characterization result obtained here for LTL$^{\leq}$-definable cost functions follows the spirit of Schützenberger's theorem which links star-free languages with aperiodic monoids [Sch65].

Organisation of the paper

After some notations, and reminder on cost functions, we introduce in Section 3 LTL$^{\leq}$ as a quantitative extension of LTL, and give an explicit translation from LTL$^{\leq}$-formulae to B-automata. We then present in Section 4 a syntactic congruence for cost functions, and show that it indeed computes the minimal stabilization semigroup of any regular cost function. We finally use this new tool to show that LTL$^{\leq}$ has the same expressive power as aperiodic stabilization semigroups.

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 $\mathbb{A}^*$ and the empty word will be noted ϵ. 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 $|u|_a$. Functions $\mathbb{N} \to \mathbb{N}$ will be denoted by letters $\alpha, \beta, \ldots$, and will be extended to $\mathbb{N} \cup \{\infty\}$ by $\alpha(\infty) = \infty$.

2 Regular Cost functions

2.1 Cost functions and equivalence

If $L \subseteq \mathbb{A}^*$, we will note χ_L the function defined by $\chi_L(u) = 0$ if $u \in L$, ∞ if $u \notin L$. Let $\mathcal{F}$ be the set of functions : $\mathbb{A}^* \to \mathbb{N}_\infty$. For $f, g \in \mathcal{F}$ and α a function (see Notations), we say that $f \leq_\alpha g$ if $f \leq \alpha \circ g$, and $f \approx_\alpha g$ if $f \leq_\alpha g$ and $g \leq_\alpha f$. Finally $f \approx g$ if $f \approx_\alpha g$ for some α. This equivalence relation doesn't pay attention to exact values, but preserves the existence of bounds.

A *cost function* is an equivalence class of $\mathcal{F}/\approx$. Cost functions are noted $f, g, \ldots$, and in practice they will be always be represented by one of their elements in $\mathcal{F}$.

2.2 *B*-automata

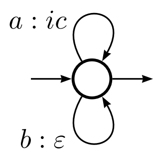
A B-automaton is a tuple $\langle Q, \mathbb{A}, In, Fin, \Gamma, \Delta \rangle$ where Q is the set of states, $\mathbb{A}$ the alphabet, In and Fin the sets of initial and final states, Γ the set of counters, and $\Delta \subseteq Q \times \mathbb{A} \times (\{i,r,c\}^*)^\Gamma \times Q$ is the set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

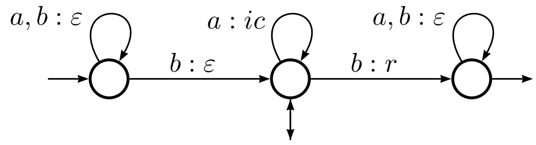
Counters have integers values starting at 0, and an action $\sigma \in (\{i,r,c\}^*)^\Gamma$ performs a sequence of atomic actions on each counter, where atomic actions are either i (increment by 1), r (reset to 0) or c (check the value). In particular we will note ε the action corresponding to the empty word : doing nothing on every counter. If e is a run, let $C(e)$ be the set of values checked during e on all counters of Γ.

A B-automaton $\mathcal{A}$ computes a regular cost function $[\![\mathcal{A}]\!]$ via the following semantic : $[\![\mathcal{A}]\!](u) = \inf\{\sup C(e), e \text{ run of } \mathcal{A} \text{ over } u\}$.

With the usual conventions that $\sup \emptyset = 0$ and $\inf \emptyset = \infty$. There exists also a dual model of B-automata, namely S-automata, that has the same expressive power, but we won't develop this further in this paper. See [Col09] for more details.

Example 1 *Let $\mathbb{A} = \{a, b\}$. The cost function $|\cdot|_a$ is the same as $2|\cdot|_a + 5$, it is computed by the following one-counter B-automaton on the left-hand side. The cost function $u \mapsto \min\{n \in \mathbb{N},\ a^n \text{ factor of } u\}$ is computed by the nondeterministic one-counter B-automaton on the right-hand side.*





Moreover, as in the case of languages, cost functions can be recognized by an algebraic structure that extends the classic notion of semigroups, called stabilization semigroups. A stabilization semigroup $\mathbf{S} = \langle S, \cdot, \leq, \sharp \rangle$ is a partially ordered set S together with an internal binary operation $\cdot$ and an internal unary operation $a \mapsto a^\sharp$ defined only on idempotent elements (elements a such that $a \cdot a = a$). The formalism is quite heavy, see appendix for all details on axioms of stabilization semigroups and recognition of regular cost functions.

3 Quantitative LTL

We will now use an extension of LTL to describe some regular cost functions. This has been done successfully with regular languages, so we aim to obtain the same kind of results. Can we still go efficiently from an LTL-formula to an automaton?

3.1 Definition

The first thing to do is to extend LTL so that it can decribe cost functions instead of languages. We must add quantitative features, and this will be done by a new operator $U^{\leq N}$. Unlike in most uses of LTL, we work here over finite words.

Formulas of LTL$^{\leq}$ (on finite words on an alphabet $\mathbb{A}$) are defined by the following grammar :

$$\phi := a \mid \phi \wedge \phi \mid \phi \vee \phi \mid X\phi \mid \phi U \phi \mid \ \phi U^{\leq N} \phi \mid \Omega$$

Note the absence of negation in the definition of LTL$^{\leq}$. The negations have been pushed to the leaves.

- a means that the current letter is a, $\wedge$ and $\vee$ are the classic conjunction and disjunction;
- $X\phi$ means that ϕ is true at the next letter;
- $\phi U \psi$ means that ψ is true somewhere in the future, and ϕ holds until that point;
- $\phi U^{\leq N} \psi$ means that ψ is true somewhere in the future, and ϕ can be false at most N times before ψ. The variable N is unique, and is shared by all occurrences of $U^{\leq N}$ operator;
- Ω means that we are at the end of the word.

We can define $\top = (\bigvee_{a \in \mathbb{A}} a) \vee \Omega$ and $\bot = \neg\top$, meaning respectively true and false, and $\neg a = (\bigvee_{b \neq a} b) \vee \Omega$ to signify that the current letter is not a.

We also define connectors "eventually" : $F\varphi = \top U \varphi$ and "globally" : $G\varphi = \varphi U \Omega$.

3.2 Semantics

We want to associate a cost function $[\![\phi]\!]$ on words to any LTL$^{\leq}$-formula ϕ.

We will say that $u, n \models \phi$ (u, n is a model of ϕ) if ϕ is true on u with n as valuation for N, i.e. as number of errors for all the $U^{\leq N}$'s in the formula ϕ. We finally define

$$[\![\phi]\!](u) = \inf \{n \in \mathbb{N} / u, n \models \phi\}$$

We can remark that if $u, n \models \phi$, then for all $k \geq n, u, k \models \phi$, since the $U^{\leq N}$ operators appear always positively in the formula (that is why we don't allow the negation of an LTL$^{\leq}$-formula in general). In particular, $[\![\phi]\!](u) = 0$ means that $\forall n \in \mathbb{N}, u, n \models \phi$, and $[\![\phi]\!](u) = \infty$ means that $\forall n \in \mathbb{N}, u, n \not\models \phi$ (since $\inf \emptyset = \infty$).

Proposition 2

- $[\![a]\!](u) = 0$ *if* $u \in a\mathbb{A}^*$, *and* ∞ *otherwise*
- $[\![\Omega]\!](u) = 0$ *if* $u = \varepsilon$, *and* ∞ *otherwise*
- $[\![\phi \wedge \psi]\!] = \max([\![\phi]\!], [\![\psi]\!])$, *and* $[\![\phi \vee \psi]\!] = \min([\![\phi]\!], [\![\psi]\!])$

– $[\![X\phi]\!](au) = [\![\phi]\!](u)$, $[\![X\phi]\!](\varepsilon) = \infty$
– $[\![\top]\!] = 0$, *and* $[\![\bot]\!] = \infty$

Example 3 *Let* $\phi = (\neg a)U^{\leq N}\Omega$, *then* $[\![\phi]\!] = |\cdot|_a$

We use LTL$^{\leq}$-formulae in order to describe cost functions, so we will always work modulo cost function equivalence $\approx$.

Remark 4 *If* ϕ *does not contain any operator* $U^{\leq N}$, ϕ *is a classic LTL-formula computing a language* L, *and* $[\![\phi]\!] = \chi_L$.

3.3 From LTL$^{\leq}$ to B-Automata

We will now give a direct translation from LTL$^{\leq}$-formula to B-automata, i.e. given an LTL$^{\leq}$-formula ϕ on a finite alphabet $\mathbb{A}$, we want to build a B-automaton recognizing $[\![\phi]\!]$. This construction is adapted from the classic translation from LTL-formula to Büchi automata [DG10].

Let ϕ be an LTL$^{\leq}$-formula. We define $\mathrm{sub}(\phi)$ to be the set of subformulae of ϕ, and $Q = 2^{\mathrm{sub}(\phi)}$ to be the set of subsets of $\mathrm{sub}(\phi)$.

We want to define a B-automaton $\mathcal{A}_\phi = \langle Q, \mathbb{A}, In, Fin, \Gamma, \Delta\rangle$ such that $[\![\mathcal{A}]\!]_B \approx [\![\phi]\!]$.

We set the initial states to be $In = \{\{\phi\}\}$ and the final ones to be $Fin = \{\emptyset, \{\Omega\}\}$ We choose as set of counters $\Gamma = \{\gamma_1, \dots, \gamma_k\}$ where k is the number of occurences of the $U^{\leq N}$ operators in ϕ, labeled from $U_1^{\leq N}$ to $U_k^{\leq N}$.

A state is basically the set of constraints we have to verify before the end of the word, so the only two accepting states are the one with no constraint, or with only constraint to be at the end of the word.

The following definitions are the same as for the classical case (LTL to Büchi automata) :

Definition 5 – *An atomic formula is either a letter* $a \in \mathbb{A}$ *or* Ω
– *A set* Z *of formulae is consistent if there is at most one atomic formula in it.*
– *A reduced formula is either an atomic formula or a Next formula (of the form* $X\varphi$*).*
– *A set* Z *is reduced if all its elements are reduced formulae.*
– *If* Z *is consistent and reduced, we define* $\mathrm{next}(Z) = \{\varphi / X\varphi \in Z\}$.

Lemma 6 (Next Step) *If* Z *is consistent and reduced, for all* $u \in \mathbb{A}^*, a \in \mathbb{A}$ *and* $n \in \mathbb{N}$,

$$au, n \models \bigwedge Z \text{ iff } u, n \models \bigwedge \mathrm{next}(Z) \text{ and } Z \cup \{a\} \text{ consistent}$$

We would like to define $\mathcal{A}_\phi$ with $Z \longrightarrow \text{next}(Z)$ as transitions.

