AAPG2024	UNREAL	Funding instrument: **PRME**	
Coordinated by:	Marc Zeitoun	Duration: **5 years**	104.8K€
Axe E.01. Fondements du numérique : informatique, automatique, traitement du signal et des images			

UNREAL: UNderstanding Regular Expressions, Automata and Logics

Summary table of persons involved in the project

Partner	Name	First name	Current position	Role & responsibilities	Involvement
Bordeaux	Place	Thomas	Assistant professor	Member, all tasks	54 p.month
Bordeaux	Zeitoun	Marc	Professor	Coordinator, all tasks	54 p.month
Bordeaux			Post-doctoral student	Member, task 2	12 p.month

Any changes made in the full proposal compared to the pre-proposal

The scientific proposal for this project remains the same. However, we have increased the requested budget by 16.4% compared with the pre-proposal: instead of 90000, the request is for 104800€. The first reason is technical: we had not anticipated the amount collected by our managing university. The second reason is that, over the five years of the project, we aim for a dissemination of research results to young researchers that is more ambitious than initially planned.

I. Proposal's context, positioning and objectives

This project is motivated by open questions for *regular languages of words and trees*. **Regularity** is a fundamental and robust notion. Indeed, regular languages are, equivalently, those accepted by finite automata, those described by regular expressions or those defined by sentences in monadic second-order logic **MSO**. Our goal is to *understand classes* of such languages. As in most areas of theoretical computer science, progress in this field often stems from *challenging questions*. Many of them, though long-standing, are still open and still being investigated in their original formulation, which testifies to both their difficulty and their importance. They form the backbone on which the research domain develops, and some of these questions form the motivation of this project.

One of the main objectives in this area is to **understand the expressive power** of logical formalisms that are *weaker* than monadic second-order logic. Since this logic captures exactly regular languages, any fragment of **MSO** defines a subclass of that of regular languages. Since the 1960s, research has focused on *first-order logic* (a yardstick formalism in mathematics) and the levels of its **quantifier alternation hierarchy** [BC71, Tho82, Str81, Thé81]. Let us briefly recall what these notions mean in this context. A word made of n letters is seen as a sequence of positions $1, 2, \ldots, n$, each position being labeled by some letter. Thus, it is possible to write first-order sentences that describe properties of a word. A sentence can quantify over the positions in the sequence and check properties of these quantified positions using a predetermined set of predicates. Thus, a first-order sentence defines the language of all words satisfying it. The simplest variant, denoted by **FO**(<), can only test the letter at a position, and compare two positions with the (strict) order predicate "<". For instance, the sentence,

$$\exists x \exists y\ a(x) \wedge b(y) \wedge (x < y)$$

states that there exists a position x carrying an "a" followed by a position carrying a "b". Therefore, the language of words that satisfy this sentence is given by the regular expression $A^* a A^* b A^*$ where A is the underlying alphabet. *Quantifier alternation* is used as a natural complexity measure for first-order logic.

One classifies the sentences into levels $\mathcal{B}\Sigma_n$, which consist in all sentences having at most n quantifier blocks of the form $\exists^*$ or $\forall^*$. For instance, the above sentence belongs to $\mathcal{B}\Sigma_1(<)$.

Quantifier alternation hierarchies correspond to a fundamental concept in automata and language theory: *concatenation hierarchies*. These hierarchies, rooted in regular expressions, have a uniform construction process. Each concatenation hierarchy originates from a unique base class of regular languages, serving as its level zero. Each subsequent level n is constructed by applying the operator $\mathcal{C} \mapsto \mathrm{BPol}(\mathcal{C})$ to the previous level $n-1$: given a class of regular languages $\mathcal{C}$, it generates an expanded class $\mathrm{BPol}(\mathcal{C})$, comprising all Boolean combinations of languages $L_0 a_1 L_1 \cdots a_k L_k$ where $L_1, \ldots, L_k$ are languages in $\mathcal{C}$. Additional "half levels" are also considered, constructed with a weaker operator Pol which does not involve Boolean combinations. A prime illustration is the Straubing-Thérien hierarchy [Str81, Thé81] whose basis is $\mathrm{ST} = \{\emptyset, A^*\}$. Perrin and Pin [PP86] proved that the languages at level n in this hierarchy precisely align with those defined by a formula at level $\mathcal{B}\Sigma_n(<)$ is the alternation hierarchy of $\mathbf{FO}(<)$. This extends an earlier result by Thomas [Tho82] who showed that the dot-depth hierarchy [BC71] of basis $\mathrm{DD} = \{\emptyset, \{\varepsilon\}, A^+, A^*\}$ corresponds to the alternation hierarchy of $\mathbf{FO}(<, +\mathbf{1})$, with predicates extended to include the successor "+1". These two hierarchies stand out as the most prominent in literature, and the breakthroughs in this field are frequently linked to them. Nonetheless, many more natural base classes have been considered (see *e.g.* [MP85, BCST92, Pin98, CPS06, KW15, PZ19d]). In particular, the connection with logic is generic: for each base class, there exists a distinct set of predicates such that the quantifier alternation hierarchy of first-order logic, when equipped with these predicates, aligns with concatenation hierarchy of basis $\mathcal{C}$ [PZ19a].

Now, let us delve deeper into what we meant initially by "*understanding classes of regular languages*". Over the past six decades, pivotal inquiries have revolved around two fundamental decision problems: **membership** and **separation**. They depend on a fixed class $\mathcal{C}$ under investigation. Membership asks for an algorithm deciding whether an input regular language belongs to $\mathcal{C}$. Separation is more general. It asks for an algorithm that, given two input languages K and L, tests whether there exists a language in $\mathcal{C}$ containing K that does not intersect L. These questions pose significant challenges, directly tied to our motivation. Beyond the algorithms themselves, the driving force behind this approach lies in the deep insight on $\mathcal{C}$ necessary to surmount the obstacles inherent in designing and validating them.

