Automatic Presentations of Infinite Structures

Vince Bárány

Automatic Presentations of Infinite Structures

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Vince Bárány, M.Sc.
aus Budapest, Ungarn

Berichter: Universitätsprofessor Dr. Erich Grädel
Universitätsprofessor Dr. Jean-Éric Pin

Tag der mündlichen Prüfung: 5. September 2007

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar.

Abstract

The work at hand studies the possibilities and limitations of the use of finite automata in the description of infinite structures. An automatic presentation of a countable structure consists of a labelling of the elements of the structure by finite words over a finite alphabet in a consistent way so as to allow each of the relations of the structure to be recognised, given the labelling, by a synchronous multi-tape automaton. The collection of automata involved constitutes a finite presentation of the structure up to isomorphism. More generally, one can consider presentations over finite trees or over infinite words or trees, based on the appropriate model of automata. In the latter models, uncountable structures are also representable.

Automatic presentations allow for effective evaluation of first-order formulas over the represented structure in line with the strong correspondence between automata and logics. Accordingly, automatic presentations can be recast in logical terms using various notions of interpretations. The simplicity and robustness of the model coupled with the diversity of automatic structures makes automatic presentations interesting subject of investigation within the scope of algorithmic model theory.

Although automata have been in use in representations of infinite structures in computational group theory, in the analysis of numeration systems and finitely generated infinite sequences as well as in the theory of term rewriting systems, a systematic investigation of automatic structures using model theoretic methods has only just begun in the last twelve years.

There are two main lines of research in this field. One would like to have a classification of automatic models of certain first-order theories of common interest, such as linear orderings, trees, boolean algebras, groups, etc. Though efforts aimed at obtaining structure theorems have produced considerable advancement in recent years, this programme is still in an early stage. Even further lacking is our understanding of the richness of automatic presentations of key individual structures. A prominent result in this area is the deep theorem of Cobham and Semenov concerning numeration systems. In this style, one would like to know the degree of freedom in choosing automatic presentations of a particular structure.

In this thesis we present contributions to both of these problem areas. We also study restricted notions of presentations and clarify the relationship of automatic presentations over finite and infinite words. The peculiarities of using automata to represent structures up to isomorphism introduce problems out of the range of classical automata theory. We present some new techniques developed to tackle these difficulties.

Zusammenfassung

In dieser Arbeit werden die Möglichkeiten zur Darstellung unendlicher Strukturen mithilfe von endlichen Automaten sowie die Grenzen dieser Methode untersucht. Eine automatische Darstellung einer abzählbaren Struktur besteht aus einer Beschriftung der Elemente der Struktur mit endlichen Wörtern über einem endlichen Alphabet in einer konsistenten Art und Weise, so dass jede Relation der Struktur, in der gewählten Beschriftung, sich durch einen synchronen vielbändigen Automaten erkennen lässt. Ein Tupel geeigneter Automaten, einer für jede Relation, liefert eine endliche, bis auf Isomorphie eindeutige Beschreibung der Struktur. Allgemeiner kann man Darstellungen über endlichen Bäumen oder über unendlichen Wörtern oder Bäumen betrachten. Letztere sind auch geeignet überabzählbare Strukture zu beschreiben.

Infolge klassischer Korrespondenzen zwischen Logiken und Automaten wird eine algorithmische Auswertung logischer Formeln erster Stufe über jeder einzelnen durch Automaten dargestellten Struktur möglich. Ferner kann man automatische Präsentationen in logische Interpretationen übersezten bzw. als solche wahrnehmen. Die Einfachheit und Robustheit dieses Modells und die Vielfalt automatischer Strukturen motivieren eine ausführlichere Untersuchung automatischer Präsentationen in Rahmen der algorithmischen Modelltheorie.

Obwohl Automaten längst zur Darstellung unendlicher Strukturen in diversen Bereichen, u.a. in der algorithmischen Gruppentheorie, Zahlensysteme, endlich generierten undendlichen Folgen und Termersetzungssysteme in Gebrauch gewesen sind, wurde erst vor etwa zwölf Jahren eine systematische Untersuchung automatischer Strukturen mithilfe modelltheoretischer Methoden veranlasst.

Zwei wichtige Forschungsrichtungen werden in diesem Bereich sichtbar. Einerseits wird eine Klassifizierung automatischer Modelle bestimmter Theorien erster Stufe von allgemeinerer Bedeutung, wie z.B. lineare Ordnungen, Bäume, Boolsche Algebren, Gruppen usw. angestrebt. Trotz anhaltender Bemühungen struktursche Sätze zu finden und früher Erfolge befindet sich dieses Programm noch in der Anfangsphase. Noch mangelhafter ist unser Verständnis der reichen Möglichkeiten verschiedener automatischer Darstellungen einzelner Strukturen von zentraler Bedeutung. Ein prominentes Ergebnis in diesem zweiten Bereich ist der tiefgehende Satz von Cobham und Semenov über wohl bekannte Zahlensysteme. In dieser Tradition wollen wir den Freiheitsgrad der Wahl einer automatischen Präsentation gewisser Strukturen genauer verstehen.

In dieser Dissertation werden Beiträge zu den beiden erwähnten Problembereichen vorgelegt, mit Schwerpunkt auf dem letzteren. Ferner werden eingeschränkte Präsentationen untersucht und das Verhältnis automatischer Darstellungen über endlichen Wörtern im Vergleich zur Präsentationen mit unendlichen Wörtern geklärt. Die Eigentümlichkeiten des Gebrauchs von Automaten zur Darstellung von Strukturen bis auf Isomorphie erzeugen Probleme ausserhalb der Reichweite klassischer Automatentheorie. Es werden einige neue Techniken zur Bewältigung dieser Schwerigkeiten präsentiert.

Acknowledgement I am most grateful to Erich Grädel for giving me the opportunity to pursue my interests, for introducing me into an inspiring international community of researchers, and for his continued support throughout my work. For the excellent working and learning environment I am equally grateful to Wolfgang Thomas, and also to Renate Eschenbach-Thomas and her team for creating a wonderful atmosphere at the Informatik Bibliothek and for acquiring even the rarest of references.

I have gained much from collaboration with Łukasz Kaiser, Christof Löding, Sasha Rubin, and Olivier Serre. In addition I thank Arnaud Carayol, Didier Caucal, Thomas Colcombet, and Luc Segoufin for their illuminating thoughts which have contributed to this thesis in many ways.

Very special thanks to my brother-in-arms Łukasz once more for his insightful remarks at every stage of my work and for his companionship during the last forty some months. Similarly, to Sasha, for always being ready to discuss whatever i had on my mind. Additionally, I am particularly grateful to Arnaud Carayol, Tobias Ganzow, Christof Löding and Michael Ummels for careful corrections to this text.

I thank my friends and colleagues Alex, Dietmar, Jan, Kari, Michael, Nico, Philipp, Stefan, also Bernd, Diana, Frédéric, Henrik, Roman, Wenyun, the respected members of the hungarian football team: Dénes, Feri, Gergő, Norbert, Roland, Tamás, as well as my dear friends Balázs, Kata, Mathis, Jutka, Petra, and the Breuer family heartily for making me feel at home in Aachen. My heart goes to those at home or abroad Ákos, András, Misi, Orsi, Tamás, and to my loved ones for spoiling me with their love, trust and constant encouragement and to Panni for all that we share.

Contents

List of Figures

1 Introduction

1.1 From finite to algorithmic model theory

The 1980s and early 1990s have seen the emergence and rapid development of *finite model theory* as a branch of mathematical logic with deep connections to theoretical computer science. Relational databases have provided an important source of motivation and field of application. In this setting, the correspondence between models and databases, respectively logics and query languages is a very natural one. *Descriptive complexity theory* was born out of definability questions in finite model theory. In this area, a number of deep connections to computational complexity theory have been unearthed [Imm99, GKL$^+$07]. Finite model theory has thus established itself as an important research field within the widely general scopes of mathematical foundations of computer science and logic in computer science [EF95, Lib04, GKL$^+$07].

There is, however, no reason to stop here. Design and verification of infinite state reactive systems, as well as constraint databases [KLP00] and knowledge bases are important application areas for a model theory of *finitely presentable infinite structures*. Of course, the domain of infinite structures considered has to be chosen with care for logical problems to be amenable to algorithmic solutions. Accordingly, differences in motivation have lead to the introduction of a multitude of classes of finitely presentable structures as suitable domains of applications in computer science.

Classes of finitely presentable structures of special interest in mathematics and computer science include groups and semigroups [CEH$^+$92, CRRT01], numeration systems [Fro02] and infinite sequences [AS03], databases [KLP00, BLSS03] and transition graphs of infinite state processes [May00, Cau03].

Finitely presentable infinite structures

The programme of *algorithmic model theory* is to extend the range of applications of model theory in computer science from finite structures to various classes of finitely presentable infinite structures enjoying basic decidability- and closure properties depending on the intended applications. In particular,

- the structures should have decidable first-order or monadic second-order theories, or something in between, e.g. first-order with reachability;
- the class of structures should be closed under basic operations and/or logical interpretations, e.g. definable extensions.

The notion of recursive structures, though well established, is far too general yielding undecidability of small fragments of first-order logic. To meet the above aims one must severely restrict the notion of computation allowed in the definition of structures. Typically, this means using some model of *finite automata* or very restricted forms of rewriting. Although there are a few purely model-theoretic techniques yielding decidability, such as the Feferman-Vaught technique, the composition method, and quantifier elimination, the most successful and broadly applicable methods are based on finite automata theory [Tho97, GTW02].

Finite presentations may involve logical interpretations, finite axiomatisations, rewriting of terms, trees, or graphs, equational specifications, the use of synchronous or asynchronous automata, etc. The various approaches can be classified according to the following disciplines:

internal: a structure is represented by explicitly describing an isomorphic copy over a universe of finite or infinite words, trees or terms and by rewriting rules or by finite automata on these objects which compute the individual relations;

algebraic: a structure is represented as the least solution of a finite set of recursive equations in an appropriately chosen algebra of finite and countable structures;

logical: structures are described by interpreting them, using a finite tuple of formulas, in a fixed structure. A different approach consists in (recursively) axiomatising the isomorphism class of the structure to be represented.

transformational: structures are defined by sequences of prescribed transformations applied to a finite structure. Transformations can be transductions, unfolding or some kind of iteration, logical interpretations, etc.

The latter two approaches thus overlap somewhat. Also, alternative to the algebraic approach one can take a *generative* view of the solution process of an equational system. This method consists in converting the equational system into an appropriate *deterministic grammar* generating the solution graph. The grammar can thus be seen as a finite presentation of the graph.

1.1.1 Automatic structures

The most general forms of internal presentations meeting the requirements stated above are based on finite automata. Here one has a choice among models of automata on finite or infinite words or trees. The principle idea of providing an *automatic presentation* of a given structure is to identify its elements with words or trees, according to the automaton model chosen, in a consistent way as to allow the recognition of both the set of representants of the structure's universe as well as its functions and relations by appropriate automata. Assuming a structure comprising only a finite number of functions and relations, every such collection of finite automata thus constitutes a finite presentation of the structure, which is henceforth said to be automatic.

The advantage of having an automatic presentation of a structure lies in the fact that first-order formulas can be effectively evaluated using basic automata constructions. The first-order theory of every automatic structure is thus decidable. Moreover, using the same construction, one can effectively transform every first-order interpretation over a given automatic structure into an automatic presentation of the interpreted structure. Thus meeting our requirements, automatic presentations provide a robust framework eligible for investigation in algorithmic model theory.

Although the notion of automatic presentation on infinite trees subsumes the other models, presentations on finite trees (term trees) suffice for most purposes, as far as countable structures are considered. The class of tree automatic structures is slightly more robust than that of word-automatic structures, being closed e. g. under weak direct products. However, word automatic structures form a class already too rich in some sense. For instance, configuration graphs of Turing machines are easily seen to be automatically presentable on finite words. This implies that extensions of first-order logic with reachability or transitive closure operators or any other means of iteration are generally undecidable on automatic structures. Therefore, automatic presentations are unsuitable for modelling infinite state systems for verification purposes. For this reason it is meaningful to consider subclasses obtained by restricting presentations in one way or another.

A bit of history

The notion of automatic presentations on finite words and on ω-words first appeared in [Hod82, Hod83]. Hodgson identified basic properties of structures presentable in this way, namely, the decidability of their first-order theories and their closure under taking direct products. The first-order decidability result is a straightforward consequence of the closure and decidability properties of the automaton model. The notion of automatic presentations on finite trees and the same approach to decidability is manifest in the work of Dauchet and Tison [DT90].

In their seminal paper [KN95], Khoussainov and Nerode reintroduced automatic structures as a robust subclass of recursive structures and initiated their systematic investigation in the style of model theory. A further boost to the establishment of this theory was provided by the thorough work of Blumensath [Blu99] and Gräedel [BG00, BG04] who characterised automatic structures in terms of interpretations, studied automaticity of certain algebraic structures, and the complexity of model checking fragments of first-order logic on automatic structures. Since then Colcombet has given an alternative logical as well as an equational characterisation of automatic structures [Col04b, Col04a]. Diverse contributions to this field are to be found in the work of Kuske and Lohrey [Kus03, Loh03, KL06]. The programme of proving structure theorems characterising automaticity of various algebraic structures was carried on by Khoussainov, Nies, Rubin and Stephan in a series of papers [KR01, KRS03c, KR03, KNRS04, KRS05]. Other key results have been obtained by Delhommé, Goranko and Knapik [DGK], and by Oliver and Thomas [OT05]. For a concise reference on structural theorems we recommend the PhD thesis of Rubin

[Rub04] and the forthcoming survey [Rub07].

Prior to and independently of the above line of work, specialised notions of automatic presentations have been employed in the mathematical fields of computational group theory, symbolic dynamics [BP97], numeration systems [Fro02], and infinite sequences [AS03]. The latter are tightly related, and are concerned with "natural" automatic presentations of the ordered sets of the reals or the naturals with or without addition [BHMV94, Fro02]. The theory of automatic groups was developed in the 1980's by Cannon, Epstein, Thurston et al. based on a geometric approach. Tailored for the application, they consider naturally restricted automatic presentations of the Cayley graph of a group associated to a finite set of generators. However, the choice of generators is irrelevant, making this a robust notion. Virtually abelian groups and Gromov's word hyperbolic groups constitute important examples of automatic groups in this sense. Major results of this programme are presented in [CEH$^+$92], cf. also the introductions by Farb [Far92] and by Choffrut [Cho02]. More recently, this notion has been extended to monoids and semigroups [CRRT01] and has caught on considerable attention [HKOT02, SS04].

In light of the above we dare to say that automatic structures occupy a central place in algorithmic model theory.

Challenges

The work presented in this thesis has been chiefly motivated by the problems of proving non-automaticity of structures, *classifying automatic presentations* of some well-known structures, and of identifying properties invariant under the choice of a particular automatic presentation of individual structures. Let us point out once again, that many of the mentioned works concern only specialised or naturally restricted automatic presentations of the structures involved. In contrast to this, we pursue a description of *all* automatic presentations of particular structures. There are two major sources of difficulty.

Automata theory delivers elegant solution of key problems in logic [Gur85, Tho97, GTW02]. However, automata theory has been traditionally concerned with languages, that is, unary relations. In automatic presentations we make use of automata working on tuples of words or trees. Technically, this can be reduced to the unary case by forming a convolution of the components to produce a combined word or tree. Alternatively, automata on tuples of words can be intuitively thought of as automata with multiple tapes. The study of this model was initiated by Elgot and Mezei in [EM65]. Despite of this early start, insightful results on multi-tape automata are extremely scarce.

The true difficulty of our tasks is, however, due to the extraordinary circumstance that we are engaging in an investigation of automaton-recognisable relations (structures) *modulo isomorphism*. In [KNRS04] Khoussainov et al. write

> *...the Σ^1_1-completeness of the isomorphism problem of the class of all automatic structures tells us that the language of first-order arithmetic is*

not powerful enough to give a structure theorem for the class of all automatic structures. In other words we should not expect a 'nice' structure theorem for the class of all automatic structures.

Nonetheless, for many common algebraic classes there are simple conditions, in some cases full structural characterisations of automaticity [Rub07].

The analysis of automatic presentations thus provides entirely new challenges for automata theory. In [Bár06b] and in [BKR07], the latter in joint work with Łukasz Kaiser and Sasha Rubin, we have developed new techniques for meeting these challenges. These results are also presented in this thesis.

1.1.2 Transition graphs of infinite state processes

A key application area as well as a source of motivation for algorithmic model theory is that of *verification of infinite state processes*. Although this discipline is fundamentally different from that of representation of infinite algebraic or data structures and does not constitute the topic of this thesis, we feel compelled to mention some of the aspects in algorithmic model theory pertaining to modelling and verification of infinite state processes.

In the context of verification one is naturally interested in the dynamic behaviour of processes. Therefore, one considers structures (processes) equivalent not only when they are isomorphic, but when they share the same behaviour, or when they are observationally equivalent. This is most appropriately modelled by some notion of bisimulation equivalence, which is strongly related to indistinguishability in modal logics. To capture dynamic behaviour one has to employ logics powerful enough to express some form of recursion, e.g. reachability, transitive closure, fixed points etc. To retain decidability the types of processes are necessarily constrained. Best fitting this description is the theory of process rewriting [May98, May00] encompassing the well-known frameworks of Petri nets, process algebra and pushdown processes, that are internal or algebraic in manner according to the above classification. Ramifications of this approach constitute the theme of Otto's programme of "domain specific algorithmic model theory" [Ott01].

In the domain of sequential processes the most general frameworks known are the hierarchies of configuration graphs of *higher-order pushdown automata*, as well as of functions defined by *higher-order recursion schemes* [Cau02, KNU02, Ong06, HMOS]. While higher-order pushdown automata provide an internal presentation of the former graphs, an alternative transformational approach consists in applying a sequence of MSO-*compatible transformations* (interpretations or transductions, substitutions, unfolding [CK02]) to a finite graph. In this case the sequence of particular transformations together with the initial graph constitute a finite presentation. By one of the strongest decidability results in logic, Muchnik's generalisation of Rabin's tree theorem based on automata and infinite games, the configuration graphs of higher-order pushdown automata are known to have a decidable monadic second-order theory [Wal02, GTW02]. Processes defined by simply-typed recursion schemes have recently received a revived attention. Trees obtained by unfolding higher-order recursion schemes form a hierarchy extending corresponding levels of the pushdown hierarchy [KNU02, Ong06]. Verification of these processes involve such diverse tools as game semantics, collapsible pushdown automata, parity games, and tree automata techniques [HMOS]. Only the first few levels of these hierarchies are well understood and little is known regarding their relationship to other classes of graphs.

In addition to the above, we will briefly encounter graphs defined by ground rewriting on trees, respectively, on terms. These graphs are of course tree automatic, and the latter also allow equational presentations in a suitable algebra of finite and countable graphs.

The exact relationship of most of the above mentioned classes to that of automatic structures is not yet known. Also, in the cases understood, we have no "natural" characterisation of the intersections. We will present some results and ideas contributing to a clearer understanding of this issue.

1.2 Outline of the thesis

The dissertation is organised into seven chapters briefly summarised below.

In **Chapter 2** we review basic notions of the theory of automata, semigroups, and logic, fixing notation and recalling some of the most essential and well-known facts that will be used throughout the text. An exception is Section 2.2.2 introducing semi-synchronous rational relations and presenting some of their basic properties, among them a solution to [Sak03, Probléme 6.3], based on the work of the author [Bár06b].

In **Chapter 3** we survey various classes of finitely presentable infinite structures with emphasis on automatic presentations. We have classified the different approaches to finite presentations of structures as being internal (e.g. in the case of automatic, prefix-recognisable, GTR, and rational graphs), logical (i.e. defined in terms of interpretations), algebraic (such as HR- and VR-equational graphs and their extensions, solutions of recursion schemes) or transformational (prominently Caucal's pushdown hierarchy). The presentation loosely follows this structure.

Section 3.1 begins with the definition of the four basic classes of automatic structures, followed by numerous examples, and a presentation of their fundamental properties, most notably decidability of the first-order theory of every automatic structure and closure of the classes under first-order interpretations. We briefly mention some decidable and undecidable problems on automatic structures, and continue with a discussion of the role of injectivity.

In Section 3.1.1 we present a result recently obtained in joint work with Łukasz Kaiser and Sasha Rubin [BKR07] establishing that all *countable ω-automatic structures are automatic*. More precisely, using the formalism of ω-semigroups, we show that every ω-automatic presentation of a countable structure (i.e. one in which elements of the structure are represented by ω-words) can be filtered in a regular fashion to yield an injective presentation, therefore also an automatic presentation (i.e. one over finite words). This complements the work of Kuske and Lohrey [KL06] and answers a question of Blumensath [Blu99].

In subsequent sections 3.2 – 3.4 we review both logical and equational characterisations of the classes of automatic structures, and discuss classes obtained by restricting, respectively, by generalising the notion of automatic presentations. Section 3.5 provides a very brief overview of rational graphs, Caucal's hierarchy of higher-order pushdown graphs, and solutions of higher-order recursive schemes.

We close Chapter 3 by giving a summary and landscape illustrating how the various notions of finite presentations compare to one another.

Chapter 4 is devoted to portrayal of techniques in the analysis of automatic presentations. There are two main issues that need to be investigated. One concerns the restrictions on the local structure imposed by the combinatorics of automaton recognisable relations, with the aim of proving that certain structures cannot be presented by finite automata. The other revolves around the problem of identifying characteristic features of all automatic presentations of a given structure aiming towards their complete classification.

In Section 4.1 we review the basic combinatorial arguments used to prove that certain structures are not automatically presentable. These methods are based on bounds on the growth of the number of definable elements or sets as certain parameters are varied. The applicability of these techniques is, however, very limited. New or refined tools are needed to prove non-automaticity of some of the more notorious and stubborn examples, such as $(\mathbb{Q}, +)$ or the GTR structure generated by the rule $c \mapsto f(c, c)$ [Löd03].

In Section 4.2 the notion of *equivalence of automatic presentations* is introduced as a basic tool in the classification of automatic presentations of individual structures. As our main technical result we prove that two presentations are equivalent if and only if the mapping translating names of elements from one presentation into the other is computable by a *semi-synchronous transducer*. The latter is a rational transducer operating in a synchronous fashion on blocks of symbols, with fixed block sizes on the input- and output tapes. The notion of semi-synchronous transducer appears in [Sak03], but has been first studied by the author [Bár06b] in this connection. As a consequence of this characterisation we obtain that the complete structures of Blumensath and Grädel (those studied by Elgot and Rabin), those of Büchi-Boffa-Bruyére, as well as that of Colcombet are *rigidly automatic* meaning that all of their automatic presentations are equivalent.

Chapter 5 is to a large extent based on the paper [Bár06a] with some extensions and minor modifications. It is devoted to the investigation of *automatically presentable infinite words* over a finite alphabet. More precisely, we study automatic presentations of expansions of the ordered set of naturals $(\mathbb{N}, <)$ by unary predicates. Motivation for this investigation is the structural simplicity potentially enabling a complete characterisation of automatic presentations, which, on the other hand, is matched by the richness and robustness of the class of words presentable by finite automata. We study properties of presentations involving a k-fold nested length-lexicographic ordering. Starting point is the observation that length-lexicographically presentable words are precisely those morphic, moreover, that many of the key features of morphic words (decidability of monadic second-order theory, closure properties) are derivable from their canonical automatic presentations. Indeed, there is a canonical way of transforming the morphisms defining a morphic word into an automatic presentation involving the length-lexicographic ordering.

The notion of *morphic words* is a classical one going back to Thue. Morphic words have been thoroughly studied in the context of combinatorics on words, have applications in formal language theory, numeration systems, number theory and appear in various disguises in different branches of mathematics. The length-lexicographic ordering is immanent in the presentation of generalised numeration systems.

We introduce a generalisation of morphic words to higher orders using a notion of morphism of level k stacks, and show, that for every k, the level k morphic words coincide with those representable using the k-fold nested length-lexicographic ordering. Underlining the robustness of these notions we show that for each k the class of level k morphic words is closed under transformations by deterministic generalised sequential machines. We prove that each of these word structures have a *decidable monadic second-order theory* and that the *hierarchy of higher-order morphic words* is strictly increasing with each level and is thus infinite. Finally we show that for every k all level k morphic words are constructible on the $2k$-th level of pushdown hierarchy.

Our results thus generalise those of Pansiot [Pan84], Carton and Thomas [CT02], Caucal [Cau02], and of Rigo and Maes [RM02] related to morphic words. We close the chapter with a discussion of further generalisations and key open questions.

In **Chapter 6** we explore the use of transductions as devices transforming one automatic presentation into another. Transductions constitute an important tool in the theory of formal languages, and have been thoroughly studied also for their own right. There are numerous classes of transductions, key properties of which are well understood. Accordingly, there is a great volume of literature on various problems related to transductions, providing valuable asset in our effort.

Our interest in transduction is motivated by the fact, that, given a natural automatic presentation of a structure, questions concerning the existence of another presentation having certain properties can be rephrased as such concerning the existence of transductions satisfying some *regularity constraints*. Note that we are primarily interested in bijective transductions.

Of utmost utility in formal language theory are *continuous transductions* satisfying the constraint that the inverse image of every regular language must again be regular. Constraints naturally arising in the context of automatic presentations concern regularity of multi-ary relations. In Section 4.2 we have encountered regularity constraints capturing the notion of equivalence of presentations. These constraints are in some sense maximal.

In Section 6.2 we show how a result of Colcombet yields a characterisation of transductions inversely mapping all prefix-recognisable relations to regular ones. In addition to continuity the single constraint that the inverse image of the prefix relation be regular is sufficient. We call these mappings *PR-transductions*. It is then observed that with some adjustments the embeddings of k-morphic words into the pushdown hierarchy are particular PR-transductions, yielding an almost effortless, though less in-depth proof of the results of Section 5.4.

In Section 6.4 we introduce *generalised run-length encodings* based on automatic presentations of infinite words. These are transductions, which can be considered as compression schemes and can be computed sequentially, however, using unbounded memory. Results of Chapter 5 imply that each such compression scheme associated to a k-morphic word is a PR-transduction. Consequently, we can provide automatic presentations of arbitrary prefix-recognisable structures based on any of these compression schemes. We conclude that prefix-recognisable structures have more than one automatic presentation up to equivalence.

In **Chapter 7** we consider the problem of understanding what different automatic presentations of individual structures have in common. More precisely, which relations over a given structure are *intrinsically regular*, meaning, invariantly regular in all automatic presentations of the structure.

Intrinsic regularity was introduced in [KRS04], where natural as well as "unnatural" automatic presentations of simple fragments of Presburger arithmetic were analysed, respectively, forged in an attempt to isolate the intrinsically regular relations over these structures. The difficulty of this task cannot be overestimated, as prompted by the pathological presentations of the innocent looking successor structure $(\mathbb{N}, \mathrm{succ})$.

A natural and fundamental question asked by Khoussainov et al. is whether there is a logical characterisation of intrinsically regular relations over arbitrary automatic structures. In his dissertation, Rubin asked whether first-order logic enhanced with modulo counting quantifiers and with the infinity quantifier is expressive enough. Our contribution to this problem is the observation that, on the one hand, relations defined by order-invariant formulas are intrinsically regular, while on the other hand, there are order-invariantly definable relations that are not definable using generalised counting quantifiers. These results were published in [Bár06b] and prompted an investigation of what we now call *regularity preserving generalised quantifiers*.

2 Preliminaries

2.1 Words and trees

Let Σ be a finite alphabet. Σ^* and Σ^ω denote the set of finite words, respectively, the set of words of length ω over Σ. The length of a word $w \in \Sigma^*$ is written $|w|$, the empty word is ε, for every $0 \leq i < |w|$ the ith symbol of w is written as $w[i]$, and when $\mathcal{I}$ denotes some interval of positions then $w\mathcal{I}$ (e.g. $w[n, m)$) is the factor of w on these positions. Note that we start indexing with 0. Accordingly, for every $n \in \mathbb{N}$, we let $[n] = \{0, \ldots, n-1\}$.

Subsets of Σ^* and of Σ^ω are languages of finite, respectively, ω-words. The class of regular languages is a fundamental, robust, most thoroughly studied family of languages. As it is well known these are the languages recognised by finite automata and by finite semigroups, described by regular expressions, and defined by monadic second-order formulas in the signature of words (see below). We will also be interested in relations on words accepted by finite automata. Next we will briefly recall these and related notions while fixing notation.

For a language $L \subseteq \Sigma^*$ let $L_{=n} = L \cap \Sigma^n$ and $L_{\leq n} = L \cap \Sigma^{\leq n}$ denote the set of members of L of length precisely n and at most n, respectively. Further, let $\mathsf{Pref}(L)$ be the set of prefixes of words in L. Note that $\mathsf{Pref}(L)$ is regular for every regular L. The *growth* of a language $L \subseteq \Sigma^*$ is the function $g_L : \mathbb{N} \to \mathbb{N}$ mapping each $n \in \mathbb{N}$ to the number of words in L of length n, that is $g_L(n) = |L_{=n}|$.

Trees

We consider finite and infinite trees with bounded branching. For our purposes the following definition suffices. A Σ-labelled tree is a function $t : \mathsf{dom}(t) \to \Sigma$, such that $\mathsf{dom}(t) \subseteq [r]^{\leq\omega}$ is 1) non-empty, 2) prefix-closed, and 3) if $xl \in \mathsf{dom}(t)$ for some $x \in [r]^*$ and $l < r$ then $xj \in \mathsf{dom}(t)$ for every $0 \leq j < l$. A tree t is finite if $\mathsf{dom}(t)$ is finite.

Nodes of a tree t are elements of $\mathsf{dom}(t)$. The nodes are partially ordered by the prefix (ancestor) relation $\preceq$. The root of a tree is the single minimal element, ε, of its nodes. A node is a leaf if it is maximal with respect to the prefix relation. A prefix of a tree t is a restriction $t|_P$ to a non-empty prefix-closed subset $P \subseteq \mathsf{dom}(t)$. An antichain is a set of nodes pairwise incomparable by $\preceq$.

There is a natural way to represent a subset $L \subseteq [r]^*$ as a $\{0,1\}$-labelled tree t_L, the characteristic tree of L, with $\mathsf{dom}(t_L) = [r]^*$ and $t_L(x) = 1$ iff $x \in L$. A tuple $(L_i)_{i<n}$ of subsets $L_i \subseteq [r]^*$ can similarly be identified with a $\{0,1\}^n$-labelled tree obtained by overlapping the t_{L_i}.

This representation of subsets of trees facilitates the well-known correspondence between tree automata and monadic second-order logic [Tho97]. Next we will recall key notions and results of automata theory, semigroups, and related logics.

2.2 Finite automata on finite words

A finite labelled transition system (TS) is a tuple $\mathcal{T} = (Q, \Sigma, \Delta)$, where Q is a finite, nonempty set of states, Σ is a finite set of labels, and $\Delta \subseteq Q \times \Sigma \times Q$ is the transition relation. $\mathcal{T}$ is called deterministic (DTS) if Δ is a function of type $Q \times \Sigma \to Q$, in this case we write δ instead of Δ, and δ^* for the unique homomorphic extension of δ to all words over Σ.

A *finite automaton* (FA) is a finite transition system together with sets of initial and final states $\mathcal{A} = (\mathcal{T}, I, F) = (Q, \Sigma, \Delta, I, F)$. $\mathcal{A}$ is *deterministic* (DFA) if $\mathcal{T}$ is determinitic and I contains a single initial state q_0.

The language $L(\mathcal{A})$ *recognised* by an FA $\mathcal{A}$ as above is the set of words w that label an accepting path in its graph, i.e. a path leading from an initial to a final state.

The *completion* of a DFA $\mathcal{A}$ is the DFA $\overline{\mathcal{A}}$ obtained by introducing a new state $\perp$ and setting it the target of all yet undefined transitions. Thus, the transition function $\overline{\delta}$ of $\overline{\mathcal{A}}$ is defined for all pairs (q, a) with $q \in Q \cup \{\perp\}$. Note that $L(\overline{\mathcal{A}}) = L(\mathcal{A})$.

2.2.1 Multi-tape automata

One can consider a finite automaton recognising a regular set of words as a finite presentation of this set. A natural extension of this concept is to consider multi-tape finite automata to represent relations on words in a similar manner. This raises the issue of how the automata should be allowed to access their individual tapes, e.g. in a synchronous or asynchronous fashion. Different operation modes give rise to different classes of relations, most notably to the classes of *recognisable*, *synchronised rational* and *rational* relations. These automata classes and their algebraic analogues have been studied in [EM65, Ber79, FS93].

Synchronised multi-tape automata constitute the fundament of the notion of automatic presentations, while rational transductions recognised by two-tape finite automata will be one of our main tools in their investigation. Let us therefore recall these basic definitions.

We consider relations on words, i.e. subsets R of $(\Sigma^*)^n$ for a finite alphabet Σ and some $n > 0$. *Asynchronous* n-tape automata accept precisely the *rational relations*, i.e., rational subsets of the product monoid $(\Sigma^*)^n$. A relation $R \subseteq (\Sigma^*)^n$ is *synchronised rational* [FS93] if it is accepted by a *synchronous* n-tape automaton. Synchronised rational relations are also called *regular relations* (cf. [KR03]), an alternative we shall frequently use as well. Finally, $R \subseteq (\Sigma^*)^n$ is *semi-synchronous rational* if it is accepted by an n-tape automaton reading each of its tapes at a fixed speed. This is made more precise below.

2.2.2 Semi-synchronous Rational Relations

The class of semi-synchronous rational relations has been introduced by Sakarovich in [Sak03] and independently by the author in [Bár06b], where the importance of semi-synchronous transductions in the study of automatic presentations was shown. Those results are presented in Section 4.2. Here we give a formal definition of semi-synchronous rational relations as well as their most rudimentary properties, essentially identical to those of regular relations, with the notable exception of Theorem 2.2.4 below.

Definition 2.2.1 (Semi-synchronous rational relations, cf. [Sak03, p. 660], [Bár06b]). Let $\Box$ be a special end-marker symbol, $\Box \notin \Sigma$, and $\Sigma_\Box = \Sigma \cup \{\Box\}$. Let $\alpha = (a_1, \dots, a_n)$ be a vector of positive integers and consider a relation $R \subseteq (\Sigma^*)^n$. Its *α-convolution* is $\boxtimes_\alpha R = \{(w_1\Box^{m_1}, \dots, w_n\Box^{m_n}) \mid (w_1, \dots, w_n) \in R$ and the m_i are minimal, such that there is a k, with $ka_i = |w_i| + m_i$ for every $i\}$. This allows us to identify $\boxtimes_\alpha R$ with a subset of the monoid $((\Sigma_\Box)^{a_1} \times \cdots \times (\Sigma_\Box)^{a_n})^*$. If $\boxtimes_\alpha R$ thus corresponds to a regular set, then we say that R is *α-synchronous (rational)*. R is *semi-synchronous* if it is α-synchronous for some α.