The problem is that $\text{next}(Z)$ is not consistent and reduced in general. If $\text{next}(Z)$ is inconsistent we remove it from the automaton. If it is consistent, we need to apply some reduction rules to get a reduced set of formulae. This consists in adding ε-transitions (but with possible actions on the counter) towards intermediate sets which are not actual states of the automaton (we will call them "pseudo-states"), until we reach a reduced set.

Let ψ be maximal (in size) not reduced in Y, we add the following transitions

- If $\psi = \varphi_1 \wedge \varphi_2 : Y \xrightarrow{\varepsilon:\varepsilon} Y \setminus \{\psi\} \cup \{\varphi_1, \varphi_2\}$
- If $\psi = \varphi_1 \vee \varphi_2 : \begin{cases} Y \xrightarrow{\varepsilon:\varepsilon} Y \setminus \{\psi\} \cup \{\varphi_1\} \\ Y \xrightarrow{\varepsilon:\varepsilon} Y \setminus \{\psi\} \cup \{\varphi_2\} \end{cases}$
- If $\psi = \varphi_1 U \varphi_2 : \begin{cases} Y \xrightarrow{\varepsilon:\varepsilon} Y \setminus \{\psi\} \cup \{\varphi_1, X\psi\} \\ Y \xrightarrow{\varepsilon:\varepsilon} Y \setminus \{\psi\} \cup \{\varphi_2\} \end{cases}$
- If $\psi = \varphi_1 U_j^{\leq N} \varphi_2 : \begin{cases} Y \xrightarrow{\varepsilon:\varepsilon} Y \setminus \{\psi\} \cup \{\varphi_1, X\psi\} \\ Y \xrightarrow{\varepsilon:ic_j} Y \setminus \{\psi\} \cup \{X\psi\} \text{ (we count one mistake)} \\ Y \xrightarrow{\varepsilon:r_j} Y \setminus \{\psi\} \cup \{\varphi_2\} \end{cases}$

  where action r_j (resp. ic_j) perform r (resp. ic) on counter γ_j and ε on the other counters.

The pseudo-states don't (a priori) belong to $Q = 2^{\text{sub}(\phi)}$ because we add formulae $X\psi$ for $\psi \in \text{sub}(\phi)$, so if Z is a reduced pseudo-state, $\text{next}(Z)$ will be in Q again since we remove the new next operators.

The transitions of automaton $\mathcal{A}_\phi$ will be defined as follows:

$$\Delta = \left\{ Y \xrightarrow{a:\sigma} \text{next}(Z) \mid Y \in Q, Z \cup \{a\} \text{ consistent and reduced}, Y \xrightarrow{\varepsilon:\sigma}_* Z \right\}$$

where $Y \xrightarrow{\varepsilon:\sigma}_* Z$ means that there is a sequence of ε-transitions from Y to Z with σ as combined action on counters.

Definition 7 *If σ is a sequence of actions on counters, we will call* $\text{val}(\sigma)$ *the maximal value checked on a counter during σ with 0 as starting value of the counters, and* $\text{val}(\sigma) = 0$ *if there is no check in σ. It corresponds to the value of a run of a B-automaton with σ as combined action of the counter.*

Lemma 8 *Let $u = a_1 \dots a_m$ be a word on $\mathbb{A}$ and $Y_0 \xrightarrow{a_1:\sigma_1} Y_1 \xrightarrow{a_2:\sigma_2} \dots \xrightarrow{a_m:\sigma_m} Y_m$ an accepting run of $\mathcal{A}_\phi$.*

Then for all $\psi \in \text{sub}(\phi)$, for all $n \in \{0, \dots, m\}$, for all $Y_n \xrightarrow{\varepsilon:\sigma}_* Y \xrightarrow{\varepsilon:\sigma'}_* Z$ with $Z \cup \{a_{n+1}\}$ consistent and reduced, and $Y_{n+1} = \text{next}(Z)$

$$\psi \in Y \implies a_{n+1}a_{n+2} \dots a_m, N \models \psi$$

where $N = \text{val}(\sigma'\sigma_{n+1} \dots \sigma_m)$.

Lemma 8 implies the correctness of the automaton $\mathcal{A}_\phi$:
Let $Y_0 \overset{a_1:\sigma_1}{\to} Y_1 \overset{a_2:\sigma_2}{\to} \dots \overset{a_m:\sigma_m}{\to} Y_m$ be a valid run of $\mathcal{A}_\phi$ on u of value $N = [\![\mathcal{A}_\phi]\!]_B$, applying Lemma 8 with $n = 0$ and $Y = Y_0 = \{\phi\}$ gives us $u, N \models \phi$. Hence $[\![\phi]\!] \leq [\![\mathcal{A}_\phi]\!]_B$.

Conversely, let $N = [\![\phi]\!](u)$, then $u, N \models \phi$ so by definition of $\mathcal{A}_\phi$, it is straightforward to verify that there exists an accepting run of $\mathcal{A}_\phi$ over u of value $\leq N$ (each counter γ_i doing at most N mistakes relative to operator $U_i^{\leq N}$). Hence $[\![\mathcal{A}_\phi]\!]_B \leq [\![\phi]\!]$.

We finally get $[\![\mathcal{A}_\phi]\!]_B = [\![\phi]\!]$, the automaton $\mathcal{A}_\phi$ computes indeed the exact value of function $[\![\phi]\!]$ (and so we have obviously $[\![\mathcal{A}_\phi]\!]_B \approx [\![\phi]\!]$).

4 Algebraic characterization

We remind that as in the case of languages, stabilization semigroups recognize exactly regular cost functions, and there exists a quotient-wise minimal stabilization semigroup for each regular cost function [CKL10].

In standard theory, it is equivalent for a regular language to be described by an LTL-formula, or to be recognized by an aperiodic semigroup. Is it still the case in the framework of regular cost functions? To answer this question we first need to develop a little further the algebraic theory of regular cost functions.

4.1 Syntactic congruence

In standard theory of languages, we can go from a description of a regular language L to a description of its syntactic monoid via the syntactic congruence. Moreover, when the language is not regular, we get an infinite monoid, so this equivalence can be used to "test" regularity of a language.

The main idea behind this equivalence is to identify words u and v if they "behave the same" relatively to the language L, i.e. L cannot separate u from v in any context : $\forall (x,y), xuy \in L \Leftrightarrow xvy \in L$.

The aim here is to define an analog to the syntactic congruence, but for regular cost functions instead of regular languages. Since cost functions look at quantitative aspects of words, the notions of "element" and "context" have to contain quantitative information : we want to be able to say things like "words with a lot of a's behave the same as words with a few a's".

That is why we won't define our equivalence over words, but over $\sharp$-expressions, which are a way to describe words with quantitative information.

4.2 $\sharp$-expressions

We first define general $\sharp$-expressions as in [Has90] and [CKL10] by just adding an operator $\sharp$ to words in order to repeat a subexpression "a lot of times". This differs from the stabilization monoid definition, in which the $\sharp$-operator can only be applied to specific elements (idempotents).

The set Expr of $\sharp$-expressions on an alphabet $\mathbb{A}$ is defined as follows:

$$e := a \in \mathbb{A} \mid ee \mid e^{\sharp}$$

If we choose a stabilization semigroup $\mathbf{S} = \langle S, \cdot, \leq, \sharp \rangle$ together with a function $h : \mathbb{A} \to S$, the eval function (from Expr to $\mathbf{S}$) is defined inductively by $\mathrm{eval}(a) = h(a), \mathrm{eval}(ee') = \mathrm{eval}(e) \cdot \mathrm{eval}(e')$, and $\mathrm{eval}(e^{\sharp}) = \mathrm{eval}(e)^{\sharp}$ ($\mathrm{eval}(e)$ has to be idempotent). We say that e is *well-formed for* $\mathbf{S}$ if $\mathrm{eval}(e)$ exists. Intuitively, it means that $\sharp$ was applied to subexpressions that corresponds to idempotent elements in $\mathbf{S}$.

If f is a regular cost function, e is *well-formed for f* iff e is well-formed for the minimal stabilization semigroup of f.

Example 9 *Let f be the cost function defined over $\{a\}^*$ by*

$$f(a^n) = \begin{cases} n & \text{if } n \text{ even} \\ \infty & \text{otherwise} \end{cases}$$

The minimal stabilization semigroup of f is : $\{a, aa, (aa)^{\sharp}, (aa)^{\sharp}a\}$, with $aa \cdot a = a$ and $(aa)^{\sharp}a \cdot a = (aa)^{\sharp}$. Hence the $\sharp$-expression $aaa(aa)^{\sharp}$ is well-formed for f but the $\sharp$-expression $a^{\sharp}$ is not.

The $\sharp$-expressions that are not well-formed have to be removed from the set we want to quotient, in order to get only real elements of the syntactic semigroup.

4.3 $\omega\sharp$-expressions

We have defined the set of $\sharp$-expressions that we want to quotient to get the syntactic equivalence of cost functions. However, we saw that some of these $\sharp$-expressions may not be well-typed for the cost function f we want to study, and therefore does not correspond to an element in the syntactic stabilization semigroup of f.

Thus we need to be careful about the stabilization operator, and apply it only to "idempotent $\sharp$-expressions". To reach this goal, we will add an "idempotent operator" ω on $\sharp$-expressions, which will always associate an idempotent element (relative to f) to a $\sharp$-expression, so that we can later apply $\sharp$ and be sure of creating well-formed expressions for f.

We define the set Oexpr of $\omega\sharp$-expressions on an alphabet $\mathbb{A}$:

$$E := a \in \mathbb{A} \mid EE \mid E^{\omega} \mid E^{\omega\sharp}$$

The intuition behind operator ω is that x^{ω} is the idempotent obtained by iterating x (which always exists in finite semigroups).

A *context* $C[x]$ is a $\omega\sharp$-expression with possible occurrences of a free variable x. Let E be a $\omega\sharp$-expression, $C[E]$ is the $\omega\sharp$-expression obtained by replacing all occurrences of x by E in $C[x]$, i.e. $C[E] = C[x][x \leftarrow E]$. Let $\mathrm{C_{OE}}$ be the set of contexts on $\omega\sharp$-expressions.

We will now formally define the semantic of operator ω, and use $\omega\sharp$-expressions to get a syntactic equivalence on cost functions, without mistyped $\sharp$-expressions.

Definition 10 *If $E \in$ Oexpr and $k, n \in \mathbb{N}$, we define $E(k,n)$ to be the word $E[\omega \leftarrow k, \sharp \leftarrow n]$, where the exponential is relative to concatenation of words.*

Lemma 11 *Let f be a regular cost function, there exists $K_f \in \mathbb{N}$ such that for any $E \in$ Oexpr, the $\sharp$-expression $E[\omega \leftarrow K_f!]$ is well-formed for f, and we are in one of these two cases*

1. *$\forall k \geq K_f, \{f(E(k!, n)), n \in \mathbb{N}\}$ is bounded : we say that $E \in f^B$.*
2. *$\forall k \geq K_f, \lim_{n \to \infty} f(E(k!, n)) = \infty$: we say that $E \in f^\infty$.*

Proof. The proof is a little technical, since we have to reuse the definition of recognization by stabilization semigroup. K_f can simply be taken to be the size of the minimal stabilization semigroup of f.

