1. CONTEXT AND POSITIONING OF THE PROPOSAL

Summary. One of the challenges of computer science is to manipulate objects from an infinite set using finitary means. All data processing problems have an infinite number of potential input data. All but the simplest specifications of computer systems talk about an infinite set of possible behaviors, be it, for example, as input/output relation or as infinite sequences of possible actions. Of course mathematics is well accustomed to deal with infinite sets. But it is computer science that brings a completely new dimension to the picture, namely that of effectiveness.

One of the central concepts that have emerged from computer science in response to this challenge is that of recognizability, whose combination with logic and automata has proved incredibly fruitful. Both logic and automata theory have then seen their areas of applications extend far beyond what could be imagined at their creation. One can for example refer to an essay "On the Unusual Effectiveness of Logic in Computer Science" [HHI01] whose title appropriately summarizes this phenomenon and draws a comparison with the role of mathematics in physics.

The theory of automata and recognizability has developed in two main directions: as an ever more sophisticated and efficient tool to handle finite, sequential and discrete behaviors (languages of finite words); and through a number of extensions of the theory aiming at the analysis of more complex, possibly infinite behaviors, which may include notions of concurrency, branching, quantitative information, etc. These extensions are motivated by increasingly important applications, and in particular by the fields of verification and computational linguistics. **With this project, we want to push further the frontiers** of these developments in important directions, described below and which center around, trees, λ-terms and the models of quantitative automata. Our main strategy will be to exploit and expand the most recent developments of the theory of recognizable languages of finite words, of an algebraic and topological nature, as a guide and a blueprint for the construction of the theory of more complex structures required in applications.

**

Background. Among all the types of data one meets in Computer Science, words, that is, sequences of letters, are arguably the most basic. This is not only because every information is encoded *in fine* into a sequence of bits, but more importantly, because words are often natural representations of computer science objects. However, this is not to say that other representations are just auxiliary. If we thought about natural numbers only in terms of their binary encodings we would miss a lot of structure. For similar reasons, trees and not words have become the dominant abstract model of a document or a database (e.g. XML encoding).

The cases of words or trees are representative of the development of research: on one hand we investigate how to express "reality" in this structure, and on the other hand we develop the theory of this structure motivated by needs of applications. Towards either objective, providing means for describing and manipulating languages of words and trees is a fundamental task. There are basically three kinds of approaches:

- *Logic*. The language of logic offers a sophisticated (and sometimes intuitive[1]) tool to specify very precisely the properties of a structure. We can benefit from close to a century of research on logic, including its algorithmic aspects, based on the recursive nature of logical formulas.
- *Automata*. This is again a very intuitive tool, to describe (approximations of) complex systems. This representation is interesting because it is very close to the implementation of systems. The discrete nature of automata (which can be viewed as labeled graphs) also opens the door to effective algorithms.
- *Algebra*. The third important approach is through algebra. It is not a surprise to see it enter into the

[1] Especially when one uses modal logics, such as temporal logic.

picture, since it has natural and well-established connections with logic. Yet it is fascinating to see that it is inescapable, in the sense that it provides some essential tools available neither in logic nor in automata.

Historical context. Let us look more closely at the triangle formed by logic, automata and algebra in the case of word languages. In this context, algebra brings a fresh element, namely the semigroup structure. And indeed, this structure does not appear naturally, either in logic, or in automata. But we know since the 1960s that the semigroup structure is essential to tackle certain problems. For instance, only algebra provides the means to conceptually and algorithmically classify regular languages. Schützenberger's and McNaughton-Papert's theorems characterizing first-order definable languages by aperiodic semigroups (1965 and 1970), are the flagship examples of this situation. Since then, many important decidability results and algorithms concerning automata and logic have been obtained through the algebraic connection, and can be best understood in that light.

While the three approaches (through logic, automata and algebra) are complementary, it is fair to say that in different periods of time some received more attention than the others. In the 1980s and the beginning of the 1990s, logics flourished, in particular variants of temporal and program logics. This development crystallized some fundamental questions and the right answers were obtained through automata theory. This in turn revived the interest in automata theory, which enjoyed a rapid development from the mid-1990s. Arguably, algebra has been able in recent years to provide answers to persistent problems in both logic and automata theory (e.g. Thérien and Wilke's result on hierarchies of temporal logic [TW04] or Bojanczyk and Walukiewicz's [BW06] on tree logics). We are convinced that more is to come, that an algebraic approach can provide in the next few years a new impetus to the field.

Goal of the project, organization. The goal of this project is to be a driving force behind the extension of the algebraic approach made possible by recent advances. In our opinion, a new frontier is being opened for the study of recognizability. In order to have a concrete work plan in this vast area, we propose to concentrate on the development along four particular fronts. One of these fronts (Task 2) concerns tree languages: trees have become a major model in computer science, because of the emergence of XML, by now a standard format of data exchange on the web but also an underlying structure in databases, and because of their role in verification. The second front (Task 3) concerns languages of λ-terms. While they may be represented by trees, λ-terms are much more than that, as they specify functions rather than terms. They are essential tools in computational linguistics and in semantics, but are not usually studied from the point of view of recognizability. The third front (Task 4) on which we want to progress, is the study of automata with limited counters, for word or tree languages. We will present later arguments for this choice, and in particular surprising connections of these kinds of models with classical questions. The last front that we want to develop (Task 5), which could well be termed *foundational*, aims at developing algebraic and topological tools for the other tasks, based on the emergence of the new concepts and completely new results which have thoroughly revolutionized the algebraic theory of recognizable languages of finite words.

The project will drive its impetus from recent important developments on all four fronts. On the most historically established of them, Task 5, such a development is the discovery of the role of the Stone duality and of the power of profinite equations. Research on Task 2 has also a relatively long history, but as we explain in more detail later, recent algebraic approaches have contributed to a new development of the field. The remaining Tasks 3 and 4 offer a very novel, promising and largely unexplored potential. Below we argue why these two tasks deserve to be put at the same level as the two more "classical" tasks. We believe that the interaction of tasks at different levels of maturity, can offer enormous benefits to all of them.