Intuitively, our definition expresses that although R requires an asynchronous automaton to accept it, synchronicity can be regained when processing words in blocks, the size of which are component-wise fixed by α. As a special case, for $\alpha = \vec{1}$, we obtain the regular relations. Recall that a relation $R \subseteq (\Sigma^*)^n$ is *recognisable* if it is saturated by a congruence (of the product monoid $(\Sigma^*)^n$) of finite index, equivalently, if it is a finite union of direct products of regular languages [FS93]. We denote by Rat, SRat, $\mathsf{S}_\alpha\mathsf{Rat}$, Reg, Rec the classes of rational, semi-synchronous, α-synchronous, regular, and recognisable relations, respectively. Sometimes we will give the underlying alphabet explicitly in brackets.

Evidently, $\mathsf{Reg} \subset \mathsf{SRat} \subset \mathsf{Rat}$ and both containments are strict as illustrated by the examples $\{(a^n, a^{2n}) \mid n \in \mathbb{N}\}$ and $\{(a^n, a^{2n}), (b^n, b^{3n}) \mid n \in \mathbb{N}\}$. In fact, the latter example witnesses the fact that semi-synchronous rational relations are strictly included in the class of deterministic rational relations. SRat is closed under complement but not under union, as also shown by the latter example. Obviously, for any fixed α the class of α-synchronous rational relations has all the nice properties of synchronised rational relations.

Proposition 2.2.2. $\mathsf{S}_\alpha\mathsf{Rat}$ is an effective boolean algebra for each α. The projection of every $\alpha\beta$-synchronous relation onto the first $|\alpha|$ many components, is α-synchronous.

Proposition 2.2.3. For every vector α of non-negative integers $\mathsf{S}_\alpha\mathsf{Rat}$ is closed under taking images (hence also inverse images) via semi-synchronous transductions.

Proof. Let T be a (p, q)-synchronous transduction, R an α-synchronous n-ary relation with $\alpha = (a_1, \dots, a_n)$. $T(R) = \{\vec{v} \mid \exists \vec{u} \in R \ \forall i \leq n \ (u_i, v_i) \in T\}$ is the projection of the $(pa_1, \dots, pa_n, qa_1, \dots, qa_n)$-synchronous relation $\{(\vec{u}, \vec{v}) \mid \vec{u} \in R, \ \forall i \leq$

n $(u_i, v_i) \in T\}$. Hence, by Proposition 2.2.2 and Theorem 2.2.4.*i*) below, $T(R)$ is α-synchronous. Closure under taking inverse images follows from the fact, that the inverse of a (p,q)-synchronous transduction is (q,p)-synchronous. $\square$

Observe that the composition of a (p,q)-synchronous and an (r,s)-synchronous transduction is (pr,qs)-synchronous, thus, the class of semi-synchronous transductions is closed under composition. We will next show that for (p,q)-synchronous rational transductions, with the exception of recognisable transductions, the ratio p/q is uniquely determined. To this end let us say that α and β are *dependent* if $k \cdot \alpha = l \cdot \beta$ for some $k, l \in \mathbb{N}$, where multiplication is meant component-wise. Then, comparing classes $\mathsf{S}_\alpha\mathsf{Rat}$ and $\mathsf{S}_\beta\mathsf{Rat}$ we observe the following "Cobham-Semenov-like" (cf. Theorem 4.3.4) relationship.

Theorem 2.2.4 (Cobham-Semenov-like relationship, [Bár06b, Car06]).
Let $n, p, q \in \mathbb{N}$ and $\alpha, \beta \in \mathbb{N}^n$.

i) If α and β are dependent, then $\mathsf{S}_\alpha\mathsf{Rat} = \mathsf{S}_\beta\mathsf{Rat}$.

ii) If (p,q) and (r,s) are independent, then $\mathsf{S}_{(p,q)}\mathsf{Rat} \cap \mathsf{S}_{(r,s)}\mathsf{Rat} = \mathsf{Rec}$.

Proof. *i*) Clearly, a relation R is α-synchronous if and only if it is $k\alpha$-synchronous for any $k \geq 1$. The claim follows.
ii) Recognisable relations are trivially α-synchronous for any α, therefore we only care for the other inclusion.

Let $R \in \mathsf{S}_{(p,q)}\mathsf{Rat} \cap \mathsf{S}_{(r,s)}\mathsf{Rat}$. We need to show, that R is a finite union of Cartesian products $A_i \times B_i$ of regular languages, in other terms that the following equivalence is of finite index.

$$x \sim x' \overset{\text{def}}{\iff} \forall y : R(x,y) \leftrightarrow R(x',y)$$

According to (1) R is both (pr,qr)- and (pr,ps)-synchronous, and by assumption $ps \neq qr$, w.l.o.g. $ps < qr$. Let us further assume for simplicity and w.l.o.g. that $pr = 1$ and let $k = ps$ and $l = qr$. Consider some $(1,k)$- respectively $(1,l)$-synchronous deterministic automata $\mathcal{A}$ and $\mathcal{A}'$ accepting R. Thus $\mathcal{A}$ is "slower" then $\mathcal{A}'$ in reading the second tape. Our first observation is confirmed by a straightforward pumping argument.

$$x \not\sim x' \Rightarrow \exists y : |y| < k(\max(|x|,|x'|) + C) \wedge R(x,y) \leftrightarrow \neg R(x',y) \qquad (*)$$

where $C = |\mathcal{A}|^2 + 1$.

The syntactic congruence of $\mathcal{A}'$ induces an equivalence of finite index on pairs of words $(u,z) \in \big((\Sigma \cup \{\square\}) \times (\Gamma \cup \{\square\})^l\big)^*$. ($(u,z) \approx_{\mathcal{A}'} (u',z')$ iff their actions on the states of $\mathcal{A}'$ are identical). Let K be the length of the longest word v such that $(v, \square^{l|v|})$ is the shortest such representant of its $\approx_{\mathcal{A}'}$-class.

Consider now any x long enough such that $\lceil (|x| + C)\frac{k}{l} \rceil + K < |x|$. During the run of $\mathcal{A}'$ on input (x,y) for any y shorter than $k(|x|+C)$, y will be completely read leaving a suffix v of x, v longer than K, unread. By replacing v with a shorter v' such that $(v, \square^{l|v|}) \approx_{\mathcal{A}'} (v', \square^{l|v'|})$ in x we obtain an x' shorter than x, which is by

($*$) $\sim$-equivalent to x. Thus we have shown that each $\sim$-class has a representant of bounded size, i.e. that there are finitely many such classes as required. $\square$

Adapting techniques from [Ber79, FS93], used to prove undecidability of whether a given rational relation is synchronised rational, we obtain the following undecidability results.

Theorem 2.2.5. *For any given $p, q \in \mathbb{N}$ the following problems are undecidable.*

i) Given a rational transduction $R \in \mathsf{Rat}$ is $R \in \mathsf{S}_{(p,q)}\mathsf{Rat}$?

ii) Given a rational transduction $R \in \mathsf{Rat}$ is $R \in \mathsf{SRat}$?

Proof. For i) the proof is essentially the same as for regularity, ii) requires, in addition, a slight adaptation of the technique. Let us therefore give a quick review. Given an instance $I = \{(u_i, v_i) \mid i < n\}$ of PCP consisting of pairs of words over some finite alphabet Γ we define $U = \{(ab^i, u_i) \mid i < n\}$ and $V = \{(ab^i, v_i) \mid i < n\}$. So it is clear that I has a solution iff $W = U^+ \cap V^+ \neq \emptyset$, where U^+ and V^+ are evidently rational. Although the class of rational relations is not closed under complementation, one can show that the complements $\overline{U^+}$ and $\overline{V^+}$ of U^+ and V^+, respectively, are in fact rational, hence is their union $\overline{W} = \overline{U^+} \cup \overline{V^+}$. A number of undecidability results follow from these observations (cf. [Ber79],[FS93]).

Note that in each pair of U and V the first component ab^i is used only to identify the index of the corresponding second component, their choice is irrelevant as long as they are distinct. Therefore, all of the previous remarks hold, in particular, for $U = U_{\vec{k}} = \{(ab^{k_i}, u_i) \mid i < n\}$ and $V = V_{\vec{k}} = \{(ab^{k_i}, v_i) \mid i < n\}$ for any sequence of naturals $\vec{k} = (k_1, \ldots, k_n)$. In [FS93] Frougny and Sakarovitch use this fact to show that for an appropriate choice of $\vec{k}$, $\overline{W}$ is regular iff $W = \emptyset$ iff I has no solution, which is undecidable.

A direct adaptation of their technique proves i). Indeed, for given p, q and instance I of PCP we chose distinct k_i such that $k_i \geq 2\frac{p}{q}\max(|u_i|, |v_i|)$ for all $i < n$. Assume $W = W_{\vec{k}} \neq \emptyset$. Let $(x, y) \in W$. Then $(x^m, y^m) \in W$ and $|x^m| \geq 2\frac{p}{q}|y^m|$ for any m. It follows from a direct adaptation of Proposition 4.1.1 that for any (p, q)-synchronous function f there exists a constant K such that $\big|q|x| - p|y|\big| \leq K$ for all $f(x) = y$. Therefore, since W is functional, it is not (p, q)-synchronous, hence, neither is $\overline{W}$. Thus, we see that $\overline{W}$ is (p, q)-synchronous iff I has no solution. This concludes the proof of undecidability of i).

To prove undecidability of ii) we give another variant of the previous reduction. Again, let I be a PCP instance over Γ. Let I' be a copy of I over an alphabet Γ' disjoint from Γ. Consider the PCP instance $I \cup I' = \{(u_i, v_i), (u'_i, v'_i) \mid i < n\}$ over $\Gamma \cup \Gamma'$. Let $U = \{(ab^i, u_i) \mid i < n\} \cup \{(a'b'^{2i+1}, u'_i)\}$, $V = \{(ab^i, v_i) \mid i < n\} \cup \{(a'b'^{2i+1}, v'_i)\}$, and $W = U^+ \cap V^+$ as above. If I has no solution then $W = \emptyset$, and if $(i_1, \ldots, i_t)$ is a solution of I with $y = u_{i_1} \cdots u_{i_t} = v_{i_1} \cdots v_{i_t}$ then there are $(x, y) \in W$ and $(x', y') \in W$ such that $|x'| = 2|x|$ and $|y| = |y'|$. Since W is functional, for the same reason as above, it can not be (p, q)-synchronous for any p and q. In other words it is not semi-synchronous, and hence neither is $\overline{W}$. Thus we have shown that the rational $\overline{W}$ is semi-synchronous iff I has no solution, which proves undecidability of ii). □

Theorem 2.2.4 provides solution to [Sak03, Probléme 6.3] and has independently been proved by the author [Bár06b] and by Carton [Car06].

2.2.3 Rational Transductions

A *transduction* is a binary relation $T \subseteq \Sigma^* \times M$ between a free monoid Σ^* and a monoid M, also considered as a mapping $T : \Sigma^* \to \mathcal{P}(M)$. We will mostly be concerned with transductions from words to (sets of) words, possibly over different alphabets, i.e. $M = \Gamma^*$. *Rational transductions* are those recognised by finite asynchronous 2-tape automata. They have been extensively studied in the context of formal language theory. For classical results we refer the reader to [Ber79] or to [Sak03].

Functional transductions $T : \Sigma^* \to \Gamma^*$, especially those semi-synchronous rational, will be of special interest to us as devices transforming one automatic presentation into another. Each of the families of rational transductions, functional transductions, respectively, semi-synchronous rational transductions is closed under composition. Whereas rational transductions preserve regularity as well as non-regularity of languages, but not of relations in general, it is easy to check that semi-synchronous rational transductions do preserve (non-)regularity of all relations of whatever arity. We will show (cf. Theorem 4.2.6) that semi-synchronous functional transductions are characterised by the key property of transforming regular relations to regular ones and non-regular relations to non-regular ones.

Generalised sequential machines, GSM's, are a restricted form of transducers having the distinctive feature of being input-driven and output-producing as opposed to being acceptors of input-output pairs. Of considerable interest computing functional transductions are *deterministic generalised sequential machines (DGSM)*. A DGSM $\mathcal{S} = (\mathcal{T}, q_0, \mathcal{O})$ consists of a DTS $\mathcal{T}$, an initial state q_0 and an output function $\mathcal{O} : Q \times \Sigma \to \Gamma^*$. The function computed by a DGSM maps a word, input to the machine, to the concatenation of the outputs produced in each state of the run while reading the input word. This mapping can, in a natural way, be extended to ω-words as well.

2.3 Finite Automata on infinite words

Finite automata on infinite words [1] are defined in the very same way as FA on finite words, only the notion of acceptance requires some thought. As a well-known matter of fact there are a number of reasonable acceptance criteria, defined in terms of the set of states $\mathrm{Inf}(\rho)$ occurring infinitely often in a run ρ, the prominent ones being

Büchi: given by a subset F of states, one of which must occur infinitely often;

[1] When we say infinite words we mean words of length ω.

co-Büchi: given by a subset C of states, none of which may occur infinitely often;

Rabin: given by pairs $\{(C_i, F_i)\}_{i<r}$ of co-Büchi and Büchi conditions; accepting runs must satisfy for some i both the i^{th} co-Büchi and the ith Büchi condition.

Streett: dual to the Rabin condition;

parity: a special case of the Rabin condition, also called Rabin-chain condition, with $E_0 \subset F_0 \subset E_1 \subset F_1 \subset \cdots \subset E_{r-1} \subset F_{r-1}$

Muller: a run ρ is accepting if $\mathrm{Inf}(\rho) \in \mathcal{F}$, where $\mathcal{F}$ is a given family of subsets of states.

It is a simple fact that non-deterministic Büchi automata are more powerful than deterministic ones when it comes to accepting languages of ω-words. Deterministic Büchi automata are not closed under complementation, but rather necessitate the dual co-Büchi acceptance. Similarly, straightforward dualisation of a Rabin condition is Streett and vice versa. Deterministic parity and Muller automata are easy to complement by dualising the acceptance condition. Based on (2.1) and using Ramsey's theorem, Büchi provided a non-constructive proof of complementation of non-deterministic Büchi automata. A fundamental theorem of McNaughton states that non-deterministic Büchi automata and deterministic Muller automata accept the same class of *ω-regular languages*. Safra gave an optimal construction of transforming non-deterministic Büchi automata into equivalent deterministic Rabin automata.

We will be using the fact that ω-regular languages are also defined by ω-regular expressions of the form $r_1 s_1^\omega + \cdots + r_k s_k^\omega$, where r_i, s_i are regular expressions. In other words, every ω-regular language L can be written as a finite union

$$L = \bigcup_i U_i V_i^\omega \tag{2.1}$$

where U_i and V_i are regular languages of finite words. ω-Regular languages have all the favourable properties of their junior fellows on finite words: they form an (effective) boolean algebra, are closed under (inverse) homomorphic mappings, etc.; emptiness is trivially decidable in any of the mentioned formalisms, in fact, by (2.1) every non-empty ω-regular language contains an ultimately periodic word uv^ω with bounded $|u|$ and $|v|$. For a comprehensive treatment of automata on infinite words we refer to [PP04].

In much the same way as over finite words, finite automata can be used to define relations on infinite words. We are interested here in the synchronised rational relations defined either by synchronised multi-tape FA or by single-tape FA reading the convolution of ω-words. The *convolution* $\otimes_k \vec{u}$ of $u_i \in \Sigma^\omega$, $i < k$ is defined as the letter-by-letter shuffle: $\otimes_k \vec{u}[nk + i] = u_i[n]$ for every n and $i < k$.[2] We refer to the

[2]Notice that we spare ourselves the awkward padding with blanks needed for finite words.

relations accepted this way as *ω-regular relations*, or just as regular relations, when the context is clear.

2.4 Finite automata on trees

Classical notions and results of formal language theory and automata theory have been generalised very early to labelled trees of bounded rank (branching degree). We refrain from giving a thorough introduction to automata on trees as we will mostly be using automata on finite and infinite words. Below, we only recall some of the most basic notions and results concerning tree automata. For a comprehensive introduction we refer to the books [GS84] and [CDG$^+$] and to the surveys [GS97, Tho97].

The algebraic approach of Mezei-Wright

From an algebraic point of view, Σ-labelled trees are terms, i.e. elements of the free algebra of function symbols $\Sigma \times [r]$ where r is the maximal rank of a node and (a, n) is an n-ary function symbol corresponding to a-labelled nodes of rank n. Words are simply unary terms in this context, and finite semigroups, the algebraic equivalents of finite automata, are finite algebras of unary functions generated by those associated to symbols of the alphabet. Using this analogy, automata on terms can be defined as finite algebras of the same signature. A set L of terms is then said to be recognisable if there is a finite algebra $\mathcal{A}$ of the same signature and a homomorphism φ from the free algebra of terms to $\mathcal{A}$ such that $L = \varphi^{-1}(\varphi(L))$.

This framework necessitates the use of constant(s) at the leaves of terms. Given a finite algebra $\mathcal{A}$ and a homomorphism φ as above, the image(s) of the constant(s) can be seen as "initial elements" of $\mathcal{A}$ in analogy to initial states of finite automata. Similarly, if φ accepts L as above, then the subset $\varphi(L)$ of $\mathcal{A}$ can be seen as a set of "final elements".

Observe, that this algebraic framework is in direct correspondence with finite automata working from right to left. These are of course equivalent to left-to-right automata as it is well known. On finite trees this analogy leads to the definition of bottom-up tree automata.

Bottom-up tree automata

Bottom-up tree automata are given by a finite set Q of states of which q_0 is initial and some $F \subseteq Q$ are final. The transition relation Δ comprises tuples $(q_1, \ldots, q_k, a, q)$ with the intended meaning that having arrived at states $q_1, \ldots, q_k$ on respective subtrees $t_1, \ldots, t_k$ of a tree $t = a(t_1, \ldots, t_k)$ the automaton can proceed to the root of t after entering state q. A bottom-up tree automaton is deterministic if q is uniquely determined by a and $q_1, \ldots, q_k$.

Bottom-up tree automata thus generalise right-to-left automata on words. The notions of run and acceptance are defined as expected. A language of labelled trees is defined to be *tree-regular* if it is accepted by some bottom-up tree automaton. Bottom-up tree automata can be determinised using the subset construction and, as hinted above, a set of term-trees is tree-regular iff it is recognised by a finite algebra under a homomorphism from the free algebra of terms.

Top-down tree automata

Bottom-up evaluation naturally only makes sense for finite trees. Top-down tree automata are defined as their bottom-up cousins with the difference that transitions $(q_1, \dots, q_k, a, q)$ are interpreted from right to left: mapping the state q and label a to tuple(s) $(q_1, \dots, q_k)$ of states in which the automaton is to proceed with the processing of corresponding subtrees. The conversion from non-deterministic top-down to and from non-deterministic bottom-up tree automata is thus a trivial act. The difference of the two approaches manifests itself in the fact that top-down tree automata cannot be determinised in general.

On the top side, top-down automata can be used as accepting devices for infinite trees. To this end, the acceptance condition has to be defined similarly to acceptance of infinite words. Given a (non-deterministic) top-down tree automaton $\mathcal{A}$ and an acceptance condition of Büchi-, Rabin-, parity-, or Muller style we say that the automaton accepts an infinite tree t if there is a run of $\mathcal{A}$ on t every path of which, seen as a word, satisfies the given acceptance condition. Regular languages of infinite trees are those accepted by (non-deterministic) tree automata with a Muller (equivalently with Rabin or parity) acceptance condition. Tree automata with Büchi condition are strictly weaker than Muller, Rabin, or parity tree automata on infinite trees.

Tree-regular relations

The concept of convolution of a tuple of words, defined as a kind of overlapping of its suitably padded components, can well in turn be extended to trees. This allows us to define tree-regular and ω-tree-regular relations.

The infinite r-padding of a finite or infinite tree of branching degree bounded by r is defined by hanging a copy of the uniformly $\Box$ labelled infinite r-ary tree as the ith subtree of every node missing an ith child node. The symbol $\Box$ is of course assumed to be distinct from all other labels. Thus, all r-padded trees have the same shape of a complete infinite r-ary tree, and are labelled with $\Sigma_\Box = \Sigma \cup \{\Box\}$. The convolution $\otimes\vec{t}$ of a tuple $\vec{t} = (t_1, \dots, t_n)$ of Σ-labelled trees, each of which is of branching degree bounded by r, is the $\Sigma_\Box^n$-labelled complete r-ary tree t obtained by overlapping the infinite r-paddings of $t_1, \dots, t_n$. More precisely, the label of t at node $x \in [r]^*$ is the tuple $(a_1, \dots, a_n)$ such that for every i either $a_i = t_i(x)$ or $\Box$ when $t(x)$ is undefined. When the t_i are finite, we define $\otimes\vec{t}$ to be finite by chopping off all uniformly $\Box^n$-labelled subtrees of the convolution of their infinite paddings.

A relation R of Σ-labelled $\leq r$-branching (infinite) trees is (ω-)tree-regular if the set $\otimes R = \{\otimes\vec{t} \mid \vec{t} \in R\}$ is (ω-)tree-regular.

2.5 Semigroups

A semigroup $(S, \cdot)$ is a set S equipped with an associative operation $\cdot$ usually denoted by juxtaposition. A monoid is a semigroup with a neutral element 1. Every semigroup S can be extended to a monoid S^1 by the addition of a neutral element, if needed. The set of finite nonempty words Σ^+ with concatenation is the free semigroup generated by Σ. With the addition of the empty word ε we obtain the free monoid Σ^* generated by Σ.

Morphisms of semigroups and monoids preserve the product as well as the identity. We denote by $\mathsf{Hom}(M, N)$ the set of homomorphisms from the monoid M to N. Each $\varphi \in \mathsf{Hom}(\Sigma^*, \Sigma^*)$ can be specified by the images $\varphi(a)$ of individual symbols $a \in \Sigma$. The length of φ, denoted $|\varphi|$, is the maximum of all the $|\varphi(a)|$, and φ is *uniform*, when $|\varphi(a)| = |\varphi|$ for every $a \in \Sigma$.

Monoid homomorphisms into a finite monoid can be seen as accepting devices. A monoid M *recognises* with its subset $F \subseteq M$ under the homomorphism $\varphi : \Sigma^* \to M$ the language $\varphi^{-1}(F)$. It is well-known that finite monoids recognise precisely the regular languages. At the core of this lies the following straightforward correspondence.

Each transition system can be represented as a pair (φ, M) where $M = (\mathcal{P}(Q \times Q), \circ, \mathsf{id})$ is the monoid of binary relations over Q with composition as product and $\varphi : \Sigma^* \to M$ is the unique monoid homomorphism such that $\varphi(a) = \{(q, q') \mid \Delta(q, a, q')\}$ for every $a \in \Sigma$. From (φ, M) one can again obtain the presentation (Q, Σ, Δ). In case of deterministic transition systems each of the $\varphi(a)$ is functional. One can then work with the monoid $(Q \to Q, \circ, \mathsf{id})$ of partial unary functions over Q.

Elements of a semigroup capture global information about the words they represent under a given morphism, whereas only local behaviour can be immediately deduced from an automaton. For this reason, semigroups are better suited for certain tasks. Most notably, they allow an algebraic study and classification of certain sub-families of regular languages via corresponding varieties of semigroups.

2.5.1 Omega-semigroups

The well known fundamental correspondence between recognisability by finite automata and by finite semigroups has been extended to ω-regular sets. This is based on the notion of *ω-semigroups*. Rudimentary facts on ω-semigroups are well presented in [PP95]. We only mention what is most necessary.

An ω-semigroup $S = (S_f, S_\omega, \cdot, *, \pi)$ is a two-sorted algebra, where $(S_f, \cdot)$ is a semigroup, $* : S_f \times S_\omega \mapsto S_\omega$ is the *mixed product* satisfying for every $s, t \in S_f$ and every $\alpha \in S_\omega$ the equality

$$s * (t * \alpha) = (s \cdot t) * \alpha$$

and where $\pi : S_f^\omega \mapsto S_\omega$ is the *infinite product* satisfying

$$s_0 \cdot \pi(s_1, s_2, \ldots) = \pi(s_0, s_1, s_2, \ldots)$$

as well as the associativity rule

$$\pi(s_0, s_1, s_2, \ldots) = \pi(s_0 s_1 \cdots s_{k_1}, s_{k_1+1} s_{k_1+2} \cdots s_{k_2}, \ldots)$$

for every sequence $(s_i)_{i \geq 0}$ of elements of S_f and every strictly increasing sequence $(k_i)_{i \geq 0}$ of indices. Again, for $e \in S_f$ we denote $e^\omega = \pi(e, e, \ldots)$.

Morphisms of ω-semigroups are defined to preserve all three products as expected. There is a natural way to extend finite semigroups and their morphisms to ω-semigroups. As in semigroup theory, idempotents play a central role in this extension. An *idempotent* is a semigroup element $e \in S$ satisfying $ee = e$. Every finite semigroup S has an exponent $\pi \in \mathbb{N}$ such that for every $s \in S$ its power s^π is the sole idempotent of the subsemigroup generated by s. A pair of semigroup elements (s, e) is called a *linked pair* if e is idempotent and $se = s$. A way to think of a linked pair is as of an initial path labelled s leading into a loop labelled e in a finite graph, only "on a global scale", that is starting in any state.

There is also a natural extension of the free semigroup Σ^+ to the ω-semigroup $\Sigma^{\leq\omega} = (\Sigma^+, \Sigma^\omega)$ with $*$ and π defined by concatenation. An ω-semigroup $S = (S_f, S_\omega)$ *recognises* a language $L \subseteq \Sigma^\omega$ via a morphism $\phi : (\Sigma^+, \Sigma^\omega) \to (S_f, S_\omega)$ if $\phi^{-1}(\phi(L)) = L$. This notion of recognisability coincides, as for finite words, with that by non-deterministic Büchi automata. In [PP95] constructions from Büchi automata to ω-semigroups and back are also presented.

Theorem 2.5.1 (cf. [PP95])**.**
A language $L \subseteq \Sigma^\omega$ is ω-regular iff it is recognised by a finite ω-semigroup.

We note that this correspondence not only allows one to engage in an algebraic study of varieties of ω-regular languages, but also has the advantage of hiding complications of cutting apart and stitching together runs of Büchi automata as we shall do. This is precisely our reason for utilising this algebraic framework.

2.6 Logics

Structures

First-order structures $\mathfrak{A} = \big(A, \{R_i\}_i, \{f_j\}_j\big)$, simply structures from now on, are given by a set A, the universe or domain of $\mathfrak{A}$, by a (not necessarily finite) number of relations R_i and functions f_j over A of respective arities $0 < r_i \in \mathbb{N}$ and $n_j \in \mathbb{N}$. That is, $R_i \subseteq A^{r_i}$ and $f_j : A^{n_j} \to A$. Note that all functions f_j are required to be total, i.e. defined everywhere on A^{n_j}. Functions of arity zero are constants and can be noted separately as c_l.

A relational structure is one having only relations, but no functions or constants. We will mostly be working with relational structures, sometimes also allowing constants. Every structure can naturally be coded as a relational structure simply by replacing each function f_j by its graph G_{f_j} of arity $n_j + 1$ defined as $G_{f_j}(\vec{x}, y) \iff f_j(\vec{x}) = y$.

The signature of a structure $\mathfrak{A}$ as above is $\sigma(\mathfrak{A}) = \{R_i^{(r_i)} \mid i\} \cup \{f_j^{(n_j)} \mid j\}$ where the R_i and f_j are now simply treated as symbols, not relations, with their respective arities noted in superscript. We say that the relation R_i, also denoted $R_i^{\mathfrak{A}}$ for unambiguity, is the interpretation of the relation symbol $R_i^{(r_i)}$ in $\mathfrak{A}$. Similarly, functions are interpretations of function symbols. For convenience we will often not distinguish relation- and function symbols in notation from their interpretations as long as no confusion arises.

Word and tree structures

We have already encountered some structures in the introduction. A tree $t : [r]^{\leq\omega} \to \Sigma$ can be naturally seen as a structure with universe $\mathsf{dom}(t) \subseteq [r]^*$ and equipped with the prefix relation $\preceq$ and/or the successor functions succ_i for $i < r$ defined as $\mathrm{succ}_i(x) = xi$ as well as the labelling predicates $P_a = \{x \mid t(x) = a\}$ for each $a \in \Sigma$. In the special case of $r = 0$ the tree degenerates to a finite or infinite word, $\preceq$ reduces to the standard ordering $\leq$ and $\mathrm{succ} = \mathrm{succ}_0$ to the successor function on positions within the word.

We will often refer to the structure $\Delta_2 = ([2]^*, \mathrm{succ}_0, \mathrm{succ}_1)$ of the complete unlabelled binary tree as "the tree" and to $\Delta_1 = (\mathbb{N}, \mathrm{succ})$ as "the line". Sometimes, however, we will use their expansions with the prefix ordering.

Logics

Basic notions of first-order logic, FO, are standard. First-order formulas of signature σ are built from atomic formulas ($t_1 = t_2$ and $Rt_1 \ldots t_n$, where the t_i are σ-terms over functions and constants of σ and variables) using boolean connectives (e.g. $\wedge$,$\vee$,$\neg$, etc.) and quantification over first-order variables ($\exists x \ldots$ and $\forall x \ldots$).

The semantics of first-order formulas is given in terms of interpretations in structures of an appropriate signature. Given a structure $\mathfrak{A}$ of signature σ, every σ-term $t(\vec{x})$ and every $\mathsf{FO}(\sigma)$-formula $\varphi(\vec{x})$ with free variables $\vec{x} = x_1, \ldots, x_n$ determines an n-ary function $t^{\mathfrak{A}}$, respectively, an n-ary relation $\varphi^{\mathfrak{A}}$. In particular, when φ is a sentence, its interpretation $\varphi^{\mathfrak{A}}$ is the truth value of its satisfaction in $\mathfrak{A}$. When a sentence φ is satisfied in $\mathfrak{A}$ we also say that $\mathfrak{A}$ is a model of φ and use the shorthand $\mathfrak{A} \models \varphi$. Similarly, we write $\mathfrak{A} \models \varphi[\vec{a}]$ or $\vec{a} \in \varphi^{\mathfrak{A}}$ when the tuple $\vec{a}$ of elements of $\mathfrak{A}$ satisfies the formula $\varphi(\vec{x})$. Given a structure $\mathfrak{A}$, its first-order theory, denoted $\mathrm{Th}_{\mathsf{FO}}(\mathfrak{A})$, is the set of FO-sentences of signature $\sigma(\mathfrak{A})$ holding true in $\mathfrak{A}$.

First-order sentences of signature $\{<, (P_a)_{a\in\Sigma}\}$ can express properties of words, thereby defining languages. More precisely, a sentence φ defines the language L_φ of words w whose associated word structure $\mathfrak{W}_w = ([|w|], <, (P_a = w^{-1}(a))_{a\in\Sigma})$ is a model of φ. Similarly, one can define tree languages. To give an example, consider the following.

Example 2.6.1. Let $\Sigma = \{a, b, c\}$. The language $a^*b^*c^*$ is defined, say, by the following formula.

$$\varphi = \exists x \exists y (x \leq y \wedge (\forall z < x) P_a z \wedge (\forall x \leq z \leq y) P_b z \wedge (\forall z > y) P_c z) \quad .$$

Note that we have allowed ourselves a more liberal syntax, as customary, to abbreviate $x < y \vee x = y$ as $x \leq y$ and to write e.g. $(\forall z < x) P_a z$ instead of $\forall z (z < x \rightarrow P_a z)$. Throughout this thesis we will regularly take advantage of such practices.

It is well-known that, intuitively speaking, first-order logic cannot count. For instance, there is no first-order sentence holding true in precisely those finite structures of even cardinality, and no first-order sentence can define the language of words containing, say, an even number of a's. In fact, the first-order definable languages of finite words (with $<$) are precisely the star-free languages (McNaughton-Papert), equivalently, those recognised by an aperiodic monoid (Schützenberger). When, on the other hand, only the successor relation, but no ordering is available, the first-order definable languages of words are the so called locally threshold testable languages. For these results and refinements as well as extensions consult [Str94].

To remedy this deficiency of first-order logic one can enhance its expressiveness by the introduction of modulo-counting quantifiers $\exists^{(r,m)}$ with the intended meaning of formulas $\exists^{(r,m)} x\, \varphi$ being that the number n of distinct x satisfying φ is finite and n has remainder r modulo m. Additionally, the infinity quantifier $\exists^{\infty} x\, \varphi$ meaning that there are infinitely many x satisfying φ will also be of interest to us. Various other extensions of first-order logic will be considered in Section 7.1.

Monadic second-order logic, MSO, extends first-order logic with quantification over sets of elements. Set variables are conventionally written as capital letters $X, Y, Z, \ldots$ to distinguish them from first-order variables. In addition to the atomic formulas of first-order logic, Xx is an atomic formula of MSO for every set variable X and first-order variable x. The monadic theory, $\mathrm{Th}_{\mathsf{MSO}}(\mathfrak{A})$ of a structure $\mathfrak{A}$ is the set of MSO-sentences of which $\mathfrak{A}$ is a model.

Quantification over sets results in considerable increase in expressive power. MSO is able to express reachability in graphs, transitive closure and fixpoint constructions of definable binary relations, as well as some NP-complete problems (e.g. 3-colorability). For example, the following formula defines the prefix relation in $\{0, 1\}$-branching trees.

$$\varphi_{\preceq}(x, y) = \forall X (Xx \wedge \forall z, z' (Xz \wedge (\mathrm{succ}_0 zz' \vee \mathrm{succ}_1 zz') \rightarrow Xz') \rightarrow Xy)$$

There is therefore no loss in expressive power, as far as MSO is considered, over word- and tree structures if we omit the (prefix) ordering from the signature. The fundamental results of Büchi and Elgot (for finite and infinite words), Thatcher and Wright, and Doner (for finite trees) and of Rabin (for infinite trees) establish the close correspondence of finite automata and monadic second-order logic.

Theorem 2.6.2 (Büchi, Elgot, Thatcher-Wright, Doner, Rabin – cf. [Tho97])**.** *A language of finite or infinite words or trees is finite automaton recognisable iff it is monadic second-order definable.*

In all cases, the direction from automata to logic is rather straightforward. It consists of existentially guessing a run of the automaton, coded by a tuple of sets, and checking its compliance with the transition relation as well as it satisfying the acceptance condition. Thus, as a byproduct, together with the converse direction, we obtain that every MSO formula is equivalent to one in "automaton normal form", in particular, to a Σ^1_1-formula on words (McNaughton provided a tighter normal-form) and to a Σ^1_2 (also to a Π^1_2) formula on trees [TL94]. The other direction, establishing translation from logic to automata proceeds via induction on formulas, taking advantage of closure properties of the automata involved, the most critical of which is complementation. Indeed, existentially quantified formulas naturally translate to non-deterministic automata, the complementation of which is far from trivial as soon as the automaton model does not allow determinisation in general. That is the case for Büchi automata on infinite words, for top-down automata on finite trees, and for Muller, Rabin, or parity automata on infinite trees as we have already remarked. The actual aim and the main achievement of the automaton method pioneered by Büchi, Elgot, McNaughton, Rabin, and others is the establishment, based on decidability of the emptiness problem of the automata involved, of the decidability of the monadic theories of one successor, S1S, respectively of two successors, S2S. In our terminology, we have the following.