Here, f^B and f^∞ are the analogs for regular cost functions of "being in L" and "not being in L" in language theory. But this notion is now asymptotic, since we look at boundedness properties of quantitative information on words. Moreover, f^∞ and f^B are only defined here for regular cost functions, since K_f might not exist if f is not regular.

Definition 12 *Let f be a regular cost function, we write $E \rightleftharpoons_f E'$ if $(E \in f^B \Leftrightarrow E' \in f^B)$. Finally we define*

$$E \equiv_f E' \textit{ iff } \forall C[x] \in \mathrm{C_{OE}}, C[E] \rightleftharpoons_f C[E']$$

Remark 13 *If $u, v \in \mathbb{A}^*$, and L is a regular language, then $u \sim_L v$ iff $u \equiv_{\chi_L} v$ ($\sim_L$ being the syntactic congruence of L). In this sense, $\equiv$ is an extension of the classic syntactic congruence on languages.*

Now that we have properly defined the equivalence $\equiv_f$ over Oexpr, it remains to verify that it is indeed a good syntactic congruence, i.e. Oexpr$/\equiv_f$ is the syntactic stabilization semigroup of f.

Indeed if f is a regular cost function, let $\mathbf{S}_f =$ Oexpr$/\equiv_f$. We can provide $\mathbf{S}_f$ with a structure of stabilization semigroup $\langle \mathbf{S}_f, \cdot, \leq, \sharp \rangle$.

Theorem 14. *$\mathbf{S}_f$ is the minimal stabilization semigroup recognizing f.*

The proof consists basically in a bijection between classes of Oexpr for $\equiv_f$, and elements of the minimal stabilization semigroup as defined in appendix A.7 of [CKL10].

4.4 Expressive power of LTL$^\leq$

If f is a regular cost function, we will call $\mathbf{S}_f$ the syntactic stabilization semigroup of f.

A finite semigroup $\mathbf{S} = \langle S, \cdot \rangle$ is called *aperiodic* if $\exists k \in \mathbb{N}, \forall s \in \mathbf{S}, s^{k+1} = s^k$. The definition is the same if $\mathbf{S}$ is a finite stabilization semigroup.

Remark 15 *For a regular cost function f, the statements "f is recognized by an aperiodic stabilization semigroup" and "$\mathbf{S}_f$ is aperiodic" are equivalent, since $\mathbf{S}_f$ is a quotient of all stabilization semigroups recognizing f.*

Theorem 16. *Let f be a cost function described by a LTL$^{\leq}$-formula, then f is regular and the syntactic stabilization semigroup of f is aperiodic.*

The proof of this theorem will be the first framework to use the syntactic congruence on cost functions.

If ϕ is a LTL$^{\leq}$-formula, we will say that ϕ verifies property AP if there exists $k \in \mathbb{N}$ such that for any $\omega\sharp$-expression E, $E^k \equiv_{[\![\phi]\!]} E^{k+1}$, which is equivalent to "$[\![\phi]\!]$ has an aperiodic syntactic stabilization semigroup".

With this in mind, we can do an induction on LTL$^{\leq}$-formulaes : we first show that $\mathbf{S}_{\Omega}$ and all $\mathbf{S}_a$ for $a \in \mathbb{A}$ are aperiodic.

We then proceed to the induction on ϕ : assuming that φ and ψ verify property AP, we show that $X\psi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi U \psi$ and $\varphi U^{\leq N} \psi$ verify property AP.

Theorem 17. *Let f be a cost function recognized by an aperiodic stabilization semigroup, then f can be described by an LTL$^{\leq}$-formula.*

The proof of this theorem is a generalization of the proof of Wilke for aperiodic languages in [Wil99]. However difficulties inherent to quantitative notions appear here.

The main issue comes from the fact that in the classical setting, computing the value of a word in a monoid returns a single element. This fact is used to do an induction on the size of the monoid, by considering the set of possible results as a smaller monoid. The problem is that with cost functions, there is some additional quantitative information, and we need to associate a sequence of elements of a stabilization monoid to a single word. Therefore, it requires some technical work to come back to a smaller stabilization monoid from these sequences.

Corollary 18 *The class of LTL$^{\leq}$-definable cost functions is decidable.*

Proof. Theorems 16 and 17 imply that it is equivalent for a regular cost function to be LTL$^{\leq}$-definable or to have an aperiodic syntactic stabilization semigroup. If f is given by an automaton or a stabilization semigroup, we can compute its syntactic stabilization semigroup $\mathbf{S}_f$ (see [CKL10]) and decide if f is LTL$^{\leq}$-definable by testing aperiodicity of $\mathbf{S}_f$. This can be done simply by iterating at most $|\mathbf{S}_f|$ times all elements of $\mathbf{S}_f$ and see if each element a reaches an element a^k such that $a^{k+1} = a^k$.

5 Conclusion

We first defined LTL$^{\leq}$ as a quantitative extension of LTL. We started the study of LTL$^{\leq}$ by giving an explicit translation from LTL$^{\leq}$-formulae to B-automata, which preserves exact values (and not only boundedness properties as it is usually the case in the framework of cost functions). We then showed that the expressive power of $\mathrm{LTL}^{\leq}$ in terms of cost functions is the same as aperiodic stabilization semigroups. The proof uses a new syntactic congruence, which has a general interest in the study of regular cost functions. This result implies the decidability of the $\mathrm{LTL}^{\leq}$-definable class of cost functions.

As a further work, we can try to put $\omega\sharp$-expressions in a larger framework, by doing an axiomatization of $\omega\sharp$-semigroups. We can also extend this work to infinite words, and define an analog to Büchi automata for cost functions. To continue the analogy with classic languages results, we can define a quantitative extension of FO describing the same class as $\mathrm{LTL}^{\leq}$, and search for analog definitions of counter-free B-automata and star-free B-regular expressions. The translation from $\mathrm{LTL}^{\leq}$-formulae to B-automata can be further studied in terms of optimality of number of counters of the resulting B-automaton.

Acknowledgments

I am very grateful to my advisor Thomas Colcombet for our helpful discussions, and for the guidelines he gave me on this work, and to Michael Vanden Boom for helping me with language and presentation issues.

References

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

[CKL10] Thomas Colcombet, Denis Kuperberg, and Sylvain Lombardy. Regular temporal cost functions. In *ICALP (2)*, pages 563–574, 2010.

[Col09] Thomas Colcombet. The theory of stabilization monoids and regular cost functions. *ICALP, Lecture Notes in Computer Science*, 2009.

[DG10] Stéphane Demri and Paul Gastin. Specification and verification using temporal logics. In *Modern applications of automata theory*, volume 2 of *IISc Research Monographs*. World Scientific, 2010. To appear.

[Has88] Kosaburo 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] Kosaburo Hashiguchi. Improved limitedness theorems on finite automata with distance functions. *Theor. Comput. Sci.*, 72(1):27–38, 1990.

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

[KPV09] Orna Kupferman, Nir Piterman, and Moshe Y. Vardi. From liveness to promptness. *Formal Methods in System Design*, 34(2):83–103, 2009.

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

[VW86] Moshe Y. Vardi and Pierre Wolper. Automata-theoretic techniques for modal logics of programs. *J. Comput. Syst. Sci.*, 32(2):183–221, 1986.

[Wil99] Thomas Wilke. Classifying discrete temporal properties. In Christoph Meinel and Sophie Tison, editors, *STACS*, volume 1563 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 1999.

6 Appendix

We will start by reviewing all notions needed to work with stabilization semigroups.

6.1 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 infinite 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 to be *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 transitive, but the following property guarantees a certain kind of transitivity.

Fact 19 *$\boldsymbol{a} \preceq_\alpha \boldsymbol{b} \preceq_\alpha \boldsymbol{c}$ 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 **??** shows that a transitivity step costs some precision. For any α, the relation $\preceq_\alpha$ coincides over constant sequences with order $\leq$ (up to identification of constant sequences with their constant value). Consequently, the infinite sequences 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.

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

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

In particular, if f is α-monotone, 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 $f : E \rightarrow F^{\mathbb{N}}$ we associate $\tilde{f} : E^{\mathbb{N}} \rightarrow F^{\mathbb{N}}$ 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 20 *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 21 *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 notion 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 $i = 1 \dots n$.

6.2 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 $I[\boldsymbol{a}] = \sup\{n + 1 \ : \ \boldsymbol{a}(n) \in I\}$.[1]

Proposition 22 *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 23 *Let $a, b \in E$ and $n \in \mathbb{N}$, we define the sequence $a|nb$ by:*

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

6.3 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

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

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 24 *Let $\mathbf{S} = \langle S, \cdot, \leq, \sharp \rangle$ be stabilization semigroup. A function ρ from S^+ to $S^{\mathbb{N}}$ is* 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 $e \in E(\mathbf{S})$, $m \in \mathbb{N}$, $\rho(e^m) \sim_\alpha (e^\sharp | me)$,*
Substitution. *for all $u_1, \ldots, u_n \in S^+$, $n \in \mathbb{N}$, $\rho(u_1 \ldots u_n) \sim_\alpha \tilde{\rho}(\rho(u_1) \ldots \rho(u_n))$ (we identify sequence of words and word of sequences)*

Example 1. Let $\mathbf{S}$ be the stabilization semigroup with 3 elements $\perp \leq a \leq b$, with product defined by : $x \cdot y = \min_\leq(x, y)$ (b neutral element), and stabilization by $b^\sharp = b$ and $a^\sharp = \perp^\sharp = \perp$. Lett $u \in \{\perp, a, b\}^+$, we define ρ by:

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

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

Remark 25 *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 26 ([Col09]). *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 1. *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 $\gamma(n) = \alpha(\gamma(\alpha(n-1)))$. ∎

6.4 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 recognizable if it is $\approx$-equivalent to a function recognized by some $\mathbf{S}, h, I$. By Proposition **??**, the recognized cost function does not depend on the choice of ρ.

Example 2. Let $\mathbb{A} = \{a, b\}$, the cost function $|\cdot|_a$ is recognizable. We take the stabilization semigroup from Example **??**, h defined by $h(a) = a, h(b) = b$, and $I = \{\bot\}$. We have then $|u|_a = I[\rho(h(u))]$ for all $u \in \mathbb{A}^+$.

6.5 Proof of Lemma 8

Proof. We do a reverse induction on n. If $n = m$, Y_n is a final state so $Y_n = \emptyset$ or $Y_n = \{\Omega\}$. If $Y_n \overset{\varepsilon:\sigma}{\to}_* Y$, then $Y = Y_n$ (no outgoing ε-transitions defined from $\emptyset$ or $\{\Omega\}$). Then if $\psi \in Y$, the only possibility is $\psi = \Omega$, but $a_{n+1} \dots a_m = \varepsilon$, and $\varepsilon, 0 \models \Omega$, hence the result is true for $n = m$.