Specifically, we are interested in solving this problem for levels of alternation hierarchies. Unfortunately, after decades of attempts, we can only understand the very first levels of such hierarchies. In particular, we still do not know how to decide whether an input language belongs to a given level. In this research field, the **question/breakthrough** milestones are as follows:

- **Question 1**: *Decide whether a language can be described by a first-order sentence.*
  Breakthrough 1: [Sch65, MP71] Membership algorithm of first-order definable languages.
- **Question 2**: *Decide membership for all levels of the quantifier alternation hierarchy.*
  Breakthrough 2a: [Sim75, Kna83] Membership algorithms for level one in the Straubing-Thérien hierarchy and the dot-depth hierarchy.
  Breakthrough 2b: [Str85] The Straubing-Thérien hierarchy is more fundamental: membership for a level in the dot-depth hierarchy reduces to the corresponding level in the Straubing-Thérien hierarchy.
  Breakthrough 2c: [PW97] Generic investigation of the operator BPol used in the construction. Membership for level $\frac{3}{2}$ in the Straubing-Thérien hierarchy.
- **Question 3**: *Investigate problems that are more general than membership, such as "separation".*
  Breakthrough 3a: [PZ14, Pla15, PvRZ13, CMM13] *Separation* algorithms for levels $\frac{1}{2}$, one, $\frac{3}{2}$ and $\frac{5}{2}$ of the Straubing-Thérien hierarchy. These algorithms are then lifted via transfer results to decide *membership* for levels two and $\frac{7}{2}$.
  Breakthrough 3b: [PZ16] Design of a clean mathematical framework to solve separation.

While the first question found resolution long ago, the second remains *wide open*. Our overarching aspiration is to decide membership for all levels of the quantifier alternation hierarchy, but this goal appears elusive in the near future. Question 3 was raised partly in anticipation of addressing Question 2. While this endeavor met with partial success, it now appears unlikely that further headway can be achieved solely by advancing existing techniques. *New approaches are imperative for gaining deeper insights into alternation hierarchies.*

I.a. Objectives and research hypothesis

New problems, particularly separation, lie at the core of this research project. Our objectives are shaped by these problems and are built upon them. Let us first examine the reasons behind this. The motivation for delving into separation and its broader forms, such as the covering problem discussed in [PZ18], is twofold. Firstly, despite being more complex, creating an algorithm that tackles separation or covering for a language class $\mathcal{C}$ is ultimately more fulfilling regarding our main goal: gaining insights into $\mathcal{C}$. In fact, this aspect is what initially drove the exploration of separation in the early 2010s. Secondly, exploring these problems has catalyzed significant progress in understanding the simpler membership problem within concatenation hierarchies [PZ14]. On the surface, these two assertions might appear contradictory:

1. Although more rewarding, tackling more general problems is also considerably more challenging than addressing membership alone.

2. However, it is instrumental in resolving many of the already complex membership questions.

Let us reconcile the apparent contradiction. We have developed a clear framework for formalizing separation and its extension, the covering problem [PZ18]. This framework has facilitated recent advancements by significantly simplifying challenging proofs. It enables us to achieve results that are more comprehensive than previous ones and produce proofs that are *generic*.

This genericity has shifted the focus from studying *individual classes* to examining *operators*. We discussed this notion earlier when introducing concatenation hierarchies, constructed using the operators Pol (polynomial closure) and BPol (Boolean polynomial closure). An operator "Op" takes an arbitrary class of languages $\mathcal{C}$ as input, and generates a larger one denoted $\mathrm{Op}(\mathcal{C})$ as output. Operators are fundamental concepts and while their definitions have evolved over the years, most of the natural and significant operators that we consider today were defined in the early days of automata theory, alongside concatenation hierarchies [BC71, Sch75, Str79, Pin80]. They emerge naturally when examining the principal classes of languages typically addressed in the literature. Though numerous, these classes can be categorized into families based on "variants" of the same syntax. For example, all concatenation hierarchies are uniformly constructed using the same syntactic process. Thus, each hierarchy can be viewed as a variant corresponding to a particular base class. This perspective also aligns in the logical point of view. First-order logic and its quantifier alternation hierarchy levels can be equipped with various distinct sets of predicates, giving rise to different classes. For instance, predicates such as the linear order "$<$" [MP71, Sch65], the successor "$+1$" [BP91] or the modular predicates "*MOD*" [BCST92] are commonly encountered. While examining multiple variants of prominent classes is valuable, doing so individually for each presents a disadvantage: the argument must be systematically adjusted to accommodate each change. This can be tedious, difficult, and not necessarily enlightening. To overcome this limitation, a natural approach is to encapsulate an entire family of variants with an operator, enabling the simultaneous study of all classes $\mathrm{Op}(\mathcal{C})$. Thus, the question arises:

"What hypotheses about $\mathcal{C}$ guarantee the decidability of $\mathrm{Op}(\mathcal{C})$-membership?".

The aspiration is that by working with operators instead of specific classes, we can achieve unified proofs for several variants of a class. This approach is particularly fitting when examining concatenation hierarchies, which are constructed by iteratively applying a single operator BPol.

Recently, this approach has garnered renewed interest as the exploration of separation and its extensions has provided solutions to the aforementioned question for specific operators. Specifically, it has been demonstrated that for several operators Op, *membership* for the resulting class $\mathrm{Op}(\mathcal{C})$ boils down to *separation* for the input class $\mathcal{C}$. For instance, this holds true for polynomial closure [PZ19a] (Pol), utilized in constructing concatenation hierarchies, and for star-free closure [PZ19c] (SF), which forms the union of all levels within a concatenation hierarchy. In fact, most recent advancements on this matter for finite words are grounded in such transfer theorems (see [PZ19b]). This resolves the previously noted apparent contradiction: one can leverage the more challenging separation problem for an input class $\mathcal{C}$ to address the simpler membership problem for the more complex class $\mathrm{Op}(\mathcal{C})$. Naturally, these findings prompt an immediate new question regarding operators:

"What hypotheses about $\mathcal{C}$ guarantee the decidability of $\mathrm{Op}(\mathcal{C})$-**separation**?".

This question is particularly significant in the context of concatenation hierarchies, where the construction process iteratively employs the BPol operator. Of course, it is even more challenging than the previous one. This leads us to the **two main scientific objectives** of the project, described in greater detail below:

1. The **first objective** is to study a novel problem, harder than separation (we introduce it below). We hope that this problem, or a variant of it, will help us to understand how to "solve" new levels in alternation hierarchies. Unlike membership and separation, it involves not one, but *two or more classes* of languages.
2. The **second objective** is to extend to more complicated *structures* the framework developed in the current answer to Question 3 (page 2), as well as the separation algorithms and the transfer results based on operators that were designed for finite words. We are particularly interested in infinite words and finite trees.