Theorem 2.6.3 (Büchi, Rabin, cf. [Tho97],[GTW02])**.** *The monadic second-order theories of Δ_1 and of Δ_2 are decidable.*

2.6.1 Interpretations

Logical interpretations provide a means of transforming a structure, the host, into another one, the interpreted structure, in a way that an associated transformation allows us to reduce, in an effective way, the logical theory of the interpreted structure onto the theory of the host. Decidability of the theory of the host thus yields a decision procedure for that of the interpreted structure. Let us first formally define first-order interpretations.

Let σ and τ be relational signatures. A *first-order interpretation* $\mathcal{I}$ transforming σ-structures into τ-structures is a collection of $\mathsf{FO}(\sigma)$-formulas

$$\mathcal{I} = (\delta(\vec{x}), \{\psi_R(\vec{x}_1, \dots, \vec{x}_r) \mid R^{(r)} \in \tau\})$$

where each vector $\vec{x}$, respectively, $\vec{x}_i$ of free variables is of the same length n, which we call the *dimension* of the interpretation.

An interpretation $\mathcal{I}$ as above transforms a σ-structure $\mathfrak{A}$ into the τ-structure $\mathfrak{B} = \mathcal{I}(\mathfrak{A})$ defined as follows. The universe of $\mathfrak{B}$ is the set $\delta^{\mathfrak{A}}$ of n-tuples of elements of $\mathfrak{A}$ satisfying δ. Similarly, for each relation symbol $R^{(r)} \in \tau$, $R^{\mathfrak{B}}$ is the set of r-tuples of n-tuples $\vec{a}_1, \dots, \vec{a}_r$ from $\mathfrak{A}$ such that $\mathfrak{A} \models \varphi_R(\vec{a}_1, \dots, \vec{a}_r)$.

If $\mathcal{I}$ is a one-dimensional interpretation, we also say that $\mathfrak{B}$ is definable in $\mathfrak{A}$. Clearly, $\mathfrak{B}$ is interpretable in $\mathfrak{A}$ iff it is definable in the n-fold direct product $\mathfrak{A}^n$ for some n.

To an interpretation $\mathcal{I}$ as above is naturally associated a transformation $\cdot^{\mathcal{I}}$ of τ-formulas into σ-formulas defined inductively by stipulating

$$\begin{array}{lcl} (R(x_1,\ldots,x_r))^{\mathcal{I}} & = & \varphi_R(x_{1,1},\ldots,x_{1,n},\cdots,x_{r,1},\ldots,x_{r,n}) \\ (\exists x\psi)^{\mathcal{I}} & = & \exists\vec{x}(\delta(\vec{x})\wedge\psi^{\mathcal{I}}) \\ (\forall x\psi)^{\mathcal{I}} & = & \forall\vec{x}(\delta(\vec{x})\rightarrow\psi^{\mathcal{I}}) \end{array}$$

and that $\cdot^{\mathcal{I}}$ distributes over boolean combinations. Intuitively speaking, $\psi^{\mathcal{I}}$ is obtained from ψ by substituting each atomic formula by its definition in $\mathcal{I}$ and by restricting quantification to tuples satisfying δ. This transformation provides an effective uniform reduction of $\mathrm{Th}_{\mathsf{FO}}(\mathcal{I}(\mathfrak{A}))$ to $\mathrm{Th}_{\mathsf{FO}}(\mathfrak{A})$ for all prospective $\mathfrak{A}$'s.

Regarding the complexity of the reduction we note that the transformation $\psi\mapsto\psi^{\mathcal{I}}$ is logspace-lineartime computable (even DGSM computable if τ does not contain function symbols) and only increases the length of formulas by at most a linear factor.

Consequently, if $\mathrm{Th}_{\mathsf{FO}}(\mathfrak{A})$ is in PSPACE (the first-order theory of every structure with two elements is at least PSPACE-hard) or is *elementary recursive* (i.e. in time/space bounded by some tower of exponentials), then the first-order theory of every $\mathfrak{B}\leq_{\mathsf{FO}}\mathfrak{A}$ is in PSPACE, respectively, is elementary recursive (and is in time/space bounded by the same tower of exponentials).

For more intricate theories (like most of the theories we will encounter), the complexity is usually measured in terms of the number of quantifier alternations in the prenex normal form of formulas, which can also be linearly increased by the reduction associated to an interpretation. However, in case of interpretations involving only quantifier-free formulas (more generally, in case of so called simple interpretations [Grä90]), the quantifier prefix of formulas is preserved by the transformation, hence this complexity blow-up is avoided.

Monadic second-order interpretations $\mathcal{I}$ and their associated contravariant transformations of formulas $\cdot^{\mathcal{I}}$ are defined analogously to the above, mutatis mutandis. Note, however, that in order for $\cdot^{\mathcal{I}}$ to transform MSO-formulas into MSO-formulas, i.e. avoiding quantification over sets of tuples, $\mathcal{I}$ must be one-dimensional. This restriction is necessary, as prompted by the example of the infinite grid $(\mathbb{N}\times\mathbb{N},(\mathrm{succ},\_),(\_,\mathrm{succ}))$ which can be interpreted in $(\mathbb{N},\mathrm{succ})$ via a trivial two-dimensional first-order interpretation, and is a prominent example of a structure with an undecidable MSO-theory [See91].

As an example let us show that the monadic theory SωS of the complete ω-branching tree of infinite depth is decidable. Our aim is to give an MSO-interpretation of the tree $\mathfrak{T}_\omega=(\mathbb{N}^*,\{\mathrm{succ}_n\mid n\in\mathbb{N}\})$ in $\Delta_2=([2]^*,\mathrm{succ}_0,\mathrm{succ}_1)$. The idea is to code a finite initial branch segment $n_1,n_2,\ldots,n_l$ of $\mathfrak{T}_\omega$ by the branch $0^{n_1}10^{n_2}1\ldots0^{n_l}1$ in Δ_2. It is then straightforward to write, for each n, a formula $\varphi_{\mathrm{succ}_n}(x,y)$ defining succ_n. Alternatively, one can consider $\mathfrak{T}_\omega$ to be of the finite signature $(\mathbb{N}^*,\preceq,<_{\mathrm{lex}})$ equipped with the prefix and the lexicographic orders. To give an interpretation of this structure, in which each succ_n is definable, using the same coding as above, prefix is simply interpreted by prefix itself (restricted to nodes with an incoming 1-edge) and the lexicographic order on $\mathbb{N}^*$ is defined by the lexicographic order on $\{0,1\}^* \cdot 1$ stressing that $1 < 0$.

MSO-transductions [Cou94] were introduced as a generalisation of monadic interpretations. We have noted that every first-order interpretation can equivalently be thought of as a one-dimensional interpretation (i.e. definition) in a k-fold direct product of the host structure with itself, and that taking direct products is not an MSO-compatible operation, e.g. in that it easily produces structures with an undecidable MSO-theory. MSO-interpretations have the drawback that they do not allow one to interpret a larger finite structure in a smaller one. This can be remedied by an initial *k-copying* operation transforming a structure $\mathfrak{A}$ into the union $\bigcup_{0 \leq i < k} \mathfrak{A} \times \{i\}$ of k disjoint copies of $\mathfrak{A}$ endowed with auxiliary edges $E_{i,j} = \{((a,i),(a,j)) \mid a \in \mathfrak{A}\}$ pointwise connecting the copies in parallel. Using Ehrenfeucht-Fraïssé games it is easy to check that k-copying is MSO-compatible in the sense that it induces a similar k-copying operation on (bounded) MSO-theories as well. Furthermore, k-copying preserves decidability of MSO-theories. An MSO-transduction is composed of a k-copying for some $k > 1$ followed by an MSO-interpretation.

3 Finite Presentations of Structures

The central topic of this thesis being automatic presentations, we shall begin by introducing this notion mentioning some restrictions followed by some of its closer and more distant relatives and generalisations.

Unless otherwise stated, we will always think of structures as being relational, that is, with n-place functions given by their graphs as relations of arity $(n+1)$. This makes perfect sense as far as presentations are concerned and it comes at no sacrifice for our purposes. Nonetheless, for the sake of readability, we will occasionally, when appropriate, allow ourselves to use a functional signature.

3.1 Automatic Presentations

It is time to formally define what we mean by an automatic presentation of a structure. The following definition covers both finite and infinite word-automatic presentations as well as finite and infinite tree-automatic presentations, the sole difference lying with the kind of finite automata being used.

Definition 3.1.1 (Automatic structures)**.**
Consider a relational structure $\mathfrak{A} = (A, \{R_i\})$ comprising relations R_i over the universe $\mathsf{dom}(\mathfrak{A}) = A$. An *(omega-)(word-/tree-)automatic presentation of* $\mathfrak{A}$ is a tuple $\mathfrak{d} = (\mathcal{A}, \mathcal{A}_{\approx}, \{\mathcal{A}_i\})$ of finite synchronised (omega-)(word-/tree-)automata, such that

- $\mathcal{A}$ recognises an (omega-)(word-/tree-)regular set D called the *domain* of the presentation,
- each $\mathcal{A}_i$ recognises an (omega-)(word-/tree-)regular relation S_i of the same arity as R_i, and
- there exists a surjective function $f : D \to \mathsf{dom}(\mathfrak{A})$ referred to as the *naming function* or *co-ordinate mapping* of the presentation, such that
- $\approx\, = \{(u, w) \in D^2 \mid f(u) = f(w)\}$ (the kernel of f) is a congruence relation on $(D, \{S_i\})$ and is recognised by $\mathcal{A}_{\approx}$, and
- $f/_{\approx}$ is an isomorphism from $(D, \{S_i\})/_{\approx}$ to $\mathcal{A}$.

We say that the presentation is *injective* whenever f is, in which case $\mathcal{A}_{\approx}$ can be omitted.

A structure $\mathfrak{A}$ is *(omega-)(word-/tree-)automatic* if it allows an (omega-)(word-/tree-)automatic presentation. The classes of (omega-)(word-/tree-)automatic structures are denoted by AutStr, ωAutStr, TAutStr and ωTAutStr, respectively.

This is more or less the standard definition used throughout the literature. A tuple $\mathfrak{d}$ of finite automata as specified above does indeed provide a finite description of the represented structure *up to isomorphism*. This is of course the most one can expect, and just means, perfectly reasonably, that the classes $(\omega)(\mathrm{T})\textsc{AutStr}$ are closed under isomorphisms. Although we do not distinguish between isomorphic structures, we very much intend to distinguish between "essentially different" automatic presentations of individual structures. In doing so, we are not so much interested in the actual automata of a presentation, i.e in $\mathfrak{d}$, but rather in which relations are regularly represented under a certain naming function. This will be formalised and elaborated in Section 4.2. It is worth noting that given a structure $\mathfrak{A}$ and an appropriate naming function f – one under which the inverse images of the universe and the relations of $\mathfrak{A}$ are regular – an automatic presentation $\mathfrak{d}$ of $\mathfrak{A}$ can be defined in a canonical way, e.g. as the tuple of automata determined by the syntactic congruence classes of the regular inverse images of the respective relations of $\mathfrak{A}$.

As a first observation on the class AutStr we note that all finite structures are trivially (though not necessarily efficiently) automatically presentable. Indeed, as all finite relations on words are regular, a naming function can be chosen arbitrarily. Our intention is, of course, to represent infinite structures. This raises the question whether finiteness of (omega-)(tree-)automatic structures is decidable. We will address this question in Section 3.1.2. Aside of that we shall not be concerned with presentations of finite structures.

From Definition 3.1.1 it is immediate that AutStr is included in TAutStr, that they contain only finite and countable structures and are thus strictly contained in ωAutStr, respectively in ωTAutStr, the latter also subsuming the former. In fact, AutStr is a proper subclass of TAutStr as shown by Blumensath [Blu99, BG04] (see Example 3.1.2(v) and Section 4.1). In Section 3.1.1 we will prove the non-trivial fact that AutStr is the restriction of ωAutStr to finite and countable structures.

We will, for the most part, be concerned with automatic presentations on finite words. Therefore, when the type of underlying objects is left unspecified, an automatic presentation will from now on mean one over finite words. When confusion could arise the kind of presentation meant will always be specified. The term automatic presentation will often be abbreviated as aut. pres. or simply as a. p.

In general, neither the naming function nor the tuple of automata comprising a particular presentation determines the other. Nonetheless, depending on the context of our investigation we might just be interested in either one of these constituents while tacitly omitting the other. Frequently, we shall allow ourselves to refer to the regular relations comprising a presentation in place of actual automata recognising them or to use other means, such as regular expressions, of describing the relations. Let us illustrate this practice on the examples below.

Example 3.1.2.

(i) The ordinal ω is automatic. Indeed, the simplest presentation thinkable is the unary one: $(0^*, \{(0^k, 0^l) \mid k < l\})$.

(ii) In fact, every ordinal below ω^ω is automatic. A presentation of ω^k generalising the above one, is $((0^*1)^k, <_{\text{lex}})$ where $<_{\text{lex}}$ denotes the lexicographic order (now on the binary alphabet) which is clearly regular. In this presentation the coordinate map is

$$0^{n_{k-1}}1\ldots 0^{n_0}1 \mapsto n_{k-1}\omega^{k-1} + \ldots + n_1\omega^1 + n_0 \quad .$$

(iii) The ordering of the rationals $(\mathbb{Q}, <)$ is automatic. The lexicographic ordering[1] on binary words ending with a 1 is of order type $(\mathbb{Q}, <)$ as can easily be checked. Thus, $(\{0,1\}^*1, <_{\text{lex}})$ constitutes an a. p. of $(\mathbb{Q}, <)$ as claimed.

(iv) Presburger arithmetic $\mathcal{N} = (\mathbb{N}, +)$ is automatic. Indeed, for every natural $k > 1$, the base k least significant bit first presentation of naturals (with or without leading zeros) constitutes a naming function of an a. p. A finite automaton can easily perform the schoolbook addition method while keeping track of the carry in its state. Such a presentation is injective when leading zeros are suppressed.

(v) Skolem arithmetic, $(\mathbb{N}, \cdot)$, the structure of the naturals with multiplication is tree automatic. The presentation is based on the unique factorisation of naturals into prime powers. Each number n is represented by a tree coding the finite sequence of powers $2^{n_2}3^{n_3}\cdots p^{n_p}\cdots$ in the factorisation of n representing each n_p by a single branch, i.e. as a word, say in binary notation. Multiplication is thus reduced to the addition of corresponding exponents, carried out in parallel on corresponding branches as in the word-automatic presentation of Presburger arithmetic.
This construction can naturally be generalised to give tree-automatic presentations of weak direct products of word-automatic structures [Blu99, BG04].

(vi) The infinite grid $(\mathbb{N}\times\mathbb{N}, \text{right}, \text{up})$ with the functions right : $(n, m) \mapsto (n+1, m)$ and similarly up : $(n, m) \mapsto (n, m+1)$ can be automatically presented on the domain a^*b^* with relations

$$R = \binom{a}{a}^* \binom{b}{a} \binom{b}{b}^* \binom{\Box}{b}$$

and U defined by a similar regular expression.

(vii) The complete infinite binary tree $(\{0,1\}^*, \text{succ}_0, \text{succ}_1, \preceq)$ equipped with the left and right successor relations as well as the prefix relation $\preceq$ is trivially automatic with the identity naming function. By adding to this tree structure the equal-length relation we obtain a *maximal* automatic expansion (cf. Theorems 3.2.4, 4.3.1 and Corollary 7.0.5).

[1] Given an ordering on the symbols of the alphabet a word u is lexicographically smaller than w if either u is a proper prefix of w or if in the first position where u and w differ one finds a smaller symbol in u than in w.

(viii) The transition graphs of pushdown automata are automatic. Given a pushdown automaton $\mathcal{A}$ with states Q, stack alphabet Γ, input alphabet Σ and transition relation Δ we can construct an automatic presentation of the transition graph of its configurations [2] as follows. We take $Q\Gamma^*$ to be the domain of the presentation in which $q\gamma$ represents the configuration of state q and stack $\gamma \in \Gamma^*$. For each $a \in \Sigma$ there is an a-transition from $q\gamma$ to $q'\gamma'$ iff $\gamma = z\alpha$, $\gamma' = w\alpha$ and $(q, z, q', w) \in \Delta$ for some $z \in \Gamma$ and $w \in \Gamma^*$. Δ being finite, this relation is obviously regular for each a. Notice that in these presentations the transition relations are not only regular but in fact defined by prefix-rewriting rules (cf. Sections 3.3.3 and 3.4 below).

(ix) The transition graphs of Turing machines are automatic [KN95]. We can give an automatic presentation of each TM $\mathcal{M}$ similarly to those of pushdown automata. Configurations are encoded as strings $\alpha q\beta \in \Gamma^* Q\Gamma^*$ where α and β are the tape contents to the left, respectively, to the right of the head of $\mathcal{M}$, and q is the current state. Observe that, as opposed to presentations of pushdown graphs, the state is now positioned not at the left of the string but at the location of the head. Consequently, rewriting is not confined to prefixes, but rather occurs around the state symbol: transitions are of the form $\alpha aqb\beta \mapsto \alpha uq'w\beta$ for adequate a, b, u, w and q, q' as determined by the transition function of $\mathcal{M}$. The fact that TM graphs are presentable using infix-rewriting has the profound consequence that reachability questions in infix-rewriting systems are generally undecidable, as opposed to graphs of prefix-rewriting systems, whose monadic second-order theory is decidable (cf. Section 3.3.3).

The advantage of having an automatic presentation of either kind of a structure lies in the fact that first-order formulas can be effectively evaluated over it using classical automata constructions. Closure properties of the underlying class of automata allow one to translate formulas into equivalent automata. Together with decidability of emptiness for the automaton model this yields a decision procedure for the first-order theory of the structure. This “automaton method” towards logic is the very essence of the tight correspondence between automata and logics as pioneered by Büchi, Elgot and Trahtenbrot in the early 1960’s. Hodgson calls such theories “automaton decidable” [Hod83]. These well-known facts are gathered in the following fundamental theorem (cf. [Hod83],[KN95],[Blu99, BG04],[Rub04]).

Theorem 3.1.3 (Fundamental Theorem of Automatic Presentations)**.**

(i) Let $\mathfrak{A}$ be (ω-)(tree-)automatic with presentation $\mathfrak{d}$ and naming function f. Then one can effectively construct, for each FO*-formula $\varphi(\vec{a}, \vec{x})$ with parameters $\vec{a}$ from $\mathfrak{A}$ defining a k-ary relation R over $\mathfrak{A}$, a synchronous (ω-)(tree-)automaton recognising $f^{-1}(R)$.*

[2] For visibly pushdown automata the same representation of configurations also allows for the trace equivalence relation to be recognised by a finite automaton. In [BLS06] this presentation was utilised to obtain a decidability result.

(ii) The FO*-theory of every (*ω*-)(tree-)automatic structure is decidable.*

(iii) The classes (ω)(T)AutStr *are closed under* FO*-interpretations.*

Let us point out now that by (*i*) above every set and relation first-order definable from (ω-)(tree-)regular sets and relations is itself (ω-)(tree-)regular. We shall use this fact frequently without explicit reference.

The Fundamental Theorem provides a very efficient tool for constructing automatic presentations of structures by defining them in other ones for which a presentation is at hand. Let us illustrate this on the following example.

Example 3.1.4 (Finitely generated abelian groups are automatic [KN95])**.** Every finitely generated abelian group G is a product of cyclic groups, that is a direct product of $(\mathbb{Z},+)^r$ with a finite commutative group G_0 (which is a product of cyclic groups of finite order). In Example 3.1.2 (iv) we gave several automatic presentations of $\mathcal{N} = (\mathbb{N},+)$. A straightforward interpretation of $(\mathbb{Z},+)$, and hence of $(\mathbb{Z},+)^r$, in $\mathcal{N}$ shows, by the Fundamental Theorem, that $(\mathbb{Z},+)^r$ is automatic for each $r > 0$. We have already noted that all finite structures are automatic. In fact, they can also be trivially interpreted in $\mathcal{N}$. Finally observe that the ($2r$-dimensional) interpretation of $(\mathbb{Z},+)^r$ and the (one-dimensional) interpretation of G_0 in $\mathcal{N}$ can be effortlessly combined into a ($(2r+1)$-dimensional) interpretation of their direct product in $\mathcal{N}$, providing, via the Fundamental Theorem, an automatic presentation of G. We add that finitely generated abelian groups are also automatic in the more restrictive sense of [CEH$^+$92].

Note that taking direct products of automatic structures always yields an automatic structure [Hod82]. This can either be verified by a direct construction or be inferred from the existence of complete structures (cf. Theorem 3.2.4) [Blu99, BG04]. The idea of the direct construction of Hodgson [Hod82] is to encode pairs of elements by the convolution of their representations and simulate corresponding automata of both presentations synchronously on their respective components of convoluted pairs. Observe that this is precisely what happens when we combine interpretations as in Example 3.1.4 above.

It is also an immediate consequence of the Fundamental Theorem that each of the classes (ω)(T)AutStr is closed under factorisation by FO-definable congruences.

3.1.1 Injective presentations

A very natural question to ask is whether every (ω-)(tree-)automatic structure can actually be automatically presented with unique representants. In other words, whether injectivity of (ω-)(tree-)automatic presentations can always be assumed. This issue has a relevance for instance for the decidability of extensions of first-order logic, e.g. with infinity (FO^∞) and/or modulo counting quantifiers ($\mathsf{FO}^{\infty,\mathrm{mod}}$). Indeed, the Fundamental Theorem has been extended to FO^∞ in [Blu99, BG04] over (ω)AutStr and further to $\mathsf{FO}^{\infty,\mathrm{mod}}$ over AutStr in [KRS03b, KRS04] and over injectively presentable ω-automatic structures in [KL06], finally, to $\mathsf{FO}^{\infty,\mathrm{mod}}$ over (injective) tree-automatic presentations in [Col04a].

Theorem 3.1.5 ([Blu99, BG04],[KRS04],[KL06],[Col04a])**.** *The statements of the Fundamental Theorem (definability, decidability and closure under interpretation) hold true for* $\mathsf{FO}^{\infty,\mathrm{mod}}$ *over all* injectively *presentable (ω-)(tree-)automatic structures.*

It is a simple observation that words can be well ordered using a regular ordering, e.g. the length-lexicographic one, implying, by the Fundamental Theorem, that finite word automatic presentations can be made injective by restricting their universe to a regular set of unique representants. However, this approach extends neither to trees nor to ω-words, as these cannot be well ordered in a regular fashion. Therefore, it is natural to ask in which cases is injective presentability an actual restriction.

Tree-automatic presentations

Since there is no regular well-ordering of finite, let alone infinite, labelled trees [CL07], the approach of selecting a regular set of unique representants of each equivalence class of an arbitrary tree-automatic presentations seems problematic.

Nonetheless, in [CL06, Theorem 8] it is shown that every tree-automatic structure admits an injective tree-automatic presentation. Instead of trying to find a tree-regular cross-section of a tree-regular equivalence, the construction of [CL06] associates, in a wMSO-definable, hence tree-regular manner, a finite number of trees to every equivalence class, the lexicographically least of which can then be taken to represent the class. Thus, injectivity poses no restriction on TAutStr either.

ω-Automatic presentations of countable structures

The case of ω-automatic presentations is more obscure. An example witnessing that not all ω-regular equivalence relations have an ω-regular set of unique representants is the *equal-ends* relation of Kuske and Lohrey [KL06]. Two ω-words are of equal end if they agree on all but finitely many positions. This equivalence relation is accepted by the following non-deterministic Büchi automaton.

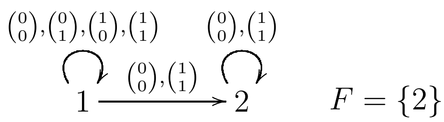


Figure 3.1: An automaton for the equal-ends relation.

Note that equal-ends has uncountably many equivalence classes each of countable cardinality. While it is not yet known whether every ω-automatic structure has an injective ω-automatic presentation, and although the example of the equal-ends equivalence shows that not all ω-regular equivalence relations have an ω-regular set of unique representants, we are able to show that for every ω-regular equivalence relation having only *countably many classes* an ω-regular set of unique representants can effectively be found. This is joint work with Łukasz Kaiser and Sasha Rubin [BKR07].

A question raised in [Blu99] is whether every *countable* ω-automatic structure is also automatic. It is easy to see that every injective ω-automatic presentation of a countable structure can be "packed" into an automatic presentation. For the reader's convenience we sketch a proof of this fact.

Proposition 3.1.6. ([Blu99, Theorem 5.32]) Let $\mathfrak{d}$ be an injective ω-automatic presentation of a countable structure $\mathcal{A}$. Then, an (injective) automatic presentation $\mathfrak{d}'$ of $\mathcal{A}$ can effectively be constructed.

Proof. By injectivity of the presentation, its domain D is a countable ω-regular set and therefore of the form $\bigcup_{k=1}^{n} U_k w_k^\omega$ for some finite periods w_k and regular sets $U_k \subseteq \Sigma^*$. Wlog. $\{1,\dots,n\} \cap \Sigma = \emptyset$. To obtain a presentation using finite words simply take as domain $D' = \bigcup_k k \cdot U_k$. To represent the relations, construct for each $\mathcal{A}_i$ an automaton $\mathcal{A}_i'$ simulating it as follows. On reading, as the first symbols of its input, a tuple $(k_1,\dots,k_{n_i})$ of indices, each $\mathcal{A}_i'$ enters a state $(q_0, w_{k_1},\dots,w_{k_{n_i}})$ where q_0 is the initial state of $\mathcal{A}_i$ and the w_{k_j}'s are the periods corresponding to the indices k_j. $\mathcal{A}_i'$ simulates $\mathcal{A}_i$ and, on reaching the end of certain input words, $\mathcal{A}_i'$ proceeds by treating the corresponding period w_{k_j} stored in its state as part of the input, and rotating it in each step by one letter. Upon termination, the state of $\mathcal{A}_i'$, some $(q, \widetilde{w_{k_1}},\dots,\widetilde{w_{k_{n_i}}})$, is accepting precisely if $\mathcal{A}_i$ accepts the tuple $(\widetilde{w_{k_1}}^\omega,\dots,\widetilde{w_{k_{n_i}}}^\omega)$ from state q. $\square$

Note that in the proof above we have only relied on the countability of the domain of the ω-automatic presentation, not on it actually being injective. The crux of our construction answering the question of Blumensath is encompassed in the following lemma.

Lemma 3.1.7. Let E be an ω-automatic equivalence relation over $(\Sigma^2)^\omega$ and let $S = (S_f, S_\omega)$ be the finite ω-semigroup that recognises E via ϕ. If E has countably many equivalence classes, then for every $w, u, v \in \Sigma^*$ there is a natural k such that

$$\phi\begin{pmatrix}u\\u\end{pmatrix} = \phi\begin{pmatrix}v\\v\end{pmatrix} \implies (wv^k u^\omega, wv^\omega) \in E.$$

Let us postpone the proof of this lemma for now and derive first our injectivity results. To this end we will need the following simple facts guaranteeing the existence of short witnesses and idempotents.

Proposition 3.1.8. For a regular language L recognised by a Büchi automaton and the corresponding semigroup morphism $\phi : (\Sigma^+, \Sigma^\omega) \to (S_f, S_\omega)$ there exist

(i) a number M so that for every $u \in \Sigma^+$ there is a word v with $|v| \leq M$ so that $\phi(u) = \phi(v)$,

(ii) a number K, the *exponent of* S_f, so that for every word u the element $\phi(u^K)$ is an idempotent.

Theorem 3.1.9 ([BKR07]). *For every ω-automatic presentation $(\mathcal{A}, \mathcal{A}_\approx, \{\mathcal{A}_i\}_i)$ of a* countable *structure there exists an ω-regular set D of unique representants of each $\approx$-class, thus yielding an* injective *ω-automatic presentation $(\mathcal{D}, \{\mathcal{A}_i\}_i)$.*

Proof. Let us consider the ω-regular equivalence relation $\approx$, the corresponding semigroup (S_f, S_ω) and morphism ϕ recognising $\approx$, and let M be the constant from Proposition 3.1.8(i) adequate for the morphism $(x \mapsto \phi\binom{x}{x})$. Let B be the set of ω-words of the form su^ω for some s and u with $|u| \leq M$. In other words, B is the finite union $\bigcup_{|u|\leq M} \Sigma^* u^\omega$ and is thus regular.

Let us now show that every equivalence class of $\approx$ has a representant in B. Assume to the contrary that there is a word x that is not equivalent to any element of B. Such words can be recognised by an automaton since they are defined by the formula $\forall y\, (By \rightarrow \neg x \approx y)$ where both B and $\approx$ are regular relations. Therefore, there is an ultimately periodic word with this property, let us denote it by wv^ω. By the choice of M there is a word u with $|u| \leq M$ such that $\phi\binom{u}{u} = \phi\binom{v}{v}$. By definition of B we have that $wv^k u^\omega \in B$ for every k and therefore wv^ω and $wv^k u^\omega$ are not equivalent, which contradicts Lemma 3.1.7 since, by assumption, $\approx$ has only countably many equivalence classes.

It remains to prune B to select unique representants of each $\approx$-class. Given the structure $\bigcup_{|u|\leq M} \Sigma^* u^\omega$ of B it is easy to give an ω-regular well-founded linear order on its elements. Define $xu^\omega < yv^\omega$ iff u is lexicographically smaller than v or $|u| = |v|$ and x is length-lexicographically smaller than y. Note that this definition involves only finitely many case distinctions, ensuring ω-regularity. Finally, define D as the set of minimal elements of every class. □

From the above theorem using Proposition 3.1.6 we immediately obtain as a corollary the following result.

Corollary 3.1.10. *A countable structure is ω-automatic iff it is automatic. Transforming a presentation of one type into the other can be done effectively.*

Further note that the proof of Theorem 3.1.9 yields a decidable criterion for countability of an ω-automatic structure. Indeed, given an ω-automatic presentation one simply has to construct the set B as above and check whether every equivalence class is represented by an element of B.

Corollary 3.1.11. *It is decidable whether a given ω-automatic presentation represents a countable structure.*

To prove Lemma 3.1.7 we need to find elements of the ω-semigroup corresponding to the equivalence relations that satisfy certain algebraic properties. In order to enhance readability, we will use the shorthand $\binom{x}{y}$ in place of $\phi\binom{x}{y}$ throughout the rest of this section.

Lemma 3.1.12. Let T be an ω-automatic relation over $(\Sigma^2)^\omega$ and let $S = (S_f, S_\omega)$ be the finite ω-semigroup that recognises T via ϕ. If there exist words $u, v \in \Sigma^*$ for which

$$\begin{pmatrix} u \\ u \end{pmatrix} = \begin{pmatrix} v \\ v \end{pmatrix} \text{ and } u^\omega \neq v^\omega$$

then for any $w \in \Sigma^*$ we can find words $w', a, b \in \Sigma^*$, $a \neq b$, satisfying the following properties:

(i) $\binom{a}{a}$ and $\binom{b}{b}$ are idempotent,

(ii) $\binom{a}{a} = \binom{b}{b}$,

(iii) $|a| = |b|$,

(iv) $\binom{a}{b}\binom{a}{a} = \binom{a}{b}\binom{b}{b} = \binom{a}{b}$,

(v) $\binom{b}{a}\binom{a}{a} = \binom{b}{a}\binom{b}{b} = \binom{b}{a}$,

(vi) $\binom{w'}{w'}\binom{a}{a} = \binom{w'}{w'}\binom{b}{b} = \binom{w'}{w'}$.

Moreover, if T is transitive and $(wv^k u^\omega, wv^\omega) \notin T$ for every natural k, then

(vii) $(w'a^\omega, w'b^\omega) \notin T$.

Proof.
Let K be the exponent of S_f, i.e. the least positive natural such that s^K is idempotent for every $s \in S_f$. Set $\hat{u} = (u^K)^{|v|}$ and $\hat{v} = (v^K)^{|u|}$.

The assumption $\binom{u}{u} = \binom{v}{v}$ implies $\binom{\hat{u}}{\hat{u}} = \binom{\hat{v}}{\hat{v}}$, by definition we have $|\hat{u}| = |\hat{v}|$, and by the choice of K, $\binom{\hat{u}}{\hat{u}} = \binom{u^{|v|}}{u^{|v|}}^K$ is idempotent. This shows that $\hat{u}$ and $\hat{v}$ satisfy properties (i)-(iii).

Let us now put $a = \hat{u}\hat{u}$, $b = \hat{v}\hat{u}$ and $w' = w\hat{v}$. Then property (i) follows directly from idempotency of $\binom{\hat{u}}{\hat{u}}$ and $\binom{\hat{v}}{\hat{v}}$, property (ii) is checked by the identities

$$\begin{pmatrix} a \\ a \end{pmatrix} = \begin{pmatrix} \hat{u} \\ \hat{u} \end{pmatrix}\begin{pmatrix} \hat{u} \\ \hat{u} \end{pmatrix} = \begin{pmatrix} \hat{v} \\ \hat{v} \end{pmatrix}\begin{pmatrix} \hat{u} \\ \hat{u} \end{pmatrix} = \begin{pmatrix} b \\ b \end{pmatrix}.$$

Properties (iv) and (v) can be proved using idempotency in a similar way, e.g.

$$\begin{pmatrix} a \\ b \end{pmatrix}\begin{pmatrix} a \\ a \end{pmatrix} = \begin{pmatrix} \hat{u}\hat{u} \\ \hat{v}\hat{u} \end{pmatrix}\begin{pmatrix} \hat{u}\hat{u} \\ \hat{u}\hat{u} \end{pmatrix} = \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix}\begin{pmatrix} \hat{u} \\ \hat{u} \end{pmatrix}^3 = \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix}\begin{pmatrix} \hat{u} \\ \hat{u} \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}.$$

Since (iii) is obviously satisfied and (vi) follows from the idempotency of $\binom{\hat{v}}{\hat{v}} = \binom{\hat{u}}{\hat{u}}$, we only need to prove (vii), i.e. that $(w'a^\omega, w'b^\omega) \notin T$ provided that $(wv^k u^\omega, wv^\omega) \notin T$ for every k and that T is transitive.