Positioning of the project, originality. There are other active research groups that deal with some of the topics we want to discuss. For instance, Aachen, LSV (Cachan), LIFL (Lille), Szeged and Warsaw are very well placed on the interaction between logic and automata for tree languages, including

applications to verification, with also a strong algebraic flavor in Szeged and Warsaw. Boston, Brno and Porto have strong expertise on the algebraic approach for automata on words, with a tree flavor in Boston as well. Expertise on the topological aspects of automata theory can be found in Lausanne, Novosibirsk, Nijmegen and Würzburg. Aachen again, Darmstadt and Leipzig are important centers for what concerns automata with quantitative data. There are many departments with expertise on λ-calculus and semantics, or λ-calculus and computational linguistics, but none approaches it from the point of view of recognizability.

The originality of the consortium formed by this project certainly lies in the blend it offers: the process that led to this project was initiated by specialists of automata theory, logic and algebra, but it rapidly attracted specialists from other communities (namely verification and computational linguistics), who saw the potential of harnessing the power of modern automata-theoretic methods to a wider variety of situations. The work we plan on algebraic and topological tools, and on the theory of recognizable tree languages relies on very recent results and is cutting-edge. What we propose on λ-terms and on cost automata is extremely novel. Another original feature is the critical mass of internationally recognized experts gathered in this project. The LaBRI and LIAFA teams both strongly contributed to the development of the algebraic approach in recent years. At the same time, they are well-established contributors to the other problems we want to tackle, and their recent results on forest algebras and stabilization monoids are at the forefront of research. In our opinion, the importance of algebraic and topological tools will grow in computer science, and the ambition of this project is to be one of the focal points of this development.

2. SCIENTIFIC AND TECHNICAL DESCRIPTION

2.1. BACKGROUND, STATE OF THE ART

Automata, logic and algebra: this fruitful triangle has been known and exploited for decades for the study and the classification of word languages, see the books of Pin [Pin86], Straubing [St94], Perrin and Pin [PP04], and the surveys of Thomas [T97], Tesson and Thérien [TT02, TT07]. Some of the prominent results in this domain are the characterization of first-order definable languages as exactly those that are definable in temporal logic, those that are star-free and those whose syntactic semigroup is aperiodic; the characterization of locally testable as those whose syntactic semigroup is locally J-trivial; and in the more recent period, the characterization of fragments or extensions of first-order logic (FO+MOD, FO[Succ], FO^2) by properties of syntactic semigroups; down to the very recent characterization of hierarchies within temporal logic [TW02, TW04] and within FO [DK09]. This branch of research remains lively, with the investigation, e.g., of quantifier alternation hierarchies or of communication complexity. It is important to note that, for all the results mentioned above, it is the algebraic characterization that yields the decision algorithms for these classes of languages: the algebraic approach combines the two essential qualities of **expressiveness** and **effectiveness**.

These results provide the paradigm of what we want to achieve: to identify connections between classes of automata-theoretically defined languages (of trees, λ-terms, weighted words, etc), logical specifications and algebraic structures, extending the very refined results known for word languages. Ideally, one would wish to formulate our results within a unifying framework like the theory of varieties. Many connections between classes of automata and logical formalisms can be found in the literature, arguably too many: they often have an ad hoc flavor, and it is not clear what general figure emerges from these many results. One of the first difficulties, which we will confront explicitly, is the determination of an appropriate algebraic framework in which to view the universe of our structures (the set of all trees, of all λ-terms, of words with multiplicity, etc). In our view, this is a first step towards understanding the correct algebraic language in which our characterization or decision results should be formulated.

As explained earlier, our work will be structured along four fronts: the study of recognizable tree

languages, of recognizable sets of λ-terms and of recognizable cost functions, and the development of algebraic and topological tools that have proved to be crucial for the study of word languages, and which are going to be extended to trees, λ-terms and cost functions. The following survey of the state-of-the-art is organized accordingly.

Tree languages. We see trees in almost any part of computer science. Traditionally, ranked trees, that are nothing else but terms, have grabbed most of the attention, although exceptions could be found in graph theory or linguistics [Car92]. Unranked trees have recently become a subject of renewed interest, mainly because of the development of XML [Via01]. It is also quite common nowadays to see trees with infinite paths, especially in the context of verification.

Trees are generalizations of words, and share many common properties. It is straightforward to generalize MSO (monadic second-order logic) from words to trees. A generalization of automata for trees goes back to the classical work of Thatcher and Wright in the 1960s.

Despite these similarities, the past research has shown that generalizing from words to trees is sometimes a very difficult task. For example, the generalization of the above-mentioned 1965 theorem of Schützenberger is still a major open problem. Maybe even more unexpectedly, a suitable algebraic approach to tree languages has been missing for a long time. The proposals that have appeared in the recent years represent a real progress, but as long as the main open problems remain unsolved, one cannot claim that they are fully adequate.

The first attempt to extend the algebraic approach was done by Eilenberg and Wright [EW67] who introduced a notion of automaton in general algebras. Later Courcelle [C89] made it explicit that recognizability is a robust algebraic notion. Yet algebras obtained by this general approach are too close to automata to provide fresh insights. In the quest for an appropriate algebraic framework, Nivat and Podelski [NP89] looked at the algebra of tree contexts, trying to get closer to the structure of semigroups. Unfortunately, as Potthoff showed later, this approach is too weak to capture most of the interesting classes of tree languages. Wilke [W93] proposed a three-sorted algebra for trees, and initiated the study of the theory of these algebras. Later Salehi and Steinby [SS07] proved a variety theorem in this setting, but once again, the approach seems to be too close to automata to help characterize significant classes. More recently, Esik and Weil [ES05] introduced the notion of pre-clones. They proved a variety theorem for pre-clones and showed that many interesting classes can be characterized in their setting, including FO[<]; however, their results do not yield effective decision procedures for the logic. Bojanczyk and Walukiewicz proposed another algebraic notion called forest algebras. They developed the basic theory of these algebras [BW07, BSW09], including an analog of the wreath product principle and used it to give (ineffective yet again) characterizations of many interesting logics.

