Description of the thesis: Study of classes of regular cost functions.

Denis Kuperberg

1 Brief Description

The goal of the thesis was to study the recent theory of regular cost functions. This theory extends strictly the theory of regular languages by quantitative features, hence allowing to define functions instead of languages. These quantitative features are of counting nature, for instance it is possible to count occurrences of letters, length of segments, etc, and to combine this quantities using operators like minimum, maximum among others. Cost functions can be effectively represented by means of logic (cost monadic logic), algebra (stabilisation monoid), automata (B- and S-automata), or expressions.

An important and deep part of language theory is devoted to the analysis and comparison of the expressiveness of various formalisms that capture fragments of regular languages. In the thesis, prolongating this branch of research, we characterized and compared classes of regular cost functions. The central contributions of the thesis are the following.

Over finite words, the fragment of "temporal regular cost functions" is characterized in several ways and membership is shown to be decidable (this class has no equivalent in the theory of regular languages). The second contribution is to raise the deep classical Schützenberger-McNaughton-Papert-Kamp theorem for languages to the level of cost functions. The study of these classes requires the introduction of a new theoretical tool, the syntactic stabilization monoid. Over infinite words, the equivalence between weak and full cost monadic logic is established. Over infinite trees, we have studied the result of Rabin characterizing (non effectively) the weak fragment of monadic logic and shown, first, that it was not extendable as itself to the framework of cost functions, and, second, that it can be recovered if one introduces the new class of quasi-weak regular cost functions. This class, that has no meaning in the theory of languages, is strictly more expressive than weak cost functions. Finally, we showed that the theory of cost functions can bring further insights to more classical problems, by obtaining a new decidability result regarding languages of infinite trees. This result settles a particular case in the open problem of decidability of the level in the Mostowski hierarchy for regular languages of infinite trees.

The consequence of these contributions is that the theory of regular cost functions is now much better understood. We now have a clear picture of the many similarities it has with regular languages, as well as the important differences they have, both at a technical and result level.

2 Scientific context

The subject of the thesis is the study of regular cost functions. This theory generalizes the notion or regular languages in all their aspects (automata, algebraic, logics temporal or not, games, etc). All these notions are now well understood in the classical case of languages and the subject of a large body of works. In this presentation of the scientific context, we will concentrate on the notion of cost function, and try to avoid to spend too much time on the classical results on regular languages.

Regular cost functions were introduced by Colcombet [Col09], thus unifying several branches of research involving formal language theory and limitedness related results. The principle of cost functions is to extend the notion of regular languages with quantitative aspects. A **language** is a set of finite words (or by extensions of infinite words, or trees, or infinite trees, depending on the context). Seen differently, a language can be seen as a function from words (or infinite words,...) to two values, 'inside' or 'outside'. The subject of cost functions is to consider instead of languages functions from words to $\mathbb{N} \cup \{\infty\}$. Under this view, 0 can be understood as 'inside' and ∞ as 'outside'. A range of intermediate values becomes now available.

As for regular languages, that can be defined by means of automata, Kirsten and then Colcombet and Bojanczyk [BC06] define two dual notions of counter automata, named B-automata and S-automata. These automata compute functions from words to $\mathbb{N} \cup \{\infty\}$. These automata can typically count events and combine them in an intricated way involving non-determinism.

However, these two forms of automata are not equivalent as such, and even worth, it is not possible to decide, given two B-automata, if the functions computed are equal (this result, due to Krob, holds in fact for very weak forms of such automata [Kro94]). This result, *a priori*, rules out all possibilities of effectively manipulating such computational models and have meaningful decision procedures for them. To achieve decidability, the precision has to be relaxed, and functions are considered modulo an equivalence relation, as follows.

We define the **equivalence** relation $\approx$ between function from words (or other domains) to $\mathbb{N} \cup \{\infty\}$, by

$$f \approx g \text{ if } f \text{ and } g \text{ are bounded on the same sets of inputs.}$$

Formally, a **cost function** is an equivalence class of functions, for the relation $\approx$. Once considered modulo this equivalence relations, the two models of B- and S-automata become effectively equivalent, as well as equivalent with several other formalisms (finite stabilisation monoids, cost monadic logic, ...).

Notice that every language $L \subseteq A^*$ can be seen as the cost function χ_L, defined by $\chi_L(u) = 0$ if $u \in L$ and $\chi_L(u) = \infty$ if $u \notin L$. This allows to consider cost function theory as a strict extension of language theory. Colcombet pushes the extension further by defining an extension of the notion of monoid (called *stabilisation monoid*), and an extension of monadic second order logic, that recognize functions instead of languages. He proves the equivalence between all these formalisms (automata, logic, and stabilization monoid), yielding a clean notion of *regular cost function.* In particular, questions of the form $f \approx g$ are decidable in this framework.

Remark that with this definition, a function f is bounded (by a non-negative integer) if and only if $f \approx 0$. For this reason, this decision procedure subsume every limitedness (aka, boundedness) problem know in the litterature ([Has82, Kir05]).

This new theory allows to develop in a more general way some ideas introduced to solve important decidability problems in language theory. Here are three such problems, that were proved decidable by reducing to boundedness problems for weighted automata:

- **The finite Power property** (finite words) [Sim78, Has79]:
  Is there n such that $(L+\varepsilon)^n = L^*$?
- **Fixed Point Iteration** (finite words)[BOW09]:
  Can we bound the number of fixpoint iterations in a MSO formula ?
- **Star-Height** (finite words/trees) [Has88, Kir05, CL10]:
  Given a non-negative integer n: Does there exists a regular expression recognizing L, and that uses at most n nesting of Kleene stars?