Let $n < m$, we assume the result is true for $n+1$, and we take same notations as in the lemma, with $\psi \in Y$. By definition of Δ, there exists a transition $Y_n \overset{a_{n+1}:\sigma\sigma'}{\longrightarrow}_* \mathrm{next}(Z) = Y_{n+1}$ in $\mathcal{A}_\phi$.

We do an induction on the length k of the path $Y \overset{\varepsilon:\sigma'}{\to}_* Z$.

If $k = 0$, then $Y = Z$ consistent and reduced, so ψ is either atomic or a Next formula.

If ψ is atomic, the only way $Z \cup \{a_{n+1}\}$ can be consistent is if $\psi = a_{n+1}$. In which case $a_{n+1} \dots a_m, N \models \psi$ without difficulty.

If $\psi = X\varphi$, $\varphi \in \mathrm{next}(Z) = Y_{n+1}$, then by induction hypothesis (on n), $a_{n+2} \dots a_m, N \models \varphi$ (N does not change because σ' is empty). Hence $a_{n+1}a_{n+2} \dots a_m, N \models X\varphi$ which shows the result.

If $k > 0$, we assume the result is true for $k-1$, and we show it for k. We have $Y \overset{\varepsilon:\sigma'_1}{\to} Y' \overset{\varepsilon:\sigma'_2}{\to}_* Z$ with $\sigma'_1\sigma'_2 = \sigma'$, and for all $\psi' \in Y', a_{n+1}a_{n+2} \dots a_m, N' \models \psi'$ with $N' = \mathrm{val}(\sigma'_2\sigma_{n+1} \dots \sigma_m)$.

We now look at the different possibility for the ε-transition $Y \overset{\varepsilon:\sigma'_1}{\to} Y'$. Let us first notice that either $N = N'$ or $N = N' + 1$, since $\sigma'_1 \in \{\varepsilon, ic, r\}$.

Let $u_{n+1} = a_{n+1}a_{n+2} \dots a_m$. If $\psi \in Y'$, then $u_{n+1}, N' \models \psi$, but $N \geq N'$ so $u_{n+1}, N' \models \psi$.

We just need to examine the cases where $\psi \notin Y'$:

- If $\psi = \varphi_1 \wedge \varphi_2$, $\sigma'_1 = \varepsilon$, and $Y' = Y \setminus \{\psi\} \cup \{\varphi_1, \varphi_2\}$, then $u_{n+1}, N \models \varphi_1$ and $u_{n+1} \dots a_m, N \models \varphi_2$, hence $u_{n+1}, N \models \psi$.
- Other classic cases where $\sigma'_1 = \varepsilon$ are similar and come directly from definition of LTL operators.

– If $\psi = \varphi_1 U_j^{\leq N} \varphi_2$, $\sigma'_1 = \varepsilon$ and $Y' = Y \setminus \{\psi\} \cup \{\varphi_1, X\psi\}$, then $u_{n+1}, N \models \varphi_1$ and $u_{n+1}, N \models X\psi$, hence $u_{n+1}, N \models \psi$
– If $\psi = \varphi_1 U_j^{\leq N} \varphi_2$, $\sigma'_1 = ic_j$ and $Y' = Y \setminus \{\psi\} \cup \{X\psi\}$, then $u_{n+1}, N' \models X\psi$.
If γ_j reaches N' before its first reset in $\sigma'_2\sigma_{n+1}\ldots\sigma_m$, then $N = N'+1$, and we can conclude $u_{n+1}, N \models \psi$.
On the contrary, if $N = N'$ and there are strictly less than N' mistakes on φ_1 before the next occurence of φ_2, we can allow one more and keep respecting the constraint on N', so $u_{n+1}, N \models \psi$.
– If $\psi = \varphi_1 U_j^{\leq N} \varphi_2$, $\sigma'_1 = r_j$ and $Y' = Y \setminus \{\psi\} \cup \{\varphi_2\}$ then $N = N'$, and $u_{n+1}, N' \models X\varphi_2$, hence $u_{n+1}, N \models \psi$.

Hence we can conclude that for all k, $a_{n+1}a_{n+2}\ldots a_m, N \models \psi$, which concludes the proof of the lemma.

□

6.6 Details on $\omega\sharp$-expressions

Proof of Lemma 11

Proof. Let f be a regular cost function recognized by $\mathbf{S}_f, h, I$. Let $N = |\mathbf{S}_f|$. It suffices to take $K_f \geq N$ to verify that for any $E \in$ Oexpr, the $\sharp$-expression $E[\omega \leftarrow K_f!]$ is well-formed for f. Moreover, if $s \in \mathbf{S}_f$, $s^{k!} = s^{K_f!}$ for all $k \geq K_f$. Let us show that $f^\infty \uplus f^B =$ Oexpr. Let $E \in$ Oexpr, and $k \geq K_f$. Let $e = E[\omega \leftarrow k!]$, e is well-formed for $\mathbf{S}_f$. For all $n \in \mathbb{N}$, let $u_n = e(n) = E(k!, n)$. From [CKL10], we know that there exists α such that $\rho(h(u_n)) \sim_\alpha \mathrm{eval}(e)|_n \mathrm{eval}(u_n)$.

Therefore,

$$\mathrm{eval}(e) \in I \Rightarrow \forall n, I[\rho(h(u_n))] \geq_\alpha n \Rightarrow \forall n, f(u_n) \geq_\alpha n \Rightarrow \lim f(u_n) = \infty$$

and $\mathrm{eval}(e) \notin I \Rightarrow \forall n, I[\rho(h(u_n))] \leq \alpha(1) \Rightarrow \forall n, f(u_n) \leq \alpha(1) \Rightarrow E \in f^B$. We get that $f^\infty = \{E \in \mathrm{Oexpr}, \mathrm{eval}(E) \in I\}$ and $f^B = \{E \in \mathrm{Oexpr}, \mathrm{eval}(E) \notin I\}$ which shows the result.

Lemma 27 *If $E \equiv_f E'$, then for any context $C_1[x] \in \mathrm{C_{OE}}$, $C_1[E] \equiv_f C_1[E']$.*

Proof. Let E, E' and $C_1[x]$ defined by the Lemma. Let $C[x]$ be a context. We define $C'[x] = C[C_1[x]]$. The definition of the $\equiv_f$ relation implies $C'[E] \rightleftharpoons_f C'[e']$. Hence $C[C_1[e]] \rightleftharpoons_f C[C_1[E']]$.

This is true for any context $C[x]$ so $C_1[E] \equiv_f C_1[E']$. □

Proposition 28 *The relation $\equiv_f$ does not change if we restrict contexts to having only one occurence of x, as it was done for* Expr *in [CKL10].*

Proof. Let $\equiv'_f$ be the equivalence relation defined with single-variable contexts. we just need to show that $E \equiv'_f E' \implies E \equiv_f E'$ (the converse is trivial). Let us assume $E \equiv'_f E'$, and let $C[x_1, x_2]$ be a context with two occurences of x, labelled x_1 and x_2. Then $C[E] = C[x_1 \leftarrow E, x_2 \leftarrow E] \rightleftharpoons_f C[x_1 \leftarrow E, x_2 \leftarrow E'] \rightleftharpoons_f C[x_1 \leftarrow E', x_2 \leftarrow E'] = C[e']$. The generalization to an arbitrary number of occurences of x is obvious, and we get $E \equiv_f E'$. □

Structure of Oexpr/$\equiv_f$ If f is a regular cost function, let $\mathbf{S}_f = \text{Oexpr}/\equiv_f$. We show that we can provide $\mathbf{S}_f$ with a structure of stabilization semigroup $\langle \mathbf{S}_f, \cdot, \leq, \sharp \rangle$.

If $E \in \text{Oexpr}$, let $\overline{E}$ be its equivalence class for the $\equiv_f$ relationship. We first naturally define the stabilization semigroup operators : $\overline{E} \cdot \overline{E'} = \overline{EE'}$ and if $\overline{E}$ idempotent we have $\overline{E} = \overline{E^\omega}$ and $(\overline{E})^\sharp = \overline{E^{\omega\sharp}}$. $\leq$ is the minimal partial order induced by the inequalities $s^\sharp \leq s$ where s is idempotent, and compatible with the stabilization semigroup structure.

Let us show that these operations are well-defined :

Product If $E_1 \equiv_f E_1'$ and $E_2 \equiv_f E_2'$. By Lemma **??** with context xE_2 and $E_1'x$, $E_1E_2 \equiv_f E_1'E_2 \equiv_f E_1'E_2'$, so $\overline{E_1E_2} = \overline{E_1'E_2'}$.

Stabilization If $E \equiv_f E'$, by Lemma **??** with context $x^{\omega\sharp}$, $E^{\omega\sharp} \equiv_f E'^{\omega\sharp}$, hence $\overline{E^{\omega\sharp}} = \overline{E'^{\omega\sharp}}$.

Moreover, it is easy to check that all axioms of stabilization semigroups are verified, for example $(s^\sharp)^\sharp = s^\sharp$ because for any sequence u_n which is either bounded or tends towards ∞, u_{n^2} has same nature as u_n.

Proof of Theorem 14 Let $I_f = \left\{\overline{E}, E \in f^\infty\right\}$, and $h_f : \mathbb{A}^* \to \mathbf{S}_f^*$ the length-preserving morphism defined by $h_f(a) = \overline{a}$ for all $a \in \mathbb{A}$ (a letter is a particular $\omega\sharp$-expression).

Proof. Let $\mathbf{S}_{\min}, h, I$ be the minimal stabilization semigroup recognizing f, as defined in appendix A.7 of [CKL10]. Let ρ be its compatible mapping, and $\text{eval} : \text{Oexpr} \to \mathbf{S}_{\min}$ the corresponding evaluation function. We will show that $E \equiv_f E'$ iff $\text{eval}(E) = \text{eval}(E')$.

We know by the proof of Lemma 11 that $E \in f^\infty \Leftrightarrow \text{eval}(E) \in I$. We remind that the definition of $\mathbf{S}_{\min}$ is based on the fact that if two elements behave the same relatively to I in any context, they are the same. These facts give us the following sequence of equivalences :

$$\begin{aligned} E \equiv_f E' &\Leftrightarrow \forall C[x] \in \mathrm{C_{OE}}, C[E] \rightleftharpoons_f C[E'] \\ &\Leftrightarrow \forall C[x] \in \mathrm{C_{OE}}, (C[E] \in f^\infty \Leftrightarrow C[E'] \in f^\infty) \\ &\Leftrightarrow \forall C[x] \in \mathrm{C_{OE}}, (\text{eval}(C[E]) \in I \Leftrightarrow (\text{eval}(C[E']) \in I) \\ &\Leftrightarrow \text{eval}(E) = \text{eval}(E') \end{aligned}$$

This gives a bijection between $\mathbf{S}_f$ and $\mathbf{S}_{\min}$ (eval function is surjective on $\mathbf{S}_{\min}$, by minimality of $\mathbf{S}_{\min}$). Moreover, this bijection is an isomorphism, since in both semigroups, operations are induced by concatenation and $\sharp$ on $\sharp$-expressions. h is determined by its image on letters, so we have to define $h_f(a) = \overline{a}$ to remain coherent. Finally, we have $\text{eval}(E) \in I \Leftrightarrow E \in f^\infty$, therefore the set I_f corresponding to I in the bijection is $I_f = \left\{\overline{E}, E \in f^\infty\right\}$. □

Growing speeds lemma The following lemma will be used for technical purposes in future proofs, but it is an interesting intuitive statement which could give a better understanding of the behaviour of regular cost functions.