First objective

Our first objective is to tackle a **new problem** for finite words, with the anticipation of it leading to a *question/breakthrough* pair. While the framework is fairly well understood for finite words, it appears inadequate for resolving new levels of quantifier alternation hierarchies. This is what motivates our new question. We specifically propose the following one because it has emerged twice as a crucial element in two seemingly unrelated papers [Pla18, PZ22].

We call this new problem *"layered separation"*. Unlike all other problems we know, it depends on **several** classes. However, the question it raises is still simple to state. For the sake of simplicity, we present it here only for two classes:

Layered separation problem for classes $\mathcal{C}$ and $\mathcal{D}$

Given *three* regular languages L_0, L_1 and L_2, does there exist two languages $H \in \mathcal{C}$ and $K \in \mathcal{D}$ such that $L_0 \subseteq H$, $H \cap L_1 \subseteq K$ and $K \cap L_2 = \emptyset$?

Layered separation is more general than separation, which corresponds to the case $L_2 = \emptyset$. **The first scientific objective** proposed in this project, also constituting its first task, is to investigate this problem. Drawing from our prior experiences, we anticipate that this exploration can pave the way for significant advancements – although, as is customary in research, this remains somewhat speculative.

Objective 1

Investigate the layered separation problem and its consequences:

(a) First, for simple pairs of classes $\mathcal{C}$ and $\mathcal{D}$.

(b) Then, for levels of quantifier alternation hierarchies.

(c) Look for transfer results involving this problem.

Let us comment on points (b) and (c), and in particular what we mean by "transfer results" . As we explained above, the layered separation problem has already appeared in two independent papers, [Pla18] and [PZ22]. However, each of these articles concentrated on resolving the separation problem for a specific level of a concatenation hierarchy. Our aim here is to comprehend why it has emerged twice, not solely to unify proofs, but also to employ layered separation in a more versatile manner.

In both [Pla18] and [PZ22], it was used in connection with the question asked at the top of page 4: "What hypotheses about $\mathcal{C}$ guarantee the decidability of $\mathrm{Op}(\mathcal{C})$-separation?", for an operator Op involved in the construction of levels in concatenation hierarchies. This is the kind of desired transfer result we are after: certain properties of $\mathcal{C}$ *transfer* to decidability of $\mathrm{Op}(\mathcal{C})$-separation.

From our experience, we find it unlikely that decidability of $\mathcal{C}$-separation alone is sufficient condition to ensure the decidability of $\mathrm{Op}(\mathcal{C})$-separation, even in the specific case of the BPol operator. When $\mathcal{C}$ is a particular level of a concatenation hierarchy, we believe that one can replace the "hypotheses about $\mathcal{C}$" of our question, by the decidability of a layered separation problem, in which the classes involved (such as $\mathcal{C}$ and $\mathcal{D}$ in the statement of this problems) are $\mathcal{C}$ and the levels of the concatenation hierarchy lying below $\mathcal{C}$. Let us explain this intuition more precisely.

Why layered separation?

While this leans on the technical side, let us elaborate on why we believe this problem holds significance and has the potential to drive substantial progress. In a concatenation hierarchy, the levels are constructed through a layered *process*, iteratively applying the BPol operator. In practice, it appears that as one ascends to higher levels, scrutinizing each new level in isolation is an inadequate approach. Instead, we advocate for addressing each new level through a layered separation problem that considers this level *and all preceding ones simultaneously*.

This idea is actually quite intuitive. Typical separation algorithms rely on (least or greatest) fixpoint procedures. The premise here is that the additional information provided by the layered separation problem is necessary to carry out such fixpoint computations for higher levels of concatenation hierarchies. This is precisely what occurs in [Pla15]. In this paper, separation for level $\frac{5}{2}$ in the Straubing-Thérien hierarchy is addressed, utilizing an *ad-hoc* variant of the layered separation problem, where the classes $\mathcal{C}$ and $\mathcal{D}$ are the levels $\frac{3}{2}$ and $\frac{5}{2}$, respectively.

Despite the straightforward formulation of the layered separation problem, we find it challenging. However, given its occurrence in two distinct contexts, both linked to quantifier alternation hierarchies, we view it as a promising candidate for reexamining new questions regarding the separation problem for levels of quantifier alternation hierarchies. We are hopeful that this fresh perspective will spur new developments.

Second objective

The second question we want to address is whether the techniques that have been developed for finite words can be **pushed for other structures**. We have two of them in mind: *infinite words*, and *finite trees*.

Objective 2

Extend the results known on finite words to more general structures:

1. To **infinite words**: define relevant operators; lift the results on separation known for finite words; reduce natural problems for infinite words to corresponding ones on finite words.
2. To **finite trees**, where separation has never been looked at: investigate it for " simple" classes.

These two extensions are of different nature. Let us first explain why the first extension (generalizing the results from finite to infinite words) is interesting and realistic.

Why infinite words?

Historically, results for **infinite words** have consistently paralleled those established for their finite counterparts. While the setting of infinite words presents added complexity, there exist classic tools at our disposal [PP04]. For this reason, we think that this objective is both natural and reachable in the short term. In fact, we already made a short excursion in this setting [PPZ16] (see also [KW18], and [CvGM22] for an even more general context).

In fact, there are three distinct sub-objectives concerning infinite words:

- Address membership and separation for specific classes.
- Identify operators that generalize those existing for finite words to establish transfer results.
- Adapt the framework introduced in [PZ18] to the setting of infinite words.

There are two natural approaches to achieve the first two sub-objectives: either by leveraging reductions to already established results on finite words or by adapting existing proofs for finite words. A challenge lies in identifying which operators are relevant for infinite words, as such operators do not currently exist. However, first-order logic on infinite words is well-defined, and thus, so are quantifier alternation hierarchies. Yet, characterizing levels of such hierarchies is open, even for the simplest variant $\mathbf{FO}(<)$ and low levels. There is hope to adapt proof techniques that have proven successful for finite words to infinite words, as demonstrated in [PPZ16, KW18] in specific cases. As of now, there are no known generic reduction techniques. The third sub-objective is more ambitious. We have a clean and elegant framework in the context of finite words to investigate the covering problem [PZ18]. The aim of this third sub-objective is to extend this framework to infinite words.