Let us assume to the contrary that $(w'a^\omega, w'b^\omega) = (w'(\hat{u}\hat{u})^\omega, w'(\hat{v}\hat{u})^\omega) \in T$. Using the fact that ϕ composes with the infinite product π of the ω-semigroup and taking advantage of properties (i)-(iii) for $\hat{u}$ and $\hat{v}$, we calculate that

$$\begin{aligned}
\binom{w'(\hat{u}\hat{u})^\omega}{w'(\hat{v}\hat{u})^\omega} &= \pi(\binom{w}{w}\binom{\hat{v}}{\hat{v}}, \binom{\hat{u}}{\hat{v}}\binom{\hat{u}}{\hat{u}}, \binom{\hat{u}}{\hat{v}}\binom{\hat{u}}{\hat{u}}, \ldots) \\
&= \pi(\binom{w}{w}\binom{\hat{v}}{\hat{v}}, \binom{\hat{u}}{\hat{v}}\binom{\hat{v}}{\hat{v}}, \binom{\hat{u}}{\hat{v}}\binom{\hat{v}}{\hat{v}}, \ldots) \\
&= \pi(\binom{w}{w}\binom{\hat{v}}{\hat{v}}\binom{\hat{v}}{\hat{v}}, \binom{\hat{u}}{\hat{v}}\binom{\hat{v}}{\hat{v}}, \binom{\hat{u}}{\hat{v}}\binom{\hat{v}}{\hat{v}}, \ldots) \\
&= \pi(\binom{w}{w}\binom{\hat{v}}{\hat{v}}, \binom{\hat{v}}{\hat{v}}\binom{\hat{u}}{\hat{v}}, \binom{\hat{v}}{\hat{v}}\binom{\hat{u}}{\hat{v}}, \ldots) \\
&= \binom{w'(\hat{v}\hat{u})^\omega}{w'(\hat{v}\hat{v})^\omega}.
\end{aligned}$$

Therefore $(w'(\hat{v}\hat{u})^\omega, w'(\hat{v}\hat{v})^\omega) \in T$, and by transitivity also $(w'(\hat{u}\hat{u})^\omega, w'(\hat{v}\hat{v})^\omega) \in T$ i.e. $(w'\hat{u}^\omega, w'\hat{v}^\omega) \in T$. But this means that $(w\hat{v}\hat{u}^\omega, w\hat{v}^\omega) \in T$ contradicting the assumption that $(wv^k u^\omega, wv^\omega) \notin T$ for any k. □

Remark 3.1.13. Below we will make use of the observation that whenever a, b, w' satisfy the conditions of Lemma 3.1.12 then $\hat{a} = a^K$,$\hat{b} = b^K$ do as well with the same w' and for K the exponent of S_f. It then holds additionally that both $\binom{\hat{a}}{\hat{b}}$ and $\binom{\hat{b}}{\hat{a}}$ are idempotent.

Proof of Lemma 3.1.7. Towards a contradiction let us take an equivalence relation E with the corresponding finite ω-semigroup $S = (S_f, S_\omega)$ and morphism ϕ that recognises E and the words w, u, v so that $\phi\binom{u}{u} = \phi\binom{v}{v}$ and $(wu^\omega, wv^\omega) \notin E$ for all $k \in \mathbb{N}$.

Since E is transitive, let us take the words w', a, b given by Lemma 3.1.12 additionally assuming, by way of Remark 3.1.13, that $\binom{\hat{a}}{\hat{b}}$ and $\binom{\hat{b}}{\hat{a}}$ are idempotent as well. We are going to show that no two words in $w \cdot \{ab, ba\}^\omega$ that differ on infinitely many positions are in E and thus we find uncountably many words that are not equivalent.

We first show that the words $w'(ab)^\omega$ and $w'(ba)^\omega$ are not equivalent. Let us assume to the contrary that $(w'(ab)^\omega, w'(ba)^\omega) \in E$ and consider the pair of words $(w'(ba)^\omega, w'(abaa)^\omega)$.

$$\begin{aligned}
\binom{w'(baba)^\omega}{w'(abaa)^\omega} &= \pi(\binom{w'}{w'}, \binom{b}{a}, \binom{a}{b}, \binom{b}{a}, \binom{a}{a}, \binom{b}{a}, \binom{a}{b}, \binom{b}{a}, \binom{a}{a}, \ldots) \\
&= \pi(\binom{w'}{w'}, \binom{b}{a}, \binom{a}{b}, \binom{b}{a}, \binom{a}{b}, \ldots) \\
&= \binom{w'(ba)^\omega}{w'(ab)^\omega}
\end{aligned}$$

So $(w'(ba)^\omega, w'(ab)^\omega) \in E$ implies that $(w'(ba)^\omega, w'(abaa)^\omega) \in E$, and as E is an equivalence relation we have that $(w'(ab)^\omega, w'(abaa)^\omega) \in E$. However

$$\begin{aligned}
\binom{w'(abab)^\omega}{w'(abaa)^\omega} &= \pi(\binom{w'}{w'}, \binom{a}{a}, \binom{b}{b}, \binom{a}{a}, \binom{b}{a}, \ldots) \\
&= \pi(\binom{w'}{w'}, \binom{b}{a}, \binom{b}{a}, \ldots) \\
&= \binom{w'b^\omega}{w'a^\omega}
\end{aligned}$$

and thus $(w'b^\omega, w'a^\omega) \in E$ which contradicts Lemma 3.1.12.

Now, knowing that $(w'(ab)^\omega, w'(ba)^\omega) \notin E$, let us take two words x_1, x_2 of the form $w'(ab, ba)^\omega$ that differ on infinitely many positions. Using idempotency (i) and the right-identity property (v) we can show that either $\binom{x_1}{x_2} = \binom{w'(ab)^\omega}{w'(ba)^\omega}$ or $\binom{x_1}{x_2} = \binom{w'(ab)^\omega}{w'(ba)^\omega}$, depending on the first pair of letters that differ. Let us assume wlog. that the first pair of letters that differ is $\binom{a}{b}$, then group all positions where the letters in the pair are equal to $p_0, p_1, \ldots$, use idempotency and finally collect the other elements in the following way:

$$\begin{aligned}
\binom{x_1}{x_2} &= \pi\Big(\binom{w'}{w'}, \binom{p_0}{p_0}, \binom{a}{b}\binom{b}{a}, \binom{p_1}{p_1}, \binom{a}{b}\binom{b}{a}, \binom{p_2}{p_2}, \binom{a}{b}\binom{b}{a}, \ldots, \binom{p_3}{p_3}, \binom{b}{a}\binom{a}{b}, \ldots\Big) &&= \\
&= \pi\Big(\binom{w'}{w'}\binom{a}{a}, \binom{a}{b}, \binom{b}{a}\binom{a}{a}, \binom{a}{b}, \binom{b}{a}\binom{a}{a}, \ldots \binom{a}{b}, \binom{b}{a}\binom{a}{a}, \binom{b}{a}, \binom{a}{b}\binom{a}{a}, \ldots\Big) &&= \\
&= \pi\Big(\binom{w'}{w'}, \binom{a}{b}\binom{b}{a}, \binom{a}{b}\binom{b}{a}, \ldots \binom{a}{b}\binom{b}{a}, \binom{b}{a}\binom{a}{b}, \binom{b}{a}\binom{a}{b}, \ldots, \binom{b}{a}\binom{a}{b}, \binom{a}{b}\binom{b}{a}, \ldots\Big) &&= \\
&= \pi\Big(\binom{w'}{w'}, \binom{a}{b}\binom{b}{a}, \binom{a}{b}\binom{b}{a}, \binom{a}{b}\binom{b}{a}, \ldots\Big) &&= \\
&= \binom{w'(ab)^\omega}{w'(ba)^\omega} \quad .
\end{aligned}$$

By this calculation, any pair of words of the form $w'(ab, ba)^\omega$ that differ on infinitely many positions is in E exactly if $(w'(ab)^\omega, w'(ba)^\omega) \in E$ or $(w'(ba)^\omega, w'(ab)^\omega) \in E$, so by the previous argument these are not in E and thus represent different elements.

This concludes the proof of Lemma 3.1.7. $\square$

3.1.2 Decidable and Undecidable Problems

The Fundamental Theorem tells us that first-order properties of (*omega*-)(tree-)automatic structures are decidable. Given an injective presentation, decidability can be extended to $\mathsf{FO}^{\infty,\mathrm{mod}}$, the extension of first-order logic with infinity and modulo counting quantifiers (Theorem 3.1.5). We have seen that for finite word- and tree-automatic presentations injectivity does not constitute a restriction, but the case of infinite word- and tree-automatic presentations is still unsettled.

In Section 3.3.1 and Chapter 5 we will see, using interpretations or automata techniques, that in some cases the monadic theory of certain automatic structures is decidable. However, the example of the infinite grid (cf. Example 3.1.2 (vi)) shows that monadic-second order theories of automatic structures are in general undecidable.

Moreover, as seen in Example 3.1.2 (ix), configuration graphs of Turing machines are automatic, it is thus not hard to show that e. g. reachability, connectivity, isomorphisms or bisimulation of automatic graphs are undecidable by a reduction from the halting problem [BG04, Rub04, Rub07]. In [Rub04, Rub07] it is observed that isomorphism of automatic graphs is in fact much harder than that: it is complete for the Σ^1_1 level of the analytic hierarchy.

A strengthening of the fundamental decidability result can therefore only be hoped for very modest extensions of first-order logic (see Section 7.1), or for appropriate subclasses of automatic structures (see below).

Finiteness

As remarked after Definition 3.1.1 all finite structures are automatic. It is natural to ask whether given an automatic presentation of either kind finiteness of the represented structure is decidable. In general this amounts to deciding whether an (ω-)(word-/tree-) automatic equivalence relation is of finite index. Given an injective presentation, however, the problem is not new, it asks finiteness of the domain. This is well-known to be decidable for regular languages as well as for tree-regular languages. Since both word-automatic and tree-automatic presentations can be effectively converted to injective ones, we have a decision procedure for these two models. Finiteness of ω-regular languages is also easily seen to be decidable, for instance by appealing to Eq. (2.1) on page 17. A decision procedure for non-injective ω-automatic presentations is obtained from Theorem 3.1.9 and Corollary 3.1.11 of the previous section. In the case of automatic presentations over infinite trees a similar result is conjectured, however, at this point we cannot provide a proof.

3.2 Logical Interpretations as Presentations

Logical interpretations transform structures into structures in a way that an associated transformation reduces the logical theory of the interpreted structure to the theory of the host structure.

3.2.1 First-Order Interpretations

We have seen in the Fundamental Theorem that each of the classes (ω)(T)AutStr is closed under first-order interpretations. It is natural to ask whether there are maximal objects in each of these classes with respect to the partial order of FO-interpretability.

Definition 3.2.1 (Complete structures)**.** Following [BG04] we say that a structure $\mathfrak{A}$ is *complete* for a class $\mathcal{K}$ wrt. a class $\mathcal{L}$ of interpretations if $\mathfrak{A} \in \mathcal{K}$ and every $\mathfrak{B} \in \mathcal{K}$ is $\mathcal{L}$-interpretable in $\mathfrak{A}$.

The approach of Büchi to decidability of Presburger arithmetic $\mathcal{N} = (\mathbb{N}, +)$ is based on finite subset interpretations reducing the FO theory of $\mathcal{N}$ to the wMSO theory of $(\mathbb{N}, \mathrm{succ})$. The well-known correspondence of automata on finite words and (w)MSO on $(\mathbb{N}, \mathrm{succ})$ can be reformulated as $(\mathbb{N}, \mathrm{succ})$ is complete for AutStr wrt. subset interpretations, to be introduced in Section 3.2.2 below. Presburger arithmetic is, however, not complete for AutStr wrt. first-order interpretations. Büchi suggested the expansion $(\mathbb{N}, +, \{2^n \mid n \in \mathbb{N}\})$, which is still not complete. Expansions of $\mathcal{N}$ by relations of the form $x \,|_k\, y$ defined to hold precisely when x is a power of k dividing y were considered by Boffa and Bruyére, whence the following theorem.

Theorem 3.2.2 (Büchi-Bruyére, cf. [BHMV94]).
A relation $R \subseteq \mathbb{N}^r$ is regular in the (least-significant digit first) base k presentation of $\mathcal{N}$ iff R is first-order definable in the expanded structure $\mathcal{N}_k = (\mathbb{N}, +, |_k)$.

The above theorem implies that each $\mathcal{N}_k$ with $k > 1$ is complete with respect to FO-interpretations. This can be seen by appealing to the fact that every automatic presentation over an alphabet Σ can be trivially encoded as a presentation over $[k]$, provided $k > 1$, by encoding symbols of Σ on blocks of $[k]$-digits of uniform length [Blu99].

There are somewhat more natural structures complete for the classes AutStr and ωAutStr with respect to first-order interpretations.

Example 3.2.3 (Complete structures of [Blu99, BG04], cf. [ER66][Nab77]). Consider a finite alphabet Σ and let

$$\mathcal{S}_\Sigma = (\Sigma^*, \{\mathrm{succ}_a\}_{a\in\Sigma}, \preceq, \mathrm{el})$$

and

$$\mathcal{S}^\omega_\Sigma = (\Sigma^{\leq\omega}, \{\mathrm{succ}_a\}_{a\in\Sigma}, \preceq, \mathrm{el})$$

be structures defined on finite, respectively on finite and ω-words, comprising the successor relations $\mathrm{succ}_a = \{(w, wa) \mid w \in \Sigma^*\}$, the prefix relation $u \preceq w$ (where u is finite and w is finite or infinite) and with the equal-length relation: $u \,\mathrm{el}\, w$ iff $|u| = |w|$. These relations are clearly regular, respectively, ω-regular, thus $\mathcal{S}_\Sigma \in$ AutStr and $\mathcal{S}^\omega_\Sigma \in \omega$AutStr. Note that if Σ is unary, then $\mathcal{S}_\Sigma$ reduces to $(\mathbb{N}, \leq)$.

The structures of Example 3.2.3 are powerful enough to allow us to express, using first-order formulas, the existence of an accepting run of any given finite automaton, hence to define all regular relations. Therefore, they are indeed complete for AutStr wrt. FO-interpretations.

Theorem 3.2.4 (Complete automatic structures [Blu99, BG04]). *Let Σ be a finite,* non-unary *alphabet.*

(i) A relation R over Σ^* is regular if and only if it is definable in $\mathcal{S}_\Sigma$.

(ii) A structure $\mathfrak{A}$ is automatic if and only if it is first-order interpretable in $\mathcal{S}_\Sigma$.

Natural complete structures for the classes TAutStr and ωTAutStr will be derived from their characterisations via subset interpretations.

3.2.2 Subset Interpretations

Subset interpretations allow one to reduce the first-order theory of one structure to the monadic second-order theory of another. This is feasible when elements of the former structure can be interpreted as subsets of the latter. Büchi used this idea to show decidability of $(\mathbb{N}, +)$ by reducing it via a finite subset interpretation to the monadic theory of one successor (see Example 3.2.7 below), thus yielding an alternative proof of Presburger's decidability theorem using the "automaton method". Later, Elgot and Rabin [ER66] investigated decidability of extensions of both of these theories using automata techniques. Subset interpretations are defined as follows.

Definition 3.2.5. A *subset interpretation* $\mathcal{I}$ is given by a collection of monadic second-order formulas $(\varphi(X), \varphi_i(\vec{X}))$ each φ_i having only set variables free.
Given a structure $\mathfrak{A}$ of the appropriate signature the structure $\mathfrak{B}$ (finite) subset interpreted by $\mathcal{I}$ in $\mathfrak{A}$ has as its elements the (finite) subsets of $\mathfrak{A}$ satisfying φ and as its relations those defined by each of the φ_i.
We use the notation $\mathfrak{B} \leq_{\mathcal{P}}^{\mathcal{I}} \mathfrak{A}$ respectively $\mathfrak{B} \leq_{\mathcal{P}_\mathrm{f}}^{\mathcal{I}} \mathfrak{A}$ to specify which interpretation is meant.

To every (finite) subset interpretation $\mathcal{I}$ we associate, as usual, a transformation of formulas $\psi \mapsto \psi^{\mathcal{I}}$, in this case mapping first-order formulas to monadic second-order formulas as done in Section 2.6.1, mutatis mutandis. In the case of subset interpretations this transformation reduces the FO-theory of the interpreted structure to the MSO-theory of the host. The case of finite subset interpretations is a bit more subtle for formulas $\psi^{\mathcal{I}}$ yielded by the transformation use the auxiliary predicate of finiteness. Of course, whenever finiteness of subsets is MSO-definable in the host structure, fortunately for us we will only deal with this case, the reduction is sound. Another option is to use wMSO-formulas in the finite subset interpretation thereby reducing the first-order theory of the interpreted structure to the wMSO-theory of the host.

The next result relating the first-order theory of the interpreted structure to the monadic theory of the "host" and thus justifying the definition is, with the added remarks, commonplace.

Proposition 3.2.6. Let $\mathfrak{A}$ be a structure in which finiteness is MSO-definable, e.g. a finitely branching tree or a linear ordering, and let $\mathfrak{B} \leq_{\mathcal{P}_{(\mathrm{f})}}^{\mathcal{I}} \mathfrak{A}$ be a (finite) subset interpretation. Then to every first-order formula $\psi(\vec{x})$ in the signature of $\mathfrak{B}$ one can effectively associate a monadic formula $\psi^{\mathcal{I}}(\vec{X})$ in the signature of $\mathfrak{A}$ such that for every matching tuple $\vec{A}$ of elements of $\mathfrak{B}$, also seen as (finite) subsets of $\mathfrak{A}$, it holds that

$$\mathfrak{B} \models \psi(\vec{A}) \iff \mathfrak{A} \models \psi^{\mathcal{I}}(\vec{A}) \quad .$$

Consequently, if the monadic-second order theory of $\mathfrak{A}$ is decidable then so is the first-order theory of $\mathfrak{B}$.

Of course, the restriction of the definability of finiteness is only needed for finite subset interpretations. As a canonical example let us give here the natural finite subset interpretation of $(\mathbb{N}, +)$ in $(\mathbb{N}, \mathrm{succ})$.

Example 3.2.7. An interpretation $(\mathbb{N}, +) \leq_{\mathcal{P}_\mathrm{f}}^{\mathcal{I}} (\mathbb{N}, 0, \mathrm{succ})$ based on the binary representation is given by $\mathcal{I} = (\varphi(X), \varphi_+(X, Y, Z))$ with $\varphi(X)$ always true and

$$\varphi_+(X, Y, Z) = \exists C \forall n \left((Zn \leftrightarrow Xn \oplus Yn \oplus Cn) \wedge (C\mathrm{succ}n \leftrightarrow M(Xn, Yn, Cn)) \wedge \neg C0\right)$$

where C is of course for carry, $\oplus$ is exclusive or, and $M(x_0, x_1, x_2)$ is the majority function, in this case definable as $\bigvee_{i \neq j} x_i \wedge x_j$.

Next we give a finite subset interpretation of the complete structure $\mathcal{S}_2$ of Example 3.2.3 in $(\mathbb{N}, \mathrm{succ})$.

Example 3.2.8. The complete structure $\mathcal{S}_2 = (\{0,1\}^*, \mathrm{succ}_0, \mathrm{succ}_1, \preceq, \mathrm{el})$ is finite subset interpretable in $(\mathbb{N}, \mathrm{succ})$ in a straightforward way representing each finite word $u \in \{0,1\}^*$ by $U = \{n \mid u_n = 1\} \cup \{|u|\}$. The maximal element of each non-empty set is used to mark the length of the word, the correspondence is otherwise the standard one. The relations of $\mathcal{S}_2$ are easy to define, succ_0 for instance by

$$\varphi_0(X,Y) = \exists m(Xm \wedge \forall n(Xn \to n \leq m) \wedge \forall n(Yn \leftrightarrow Xn \wedge n < m \vee n = \mathrm{succ} m) \quad .$$

Given the Büchi-Rabin equivalence of word, respectively tree automata and monadic second-order logic on "the line" $\Delta_1 = (\mathbb{N}, \mathrm{succ})$, respectively on "the tree" $\Delta_2 = (\{0,1\}^*, \mathrm{succ}_0, \mathrm{succ}_1)$ all four notions of automatic presentations introduced in Definition 3.1.1 allow a straightforward yet fundamental reformulation in terms of subset interpretations. In the finite word case this was first discovered by Büchi [Büc60] and Elgot [Elg61], cf. also [ER66], [Blu99, BG04], [Rub04] and [Col04a] for generalisations.

Theorem 3.2.9 (Automatic presentations as subset interpretations). *Each of the four notions of presentation can be characterised in terms of subset interpretations in the line Δ_1 or in the tree Δ_2 as follows.*

(i) $\mathfrak{A} \in \textsc{AutStr}$ *iff* $\mathfrak{A} \leq_{\mathcal{P}_\mathrm{f}} \Delta_1$

(ii) $\mathfrak{A} \in \omega\textsc{AutStr}$ *iff* $\mathfrak{A} \leq_{\mathcal{P}} \Delta_1$

(iii) $\mathfrak{A} \in \mathrm{T}\textsc{AutStr}$ *iff* $\mathfrak{A} \leq_{\mathcal{P}_\mathrm{f}} \Delta_2$

(iv) $\mathfrak{A} \in \omega\mathrm{T}\textsc{AutStr}$ *iff* $\mathfrak{A} \leq_{\mathcal{P}} \Delta_2$

And the transitions from automatic presentations to subset interpretations and back are effective.

Let us define the *(finite) subset enveloping* $\mathcal{P}_{(f)}(\mathfrak{A})$ of a structure $\mathfrak{A}$ by adjoining its (finite) subsets as new elements endowed with the subset relation and identifying singleton subsets with the original elements they contain. Formally, given $\mathfrak{A} = (A, \{R_i\})$ we define $P_{(f)}(A)$ as the set of all (finite) subsets of A and let $\mathcal{P}_{(f)}(\mathfrak{A}) = (P_{(f)}(A), \{R_i'\}, \subseteq)$ with $\subseteq$ defined on $P_{(f)}(A)$ and $R_i' = \{(\{a_1\}, \ldots, \{a_n\}) \mid (a_1, \ldots, a_n) \in R_i\}$ for every n-ary R_i. It is now clear that

$$\mathfrak{B} \leq_{\mathcal{P}_{(\mathrm{f})}} \mathfrak{A} \iff \mathfrak{B} \leq_{\mathsf{FO}} \mathcal{P}_{(f)}(\mathfrak{A}) \qquad .$$

In particular, this yields natural complete structures with respect to first-order interpretations within each of the four classes.

Corollary 3.2.10.
$\mathcal{P}_f(\Delta_1)$ *is complete for* AutStr,
$\mathcal{P}(\Delta_1)$ *is complete for* ωAutStr,
$\mathcal{P}_f(\Delta_2)$ *is complete for* TAutStr *and*
$\mathcal{P}(\Delta_2)$ *is complete for* ωTAutStr.

3.3 Restrictions

Since their introduction there have been some attempts to find interesting subclasses of automatic structures via restricting presentations in certain aspects. Mainly there are three ways to go about defining subclasses: 1) by restricting the domain of presentations; 2) by restraining presentations of relations to having simpler than regular form; or 3) by considering the class of structures definable in a fixed automatic structure.

While the third option is unproblematic but rarely justified, the first two approaches are, despite some attempts (see the quite thorough [Blu99, Chapter 8.] on this matter), hard to apply. The difficulty seems to have been related to finding robust subclasses of synchronised rational relations as the usual restrictions on regular languages fail to extend to relations in a suitable way.

In this section we present some of the more reasonable, robust, and well studied restrictions of the notion of automaticity. The only new class introduced here is that of p-automatic structures which extends unary automatic structures in a natural way and can be characterised both in the spirit of 1) and 3) above.

3.3.1 Unary Presentations

A natural and strong restriction of the notion of automaticity is obtained when we confine ourselves to words over a unary alphabet. Unary automatic structures were introduced in [Blu99] and have since been repeatedly studied serving as a test-bed for analysing automaton presentable algebraic structures.

Definition 3.3.1 (Unary automatic structures)**.** 1AutStr is the subclass of AutStr containing those structures, which allow for an automatic presentation over a unary alphabet.

The class of unary automatic structures can easily be characterised both from a logical [Nab77, Blu99] as well as from a structural point of view (cf. notion of unwinding in [Rub04]). It is clear at a glance that the power of finite automata on unary words is extremely limited as unary words carry only the information of their length, which when sufficiently large can only be tested modulo some constant by any given finite automaton. Owing to this simplicity linear orderings, equivalence structures, permutation structures and the like having unary a.p. have been completely characterised. These results are concisely presented in [Blu99, Rub04].

We merely recite here the logical characterisations, which can already be found in [Nab77]. To this end, let $\mathfrak{M}$ be the structure $(\mathbb{N}, <, \{\equiv_{(\mathrm{mod}\, m)}\}_{m>1})$ and recall that Δ_1 denotes $(\mathbb{N}, succ)$.

Theorem 3.3.2. *For any structure $\mathfrak{A}$ the following are equivalent:*

1) $\mathfrak{A} \in 1\mathrm{AutStr}$

2) $\mathfrak{A} \leq_{\mathrm{one-dim-FO}} \mathfrak{M}$, *i.e.* $\mathfrak{A}$ *is* FO*-definable in* $\mathfrak{M}$

3) $\mathfrak{A} \leq_{\mathsf{MSO}} \Delta_1$

It follows from the last item that the MSO theory of every unary automatic structure $\mathfrak{A}$ is decidable. We can even be more specific by observing that the extension of $\mathfrak{A}$ obtained by adjoining the partial order of its finite subsets, its *finite subset envelope* (cf. Section 3.2.2), is trivially automatic. Indeed, finite subsets can be represented by their characteristic sequences (see Figure 7.1 in Section 7.1.3 for an example). Conversely, if the finite subset envelope of $\mathfrak{A}$ is automatic then $\mathfrak{A}$ has to be unary automatic. This follows from a far more general result of [CL06], a simple and direct proof will be given in Section 4.3.2.

3.3.2 p-Automatic Presentations

Growth arguments have proved to be a successful means of analysing different automatic presentations of certain structures, as well as for proving the impossibility of automatically presenting others. These methods are surveyed in Section 4.1 below. Here we look at presentations restricted in the growth of their underlying domain.

Our first observation is straightforward, therefore we omit the proof.

Proposition 3.3.3. A structure is unary automatic iff it has an automatic presentation over a universe with a growth rate bounded by a constant.

Thus the restriction to constant growth yields a fairly robust subclass, though a very confined one. One deficiency of unary presentations is that they are not preserved by multi-dimensional interpretations. Consider e.g. the grid as a direct product of two copies of $(\mathbb{N}, succ)$ having a two-dimensional interpretation in $(\mathbb{N}, succ)$. The latter is unary automatic, but the grid is clearly not as its monadic theory is undecidable. To overcome this we introduce the subclass of *p-automatic structures.*

Definition 3.3.4. A relational structure $\mathfrak{A}$ is *p-automatic* ($\mathfrak{A} \in \mathrm{PAutStr}$) if it has an injective automatic presentation over a domain of polynomial growth.

To underline the robustness of this class we first observe the following.

Proposition 3.3.5. The class PAutStr is closed under disjoint union, direct product, and first-order interpretations.

Proof. The claim follows from the fact that the class of regular languages of polynomial growth are closed under union, convolution product and under taking subsets. To quickly check the growth of the convolution we note that $|(D \otimes D')_{=2n}| \leq |D_{=n}| \cdot |D'_{\leq n}| + |D'_{=n}| \cdot |D_{\leq n}|$. □

Using pumping arguments, e.g. based on Proposition 4.1.1, it has already been shown in [Blu99] and [Rig01] that Presburger arithmetic $(\mathbb{N}, +)$ has no p-automatic presentation. On the other hand, the grid is p-automatic as shown in Example 3.1.2 (vi). Hence

$$1\textsc{AutStr} \subsetneq \textsc{PAutStr} \subsetneq \textsc{AutStr}\ .$$

Are there complete structures, with respect to first-order interpretations, within the classes of unary automatic or p-automatic structures?

As mentioned above, $\mathfrak{M} = (\mathbb{N}, <, \{\equiv_{(\mathrm{mod}\, m)}\}_{m>1})$ is complete for unary automatic structures under *one-dimensional* FO-interpretations (see Theorem 3.3.2). Obviously, $(\mathbb{N}, <)$ suffices if we allow modulo counting quantifiers (cf. Sec. 7.1) in interpretations. By the above, every structure FO-interpretable in $\mathfrak{M}$ is p-automatic irrespective of the dimension of the interpretation. We are able to show the converse as well allowing for a number of reformulations.

Theorem 3.3.6 (Logical characterisation of PAutStr)**.** *A structure is p-automatic iff it is first-order interpretable in $\mathfrak{M}$. In fact, for every structure $\mathfrak{A}$ the following are equivalent*[3]*:*

(1) $\mathfrak{A} \in \textsc{PAutStr}$

(2) $\mathfrak{A} \leq_{\mathsf{FO}} \mathfrak{M}$

(3) $\mathfrak{A} \leq_{\mathsf{FO}^{\mathrm{mod}}} (\mathbb{N}, <)$

(4) $\mathfrak{A} \leq_{\mathrm{multi-dim-}\mathsf{MSO}} \Delta_1$

(5) $\mathfrak{A} \leq_{\mathcal{P}_\mathrm{b}} \Delta_1$

Proof. (2) $\Rightarrow$ (1) : Clearly, $\mathfrak{M}$ is p-automatic and we have already seen that PAutStr is closed under first-order interpretations.

(1) $\Rightarrow$ (2) : Let us start by noting that if D is regular of polynomial growth, then so is the set $\mathsf{Pref}(D)$ of all prefixes of all words of D [SYZS92]. An automaton for $\mathsf{Pref}(D)$ can be obtained by setting all states final in an automaton for D. Regular sets of polynomial growth have been characterised by Szilard et al. [SYZS92] as those being a finite union of the form

$$D = \bigcup_{i<N} u_{i,1} v_{i,1}^* u_{i,2} \dots u_{i,n_i} v_{i,n_i}^* u_{i,n_i+1} \tag{3.1}$$

[3]Here $\leq_{\mathcal{P}_\mathrm{b}}$ stands for bounded subset interpretation, that is subset interpretation (Definition 3.2.5) over finite subsets of bounded size. Cf. Theorem 3.2.9

In terms of a minimal deterministic automaton this amounts to it not having two cycles reachable from one another. Such an automaton can thus be represented as a DAG of simple cycles labelled by the $v_{i,j}$ and intermediate edges labelled by the $u_{i,j}$ as above. Consider $w \in \mathsf{Pref}(D)$ with

$$w = u_{i,1}v_{i,1}^{r_1}u_{i,2}\ldots u_{i,j}v_{i,j}^{r_j}w'$$

for some i, $j \leq n_i - 1$, and w' a prefix of either $v_{i,j}$ or $u_{i,j+1}$. Let w' for the sake of illustration be a prefix of $v_{i,j}$. The idea is to represent w by the tuple

$$t(w) = \big(i, |u_{i,1}v_{i,1}^{r_1}|, |u_{i,1}v_{i,1}^{r_1}u_{i,2}v_{i,2}^{r_2}|, \ldots, |u_{i,1}v_{i,1}^{r_1}\ldots u_{i,j}v_{i,j}^{r_j}w'|, |w|, \ldots, |w|\big)$$

where the number of components k is the maximum of all the $n_i + 2$. Every such tuple $t = t(w) = (i, m_1, m_2, \ldots, m_k)$ can be identified with the convolution of its unary presentation $u = \otimes_{k+1}(1^i, 1^{m_1}, \ldots, 1^{m_k})$, which in turn can be represented as $v = f(w) = i\, a_1^{m_1} a_2^{m_2-m_1} \cdots a_k^{m_k-m_{k-1}}$, noting that the m_j do indeed form an increasing sequence.

Claim 3.3.7. f is a $(1,1)$-synchronous, and t is a $(1,k)$-synchronous translation

The translation from $f(w)$ to $t(w)$ is clearly $(1,k)$-synchronous, with the stretching by k needed to compensate for the convolution. More precisely it works by mapping each a_j to the sequence $\square^{j-1}1^{k-j+1}$. Moreover f is easily seen to be computable by a letter-to-letter transducer that counts the number of consecutive a_j's seen first up to $|u_{i,j}|$ then modulo $|v_{i,j}|$ while checking that the corresponding subsequence on the first tape matches $u_{i,j}v_{i,j}^*$. This proves the above claim.

Thus, by Corollary 4.2.14, the original presentation is equivalent to both the one obtained via the coding f and that via t. Our aim is now to characterise the relations regular under the coding t, or those regular under the coding f for that, as those being first-order definable in $\mathfrak{M}$. It is convenient to deal with relations under the coding f, i.e. with regular relations over the domain $U = a_1^* a_2^* \cdots a_{k+1}^*$.

For unary relations, that is subsets of U it is obvious that the corresponding sets of tuples of exponents are $\mathsf{FO}^{\mathrm{mod}}$-definable as on each subword $a_j^{r_j}$ the behaviour of any automaton is eventually periodic. Finally we note that the case of relations of higher arity simply reduces to the one just handled, as e.g. $U^{\otimes 2}$ is the finite union of all sets of the form $[a_{j_1}a_{l_1}]^*[a_{j_2}a_{l_2}]^* \cdots [a_{j_{k+1}}a_{l_{k+1}}]^*$ with $1 \leq j_1 \leq j_2 \leq \ldots \leq j_{k+1} \leq k+1$ and $1 \leq l_1 \leq l_2 \leq \ldots \leq l_{k+1} \leq k+1$. There is therefore technically hardly any difference between the unary and the higher arity cases.

(2) $\Rightarrow$ (3) : is obvious as for each m being divisible by m is definable using modulo counting quantifiers and only finitely many of these predicates can be used in an interpretation.

(3) $\Rightarrow$ (4) : is again trivial, since $<$ as well as the modulo counting quantifiers are definable in weak monadic second-order logic.

(2) $\Rightarrow$ (5) : holds for much the same reason as in (2) $\Rightarrow$ (3) $\Rightarrow$ (4) above, noting that the tuples we are actually using are all ordered according to length and thus can be identified with the set of their components.

(5) $\Rightarrow$ (4) : Once identifies each subset $\{n_1, \ldots, n_l\}$ of $\mathbb{N}$ of size $l \leq k$, wlog. $n_1 < \ldots < n_l$, with the extended k-tuple $(n_1, \ldots, n_l, n_l, \ldots, n_l)$ and adjusts the formulas of the interpretation accordingly.

(4) $\Rightarrow$ (1) : Appealing to the well-known correspondence between automata and monadic-second order logic, every multi-dimensional MSO interpretation in $(\mathbb{N}, succ)$ is easily seen to produce an automatic structure having a presentation in which each element, i.e. k-tuple $\{n_1, \ldots, n_k\}$, is represented by the convolution $\otimes_k(1^{n_1}, \ldots, 1^{n_k})$. The set of all such convolutions is of polynomial growth.

$\square$

Note that in the proof of Theorem 3.3.6 we made use of the fact that every p-automatic presentation over an $\mathcal{O}(n^k)$-growing domain is *equivalent* (see Section 4.2) to one over a subset of $a_0^* a_1^* \cdots a_k^*$. The latter presentations constitute therefore a kind of normal form of p-automatic presentations and can be used e.g. to show that a structure is not p-automatic.