Indeed, every classical problem that can be reformulated in terms of existence of bounds for automata with counters can be solved using the cost function theory. So we can see this new theory as a "toolbox" that allows us to show decidability for a range of problems, for which other techniques were unsufficient. The more cost function theory develops, the more we can hope solving decidability problems that were left open because we lacked the tools to approach them. It is in particular likely to help with open problems related to regular languages such as deciding the Mostowski hierarchy (this will be presented below).

Cost automata

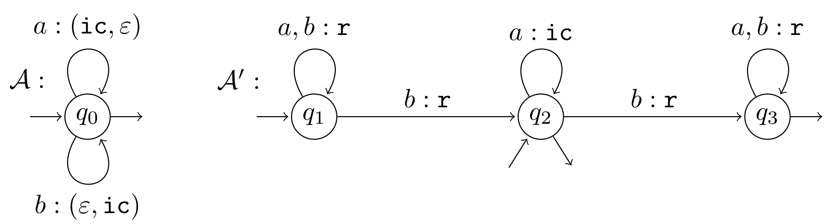
To give some insight concerning the expressive power of regular cost functions, we introduce below the formalism of B- and S-automata, that we call generally **cost automata**. For the sake of keeping the description concise, we chose not to present the other formalisms, that are necessary in the development of the theory, at least in this part devoted to the scientific context. We will stay general enough so that the definitions below can be used for any kind of input structure, from finite word to infinite trees. That is we will not speak precisely of the accepting condition, or the precise structure of the transition function.

B-automata :

A **B-automaton** is a nondeterministic finite automaton, with a finite set of **counters**. On each transition, **actions** on the counters can be performed. For each counter, the following actions are allowed:

- Do nothing (ε),
- **Increment** (`ic`), and;
- **Reset** (`r`).

Typically, B-automata would be drawn as the following automata $\mathcal{A}$ and $\mathcal{A}'$:



The convention is that each transition is labelled $a : (\tau_1, \dots)$, where a is the input letter for which the transition is allowed, and τ_i is the action (among ε, `ic` and `r`) that should be performed on the ith counter when the transition is taken. Ingoing edges without origin denote **initial states**, and outgoing edges without destination denote **final states**. When writing, $a, b : \tau$, this represents the two transitions that are labelled $a : \tau$ and $b : \tau$. We also remove the parentheses from the notation when there is only one counter (case of the automaton $\mathcal{A}'$).

The semantics of these automata is informally described as follows. The counters start with value 0, and along a run, their values are updated according to the transitions (ε leaves the value of the counter unchanged, `ic` adds 1 to its value, and `r` sets it back to 0).

The semantic of a B-automaton $\mathcal{A}$ is the cost function $[\![\mathcal{A}]\!]_B$ from words to $\mathbb{N} \cup \{\infty\}$ defined for all input words u by:

$$[\![\mathcal{A}]\!]_B(u) = \inf\{n \in \mathbb{N} : \text{there is a valid run of } \mathcal{A} \text{ in which no counter value ever exceeds } n\}.$$

In the above definition, the infimum is considered in $\mathbb{N} \cup \{\infty\}$. This means that when there is no run of the automaton over some input, the corresponding output value is ∞.

Let us remark that since this is considered as a cost function, replacing "exceeds" by "exceeds or is equal" in the definition would not change the semantic of $\mathcal{A}$ (up to $\approx$).

Example 2.1 *Let us work on finite words on alphabet $\{a, b\}$, and consider the automata $\mathcal{A}$ and $\mathcal{A}$' described above. On an input word u, the automaton $\mathcal{A}$*

computes $\max(|u|_a, |u|_b)$*, while* $\mathcal{A}'$ *computes the size of the minimal block of* a*'s. The automaton* $\mathcal{A}$ *is deterministic and uses two counters: one for* $|u|_a$ *and the other for* $|u|_b$*. Since both counter have to stay below* n*, the optimal value for* n *is the maximum of the two.*

On the other hand, $\mathcal{A}'$ *is non-deterministic, and has only one counter. Each run of this automaton counts a block of* a*'s, so by using the best (i.e. minimal) run, the automaton computes the minimal block of* a*'s in the input word.*

S-automata

As before, an S-automaton is a nondeterministic finite automaton, with a finite set of counters which are updated at each transition. However, the set of atomic counter actions is changed:

- Do nothing (ε),
- **Increment** (`i`),
- **Reset** (`r`).
- **Check-Reset** (`cr`).

As before, counters start with value 0, but this time only the checked values (when action `cr` is performed) will be taken into account. The same explanation justifies the notation `ic` (short for increment-check) for increments in B-automata.

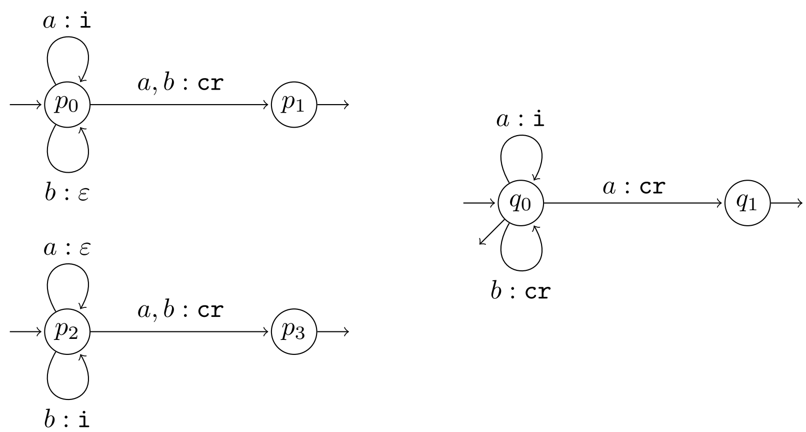
The semantic of an S-automaton is the cost function defined by:

$$[\![\mathcal{A}]\!]_S(u) = \sup\{n \in \mathbb{N} : \text{there is a valid run of } \mathcal{A} \text{ in which every checked value is above } n\}.$$

This can be viewed as a dualization of B-automata : inf and sup have been reversed in the definitions. Moreover, if $\mathcal{A}$ is a classical automaton for a language L, then $[\![\mathcal{A}]\!]_B = \chi_L$ and $[\![\mathcal{A}]\!]_S = \chi_{\overline{L}}$.