Parallel to the development of algebras for trees, other types of characterizations for classes of tree languages have been investigated. Thomas [T84] introduced regular expressions for trees that permit to study star-free tree languages. Heuter [H88] proposed regular expressions capturing FO logic over trees. She showed that, surprisingly, not every aperiodic tree language is FO-definable. Potthoff and Thomas [PT93] discovered the even more unexpected fact that all regular tree languages are star-free. The subsequent work of Potthoff [P94, P95] produced very interesting examples and proved that the above-mentioned approach of Nivat and Podelski is too weak to capture FO. Wilke [W96] gave a decidable characterization of frontier testable languages using his setting. A new impetus to the field was given by decidable characterizations of the EF logic [BW04], and of the first-order logic with successor [BS05]. These results indirectly led to the definition of forest algebras. The more recent decidable characterizations [B07, BSS08, BS08, PS09] have all been obtained using forest algebras.

Languages of λ-terms. The simply typed λ-calculus is a natural extension of strings and trees. Indeed, a ranked alphabet can easily be represented by means of a higher order signature: a rank n symbol r is represented as a second order constant **r** of type $o \to \cdots \to o \to o$ (n+1 occurrences of o) and a tree $r(t_1,\ldots,t_n)$ is isomorphically interpreted as the closed λ-term of type o, $\underline{r(t_1,\ldots,t_n)} = \mathbf{r}\ \underline{t_1}\ldots\underline{t_n}$. As usual,

words can be viewed as trees whose nodes have rank 1, but with the difference that instead of completing the trees representing strings with a dummy leaf, this leaf is replaced by an abstracted variable, so that when strings are seen as λ-terms, function composition plays the role of concatenation and the identity function stands for the empty word. In a nutshell, this representation of strings as lambda-terms is a representation of a free monoid with monadic functions (similar to the consideration of contexts when investigating trees).

In various contexts, simply typed λ-calculus is used to define languages. In particular, after the work by Engelfriet and Schmidt [ES77] about IO and OI context-free tree grammars, Damm [Dam82] studied a generalization of the notions of context free languages for strings and trees using simply typed λ-calculus, which yielded the so-called IO and OI hierarchies. The OI hierarchy is closely related to higher-order stack automata and it is a grammatical alternative for the study of the languages defined by these automata. Because strings and trees can be easily embedded in the simply typed λ-calculus, it is convenient to generalize the IO and OI hierarchies, and to consider them as defining languages of λ-terms rather than languages of strings or of trees.

In the context of formal linguistics, languages of λ-terms appear naturally in two contexts: 1) in formal semantics where meaning representations are often obtained by means of λ-terms; 2) in syntactic structures where certain grammatical derivations obtained in some logics can be seen as λ-terms through the Curry-Howard isomorphism. Two similar proposals were made by Muskens [Mus01] and de Groote [dGr01] in which λ-calculus was considered as the main tool for describing language.

Just like in the case of words, where the notion of recognizable languages is very useful in the study of context free grammars, a notion of recognizable sets of λ-terms is of great help in the study of languages of λ-terms. Salvati proposed such a notion in [Sal09]. This notion has two remarkable properties that make it quite useful. Firstly, it generalizes the notion of recognizability of strings and trees, in the sense that a set of strings (resp. trees) is recognizable in the usual sense if and only if the set of λ-terms that represent those strings (resp. trees) is a recognizable set of λ-terms. Secondly, recognizable sets of λ-terms are closed under βη-conversion. The first property pleads in favor of the proposed notion by showing that it is natural, while the second property shows its power and its relevance for λ-terms. Indeed, recognizability is usually about static objects that have a given form, e.g. the succession of letters for a string, while this extension of recognizability to λ-terms shifts it to dynamic objects and programs (expressed as simply typed λ-terms) that produce a given string. These dynamic objects are handled like static ones in this notion of recognizability, without losing their dynamicity. One may draw a parallel with the notion of recognizable trace languages, used in the theory of distributed computing: traces are equivalence classes of strings up to the commutation of certain letters, and the relevant notion of recognizability is closed under these partial commutations.

For the moment, the study of this notion is at a very early stage. It has two equivalent definitions: one is in terms of finite models of the simply typed λ-calculus which corresponds to the usual algebraic definitions of recognizability for strings in terms of finite semigroups and for trees in terms of finite algebras; the second is expressed in terms of typing and corresponds to an automata-like presentation of recognizability. The usual Boolean closure properties are proved, by generalizing the usual constructs for automata. Finally the closure under inverse homomorphisms is proved by a rather simple construction. This theorem has many applications. It yields a simplification of the proof by Damm [Dam82] of the closure the IO hierarchy under intersection with recognizable sets of λ-terms. It also gives for this class of languages a simple generalization of Thatcher's result about the recognizability of the parse forest of a context free grammar. It proves that the problem of text generation, when λ-terms are used to represent meaning, is decidable — which simplifies the proof given by [Sal]. Finally, the decidability of fourth-order matching can be seen as a corollary of the closure of the IO hierarchy under intersection with recognizable sets of λ-terms.

Cost automata: beyond recognizability. Rational languages are so central in computer science and logic that it became quite clear already in the 1950s that a similar theory allowing to reason about

quantitative notions would be of great interest. A first solution, advocated by Schützenberger, is to add to automata multiplicities ranging in a semiring [Sch61]. This well known method for enriching language theory with quantitative capabilities led to the development of the theory of rational power series, a rich branch in automata and combinatorial complexity. Many other models allowing automata to count were proposed with some specific goals in mind, such as modelization and verification.