Lemma 29 *Let f be a regular cost function, and $e \in \mathrm{Expr}$ containing N $\sharp$-operators $\sharp_1, \dots, \sharp_N$. For all $i \in \{1, \dots, N\}$, let σ_i be a function $\mathbb{N} \to \mathbb{N}$ with $\sigma_i(n) \to \infty$. Then $f(e[\sharp_i \leftarrow \sigma_i(n)$ for all $i]) \to \infty \Leftrightarrow f(e(n)) \to \infty$. In other words, we can replace some of the n exponents by any function $\sigma(n) \to \infty$ when approximating a $\sharp$-expression by a sequence of words. It does not change the nature of the sequence relatively to f.*

Proof. This result is intuitive : since we always work up to cost equivalence, growing at different speeds has an effect on correction functions, but not on qualitative behaviour.

We will use notation $\underset{n\to\infty}{\bowtie}$: $g_1(n) \underset{n\to\infty}{\bowtie} g_2(n)$ means "$g_1(n)$ is bounded iff $g_2(n)$ is bounded", but remark that here all functions will either be bounded or tend towards ∞ (this notation will be reused in the next section).

For convenience we will note $e_n = e[\sharp_i \leftarrow \sigma_i(n)$ for all $i]$. We want to show that $f(e_n) \underset{n\to\infty}{\bowtie} f(e(n))$. Let $\mathbf{S}_f$ be the minimal stabilization semigroup of f, with compatible function ρ. We will in fact show that there exists α such that for all n, $\rho(e_n) \sim_\alpha \rho(e(n))$, which implies the result. We proceed by induction on N. If $N = 0$, then $e_n = e(n)$ so the result is trivial. We assume the result is true for all $k < N$ (with function $\alpha_<$), and we choose $\sharp_N$ to be outermost (not under another $\sharp$). We can write $e = rs^{\sharp_N}t$, with $r, s, t \in \mathrm{Expr}$.

Let β be a witness of ρ compatible with $\mathbf{S}_f$, and γ such that $n \sim_\gamma \sigma_N(n)$. We have

$$\begin{aligned}
\rho(e_n) &= \rho(r_n(s_n)^{\sigma_N(n)}t_n) \\
&\sim_\beta \tilde{\rho}(\rho(r_n)\rho((s_n)^{\sigma_N(n)})\rho(t_n)) \\
&\sim_\gamma \tilde{\rho}(\rho(r_n)\rho((s_n)^n)\rho(t_n)) \\
&\sim_{\alpha_<} \tilde{\rho}(\rho(r(n))\rho(s(n)^n)\rho(t(n))) \\
&\sim_\beta \rho(e(n)).
\end{aligned}$$

This give us a function α which completes the induction. □

6.7 Proof of Theorem 16

We remind the theorem we want to prove :

Let f be a cost function described by a LTL$^\leq$-formula, then f is regular and the syntactic stabilization semigroup of f is aperiodic.

Proof. We want to show that for all LTL$^\leq$-formula ϕ, $\mathbf{S}_{[\![\phi]\!]}$ is aperiodic.

We proceed by an induction on ϕ and use the characterization of $\mathbf{S}_{[\![\phi]\!]}$ provided by Theorem 14.

If $\phi = a$,

then $S_{[\![\phi]\!]} = \{a, b\}$ with $a \cdot b = a \cdot a = a$, and $b \cdot a = b \cdot b = b$, it is aperiodic (also trivial if $\phi = \neg a$).

If $\phi = \Omega$,

then $S_{[\![\phi]\!]} = \{1, a\}$ with 1 neutral element and $a \cdot a = a$, it is aperiodic.

If $\phi = \varphi_1 \wedge \varphi_2$ or $\phi = \varphi_1 \vee \varphi_2$,

ϕ is recognized by the product semigroup of $\mathbf{S}_{[\![\varphi_1]\!]}$ and $\mathbf{S}_{[\![\varphi_2]\!]}$, which is aperiodic by induction hypothesis.

If $\phi = X\psi$,

we know by induction hypothesis that $\mathbf{S}_{[\![\psi]\!]}$ is aperiodic, so there exists $k \in \mathbb{N}$ such that for any $\omega\sharp$-expression E, $E^k \equiv_{[\![\psi]\!]} E^{k+1}$. We want to show that it is also true for $[\![\phi]\!]$. Let E be a $\omega\sharp$-expression, and $e = E[\omega \leftarrow \max(K_{[\![\phi]\!]}!, K_{[\![\psi]\!]}!)]$ (from Lemma 11).

We want to show that $E^{k+2} \equiv_{[\![\phi]\!]} E^{k+1}$ i.e. for any context $C[x]$, $[\![\phi]\!](C[e^{k+2}](n)) \underset{n\to\infty}{\bowtie} [\![\phi]\!](C[e^{k+1}](n))$.

Let $C[x]$ be a context. If $C[x] = aC'[x]$, then $[\![\phi]\!](C[e^{k+2}](n)) = [\![\psi]\!](C'[e^{k+2}](n)) \underset{n\to\infty}{\bowtie} [\![\psi]\!](C'[e^{k+1}](n)) = [\![\phi]\!](C[e^{k+1}](n))$ (by proposition **??** with context xe).

If the beginning of $C[x]$ is a letter a under (at least) a $\sharp$, we have a context $C'[x]$ such that for any $\sharp$-expression e', $C[e'](n+1) = aC'[e'](n)$. For instance if $C[x] = ((ax)^\sharp b)^\sharp$ then $C'[x] = x(ax)^\sharp b((ax)^\sharp b)^\sharp$. Then we can write $[\![\phi]\!](C[e^{k+2}](n+1)) = [\![\psi]\!](C'[e^{k+2}](n)) \underset{n\to\infty}{\bowtie} [\![\psi]\!](C'[e^{k+1}](n)) = [\![\phi]\!](C[e^{k+1}](n+1))$.

Finally, if $C[x]$ starts with x (possibly under $\sharp$), we expand x in ex in $C[x]$, so that it does not start with x anymore. As before we can get $C'[x]$ such that $C[e^{k+1}](n+1) = aC'[e^k](n)$ and $C[e^{k+2}](n+1) = aC'[e^{k+1}](n)$ for all n, hence

$$\begin{aligned}
[\![\phi]\!](C[e^{k+2}](n+1)) &= [\![\phi]\!](aC'[e^{k+1}](n)) \\
&= [\![\psi]\!](C'[e^{k+1}](n)) \\
&\underset{n\to\infty}{\bowtie} [\![\psi]\!](C'[e^{k}](n)) \\
&= [\![\phi]\!](aC'[e^{k}](n)) \\
&= [\![\phi]\!](C[e^{k+1}](n+1))
\end{aligned}$$

If $\phi = \varphi U \psi$,

we know by induction hypothesis that $\mathbf{S}_{[\![\varphi]\!]}$ and $\mathbf{S}_{[\![\psi]\!]}$ are aperiodic, so there exists $k \in \mathbb{N}$ such that for any $\omega\sharp$-expression E, $E^k \equiv_{[\![\varphi]\!]} E^{k+1}$ and $E^k \equiv_{[\![\psi]\!]} E^{k+1}$. Let E be a $\omega\sharp$-expression. We will show that $E^{k+1} \equiv_{[\![\phi]\!]} E^{k+2}$

Let $C[x]$ be a context in C_{OE}, $K = \max(K_{[\![\varphi]\!]}, K_{[\![\psi]\!]})$, $u_n = C[E^{k+1}](K!, n)$ and $v_n = C[E^{k+2}](K!, n)$. We want to show that $C[E^{k+1}] \rightleftharpoons_{[\![\phi]\!]} C[E^{k+2}]$, i.e. $[\![\phi]\!](u_n) \underset{n\to\infty}{\bowtie} [\![\phi]\!](v_n)$. Assume for example that $[\![\phi]\!](u_n)$ is bounded by m We have $u_n, m \models \phi$ for all n. We can write $u_n = y_n z_n$ with $z_n, m \models \psi$ and for any strict suffix y_n^i of y_n, $y_n^i z_n, m \models \varphi$. Let p_n be the starting position of z_n (position 0 being the beginning of the word). We define y_n^i to be the suffix of y_n starting at position i for all $i \in [\![0, p-1]\!]$. In this way $y_n^0 = y_n$.

Let us focus on the position p_n of the beginning on z_n. The $\sharp$-expression $e = C[E^{k+1}](K!)$ is finite so we can extract a sequence $u_{\delta(n)}$ from u_n such that the beginning position $p_{\delta(n)}$ of $z_{\delta(n)}$ corresponds to the same position p in e. Let $\{e_j, j \in J\}$ be the finite set of $\sharp$-expression such that $e_j^\sharp$ contains position p in e. We choose $J = \{1, r\}$ with $1 \leq j < j' \leq r$ implies $e_j^\sharp$ is a subexpression of $e_{j'}$. For convenience, we label the $\sharp$-operator of $e_j^\sharp$ with j. Note that J can be empty, if p does not occur under a $\sharp$ in e.

We denote by $\overleftarrow{f_j}(\delta(n))$ the number of occurences of $e_j(\delta(n))$ (coming from the corresponding $e_j^\sharp$) in $y_{\delta(n)}$ and we define $\overrightarrow{f_j}(\delta(n))$ in the same way relatively to $z_{\delta(n)}$. We have for all $n \in \mathbb{N}$, $\delta(n) - 1 \leq \overleftarrow{f_j}(\delta(n)) + \overrightarrow{f_j}(\delta(n)) \leq \delta(n)$. The $\delta(n) - 1$ lower bound is due to the fact than p can be in the middle of one occurence of e_j, therefore this occurence does not appear in $y_{\delta(n)}$ nor in $z_{\delta(n)}$.

This implies that for each $j \in J$, we are in one of these three cases :

- $j \in J_1$: $\overleftarrow{f_j}(\delta(n))$ is unbounded and $\overrightarrow{f_j}(\delta(n))$ is bounded.
- $j \in J_2$: $\overleftarrow{f_j}(\delta(n))$ is bounded and $\overrightarrow{f_j}(\delta(n))$ is unbounded.
- $j \in J_3$: $\overleftarrow{f_j}(\delta(n))$ and $\overrightarrow{f_j}(\delta(n))$ are both unbounded .