Given that techniques for infinite words are frequently inspired by those for finite words, we are confident that these three sub-objectives can be explored in the short term. Either of these questions would serve as an **ideal topic for a post-doc** (in contrast to our first objective and the extension for trees, as discussed below).

Now, we shift our focus to the second extension and explain why we believe that certain tools developed for words can be effectively employed for finite trees.

Why trees?

Developing membership algorithms for languages of finite trees is widely recognized as a very challenging question. Such algorithms for trees are scarce, and there are almost no results on separation for trees. Even characterizing full first-order logic for trees appears to be beyond reach [Heu88]. Hence, proposing to tackle separation for trees may seem overly ambitious.

Nevertheless, we perceive the development of separation algorithms for tree languages as a natural progression within this project. Recent successes in membership achieved through this approach for finite words are applicable to classes closely related to the few classes already solved for trees (as opposed to complete first-order logic). To clarify, manageable classes of tree languages are those where specifying much about branching in the trees is limited. There exists a parallel with the classes of finite words that have been effectively managed: these are classes that prohibit Kleene's star. In essence, in both scenarios, manageable cases entail expressible properties utilizing "local" combinations of simple properties.

Therefore, we think that it is feasible to adapt some of the techniques that have demonstrated efficacy for finite words to finite trees. In essence, we hold the belief that this objective is realistic, meaning that separation techniques developed for words can be tailored to trees in cases where membership has already been resolved.

Third objective

Our third objective in the project is not scientific in nature. Its goal is to disseminate the results.

Objective 3

Disseminate the results gathered during the last 10 years in this area, in particular:

- regarding the latest developments,
- towards young scientists and students.

We propose several means to reach this objective, described below.

Research on the classification of regular languages has advanced significantly in recent years. Additionally, regular languages serve as the foundation of entire domains within fundamental computer science. Finite automata, in particular, inspire research in related fields like software verification and learning. Consequently, we believe that the topics explored in this project could captivate numerous young researchers, enriching their understanding and appreciation of regular language theory. We propose disseminating these latest developments throughout the community through two complementary avenues:

- Firstly, through the creation of educational resources: an extensive book on the topic and an open-source software package that implements the membership and separation algorithms.
- Secondly, by directly imparting the knowledge acquired over the past 10 years to the community via a summer or spring school tailored for young researchers, and a specialized workshop geared towards established researchers closely aligned with the domain of this project.

In Section I.3, we elaborate on these two objectives.

I.b. Position of the project as it relates to the state of the art

Let's reposition each scientific objective of this proposal in relation to the current state of the art:

- The first objective pertains to a novel problem, *layered separation*, which extends separation and has not been systematically studied before. As mentioned earlier, it has been encountered in two instances [Pla18, PZ22]. The aim is to generate new findings for specific levels in concatenation hierarchies.

  The current status regarding the Straubing-Thérien and dot-depth hierarchies is as follows: separation and membership are known to be decidable up to level two [PZ19b]. For other standard hierarchies, separation and membership are known to be decidable up to level one [PZ19d]. Leveraging layered separation, we aim to achieve decidability of membership for higher levels in these hierarchies.

- The first part of the second objective revolves around infinite words. In this domain, separation is known to be decidable for the quantifier alternation hierarchy of first-order logic over infinite words up to level one, and membership is decidable up to level two [PPZ16, KW18]. Furthermore, unlike the case of finite words, no comprehensive framework has been devised to explore separation for classes of languages of infinite words. As previously mentioned, the notion of an operator is absent.

- Regarding the latter part of the second objective, membership has been resolved for a few classes of tree languages [BW06, BS09, BS10, PS11, BSS12, BP12, PS16]. These classes will serve as natural candidates for investigating the separation problem.

I.c. Methodology and risk management

Let's outline how we will break down our three objectives into manageable tasks to achieve them. There is a preliminary task that corresponds to the project management. The three next tasks correspond to the scientific objectives 1 and 2, as we divide the second objective into two tasks focused on infinite words and finite trees. Lastly, we will tackle the dissemination objective through four tasks.

Except for the organization of the project (Task 0) and the extension of results to infinite words (Task 2), all tasks involve the two permanent members of the team, Thomas Place and Marc Zeitoun. Task 0 will be handled by Marc Zeitoun and Task 2 involves the two permanent members plus the post doctoral student that we intend to hire.

Preliminary task: organization of the project

Task 0: Management of the project

The primary aim of this task is to oversee the project comprehensively, encompassing administrative, financial, and scientific aspects. The project coordinator is tasked with operational management responsibilities.

Task for Objective 1: layered separation

Task 1: Investigating layered separation

This task involves exploring the layered separation problem and its implications. As detailed earlier (under "Why layered separation"), the objective is to gather concurrent insights across various levels of concatenation hierarchies to derive "transfer result" in the following format:

If level n fulfills a certain condition, then level $n+1$ has decidable separation.

Tasks for Objective 2: extension to infinite words and finite trees

Task 2: Extending results from finite words to infinite words

This second task pertains to the first intended expansion towards infinite words. The objectives are to extend the existing separation results from finite words, either through reductions or by adapting techniques tailored for finite words. Additionally, we aim to construct a specialized mathematical framework dedicated to separation, similar to what was accomplished for finite words.

As previously discussed, this task presents an **excellent opportunity for a young researcher**. The rationale behind this is that algorithms for infinite words can draw inspiration from well-understood algorithms for finite words, providing a clear roadmap for investigation. The project would allow us to hire a post-doc to investigate these questions.

Task 3: Extending results from finite words to finite trees

This task aims to explore separation within classes of finite trees where membership algorithms are available, as discussed above.

Objective 3: Knowledge dissemination

Our third objective focuses on knowledge dissemination and comprises four distinct tasks. Two tasks involve creating educational resources:

- a book on one hand,
- an open-source software on the other hand.