A dichotomy

Recall the complete structures $\mathcal{S}_\Sigma = (\Sigma^*, \preceq, \{succ_a\}_{a\in\Sigma}, \mathrm{el})$ and consider their restriction to $\mathsf{Pref}(D)$ for some regular D. We observe the following dichotomy (see [SYZS92] for gap theorems on growths of regular languages).

Proposition 3.3.8. Let D be a regular set over some alphabet Σ and consider the structure $\mathcal{S}_D = (\mathsf{Pref}(D), \preceq, \{succ_a\}_{a\in\Sigma}, \mathrm{el})$.

(1) If D is of polynomial growth, then $\mathcal{S}_D$ is p-automatic, i.e. $\mathcal{S}_D \leq_{\mathsf{FO}} \mathfrak{M}$.

(2) If D is of exponential growth, then $\mathcal{S}_D$ is complete for AutStr wrt. $\mathsf{FO}^{\mathrm{mod}}$-interpretations, i.e. $\mathcal{S}_{[2]} \leq_{\mathsf{FO}^{\mathrm{mod}}} \mathcal{S}_D$.

Proof. If D is polynomially growing then so is $\mathsf{Pref}(D)$, which proves (1). To prove (2) we need to find a regular subset of $\mathsf{Pref}(D)$ that is binary branching. Consider a trim deterministic automaton $\mathcal{A}$ for D. Then $\mathcal{A}$ has a state q with two outgoing edges $q \xrightarrow{a} q_1$ and $q \xrightarrow{b} q_2$ for some $a \neq b \in \Sigma$ and states q_1 and q_2 from both of which q can be reached. For otherwise $\mathcal{A}$ would have the structure of a DAG of simple loops (each state would have at most one outgoing transition contributing to a simple loop all other edges contributing to the DAG structure) yielding a description of D as in (3.1) contradicting exponential growth.

Let v' and w' be words leading from q_1, respectively from q_2 back to q. Take $v = (av')^{|bw'|}$ and $w = (bw')^{|av'|}$, thus v and w are distinct labels of two loops of length $|v| = |w|$ from q to q. Let $u\Sigma^*$ be the label of a path leading to q from the initial state of $\mathcal{A}$.

The language $L = u\{v, w\}^*$ is an $\mathsf{FO}^{\mathrm{mod}}$-definable subset of $\mathsf{Pref}(D)$ containing exactly 2^n words of length $|u| + n|v|$. [4] To interpret $\mathcal{S}_{[2]}$ take the formula defining L and formulas defining succ_0 and succ_1 on L by appending v, respectively, w to each word. Prefix and equal length need only be restricted to L. $\square$

Complexity

The expansion of $\mathfrak{M}$ with the successor function succ and a constant for 0 admits quantifier elimination, meaning that every first-order formula of this expanded structure $\mathfrak{M}'$ is equivalent to quantifier-free formula. Hence, every p-automatic structure can be interpreted in $\mathfrak{M}'$ using quantifier-free formulas. Concerning the computational complexity of theories of p-automatic graphs we note that the PSPACE complexity bound of Blumensath for unary automatic structures [Blu99] extends to p-automatic structures as well, since it is preserved by first-order interpretations of arbitrary dimension. Although first-order model checking is as low of complexity as can be, adding even the most confined form of iteration to FO leads to undecidability.

The following example was pointed out to the author by Th. Colcombet.

Example 3.3.9 (Configuration graphs of Minsky-machines)**.** Minsky-machines are two-counter machines with finite control, they are Turing-complete and hence have an undecidable halting problem. To every k-counter machine with n states (wlog. $0 \leq q < n$) it is straightforward to construct a p-automatic presentation of its configuration graph representing each configuration $(q, n_1, \ldots, n_k)$ by the word $a^q c_1^{n_1} \cdots c_k^{n_k}$.

It follows that the first-order theory with reachability, $\mathsf{FO}[\mathbf{R}]$, of a p-automatic structure is in general undecidable.

3.3.3 Prefix-Recognisable Presentations

Prefix-recognisable graphs were introduced by Caucal in [Cau96] as a generalisation of context-free graphs, they have many equivalent characterisations, see Theorem 3.4.3 below, e.g. as those graphs monadic second-order interpretable in the infinite binary tree. The notion of prefix-recognisability was extended to relational structures by Blumensath in [Blu02].

In the context of automatic presentations we may say that a structure is prefix-recognisable if it has an automatic presentation in which every relation is prefix-recognisable.

Definition 3.3.10 (Prefix-recognisable relations)**.** A unary relation is prefix recognisable iff it is regular. For every $k > 0$ a $(k+1)$-ary relation $R(\vec{x})$ is PR if

[4] We could define L without counting quantifiers provided every bi-infinite word $\{v, w\}^{\mathbb{Z}}$ had a unique factorisation into v and w segments. This could possibly be ensured by a more clever choice of v and w yielding a sharper result.

$R = \bigcup_{\pi \in \mathrm{Perm}([k+1])} R_\pi(x_{\pi(0)}, \ldots, x_{\pi(k)})$ where each R_π is a finite union of relations of the form $\mathrm{id}_U(V \times W)$ where id_U is the $(k+1)$-ary identity restricted to the regular set U and both V and W are PR of arity $n \leq k$, respectively, $m \leq k$ such that $n + m = k + 1$.
The class of prefix-recognisable relations is denoted as PR, or as $\mathsf{PR}(\Sigma)$ when we wish to specify the alphabet.

As a simple example let us show that the *lexicographic ordering*, $<_{\mathrm{lex}}$, on an ordered alphabet Σ is prefix-recognisable. It is defined as the union of

$$\mathrm{id}_{\Sigma^*}(\varepsilon \times \Sigma^*) \quad \text{and} \quad \mathrm{id}_{\Sigma^*}(a\Sigma^* \times b\Sigma^*) \quad \text{for each } a < b \in \Sigma \ .$$

Caucal has shown that PR graphs are MSO-interpretable in the infinite binary tree and hence have a decidable MSO-theory. Actually, in [LS87] Läuchli and Savioz proved that MSO-definable relations on the binary tree coincide with prefix-recognisable relations over the binary alphabet (see also Proposition 6.2.1). Since then a number of equivalent characterisations of PR have been found, see Theorem 3.4.3 below, the strongest and most recent one of which is due to Colcombet [Col07b], as those structures having a one-dimensional first-order interpretation in the binary tree using the successors *and* the prefix relation, cf. Theorem 3.4.3 (7).

In our notation prefix-recognisable relations are in fact defined by suffix rewriting via rules of the form $s \to t$ for $s \in V, t \in W$ with the above notation. On words there is clearly no difference between the two ends as long as rewriting takes place consistently on one end only. On trees, however, this issue is much more subtle, see e.g. [Cau92a, Mey05].

The naming is due to Caucal owing to the representation of configurations of pushdown automata (PDA) as words $qw \in Q\Gamma^*$ where q is the current state and w represents the stack with its top symbol on the left. Transitions of a PDA correspond to prefix-rewriting steps. Conversely, Caucal [Cau92a] has shown that *prefix-rewriting* with rules of the form $v \to w$ where both v and w are words produce graphs effectively isomorphic to pushdown graphs. Thus, pushdown graphs are prefix-recognisable. See Section 3.4 below for more.

3.3.4 (Regular) Ground Term Rewriting

In the theory of term rewriting systems it is a natural restriction to consider systems defined by ground rules only. Ground terms are terms without variables, the leaves of the term tree of a ground term are labelled with constants. Ground rewrite rules $t \to t'$ consist of ground terms both on their left and right sides. A *ground term rewriting (*GTR*) system* is given by a finite set of rules $t \to t'$. Since the left hand sides of such rules are terms without free variables they can only be matched identically without substitution. Hence, ground rewriting always occurs around the leaves of term trees.

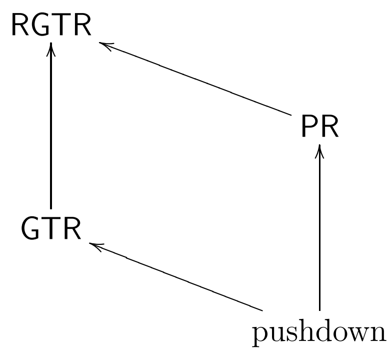
In the special case of trees consisting of a single branch, that is on words, these rules act as prefix-rewriting, i.e. pushdown transitions (see above). In this sense, ground rewriting of terms generalises prefix rewriting of words. Also observe how automata on words and bottom-up tree automata can be perceived as prefix-rewriting-, respectively, as ground term rewriting systems. E.g. a bottom-up transition $a(q_1, \dots, q_k) \mapsto q$ can be interpreted as a ground rule. The rules are thus monotone decremental and a tree is accepted by an automaton iff it reduces, after having attached initial states on its leaves, to a final state under the transition rules.

Prefix-recognisable relations extend prefix rewriting by being defined (in the binary case) by rules of the form $V \to W$, where V and W are arbitrary regular sets instead of individual words. Ground term rewriting can analogously be extended. A *regular ground term rewriting (*RGTR*) system* is given by a finite number of rules $T \to T'$ with T and T' regular sets of ground terms.

Every RGTR (GTR) system determines a graph. The vertices are the ground terms of a fixed signature (that are reachable from a given initial term[5]). Vertices are connected by i-labelled edges corresponding to one-step derivations according to the i'th rule. It follows immediately from the definition that GTR graphs are finite degree RGTR graphs (in fact, the converse is also true), which are in turn tree-automatic.

Regular ground term rewriting systems have been studied by Löding [Löd03] from the internal point of view of rewriting rules; and by Colcombet [Col02] form the external point of view of logical interpretations and equational definitions (see below). They have shown that every RGTR graph has a decidable first-order theory with reachability, gave different characterisations of these graphs and compared them to other classes of finitely presentable graphs. Some of these key results are postponed to sections to follow.

As noted, GTR systems generalise prefix-rewriting systems from words to trees, and similarly, RGTR systems generalise regular prefix-rewriting systems. In terms of their graphs this gives the following inclusion diagram:



One can in fact be more precise. Pushdown graphs are the prefix-recognisable graphs of finite degree, and analogously, GTR graphs are precisely the finite degree

[5]Löding [Löd03] considers terms reachable from a fixed "axiom" via rewriting steps, whereas Colcombet [Col02] works with the graph of all well-formed typed term trees, a condition recognisable by a deterministic top-down tree automaton.

RGTR graphs. Moreover, Löding proved that the pushdown graphs are precisely those GTR graphs of bounded tree-width. On the other hand, RGTR graphs of bounded tree-width are the HR-equational graphs forming a proper subclass of PR graphs (cf. Section 3.4 below). A structural characterisation of PR graphs relative to RGTR graphs is not known. [6] In [Löd03] it is also shown that a GTR graph is of bounded clique-width iff it is of bounded tree-width.)

We have observed that PR graphs are automatic, and similarly, that RGTR graphs are tree-automatic. Another result of [Löd03] that needs to be mentioned here states that GTR graphs of bounded out-degree are in fact automatic. This result relies on the observation that terms having only a bounded number of ground rewrites are "thin" in a certain sense which allows one to actually encode them as words over an appropriate alphabet of subtrees. Let us point out that for this construction it is crucial that one considers the rewriting graph restricted to the set of terms reachable from a given initial term.

3.4 Equational Presentations

In this section we briefly outline the general and novel algebraic approach of describing (hyper)graphs as (minimal) solutions of characteristic equations. This necessitates the introduction of a suitable algebraic structure over the universe of, say, countable vertex- and edge labelled (hyper)graphs. Note that hypergraphs with distinguished source nodes are just a synonym for relational structures with constants.

Historically, this approach is rooted in (hyper)graph rewriting and semantics of process calculi, originally conceived as a means of defining sets of finite (hyper)graphs akin to the use of grammars in formal language theory. Graph grammars operate much the same way as word grammars do by rewriting nonterminal vertices and (hyper)edges. Depending on the restrictions on the replacement rules one obtains different classes of "languages" of finite graphs closed under various algebraic operations. Alternatively, one may wish to start with trivial languages and proceed by defining complexer ones using a set of algebraic operations. Each term thus defines a set of (hyper)graphs much the same way as rational expressions define rational languages. For a comprehensive survey of graph rewriting see [Eng97].

Another use of grammars at the centre of our attention is to generate countable, typically infinite, graphs via complete rewriting. This can be formalised as a limit construction by way of iterative rewriting. On the algebraic side one describes such a limit as a least fixed-point solution of a system of equations in the appropriate algebraic setting. The equivalence of these two approaches is thus quite natural. What is more interesting for us now is how they compare to conceptually different means of finite presentations, primarily, as far as this work is concerned, to various automatic presentations.

[6] Löding conjectured that the right condition might be that of bounded clique-width, disproved by Colcombet [Col04b].

There is a great body of literature on various (hyper)graph rewriting frameworks, numerous variations on (hyper)graph algebras built on standard operations such as disjoint union, vertex recoloring, introduction of edges, series- and parallel composition, asynchronous product, synchronised product, fusion, quantifier-free definable operations. Whatever the set of operations of choice, finite terms represent finite (hyper)graphs via the evaluation mapping, which is a homomorphism from the free algebra of terms into the algebra of graphs under the chosen operations. Furthermore this homomorphism extends by continuity to infinite terms, which thus represent countable graphs [Cou90a]. Having established this correspondence places the powerful artillery of model-theoretic as well as tree-automata techniques at our disposal in the definition, analysis, algorithmics of classes of finite (hyper)graphs, respectively of countable (hyper)graphs. Moreover, it has been observed that for suitable choices of operations, most notably avoiding products, evaluation turns out to be a monadic second-order interpretation or transduction, thus strengthening the above tree-automata bond [CM02]. If one includes either the synchronous- or the asynchronous product among the operations then (appropriately restricted) subset interpretations provide a natural means of evaluating term-trees. Moreover, each of the mentioned evaluating interpretations extend, by continuity, to infinite term-trees [CC03, Col04a].

3.4.1 HR-equational graphs

Hyperedge replacement (HR) grammars are a very natural generalisation of grammars known in formal language theory going back to the 1980's. We shall not define (context-free) hyperedge replacement grammars formally here, but rather illustrate their working on an example. We only note that as the name tells they are given as a finite collection of rules that allow the replacement of a non-terminal hyperedge in a hypergraph by the right hand side of a matching rule, which is a given finite hypergraph with a number of distinguished vertices equal to the arity of the hyperedge to be replaced.

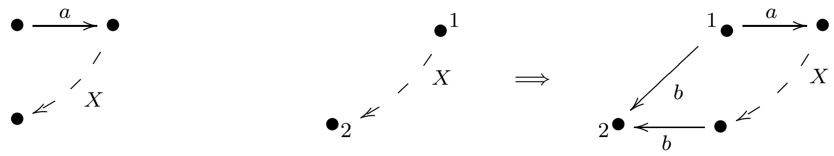
The class of HR-equational graphs properly extends that of pushdown graphs. Context-free graphs, that is configuration graphs of pushdown automata can be characterised as rooted HR-equational graphs of finite degree, which are also isomorphic to derivation graphs of prefix-rewriting systems on strings, a special case of prefix-recognisable graphs $\mathrm{id}_{U_i}(V_i \times W_i)$ with every V_i and W_i finite [Cau92b].

It is indeed quite easy to visualise the generation of a pushdown graph by a grammar. We illustrate this on the following example.

Example 3.4.1. Consider the pushdown automaton, depicted below, accepting the language $\{a^n b^n \mid n > 0\}$.

$$\begin{tikzcd}
q[\perp] \arrow[r, "a"] & q[a\perp] \arrow[r, "a"] \arrow[dl, "b"] & q[aa\perp] \arrow[r, "a"] \arrow[dl, "b"] & q[aaa\perp] \arrow[r, "a"] \arrow[dl, "b"] & \cdots \arrow[dl, "b"] \\
p[\perp] & p[a\perp] \arrow[l, "b"] & p[aa\perp] \arrow[l, "b"] & p[aaa\perp] \arrow[l, "b"] & \cdots \arrow[l, "b"]
\end{tikzcd}$$

It is generated by the deterministic hyperedge replacement grammar with initial graph and replacement rule given as follows.



Notice how the linearity of the pushdown graph is reflected in the linearity of the replacement rule having a single occurrence of the non-terminal X-labelled hyperedge on the right. To generate e.g. the infinite binary tree, which is of course a pushdown graph, a rule having at least two non-terminals on its right is needed.

The HR-algebra of graphs is many-sorted, having a separate sort n for graphs with n *sources* (i.e. constants). There are constants of each sort n: hypergraphs on n vertices, each a source, and with at most one hyperedge; and the following (overloaded) operations: *disjoint union* $\oplus$ (mapping sort n and m to sort $n+m$, involves shifting of source names), *renaming of sources* $\rho_{c\mapsto c'}$, and *fusion of sources* θ_ε according to an equivalence on sources. It should be clear how a vertex replacement step can be expressed using disjoint union with the right-hand side of the rule followed by a fusion and renaming of sources. For a detailed presentation of the HR framework the reader should consult [Cou90a, Bar98]. Notice that edges can only be "created" by fusion of sources. Since in a finite HR-equational system only a bounded number of source names are used, this considerably limits the pattern in which edges can be created connecting parts of the (hyper)graph defined by the system built in different stages of the iterative solution process. In particular, there is a bound on the size of complete bipartite subgraphs $K_{n,n}$ which can be created this way [Bar98], a feature which distinguishes HR-equational graphs from VR-equational ones, to be introduced right after the following characterisation theorem.

Theorem 3.4.2 (Barthelmann,Courcelle – cf. [Cou90a, Bar98, Blu01]).
For every countable infinite graph G the following are equivalent.

(1) G is generated by a deterministic HR *grammar;*

(2) G is HR*-equational, i.e. the interpretation of a regular* HR*-term, i.e. the least solution of a finite systems of* HR*-equations;*

(3) $\hat{G} \leq_{\mathsf{MSO}} \Delta_2$, the two-sorted adjacency graph $\hat{G}$ of G is monadic second-order interpretable in the infinite binary tree;

(4) G is VR*-equational and has bounded tree-width;*

(5) G is VR*-equational and, undirected, it does not contain $K_{n,n}$ for large enough n.*

3.4.2 VR-equational graphs

Vertex replacement systems are a finite collection of graph rewriting rules that allow one to substitute given finite graphs in place of single vertices. The corresponding VR-algebra of graphs is built on the following operations: constant graphs of a single c-coloured vertex $\mathbf{c}$, disjoint union $\oplus$, recolorings of vertices $\rho_{c\mapsto c'}$, and introduction of a-coloured edges from every c-coloured vertex to every d-coloured vertex.

The evaluation of VR-terms, whether finite or infinite, is realisable as a monadic second-order interpretation. As VR-equational graphs are interpretations of regular terms obtained by unfolding the finite system of equations, they can be MSO-interpreted in a regular tree, hence also in Δ_2. As a matter of fact, the converse also holds, together with a host of other equivalent characterisation.

Theorem 3.4.3 (Barthelmann,Caucal,Courcelle,Stirling – cf. [Blu01])**.**
For every countable infinite graph G the following are equivalent.

(1) G is generated by a deterministic VR *grammar;*

(2) G is VR*-equational, i.e. the interpretation of a regular* VR*-term, i.e. the least solution of a finite system of equations of the form $X_i = t_i(\vec{X})$ with finite* VR*-terms $t_i(\vec{X})$;*

(3) $G \leq_{\mathsf{MSO}} \Delta_2$;

(4) G is prefix-recognisable;

(5) $G = h^{-1}(\Delta_2)|_C$, i.e. the vertices of G are obtained by restricting the nodes of Δ_2 to a regular set C, and its edges by applying an inverse rational substitution h to Δ_2;

(6) G is the configuration graph of a pushdown automaton modulo ε-transitions.

Recently, Colcombet has proved that over trees every MSO-interpretation can be decomposed into a preparatory MSO-definable "marking" and a FO-interpretation using the prefix relation.

Theorem 3.4.4 (Colcombet [Col07b, Col07a])**.** *Every* MSO*-interpretation $\mathcal{I}$ can be effectively decomposed into an appropriate* MSO*-definable marking $\mathcal{M}$ (i.e. an interpretation keeping the original structure and enhancing it with additional labellings of vertices) and a suitable* FO*-interpretation $\mathcal{J}$, such that on all prefix-ordered trees $\mathcal{I}$ and the composition $\mathcal{J} \circ \mathcal{M}$ produce identical structures.*

The ingenious technique uses a deterministic, i.e. simultaneous, factorisation of the branches in the style of Simon's factorisation according to the finite semigroup recognising all word languages involved in the MSO-interpretation. (Provided that labels of nodes in a tree are augmented by sufficient type information on respective subtrees, every MSO formula has an equivalent normal form whose relevant constituents define regular path segments – this is precisely the idea behind the equivalence of (3) and (4) above). In the above decomposition the MSO marking is used to produce a labelling of the tree with information coding the "jumps" in the factorisation tree of the finite type-augmented path leading to each node (this is where the existence of a deterministic factorisation is vital) from which the first-order interpretation can recover the semigroup element corresponding to the path segment between any given pair of nodes $x \preceq y$. We note that the first-order interpretation only depends on the set of labels to be produced by the marking $\mathcal{M}$.

Since MSO-definable markings over Δ_2 are regular, and regular markings are themselves FO-interpretable in $(\Delta_2, \preceq)$ Colcombet obtains as a corollary the following characterisation.

Theorem 3.4.3 (... continued – cf. [Col07b, Col07a]).
(7) $G \leq_{\mathsf{FO}} (\Delta_2, \preceq)$, *i.e. G is first-order interpretable in the infinite binary tree using the prefix relation;*

Please note that this last characterisation is only valid when considering graphs *up to isomorphism*. The characterisation in terms of MSO-interpretability (3) is actually stronger in this sense. Indeed each prefix-recognisable relation over $\{0, 1\}$ is actually MSO-definable in Δ_2 *as is* (when one identifies words with nodes of the tree) in a very natural way. PR relations over a different alphabet can only be defined in Δ_2 modulo a coding of the alphabet, but can naturally be defined in the Σ-branching tree $(\Sigma^*, \{\mathrm{succ}_a\}_{a\in\Sigma})$ without any coding.

3.4.3 VRA-equational graphs

In the presence of some restricted product operations evaluation is no longer an MSO interpretation, however, it can be captured by subset interpretations. Though subset interpretations are not MSO-compatible, in the sense of Courcelle, they reduce the first-order theory of the interpreted structure to the monadic second-order theory of the host, but retain decidability of the FO theory. So those structures that are thus obtained by evaluating a regular term tree, or any term tree with a decidable MSO theory for that, do have a decidable first order theory, or even a $\mathsf{FO}[\mathbf{R}]$, first-order with reachability in a certain case. These results of Colcombet [Col02, Col04b] are summed up below.

The system VRA (in [Col02] VRP with P for product) extends the VR operations with *asynchronous* product. Although the evaluation of VRA-terms is no longer MSO-definable, it can be coded as a finite subset interpretation of a restricted kind, namely, in which, only antichains (equivalently prefix-closed subsets) of nodes of the term-tree represent elements of the encoded structure. In the following characterisation theorem RGTRS refers to Colcombet's model of regular ground term rewriting systems on well-typed term-trees. [7]

[7] This model differs from that of Löding, presented in Section 3.3.4, in that the trees representing nodes of a graph are not confined to those reachable from a chosen initial tree, but are

Theorem 3.4.5 ([Col02],[CC03]).
(1) The class of RGTRS*-graphs coincides with that of* VRA*-equational graphs, which are further characterised by being finite subset interpretable in regular trees using only antichains (alternatively, prefix-closed sets) of nodes.*

(2) Prefixset-, i.e. antichain-interpretations transform trees with a decidable MSO *theory into graphs with a decidable* $\mathsf{FO}[\mathbf{R}]$ *theory.*

In [CC03] the second statement of the above theorem was extended to arbitrary term-trees corresponding to solutions of infinite systems of VRA-equations. Applying antichain interpretations to e.g. tree-unfoldings of deterministic higher-order push-down graphs thus yields richer classes of graphs having many of the advantageous features of RGTRS graphs.

3.4.4 VRS-equational graphs

The system VRS (in [Col04a] VRC with C for Cartesian product) is obtained by adding the *synchronised* product operation to VR, whereas $\mathsf{VRS}_{\mathsf{fin}}$ is the extension by the unary operations of taking synchronised product of the argument with fixed finite graphs. Colcombet has shown that finite equational systems built with these operators define (after forgetting some auxiliary (hyper)edges) precisely the classes of tree-automatic, respectively, automatic (hyper)graphs.

To be precise, let $\mathfrak{d} = (\mathcal{A}, \{\mathcal{A}_i\})$ be an automatic presentation of $\mathfrak{A} = (A, \{R_i\})$, wlog. $A = L(\mathcal{A})$ and $\otimes R_i = L(\mathcal{A}_i)$ are regular relations over an alphabet Σ. Note that for now we do not have to distinguish between injective and non-injective presentations, the congruence of a presentation is now treated simply as one of the relations R_i. Consider the structure $\mathfrak{A}_{\mathfrak{d}} = (\Sigma^*, \{A^{p,q}\}, \{R_i^{p,q}\})$ with

$$R_i^{p,q} = \{\vec{u} \in (\Sigma^*)^{r_i} \mid \delta_i^*(p, \otimes\vec{u}) = q\}$$

for every i and every pair of states p, q of $\mathcal{A}_i$, and with the $A^{p,q}$ similarly defined. The structure $\mathfrak{A}_{\mathfrak{d}}$ is by definition automatic, in fact it encodes the presentation $\mathfrak{d}$ started with.

We are now in a position to state Colcombet's theorem: a relational structure $\mathfrak{A}$ is automatic iff it has an augmentation $\mathfrak{A}_{\mathfrak{d}}$ as above that is $\mathsf{VRS}_{\mathsf{fin}}$-equational. Clearly, $\mathfrak{A}$ can be obtained from $\mathfrak{A}_{\mathfrak{d}}$ by restricting the domain to $A^{q_0,F}$ and dispensing with those relations $R_i^{p,q}$ encoding partial computations. Thus, if we allow a quantifier-free interpretation to be applied as a final step after having obtained a least fixpoint solution of $\mathsf{VRS}_{\mathsf{fin}}$-equational systems then we can obtain all automatic relations and only these.

Moreover, combining Theorem 3.2.9 with the fact that the evaluation of VRS- and $\mathsf{VRS}_{\mathsf{fin}}$-terms can be realised as a subset interpretations we obtain the following threefold characterisation.

rather restricted to well-typed term-trees, a property recognisable by a deterministic top-down automaton

Theorem 3.4.6 ($\mathsf{VRS_{fin}}$-equational structures are automatic [Col04b, Col04a]).

$$\begin{array}{rcccl} \textsc{AutStr} & = & \leq_{\mathcal{P}_f}(\Delta_1) & = & \mathsf{VRS_{fin}}\text{−equational} \\ \textsc{TAutStr} & = & \leq_{\mathcal{P}_f}(\Delta_2) & = & \mathsf{VRS}\text{−equational} \end{array}$$

Finite subset interpretations transform trees with a decidable MSO *theory into graphs with a decidable* $\mathsf{FO}^{\infty,\mathrm{mod}}$ *theory.*

We conclude that the decidability result for each of the classes of (hyper)graphs defined in Sections 3.1 – 3.4.4 and the respective logics derive, via the appropriate notion of interpretation, from Rabin's MSO-decidability result for the infinite binary tree. For finite equational systems, that is. As already noted, the results via interpretations extend to arbitrary trees having a decidable MSO theory, e.g. to those obtained via repeated applications of the Shelah-Muchnik-Walukiewicz iteration (see Section 3.5.2 below). All in all, these results and techniques are grounded in the theory of tree automata and infinite games [GTW02].

3.5 Other means of Presentations

3.5.1 Rational graphs

Rational graphs are defined similarly to automatically presentable graphs. In a rational presentation vertices are labelled with finite words of a rational (i.e. regular) language over some finite alphabet Σ, and the edge relation(s) are require to be rational subsets of $\Sigma^* \times \Sigma^*$. Hence, this definition is more liberal in that it allows *asynchronous* automata in presentations of relations. The price to pay is the loss of tractability: rational graphs do not have a decidable first-order theory in general. The class of rational graphs strictly includes that of automatic graphs. With no appeal to completeness we list below some results on (asynchronous) rational graphs relevant in comparison to automatic (synchronised) ones. For a comprehensive treatment of rational graphs the reader is referred to the PhD thesis of C. Morvan [Mor01].

As noted, the undecidability of FO over rational graphs renders them useless for representing data, let alone programs for any practical means. In the context of formal language theory, however, rational graphs seem to fill a gap. Considering rational graphs as infinite automata, i.e. as acceptors of languages, Morvan and Stirling have shown that they trace all context-sensitive languages and only those [MS01, MR05], see also [CM05] for a simplified approach. In fact, this holds true for synchronised rational, i.e. for automatic graphs as well as first observed by Rispal [Ris02, MR05, CM05].

Although first-order queries on rational graphs are in general intractable, it has recently been shown by Carayol and Morvan that on rational graphs, which happen to be trees (an undecidable property) first-order logic is decidable [CM06, Mor06]. The decision method is based on locality of FO as formulated by Gaifman and uses a compositional technique. The authors also exhibit a finitely branching tree whose graph is rational but not automatic.

Another subclass of rational graph having a decidable first-order theory is that of rational graphs over a free commutative monoid [Mor01, Mor06]. These are the analogues of p-automatic graphs in the asynchronous model. Over the unary alphabet the monoid structure is isomorphic to $(\mathbb{N}, +)$ whence the unary rational graphs are those first-order definable in $(\mathbb{N}, +)$ [Mor01].

In their seminal paper [KN95] Khoussainov and Nerode also introduced asynchronous automatic structures. As examples they give an asynchronous automatic presentation of ω^ω, known to be non-automatic [KRS05, DGK].

Asynchronous automatic presentations of Cayley-graphs of finitely generated groups have also been considered as generalisations of "automatic groups" [CEH+92].

3.5.2 Caucal's pushdown hierarchy

As a generalisation of Rabin's tree theorem (Theorem 2.6.3), which also crucially relies on Rabin's tree automata technique is a fundamental result of Muchnik [Wal02, GTW02] establishing that the MSO-theory of a certain iteration $\mathfrak{A}^\#$ of a structure $\mathfrak{A}$ can be reduced to the MSO-theory of the original structure $\mathfrak{A}$. Thus, $Th_{\mathsf{MSO}}(\mathfrak{A}^\#)$ is decidable whenever $Th_{\mathsf{MSO}}(\mathfrak{A})$ is decidable.

The universe of $\mathfrak{A}^\#$ is the set A^+ of non-empty finite sequences of elements of $\mathfrak{A}$. For each relation R of $\mathfrak{A}$, $\mathfrak{A}^\#$ has a relation $R^\#$ defined as

$$R^\# = \{(\alpha a_1, \ldots, \alpha a_n) \mid \alpha \in A^+,\ (a_1, \ldots, a_n) \in R\} \ .$$

In addition to the above, the iteration is also equipped with relations

$$\mathsf{son} = \{(\alpha, \alpha a) \mid \alpha \in A^+,\ a \in A\} \quad \text{and} \quad \mathsf{clone} = \{(\alpha a, \alpha a a) \mid \alpha \in A^+\} \ .$$

Observe that if A is the set of level k stacks then A^+ is the set of level $k+1$ stacks and how the son and clone relations facilitate the definition of higher level push and pop operations.

Also observe that the tree *unfolding* $\mathfrak{T}_{\mathfrak{G},v}$ of a graph $\mathfrak{G}$ from a (definable, e.g. constant) vertex v is definable in $\mathfrak{G}^\#$. Thus, by the theorem of Muchnik, the decidability of the monadic theory of a graph is inherited by its tree unfolding. This result is considerably simpler to prove if the graph is assumed to be deterministic, i.e. if the neighbours of each node are unambiguously determined by the label of the edge leading there [CW98].

Together with the easy fact that MSO-interpretations preserve decidability of monadic theories of structures, these two MSO-*compatible* operations allow us to define a rich class of structures starting with finite graphs and alternatingly applying unfoldings and MSO-interpretations:

$$\begin{aligned}
\mathcal{G}raphs_0 &= \{\text{finite edge- and vertex-labelled graphs}\}\\
\mathcal{T}rees_{n+1} &= \{\mathfrak{T}_{\mathfrak{G},v} \mid (\mathfrak{G}, v) \in \mathcal{G}raphs_n\}\\
\mathcal{G}raphs_{n+1} &= \{\mathcal{I}(\mathfrak{T}) \mid \mathfrak{T} \in \mathcal{T}rees_{n+1}, \mathcal{I} \text{ is an } \mathsf{MSO} \text{ interpretation}\}
\end{aligned}$$

This hierarchy of trees and graphs was proposed by Caucal in [Cau02] using inverse rational mappings instead of the syntactically more general MSO-interpretations. In [Cau02] it was also shown that the hierarchy of term-trees within $\mathcal{T}rees_n$ coincides with that of term-trees generated by *safe* higher-order recursion schemes of level at most n of [KNU02]. In [CW03] Carayol and Wöhrle proved that graphs of $\mathcal{G}raphs_n$ are precisely the ϵ-closures of configuration graphs of higher-order pushdown automata of level n. Hence the name: pushdown hierarchy.

In [CW03] Carayol and Wöhrle show that all graphs of $\mathcal{G}raphs_n$ can be obtained via inverse rational mappings from deterministic trees of $\mathcal{T}rees_n$, implying that the assumption of unfoldings from definable vertices is not necessary. Moreover, the characterisation in terms of higher-order pushdown automata also yields that the same classes of trees and graphs are obtained if we use iteration instead of unfolding and MSO-transductions instead of interpretations. All of these various characterisations underline the robustness of these classes and the key role of the hierarchy in the study of transition systems.

A further strengthening was recently delivered by Colcombet. Recall Theorem 3.4.4 stating that MSO-interpretations can be written as a composition of an MSO-definable marking and a FO-interpretation. Thus, since $\mathcal{T}rees_n$ is closed under MSO-definable markings for every n [CW03], we could have defined $\mathcal{G}raphs_n$ as the set of graphs obtainable via FO-interpretations from trees of $\mathcal{T}rees_n$.

The level-zero graphs are the finite graphs, trees of level one are the regular trees, and as we have seen in Theorem 3.4.3 the level-one graphs are prefix-recognisable ones. Level-two trees are algebraic trees. From the second level onward we have no clear structural understanding of what kind of graphs inhabit the individual levels. While with considerable experience and effort one can construct individual graphs or families of graphs inside the hierarchy, on the other hand, it can be extremely challenging to prove that a given graph is not to be found on any level.

3.5.3 Simply-typed recursion schemes

In the previous subsection we have already mentioned higher-order recursion schemes. *Safe* schemes, to be precise. The general notion of higher-order schemes is a classical one [Dam82, Cou90b]. Schemes are a kind of deterministic grammars, a generalisation of context-free grammars, on simply-typed terms. The left- and right hand side of each rule can be understood as a name and a definition of a higher-order functional (combinator). The definitions may refer recursively to any of the functionals being defined. Thus, the solution is obtained by taking the simultaneous fixed points of the right hand side of each rule. The (typically infinite) term defined by a scheme is the fixed-point of a designated rule.