But J is finite, hence we can extract $\sigma(n)$ from $\delta(n)$ such that for each $j \in J$:

- If $j \in J_1$, $\overleftarrow{f_j}(\sigma(n)) \to \infty$ and $\overrightarrow{f_j}(\sigma(n))$ is constant.
- If $j \in J_2$, $\overleftarrow{f_j}(\sigma(n))$ is constant and $\overrightarrow{f_j}(\sigma(n)) \to \infty$.
- If $j \in J_3$, $\overleftarrow{f_j}(\sigma(n)) \to \infty$ and $\overrightarrow{f_j}(\sigma(n)) \to \infty$.

Remark that if $j < j'$ and $\overrightarrow{f_j} \circ \sigma \neq 0$, then $j \notin J_1$. Symmetrically, if $j < j'$ and $\overleftarrow{f_j} \circ \sigma \neq 0$, then $j \notin J_2$.

We can distinguish three cases for the position of p in $e = C[E^{k+1}](K!)$:

First case : p is before the first occurence of E in e.

We then consider $C'[x] \in \mathrm{C}_{\mathrm{OE}}$ obtained from $C[x]$ by replacing $\sharp_j$ by the constant value of $\overrightarrow{f_j}(\sigma(n))$ for all $j \in J_1$. We have $[\![\psi]\!](z_{\sigma(n)}) \leq m$ for all n, but by Lemma **??**, $[\![\psi]\!](z_n)$ is bounded iff $C'[E^{k+1}] \in [\![\psi]\!]^B$. By induction hypothesis, $C'[E^{k+1}] \in [\![\psi]\!]^B \Leftrightarrow C'[E^{k+2}]] \in [\![\psi]\!]^B$. Let z'_n be the suffix of $C[E^{k+2}](K!, n)$ starting at position p_n. By reusing Lemma **??**, we get that $[\![\psi]\!](z'_{\sigma(n)}) \leq m'$ for some m'.

We still have to show that there exists a constant M such that $[\![\varphi]\!](y^i_{\sigma(n)} z'_{\sigma(n)}) \leq M$ for all n and all $i \in [\![1, p_{\sigma(n)}]\!]$ (the $y^i_{\sigma(n)}$ are not affected by the change from E^{k+1} to E^{k+2}). Let us call $g^i_{\sigma(n)} = [\![\varphi]\!](y^i_{\sigma(n)} z'_{\sigma(n)})$ for more lisibility. Let us assume that no such M exists, then $\left\{g^i_{\sigma(n)}, n \in \mathbb{N}, 1 \leq i \leq p_{\sigma(n)}\right\}$ is unbounded. For all n, we define i_n such that $g^{i_{\sigma(n)}}_{\sigma(n)} = \max\left\{g^i_{\sigma(n)}, 1 \leq i \leq p_{\sigma(n)}\right\}$. By construction, the sequence $g^{i_{\sigma(n)}}_{\sigma(n)} = [\![\varphi]\!](y^{i_{\sigma(n)}}_{\sigma(n)} z'_{\sigma(n)})$ is unbounded. We first extract $\sigma'(n)$ from $\sigma(n)$ such that $g^{i_{\sigma'(n)}}_{\sigma'(n)} \to \infty$.

We can now repeat the same process as before to extract a sequence $\gamma(n)$ from $\sigma'(n)$, such that the starting positions of $y^{i_{\gamma(n)}}_{\gamma(n)}$ for all n correspond to the same position in e, and such that there exists a context $C''[x]$ with $[\![\varphi]\!](y^{i_{\gamma(n)}}_{\gamma(n)} z_{\gamma(n)}) \underset{n\to\infty}{\bowtie} [\![\varphi]\!](C''[E^{k+1}](K!, \gamma(n)))$ (by Lemma **??** again). By adding an extra E (from $k+1$ to $k+2$) and changing z by z' (the y factors are not concerned by occurences of E), we get $g^{i_{\gamma(n)}}_{\gamma(n)} \underset{n\to\infty}{\bowtie} [\![\varphi]\!](C''[E^{k+2}](K!, \gamma(n)))$. By hypothesis, $[\![\varphi]\!](y^{i_{\gamma(n)}}_{\gamma(n)} z_{\gamma(n)})$ bounded by m, and $C''[E^{k+1}] \rightleftharpoons_{[\![\varphi]\!]} C''[E^{k+2}]$, so $g^{i_{\gamma(n)}}_{\gamma(n)}$ is bounded, but we already know that $g^{i_{\gamma(n)}}_{\gamma(n)} \to \infty$. We have a contradiction, so M must exist.

We finally obtain the existence of M such that for all n and valid i, $[\![\varphi]\!](y^i_{\sigma(n)} z'_{\sigma(n)}) \leq M$. This together with the previous result on ψ gives us that $[\![\varphi U \psi]\!](C[E^{k+2}](K!, n)) \leq \max(m', M)$. We got $C[E^{k+1}] \in [\![\phi]\!]^B \implies C[E^{k+2}] \in [\![\phi]\!]^B$. The other direction works exactly the same, by removing one E instead of adding one. Hence we have $C[E^{k+1}] \rightleftharpoons_{[\![\phi]\!]} C[E^{k+2}]$.

Second case : p is after the last occurence of E in e.

This time z_n is not affected by changing from E^{k+1} to E^{k+2}, however it affects some of the y^i_n. Let $y'^i_n z_n$ be the suffixes of $v_n = C[E^{k+2}](K!, n)$, and p'_n the position of the beginning of z_n in v_n. As before, we assume that $\left\{[\![\varphi]\!](y'^i_{\sigma(n)} z_{\sigma(n)}), n \in \mathbb{N} 1 \leq i \leq p_{\sigma(n)}\right\}$ is unbounded, and we build a sequence $y'^{i_{\gamma(n)}}_{\gamma(n)}$ with the same start position in e, such that $[\![\varphi]\!](y'^{i_{\gamma(n)}}_{\gamma(n)} z_{\gamma(n)}) \to \infty$.

We can again extract context $C''[x]$, but we may need to use again Lemma **??**, in order to map the $\sharp$'s of $C''[x]$ with the remaining repetitions of idempotent elements, (which could be any functions $g(n) < n$). The main idea is to map positions in $v_{\gamma(n)}$ with positions in $u_{\gamma(n)}$ in order to be able to bound the values $[\![\varphi]\!](y'^{i_{\gamma(n)}}_{\gamma(n)} z_{\gamma(n)})$ with what we know about the behaviour on $u_{\gamma(n)}$, and so get a contradiction. Three cases are to be distinguished :

- If a factor corresponding to E^{k+2} occurs in the $y'^{i_{\gamma(n)}}_{\gamma(n)}$, the precedent proof stays valid, and we can map $y'^{i_{\gamma(n)}}_{\gamma(n)}$ with some $y^{j_{\gamma(n)}}_{\gamma(n)}$ ($j_{\gamma(n)}$ may be different from $i_{\gamma(n)}$) in order to get the contradiction. The mapping just need to take in account the shift due to the new occurences of E, but the positions in the words are essentially the sames.
- If the remaining factors contain at most k occurences of E, then the position can be matched with positions in u_n without any changes, and we get the contradiction.
- If the remaining factors contain $k+1$ occurences of E, then we can use the equivalence $E^{k+1} \equiv_{[\![\varphi]\!]} E^k$ to match positions in v_n with positions in u_n and get the contradiction. This time we map positions in the first E of each sequence E^k with the corresponding position in the second one. Informally, we "duplicate" the first E of each sequence.

Third case : In all other situations, a combination of the techniques used above gives us the wanted result. We just need to do with ψ what we did with φ in the second case : for instance we may use $E^{k+1} \equiv_{[\![\psi]\!]} E^k$ if $z'_{\sigma(n)}$ contains $k+1$ occurences of E.

As before, the other way is similar, and we finally get $E^{k+1} \equiv_{[\![\phi]\!]} E^{k+2}$. In conclusion, $\mathbf{S}_{[\![\phi]\!]}$ is aperiodic.

If $\boldsymbol{\phi = \varphi U^{\leq N} \psi}$ We just need to adapt the precedent proof to take in account some exceptions in the validities of φ formulae. Indeed removing an occurence of E does not change the number of possible mistakes, but adding one can double it (at worse), since at most two positions in v_n are mapped to the same position in u_n. Hence , under the hypotheses $E^k \equiv_{[\![\psi]\!]} E^{k+1}$ and $E^k \equiv_{[\![\varphi]\!]} E^{k+1}$, we get $E^{k+1} \equiv_{[\![\varphi U^{\leq N}\psi]\!]} E^{k+2}$, with a correction function that doubles the one in the precedent proof. We can conclude that $\mathbf{S}_{[\![\phi]\!]}$ is also aperiodic in this case.

6.8 Proof of Theorem 17

We remind the theorem we want to prove :

Let f be a cost function recognized by an aperiodic stabilization semigroup, then f can be described by a LTL$^{\leq}$-formula.

Proof. This proof is a generalization of the proof from Wilke for aperiodic languages in [Wil99].

Let us first notice that "$\mathbf{S}_f$ is aperiodic" is equivalent to "f is computed by an aperiodic stabilization monoid", since aperiodicity is preserved by quotient and by addition of a neutral element.

We take an alphabet $\mathbb{A} \subseteq \mathbf{M}$ to avoid using a morphism h and simplify the proof. The LTL$^{\leq}$-formulae are about elements of $\mathbf{M}$, and are monotonic in the sense that $[\![a]\!](bu) = 0$ iff $b \geq a$, ∞ otherwise. It is easy to get from this to the general case by substituting in the formula an element m by $\vee_{h(a)\geq m} a$. We also will be sloppy with the empty word ε. It is not more difficult to take it in account, but the addition of a lot of special cases for ε in the proof would make it harder to understand.

We assume that f on alphabet $\mathbb{A} \subseteq \mathbf{M}$ is computed by $\mathbf{M}, I$ with $\mathbf{M}$ aperiodic. Let ρ be compatible with $\mathbf{M}$.

If $m \in \mathbf{M}$, we note f_m the cost function $f_m(u) = \inf\{n/\rho(u)(n) \geq m\}$. It is sufficient to show that the f_m functions are LTL$^{\leq}$-computable, since $f \approx \min_{m \notin I} f_m$.

We proceed by induction on both the size of the stabilization monoid and on the size of the alphabet, the induction parameter being $(|\mathbf{M}|, |\mathbb{A}|)$ for order $<_{lex}$.

We add in the induction hypothesis that $\mathbf{M}$ has a neutral element 1 for multiplication.

If $|\mathbf{M}| = 1$ then f is the constant function 0 or ∞, which is LTL$^{\leq}$-computable.