In this sense, switching B- and S- generalizes language complementation.

Example 2.2 *We define the two cost functions of Example 2.1, but this time with* S*-automata:*



The first automaton (union of the two on the left) is non-deterministic: on input u, it guesses whether it will count $|u|_a$ or $|u|_b$. Therefore, the S-semantic of the automaton is $\max(|u|_a, |u|_b)$ *by definition*

The second automaton is (almost) deterministic: every block of a is counted (the automaton is forced to guess the end of the word if it ends with a, to perform a `cr` *on the last block). This way, it computes the minimal size of these blocks, since they all have to be bigger than the wanted n. Note that the last a can be lost, but this does not change the semantic up to $\approx$.*

Contributions of the thesis

The theory of regular languages is developed in many related, yet different, directions. In particular, the notion of regularity of languages is studied for usual finite words (the seminal works of Büchi, Elgot, Rabin, Scott, Trakhtenbrot, ...), for infinite words (the key results of Büchi, McNaughton and Safra), for trees (Thatcher and Wright), and for infinite trees (the famous result of Rabin [Rab69]).

The thesis covers essentially this whole range of possibilities (but finite trees). The structure of the rest of the document follows this classification. In Section 3, the contribution for regular cost functions over finite words are presented. The case of infinite words is the subject of section 4. Finally, Section 5 addresses the case of infinite trees.

3 Contributions : Finite words

As the theory of regular languages can be instantiated on various models of words or trees, regular cost functions need to be considered in several such situations. The thesis contributes to the theory of cost functions in all these directions. In this first section, the contributions that are concerned with finite words are presented.

What was known prior to the thesis was essentially decision procedures, and procedures for effectively performing closure under certain operations, and translation among various models of acceptors. For languages, another kind of questions is the subject of intense research, the characterization of classes of regular languages. The general question is of the form:

> Can we characterize the regular languages that belong to a class $\mathcal{C}$? And if possible, can we decide whether a given regular language belongs to the class?

where the class $\mathcal{C}$ can be the set of languages definable using a certain logic, using a certain family of automata, etc. Such results witness usually a very strong and deep understanding of the class under consideration.

In the thesis, the characterization of several classes of regular cost functions over finite words has been considered. The first one, the **aperiodic fragment** generalizes a very famous result (in fact several) of Schützenberg, McNaughton, Papert and Kamp. It is the subject of Section 3.1. In particular, the development of this result requires the introduction of a not yet available notion, the syntactic stabilisation monoid. The second class under consideration, **temporal cost functions** has no equivalent for languages. This is the subject of Section 3.2.

3.1 Aperiodic Class

The class of star-free languages offers a nice example of fruitful interaction between algebra and computer science. The original theorem by Schützenberger [Sch65] states that languages expressible by star-free regular expressions are exactly those recognizable by aperiodic monoids. This algebraic characterization allows to show decidability of membership for this class. Later, Papert, McNaughton and Kamp [MP71, Kam68] showed that this class is also characterized by counter-free automata, first-order logic and linear temporal logic. This extraordinary robustness of the class of first-order definable language made of the Schützenberger-McNaughton-Papert-Kamp theorem the origin of a productive branch of research.

I was attached to generalise these equivalences, starting with the different formalisms involved. In the remainder of this section, various (equivalent) formalisms are introduced, that represents the generalization of all the formalisms involved in the classical result.

Cost Logics

We start our description of the models of computations involved in our description of the aperiodic case by the logical formalisms. There are two of them, the temporal logic CLTL and the first order variant CFO. Let us introduce successively their syntax, and then their semantics.

We will start with CLTL (for Cost Linear Temporal Logic), which generalises LTL to the cost setting. Let us remind that LTL is a modal logic that is interesting from both pratictal and theoretical points of view. In particular, it is well-suited for specifying a desired system behaviour, and its syntax without variables is easy to handle, so it is often used for practical purpose in a context of verification. See for instance [GHR94] for an introduction to temporal logic.

We will directly give the grammar that describes the syntax of CLTL-formulae over a finite alphabet $\mathbb{A}$:

$$\varphi := a \in \mathbb{A} \mid \Omega \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathbf{X}\varphi \mid \varphi \mathbf{U} \psi \mid \varphi \mathbf{U}^{\leq N} \psi$$

Notice the absence of negations. However, any classical LTL-formula (where negations are available) can still be represented, by pushing negations to the leaves. The special symbol Ω stands for the end of the word. LTL is usually used on infinite words, where this symbol is not needed, but either negation or the "Release" operation (dual of Until) is needed then. The other symbols, except for $\mathbf{U}^{\leq N}$ are from classical LTL: a means that the current letter is a, $\wedge$ and $\vee$ are conjunction and disjunction, $\mathbf{X}\varphi$ means that φ is true at the next position, and $\varphi\mathbf{U}\psi$ stands for "φ Until ψ", i.e. ψ is true somewhere in the future, and φ is true until then.