The other two tasks aim to share acquired knowledge with the community:

- on one hand, to young researchers through a summer or spring school,
- on the other hand, to established researchers close to the domain of this project, through a more specialized workshop.

While presented separately, these tasks are interconnected and mutually reinforcing. For instance, algorithms outlined in the book will be directly integrated into the software, and some of the book's content will serve as foundational material for both the summer school and the specialized workshop.

Task 4: Finalizing and publishing a textbook

This tasks consists of finalizing a textbook and publishing its first part. The book is already partly written and available[ab] in two versions:

- Full version: `https://mycore.core-cloud.net/index.php/s/RtaJjDpmGVjemXU`.
- First 13 chapters: `https://mycore.core-cloud.net/index.php/s/nnAOuzL9MmN67Oc`

[a] Password AutoBook, do not distribute.
[b] Browsers do not render the pdf of this book correctly: please download it if you want to read it.

We commenced the writing of this book a few years ago, with the backing of the ANR under the DeLTA project (ANR-16-CE40-0007). Presently, the book has made significant progress, comprising over 1000 pages across 7 parts and encompassing thirty chapters. Nevertheless, substantial effort is still required to render it ready for distribution and publication.

The book's structure is unlikely to undergo significant changes, but it does necessitate multiple proofreading passes to incorporate remarks, examples, exercises, etc., rectify typographical errors, and enhance language quality. Additionally, not all chapters have undergone the same level of proofreading: introductory chapters are more refined compared to subsequent ones. Moreover, the book's parts cover topics ranging in specialization: it begins with accessible content suitable for bachelor's or master's students, gradually progressing to material tailored for doctoral students or researchers in related fields after Chapter 14.

Due to these considerations, **Task 4** initial aim is to complete and publish the first 13 chapters only, totaling approximately 450 pages. This first volume would encompass three primary sections, excluding the introduction (which is yet to be written) and the mathematical preliminaries.

- A part introducing standard results on regular languages and their various definitions (*e.g.*, regular expressions, finite automata, MSO logic, or recognition by monoids). This section serves as a textbook for bachelor's and master's students.
- A part outlining the problems addressed in the book, such as the membership problem or the separation problem.
- A part detailing the most classical classes of regular languages studied to date, including star-free languages or piecewise testable languages. These latter sections are accessible to master's and PhD students.

Task 5: Developing an open source software on automata theory

This task consists in implementing all the membership and separation algorithms presented in the book, which is described in Task 4.

Presently, a multitude of algorithms tackle membership and separation problems for various classes of regular languages. While some algorithms are established classics, others have emerged more recently, especially in the domain of separation algorithms. Despite the availability of several software packages for manipulating finite automata, such as the Awali package in SageMath, we are aware of only one recent tool dedicated to the membership problem. Developed by Charles Paperman in Python, this tool is accessible via `https://gitlab.inria.fr/cpaperma/pysemigroup` and `https://paperman.name/semigroup/`. It facilitates testing regular language properties across various language classes, effectively resolving the membership problem for many classes. However, there are several areas where it could be improved:

- Currently, it solely manages predetermined classes of languages, lacking the capability to construct classes using operators and to address questions dependent on such constructed classes.
- It only addresses the membership problem.
- Certain algorithms fail to terminate even for relatively small input languages, contradicting the expectations set by their theoretical complexity.

Ideally, we seek a similar software capable of handling a class $\mathcal{C}$ of languages by:

- Determining if an input language belongs to class $\mathcal{C}$ and providing a proof if so.
- Establishing whether two input languages are separable by a language from a class $\mathcal{C}$, offering a proof if applicable, and potentially furnishing a separator if it's not excessively large.
- Conducting these computations in an optimized manner.

We have already developed a prototype of such software named **MeSCaL** (acronym for MEmbership and Separation for ClAsses of Languages), as a proof of concept. The prototype of this software can be accessed at `https://github.com/thomas-place/mescal`. Nonetheless, there are several areas in which this software can be enhanced:

1. Currently, only membership algorithms are implemented, and almost none are dedicated to separation. Consequently, we plan to incorporate the implementation of separation algorithms.
2. The software is presently in a prototype stage. Specifically, it was developed in a fixed environment (MacOS) and would benefit from being adapted to other operating systems, such as Linux. For instance, automaton visualization currently occurs directly in the terminal, utilizing a protocol specific to the iTerm2 terminal.
3. The software requires extensive testing.
4. If time permits, we intend to create a web interface. Nevertheless, this is not the primary focus of this task; our priority is to concentrate on developing algorithms.

We do not intend to hire an engineer for the software development tasks in this project. These tasks demand a dual expertise: understanding complex algorithms and possessing strong programming and software engineering skills. While we lack the latter expertise, we are capable of programming the algorithms, prioritizing the efficiency of implementations. Additionally, we believe we can leverage internal support from research engineers in our laboratory who are accustomed to collaborating with researchers. Notably, our research department includes an engineer with expertise in automata theory.

Task 6: Organizing a summer school for young researchers

This task consists of organizing a spring or summer school aimed at master's or doctoral students.

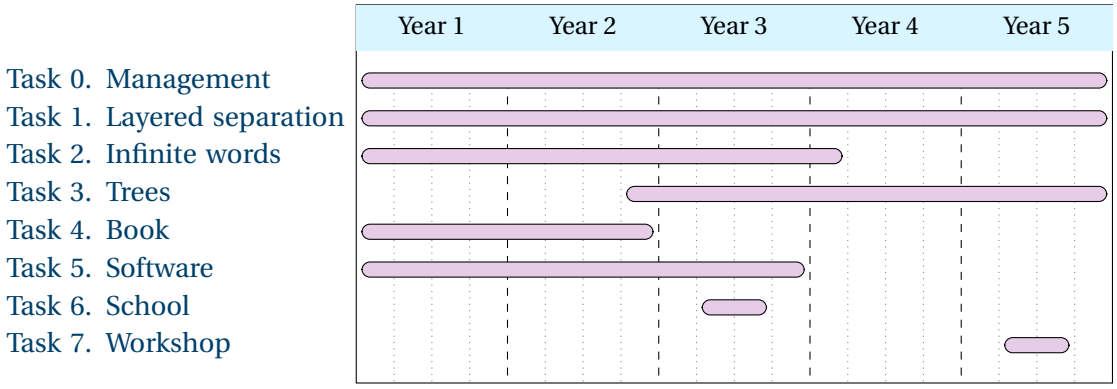
Recent advancements in the field of regular languages exist within a broader domain. Specifically, the separation problem has been explored in various contexts, as referenced in works such as [CLP20, CL19, KZ23, BMZ23, CZ20]. While workshops have previously been convened on this topic, they were either relatively confidential (*e.g.*, organized as part of ANR projects involving students), aimed at senior researchers, or covered broader thematic scopes than those of this project. Notably, these new findings have yet to be the focal point of a thematic school for young researchers (master's or doctoral students). Our goal is to host a school for 20 to 30 students over the course of the project, spanning 4 or 5 days and held at a convenient location in France.