Safety is a technical restriction (implicit in [Dam82]) ensuring that no renaming of variables (α-conversion) is needed during the generative substitutive reduction (β-reduction) process constructing the solution term [AdMO05, Ong06]. We have mentioned that safe schemes are intimately related to the pushdown hierarchy. This connection is well explained in [AdMO05] showing that while on the one hand order-n schemes can define the behaviour and hence (the unfolding of) the configuration graphs of level-n deterministic pushdown automata, on the other hand, deterministic pushdown automata of level n can evaluate order-n schemes. For the latter, however, safety is essential.

In order to evaluate arbitrary schemes Ong et al. introduce higher-order collapsible pushdown automata, a kind of generalisation of panic automata. In [HMOS] a characterisation of term-trees of solutions of arbitrary higher-order schemes and graphs interpretable in them is given in terms of collapsible pushdown automata in the spirit of [CW03].

While convenient [KNU02] it is not necessary to assume safety for establishing decidability of the MSO-theory of the term-tree of the solutions of higher-order schemes. Indeed, Ong et al. [Ong06, HMOS] show that the term-trees of solutions of arbitrary higher-order recursion schemes have a decidable MSO-theory. We have to point out that their solution method, although, naturally involves tree automata, is radically different from that of the previous section based on unfoldings and interpretations. So for a good reason: there exists a collapsible pushdown automaton of level 2, the configuration graph of which has an undecidable MSO-theory [HMOS], and therefore cannot be constructed using MSO-compatible transformations from finite structures. Note, however, that μ-calculus remains decidable for higher-order collapsible pushdown graphs [Ong06, HMOS].

3.5.4 Generalised automatic structures

Finite subset interpretations transform trees with a decidable MSO theory into structures having a decidable $\mathsf{FO}^{\infty,\mathrm{mod}}$ theory. This facilitates a broad and novel generalisation of the notion of automaticity by classes of structures (finite) subset interpretable in given trees whose MSO theory is decidable [CL06]. The underlying idea is very simple. Fix a tree $\mathfrak{T}$, which is finitely presentable and which has a decidable MSO theory. Define the class of $\mathfrak{T}$-*automatic* structures as those finite subset interpretable in $\mathfrak{T}$. The interpretation together with the finite presentation of $\mathfrak{T}$ thus constitute a finite presentation of the interpreted structure. Moreover, every such structure has a decidable $\mathsf{FO}^{\infty,\mathrm{mod}}$ theory, and is first-order interpretable in $\mathcal{P}_f(\Delta_1)$

Recently Colcombet and Löding [CL06] investigated the power of finite subset interpretations. As their main combinatorial tool they proved the following theorem.

Theorem 3.5.1 ([CL06]). *Assume that for some $\mathfrak{A}$ its finite subset envelope, $\mathcal{P}_f(\mathfrak{A})$, is finite subset interpretable in some tree $t : [r]^* \to \Sigma$ seen as a structure $\mathfrak{T} = (\mathsf{dom}(t), \{P_a\}_{a\in\Sigma}, \{succ_i\}_{i<r})$. Then $\mathfrak{A}$ is* wMSO*-interpretable in $\mathfrak{T}$.*

In fact, the following stronger statement is also valid. To each finite subset interpretation $\mathcal{I}$ there exists an wMSO*-interpretation $\mathcal{J}$ such that for every tree $\mathfrak{T}$ and structure $\mathfrak{A}$ if $\mathcal{P}_f(\mathfrak{A}) \leq^{\mathcal{I}}_{\mathcal{P}_\mathrm{f}} \mathfrak{T}$, then $\mathfrak{A} \leq^{\mathcal{J}}_{\mathsf{wMSO}} \mathfrak{T}$.*

Observe that in the special cases of Δ_2 and Δ_1, the above theorem tells us that $\mathcal{P}_f(\mathfrak{A})$ is tree-automatic, respectively, automatic, iff $\mathfrak{A}$ is prefix-recognisable, respectively, unary automatic. In Section 4.3.2 we give a rather straightforward combinatorial proof the latter also extended to subset interpretations.

Using the above theorem it is easy to transfer strictness of the Caucal hierarchy to obtain an infinite hierarchy of generalised automatic structures [CL06]. Indeed, each level of the pushdown hierarchy contains a tree $\mathfrak{T}_n \in \mathcal{T}rees_n$ in which all of $\mathcal{G}raphs_n$ is wMSO-interpretable.[8] To show that $\mathfrak{T}_{n+1}$-automatic structures are all $\mathfrak{T}_n$-automatic, it is sufficient to check that $\mathcal{P}_f(\mathfrak{T}_{n+1})$ is not finite subset interpretable in $\mathfrak{T}_n$. If it were, then by Theorem 3.5.1 we had $\mathfrak{T}_{n+1} \leq_{\mathsf{wMSO}} \mathfrak{T}$ contradicting strictness of the pushdown hierarchy.

Another application of the above theorem is presented in Theorem 4.1.5.

3.6 Landscape and Summary

In this chapter we have surveyed numerous classes of finitely presentable infinite structures. The literature on these notions is vast and diverse, we have but highlighted a few of the key results, especially concerning equivalence of different approaches and comparison of the corresponding classes of graphs and structures. Automatic presentations being the central topic of this thesis, most attention has been given to the variants, restrictions, logical and equational characterisations of this notion. We have stated the most fundamental properties of automatic structures and provided a dozen or so examples. Below we will proceed with a more in-depth investigation of the potentials and the limitations of automatic presentations.

To close this chapter we present the inclusion graph of the various classes introduced illustrating the relationships among the notions given throughout this chapter. The diagram is an extension of that given in Löding's thesis [Löd03] and represents the work of the many researchers cited above. Our contribution is the introduction and characterisation of p-automatic graphs and the establishment of the fact that countable ω-automatic structures are automatic. The picture is still far from being entirely clear, some challenging problems remain, including: separating RGTR (VRA) from AutStr (see [Löd03] for a candidate graph); extending the results of Section 3.1.1 to (ω)TAutStr.

[8] $\mathfrak{T}_1 = \Delta_2$ and $\mathfrak{T}_{n+1}$ is the infinitely branching tree $\mathfrak{T}_n^{\#}$, i.e. for each n, $\mathfrak{T}_n$ is the tree of all level n pushdown stores over two stack symbols with edges marked with push operations of the appropriate level, in other words the free algebra over the unary operations $push_1(0), push_1(1), push_2, \ldots, push_n$ with a constant for the empty stack of level n.

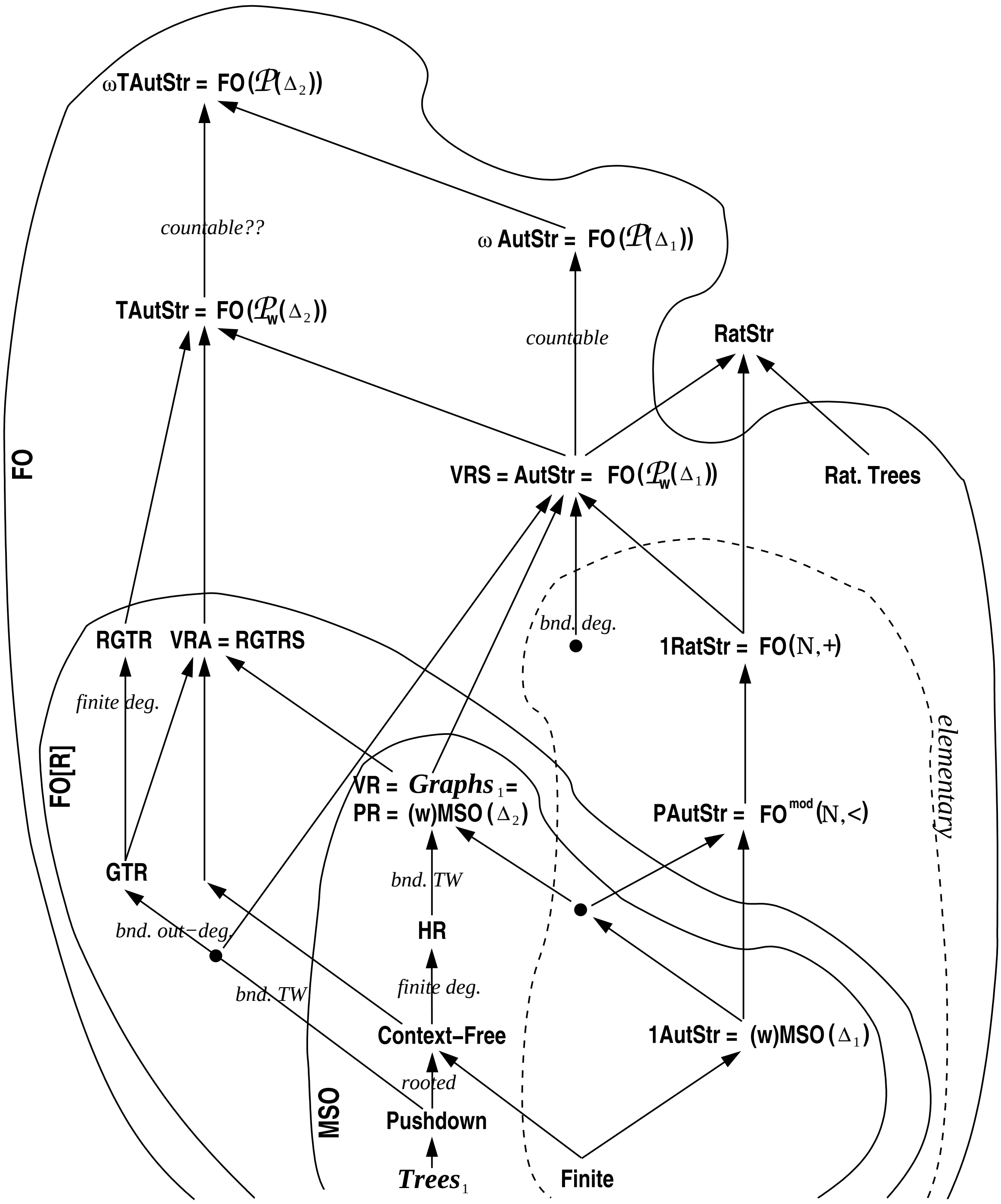


Figure 3.2: Landscape of classes of finitely presentable graphs (structures)

4 Analysis of Presentations

Growth functions of regular sets have been thoroughly studied in the context of formal language theory. We have already cited Szilard et al. [SYZS92] on the characterisation of regular sets of polynomial growth (polynomial density, in their terminology). In [PS95] Păun and Salomaa studied regular languages of bounded growth.

The classification of regular languages according to asymptotic growth is further underlined by the result of Maurer and Nivat [MN80] showing that there is a rational bijection between two infinite regular languages if and only if they are both of polynomial growth of the same degree or if they are both of exponential growth. Recently, Béal, Lombardy and Sakarovitch have proved the existence of a letter-to-letter bijection between any two regular languages of identical growth. In Section 4.2 below we will introduce the notion of equivalence of automatic presentations of a given structure and characterise it in a similar fashion in terms of semi-synchronous rational bijections between their domains.

4.1 General Tools: Pumping and Growth Arguments

This section is devoted to the investigation of how growth arguments can be used to obtain information about potential automatic presentations of particular structures. We have already provided an example of growth analysis applied to the domain of presentations in Section 3.3.2 where this simple approach has proved fruitful establishing the structural, expressive and computational complexity gap between automatic structures allowing a presentation over a polynomially growing domain and those which do not. The analysis of the latter call for more sophisticated techniques, e.g. measuring growth in reference to the structure.

Pumping and counting

To begin with we gather the most basic combinatorial facts on regular relations. The first of these is a straightforward consequence of the well-known "pumping lemma" of automata theory. A relation R of arity $n+m$ is *locally finite* if for every $(x_1, \dots, x_n)$ there are only finitely many $(y_1, \dots, y_m)$ such that $R(\vec{x}, \vec{y})$ holds. Obviously, every functional relation $f(\vec{x}) = y$ is locally finite. Other examples of locally finite relations are equal-length el, $|x| > |y|$ and the prefix relation $y \preceq x$. Note that local finiteness depends on the partitioning of the variables, e.g. $x \preceq y$ is not locally finite.

Proposition 4.1.1. (Elgot and Mezei [EM65]) Let $R \subseteq (\Sigma^*)^{n+m}$ be a regular and locally finite relation. Then there is a constant k such that $max_j|y_j| \leq max_i|x_i| + k$ holds for every $R(\vec{x}, \vec{y})$. In particular, if f is a regular function then there is a constant k such that for every $\vec{x}$ in its domain we have $|f(\vec{x})| \leq max_i|x_i| + k$.

The following helpful lemma on growth of regular sets appeared in [KNRS04]. To fix notation, for any regular set $D \subseteq \Sigma^*$ let $D_{=n} = D \cap \Sigma^n$ and $D_{\leq n} = D \cap \Sigma^{\leq n}$ denote the set of members of D of length precisely n and at most n respectively. Further let $\mathsf{Pref}(D)$ be the (regular) set of prefixes of words in D.

Proposition 4.1.2. [KNRS04, Lemma 3.12] Let $D \subseteq \Sigma^*$ be a regular set. Then

(i) $|\mathsf{Pref}(D)_{=n}| = \mathcal{O}(|D_{\leq n}|)$ and

(ii) for every fixed $c \in \mathbb{N} : |D_{\leq(n+c)}| = \Theta(|D_{\leq n}|)$

Growth of generations

From Proposition 4.1.1 one directly obtains a bound on the number of elements that can be generated by iterated applications of some automatic functions. Consider for instance a binary function $f(x, y)$ and assume it is automaton computable. The number $n(h)$ of f-terms of depth $\leq h$ satisfies the recurrence $n(h+1) = n(h)(n(h)+1)$ and is thus in the doubly exponential $2^{\mathcal{O}(2^h)}$ range. A word resulting from applying an f-term of depth $\leq h$ to words of length at most l is bounded in length by $kh + l$ for some constant k as in Proposition 4.1.1. As there are only $2^{\mathcal{O}(h)}$-many words of length at most $kh+l$ we see that no pairing function (one mapping pairs of elements to elements injectively, hence for which distinct terms yield distinct elements) can be automatic [BG04]. Even assuming associativity of f there are $2^{2^{\mathcal{O}(h)}}$ inequivalent f-terms in $r > 1$ variables, implying that no function acting freely on a subalgebra generated by more than one element is automatically presentable. In other words, the free monoid $(\{a, b\}^*, \cdot)$ of concatenation is not automatic [BG04].

In [KN95] and in [Blu99] the general approach illustrated on the previous examples is captured by the notion of generations. Consider a structure $\mathfrak{A}$ with functions $\mathcal{F} = \{f_1, \ldots, f_s\}$ and a sequence $E = \{e_0, e_1, e_2, \ldots\}$ of elements of $\mathfrak{A}$. The generations of E with respect to $\mathcal{F}$ are defined recursively as follows.

$$\begin{array}{lcl} G^0_{\mathcal{F}}(E) & = & \{e_0\} \\ G^{n+1}_{\mathcal{F}}(E) & = & G^n_{\mathcal{F}}(E) \bigcup \{e_{n+1}\} \bigcup \{f(\vec{a}) \mid f \in \mathcal{F}\,,\ \vec{a} \vec{\in} G^n_{\mathcal{F}}(E)\} \end{array}$$

From Proposition 4.1.1 one obtains.

Proposition 4.1.3 ([KN95],[Blu99, BG04])**.** Let $\mathfrak{A}$ be automatic and consider an injective presentation $\mathfrak{d}$ with naming function ν. Let $\mathcal{F}$ be a finite set of functions FO-definable in $\mathfrak{A}$ and $E = \{e_0, e_1, \ldots\}$ a definable set of elements ordered according to length in $\mathfrak{d}$, i.e. $|f^{-1}(e_0)| \leq |f^{-1}(e_1)| \leq \cdots$. Then there is a constant k such that for every n and for every $a \in G^n_{\mathcal{F}}$ $|f^{-1}(a)| \leq kn$. Therefore, $|G^n_{\mathcal{F}}| = 2^{\mathcal{O}(n)}$.

In other words, the number of elements that can be generated using functions in any automatic structure is at most a single exponential in the number of iterations. In [Blu99, BG04] this observation is cleverly used to show that Skolem arithmetic $(\mathbb{N}, \cdot)$ is not automatic (cf. Example 3.1.2(v) where a tree-automatic presentation of Skolem arithmetic is given.)

Using either of the above propositions it is straightforward to derive that if a group structure $(G, \cdot)$ is automatic, then every finitely generated subgroup of G has polynomial growth, meaning that for any chosen finite set $A = \{a_1, \dots, a_k\}$ the function $\gamma(n) = |\{\prod_{i<n} c_i^{\sigma_i} \mid \forall i < n : \ c_i \in A, \ \sigma_i \in \{1, -1\}\}|$ is bounded by a polynomial. Note that the presentation of G is not p-automatic unless G is finite. Using this easy fact, and powerful theorems of Gromov and Ersov, Nies, Oliver and Thomas concluded that if a group structure $(G, \cdot)$ is automatic then every finitely generated subgroup of G is virtually abelian. In particular, a finitely generated group has an automatic group structure iff it is virtually abelian [OT05]. Example 3.1.4 thus exhausts all automatic finitely generated group structures.

In [KNRS04] the applicability of Propositions 4.1.1 and 4.1.2 are pushed to their limits in showing, among a host of similar non-automaticity results, that no monoid having $(\mathbb{N}, \cdot)$ as submonoid is automatic, that no infinite integral domain is automatic, and that the countable atomless boolean algebra is not automatic.

Number of definable subsets

With the aim of proving non-automaticity of various structures obtainable as the Fraïssé limit of a suitable class of finite structures Khoussainov et al. develop in [KNRS04] a different approach more model theoretic in nature. Their technique involves counting the number of definable subsets of elements represented, in a tentative automatic presentation, on words of bounded length.

Consider $\mathfrak{A} \in \textsc{AutStr}$ with automatic presentation $\mathfrak{d}$ on domain $D \subseteq \Sigma^*$. Recall that $D_{\leq n} = D \cap \Sigma^{\leq n}$. To each first-order formula $\varphi(x, y, \vec{a})$ in the language of $\mathfrak{A}$ with parameters $\vec{a}$ from $\mathfrak{A}$ and to every $n \in \mathbb{N}$ we associate the function

$$C_{\mathfrak{d},n,\varphi}(b) = \{u \in D_{\leq n} \mid \mathfrak{A} \models \varphi(f^{-1}(u), b, \vec{a})\} \ .$$

The functions $C_{\mathfrak{d},n,\varphi}$ thus measure the extent to which definable families of subsets of the domain shatter the finite sets $D_{\leq n}$. As it happens, in an automatic structure for every φ however the parameter b is varied only linearly many subsets $C_{\mathfrak{d},n,\varphi}(b)$ of each $D_{\leq n}$ can occur.

Proposition 4.1.4 ([KNRS04])**.** In every automatic presentation $\mathfrak{d}$ of a structure $\mathfrak{A}$ and for every formula φ it holds that

$$|\{C_{\mathfrak{d},n,\varphi}(b) \ midb \in \mathfrak{A}\}| = \mathcal{O}(|D_{\leq n}|)$$

Khoussainov et al. conclude that the *random graph* $\mathfrak{R}$ together with various other random structures, equivalently, Fraïssé limits are not automatic. Indeed, the random graph, being the Fraïssé limit of all finite graphs, is characterised by the property that for every two disjoint finite sets of vertices U,V there is a vertex w connected to all elements of U and to no element of V. In other words, every finite set X of vertices is fully shattered (all $2^{|X|}$ subsets of X are isolated) by the edge relation as the parameter w is varied, contradicting Proposition 4.1.4.

Using subset interpretations Colcombet and the author have established non-automaticity of the random graph in a far more general sense: there is no finite subset interpretation of the random graph in any tree [CL06, Theorem 16], e.g. $\mathfrak{R}$ is also not tree-automatic. The result of [KNRS04] corresponds, in line with Theorem 3.2.9, to the case of the degenerate tree Δ_1.

Theorem 4.1.5 (The random graph is not automatic [CL06, Theorem 16])**.** *There is no tree t and no finite subset interpretation $\mathcal{I}$ such that $\mathfrak{R} \leq^{\mathcal{I}}_{\mathcal{P}_\mathrm{f}} t$.*

Proof idea. Intuitively speaking, the random graph contains its own finite subset envelope, respectively, the subset envelope of every finite graph as a subgraph. This can be exploited to devise, assuming a finite subset interpretation $\mathcal{I}$ of $\mathfrak{R}$ in a tree t, a finite subset interpretation $\mathcal{J}$, such that for every finite graph G there is an appropriate additional labelling t' of t such that $\mathcal{P}_f(G) = \mathcal{J}(t')$. Applying the main combinatorial theorem of [CL06] yields an MSO-interpretation $\mathcal{K}$ mapping each t' to the corresponding graph G, contradicting the fact that a class of graphs interpretable in a class of trees using a fixed MSO-interpretation is of bounded clique-width. □

Consider the more usual definition of a family of sets defined by φ with parameter b:

$$C_\varphi(b) = \{c \in \mathfrak{A} \mid \mathfrak{A} \models \varphi(c, b, \vec{a})\} \ .$$

In model theory, the *VC-dimension* of such a family is defined as the supremum of the sizes of finite subsets fully shattered by the family, i.e.

$$\sup\{|X| \text{ such that } |\{C_\varphi(b) \cap X\}| = 2^{|X|}\} \ .$$

Proposition 4.1.4 tells us that the sets $D_{\leq n}$ can only be shattered to a minimal extent by definable families. This is in contrast with the observation of Benedikt et al. [BLSS03] that in $\mathcal{S}_{[2]}$ each of the sets $\{0, 00, \ldots, 0^n\}$ can be fully shattered by the formula $\varphi(x, y) = \exists z(\succ_1 z \preceq y \wedge \mathrm{el}(z, x) \wedge)$ as y is varied.

4.2 Equivalent Presentations

In this section we develop a concise theory of simple transformations of automatic presentations in order to be able to distinguish between essentially different presentations as opposed to presentations identical modulo some trivial coding. Indeed, we should not consider a presentation different from an other one obtained from the former, say, by a permutation of the alphabet. In fact, we propose the following notion of equivalence, while arguing that results of this section will support our claim that this is indeed *the* right notion of equivalence.

Definition 4.2.1 (Equivalence of automatic presentations)**.**
Two presentations $(\mathfrak{d}_1, \nu_1)$ and $(\mathfrak{d}_2, \nu_2)$ of some $\mathfrak{A} \in$ AutStr are *equivalent* if for every relation R over $\mathfrak{A}$, $\nu_1^{-1}(R)$ is regular iff $\nu_2^{-1}(R)$ is regular.

In the following we shall consider bijective transformations, referred to as *translations*, of *injective* automatic presentations in connection with the notion of equivalence.

Definition 4.2.2 (Translations)**.** A *translation* is a bijection $t : D \to C$ between regular sets of words $D \subseteq \Sigma^*$ and $C \subseteq \Gamma^*$. If $D = \Sigma^*$ then t is a *total*- otherwise a *partial* translation. A translation t *preserves regularity (non-regularity)* if the image of every regular relation under t (respectively under t^{-1}) is again regular. Finally, t is *weakly regular* if it preserves both regularity and non-regularity.

Clearly, every bijective rational transduction qualifies as a translation, however, not necessarily weakly regular. It is easy to check that bijective semi-synchronous rational transductions are weakly regular. The aim of this section is to establish the exact converse of this.

We associate to each translation f its *growth function* G_f defined as $G_f(n) = max\{|f(u)| : u \in \Sigma^*, |u| \leq n\}$ for each n and say that f is *length-preserving* if $|f(x)| = |x|$ for every word x, further, f is *monotonic* if $|x| \leq |y|$ implies $|f(x)| \leq |f(y)|$ for every x and y, finally, f *has bounded delay* if there exists a constant δ such that $|x| + \delta < |y|$ implies $|f(x)| < |f(y)|$ for every x and y.

Let us look at the special case of length-preserving translations. Now it is known that every length-preserving rational transduction is in fact synchronised rational, cf. [FS93]. We show that this is true of all length-preserving and regularity-preserving translations. Note, however, that we do not assume a priori that translations are rational transductions. This result is interesting in its own right, and will also be key to our general characterisation.

Proposition 4.2.3. Let $f : D \to C$ be a length-preserving translation. If f preserves regularity of all relations on D then (the graph of) f is regular.

Proof. Consider $S_z = \{(u, v) \in D^2 \mid \exists v' : |v'| = |u| \wedge v'z \preceq v\}$, which is clearly regular for every $z \in \Sigma^*$. Their images under f are by assumption regular relations over C and in fact, since only the length of the first component plays a role in these relations, and it is preserved by f, the following "variants" over $D \times C$ are also regular.

$$R_z = \{(u, f(v)) \in D \times C \mid \exists v' : |v'| = |u| \wedge v'z \preceq v\} \quad (z \in \Sigma^*)$$

Let K be such that for every $n \in \mathbb{N}$ there is a word w in D of length $nK \leq w < (n+1)K$. Observe, that then every $u \in D$ is completely determined by the set of pairs (v, z) with $|z| \leq K$ and such that $S_z(v, u)$ holds. We can therefore define f using relations R_z with $|z| \leq K$ as

$$\text{graph}(f) = \{(u, x) \in D \times C \mid |u| = |x| \wedge \forall v \in D \bigwedge_{z \in \Sigma^{\leq K}} S_z(v, u) \to R_z(v, x)\}$$

This shows that the graph of f is indeed regular, i.e. that f is synchronised rational. □

Let it be mentioned that by a clever construction of [BLS06, Theorem 6] there is always a letter-to-letter automatic bijection between any two regular languages having exactly the same number of elements of every length.

Theorem 4.2.4 ([BLS06]).
Let $D \in \mathsf{Rat}(\Sigma^*)$ and $C \in \mathsf{Rat}(\Gamma^*)$ be two regular languages of identical growth, i.e. $g_D = g_C$. Then there is a length-preserving translation $t : D \to C$ computed by a letter-to-letter automaton.

Back to our task, we treat the general case via a series of equivalent transformations. Whereby we mean that two translations $f : D \to C$ and $g : D \to E$ over the same domain are *equivalent ($f \sim g$)* if for every $n \geq 1$ and for every relation $R \subseteq D^n$ either both $f(R)$ and $g(R)$ are regular or neither of them is. Obviously, composing a translation f with a weakly regular translation t we obtain $t \circ f \sim f$. In fact, keeping in mind that translations are by definition bijective, we have $f \sim g$ iff $f \circ g^{-1}$ is weakly regular. The next lemma gives a handy example of an equivalent transformation.

Lemma 4.2.5 (Padding).
To every translation $f : D \to C$ preserving the regularity of the relation $L(x, y) = |y| \leq |x|$ one can construct an equivalent monotonic translation $g : D \to C'$.

Proof. The relation $L(x, y) = |y| \leq |x|$ is locally finite and regular, so is its image $f(L)$. Therefore, by Proposition 4.1.1, there is a constant K such that $|y| \leq |x| \to |f(y)| \leq |f(x)| + K$ for every $x, y \in \Sigma^*$. The idea is to pad each image word $f(x)$ by an appropriate ($\leq K$) number of @'s, where @ does not occur in any of the alphabets involved.

By the choice of K above, we have $G_f(|x|) \leq |f(x)| + K$, and for each $s = 0..K$ the set $D_s = \{x : G_f(|x|) - |f(x)| = s\}$ is regular, being definable. This observation allows us to pad each codeword accordingly:

$$g(x) = f(x)@^{G_f(|x|)-|f(x)|} \qquad (\forall x \in D)$$

Note that $C' = g(D) = \bigcup_{s=1}^{k} f(D_s) \cdot @^s$ is regular.

To confirm that f and g are indeed equivalent one merely has to check that the translation $f(x) \mapsto g(x)$ is weakly regular, which is obviously true, it being definable, hence regular.

Finally, it is clear that g is monotonic, because $|g(x)| = G_f(|x|) = G_g(|x|)$ holds for every word x, and the growth function G_f is by definition always monotonic. □

Our next result shows that every regularity-preserving translation of bounded delay also preserves non-regularity of all relations, is thus weakly regular. This is achieved by showing that f is equivalent to a length-preserving translation satisfying the conditions of Proposition 4.2.3.

Theorem 4.2.6.
A translation is weakly regular iff it is regularity-preserving and has bounded delay.

Proof. The "only if" direction is easy to prove. We only need to show that every weakly regular f has bounded delay. Consider the equivalent presentation g obtained from f by padding each codeword with at most K new symbols as in Lemma 4.2.5. If g has bounded delay with bound δ then f has bounded delay with bound $\leq K\delta$. Assume therefore that f is monotonic. Then it suffices to consider the inverse image of the locally finite relation $L(x, y) = |x| \leq |y|$. Since $f^{-1}(L)$ is regular, the usual pumping argument by way of Proposition 4.1.1 shows that there is a constant d such that $|f^{-1}(x)| \leq |f^{-1}(y)| + d$ whenever $|x| \leq |y|$. In other words $|u| > |v| + d$ implies that $|f(u)| > |f(v)|$, which is to say, that f has bounded delay.

The converse "if" implication is proved by constructing in two steps of transformations a length-preserving and regularity-preserving translation h equivalent to f. The claim then follows by Proposition 4.2.3.

Again, as a first step we transform f using Lemma 4.2.5 into an equivalent monotonic translation. Henceforth we assume that f is monotonic. Next we establish that the growth function of f is in fact of a very restricted kind. This is key to showing that the second and decisive transformation performed in Lemma 4.2.8 does indeed produce an equivalent translation.

Lemma 4.2.7. Let $f : D \to C$ be a monotonic regularity-preserving translation of bounded delay. Then the infinite sequence of increments of the growth function of f, $\partial G_f = \langle G_f(1) - G_f(0), G_f(2) - G_f(1), \ldots \rangle$, is ultimately periodic.

Proof. Consider the language

$$L = \{x = f(u) \mid \forall y = f(v)(|u| = |v| \to x \leq_{llex} y)\}$$

collecting the length-lexicographically least element of $f(D_{=n})$ for each $n \in \mathbb{N}$. Because f preserves regularity of the equal-length relation the above definition yields that L is regular. Furthermore, since f has bounded delay, say with bound δ, it is δ-*thin*, meaning that there are at most δ many words in L of each length. We can thus write L as disjoint union of the regular languages $L_k = \{x \in L \mid \exists^{=k} y \in L\ |x| = |y|\}$ for $k = 1, \ldots, \delta$. Let us project L as well as L_k's onto 1^* in a length-preserving manner. G_f is a non-decreasing sequence of naturals in which each number can occur at most δ times. Due to monotonicity of f this projection of L corresponds, in the unary encoding, to the pruned sequence obtained from G_f by omitting the repetitions, whereas L_k is mapped onto those 1^n for which n is repeated exactly k times in G_f. All these projections are regular unary languages, which is the same as saying that the corresponding sets of naturals are ultimately periodic. The claim follows. □

We are now prepared to make the final transformation step. Lemma 4.2.7 below allows us to construct an equivalent length-preserving translation h by "factoring" each word $f(u)$ of length $G_f(|u|)$ into "blocks" according to ∂G_f.

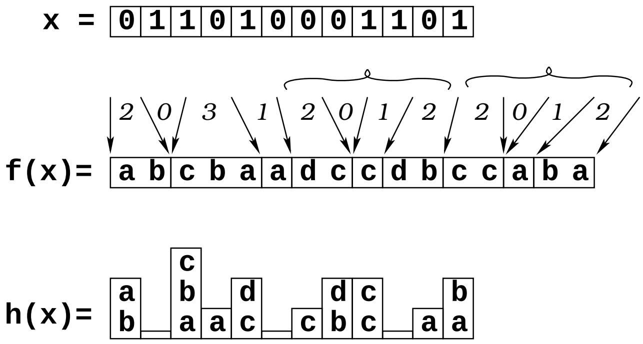


Figure 4.1: Factoring $f(x)$ along G_f with block sizes shown.

Lemma 4.2.8. Let $D \subseteq \Sigma^*$, $C \subseteq \Gamma^*$ be regular languages and let $f : D \to C$ be a monotonic regularity-preserving translation of bounded delay. Then one can construct an equivalent length-preserving translation $h : D \to C'$.

Proof. The fact that ∂G_f is ultimately periodic allows us to construct an equivalent length-preserving presentation h by subdividing codewords produced by f into blocks according to ∂G_f. (For this we need to assume that $G_f(0) = 0$, i.e. the empty word is represented by itself. Clearly, this is no serious restriction as changing a translation on a finite number of words always yields an equivalent translation.)

Consider some word $u \in D$ of size n and its image $v = f(u) \in C$. Since f is monotonic $|v| = G_f(|u|) = G_f(n)$ and we can factorise v as $v_1 v_2 \cdots v_n$ where $|v_i| = \partial G_f[i]$ for each $i \leq n$. Let $c = \max_n \partial G_f[n]$. Since $\partial G_f[i] \leq c$ for every i, we can consider each v_i as a single symbol of the alphabet $\Theta = \Gamma^{\leq c} = \{w \in \Gamma^* : |w| \leq c\}$. Let β be the natural projection mapping elements of Θ to the corresponding words over Γ, and let $\lambda(w) = |\beta(w)|$ for each $w \in \Theta$.

We define the mapping $h : D \to \Theta^*$ by setting for each $u \in D$, with factorisation as above, $h(u) = v_1 \cdot v_2 \cdot \ldots \cdot v_n$ when considered as a word of length n over Θ. This construction is illustrated in Figure 4.1.

Thus, h is by definition length-preserving and maps D injectively onto the set $C' = \{x \in \Theta^* \,|\, \beta(x) \in C \wedge (\forall i = 1..|x|)\ \lambda(x[i]) = \partial G_f(i)\}$. Because β is a homomorphism, C regular and ∂G_f ultimately periodic, C' can clearly be accepted by a finite automaton. Moreover, the fact that any two words w,w' belonging to C' are synchronously blocked (in the sense that $x[i]$ and $x'[i]$ have the same length for all $i \leq |x|, |x'|$) enables us to easily simulate any n-tape automaton $\mathcal{A}$ accepting a relation over C by an automaton $\mathcal{A}'$ accepting the "same" relation over C' and vice versa. $\square$

This concludes the proof of Theorem 4.2.6. $\square$

Observe that to establish Theorem 4.2.6 we have only made use of the fact that f preserved regularity of a handful of relations having to do with length comparison as well as those S_z of Proposition 4.2.3 matching a constant word z in one word at a position given by another. This does not come as a great surprise, as alone the relations S_z with $|z| \leq 1$ can define prefix, equal length, and the successor relations, and thus constitute a complete automatic structure. The following consequence of Theorem 4.2.6 is probably more surprising.