The new construction $\varphi\mathbf{U}^{\leq N}\psi$ is a slight modification of the $\mathbf{U}$ operator. It means that ψ is true somewhere in the future, and φ is false at most N times in the meantime. The "Error variable" N is free and unique in the formula, i.e. it is shared by all occurences of $\mathbf{U}^{\leq N}$. We can also define $\mathbf{G}^{\leq N}\varphi$ as $\varphi\mathbf{U}^{\leq N}\Omega$, meaning that φ is false at most N times in the future.

In the same way, we generalize first-order logic to CFO, defined with the following grammar, where variables quantify over positions in the word:

$$\varphi := a(x) \mid x = y \mid x < y \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \forall x \varphi \mid \forall^{\leq N} x \varphi$$

The only differences with classical FO is the new operator $\forall^{\leq N} x\varphi$, and the absence of negations. Again, since negations can be pushed to the leaves, this logics embeds FO. As before, N is a unique free variable measuring a number of errors. That is to say, $\forall^{\leq N} x\varphi(x)$ means φ is false on at most N positions.

Semantics of Cost Logics

We will now describe how a formula of one of these logics can define a cost function. We start by extending the notion of a word being a model of a formula. Here we additionally need to provide a value for the free variable N, in order to be able to evaluate a formula over a word of $\mathbb{A}^*$.

More formally, let φ be a CLTL or CFO formula. We write $(u, n) \models \varphi$ to signify that $u \in \mathbb{A}^*$ satisfies the formula φ, with $n \in \mathbb{N}$ as value for N in φ (this can be defined more rigorously by induction on the structure of φ).

Example 3.1 *Let φ be the CLTL-formula $\mathbf{G}(a\mathbf{U}^{\leq N}(b \vee \Omega))$. This formula requires that at any position of the input word, all positions but at most N until the next b or the end of the word are labelled by a. Therefore, if we take as input word $u = aabbcbccacacbac$, we have $(u, 5) \models \varphi$ but $(u, 3) \not\models \varphi$, because of the factor ccacacb which causes 4 errors.*

We now want to associate a cost function $[\![\varphi]\!] : \mathbb{A}^* \to \mathbb{N} \cup \{\infty\}$ to any formula φ, of CLTL or CFO. This cost function is defined as follows:

$$[\![\varphi]\!](u) = \inf \{n \in \mathbb{N} : (u, n) \models \varphi\}.$$

That is to say, we take the minimal value for N which makes the formula true on the input word u. This value will contain information about quantitative behaviours in u, since it is the threshold for u to be accepted by φ.

Example 3.2

- *With φ and u from Example 3.1, we have $[\![\varphi]\!](u) = 4$. The function $[\![\varphi]\!]$ maps to any word $u = v_1 b v_2 b \dots b v_k$, where each v_i does not contain any b, the value* $\max_{1 \le i \le k} |v_k|_c$.
- *On alphabet $\mathbb{A} = \{a, b\}$, we can represent the function f counting the number of occurences of a in the word, with both CLTL and CFO: we have $f = [\![\mathbf{G}^{\le N} b]\!] = [\![\forall^{\le N} x\ b(x)]\!]$.*

It is also important to notice that if φ is a classical formula for a language L, then we can push negations to the leaves, and interpret φ as a cost logic formula. It is easy to verify that $[\![\varphi]\!] = \chi_L$. Therefore, classical formulas and languages are particular cases of cost functions, and all results we will show inthe framework of cost functions can be in particular interpreted as theorem on languages.

Aperiodic stabilisation monoids

Although we will not describe in detail the definition of stabilisation monoids here, we give a brief description of the idea behind this model. Stabilisation monoids are classical monoids extended with an operator $\sharp$, acting on idempotents (i.e. elements e such that $e \cdot e = e$). This operator intuitively describes the effect of "repeating a lot of times" an element. In a nutshell, a stabilisation monoid is a monoid with additional structure describing quantitative behaviour.

Describing how these stabilisation monoids can recognize cost functions is a lot harder than in the classical case (i.e. correspondance between monoids and languages). For instance unlike in the case of language, we need deep combinatorics theorems to give a semantic to stabilisation monoids. So we do not presetn further this part of the theory.

CFO and CLTL are strictly less expressive than regular cost functions. Hence we restrict the power of stabilisation monoids to match their expressivity. We take here the same definition as in the classical case: a stabilisation monoid M is **aperiodic** if there exists $n \in \mathbb{N}$ such that for all $x \in M$, we have $x^n = x^{n+1}$. This means the monoid cannot contain any nontrivial group.

Myhill-Nerode Equivalence

Myhill-Nerode equivalence is a key tool in language theory. It allows to obtain an algebraic description of a language, starting from any other description. The generalisation to cost function was not available and is one of the contributions of the thesis.

Let us remind the definition of the classical Myhill-Nerode equivalence for a language L: we have $u \sim_L v$ if for all words x and y,

$$xuy \in L \Leftrightarrow xvy \in L.$$

Then the **syntactic monoid** of L is given by the quotient $\mathbb{A}^*/\sim_L$. This monoid is the smallest possible recognizing L, and divides any monoid recognizing L. When generalising to cost functions, the main obstacle was that quotienting the set of words cannot work anymore.

The following table sums up the concepts needed to define the generalised equivalence relation:

Languages	**Cost functions**
Words $\mathbb{A}^*$	$\sharp$-expressions $\mathbb{A}^{*\sharp}$ (example : $(b^\sharp a)^\sharp$)
Word with a hole aab_bbaa	Context $(\_^\sharp a)^\sharp$
Membership of a word in L	Boundedness of f on a sequence of words
Equivalence $\sim_L$ over $\mathbb{A}^*$	Equivalence $\sim_f$ over $\mathbb{A}^{*\sharp}$
$\mathbb{A}^*/\sim_L$ syntactic monoid	$\mathbb{A}^{*\sharp}/\sim_f$ syntactic stabilisation monoid

Some additional difficulties occur: for instance we have to restrict $\sharp$-expressions to the well-formed ones, relatively to the cost function f. Moreover, this relation cannot be properly defined if f is not regular, which was not the case with languages. It turns out that some of the behaviours of cost functions on finite words are more reminiscent of ω-languages than of finitary languages. For instance the analog of well-formed $\sharp$-expressions would be ultimately periodic words.