Given the pervasive nature of regularity in theoretical computer science, we believe this topic will be of interest to many students. Moreover, material for part of this thematic school will already be available, stemming from the book outlined in Task 4. The requested funding for this task will be allocated towards inviting colleagues to deliver lectures and for local organization expenses.

Task 7: Organizing a workshop for specialists

The final task will entail convening a small group of colleagues (fewer than ten) engaged in fields related to this project's themes. We already have several colleagues in mind whom we would like to invite. The objective will be to foster idea exchange centered on the separation problem, particularly to deepen understanding of recent developments and result articulation. Ideally, this task will be organized towards the conclusion of the project.

The following Gantt diagram shows a tentative planning of the tasks.



In this diagram, we try to give an approximate overview. Layered separation is expected to be a long-term task. On the other hand, three years seem to be enough to address infinite words. The overlap between the two parts of Objective 2 (Tasks 2 and 3) is intentionally short. Task 4 concerns only the first volume of the book (but we may continue to work on the last parts after Year 2). We think that the development will not last more than three years (in the context of this project).

Risk management

The project introduces two novel challenges departing from previous endeavors: a more intricate problem surpassing the separation problem on finite words, and the exploration of the separation problem on trees. Nonetheless, the project has been carefully structured to mitigate associated risks.

- The primary risk is evident: the aspiration to achieve substantial theoretical results without any certainty. However, the project encompasses a range of problems varying in difficulty levels.

Typically, mitigating such risks involves tackling simpler cases before advancing to more complex ones. Moreover, the project's duration of 60 months provides flexibility: in case of impasse on one problem, resources can be reallocated to focus on another or even redirect the question.

- Tasks related to writing the book and developing the software are well-defined and entail minimal risk.
- Finally, we aim to recruit a skilled postdoctoral researcher and assign them problems to solve over the planned one-year period. This is also an area of concern. However, we have chosen a well-defined problem for this postdoctoral researcher, involving the adaptation of results already obtained on finite words. As mentioned earlier, results on infinite words generally follow those of finite words, providing a pre-existing toolkit and clear guidance. Additionally, Task 2 comprises two sub-objectives of varying difficulties, allowing for further adaptation if necessary. Furthermore, there is a consistent supply of talented students in the research area, and the planned duration will aid in hiring a suitable candidate.

In summary, while we propose new and ambitious questions in this project, the associated risks are mitigated by the progressive difficulty of the questions, the multifaceted nature of the project, and its duration spanning over 5 years.

I.d. Ability of the project to address the research issues covered by the chosen research theme

The project is connected to the following scientific parts of the chosen scientific theme (E.01 axis):

- Informatique fondamentale.
- Calculabilité et décidabilité.
- Logique.
- Modèles de calcul.

II. Organization and implementation of the project

II.a. Scientific coordinator and its consortium / its team

Below is a brief introduction to the two members of our team. Both intend to allocate **90%** of their research time to this project, essentially dedicating all their research efforts to it. Marc Zeitoun has been working in automata theory since the early 1990s and Thomas Place has been active in the field since the late 2000s. Since 2013, Thomas Place and Marc Zeitoun have been collaborating on decision problems for classes of regular languages. A significant contribution came in 2014 when they developed an algorithm capable of determining whether a regular language can be expressed in the fragment of first-order logic with only one quantifier alternation [PZ14]. This breakthrough resolved a problem that had remained unsolved for 45 years.

Project's longevity and perenniality of the team.

Thomas Place and Marc Zeitoun have been working together since 2013. They are a team that develops research on open questions in automata theory. Between their first joint paper (MFCS 2013) and the last one (LICS 2023), they have published together 27 papers: 17 in conferences (including 7 papers at ICALP and LICS) and 9 papers in journals (LMCS, TOCL, JACM, ToCS, TheoretiCS).

In addition to the scientific objectives outlined in this document, the team has two other projects it intends to pursue within the proposed submission:

- One project involves writing a book on the team's research topics. This book aims to provide a comprehensive overview of the team's research to a wide scientific audience, including students, in an accessible manner, while also incorporating the latest advances in the field (Task 4).
- The other project is to develop software implementing the team's algorithms (Task 5).

Marc Zeitoun (scientific coordinator).

- **Academic record.** PhD thesis in 1993 and habilitation thesis in 2004.
- **Positions.** Full professor at LaBRI, Bordeaux University since 2005.
- **Short CV.** He supervised 4 PhD theses. He is the author of 38 publications in international conferences (LICS, ICALP, STACS, FoSSaCS, MFCS, CSL), 34 publications in international journals in computer science and mathematics and one book chapter. He gave several invited lectures (recently: Highlights in Logic, Games and Automata 2016, Computer Science in Russia 2017, forthcoming: Automatha 2024). He participated in several program committees (STACS'13, STACS'15, FCT'17 (co-chair), ICALP'20, DLT'21, DLT'22). He co-organized FCT'17 in Bordeaux. He serves as an associate editor for Journal of Computer and System Sciences.
- **Implication in the project**: 90% of the research time.

Thomas Place.

- **Academic record.** PhD Thesis in 2010.
- **Positions.** Assistant professor at LaBRI, Bordeaux University, since 2012.
- **Short CV.** He co-supervised 2 PhD theses. He is the author of 28 papers in international conferences (LICS, ICALP, STACS, FoSSaCS, MFCS, CSL) including a distinguished paper at LICS'22 and 14 papers in international journals in computer science. He gave an invited lecture in LATA'20. He participated in several program committees (LICS'18, DLT'19, MFCS'20).
- **Implication in the project**: 90% of the research time.