If $\mathbb{A} = \{a\}$, we can consider that $\mathbf{M} = \{a^i/0 \leq i \leq p\} \cup \{(a^p)^\sharp\}$ (by aperiodicity of $\mathbf{M}$) and $(a^p)^\sharp \leq a^p$ is the only pair in $\leq$. We can show that for all $b \in \mathbf{M}$, f_b is LTL$^{\leq}$-computable :

– If $i < p$, $f_{a^i} \approx [\![\bigwedge_{0\le j<i} X^j a \wedge X^i \Omega]\!]$,
– $f_{a^p} \approx [\![\bot U^{\le N} \Omega]\!]$,
– $f_{(a^p)^\sharp} \approx [\![\bigwedge_{0\le j<p} X^j a]\!]$

Let us assume that $|\mathbf{M}| > 1$, $|\mathbb{A}| > 1$, and the theorem is true for all $(|\mathbf{M}'|, |\mathbb{A}'|) <_{lex} (|\mathbf{M}|, |\mathbb{A}|)$. We choose a letter $b \neq 1 \in \mathbb{A}$, let $\mathbb{B} = \mathbb{A} \setminus \{b\}$.

Let $L_0 = \mathbb{B}^*$, $L_1 = \mathbb{B}^* b \mathbb{B}^*$, and $L_2 = \mathbb{B}^* b(\mathbb{B}^* b)^+ \mathbb{B}^*$. We have $\mathbb{A}^* = L_0 \cup L_1 \cup L_2$.

We define restrictions of $f_m : f_0, f_1, f_2$ on L_0, L_1, L_2 respectively (giving value ∞ outside of the domain). We have $f_m = \min(f_0, f_1, f_2)$. Hence it suffices to show that the f_i's are LTL$^\le$-computable to get that f_m is also LTL$^\le$-computable.

f_0 is computed by $\mathbf{M}$ on alphabet $\mathbb{B}$, so by induction hypothesis there is a formula φ_0 on $\mathbb{B}$ computing f_0. The formula $\varphi_0' = \varphi_0 \wedge G\neg b$ is a formula on $\mathbb{A}$ computing f_0.

For all $x \in \mathbf{M}$, let φ_x be the LTL$^\le$-formula on $\mathbb{B}$ computing f_x (restricted to $\mathbb{B}^*$), these formulae exist by induction hypothesis, since $|\mathbb{B}| < |\mathbb{A}|$.

If φ is a LTL$^\le$-formula on $\mathbb{B}$, we define its "relativisation" φ' on $\mathbb{A}$ which has the effect of φ on the part before b in a word. We define φ' by induction in the following way :

$$\begin{aligned}
a' &= a \wedge XFb \\
\Omega' &= b \\
(\varphi \wedge \psi)' &= \varphi' \wedge \psi' \\
(X\varphi)' &= X\varphi' \wedge \neg b \\
(\varphi U \psi)' &= (\varphi' \wedge \neg b) U \psi' \\
(\varphi U^{\le N} \psi)' &= (\varphi' \wedge \neg b) U^{\le N} \psi'
\end{aligned}$$

With this definition, $[\![\varphi']\!](u_1 b u_2) = [\![\varphi]\!](u_1)$ for any $u_1 \in \mathbb{B}^*$ and $u_2 \in \mathbb{A}^*$.

We define the following formula on $\mathbb{A}$:

$$\varphi_1 = (\bigvee_{xby=m} (\varphi_x' \wedge F(b \wedge X\varphi_y)) \wedge (\neg b U(b \wedge XG\neg b))$$

The second part controls that the word is in L_1. We show $[\![\varphi_1]\!] \approx f_1$.

Let $u \in L_1$, we can write $u = u_1 b u_2$ with $u_1, u_2 \in \mathbb{B}^*$.
By definition of φ_1,
$[\![\varphi_1]\!](u) = \min_{xby=m} \max([\![\varphi_x']\!](u), [\![\varphi_y]\!](u_2))$
$\quad = \min_{xby=m} \max([\![\varphi_x]\!](u_1), [\![\varphi_y]\!](u_2))$
$\quad = \min_{xby=m} \max(f_x(u_1), f_y(u_2))$.

We have for any $z \in \mathbf{M}$ and $v \in \mathbb{B}^*$, $\rho(v) \succeq \bot|_{f_z(v)} z$ where $\bot$ is an extra smallest element (by definition of f_z).

But for any x, y such that $xby = m$,
$\rho(u) \sim \tilde{\rho}(\rho(u_1) b \rho(u_2))$
$\quad \succeq \tilde{\rho}(\bot|_{f_x(u_1)} x \cdot b \cdot \bot|_{f_y(u_2)} y)$
$\quad \succeq \bot|_{\max(f_x(u_1), f_y(u_2))} m$.

It implies that for some β (not depending on u), $\forall x, y$ such that $xby = m$, $f_m(u) \le_\beta \max(f_x(u_1), f_y(u_2))$.

In particular, $f_1(u) = f_m(u) \le_\beta \min_{xby\in I} \max(f_x(u_1), f_y(u_2)) = [\![\varphi_1]\!](u)$. We can conclude $f_1 \preccurlyeq [\![\varphi_1]\!]$.

Conversely, let us assume that $f_1(u) \leq n$, it means that $\rho(u)(n) \geq m$. but $\rho(u) \sim_\alpha \rho(u_1) \cdot b \cdot \rho(u_2)$, so $\rho(u_1)(\alpha(n)) \cdot b \cdot \rho(u_2)(\alpha(n)) \geq m$.

Let $x = \rho(u_1)(\alpha(n))$ and $y = \rho(u_2)(\alpha(n))$, we have $f_x(u_1) \leq \alpha(n)$ and $f_y(u_2) \leq \alpha(n)$, so $\max(f_x(u_1), f_y(u_2)) \leq \alpha(n)$. We get $[\![\varphi_1]\!](u) \leq \alpha(n)$, and in conclusion $[\![\varphi_1]\!] \preccurlyeq f_1$. This concludes the proof of $[\![\varphi_1]\!] \approx f_1$.

Last but not least, we have to show that f_2 is LTL$^{\leq}$-computable. For that we will finally use the induction hypothesis on the size of the monoid (until now we only have decreased the size of the alphabet and kept the monoid unchanged).

We define the stabilization monoid $\mathbf{M}' = \langle Mb \cap bM, \circ, \natural, \leq' \rangle$ in the following way : $xb \circ by = xby$, and for xb idempotent $(xb)^\natural = (x^\omega)^\sharp b$ where $x^\omega = x^{|\mathbf{M}|}$ is idempotent, since $\mathbf{M}$ is aperiodic. $\mathbf{M}'$ is a stabilization monoid, let ρ' be compatible with $\mathbf{M}'$. We can first notice that this definition implies $(xb)^k = x^k b$, so $\mathbf{M}'$ is also aperiodic. Moreover, if $1 \in \mathbf{M}'$, let $n = |\mathbf{M}|$, $1 = xb = (xb)^k = x^k b^k = x^k b^{k+1} = (xb)^k b = 1b = b$, but $b \neq 1$ so $1 \notin \mathbf{M}'$, b is the neutral element for $\circ$ in $\mathbf{M}'$, and $|\mathbf{M}'| < |\mathbf{M}|$, which allows us to use induction hypothesis on $\mathbf{M}'$ with alphabet $\mathbf{M}'$.

Let $\Delta = b(\mathbb{B}^* b)^+$, then $L_2 = \mathbb{B}^* \Delta \mathbb{B}^*$.

Let $d \in \mathbf{M}$, we first want to show that f_d over language Δ is LTL$^{\leq}$-computable.

Let $\begin{array}{l} \sigma : \Delta \to (\mathbf{M}'^{\mathbb{N}})^* \\ bu_1 b \ldots u_k b \mapsto (b\rho(u_1)b) \ldots (b\rho(u_k)b) \end{array}$

By induction hypothesis, for any $x \in \mathbf{M}'$, there exists a LTL$^{\leq}$-formula ψ_x on alphabet $\mathbf{M}'$ and a correction function α such that for any $v \in \mathbf{M}'^*$,
$[\![\psi_x]\!](v) \approx_\alpha \inf\{n/\rho'(v)(n) \geq x\}$.

Definition 30 *Let $\mathbf{S}$ be a stabilization monoid. Let f be a cost function $\mathbf{S}^* \to \mathbb{N}^\infty$, and $\mathbf{S}^\uparrow$ be the set of α-increasing sequences of elements of $\mathbf{S}$ (for some α). we define $\tilde{f} : \mathbf{S}^\uparrow \to \mathbb{N}_\infty$ by $\tilde{f}(\boldsymbol{u}) = \inf\{n/f(u_n) \leq n\}$.*

Remark that this notation is coherent with the $\tilde{}$ operator previously defined for functions $E \to F^{\mathbb{N}}$ in the sense that if f is recognized by $\mathbf{S}, h, I$ with compatible function ρ, then $\tilde{f} \approx \boldsymbol{u} \mapsto I[\tilde{\rho}(h(\boldsymbol{u}))]$.

Lemma 31 *We claim that there exists α and ϕ_d a LTL$^{\leq}$-formula on alphabet $\mathbb{A}$ such that for all $u \in \Delta$ and $v \in \mathbb{B}^*$:*

$$[\![\phi_d]\!](uv) \approx_\alpha \widetilde{[\![\psi_d]\!]}(\sigma(u)) \approx_\alpha f_d(u)$$

With this result we can build a formula φ_2 computing f_2 :

$$\varphi_2 = (\bigvee_{xdy=m} (\varphi'_x \wedge F(b \wedge X\phi_d)) \wedge F(b \wedge X(G\neg b \wedge \varphi_y))) \wedge \varphi_{L_2}$$

where $\varphi_{L_2} = F(b \wedge XFb)$ controls that the word is in L_2.

By construction, lemmas and induction hypothesis, there exists α such that for all $v_1, v_2 \in \mathbb{B}^*$ and $u \in \Delta$,

$$\begin{aligned} [\![\varphi_2]\!](v_1 u v_2) &\approx_\alpha \min_{xdy=m} \max([\![\varphi'_x]\!](v_1 u v_2), [\![\phi_d]\!](u v_2), [\![\varphi_y]\!](v_2)) \\ &\approx_\alpha \min_{xdy=m} \max(f_x(v_1), f_d(u), f_y(v_2)). \end{aligned}$$

The proof that $min_{xdy=m} \max(f_x(v_1), f_d(u), f_y(v_2)) \approx f_m(v_1uv_2)$ is similar to the proof of $[\![\varphi_1]\!] \approx f_1$.

All this together gives us $[\![\varphi_2]\!] \approx f_2$, which concludes the proof. $\square$

Proof of Lemma ??