These models are interesting on their own, but they present some inherent difficulties. For instance, the equivalence problem for automata with multiplicities in the tropical semiring is undecidable by a deep result of Krob [Kro92].

One specific brand of quantitative models has been introduced for the resolution of the famous (restricted) star-height problem on classical automata: the distance automata of Hashiguchi and the distance desert automata of Kirsten [Has82, Kir09]. Those are non-deterministic automata using counters that can be incremented or reset. Each such automaton computes a value from its input as the minimum over all runs, of the maximum value taken by a counter. Hashiguchi, Simon and Kirsten showed that such models have a decidable limitedness problem, i.e., that it is decidable whether the function computed by such an automaton is bounded. Those results were used to solve difficult problems in language theory, such as the star-height problem, the finite power problem, or the finite substitution problem. They were also used in model theory [BOW09], database theory [GT06], etc.

Colcombet [Col09] unified these results with the notion of cost functions. Two functions from words to non-negative integers are considered equivalent if on every set of words, both functions are bounded, or both are unbounded. A cost function is an equivalence class of functions modulo this equivalence. The undecidability result of Krob does not hold anymore when cost functions are considered rather than functions. It is shown in [Col09] and [Col09b] that a notion of regular cost function can be introduced, which is a strict extension of the standard notion for language, and which enjoys equivalent characterizations in terms of cost automata, recognizability by (stabilization) monoids and cost MSO: the key triangle formed by logic, automata and algebra is formed once more.

As an example of the various equivalent formalisms, the logic cost MSO extends MSO by the ability to express that a set has size at most n, providing that the test appears positively in the formula. The value of a formula is the least value of n such that the formula holds. One can for instance define in cost MSO, using standard methods, the diameter of a graph as the `least value of n such that for all vertices x, y, there exists an induced subgraph of size at most n inside which it is possible to go from x to y (x excluded)'.

In [CL08a], limitedness has been shown to be decidable for alternating tree automata, yielding good perspectives for the development of a model equivalent to cost MSO over finite trees. One also knows that the same theory over infinite trees would solve a very difficult problem in automata theory: the Mostowski hierarchy problem [CL08b].

Algebraic and topological tools. The definition of the syntactic semigroup first appeared in a paper of Rabin and Scott [RS59], but its first nontrivial use is due to Schützenberger. Schützenberger's theorem [Sch65] states that a recognizable language is star-free if and only if its syntactic semigroup is finite and aperiodic. Schützenberger's theorem was later supplemented by McNaughton [McNP71], who established the equivalence between star-free languages and FO[<], the first order logic of the order relation. These early results explain the importance of the structure theory of finite semigroups in automata theory. In return, the major developments in semigroup theory that occurred since the 1960s were mainly motivated by automata theory. Two other important results date back to the early seventies: Simon [Si75] described the languages whose syntactic semigroups are J-trivial and Brzozowski-Simon [BS73] and independently, McNaughton [McN74] characterized the locally testable languages.

These successes settled the power of the algebraic approach, which was axiomatized by Eilenberg's variety theorem [Eil76]. Eilenberg's theorem states that varieties of finite semigroups are in one to one

correspondence with certain classes of recognizable languages, the varieties of languages. For instance, the rational languages are associated with the variety of all finite semigroups, the star-free languages with the variety of finite aperiodic semigroups, and the piecewise testable languages with the variety of J-trivial semigroups.

Several attempts were made to extend Eilenberg's variety theory to a larger scope. For instance, ordered syntactic semigroups were introduced in [Pin95]. The resulting extension of Eilenberg's variety theory permits to treat classes of languages that are not necessarily closed under complement, contrary to the original theory. Other extensions were developed independently by Straubing [St02] and Ésik and Ito [EI03].

Varieties of finite semigroups are the finite counterpart of the much older notion of a Birkhoff variety. Birkhoff proved [Bi35] that varieties could be defined by a set of identities. Almost fifty years later, Reiterman [Re82] extended Birkhoff's theorem to varieties of finite semigroups: any variety of finite semigroups can be characterized by a set of identities between profinite words. Profinite words can be viewed as limits of sequences of words for a certain topology, the profinite topology.

The profinite approach is not only a powerful tool for studying varieties but it also led to spectacular developments, which are at the heart of the current research in this domain. In particular, Gehrke, Grigorieff and Pin [GGP08] proved that any lattice of recognizable languages could be defined by a set of profinite equations. This result subsumes Eilenberg-Reiterman's theory of varieties and its subsequent extensions. It also shows that any class of regular languages defined by a fragment of logic closed under conjunctions and disjunctions (first order, monadic second order, temporal, etc.) admits an equational description. This is particularly important as, for words but even more frequently for trees and other more complex structures, many significant families of languages lack the closure properties of varieties of word languages (under complement, residuals, inverse morphisms).

The extension of these methods to the investigation of languages of other structures than words is still in its infancy. A rather complete framework has been developed and exploited for ω -words, see the book of Perrin and Pin [PP04]. On trees, there were a few forays in this direction, e.g. [EW05], [EW09] on preclones and [BSS08], [BSW09] on forest algebras, but almost everything remains to be done. The slate is even blanker for λ-terms or cost functions.

2.2. RATIONALE HIGHLIGHTING THE ORIGINALITY AND NOVELTY OF THE PROPOSAL

Tree languages. There are a growing number of reasons for looking closer at tree formalisms. For example, in the context of XML, the nature of data (from an infinite set of labels, on which some operations are defined) is important. It is rather difficult to come with a decidable nontrivial formalism in this context [BDMSS06, JL07]. A better understanding of expressive power in the data-less case can help substantially. Even without data, understanding query and transformation languages for trees is still a challenge. At present there are arguably too many such formalisms. A unification and clarification effort is necessary, using more classical formalisms as a yardstick. An example of this kind of research is the work of Marx and de Rijke [MdR05] on navigational XPath and first-order logic. In another context, that of computer-aided verification, trees are used to model the behavior of information systems. While, our understanding of the relevant formalisms is much more satisfactory here, some outstanding open questions remain. Below we describe the problems that will drive our research.