Algebraic characterization of CLTL fragment

We now have an abstract description of the syntactic monoid for any regular cost function. Therefore, it is possible to show, by induction on the structure of a CLTL-formula φ, that $[\![\varphi]\!]$ is recognized by an aperiodic stabilisation monoid. This part still poses technical difficulties, because we have to consider the behaviours of CLTL-formulae on sequences of words generated by $\sharp$-expressions, and in particular find some uniformity in these behaviours.

Conversely, we generalized a proof from [Wil99] to go from an aperiodic stabilization monoid M to a CLTL-formula φ. The proof goes by induction on $(|M|, |\mathbb{A}|)$. If M contains an element $b \neq 1$, we show that we can use the induction hypothesis on the stabilization monoid $\langle Mb \cap bM, \circ, \natural\rangle$, which is aperiodic and stictly smaller than M. Again, we have to deal with difficulties inherent to the quantitative setting, in addition to the technicalities of the original proof.

Finally, we obtain the following theorem:

Theorem 3.3 *For a regular cost function f, it is equivalent to be recognized by*

- *a CFO formula,*

- *a CLTL formula,*
- *aperiodic stabilisation monoid.*

As a consequence, it is decidable whether a regular cost function f can be described by a CLTL-formula: we can compute the minimal stabilization monoid for f, and test for aperiodicity.

Other results on CLTL

Since LTL is a formalism used in practice for verification purpose, we are interested in the computational complexity properties of CLTL. By generalizing the classical approach, we can obtain the following results:

Theorem 3.4 *Let φ be a CLTL-formula. We can build B- and S-automata (of size exponential in $|\varphi|$) for $[\![\varphi]\!]$. The boundedness problem for $[\![\varphi]\!]$ is PSPACE-complete.*

It is interesting to notice that generalizing from LTL to CLTL induces no cost in terms of computational complexity, for the satisfiability problem (generalised to boundedness in the framework of cost functions) .

3.2 Temporal class

Another class of regular cost functions is studied: **temporal cost functions**. Unlike the aperiodic class, this class is specific to cost functions in the sense that it does not generalize a class of languages. 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 and Bala [Kir04, Bal04]. This class can be 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. We show that for this class, it is enough for B- or S-automata to have one counter to reach their full expressive power (this is not the case for general regular cost functions).

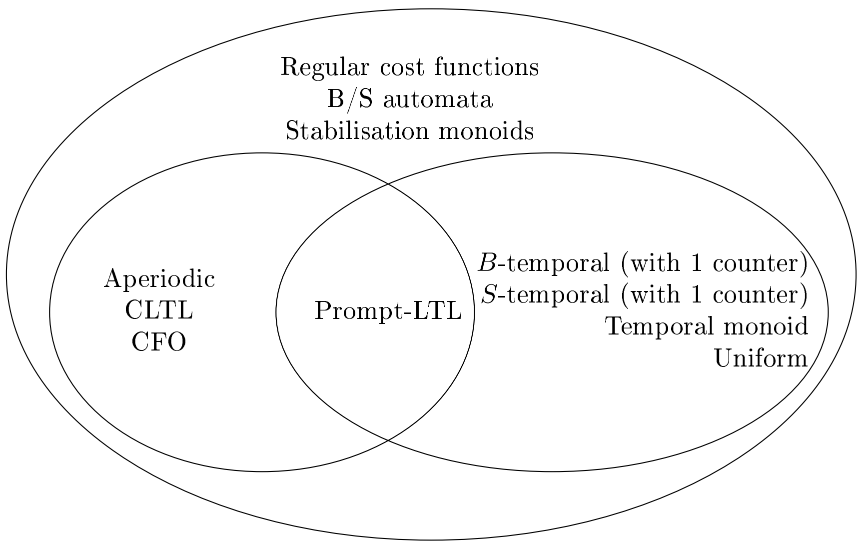
Second, we provide an equivalent *clock-based* presentation, in which the temporal cost function is represented as a regular language over words (called **Uniform** representation) labeled with the ticks of a clock as an extra information. We show all the closure results for regular temporal cost functions (e.g., min, max, etc...) using this presentation. As opposed to the general theory of regular cost functions, all these results are obtained by a translation to the theory of regular languages. These constructions are of better complexity, both in terms of number of states of automata, and in terms of technicality of the constructions themselves.

Like in the aperiodic case, we show an algebraic characterization of the temporal class. Since the corresponding property of stabilization monoids can be effectively tested, we obtain decidability of membership in the temporal class, for arbitrary regular cost functions as input.

The big picture

Finally, we take a look at the intersection between the aperiodic class and the temporal class. We show that this intersection is characterized by the logic **Prompt-LTL**. This logic has been introduced independently in a context of verification [KPV09] with the idea of bounding time intervals, but it can be seen as a restriction of CLTL where the operator $\varphi \mathbf{U}^{\leq N} \psi$ is replaced by $\bot \mathbf{U}^{\leq N} \psi$.

Some of the results on finite words can be summarized by the following picture:



4 Contributions: Infinite words

Cost functions can be defined on infinite words in the same way as for finite words. Automata now have **infinitary condition**, like **Büchi** (accepting states must be seen infinitely often in an accepting run) or **Parity** (each state has an integer **rank**, and the higher rank appearing infinitely often in an accepting run must be even).