Corollary 4.2.9. *Consider a non-unary alphabet Σ and a total translation $f : \Sigma^* \to C$. If f preserves regularity of all relations over Σ^* then f is weakly regular.*

Proof. To apply the above theorem we only need to show that f has bounded delay. This is done in the lemma to follow, noting that monotonicity of f can be assumed using Lemma 4.2.5 as in Theorem 4.2.6. □

Lemma 4.2.10. Let $s = |\Sigma| \geq 2$ and let $f : \Sigma^* \to C$ be a monotonic regularity-preserving total translation. Then f has bounded delay.

Proof. Let $C_{\leq n} = \{x \in C \mid |x| \leq n\}$ for each $n \in \mathbb{N}$. Assume, that for some n and t we find the following situation.

$$G_f(n-1) < G_f(n) = G_f(n+1) = \ldots = G_f(n+t-1) < G_f(n+t)$$

Because f is total and monotonic we have $|C_{\leq G_f(n-1)}| = (s^n - 1)/(s-1)$ and $|C_{\leq G_f(n)}| = |C_{\leq G_f(n+t-1)}| = (s^{n+t} - 1)/(s-1)$ since these sets contain precisely the images of words of length at most $n-1$ and $n+t-1$ respectively.

Let K, by way of Proposition 4.1.1, be chosen such that $G_f(n) \leq G_f(n-1) + K$, hence $C_{\leq G_f(n)} \subseteq C_{\leq G_f(n-1)+K}$ for every $n \in \mathbb{N}$. From Proposition 4.1.2 we know that $|C_{\leq n+K}| \in \Theta(|C_{\leq n}|)$. Thus, there is a constant β (certainly, $\beta \geq 1$) such that $|C_{\leq G_f(n)}| \leq |C_{\leq G_f(n-1)+K}| \leq \beta \cdot |C_{\leq G_f(n-1)}|$. By simple arithmetic, $t \leq log_s(\beta)$, which proves that f has bounded delay, namely, bounded by $\delta = log_s(\beta) + 1$. □

Observe that Corollary 4.2.9 implies that the complete automatic structures $\mathcal{S}_\Sigma = (\Sigma^*, \{\mathrm{succ}_a\}_{a \in \Sigma}, \preceq, \mathrm{el})$ of Section 3.2.1 have only a single automatic presentation up to equivalence (see Theorem 4.3.1 below). Indeed, by completeness, the inverse of the naming function of every injective automatic presentation of $\mathcal{S}_\Sigma$ is a regularity-preserving translation that is also total by the definition of $\mathcal{S}_\Sigma$. Hence, by Corollary 4.2.9 the naming function itself is weakly regular and hence equivalent to the identity presentation.

Remark 4.2.11. We have to point out, that the assumption of Σ being non-unary is indeed essential. Corollary 4.2.9 fails for unary alphabets, because, as can easily be checked, the mapping from unary to binary presentation of the naturals does preserve regularity, but also maps some non-regular relations to regular ones. The same argument shows that the condition of f being total can not be dropped either: simply take a variant of the unary presentation e.g. over the partial domain ab^*.

4.2.1 Semi-synchronous transductions

Observe that we have actually proved more than what is claimed in Theorem 4.2.6. The above proof shows indeed that every regularity-preserving translation f of bounded delay (or every weakly regular translation for that) can be decomposed as

$$f = \pi^{-1} \circ \beta^{-1} \circ h$$

where π applies the padding, β the cutting of words into blocks, and where h is length-preserving and regular. Since both π^{-1} and β^{-1} are projections the composition is in fact a rational transduction.[1] But we can be a great deal more specific. We have also shown that ∂G_f is ultimately periodic, say from threshold N with period p. Let $q = G_f(N + p) - G_f(N)$ be the total length of *any* p consecutive blocks with sufficiently high indices. This means that after reading the first N input symbols and the first $G_f(N)$ output symbols a transducer accepting f can proceed by reading blocks of p input- and q output symbols in each step, which implies that f is in fact a (p, q)-synchronous transduction.

Theorem 4.2.12.
A translation is weakly regular iff it is a semi-synchronous transduction.

Proof. The "if" part is a special case of Lemma 2.2.3. To prove the "only if" part we repeat the same steps of transformations in the proof of Theorem 4.2.6. Thus, we obtain the same decomposition $f = \pi^{-1} \circ \beta^{-1} \circ h$, which shows, as argued above, that f is a semi-synchronous transduction. □

Corollary 4.2.13. *Two translations $f : D \to C$ and $g : D \to E$ are equivalent if and only if the translation $g \circ f^{-1}$ (hence also $f \circ g^{-1}$) is a semi-synchronous transduction.*

We have defined two automatic presentations of the same structure as equivalent if there is no difference between them in terms of representability of relations via automata, in other words, if they are expressively equivalent. In this section we have established that two injective automatic presentations are precisely then equivalent when the transduction

$$T = \{(x, y) \in D \times D' \mid \nu_1(x) = \nu_2(y)\}$$

translating names of elements from one presentation to the other, is semi-synchronous rational. Equivalent injective presentations are therefore truly identical modulo such a simple coding, i.e. expressive equivalence coincides with computational equivalence. This can in fact easily be extended to not necessarily injective presentations as well.

Corollary 4.2.14. *Two presentations $(\mathfrak{d}_1, \nu_1)$ and $(\mathfrak{d}_2, \nu_2)$ of the same structure are equivalent if and only if the transduction T, defined as above, translating names of elements from one presentation to the other, is semi-synchronous rational.*

[1] Knowing this, the claim of Proposition 4.2.3 already follows from [EM65, Corollary 6.6] (see also [FS93]) stating that length-preserving rational transductions are synchronised rational.

Proof. Assuming that T is semi-synchronous, one can transform any automaton recognising a relation in one presentation into another one recognising the same relation in the other presentation as done in Proposition 2.2.3.

Conversely, we can select a regular subdomain D'_i for both $i = 1, 2$ such that the thus restricted presentations are injective. These injective presentations are of course still equivalent. Thus, by Theorem 4.2.12, $T' = T \cap (D'_1 \times D'_2)$, being a weakly regular translation from D'_1 to D'_2, is semi-synchronous, say (p, q)-synchronous. Noting that $T = \varepsilon_2 \circ T' \circ \varepsilon_1$ where each ε_i is the congruence of the respective presentation it is easy to construct a non-deterministic (p, q)-synchronous transducer computing T by guessing a pair of words accepted by T' and checking their equivalence to the respective input words. This transducer can finally be determinised. $\square$

Mauer and Nivat have shown in [MN80] that there exists a rational bijection, i.e. a bijective rational transduction, between two infinite regular sets D and C if and only if their accumulated growth functions, $G_D(n) = |D_{\leq n}|$ and G_C defined analogously, are asymptotically equal, meaning that there are real numbers $c, d > 0$ such that $G_D(n) \leq G_C(cn)$ and $G_C(n) \leq G_D(dn)$ hold for all sufficiently large n. This holds precisely if both D and C are polynomially growing and with the same polynomial degree or if both are exponentially growing. As semi-synchronous transductions are by definition rational, this asymptotic equality must also hold for domains of equivalent presentations. More precisely, if $t : D \to C$ is a (p, q)-synchronous translation then there are constants k, l such that $G_D(n) \leq G_C(\frac{q}{p}n + k)$ and $G_C(n) \leq G_D(\frac{p}{q}n + l)$ holds for all n.

Finally, we note another use of the semi-synchronous transduction translating between equivalent presentations.

Corollary 4.2.15. *Let ν_1 and ν_2 be equivalent automatic presentations of $\mathfrak{A}$. Then there is a constant C such that for every n-ary relation R over $\mathsf{dom}(\mathfrak{A})$ and for every automaton $\mathcal{A}_1$ recognising $\nu_1^{-1}(R)$ there is an automaton $\mathcal{A}_2$ of size $|\mathcal{A}_2| \leq C^n \cdot |\mathcal{A}_1|$ recognising $\nu_2^{-1}(R)$, and vice versa.*

Proof. The automaton $\mathcal{A}_2$ is obtained from $\mathcal{A}_1$ by composing it with n copies of the transducer for T, one for each component. $\square$

4.3 Case Studies

4.3.1 Complete Structures

Recall the structures $\mathcal{S}_\Sigma = (\Sigma^*, \{succ_a\}_{a\in\Sigma}, \preceq, \mathrm{el})$ and $\mathcal{N}_k = (\mathbb{N}, +, |_k)$ from Section 3.2.1 where it was shown that for $|\Sigma|, k > 1$ they are each complete for AutStr with respect to first-order interpretations. Using results of Section 4.2 we can show that all of their presentations are equivalent.

Theorem 4.3.1. *Let Σ be a non-unary finite alphabet and $k > 1$. The complete structure $\mathcal{S}_\Sigma$ has, up to equivalence, only a single automatic presentation. The same result carries over to the structures $\mathcal{P}_f(\Delta_1)$, and $\mathcal{N}_k = (\mathbb{N}, +, |_k)$ for each k.*

Proof. Observe that for every injective presentation $(\mathfrak{d}, \nu)$ of $\mathcal{S}_\Sigma$ the naming function ν^{-1} is (by completeness) a regularity preserving and (having domain Σ^*) total translation. It follows from Corollary 4.2.9 that ν^{-1} is weakly regular. Therefore all injective presentations of $\mathcal{S}_\Sigma$ are equivalent.

Every automatic presentation is equivalent to an injective one obtained from it by restriction to a regular set of unique representants. Indeed, if R is the representation of an n-ary relation then $R' = R \cap L^n$ is the representation of the same relation in the injective presentation and $R = \sim \circ R' \circ \sim$. We see that R is regular iff R' is. This proves the claim for $\mathcal{S}_\Sigma$.

Towards establishing the claim for $\mathcal{P}_f(\Delta_1)$ consider an interpretation of $\mathcal{S}_{[2]}$ in $\mathcal{P}_f(\Delta_1)$ based on the following bijective representation. For each non-empty word x let x^- denote its immediate predecessor in the length-lexicographic order, and let x be represented by the set $\{|x^-|\} \cup \{n \mid x^-[n] = 1\}$; finally, let ε be represented by the empty set. Clearly, there is such an interpretation. Thanks to the bijective encoding, we can consider every presentation of $\mathcal{P}_f(\Delta_1)$ as one of $\mathcal{S}_{[2]}$. In fact, inequivalent presentations of the former would result in inequivalent presentations of the latter, which cannot happen.

Similarly, $\mathcal{S}_{[k]}$ can be bijectively interpreted in $\mathcal{N}_k$ for each k, so the same argument applies. $\square$

Not all complete structures have this property. Let $\mathfrak{C} = \mathfrak{A} \uplus \mathfrak{B}$ be the disjoint union of $\mathfrak{A}$ and $\mathfrak{B}$ having an additional unary predicate A identifying elements belonging to $\mathfrak{A}$. Thus, $\mathfrak{A}$ and $\mathfrak{B}$ are trivially FO-interpretable in $\mathfrak{C}$, and $\mathfrak{C} \in \textsc{AutStr}$ iff $\mathfrak{A}, \mathfrak{B} \in \textsc{AutStr}$. It follows from Theorem 2.2.4 above, that $\mathfrak{A} \uplus \mathfrak{B}$ has infinitely many pairwise inequivalent automatic presentations, provided both $\mathfrak{A}$ and $\mathfrak{B}$ are infinite. In particular, this holds for the complete structure $\mathcal{S}_\Sigma \uplus \mathcal{S}_\Sigma$. Let us therefore say that a structure is *rigidly automatic* if it has but one automatic presentation up to equivalence. Finite structures are trivially rigidly automatic.

Conjecture 4.3.2. *Every infinite rigidly automatic structure is complete.*

Let us remark that by the theorem of Cobham-Semenov (Theorem 4.3.4 below) Presburger arithmetic has infinitely many pairwise inequivalent presentations, whence no structure interpretable in $(\mathbb{N}, +)$ is rigidly automatic. In particular, (cf. Theorem 3.3.6), p-automatic structures are not rigidly automatic.

Furthermore, we are able to show that infinite prefix-recognisable structures are not rigidly automatic either. See Section 6.4 and Theorem 6.4.3 below.

4.3.2 Subset Envelopings

What does Theorem 3.5.1 spell out in the special cases corresponding to the tree Δ_2 or to the line Δ_1?

Recall that the (finite) subset envelope $\mathcal{P}_{(w)}(\mathfrak{A})$ of a structure $\mathfrak{A}$ is obtained by adjoining to $\mathfrak{A}$ the $\subseteq$-ordered structure of its (finite) subsets and by identifying elements of $\mathfrak{A}$ with the corresponding singleton sets.

A structure is MSO interpretable in the tree or in the line iff it is prefix-recognisable, hence automatic (cf. Sections 3.3.3 and 3.4.2), respectively, if it is unary automatic (cf. Section 3.3.1). Thus, the result tells us that the finite subset envelope $\mathcal{P}_f(\mathfrak{A})$ of a structure $\mathfrak{A}$ is tree-automatic iff the structure is prefix-recognisable, and that $\mathcal{P}_f(\mathfrak{A})$ is word-automatic iff $\mathfrak{A}$ is unary automatic. Inspecting the proof of Theorem 3.5.1 [CL06] we can even conclude, by a remark of Colcombet, that on the line the restriction to finite subsets can be lifted. That is, a subset envelop $\mathcal{P}(\mathfrak{A})$ is ω-automatic iff $\mathfrak{A}$ is unary automatic.

We would like to note that the technique of [CL06] is far more general than what we are using in the context of automatic presentations. Indeed, they consider interpretations in arbitrarily coloured trees, which makes the combinatorics substantially more involved. It is therefore perhaps instructive to present simple combinatorial proofs of the mentioned facts concerning automatic presentations of subset envelopes.

Proposition 4.3.3. [CL06] For every countable structure $\mathfrak{A}$ the following implications hold.

(i) $\mathcal{P}_f(\mathfrak{A}) \in \textsc{AutStr} \Rightarrow \mathfrak{A} \in 1\textsc{AutStr}$
In fact, an equivalent unary presentation of $\mathfrak{A}$ can be extracted from the presentation of its envelope.

(ii) $\mathcal{P}(\mathfrak{A}) \in \omega\textsc{AutStr} \Rightarrow \mathfrak{A} \in 1\textsc{AutStr}$
Again, a unary presentation of $\mathfrak{A}$ can be extracted from the presentation of its envelope.

Proof.
The "if" direction is straightforward in both cases: every unary-automatic presentation can be seen as a monadic second-order interpretation in "the line". Each such interpretation can be extended to a (finite) subset-interpretation of the (finite) subset enveloping in the obvious way. Therefore, in each case only the "only if" direction requires some consideration.

(i) *A quick proof:* Given an a.p. $\mathfrak{d} = (D_0 \uplus D_P, \{R_i\}, R_\subseteq, R_\in)$ of $\mathcal{P}_f(\mathfrak{A})$ we can give an a.p. $\mathfrak{d}' = (D_0 \uplus D_P, \mathit{succ}, R_\subseteq, R_\in)$ of $\mathcal{P}_f(\Delta_1)$ just by defining the successor function on D_0 according to the lexicographic ordering. By Theorem 4.3.1 all automatic presentations of $\mathcal{P}_f(\Delta_1)$ are equivalent. Thus, $\mathfrak{d}$ incorporates a presentation of $\mathfrak{A}$ equivalent to a unary one and a presentation of the subset structure equivalent to the natural binary one.

(i) *A direct proof:* Consider a (wlog. injective) a.p. $\mathfrak{d} = (D, \{R_i\}, R_\subseteq, R_\in)$ of some $\mathcal{P}_f(\mathfrak{A})$. By definition D is partitioned into two regular subsets D_0 and D_P of words representing elements, respectively finite subsets of $\mathsf{dom}\mathfrak{A}$. An a.p. $\mathfrak{d}_0 = (D_0, \{R_i\}$ of $\mathfrak{A}$ can therefore be extracted from that of $\mathcal{P}_f(\mathfrak{A})$. Our aim is to prove that $\mathfrak{d}_0$ is in fact *essentially* unary. By this we mean that D_0

is thin, i.e. there is a constant k such that D_0 contains at most k words of any given length n. This can be verified by a quick counting argument.

By Proposition 4.1.1 there is a constant K such that each subset every element of which is represented by a word of length n is itself represented on length at most $n + K$. (To see this consider the locally finite relation defined as $\forall z(zR_iny \rightarrow |z| \leq |x|)$.)

Let t_n respectively s_n be the number of words of D_0 of length n, respectively of length $\leq n$. By the above all 2^{s_n} subsets of these elements are represented by words not longer than $n + K$. Assuming, wlog., a binary presentation, there are less then 2^{n+K+1} such words available. This gives $s_n \leq n + K + 1$ for every n, therefore, by a gap theorem of growth functions of regular languages [SYZS92], (t_n) must be bounded by a constant L.

Finally, a unary presentation of $\mathfrak{A}$ equivalent to $\mathfrak{d}_0$ is constructed by "stretching" by a factor of L: the k-th word (e.g. in the lexicographic ordering) of length n is mapped to 0^{Ln+k}. This transformation is obviously semi-synchronous, hence the equivalence (cf. Corollary 4.2.14).

(ii) By Theorem 3.1.9 we can assume that $\mathfrak{A}$, as a definable substructure of $\mathcal{P}(\mathfrak{A})$, is injectively represented even if the whole subset extension is not. Thus, by Proposition 3.1.6, $\mathfrak{A}$ is automatic.

In [KL06] Kuske and Lohrey proved that over injective $\omega-$automatic presentations $\mathsf{FO}^{\infty,\mathrm{mod}}$-definable relations are ω-regular. We apply their result to the injectively presented subdomain of singletons. The set of ω-words representing a finite subset of $\mathfrak{A}$ is thus ω-regular, since it is definable in $\mathcal{P}(\mathfrak{A})$ with the $\exists^\infty$ quantifier applied only to singletons. Therefore, the given ω-automatic presentation of $\mathcal{P}(\mathfrak{A})$ includes one of $\mathcal{P}_f(\mathfrak{A})$. This, being countable, can, by results of Section 3.1.1, be filtered to be injective and effectively transformed into an essentially equivalent automatic presentation on finite words showing that $\mathcal{P}_f(\mathfrak{A})$ is automatic. The claim now follows from (i).

□

In the terminology of [CL06] item (ii) above would be formulated as for every subset-interpretation $\mathcal{I}$ we can construct an MSO-Interpretation $\mathcal{I}'$ such that whenever $\mathcal{P}(\mathfrak{A}) \leq^{\mathcal{I}}_{\mathcal{P}} \Delta_1$ then $\mathfrak{A} \leq^{\mathcal{I}'}_{\mathsf{MSO}} \Delta_1$.

4.3.3 Presburger Arithmetic

By Presburger arithmetic we mean both the structure $\mathcal{N} = (\mathbb{N}, +)$ and its first-order theory. It should always be clear which is meant. Presburger proved decidability of the first-order theory of $\mathcal{N}$ (actually that of $(\mathbb{Z}, +)$) using the technique of quantifier elimination. Büchi applied his automaton method to show decidability of Presburger arithmetic. His approach consisted in interpreting $\mathrm{Th}_{\mathsf{FO}}(\mathcal{N})$ in $\mathrm{Th}_{\mathsf{MSO}}(\mathbb{N}, succ)$ by a finite subset interpretation.

The straightforward interpretation uses of course the binary presentation (as in Example 3.2.7). The automaton method easily extends to representations in any natural base k. It was observed early that these representations are inequivalent, i.e. that a set of naturals may or may not be regular in different bases. This lead to the investigation of regularity in different number bases. A concise and accessible survey of results in this area is given in [BHMV94].

We have already seen an interpretation of the Büchi-Bruyére Theorem 3.2.2 in proving completeness of certain expansions of Presburger arithmetic. In order to state the celebrated Cobham-Semenov theorem we need the following definition. We say that p and q are multiplicatively independent if they have no common power, in other words if $p^k \neq q^l$ for all $k, l \geq 1$. Otherwise they are multiplicatively dependent.

Theorem 4.3.4 (Cobham-Semenov [2], cf. [BHMV94, Bés00, Muc03])**.**
Consider $p, q \geq 2$. The following dichotomy holds.

(i) If p and q are multiplicatively dependent then every relation $R \subseteq \mathbb{N}^r$ is regularly presented in base p iff it is regularly presented in base q.

(ii) If p and q are multiplicatively independent then a relation $R \subseteq \mathbb{N}^r$ is regularly presented in both base p and base q iff R is FO-definable in $\mathcal{N}$.

In our terminology the first case can be stated as for multiplicatively dependent p and q the two presentations base_p and base_q are equivalent. More precisely, if $p^k = q^l$ then there is a (k, l)-synchronous translation from base p to base q. On the other hand, for independent bases p and q the theorem tells us that the base p and base q presentations are as far apart as they can be. Indeed, by the Fundamental Theorem 3.1.3 every relation first-order definable in $\mathcal{N}$ is regular in every presentation of $\mathcal{N}$.

Generalised numeration systems

In [Bés00] Bés has extended Theorem 4.3.4 to *generalised numeration systems*. The theory of generalised numeration systems [Fro02] is concerned with representations of the naturals as well as the reals in various bases and using different, possibly negative digits. In general, the basis $U_0 < U_1 < U_2 < \dots$ of the system does not have to be the sequence of powers of a natural, one considers bases satisfying appropriate linear recursions, alternatively, powers of a base β, which is the greatest root of a polynomial of a certain type.

The study of generalised numeration systems goes back to Rényi who in 1957 introduced β-expansions, but can be traced back much further to the work of Cauchy (cf. [Fro02]). Naturally associated to the representation based on β-expansions is the β-shift defining a symbolic dynamical system. Thus, the theory of generalised numeration systems is closely connected to symbolic dynamics, Cantor sets, notion

[2] Cobham proved it for sets, later Semenov extended it to arbitrary relations.

of topological entropy, descriptive set theory, and of course to number theory [BP97, AS03].

Without going into the particulars of this very rich field we need to point out the fact that from a practical perspective one is interested in *normalised* numerals obtained via the *greedy* algorithm suggested by Rényi. Normalised numerals are thus naturally ordered according to length and then lexicographically. This ordering being automatic and given a regular set of (normalised) numerals $N \subseteq [d]^*$ over the set of digits $0, \ldots, d-1$ one is left with an automatic presentation of $(\mathbb{N}, <)$ of the form $(N, <_{\mathrm{llex}})$. A fundamental question in this context asks under which circumstances can addition be computed by an automaton?

It is easy to see that addition is regular in a generalised numeration system if normalisation of numerals over the extended digit set $0, \ldots, 2d-1$ is automaton computable. Indeed, digit-wise addition followed by normalisation of the result provides an automatic procedure for addition. Usually one considers numeration systems associated to a sequence of basis elements satisfying a linear recurrence. It is known that if the characteristic polynomial of the linear recurrence is the minimal polynomial of a Pisot number then normalisation, hence also addition, is automaton computable [Fro02].

A prominent example of a generalised numeration system with regular addition is the so-called *Fibonacci numeration system*.

Example 4.3.5. The Fibonacci numeration system has the Fibonacci numbers $1, 2, 3, 5, 8, \ldots$ as its basis and the binary digit set. The normalised numerals delivered by the greedy algorithm are $\varepsilon, 1, 10, 100, 101, 1000$, 1001, 1010, 10000, 10001, ... in the length-lexicographic ordering. They are the binary strings avoiding 11 as a factor, since, by the recursive identity $F(n+2) = F(n+1) + F(n)$ and greedy normalisation prefers 100 to 11. Naturally, 10^n represents the nth Fibonacci number.

Automatic presentations

We have mentioned that certain generalised numeration systems considered in the field can be conceived as automatic presentations of Presburger arithmetic. But are there any essentially different automatic presentations?

There are two aspects that have to be taken into account in order to classify all automatic presentations of $\mathcal{N}$ up to equivalence. One concerns the growth of the domain of numerals, the other their ordering.

It has been observed by Blumensath [Blu99, Lemma 5.3] and by Rigo [Rig01] that $\mathcal{N}$ is not p-automatic. (In $\mathcal{N}$ we can define Δ_2, which is not p-automatic simply because the number of nodes of depth n is exponential in n necessitating exponential growth of the domain by Proposition 4.1.1.) As we have hinted, in the context of generalised numeration systems addition is strongly connected to normalisation, and that the regularity of normalisation delicately linked with the alignment of roots of the characteristic polynomial of the linear recurrence satisfied by the growth function.

Regarding the representation of the ordering we can say that all automatic presentations of $(\mathbb{N}, +)$ known to us are based on numeration systems and are thus either comprised of normalised numerals obtained by the greedy enumeration, or equivalent to one such presentation. Let us quickly remark that for addition to be automaton computable the numerals have to be supplied in a *least-significant-digit-first* manner. This means of course that the ordering of the numerals is then also reversed, i.e. defined by comparison of length followed by the reverse lexicographic comparison.

We conjecture that the greedy enumeration is in fact an essential feature.

Conjecture 4.3.6. *Every automatic presentation of Presburger arithmetic is equivalent to one in which numerals are ordered in the reverse length-lexicographic manner.*

5 Automatic Words – a hierarchy of higher-order morphic words

This chapter is devoted to the investigation of automatically presentable infinite sequences (ω-words) over a finite alphabet and is based on the paper [Bár06a]. Note that we consider word structures utilising the ordering predicate on positions, hence we are also engaging in an investigation of automatic presentations of $(\mathbb{N}, <)$.

Word structures

To every omega-word $w \in \Sigma^\omega$ we associate in a standard way its *word structure* $\mathfrak{W}_w = (\mathbb{N}, <, \{P_a\}_{a \in \Sigma})$ having unary relations $P_a = \{n \mid w[n] = a\}$ for each $a \in \Sigma$ partitioning $\mathbb{N}$ according to the symbol occurring in each position. Note that we consider the ordering, as opposed to the successor relation, as given in our word structures. When one is working with monadic second-order logic, there is of course no difference in terms of expressiveness. However, as we are engaging in an investigation of automatically presentable word structures, the presence of the ordering, as opposed to just having the successor relation, is not without significance.

Automatic presentations

In accordance with Definition 3.1.1 an automatic presentation $(D, R, \{P_a\}_{a \in \Sigma})$ of $\mathfrak{W}_w$ as above comprises a regular set D partitioned by the regular sets P_a for each $a \in \Sigma$ over some alphabet Γ, together with a regular relation R, which is a linear ordering of type ω over D and such that the i-th word in this ordering belongs to P_a iff the i-th symbol of w is a. Elements of D can be seen as numerals, each $x \in D$ representing the number $\nu(x)$ where ν is the coordinate map of the presentation. For readability we identify x with $\nu(x)$ and write e.g. $w[x]$ in place of $w[\nu(x)]$ when indexing symbols or factors of w.

5.1 Morphic words and regular numeration systems

In the literature the most frequently, if not exclusively, used regular ordering of type ω is the *length-lexicographic* ordering, also called military-, radix-, or genealogical ordering by some and shortlex by others. Starting point of our investigation is the observation that those words possessing an automatic presentation using the length-lexicographic ordering are precisely those *morphic*, as demonstrated in Proposition 5.1.4 below.

Morphic words

A particularly well understood class of ω-words is that of the so called *morphic words*. The basic idea, invented and cleverly applied by Thue, is to obtain an infinite word via iteration of a suitable morphism $\tau : \Sigma^* \to \Sigma^*$. Suitability is expressed by the condition that $\tau(a)[0] = a$ for some $a \in \Sigma$. In this case τ is said to be *prolongable on* a. This ensures that the sequence $(\tau^n(a))_{n\in\mathbb{N}}$ converges to either a finite or infinite word, which is a fixed point of τ, denoted $\tau^\omega(a)$. An ω-word $w \in \Gamma^\omega$ is morphic, if $w = \sigma(\tau^\omega(a))$ for some τ prolongable on a and some $\sigma \in \mathsf{Hom}(\Sigma^*, \Gamma^*)$ extended in the obvious way to ω-words.

Example 5.1.1. Consider $\tau : a \mapsto ab, b \mapsto ccb, c \mapsto c$ and $\sigma : a, b \mapsto 1, c \mapsto 0$ both homomorphically extended to $\{a, b, c\}^*$. The fixed point of τ starting with a is the word $abccbccccbc^6b\ldots$, and its image under σ, $11001000010^61\ldots$, is the characteristic sequence of the set of squares.
In general, as was shown in [CT02], the characteristic sequence of every set of the form $\{\sum_{k=0}^{n} s_k \mid n \in \mathbb{N}\}$, where $0 < (s_k)$ is an $\mathbb{N}$-rational sequence is morphic. This result follows trivially from the characterisation of [RM02], cf. Proposition 5.1.4.

Morphic words and their relatives have been extensively studied in the context of formal language theory, Lindenmayer systems and combinatorics on words. For once an ordering is fixed (in this case to be length-lexicographic) the emphasis is on combinatorial aspects, such as number of finite factors of given length, and the growth of their re-occurrences, etc.

We will first show that morphic words can be naturally characterised as a subclass of automatic words representable using the length-lexicographic ordering. Then we will go on to generalise both notions to obtain an infinite hierarchy of "higher-order" morphic words inside the class of automatically presentable infinite words.

Automatic sequences

The theory of the so called *automatic sequences* [AS03] studies ω-words representable in more-or-less standard numeration systems. Presentations of primary concern are of a natural base k, or of base $-k$ and possibly involving negative digits.

Sequences representable in a natural number base k using the standard digits $[k] = \{0, \ldots, k-1\}$ are very well understood. These *k-automatic sequences* have been characterised both in algebraic, and in logical terms as being definable in $(\mathbb{N}, +, |_k)$ – one of our complete structures for AutStr – and also as morphic words that are fixed points of *uniform* morphisms on k symbols [BHMV94].

The prominent example of a 2-automatic word is the "ubiquitous Prouhet-Thue-Morse Sequence" [AS99].

Example 5.1.2. Consider the morphism $\tau : 0 \mapsto 01, 1 \mapsto 10$. Its fixed point $\tau^\omega(0)$ is the *Thue-Morse sequence* $t = 01101001100101101001\ldots$. This is a truly remarkable sequence bearing a number of characterisations [AS99]. For instance, its nth digit is a 1 iff the binary numeral of n contains an odd number of 1's. A key property of t is that it is uniformly recurrent without being ultimately periodic [AS99]. Moreover, t is overlap-free in the sense that it does not contain a factor of the form $awawa$ for any $a \in \{0,1\}$ and $w \in \{0,1\}^*$ which was used by Thue to produce a square free infinite sequence on three letters, i.e. one not having any ww as factor. Such are the roots of the field of combinatorics on words [Lot83, Lot02]. The existence of square-free sequences has some other notable consequences, such as existence of infinite parties of chess, etc.[AS99].

Much the same way as uniform morphisms are related to standard base numeration systems, fixed points of non-uniform morphisms are naturally presented in generalised numeration systems. We have already mentioned generalised numeration systems in Section 4.3.3 on Presburger arithmetic, where we also conjectured that every a.p. of $\mathcal{N}$ is equivalent to a generalised numeration system. In Example 4.3.5 we presented the Fibonacci numeration system as a prominent example. Let us now consider an associated morphic word, which is generated by an appropriately non-uniform morphism.

Example 5.1.3. Let $\phi : a \mapsto ab, b \mapsto a$. Its fixed point $\phi^\omega(a)$ is the *Fibonacci word* $f = abaababaabaababaababa\ldots$, so called for the recursive dependence $\phi^{n+2}(a) = \phi^{n+1}(a) \cdot \phi^n(a)$ implying that $|\phi^n(a)|$ is the n^{th} Fibonacci number.

Recently Rigo has introduced *abstract numeration systems* [Rig01] as a further generalisation in which an arbitrary regular language is taken as a set of numerals, however, the ordering still represented length-lexicographically.

It is not hard to see, that an ω-word is length-lexicographically presentable iff it is morphic. There is a perfectly natural correspondence between the morphisms generating a word and the automaton recognising the set of "numerals", which, when length-lexicographically ordered, allow an automatic presentation of the morphic word. For the sake of completeness and to illustrate the techniques of Section 5.4 in this simple case we present a compact proof of this fact, which has appeared in [RM02].

Automata and morphisms

To each morphism $\varphi \in \mathsf{Hom}(\Sigma^*, \Sigma^*)$ with $|\varphi| = l$ we associate its *index transition system* $\mathcal{I}_\varphi = (\Sigma, [l], \delta)$ where $\delta(a,i) = \varphi(a)[i]$ for every $i < |\varphi(a)|$ and undefined otherwise. For each $a \in \Sigma$ considered as the initial state, the DFA $(\mathcal{I}_\varphi, a, \Sigma)$ accepts the set $I(a) = I_\varphi(a)$ of valid sequences of indices starting from a. Applying φ n

times to a gives the word

$$\varphi^n(a) = \prod_{x \in I(a) \cap [l]^n}^{lex} \delta^*(a,x) \tag{5.1}$$

where x is meant to run through all valid sequences of indices of length n in lexicographic order. Thus $\varphi^n(a)$ is the sequence of labels of the n^{th} level of the tree unfolding of $\mathcal{I}_\varphi$ from a.

Conversely, given a linear ordering $a_0 < a_1 < \ldots < a_s$ of Σ we associate to each DTS $\mathcal{T} = (Q, \Sigma, \delta)$ its *transition morphism* $\tau = \tau_{\mathcal{T}} \in \mathsf{Hom}(Q^*, Q^*)$ defined as $\tau(q) = \delta(q, a_{i_1})\delta(q, a_{i_2}) \ldots \delta(q, a_{i_k})$ where $a_{i_1} < a_{i_2} < \ldots < a_{i_k}$ are precisely those symbols for which a transition from q is defined. Just as in (5.1) applying τ n times to some q results in $\tau^n(q) = \prod_{w \in L(\mathcal{T},q,Q) \cap \Sigma^n}^{lex} \delta^*(q,w)$, where w runs through, in lexicographic order, all words of length n, which are labels of some path in $\mathcal{T}$ starting from q. Thus $\tau_{\mathcal{T}}^n(q)$ is the sequence of labels of the n^{th} level of the tree unfolding of $\mathcal{T}$ from q.

Proposition 5.1.4 ([RM02])**.** The word structure $\mathfrak{W}_w$ of an ω-word w is length-lexicographically presentable iff w is morphic. The transformation from one formalism to the other is straightforward.