The team behind this proposal is part of the M2F department at the LaBRI laboratory and collaborates within the LX team. They are well integrated; for instance, Thomas Place gave a talk in the M2F seminar in June 2023, and Marc Zeitoun is scheduled to give one in June 2024.

Implication of the scientific coordinator in on-going projects

The scientific coordinator is currently not involved in any on-going project. He intends to devote 90% of his research time to the project described in this proposal.

II.b. Implemented and requested resources to reach the objectives

Partner 1: Bordeaux University

Staff expenses

As explained earlier, the extension of results achieved on finite words to infinite words (Scientific Objective 2, Task 2) would significantly benefit from the assistance of a **postdoctoral researcher**. Conversely, the associated task is structured to be more approachable than the others: it entails adapting the concepts established for finite words, alongside certain standard tools utilized for infinite words. The former are presented in the book addressed in Task 4, while the latter are standard and have been covered in a textbook [PP04].

We are not seeking funding for Task 5 (software development) because, as previously stated, we believe it would be cost-prohibitive to hire an engineer solely for this purpose, given the requirement for expertise in separation concepts. Instead, if necessary, we will seek assistance within our laboratory.

Overheads costs

- Missions grants: 24000€. We regularly publish in the main conferences of our field. We also participate to research workshops (such as the workshop proposed in Task 7 of this proposal). The requested amount allows funding for about 12 missions over 5 years, which amounts to one to three per year, for both permanent members and the postdoc (of course, the actual amount of a particular mission depends on the venue, the conference fees, etc.). While this is lower than the number of conferences we attend annually as a team, we plan to request co-funding of the laboratory, in particular for missions involving the postdoc.
- Spring or summer school for youg researchers: 4000€.
  - The requested amount will enable us to invite two or three speakers, with possibly one or two coming from abroad.
  - We plan to request co-funding to cover local expenses such as coffee breaks, giveaways, and other miscellaneous costs.

  We plan to associate this school with a meeting of a working group from the CNRS GDR "Fundamental Computer Science and its Mathematics". This way, the travel expenses of the participants will already be covered.
- Workshop 3000€. This amount will enable us to invite a two or three French speakers. We plan to organize the workshop in an accessible location, in order to decrease the costs.

Requested means by item of expenditure and by partner

Staff expenses	60 500,00 €
Overheads costs	31 000,00 €
Administrative management & structure costs	13 267,50 €
Requested funding	104 767,50 €

III. Impact and benefits of the project

Since this is fundamental research, the impact primarily lies in the communication and dissemination of results, as discussed throughout the text:

- Firstly, we aim to publish our findings in renowned conferences (such as LICS or ICALP) and journals, preferably with diamond open access.
- Secondly, we intend to promote the two resources we plan to develop: the book and the software. This may involve presentations at dedicated conferences, particularly regarding the software.
- Finally, our impact will extend to young students through the summer or spring school.

Research on the topics of this project has been vibrant over the last few decades, and the theory is becoming stable and mature. Therefore, we believe this is an opportune time to disseminate these ideas within the theoretical computer science community.

IV. References

[BC71] Janusz A. Brzozowski and Rina S. Cohen. Dot-depth of star-free events. *Journal of Computer and System Sciences*, 5(1):1–16, 1971.

[BCST92] David A. Mix Barrington, Kevin Compton, Howard Straubing, and Denis Thérien. Regular languages in nc1. *Journal of Computer and System Sciences*, 44(3):478 – 499, 1992.

[BMZ23] Pascal Baumann, Roland Meyer, and Georg Zetzsche. Regular separability in büchi VASS. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany*, volume 254 of *LIPIcs*, pages 9:1–9:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[BP91] Danièle Beauquier and Jean-Eric Pin. Languages and scanners. *Theoretical Computer Science*, 84(1):3–21, 1991.

[BP12] Mikolaj Bojanczyk and Thomas Place. Regular languages of infinite trees that are boolean combinations of open sets. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*, volume 7392 of *Lecture Notes in Computer Science*, pages 104–115. Springer, 2012.

[BS09] Michael Benedikt and Luc Segoufin. Towards a characterization of order-invariant queries over tame graphs. *J. Symb. Log.*, 74(1):168–186, 2009.

[BS10] Mikolaj Bojanczyk and Luc Segoufin. Tree languages defined in first-order logic with one quantifier alternation. *Log. Methods Comput. Sci.*, 6(4), 2010.

[BSS12] Mikolaj Bojanczyk, Luc Segoufin, and Howard Straubing. Piecewise testable tree languages. *Log. Methods Comput. Sci.*, 8(3), 2012.

[BW06] Mikolaj Bojanczyk and Igor Walukiewicz. Characterizing EF and EX tree logics. *Theor. Comput. Sci.*, 358(2-3):255–272, 2006.

[CL19] Wojciech Czerwinski and Slawomir Lasota. Regular separability of one counter automata. *Logical Methods in Computer Science*, 15(2), 2019.

[CLP20] Lorenzo Clemente, Slawomir Lasota, and Radoslaw Piórkowski. Timed games and deterministic separability. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 121:1–121:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[CMM13] Wojciech Czerwiński, Wim Martens, and Tomáš Masopust. Efficient separability of regular languages by subsequences and suffixes. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming*, ICALP'13, pages 150–161, Berlin, Heidelberg, 2013. Springer-Verlag.

[CPS06] Laura Chaubard, Jean-Eric Pin, and Howard Straubing. First order formulas with modular predicates. In *Proceedings of the 21th IEEE Symposium on Logic in Computer Science (LICS'06)*, pages 211–220, 2006.

[CvGM22] Thomas Colcombet, Sam van Gool, and Rémi Morvan. First-order separation over countable ordinals. In *Proceedings of the 25th international conference on Foundations of Software Science and Computation Structures*, FOSSACS'22, pages 264–284. Springer, 2022.

[CZ20] Wojciech Czerwinski and Georg Zetzsche. An approach to regular separability in vector addition systems. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 341–354. ACM, 2020.