Decidable characterization of logical fragments. There are several important logical formalisms for describing tree properties: monadic second-order logic (MSO), first-order logic (FO), computation tree logic (CTL), CTL*, the μ-calculus, to name a few. In contrast with words, we are far from having a good insight into the expressive power of the logics in question. One of the yardsticks of our understanding consists in having an algorithm to decide whether a given regular tree language is expressible in FO. While this yardstick seems ad hoc, in almost all the cases we know, such an effective characterization has provided a deep and decisive insight into the expressive power of the

logic. At present decidable characterizations are known only for a few fragments of MSO [BW06, BS05, B07, BSS08, BS08, PS09]. All of them use the algebraic approach. The big objective, maybe unrealistic in the time-span of this project, is to characterize first-order logic. Still there are many simpler, intermediate questions, as for fragments of FO inspired by XPath constructs.

Understanding the power of order invariance. A property is order-invariant if it does not distinguish two trees that differ only in the order of siblings. In Section 3.3.2, we mention a surprising example of Potthoff that shows the power and mystery of order invariance over trees. It is easy to draw a link between this property and our motivation concerning XML. Most of the time, XML trees do not have any logical order, but they have an *ad hoc* physical order due to the fact that they are stored in memory or streamed over a network. The challenge is to understand the power of order-invariance, i.e., how this additional "uncontrollable" order can be exploited.

Formalisms having a finite base. We plan also to explore the phenomenon of the existence, or the absence, of a finite base. Among the logics mentioned above, only CTL has a finite base, meaning that all formulas of the logic are obtained from atomic formulas using a finite number of operators. This property is very useful for practical (algorithmic) as well as theoretical (inductive proofs) reasons. The goal is to understand the limits of the expressive power of formalisms having a finite base.

Profinite approach for trees. An almost purely algebraic goal is to develop the variety theory for tree languages. As we have mentioned above, both for pre-clones and forest algebras this development has been started. In both cases the classical Eilenberg approach has been pursued. Yet, the profinite approach that we plan to develop in this project seems particularly interesting in the context of trees. As characterizations are so much more difficult in the tree case, it makes a lot of sense to look at logical fragments that are not necessarily closed under negation, or inverse homomorphisms. The profinite approach is the only one that permits to address such a challenge.

Languages of λ-terms. Recognizability for the simply typed λ-calculus is defined to be a generalization of the notions of recognizability for strings and trees. The exploration of this new notion is at a very early stage. Several questions arise naturally in this context.

Recognizability classics: regular expressions, logic and infinite Böhm trees. A first challenge is to explore the usual features of recognizability such as logic, regular expression and some generalization over infinite objects, in the context of λ-terms. Even though the extension of recognizability from strings and trees to λ-terms is rather straightforward, there are many problems related to the management of typing and of free variables that make the generalizations concerning logic, regular expressions and infinite λ-terms much more difficult.

Profinite techniques, Stone duality and semantics of programming languages. Another interesting aspect, that further connects the study of recognizable sets of λ-terms with the work proposed on algebraic and topological tools, is that it bridges recognizability with the study of the semantics of functional programming languages which benefits from a very rich literature. In particular, the use of Stone duality is very common in that field, and it is likely that through the generalization of recognizability to the λ-calculus the work of [GGP08] can be connected to earlier studies of semantics in programming languages. We expect a mutual fertilization through this possible connection. Since the questions considered in the field of formal language theory and in that of programming languages are rather different, we think that, similarly to what happened to the study of fourth order matching or of context free languages of λ-terms, this connection will be very fruitful for both fields. We expect in particular that results about pseudo-varieties for words and the techniques of profinite topology will be shifted to λ-terms.

Lambda-calculus and preclones. The notion of recognizability of λ-terms not only generalizes the notions of recognizability of strings and trees, but it also unifies them in the sense that both notions are embedded into a more general one. As such, we expect another understanding of the differences and similarities of strings and trees in the context of the λ-calculus. In particular, we hope that the

reinterpretation of certain notions like pre-clones in this setting, in connection to the work done on profinite topology and Stone duality, will shed additional light on the problem of developing a variety theory for the classification of tree languages.

Cost automata. The theory of cost automata and regular cost functions is rather new, and much remains to be done to understand it as well as the standard theory of regular languages. The objectives of this task are divided into four tracks, which are concerned with four different and rather independent directions of research. Each of those tracks has its own objectives, and specific difficulties to be solved.

The first track is concerned with the fine study of the algorithmic theory of regular cost functions over words. In particular, the goal is to reach the optimal complexity (i.e., obtain lower and upper bounds) for the key construction in the theory, the cost duality theorem. This requires investigating in deep details the difficult techniques involved in the original result. In this direction we are also interested in the equivalence with the formalism of weak cost MSO over infinite words: a proof of this result is likely to provide us with new techniques and possibly new models of automata for regular cost functions. The second track is concerned with the definition of new formalisms for regular cost functions under the form of temporal logics extended with counting capabilities. The goal is to obtain formalisms that are both easier to understand, and more efficient to solve. We plan to use the algebraic approach to guide our research, the intention being that strong formalisms should have simple algebraic characterizations (as it is the case for languages). This approach is evidently linked to Task 5.