Proof. First let us show that $\widetilde{[\![\psi_d]\!]}(\sigma(u)) \approx_\alpha f_d(u)$ for some α and all $u \in \Delta$. Let $u = bu_1bu_2 \dots u_kb$ with $u_i \in \mathbb{B}^*$. For each $i \in [\![1, k]\!]$ and $t \in \mathbb{N}$, $\rho(u_i)(t) = a_{i,t} \in \mathbf{M}$. For all $t \in \mathbb{N}$, let $v_t = (ba_{1,t}b) \dots (ba_{k,t}b)$, v_t is a word on $\mathbf{M}'$ of length k, and $\sigma(u) = (v_t)_{t\in\mathbb{N}}$. Finally, let $w_t = ba_{1,t}ba_{2,t} \dots ba_{k,t}b$ of length $2k + 1$ on $\mathbf{M}$. We have :

$$\begin{aligned}\widetilde{[\![\psi_d]\!]}(\sigma(u))) &= \inf \{t/[\![\psi_d]\!](v_t) \leq t\} \\ &\approx \inf \{t/ \inf \{n/\rho'(v_t)(n) \geq d\} \leq t\}\end{aligned}$$

We can verify that $\rho'(v_t) \sim \rho(w_t)$ for any t : we check that ρ' verifies the same axioms on words $(ba_1b) \dots (ba_kb)$ than ρ does for $ba_1ba_2 \dots a_kb$. The only interesting case is the stabilization rule : let bab be an idempotent of $\mathbf{M}'$, $\rho'((bab)^p) \sim (bab)^\natural|_p(bab) \sim (ba)^{\omega\sharp}b|_p(bab)$. But if $p = |\mathbf{M}|p' + p''$ with $p'' < |\mathbf{M}|$,

$$\begin{aligned}\rho((ba)^pb) &\sim \rho((ba)^{|\mathbf{M}|})^{p'}) \cdot (ba)^{p''}b \\ &\sim^{(1)} \rho(((ba)^\omega)^{p'}) \cdot (ba)^{p''}b \\ &\sim (ba)^{\omega\sharp}(ba)^{p''}b|_{p'}(ba)^\omega(ba)^{p''}b \\ &\sim^{(2)} (ba)^{\omega\sharp}b|_p(bab).\end{aligned}$$

We get the equivalence (1) by aperiodicity of $\mathbf{M}$ ($(ba)^\omega$ is now a letter and no longer a word of length $|\mathbf{M}|$), and (2) by the fact that bab is idempotent in $\mathbf{M}'$ so $(ba)^\omega(ba)^{p''}b = bab$, and $(ba)^\omega(ba)^{p''} = (ba)^\omega$ by aperiodicity of $\mathbf{M}$ (and also $p \approx_{\times(|\mathbf{M}|+1)} p'$).

We can then apply the unicity theorem from [Col09] : ρ is unique up to $\sim$, hence we have $\rho'(v_t) \sim \rho(w_t)$ for any t.

Moreover, let $\boldsymbol{w} = (w_t)_{t\in\mathbb{N}}$, we show that

$$\inf \{n'/\tilde{\rho}(\boldsymbol{w})(n') \geq d\} \approx \inf \{t/ \inf \{n/\rho(w_t)(n) \geq d\} \leq t\} : (EQ).$$

Let $N' = \inf \{n'/\tilde{\rho}(\boldsymbol{w})(n') \geq d\}, \rho(w_{N'})(N') \geq d$ and $N' \leq N'$ so $N' \geq \inf \{t/ \inf \{n/\rho(w_t)(n) \geq d\} \leq t\}$.

Conversely, let $T = \inf \{t/ \inf \{n/\rho(w_t)(n) \geq d\} \leq t\}$ and N the corresponding value of $\inf \{n/\rho(w_t)(n) \geq d\}$, we have $N \leq T$ and $\rho(w_t)$ is α-increasing, so $\rho(w_T)(T) \geq_\alpha \rho(w_T)(N) \geq d$, i.e. $T \geq_\alpha \inf \{n'/\tilde{\rho}(\boldsymbol{w})(n') \geq d\}$.

Hence we have the equivalence (EQ).

Finally,

$$\begin{aligned}\widetilde{[\![\psi_d]\!]}(\sigma(u))) &\approx \inf \{t/ \inf \{n/\rho(w_t)(n) \geq d\} \leq t\} \\ &\approx \inf \{n/\tilde{\rho}(\boldsymbol{w})(n) \geq d\} && \text{by } (EQ) \\ &= \inf \{n/\tilde{\rho}(b\rho(u_1)b\rho(u_2) \dots \rho(u_k)b)(n) \geq d\} \\ &\approx \inf \{n/\rho(bu_1bu_2 \dots u_kb)(n) \geq d\} && \text{Substitution axiom} \\ &\approx f_d(u).\end{aligned}$$

which concludes the proof of $\widetilde{[\![\psi_d]\!]}(\sigma(u))) \approx f_d(u)$.

It remains to show that there exists a formula ϕ_d and a α such that for all $u, v \in \Delta \times \mathbb{B}^*$, $[\![\phi_d]\!](uv) \approx_\alpha \widetilde{[\![\psi_d]\!]}(\sigma(u))$.

If ψ is a LTL$^{\leq}$-formula on $\mathbf{M}'$, we define $\psi^\bigstar$ on alphabet $\mathbb{A}$ by induction on ψ:

$$
\begin{aligned}
x^\bigstar &= (b \wedge XFb) \wedge (X\varphi'_x) \\
(\psi_1 \wedge \psi_2)^\bigstar &= \psi_1^\bigstar \wedge \psi_2^\bigstar \\
(\psi_1 \vee \psi_2)^\bigstar &= \psi_1^\bigstar \vee \psi_2^\bigstar \\
(X\psi)^\bigstar &= \neg b U(b \wedge \psi^\bigstar) \\
(\psi_1 U \psi_2)^\bigstar &= (b \implies \psi_1^\bigstar) U (b \wedge \psi_2^\bigstar) \\
(\psi_1 U^{\leq N} \psi_2)^\bigstar &= (b \implies \psi_1^\bigstar) U^{\leq N} (b \wedge \psi_2^\bigstar).
\end{aligned}
$$

Where φ'_x is defined as before for any φ_x on alphabet $\mathbb{B}$.

Let us show by induction on ψ that that $[\![\psi^\bigstar]\!](uv) \approx \widetilde{[\![\psi]\!]}(\sigma(u))$ for $u = bu_1bu_2 \dots u_kb \in \Delta$ and $v \in \mathbb{B}^*$:

- If $x \in \mathbf{M}'$,
  $[\![x^\bigstar]\!](uv) = [\![\varphi'_x]\!](u_1bu_2 \dots u_kbv) = [\![\varphi_x]\!](u_1)$, and
  $\widetilde{[\![x]\!]}(\sigma(u)) = \inf\{n / [\![x]\!](\rho(u_1)(n)) \leq n\} \approx \inf\{n / (\rho(u_1)(n)) \geq x\} \approx [\![\varphi_x]\!](u_1)$.
- $\wedge$ case :
  $$
  \begin{aligned}
  [\![(\psi_1 \wedge \psi_2)^\bigstar]\!](uv) &= \max([\![\psi_1^\bigstar]\!](uv), [\![\psi_2^\bigstar]\!](uv)) \\
  &\approx \max(\widetilde{[\![\psi_1]\!]}(\sigma(u)), \widetilde{[\![\psi_2]\!]}(\sigma(u))) \\
  &\approx \widetilde{[\![\psi_1 \wedge \psi_2]\!]}(\sigma(u))
  \end{aligned}
  $$
- $\vee$ case :
  $$
  \begin{aligned}
  [\![(\psi_1 \vee \psi_2)^\bigstar]\!](uv) &= \min([\![\psi_1^\bigstar]\!](uv), [\![\psi_2^\bigstar]\!](uv)) \\
  &\approx \min(\widetilde{[\![\psi_1]\!]}(\sigma(u)), \widetilde{[\![\psi_2]\!]}(\sigma(u))) \\
  &\approx \widetilde{[\![\psi_1 \vee \psi_2]\!]}(\sigma(u))
  \end{aligned}
  $$
- X case :
  $$
  \begin{aligned}
  [\![(X\psi)^\bigstar]\!](uv) &= [\![\psi^\bigstar]\!](bu_2b \dots u_kbv) \\
  &\approx \widetilde{[\![\psi]\!]}(\sigma(bu_2b \dots u_kb)) \\
  &\approx \widetilde{[\![X\psi]\!]}(\sigma(bu_1bu_2b \dots u_kb))
  \end{aligned}
  $$
- U case :
  $$
  \begin{aligned}
  [\![(\psi_1 U \psi_2)^\bigstar]\!](uv) &= \min_{1 \leq j \leq k}(\max([\![\psi_2^\bigstar]\!](bu_jb \dots u_kbv), \max_{1 \leq i \leq j}[\![\psi_1^\bigstar]\!](bu_ib \dots u_kbv))) \\
  &\approx \min_{1 \leq j \leq k}(\max(\widetilde{[\![\psi_2]\!]}(\sigma(bu_jb \dots u_kb)), \max_{1 \leq i \leq j} \widetilde{[\![\psi_1]\!]}(\sigma(bu_ib \dots u_kb)))) \\
  &\approx \widetilde{[\![\psi_1 U \psi_2]\!]}(\sigma(u))
  \end{aligned}
  $$
- The $U^{\leq N}$ case is the same than above, allowing at most N mistakes for ψ_1.
  We now just have to take $\phi_d = \psi_d^\bigstar$ to complete the proof of Lemma **??**.

□

6.9 Case of unregular cost functions

The syntactic congruence still can defined on unregular language, and the number of equivalence classes becomes infinite, whereas we need cost functions to be regular a priori to define their syntactic congruence.

Here, if f is not regular, $\equiv_f$ may not be properly defined, since we use the existence of a minimal stabilization semigroup of f to give a semantic to the operator ω. But we can go back to $\sharp$-expressions and define $\sim_f$ on Expr for all f in the following way : $e \sim_f e'$ if for any $C[x]$ context on $\sharp$-expressions, $\{f(C[e])(n), n \in \mathbb{N}\}$ is bounded iff $\{f(C[e'])(n), n \in \mathbb{N}\}$ is bounded.

In this way if f is regular, then for all $e, e' \in \text{Expr}$, $e \sim_f e'$ iff $e[\sharp \leftarrow \omega\sharp] \equiv_f e'[\sharp \leftarrow \omega\sharp]$. In particular $\text{Expr}/{\sim_f}$ is bigger than $\text{Oexpr}/{\equiv_f}$ when f is regular : there might be equivalence classes corresponding to $\sharp$-expressions that are not well-formed for f.

However, if f is not regular, $\text{Expr}/{\sim_f}$ is not infinite in general (this differs from the results in language theory).

Example 32 *Let* $f(u) = \min_{e \in \text{Expr}} \{|e|, \exists n \in \mathbb{N}, u = e(n)\}$*, there is only one equivalence class for* $\sim_f$ *(*$f(C[e](n))$ *is always bounded by* $|C[e]|$*) so* $\text{Expr}/{\sim_f}$ *has only one element, and therefore cannot contain a stabilization semigroup computing* f*. This gives us a proof that* f *is not regular.*