[Heu88] Uschi Heuter. First-order properties of trees, star-free expressions, and aperiodicity. In *STACS'88*, pages 136–148, 1988.

[Kna83] Robert Knast. A semigroup characterization of dot-depth one languages. *RAIRO - Theoretical Informatics and Applications*, 17(4):321–330, 1983.

[KW15] Manfred Kufleitner and Tobias Walter. One quantifier alternation in first-order logic with modular predicates. *RAIRO-Informatique Théorique*, 49(1):1–22, 2015.

[KW18] Manfred Kufleitner and Tobias Walter. Level two of the quantifier alternation hierarchy over infinite words. *Theory Comput. Syst.*, 62(3):467–480, 2018.

[KZ23] Chris Köcher and Georg Zetzsche. Regular separators for VASS coverability languages. In Pa-

tricia Bouyer and Srikanth Srinivasan, editors, *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2023, December 18-20, 2023, IIIT Hyderabad, Telangana, India*, volume 284 of *LIPIcs*, pages 15:1–15:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[MP71] Robert McNaughton and Seymour A. Papert. *Counter-Free Automata*. MIT Press, 1971.

[MP85] Stuart Margolis and Jean-Eric Pin. Product of Group Languages. In *FCT Conference*, volume 199, pages 285–299. Springer-Verlag, 1985.

[Pin80] Jean-Eric Pin. Propriétés syntactiques du produit non ambigu. In *Proceedings of the 7th International Colloquium on Automata, Languages and Programming*, ICALP'80, pages 483–499, 1980.

[Pin98] Jean-Eric Pin. Bridges for concatenation hierarchies. In *Proceedings of the 25th International Colloquium on Automata, Languages and Programming*, ICALP'98, pages 431–442, Berlin, Heidelberg, 1998. Springer-Verlag.

[Pla15] Thomas Place. Separating regular languages with two quantifiers alternations. In *Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science, (LICS'15)*, pages 202–213. IEEE Computer Society, 2015.

[Pla18] Thomas Place. Separating regular languages with two quantifier alternations. *Logical Methods in Computer Science*, 14(4), 2018.

[PP86] Dominique Perrin and Jean-Eric Pin. First-order logic and star-free sets. *Journal of Computer and System Sciences*, 32(3):393–406, 1986.

[PP04] Dominique Perrin and Jean-Éric Pin. *Infinite Words*. Elsevier, 2004.

[PPZ16] Théo Pierron, Thomas Place, and Marc Zeitoun. Quantifier alternation for infinite words. In *Proceedings of the 19th international conference on Foundations of Software Science and Computation Structures International Conference*, FOSSACS'16, pages 234–251. Springer, 2016.

[PS11] Thomas Place and Luc Segoufin. A decidable characterization of locally testable tree languages. *Log. Methods Comput. Sci.*, 7(4), 2011.

[PS16] Thomas Place and Luc Segoufin. Decidable characterization of fo2(<, +1) and locality of DA. *CoRR*, abs/1606.03217, 2016.

[PvRZ13] Thomas Place, Lorijn van Rooijen, and Marc Zeitoun. Separating regular languages by piecewise testable and unambiguous languages. In *Proceedings of the 38th International Symposium on Mathematical Foundations of Computer Science*, MFCS'13, pages 729–740, Berlin, Heidelberg, 2013. Springer-Verlag.

[PW97] Jean-Eric Pin and Pascal Weil. Polynomial closure and unambiguous product. *Theor. Comp. Syst.*, 30(4):383–422, 1997.

[PZ14] Thomas Place and Marc Zeitoun. Going higher in the first-order quantifier alternation hierarchy on words. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming*, ICALP'14, pages 342–353, Berlin, Heidelberg, 2014. Springer-Verlag.

[PZ16] Thomas Place and Marc Zeitoun. The covering problem: A unified approach for investigating the expressive power of logics. In *Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science*, MFCS'16, 2016.

[PZ18] Thomas Place and Marc Zeitoun. The covering problem. *Logical Methods in Computer Science*, 14(3), 2018.

[PZ19a] Thomas Place and Marc Zeitoun. Generic results for concatenation hierarchies. *Theor. Comp. Syst. (ToCS)*, 63(4):849–901, 2019.

[PZ19b] Thomas Place and Marc Zeitoun. Going higher in first-order quantifier alternation hierarchies on words. *JACM*, 66(2):12:1–12:65, 2019.

[PZ19c] Thomas Place and Marc Zeitoun. On all things star-free. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming*, ICALP'19, pages 126:1–

126:14, 2019.

[PZ19d] Thomas Place and Marc Zeitoun. Separation and covering for group based concatenation hierarchies. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS'19, pages 1–13, 2019.

[PZ22] Thomas Place and Marc Zeitoun. A generic polynomial time approach to separation by first-order logic without quantifier alternation. In *Proceedings of the 42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, FSTTCS'22, 2022.

[Sch65] Marcel Paul Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8(2):190–194, 1965.

[Sch75] Marcel Paul Schützenberger. Sur certaines opérations de fermeture dans les langages rationnels. *Symposia Mathematica*, XV:245–253, 1975. Convegno di Informatica Teorica, INDAM, Roma, 1973.

[Sim75] Imre Simon. Piecewise testable events. In *Proceedings of the 2nd GI Conference on Automata Theory and Formal Languages*, pages 214–222, Berlin, Heidelberg, 1975. Springer-Verlag.

[Str79] Howard Straubing. Aperiodic homomorphisms and the concatenation product of recognizable sets. *Journal of Pure and Applied Algebra*, 15(3):319 – 327, 1979.

[Str81] Howard Straubing. A generalization of the Schützenberger product of finite monoids. *Theoretical Computer Science*, 13(2):137–150, 1981.

[Str85] Howard Straubing. Finite semigroup varieties of the form V ∗ D. *Journal of Pure and Applied Algebra*, 36:53–94, 1985.

[Thé81] Denis Thérien. Classification of finite monoids: The language approach. *Theoretical Computer Science*, 14(2):195–208, 1981.

[Tho82] Wolfgang Thomas. Classifying regular events in symbolic logic. *Journal of Computer and System Sciences*, 25(3):360–376, 1982.