The last two tracks aim at developing the theory of cost functions over trees. The objective of the third track is to develop an algebraic model for regular cost functions over trees, in order to characterize simple fragments of regular costs functions. Completely new ideas are required here, since the usual algebraic approaches over trees are insufficient here. This track is related to Task 2, and we expect mutual fertilization. The fourth track, which is certainly the most difficult and ambitious, aims at developing the theory of regular cost functions over infinite trees. It would at the same time extend Rabin's famous theorem on MSO [Rab69], and solve the difficult Mostowski problem [Col09b]. However, preliminary studies have shown the difficulty of the problem. We plan to investigate first the case of weak cost MSO, which seems, at first glance, more amenable to study.

Algebraic and topological tools. There are two major tasks. The first one is to complete the foundational work outlined in [GGP08], which covers several aspects. The fundamental observation is that, in Stone duality, the dual space of the Boolean algebra Reg of regular languages is the free profinite semigroup introduced by Reiterman [Re82]. Then sublattices (and not only Boolean algebras) of Reg can be viewed as quotients of the dual algebra and thus can be defined by equations. But the idea of Stone duality is so general that it can be applied to many other settings, and in particular for us to infinite words, trees or λ terms. Developing this theory in all possible directions is a major challenge of this project since it would have important consequences on all the other tasks. Advanced topology (uniform or even quasi-uniform spaces) is needed: although metric spaces give the right intuition they are not sufficient to cover all cases. A further challenge would be to explore the duality beyond regular languages, but the dual spaces are even more complicated than the profinite semigroups. Even on a one-letter alphabet, the dual space of the set of all languages is the Stone-Cech compactification of the natural numbers, a space known as the "three headed monster" [vM84].

The second task is to develop a battery of algebraic tools. Although there is an important existing literature on finite semigroups, the specific needs of this project require a deep understanding of the algebraic structures (semigroups, ordered semigroups, ω-semigroups, forest algebras, multi-sorted algebras, stabilization monoids). This encompasses structure theorems involving algebraic tools like semigroup expansions, semidirect product decompositions [PW02], Mal'cev products [PW97], pointlike sets [ACZ08], etc. For instance, a structure theorem for forest algebras similar to the Krohn-Rhodes theorem [Eil76] is probably the key to the solution of the decidability of FO on trees.

Many known results on regular languages, ω-regular languages, tree languages rely on algebraic

properties of their syntactic invariants [Sch65, Si75, PW97, PW02, PP04, BS08, BSS08, BSW09]. This includes decidability results for various logical fragments. For words, the Grail in this direction would be to solve the long-standing open problem of the decidability of the Σ_n hierarchy of FO[<], but any partial answer in this direction would be welcome. Other interesting fragments include FO[< + MOD], fragments of temporal logic or fragments occurring in database theory (words with data).

3. REFERENCES

[Abr88] S. Abramski, Domains Theory in Logical Form, *Annals of Pure and Applied Logic* **51** (1991), 1–77.

[ACZ08] J. Almeida, J. Costa and M. Zeitoun, Pointlike sets with respect to R and J. *J. Pure Appl. Algebra* **212** (2008), 486-499.

[AC98] R. Amadio and P.L. Curien, *Domains and Lambda Calculi*, Cambride University Press (1998).

[BS05] M. Benedikt and L. Segoufin, Regular languages definable in FO. In STACS'05, *Lect. Notes Comp. Sci.* **3404** (2005) 327 – 339. See the corrected version on the authors web page.

[Bi35] G. Birkhoff, On the structure of abstract algebras, *Proc. Cambridge Phil. Soc.* **31** (1935), 433–454.

[BLW09] A. Blumensath, M. Otto and M. Weyer, Boundedness of Monadic Second-Order Formulae over Finite Words, in ICALP 2009, Springer, *Lect. Notes Comp. Sci.* **5556** (2009), 67–78.

[BW06] M. Bojanczyk and I. Walukiewicz. Characterizing EF and EX tree logics. *Theoret. Comput. Sci.* **358** (2006), 255–272.

[BDMSS06] M. Bojanczyk, C. David, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data trees and XML reasoning. In PODS, pages 10–19. ACM, 2006.

[BW07] M. Bojanczyk and I. Walukiewicz. Forest algebras. In J. Flum, E. Grädel, and T. Wilke, editors, Logic and Automata, volume 2 of Texts in Logic and Games, pages 107–132. Amsterdam University Press, 2007.

[B07] M. Bojanczyk. Two-way unary temporal logic over trees. In LICS 2007, *IEEE Computer Society* (2007), 121–130.

[BSS08] M. Bojanczyk, L. Segoufin, and Howard Straubing, Piecewise Testable Tree Languages, LICS 2008, *IEEE Computer Society* (2008), pages 442-451

[BS08] M. Bojanczyk, and L. Segoufin, Tree Languages Defined in First-Order Logic with One Quantifier Alternation. ICALP 2009, *Lect. Notes Comp. Sci.* **5126** (2009), 233-245

[BSW09] M. Bojanczyk, H. Straubing, and I. Walukiewicz. Wreath products of forest algebras, with applications to tree logics. In LICS 2009, *IEEE Computer Society* (2009), 255–263.

[BP09] M. J. J. Branco and J.-E. Pin, Equations defining the polynomial closure of a lattice of regular languages, *ICALP 2009, Part II*, *Lect. Notes Comp. Sci.* **5556**, Springer Verlag, (2009), 115-126.

[BS73] J. A. Brzozowski and I. Simon, Characterizations of locally testable events, *Discrete Math.* **4** (1973), 243–271.

[Büc62] R. Büchi, On a decision method in restricted second order arithmetic, in Proceedings of the International Congress on Logic, Methodology and Philosophy of Science, Stanford Univ. Press (1962), 1–11.

[Car92] B. Carpenter, *The Logic of Typed Future Structures*, Cambridge University Press, 1992.

[CPS06] L. Chaubard, J.-E. Pin, and H. Straubing, First order formulas with modular predicates, 21st Annual IEEE Symposium on Logic in Computer Science, LICS 2006, *IEEE Computer Society* (2006), 211-220.