4.1 Regular cost functions on infinite words

Since switching between B- and S-automata is the cost function analog of complementation, the Büchi complementation theorem (stating that the class of

ω-languages recognizable by nondeterministic Büchi automata is closed under complement) is generalized in the following way:

Theorem 4.1 *On infinite words, B-Büchi automata and S-Büchi automata are effectively equivalent (in terms of recognized cost functions). Cost functions recognized by such automata will be called* **regular**. *Equivalence and boundedness are decidable for regular cost functions on infinite words.*

We are also interested in a special form of automata: **weak B-alternating automata**. The weakness describe a condition on the structure of the automata: every strongly connected component in its transition graph is either accepting or rejecting. Therefore, any infinite play of such an automaton stabilizes either in accepting states, or in rejecting states. The semantic of such automata is defined as a game between two players, one trying to minimize the counter value while satisfying the acceptance condition, and the other trying to maximize the counter value, or to force a failure of the acceptance condition. This model has been introduced in [VB11] in the more general context on infinite trees.

On the logical side, let **CMSO** (for Cost Monadic Second-Order logic) be CFO extended with quantification over set of positions. We also define the weak variant **WCMSO** where second-order quantification is restricted to **finite sets**. This last logic is equivalent to the model of weak B-alternating automata [VB11]. In the classical setting, weak alternating automata and weak MSO have received a lot of attention, because these formalisms offer a good trade-off between expressivity and simplicity of computational properties and procedures (like complementation or model-checking).

We show the following theorem:

Theorem 4.2 *For a cost function on infinite words, it is equivalent to be*

- *regular,*
- *recognized by a CMSO formula,*
- *recognized by a WCMSO formula,*
- *recognized by a weak B-alternating automaton.*

In the classical setting, equivalence between MSO and Weak MSO is usually proved by using **deterministic parity automata**. Such automata do not always exist in the framework of cost functions (deterministic B-automata are strictly weaker than nondeterministic ones), so we have to follow another route to prove this theorem. We first show a **normal form** for B-Büchi automata: for all regular cost function, there is a B-Büchi automaton recognizing it, such that all transitions leaving Büchi states perform a reset of all counters. This allows to simplify the interaction between counter behaviour and acceptance condition, and to generalize a proof from Kupferman and Vardi [KV01], going from nondeterministic Büchi automata to weak alternating automata.

4.2 Kamp Theorem

We continued the study of CLTL, this time on infinite words. Kamp Theorem states that FO and LTL have same expressive power in terms of recognized language.

The corresponding theorem on cost functions has been shown in the thesis:

Theorem 4.3 *For a regular cost function on infinite words, it is equivalent to be recognized by*

- *CFO-formula,*
- *CLTL-formula,*
- *CLTL-formula with both future and past operators.*

Going from CLTL-formula to CFO-formula is straightforward: it suffices to express all operators of CLTL by means of CFO-formulas.

The converse is a lot harder, and is proved here by generalizing a proof in [GPSS80], which starts by enriching temporal logic with past operators. One key argument of the proof is to show **separability** of such formulas, i.e. that any formula with future and past operators can be transformed into an equivalent formula which is a boolean combination of past formulas and future formulas. The technical difficulty of this lemma is increased in the framework of cost functions, because quantitative behaviours interact with temporal operators in unexpected ways.

5 Contributions: Infinite trees

All models of cost automata on infinite words can be generalized to infinite trees. That is, automata now take as input infinite binary trees, with their nodes labelled by letters from $\mathbb{A}$. We will not detail here the precise definitions of such automata. As before, they come in several variants: B or S, nondeterministic or alternating, with acceptance condition Weak, Büchi or Parity.

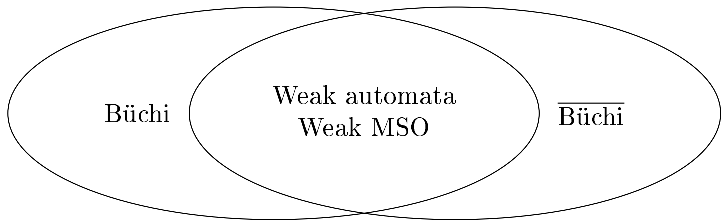
This new framework brings several new difficulties, for instance witnessed by the fact that decidability of the boundedness problem for B-Büchi automata is still open on infinite trees.

The starting point of the work on infinite trees in the thesis was to generalize the following theorem:

Theorem 5.1 *[Rab70, KV99] Let L be a language of infinite trees. The following statements are equivalent:*

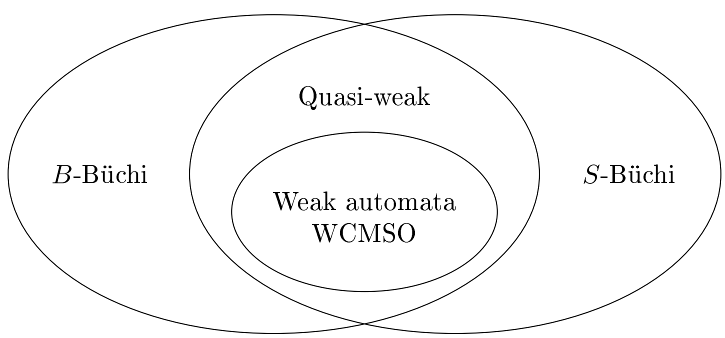
- *L is described by a weak MSO formula,*
- *L is recognized by a weak alternating automaton,*
- *Both L and its complement are recognized by nondeterministic Büchi automata.*

This can be summarized by the following picture:



The generalizations of these classes were already defined for cost functions: complementing corresponds to switching between B and S, and **weak cost functions**, i.e. cost functions definable by WCMSO or weak B-alternating automata equivalently, were studied in [VB11]. Therefore, we conjectured (in joint work with Michael Vanden Boom) that this picture could be lifted to cost functions, i.e. that cost functions definable by both nondeterministic B- and S-Büchi automata were exactly weak cost functions.