[CPP93] J. Cohen, D. Perrin and J.-E. Pin, On the expressive power of temporal logic for finite words, *Journal of Computer and System Sciences* **46** (1993), 271-294.

[Col09] T. Colcombet, The Theory of Stabilisation Monoids and Regular Cost Functions, in ICALP 2009, Springer, *Lect. Notes Comp. Sci.* **5556** (2009), 139–150.

[Col09b] T. Colcombet. Regular cost functions, Part I: logic and algebra over words, under submission.

[CL08a] T. Colcombet and C. Löding, The Nesting-Depth of Disjunctive μ-calculus for Tree Languages and the Limitedness Problem, in CSL, *Lect. Notes Comp. Sci.* **5213** (2008), 416–430.

[CL08b] T. Colcombet and C. Löding, The non-deterministic Mostowski hierarchy and distance-parity automata, in ICALP 2008, *Lect. Notes Comp. Sci.* **5126** (2008), 398–409.

[C89] B. Courcelle. On recognizable sets and tree automata. In H. A.-K. M. Nivat, editor, Resolution of equations in algebraic structures, pages 93–126. Academic Press, 1989.

[C91] B. Courcelle. The monadic second-order logic of graphs V: On closing the gap between definability and recognizability. *Theor. Comput. Sci.* **80** (1991),153–202.

[Dam82] W. Damm, The IO and OI Hierarchies, *Theoret. Comput. Sci.* 20 (1982), 95-207.

[dGr01] P. de Groote, Towards Abstract Categorial Grammars. ACL 2001, 148-155

[DK09] V. Diekert, M. Kufleitner, Fragments of First-Order Logic over Infinite Words. In STACS 2009, LIPIcs vol. 3, pp. 325-336, (Schloss Dagstuhl, 2009)

[Eil76] S. Eilenberg, *Automata, languages, and machines*, Vol. B, Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1976, Pure and Applied Mathematics, Vol. 59.

[EW67] S. Eilenberg and J. B. Wright, Automata in general algebras, *Information and Control* **11**, (1967) 452-470.

[EI03] Z. Ésik and M. Ito, Temporal logic with cyclic counting and the degree of aperiodicity of finite automata, *Acta Cybernetica* **16** (2003), 1–28.

[ES77] E. Engelfriet and E.M. Schmidt, IO and OI. I, *Journal of Computer and System Sciences* **15** (1977), 328-353.

[EW05] Z. Ésik and P. Weil, Algebraic recognizability of regular tree languages, *Theoret. Comput. Sci.* **340** (2005) 291-321.

[EW09] Z. Ésik and P. Weil, *International Journal of Algebra and Computation*, to appear.

[GGP08] M. Gehrke, S. Grigorieff et J.-E. Pin, Duality and equational theory of regular languages, in ICALP 2008, Springer, *Lect. Notes Comp. Sci.* **5126** (2008), 246–257.

[GT06] G. Grahne, A. Thomo: Regular path queries under approximate semantics. *Ann. Math. Artif. Intell.* **46** (2006), 165–190.

[HT87] T. Hafer and W. Thomas. Computation tree logic CTL and path quantifiers in the monadic theory of the binary tree. In 14th Internat. Coll. on Automata, Languages and Programming (ICALP'87), *Lect. Notes Comp. Sci.* **267** (1987), 269–279.

[HHI01] Joseph Y. Halpern, Robert Harper, Neil Immerman, Phokion G. Kolaitis, Moshe Y. Vardi, Victor Vianu, *Bull. Symbolic Logic* **7** (2001), 213-236.

[Has82] K. Hashiguchi, Limitedness Theorem on Finite Automata with Distance Functions, *J. Comput. Syst. Sci.* **24** (1982), 233–244.

[Has88] K. Hashiguchi, Relative star height, star height and finite automata with distance functions, in Formal Properties of Finite Automata and Applications, Springer, *Lect. Notes Comp. Sci.* **386** (1988), 74–88.

[H88] U. Heuter. First-order properties of trees, star-free expressions, and aperiodicity. In STACS'88, *Lect. Notes Comp. Sci.* **294** (1988) 136–148.

[JL07] M. Jurdzinski and R. Lazic. Alternation-free modal mu-calculus for data trees. In LICS 2007, *IEEE Computer Society* (2007),131–140.

[Kir05] D. Kirsten, Distance Desert Automata and the Star Height Problem, *ITA* **39** (2005), 455–509.

[Kir06] D. Kirsten, A Burnside Approach to the Finite Substitution Problem, *Theoret. Comput. Sci.* **39** (2006), 15–50.

[KO09] N. Kobayashi and C.-H. L. Ong, A Type System Equivalent to the Modal Mu-Calculus Model Checking of Higher-Order Recursion Schemes, LICS 2009, *IEEE Computer Society* (2009), 179-188.

[Kro92] D. Krob, The Equality Problem for Rational Series with Multiplicities in the Tropical Semiring is Undecidable, ICALP 1992, Springer, *Lect. Notes Comp. Sci.* **623** (1992), 101–112

[Le88] H. Leung, On the topological structure of a finitely generated semigroup of matrices, *Semigroup Forum* **37** (1988), 273-287.

[MdR05] M. Marx and M. de Rijke. Semantic characterizations of navigational xpath. *SIGMOD Record* **34** (2005), 41–46.

[McN66] R. McNaughton, Testing and generating infinite sequences by a finite automaton, *Information and Control* **9** (1966), 521–530.

[McN74] R. McNaughton, Algebraic decision procedures for local testability, *Math. Systems Theory* **8** (1974), 60–76.

[McNP71] R. McNaughton and S. Papert, *Counter-free automata*, The M.I.T. Press, Cambridge, Mass. London, 1971. With an appendix by William Henneman, M.I.T. Research Monograph, No. 65.