But it turns out that the class lying at the intersection of B-Büchi and S-Büchi strictly contains the weak class. Indeed, this new class, that we call **quasi-weak** is specific to cost functions (since its restriction to languages coincides with the weak class, by the above theorem). As for the weak class, quasi-weak cost functions can be defined by a syntactical restriction on automata recognizing them: a quasi-weak automaton is an alternating B-Büchi automaton, such that any cycle which contains both accepting states and rejecting states must increment one of the counter without resetting it.



To establish this picture, we have to show in particular that if a cost function f is definable by both non-deterministic B- and S-Büchi automata, then it is definable by a quasi-weak automaton. The proof of this fact is very technical and new kind of objects and notions were introduced to tackle it. In particular, we introduce cost tree automata equipped with both kind of counters: B and S. With suitable semantics, we show that is is possible to obtain a **normal form** for these automata, were counters behave with respect to a global hierarchy.

This allows us to generalize the proof from [KV99], where a weak alternating automaton counted "blocks" on two simultaneous runs of Büchi automata, in order to build a "trap" leading to a contradiction. We have generalized to the cost setting the notions of blocks and trap, and the new definitions can yield a proof, provided we first convert the product of the B- and S-Büchi automaton to its equivalent normal form described above.

The other way around is obtained by studying the shape of **strategies in quasi-weak B-games**, and to convert these games into different ones where classical results can be applied. From this, we show that is is possible to simulate a quasi-weak automaton by a non-deterministic B- or S-Büchi one. The simulation by S-automata allows us to describe a decision procedure for boundedness of cost functions definable by quasi-weak automata. In fact, if f and g are defined by quasi-weak automata, we show that $f \approx g$ is decidable. This class of cost functions is currently the largest one such that the boundedness (or equivalence) problem is known to be decidable on infinite trees.

Back to languages

The **Mostowski index problem** is a difficult open problem about regular language on infinite trees: given a language L and an interval of integers $[i, j]$, is there a non-deterministic parity automaton for L using parity ranks $[i, j]$?

The most significant advance on this problem has been made by Niwinski and Walukiewicz [NW05]: they showed that the problem is decidable if L is given by a deterministic tree automaton. Since regular tree languages cannot always be recognized by deterministic automata, this is not enough to answer the general question.

A special case of this problem, which has drawn interest, is the weak definability problem: given a regular tree language L, can we decide whether L is weak?

By using quasi-weak automata, we showed that a special case of this problem can be solved: if the input automaton is Büchi, then it is decidable whether the language is weak.

This is done by reducing this problem to the boundedness of quasi-weak B-automata, which we showed decidable.

Colcombet and Löding showed [CL08] that for any $[i, j]$, there is a reduction from $[i, j]$-definability to boundedness of $[i, j]$-parity B-automata, but this last problem remains open, except for intervals $[2n, 2n + 1]$ (corresponding to co-Büchi condition).

6 Conclusion

The theory of regular cost functions is a new and promising field of research in automata theory. This theory at the same time extends many key classical language theoretic theorems, and provides new techniques and decidability results. In this thesis, several new results for cost functions are developed. These go in several directions:

- The central notion of the **syntactic monoid** of a language, was generalized to the framework of cost functions. This object plays the same important rôle for cost functions.

- Following the long line of research in automata theory devoted to the analysis of fragments of regular languages, several classes of regular cost functions have been **effectively characterized**. This includes a generalization of the deep **Schützenberger-McNaughton-Papert-Kamp theorem** for languages, as well as the characterization of new classes, such as **temporal cost functions**.

- Over infinite words and trees, the relationship between between cost monadic and **weak cost monadic logic** is analysed. On infinite words, once more, these two logics are of same expressive power, but the proofs are radically different from the language case. On infinite trees, these two logic differ, as for languages, but this time, the result of Rabin stating that weak monadic logic defines exactly the languages that are both Büchi and of Büchi complement fails for cost functions. This motivates the introduction of a new class of regular cost functions over infinite trees, **quasi-weak cost functions** that corresponds to the characterization of Rabin.

- These last results over infinite trees paved the way to the effective decision of some classes of the **Mostowski hierarchy**.

Overall these results showed that a large part of automata theory can be raised to the level of cost functions. This is obtained at the price of introducing many new ideas, and yields new insights, even for classical automata theory itself.

Several aspects of this thesis were particularly challenging, for different reasons:

- The objects manipulated sometimes become a lot more complicated than their classical analogs (for instance the semantics of stabilisation monoids, or automata on infinite trees).

- Some of the theorems that are generalized were already technically challenging in the classical theory. Since cost function theory embeds language theory, the proofs described in the thesis must contain in particular classical proofs for these hard theorems.

- A lot of new intuitions had to be developed to be able to understand deeply the different objects of cost functions theory. The interplay between quantitative behaviours and other constraints raised new problems, that could not be solved with ideas taken from existing proofs. The technical machineries that were required to tackle these difficulties often were completely specific to cost functions, with no analog from language theory.

7 Publications relative to the thesis

The publications corresponding to the thesis are the following:

1 Regular temporal cost functions, with Thomas Colcombet and Sylvain Lombardy, ICALP 2010 [CKL10]

2 Linear temporal logic for regular cost functions, STACS 2011 [Kup11]

3 Quasi-weak cost automata: a new variant of weakness, with Michael Vanden Boom, FST&TCS 2011 [KV11].

4 On the expressive power of cost logics over infinite words, with Michael Vanden Boom, ICALP 2012 [KV12].

Several paper are also resulting from the work of the thesis.

5 Linear temporal logic for regular cost functions, Submitted to *Logical Methods in Computer Science*.

6 Deciding the weak definability of Büchi definable tree language, with Thomas Colcombet, Christof Löding and Michael Vanden Boom, to be submitted to CSL.

6 Two-Way Cost Automata and Cost Logics over Infinite Trees, with Achim Blumensath, Thomas Colcombet, and Michael Vanden Boom, to be submitted.

References

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

[BC06] Mikołaj Bojańczyk and Thomas Colcombet. Bounds in ω-regularity. In *Proceedings of LICS 2006*, pages 285–296. IEEE Computer Society Press, 2006.

[BOW09] Achim Blumensath, Martin Otto, and Mark Weyer. Boundedness of monadic second-order formulae over finite words. In *Automata, languages and programming. Part II*, volume 5556 of *Lecture Notes in Comput. Sci.*, pages 67–78, Berlin, 2009. Springer.

[CKL10] Thomas Colcombet, Denis Kuperberg, and Sylvain Lombardy. Regular temporal cost functions. In *Automata, languages and programming. Part II*, volume 6199 of *Lecture Notes in Comput. Sci.*, pages 563–574. Springer, Berlin, 2010.

[CL08] Thomas Colcombet and Christof Löding. The non-deterministic Mostowski hierarchy and distance-parity automata. In *Automata, languages and programming. Part II*, volume 5126 of *Lecture Notes in Comput. Sci.*, pages 398–409. Springer, Berlin, 2008.

[CL10] Thomas Colcombet and Christof Löding. Regular cost functions over finite trees. In *25th Annual IEEE Symposium on Logic in Computer Science LICS 2010*, pages 70–79. IEEE Computer Soc., Los Alamitos, CA, 2010.

[Col09] Thomas Colcombet. The theory of stabilisation monoids and regular cost functions. In *Automata, languages and programming. Part II*, volume 5556 of *Lecture Notes in Comput. Sci.*, pages 139–150, Berlin, 2009. Springer.

[GHR94] Dov M. Gabbay, Ian Hodkinson, and Mark Reynolds. *Temporal logic. Vol. 1*, volume 28 of *Oxford Logic Guides*. The Clarendon Press Oxford University Press, New York, 1994. Mathematical foundations and computational aspects, Oxford Science Publications.

[GPSS80] Dov M. Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi. On the temporal analysis of fairness. In *POPL*, pages 163–173, 1980.

[Has79] Kosaburo Hashiguchi. A decision procedure for the order of regular events. *Theoret. Comput. Sci.*, 8(1):69–72, 1979.

[Has82] Kosaburo Hashiguchi. Limitedness theorem on finite automata with distance functions. *J. Comput. Syst. Sci.*, 24(2):233–244, 1982.

[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.

[Kam68] Hans W. Kamp. *Tense Logic and the Theory of Linear Order*. Phd thesis, University of Warsaw, 1968.

[Kir04] Daniel Kirsten. Desert automata and the finite substitution problem (extended abstract). In *STACS 2004*, volume 2996 of *Lecture Notes in Comput. Sci.*, pages 305–316. Springer, Berlin, 2004.

[Kir05] Daniel Kirsten. Distance desert automata and the star height problem. *Theor. Inform. Appl.*, 39(3):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.

[Kro94] Daniel Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *International Journal of Algebra and Computation*, 4(3):405–425, 1994.

[Kup11] Denis Kuperberg. Linear temporal logic for regular cost functions. In *28th International Symposium on Theoretical Aspects of Computer Science*, volume 9 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 627–636. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2011.

[KV99] Orna Kupferman and Moshe Y. Vardi. The weakness of self-complementation. In *STACS 99 (Trier)*, volume 1563 of *Lecture Notes in Comput. Sci.*, pages 455–466. Springer, Berlin, 1999.

[KV01] Orna Kupferman and Moshe Y. Vardi. Weak alternating automata are not that weak. *ACM Trans. Comput. Log.*, 2(3):408–429, 2001.

[KV11] Denis Kuperberg and Michael Vanden Boom. Quasi-weak cost automata: a new variant of weakness. In *31st International Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 13 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 66–77. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2011.

[KV12] Denis Kuperberg and Michael Vanden Boom. On the expressive power of cost logics over infinite words. In *ICALP (2)*, pages 287–298, 2012.

[MP71] R. McNaughton and S. Papert. *Counter-free automata*. M.I.T. Press research monographs. M.I.T. Press, 1971.

[NW05] Damian Niwiński and Igor Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. In *Proceedings of the 11th Workshop on Logic, Language, Information and Computation (WoLLIC 2004)*, volume 123 of *Electron. Notes Theor. Comput. Sci.*, pages 195–208 (electronic), Amsterdam, 2005. Elsevier.

[Rab69] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141:1–35, 1969.

[Rab70] Michael O. Rabin. Weakly definable relations and special automata. In *Mathematical Logic and Foundations of Set Theory (Proc. Internat. Colloq., Jerusalem, 1968)*, pages 1–23. North-Holland, Amsterdam, 1970.

[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 *19th Annual Symposium on Foundations of Computer Science (Ann Arbor, Mich., 1978)*, pages 143–150. IEEE, Long Beach, Calif., 1978.

[VB11] Michael Vanden Boom. Weak cost monadic logic over infinite trees. In *Mathematical foundations of computer science 2011*, volume 6907 of *Lecture Notes in Comput. Sci.*, pages 580–591. Springer, Heidelberg, 2011.

[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.