[Mus01] R. Muskens, Lambda Grammars and the Syntax-Semantics Interface, in R. van Rooy and M. Stokhof, editors, *Proceedings of the Thirteenth Amsterdam Colloquium* (2001), 150-155.

[NP89] M. Nivat, A. Podelski, Tree monoids and recognizability of sets of finite trees, in: H. Aït-Kaci, M. Nivat (Eds.), Resolution of Equations in Algebraic Structures, Vol. 1, Academic Press, Boston, MA, (1989), 351-367.

[N88] D. Niwiński. Fixed points vs. infinite generation. In LICS 2007, *IEEE Computer Society* (1988), 402–409.

[Ong06] C.-H. Luke Ong: On Model-Checking Trees Generated by Higher-Order Recursion Schemes. LICS 2006, *IEEE Computer Society* (2006), 81-90.

[PP04] D. Perrin and J.-E. Pin, *Infinite Words*, Pure and Applied Mathematics Vol 141, Elsevier, (2004), ISBN 0-12-532111-2.

[Pin86] J.-E. Pin, *Varieties of formal languages*, North Oxford, London and Plenum, New-York, (1986)

[Pin95] J.-E. Pin, A variety theorem without complementation, *Russian Mathematics* (Iz. VUZ) **39** (1995), 80–90.

[PS05] J.-E. Pin and P. Silva, A topological approach to transductions, *Theoret. Comput. Sci.* **340** (2005), 443-456.

[PW97] J.-E. Pin and P. Weil, Polynomial closure and unambiguous product, *Theory Comput. Systems* **30**, (1997), 383-422.

[PW02] J.-E. Pin and P. Weil, The wreath product principle for ordered semigroups, *Communications in Algebra* **30**, (2002), 5677-5713.

[PS09] T. Place, and L. Segoufin, A Decidable Characterization of Locally Testable Tree Languages, ICALP 2009 *Lect. Notes Comp. Sci.* **5556** (2009), 285-296

[PT93] A. Potthoff and W. Thomas. Regular tree languages without unary symbols are star-free. In Fundamentals of Computation Theory, *Lect. Notes Comp. Sci.* **710** (1993), 396–405.

[P94] A. Potthoff. Modulo-counting quantifiers over finite trees. *Theoret. Comput. Sci.*, 126:97–112, 1994.

[P95] A. Potthoff. First-order logic on finite trees. In Theory and Practice of Software Development, *Lect. Notes Comp. Sci*. **915** (1995), 125–139.

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

[RS59] M. O. Rabin and D. Scott, Finite automata and their decision problems, Rap. Tech., IBM J. Res. and Develop., 1959. Reprinted in Sequential Machines, E. F. Moore (ed.), Addison-Wesley, Reading, Massachussetts, (1964), 63–91.

[Re82] J. Reiterman, The Birkhoff theorem for finite algebras, *Algebra Universalis* **14** (1982), 1–10.

[SS07] S. Salehi and M. Steinby. Tree algebras and varieties of tree languages. *Theoret. Comput. Sci.* **377** (2007), 1–24.

[Sal09a] S. Salvati, Recognizability in the Simply Typed Lambda-Calculus, WoLLIC 2009, *Lect. Notes Comp. Sci*. **5514** (2009), 48-60.

[Sal] S. Salvati, On the Membership Problem for Non-Linear Abstract Categorial Grammars, *Journal of Logic, Language and Information*, to appear.

[Sch61] M.-P. Schützenberger, On the definition of a family of automata, *Information and Control* **4** (1961), 245–270.

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

[Si75] I. Simon, Piecewise testable events, in Proc. 2nd GI Conf., H. Brackage (ed.), Springer, *Lect. Notes Comp. Sci.* **33** (1975), 214–222.

[Sim88] I. Simon, Recognizable Sets with Multiplicities in the Tropical Semiring, in MFCS, Springer, Lect. Notes Comp. Sci. 324 (1988), 107–120,

[St94] H. Straubing. *Finite automata, formal logic, and circuit complexity*, Birkhäuser Boston Inc., Boston, MA, 1994.

[St02] H. Straubing, On logical descriptions of regular languages, in LATIN 2002, Springer, *Lect. Notes Comp. Sci.* **2286** (2002), 528–538.

[TT02] P. Tesson and D. Thérien, Diamonds are forever: The variety DA. In (G. Gomes, P.Ventura, J.-E. Pin, eds), *Semigroups, Algorithms, Automata and Languages*, World Scientific, (2002) 475-500.

[TT07] P. Tesson and D. Thérien, Logic meets algebra: the case of regular languages, *Logical Methods in Computer Science* **3** (2007), 1-37.

[TW02] D. Thérien and T. Wilke, Temporal logic and semidirect products: an effective characterization of the Until hierarchy, *SIAM Journal on Computing*, **31** (2002), 777-798.

[TW04] D. Thérien, T. Wilke, Nesting Until and Since in Linear Temporal Logic, *Theory Comput. Syst.* **37** (2004), 111-131.

[T84] W. Thomas. Logical aspects in the study of tree languages. In Colloquium on Trees and Algebra in Programming (ICALP'84), pages 31–50, 1984.

[T87] W. Thomas. On chain logic, path logic, and first-order logic over infinite trees. In Logic in Computer Science, pages 245–256, 1987.

[T97] W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors,

Handbook of Formal Languages, volume III, pages 389-455. Springer, New York, 1997.

[vM84] J. van Mill, An introduction to , in Kunen, Kenneth; Vaughan, Jerry E., *Handbook of Set-Theoretic Topology*, North-Holland, (1984), 503–560.

[W93] T. Wilke. Algebras for classifying regular tree languages and an application to frontier testability. In ICALP'93, volume 700 of *Lect. Notes Comp. Sci.*, pages 347–358, 1993.

[Via01] V. Vianu. A web odyssey: From CODD to XML. In PODS'01. ACM, 2001.