Proof. Let $\tau \in \mathsf{Hom}(\Gamma^*, \Gamma^*)$ be prolongable on a and consider its index transition system $\mathcal{I} = \mathcal{I}_\tau$. It is clear from our previous observations that the language $L(\mathcal{I}, a, \Gamma)$ recognised by $\mathcal{I}$ with all states final and a as its initial state provides, equipped with the prefix-ordering, an automatic presentation of the tree unfolding $\mathcal{T} = \mathcal{T}_{\mathcal{I},a}$ of $\mathcal{I}$ from the initial state a. As also remarked, $\tau^n(a)$ is precisely the word one obtains by reading the n^{th} level of $\mathcal{T}$ from "left to right", i.e. in lexicographic order. Also note that τ being prolongable on a, $\mathcal{I}_\tau$ contains a transition $a \xmapsto{0} a$, therefore the subtree of $\mathcal{T}$ rooted at 0 is isomorphic to the whole tree. Let $\tau(a) = au$ for some $u = u_1 \ldots u_t \in \Gamma^*$ and let $\mathcal{U}_i$ be the subtree rooted at $0 < i \leq t$. Then $\tau^{n+1}(a) = au\tau(u) \cdots \tau^n(u) = \tau^n(a) \cdot \tau^n(u)$ and $\mathcal{T} \cong a(\mathcal{T}, \mathcal{U}_1, \ldots, \mathcal{U}_t)$. To obtain a length-lexicographic presentation of $\tau^\omega(a)$ we dispense with the subtree rooted at 0 so that the levels of the remaining regular tree $a(\mathcal{U}_1, \ldots, \mathcal{U}_t)$ correspond to the increments $\tau^n(u)$ between iterations of τ. We have thus shown that $D = L(\mathcal{I}_\tau, a, \Gamma) \setminus 0[|\tau|]^*$ and $P_c = L(\mathcal{I}_\tau, a, c) \setminus 0[|\tau|]^*$ for each $c \in \Gamma$ together with the natural length-lexicographic ordering provide an automatic presentation of $\tau^\omega(a)$. To give a lexicographic presentation of $w = \sigma(\tau^\omega(a))$ where $\sigma \in \mathsf{Hom}(\Gamma^*, \Sigma^*)$ we set $D' = \{xi \mid c \in \Gamma, x \in P_c, i < |\sigma(c)|\}$ and $P_b = \{xi \mid c \in \Gamma, x \in P_c, \sigma(c)[i] = b\}$ for each $b \in \Sigma$.

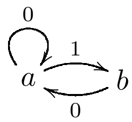
Conversely, given a lexicographic presentation $(\mathcal{A}_D, <_{\text{lex}}, \{\mathcal{A}_{P_a}\}_{a \in \Sigma})$ of some w consider the product automaton $\mathcal{A} = \prod_{a \in \Sigma} \mathcal{A}_{P_a}$. Let $\tau = \tau_{\mathcal{A}}$ be its transition morphism, and let us define $\sigma \in \mathsf{Hom}(Q(\mathcal{A})^*, \Sigma^*)$ by stipulating that $\sigma(\vec{q}) = a$

whenever the a^{th} component of $\vec{q}$ is an accepting state of $\mathcal{A}_{P_a}$ (clearly, a is then uniquely determined) and $\sigma(\vec{q}) = \varepsilon$ when no such a exists. To ensure that τ is prolongable, we introduce a new symbol $\vec{q}_0' \notin Q(\mathcal{A})$ and set $\tau(\vec{q}_0') = \vec{q}_0'\tau(\vec{q}_0)$ and $\sigma(\vec{q}_0') = \sigma(\vec{q}_0)$, where $\vec{q}_0$ is the initial state of $\mathcal{A}$. We leave it to the reader to check that $w = \sigma(\tau^\omega(\vec{q}_0'))$.

□

Example 5.1.5. Recall the Fibonacci word generated by the morphism $\phi : a \mapsto ab, b \mapsto a$ of Example 5.1.3. The index transition system of ϕ,

$\mathcal{I}$:



accepts, with a being initial and both states being final, the language $\{0,1\}^* \setminus \{0,1\}^*11\{0,1\}^*$ of Fibonacci numerals from Example 4.3.5 only *with* leading zeros. The construction of the proof of Proposition 5.1.4 dispenses precisely with those numerals starting with a zero, thus producing an injective presentation.

5.2 MSO-friendly presentations

Let us now turn our attention to more sophisticated automatic orderings of type ω. How does the choice of the ordering affects the class and the properties of words thus presentable? Note that in order to give a complete answer we need to characterise *all* automatic presentations of $(\mathbb{N}, <)$ up to equivalence in some manageable way. Short of achieving this ambitious task, we will introduce a generalisation of the length-lexicographic ordering and a corresponding notion of higher-order morphic words. We shall see that increasing the complexity of the ordering relation this way gives rise to a hierarchy of higher-order morphic words enjoying all the nice properties of morphic words.

In preparation we define the key concept of MSO-friendly presentations and derive extensions of the Fundamental Theorem 3.1.3 to MSO over word structures having an MSO-friendly presentation.

Definition 5.2.1 (MSO-friendly presentations).
An automatic presentation $\mathfrak{d} = (D, <, \{P_a\}_{a\in\Sigma})$ of some infinite word $w \in \Sigma^\omega$ is *MSO-friendly* if there is an algorithm, which constructs for every homomorphism $\psi \in \mathsf{Hom}(\Sigma^*, M)$ into a finite monoid M and for every monoid element $m \in M$ a synchronous two-tape automaton recognising the relation

$$B_m = \{(x, y) \in D^2 \mid x < y \wedge \psi(w[x, y]) = m\} \ .$$

Thus, being MSO-friendly means that membership of finite factors of w in a regular language can be decided by an effectively constructible automaton reading the representations of the two endpoints of the factor. It is very easy to derive decidability of the monadic second-order theory of words having MSO-friendly presentations.

Lemma 5.2.2. Let $\mathfrak{d} = (D, <, \{P_a\}_{a\in\Sigma})$ and ν constitute an MSO-friendly presentation of $w \in \Sigma^\omega$. Then for every deterministic Muller automaton $\mathcal{A}$ an automaton recognising the following set can be effectively constructed.

$$E_{\mathcal{A}} = \{x \in D \mid w[x, \infty) \in L(\mathcal{A})\}$$

Proof. Consider $\mathcal{A}$ as a pair (ψ, M) with $M = (Q \to Q, \circ)$ and $\psi \in \mathsf{Hom}(\Sigma^*, M)$. By MSO-friendliness of $\mathfrak{d}$ we find automata recognising $X_q = \{(x, y) \in D^2 \mid x < y \wedge \psi(w[x, y])(q_0) = q\}$ for each $q \in Q$. Using Theorem 3.1.3 we can construct automata recognising $Y_F = \{x \in D \mid \bigwedge_{q\in F} \exists^\infty y\, X_q(x, y) \wedge \bigwedge_{q\notin F} \neg\exists^\infty y\, X_q(x, y)\}$ for all $F \subseteq Q$. Finally, $E_{\mathcal{A}}$ is the union of those Y_F such that a run of $\mathcal{A}$ is accepting with F being the set of infinitely often occurring states. The claim follows. $\square$

Corollary 5.2.3. *Let w be an ω-word having an MSO-friendly automatic presentation. Then the* MSO*-theory of $\mathfrak{W}_w$ is decidable.*

Proof. In line with the well known correspondence between automata and MSO on ω-words deciding the MSO-theory of a word structure amounts to deciding acceptance of the word by any given deterministic Muller automaton $\mathcal{A}$. Given an MSO-friendly presentation this can be done by checking membership of a representation of 0 in $E_{\mathcal{A}}$ constructed as in the above lemma. $\square$

MSO-friendliness yields more than just decidability as we shall see next. Let φ be an MSO sentence in a language of word structures and let x, y be first-order variables not occurring in any subformula of φ. We define three kinds of *relativisations of* φ: $\varphi^{[0,x]}$, $\varphi^{[x,y]}$, and $\varphi^{[x,\infty)}$ obtained by relativising all first- and second-order quantifications to the noted intervals. For instance $(\exists z\vartheta)^{[x,y]} = \exists z(x \leq z \wedge z \leq y \wedge \vartheta^{[x,y]})$, and $(\forall Z\vartheta)^{[x,\infty)} = \forall Z(\forall z(z \in Z \to x \leq z) \to \vartheta^{[x,\infty)})$. The relevance of relativisation is expressed by the equivalence $\mathfrak{W}_w \models \varphi^I \iff \mathfrak{W}_{wI} \models \varphi$, where I is an interval of any of the three kinds.

Lemma 5.2.4 (Normal Form of MSO formulae over word structures)**.** Every MSO formula $\varphi(\vec{x})$ having free first-order variables $x_0, \ldots, x_{n-1}$ and no free second-order variables is equivalent to a boolean combination of formulae $x_i < x_j$ and relativised MSO sentences $\vartheta^{[0,x_i]}$, $\vartheta^{[x_i,x_j]}$, and $\vartheta^{[x_i,\infty)}$ with $i, j \in [n]$.

Proof. We present a proof through automata. Via standard construction, there is a deterministic Muller automaton $\mathcal{A}$ over the alphabet $\Sigma \times \{0, 1\}^n$ such that $\mathfrak{W}_w \models \varphi(\vec{k})$ iff $w \otimes \xi_{\vec{k}} \in L(\mathcal{A})$ for all $\vec{k} \in \mathbb{N}^n$, where $\xi_{\vec{k}} \in (\{0, 1\}^n)^\omega$ is the characteristic word of the tuple $\vec{k}$, i.e. $\xi_{\vec{k}}[i]_j = 1$ iff $k_j = i$. We collect for each pair of states (p, q) of $\mathcal{A}$ the regular language $L_{p,q} = \{u \in \Sigma^* \mid \delta^*(p, u \otimes (0^n)^{|u|}) = q\}$. Additionally, we let $L_q = \{u \in \Sigma^\omega \mid \mathcal{A}$ accepts $u \otimes (0^n)^\omega$ from state $q\}$. Again, by standard constructions, we find MSO sentences $\vartheta_{p,q}$ respectively ϑ_q defining these languages.

Each infinite word $w\otimes\xi_{\vec{k}}$ is naturally factored into segments in between consecutive k_i's, some of which can be equal. Accordingly, each run of $\mathcal{A}$ can be factored into finite number of finite segments and an infinite segment by those positions where in at least one of the last n components of the symbol read a 1 is encountered. The intermediate segments and the last infinite segment are models of the appropriate sentences $\vartheta_{p,q}$ and of ϑ_q respectively.

By summing up all possible factorisations of accepting runs we obtain in a first attempt a boolean combination of formulae of type $x_i < x_j$, $x_i = x_j$, $P_a x_i$ and of relativised sentences of the form $\vartheta_{q_0,q}^{[0,x_i)}$, $\vartheta_{p,q}^{(x_i,x_j)}$ and $\vartheta_q^{(x_i,\infty)}$. Equality can be expressed using $<$, and integrating the $P_a x_i$ into the neighbouring openly relativised segment formulae we finally arrive at a normal form as promised. □

Theorem 5.2.5 (MSO-definability)**.** *Let w be an ω-word having an MSO-friendly presentation $\mathfrak{d}$ with domain D and bijective naming function $\nu : D \to \mathbb{N}$. Then there is an algorithm transforming every* MSO *formula $\varphi(\vec{x})$ having n free first-order variables (and no free set variables) into an n-tape synchronous automaton $\mathcal{A}$ such that for every $u_1, \ldots, u_n \in D$*

$$\mathfrak{W}_w \models \varphi[\nu(\vec{u})] \iff \vec{u} \in L(\mathcal{A})$$

Proof. Using Lemma 5.2.4, we transform φ into a boolean combination of relativised sentences and comparison formulae $x_i < x_j$. MSO-friendliness and Lemma 5.2.2 yield automata recognising the relations defined by relativised sentences $\vartheta^{[0,x_i]}$, $\vartheta^{[x_i,x_j]}$, respectively $\vartheta^{[x_i,\infty)}$. Thus, by the appropriate combination of the automaton recognising $<$ and of the automata recognising the relativised subformulae of the normal form we obtain $\mathcal{A}$ as required. □

Note that a set $X \subseteq \mathbb{N}$ is definable by an MSO formula $\psi(X)$ in $\mathfrak{W}_w$ iff it is pointwise definable by one of the form $\varphi(x)$. Thus, $(\mathfrak{W}_w, X)$ is automatic for every $\mathfrak{W}_w$ presentable in an MSO-friendly way and for every X, MSO-definable in $\mathfrak{W}_w$.

5.3 k-lexicographic presentations

Let Σ be a finite non-empty alphabet. To each word $u = a_0 a_1 \ldots a_{n-1} \in \Sigma^*$ of length n and to each $0 < k$ we associate its *k-split* $(u^{(1)}, u^{(2)}, \ldots, u^{(k)})$ defined as follows. Let t be such that $tk \leq n < (t+1)k$. Then the ith word of the k-split is $u^{(i+1)} = a_i a_{k+i} a_{2k+i} \ldots a_{tk+i}$ for each $i < k$. Conversely, the *k-merge* of the components produces the original word $u = \otimes_k(u^{(1)}, \ldots, u^{(k)})$. Additionally, we define $u^{(0)} = |u| \in \mathbb{N}$ or in unary presentation as $1^{|u|}$, whichever is more convenient. For $0 \leq i < k$ we define the equivalence

$$u =_i v \stackrel{\text{def}}{\iff} \forall j \leq i \;\; u^{(j)} = v^{(j)} \; .$$

This implies, in particular, $|u| = |v|$. Let now $<$ be a linear ordering of Σ, and let $<_{\text{lex}}$ denote the induced lexicographic ordering. For each $0 \leq k$ we define the k-length-lexicographic ordering ($<_{k\text{-llex}}$) of Σ^* as

$$u <_{k\text{-llex}} v \stackrel{\text{def}}{\iff} |u| < |v| \lor \exists i < k : \; u =_i v \land u^{(i+1)} <_{\text{lex}} v^{(i+1)} \; .$$

Definition 5.3.1 (k-lexicographic words)**.** An ω-word $w \in \Sigma^\omega$ is *k-lexicographic* (short: *k-lex*) if there is an automatic presentation $(D, <_{k\text{-llex}}, \{P_a\}_{a\in\Sigma})$ of the associated word structure $\mathfrak{W}_w$. For each k, the class of k-lexicographic words is denoted $\mathcal{W}_k$, and we also let $\mathcal{W} = \bigcup_k \mathcal{W}_k$.

Observe that the 0-lexicographic ordering is just the ordering of words according to their length. Therefore, by definition, the domain of a 0-lex presentation has to be *thin*, i.e. containing at most one word of each length. All such presentations are easily seen to be equivalent to one over a unary alphabet. Thus, $\mathcal{W}_0$ is precisely the class of ultimately periodic words. Together with Proposition 5.1.4 we have a characterisation of the

Proposition 5.3.2. $\mathcal{W}_0$ is the class of ultimately periodic words.

We have already seen that 1-lex words are precisely the morphic ones. Let us now give an example of a 2-lexicographic word, which is not morphic.

Example 5.3.3. Consider the Champernowne word $s = 12345678910111213\ldots$ (also called Smarandache sequence) obtained by concatenating all decimal numerals (without leading zeros) in ascending, i.e. length-lexicographic order. To give a natural 2-lex presentation of $\mathfrak{W}_s$ we use words $\otimes_2(x^{(1)}, x^{(2)})$ such that $x^{(1)}$ is a decimal numeral (not starting with a zero) and $x^{(2)} \in 1^*01^*$. We use the single 0 in $x^{(2)}$ to mark a position within $x^{(1)}$. For each digit $d \in [10]$ we can thus define the unary predicate P_d as $([10]1)^*d0([10]1)^* \setminus 0[10]^*$.

We close this section with two simple but useful observations.

Proposition 5.3.4 (Closure under homomorphic mappings)**.** The class of automatically presentable ω-words is closed under homomorphic mappings. In particular, if w is k-lexicographic, then so is $h(w)$ for every homomorphism h.

Proof. The idea is to append each word $x \in P_a$ of a given presentation of w indexing a symbol a by $|h(a)|$ many appropriately chosen suffixes $u_{a,i}$ with $i < |h(a)|$. For k-lexicographic presentations we choose $|u_{a,i}| = k$ and take care that differences fall within the k^{th} component of the k-split of $xu_{a,i}$. ☐

Lemma 5.3.5 (Normal Form Lemma)**.** Let $1 < k \in \mathbb{N}$. Each k-lexicographic presentation $\mathfrak{d} = (D, <_{k\text{-llex}})$ of $(\mathbb{N}, <)$ over an alphabet Σ is equivalent to one $\mathfrak{d}' = (D', <_{k\text{-llex}})$ over some Γ such that $D' \subseteq (\Gamma^k)^*$. In fact, one can choose $\Gamma = \{0, 1\}$ in the above.

Proof. Let first $\Gamma = \Sigma \uplus \{\widehat{0}, \ldots, \widehat{k-1}, \diamond\}$ endowed with the ordering $\diamond < \widehat{0} < \ldots < \widehat{k-1} < a_1 < \ldots < a_s$ where $a_1 < \ldots < a_s$ is the ordering of Σ used in the presentation $\mathfrak{d}$. We define the translation $t : \Sigma^* \to (\Gamma^k)^*$ padding each word x to

$t(x) = \widehat{l}\diamond^{k-1} x\diamond^{k-l}$ where $l = |x| \bmod k$. Observe that the moduli of the positions of symbols of x are preserved in the process of this coding, i.e. $t(x)^{(i)} = \alpha x^{(i)}\diamond$ with α being $\widehat{l}$ for $i = 0$ and $\diamond$ otherwise. Consequently $x <_{k\text{-llex}} y$ iff $t(x) <_{k\text{-llex}} t(y)$ in the orderings induced by that of the symbols. Since t is a synchronised rational bijection $\mathfrak{d}' = (t(D), <_{k\text{-llex}})$ is equivalent to $\mathfrak{d}$.

Finally, to obtain an equivalent presentation over $\{0,1\}$ take any binary coding $a \mapsto b_0 \dots b_{l-1}$ of the symbols $a \in \Gamma$ uniformly of length l and such that $a < a'$ iff $b_0 \dots b_{l-1} <_{1\text{-llex}} b'_0 \dots b'_{l-1}$. Extend this into a coding of blocks of k consecutive symbols as $a^0 \dots a^{k-1} \mapsto b_0^0 \dots b_0^{k-1} \dots b_{l-1}^0 \dots b_{l-1}^{k-1}$, and extend this homomorphically to $(\Gamma^k)^*$. Due to the uniformity requirement, this translation is semi-synchronous, further it respects the k-lexicographic ordering, thus providing an equivalent k-lexicographic presentation.

$\square$

5.4 MSO-friendliness, Closure and Decidability

5.4.1 Technical tools: automata transformations

Consider a finite deterministic transition system $\mathcal{T} = (Q, \Sigma, \delta)$ and the associated pair (M, φ) consisting of the finite monoid $M = (Q \to Q, \circ)$ and the homomorphism $\varphi \in \mathsf{Hom}(\Sigma^*, M)$ induced by δ. We call $\mathsf{Hom}(\Sigma^*, M)$ the *derived state space* and denote it by $Q^{(\Sigma)}$. Furthermore, we call $M^{(\Sigma)} = Q^{(\Sigma)} \to Q^{(\Sigma)}$ the monoid of *automata transformations*. Note that both $Q^{(\Sigma)}$ and $M^{(\Sigma)}$ are finite. This terminology is justified by the fact that $Q^{(\Sigma)} = \mathsf{Hom}(\Sigma^*, M)$ is in essence the set of all Σ-labelled DTS's over the state space Q, hence $M^{(\Sigma)}$ is indeed the monoid of all transformations of such transition systems.

A particular submonoid of $M^{(\Sigma)}$ that interests us is that of *inverse homomorphic transformations* $H^{(\Sigma)}$ defined as follows. Consider a homomorphism $h \in \mathsf{Hom}(\Sigma^*, \Sigma^*)$. We can associate to h the element $\Phi(h)$ of $M^{(\Sigma)}$ defined as $(Q^{(\Sigma)} \ni \chi \mapsto \chi \circ h)$. It can be readily seen that Φ is a monoid homomorphism from $\mathsf{Hom}(\Sigma^*, \Sigma^*)$ to $M^{(\Sigma)}$, therefore $H^{(\Sigma)} \stackrel{\text{def}}{=} \Phi(\mathsf{Hom}(\Sigma^*, \Sigma^*))$ is a submonoid of $M^{(\Sigma)}$. In terms of automata transformations this amounts to mapping a transition function δ to δ' such that $\delta'(q, a) = q'$ whenever $\delta^*(q, h(a))$, where δ^* denotes as usual the extension of δ to all words over Σ. We let $h^{-1}(\mathcal{T})$ denote the transition system (Q, Σ, δ'). Thus, for every $q, q' \in Q$ and $w \in \Sigma^*$ there is a path in $h^{-1}(\mathcal{T})$ labelled w from q to q' iff there is a path in $\mathcal{T}$ labelled $h(w)$ from q to q'.

Consider a finite alphabet Θ and a mapping $\vartheta : \Theta \to \mathsf{Hom}(\Sigma^*, \Sigma^*)$. We extend ϑ to Θ^* according to the rule

$$\vartheta(x \cdot x') = \vartheta(x') \circ \vartheta(x) \tag{5.2}$$

which ensures that $\Phi_\vartheta = \Phi \circ \vartheta$ is a homomorphism from Θ^* to $H^{(\Sigma)}$.

$$\begin{aligned}\Phi(\vartheta(x \cdot x'))(\chi) &= \Phi(\vartheta(x') \circ \vartheta(x))(\chi) = \chi \circ \vartheta(x') \circ \vartheta(x)\\ &= \Phi(\vartheta(x'))(\chi) \circ \vartheta(x) = \Phi(\vartheta(x))(\Phi(\vartheta(x'))(\chi))\\ &= (\Phi(\vartheta(x)) \circ \Phi(\vartheta(x')))(\chi) \quad .\end{aligned}$$

Therefore, the pair $(H^{(\Sigma)}, \Phi_\vartheta)$ represents, in accordance with our initial correspondence, a Θ-labelled finite transition system with state space $Q^{(\Sigma)}$. Elements of Θ^* can thus be seen as words over Θ, or, via ϑ as homomorphisms from Σ^* to Σ^*, or, via Φ_ϑ, as transformations of Σ-labelled transition systems. Given a word $w \in \Sigma^*$ and a monoid element $m \in M$, we are interested in the following subset of Θ^*.

$$L_{\mathcal{T},w,m,\vartheta} = \{x \in \Theta^* \mid \text{the state transformation induced by } w \text{ in } \vartheta(x)^{-1}(\mathcal{T}) \text{ is } m\}$$

Let $n = |Q|$. Since $Q^{(\Sigma)}$ is finite, and the kernel $\mathsf{Ker}(\varphi)$ of every homomorphism $\varphi \in Q^{(\Sigma)}$ is a congruence (wrt. concatenation) of finite index, their intersection

$$\sim_n \stackrel{\text{def}}{=} \bigcap_{\varphi \in Q^{(\Sigma)}} \mathsf{Ker}(\varphi) = \{(u, u') \mid \forall \varphi \in \mathsf{Hom}(\Sigma^*, M)\ \varphi(u) = \varphi(u')\}$$

is again a congruence of finite index, i.e. the factor monoid $\widetilde{Q} = \Sigma^* / \sim_n$ is finite. Note that this equivalence depends only on the size of Q, hence the notation. Intuitively, $u \sim_n u'$ iff there is no automaton having at most n states that could distinguish u from u'. This equivalence can be used to define the Hall metric on Σ^* giving rise to a compact Hausdorff topology (cf. [PS05]), which is essentially what one obtains from the analogous equivalences wrt. MSO formulae of restricted quantifier ranks (cf. [Rab05],[RT06]).

Clearly, every homomorphism $h \in \mathsf{Hom}(\Sigma^*, \Sigma^*)$ preserves $\sim_n$-classes, and thus determines a function $\widetilde{h} : \widetilde{Q} \to \widetilde{Q}$. It is again routine to check that $\Psi : h \mapsto \widetilde{h}$ thus defined is a homomorphism from $(\mathsf{Hom}(\Sigma^*, \Sigma^*), \circ)$ into $\widetilde{M} = (\widetilde{Q} \to \widetilde{Q}, \circ)$. Furthermore, each $\sim_n$ determines an equivalence of homomorphisms $h, h' \in \mathsf{Hom}(\Sigma^*, \Sigma^*)$ defined as follows.

$$\begin{aligned}h \sim_n h' &\stackrel{\text{def}}{\iff} \forall w \in \Sigma^* \quad h(w) \sim_n h'(w)\\ &\iff \forall a \in \Sigma \quad h(a) \sim_n h'(a)\end{aligned} \tag{5.3}$$

The Hall metric on Σ^* thus induces a similar metric, thereby determining a compact Hausdorff topology, on $\mathsf{Hom}(\Sigma^*, \Sigma^*)$. Moreover, the following equivalence

$$\Phi(h_1) = \Phi(h_2) \iff h_1 \sim_n h_2 \iff \widetilde{h_1} = \widetilde{h_2} \tag{5.4}$$

can easily be checked to hold:

$$\begin{aligned} & \Phi(h_1) = \Phi(h_2)\\ \iff & \forall \chi \in \mathsf{Hom}(\Sigma^*, M) :\ \chi \circ h_1 = \chi \circ h_2\\ \iff & \forall w \in \Sigma^* \, \forall \chi \in \mathsf{Hom}(\Sigma^*, M) :\ \chi(h_1(w)) = \chi(h_2(w))\\ \iff & \forall w \in \Sigma^* :\ h_1(w) \sim_n h_2(w)\\ \iff & h_1 \sim_n h_2 \quad .\end{aligned}$$

Lemma 5.4.1 (Higher-Order Regularity (HOR) Lemma).
For every $\mathcal{T} = (Q, \Sigma, \delta)$ with associated (M, φ) and for every $w \in \Sigma^*$, $m \in M$, and every Θ and ϑ as above we can construct an automaton recognising $L_{\mathcal{T},w,m,\vartheta}$.

Proof. Observe that we can write $L_{\mathcal{T},w,m,\vartheta}$ equivalently as

$$\begin{aligned} L_{\mathcal{T},w,m,\vartheta} &\stackrel{\text{def}}{=} \{x \in \Theta^* \mid \text{the state transformation induced by } w \text{ in } \vartheta(x)^{-1}(\mathcal{T}) \text{ is } m\} \\ &= \{x \in \Theta^* \mid \text{the state transformation induced by } \vartheta(x)(w) \text{ in } \mathcal{T} \text{ is } m\} \\ &= \{x \in \Theta^* \mid \varphi(\vartheta(x)(w)) = m\} \\ &= \{x \in \Theta^* \mid \Phi(\vartheta(x))(\varphi)(w) = m\} \\ &= \{x \in \Theta^* \mid \Phi(\vartheta(x)) \in H_{m,\varphi,w}\} \\ &= \Phi_\vartheta^{-1}(H_{m,\varphi,w}) \end{aligned}$$

where $H_{m,\varphi,w} = \{\xi = \Phi(h) \in H^{(\Sigma)} \mid \xi(\varphi)(w) = \varphi(h(w)) = m\}$. Hence, $L_{\mathcal{T},w,m,\vartheta}$ is recognised by the subset $H_{m,\varphi,w}$ of the finite monoid $H^{(\Sigma)}$ under the morphism Φ_ϑ. Moreover, $H_{m,\varphi,w}$ can be determined, according to (5.4), by enumerating all $\sim_n$-classes of homomorphisms. Using the correspondence (5.3) this can be reduced to enumerating $\sim_n$-classes of words over Σ. $\square$

5.4.2 k-lexicographic presentations are MSO-friendly

Let a $(k+1)$-lex presentation $\mathfrak{d} = (D, <_{(k+1)\text{-llex}}, \{\mathcal{A}_a\}_{a\in\Sigma})$ of $w \in \Sigma^\omega$ in normal form over the alphabet Γ together with the bijective coordinate function $\nu : D \to \mathbb{N}$ as well as a homomorphism $\psi \in \mathsf{Hom}(\Sigma^*, M)$ into a finite monoid M be given. We associate to $\mathfrak{d}$ the DFA $\mathcal{A}_\mathfrak{d} = \overline{\prod_{a\in\Sigma} \mathcal{A}_a}$ consisting of the DTS $\mathcal{T}_\mathfrak{d} = (Q_\mathfrak{d}, \Gamma, \delta_\mathfrak{d})$ and having initial state $\vec{q}_0$. Further let $\sigma_\mathfrak{d} \in \mathsf{Hom}(Q_\mathfrak{d}^*, \Sigma^*)$ be such that $\sigma_\mathfrak{d}(\vec{q}) = a$ whenever the a^{th} component of $\vec{q}$ is in a final state (in which case a is uniquely determined) and $\sigma_\mathfrak{d}(\vec{q}) = \varepsilon$ otherwise. Finally, we set $w_\mathfrak{d} = \prod_{x\in\Gamma^*}^{<_{(k+1)\text{-llex}}} \delta_\mathfrak{d}^*(\vec{q}_0, x) \in Q_\mathfrak{d}^\omega$. Clearly, $w = \sigma_\mathfrak{d}(w_\mathfrak{d})$

For every $x = \otimes_{k+1}(x^{(1)}, \ldots, x^{(k+1)})$ let $x' = \otimes_k(x^{(1)}, \ldots, x^{(k)})$ be the projection of x onto its first k splitting components, when $k > 0$ and let $x' = x^{(0)} = 1^{|x|}$ when $k = 0$. We define $D' = \{x' \mid x \in D\}$ as the point-wise projection of D. The equivalence $=_k$ partitions the set D of indices into consecutive intervals. Let $c(x')$ denote the interval containing x, i.e. $c(x') = \{y \in D \mid y' = x'\}$, and consider the factorisation of w according to such intervals.

$$w = \prod_{x'\in D'}^{<_{k\text{-llex}}} w[c(x')]$$

The *contraction* (compare with that of [ER66]) of w wrt. $\mathfrak{d}$ and ψ is the ω-word

$$c_\mathfrak{d}^\psi(w) = \prod_{x'\in D'}^{<_{k\text{-llex}}} \psi(w[c(x')]) \quad \in M^\omega$$

indexed by elements of D' ordered according to $<_{k\text{-llex}}$. We can prove that $c_\mathfrak{d}^\psi(w)$ is in fact automatically presentable over $(D', <_{k\text{-llex}})$.

Lemma 5.4.2 (Contraction Lemma).
Let $\mathfrak{d} = (D, <_{k+1\text{-llex}}, \{\mathcal{A}_a\}_{a\in\Sigma})$ be a $(k+1)$-lex presentation with coordinate function ν of the word structure of an ω-word $w \in \Sigma^\omega$. Then for every finite monoid M, every $\psi \in \mathsf{Hom}(\Sigma^*, M)$ and for each $m \in M$ the following relations are regular.

$$\begin{array}{rcl} B'_m & = & \{(x,y) \in D^2 \mid x \leq_{k+1\text{-llex}} y \wedge x =_k y \wedge \psi(w[x,y]) = m\} \\ P'_m & = & \{x' \in D' \mid \psi(w[c(x')]) = m\} \end{array}$$

Whence, $(D', <_{k\text{-llex}}, \{P'_m\}_{m\in M})$ is a k-lexicographic presentation of $c^{\psi}_{\mathfrak{d}}(w)$.

Proof. We are going to employ the machinery introduced in Section 5.4.1. In order to apply the HOR Lemma first we generalise the notion of transition morphisms. Wlog. the ordered alphabet Γ of the presentation $\mathfrak{d}$ is $[t] = 0 < 1 < \ldots < t-1$. Let $Q = Q_{\mathfrak{d}} \times \{l, r, b, n\}$ (standing for left, right, both and none respectively) and $\pi : ((q,x) \mapsto q)$ be the projection onto the first component. We define the mapping $\beta : ([t]^k([t] \times [t]))^* \to \mathsf{Hom}(Q^*, Q^*)$ via homomorphic extension as in (5.2) while stipulating that

$$\begin{array}{rcl} \beta_{u(i,j)}(q,n) & = & (\delta^*_{\mathfrak{d}}(q,u0),n)(\delta^*_{\mathfrak{d}}(q,u1),n)\ldots(\delta^*_{\mathfrak{d}}(q,u(t-1)),n) \\ \beta_{u(i,j)}(q,l) & = & (\delta^*_{\mathfrak{d}}(q,ui),l)(\delta^*_{\mathfrak{d}}(q,u(i+1)),n)\ldots(\delta^*_{\mathfrak{d}}(q,u(t-1)),n) \\ \beta_{u(i,j)}(q,r) & = & (\delta^*_{\mathfrak{d}}(q,u0),n)\ldots(\delta^*_{\mathfrak{d}}(q,u(j-1)),n)(\delta^*_{\mathfrak{d}}(q,uj),r) \\ \beta_{u(i,j)}(q,b) & = & \begin{cases} \varepsilon \quad \text{for } i > j \\ (\delta^*_{\mathfrak{d}}(q,ui),b) \quad \text{for } i = j \\ (\delta^*_{\mathfrak{d}}(q,ui),l)(\delta^*_{\mathfrak{d}}(q,u(i+1)),n)\ldots(\delta^*_{\mathfrak{d}}(q,u(j-1)),n)(\delta^*_{\mathfrak{d}}(q,uj),r) \end{cases} \end{array}$$

where u ranges over Γ^k and $i,j < t$. We regard β as a mapping from pairs of $=_k$-equivalent words $x,y \in D$. Indeed, each pair (x,y) of words with $x' = y'$ determines a sequence $u_1(i_1,j_1)\ldots u_n(i_n,j_n)$, and vice versa, such that $x^{(k+1)} = i_1\ldots i_n$, $y^{(k+1)} = j_1\ldots j_n$ and $x' = y' = u_1\ldots u_n$. In accordance with (5.2) we can thus define $\beta_{x,y}$ as the composition $\beta_{u_n(i_n,j_n)} \circ \cdots \circ \beta_{u_1(i_1,j_1)}$. We further let $\tau_u = \beta_{u(0,t-1)}$. Note that, for $k = 0$, τ_ε is essentially the transition morphism τ associated to $\mathcal{T}_{\mathfrak{d}}$ as defined above. To allow for uniform treatment we set $\tau_{1^n} = {\tau_\varepsilon}^n$ when $k = 0$.

Claim *For all $k \in \mathbb{N}$ and $x, y \in (\Gamma^{k+1})^*$ such that $x' = y'$ and $x \leq_{k+1\text{-}llex} y$:*

$$\pi(\beta_{x,y}(\vec{q}, b)) \quad = \quad \textstyle\prod_{z=x}^{y} \delta^*_{\mathfrak{d}}(\vec{q}, z)$$

where the concatenation product is taken over the values of z in the $(k+1)$-lexicographic ordering. Consequently, when in addition $x, y \in D$ then we have

$$\begin{array}{rcl} \sigma_{\mathfrak{d}}(\pi(\beta_{x,y}(\vec{q}_0, b))) & = & w[x,y] \\ \sigma_{\mathfrak{d}}(\pi(\tau_{x'}(\vec{q}_0, b))) & = & w[c(x')] \end{array}$$

By the above claim we know that $\psi(w[x,y]) = \psi(\sigma_{\mathfrak{d}}(\pi(\beta_{x,y}(\vec{q}_0, b))))$ and that $\psi(w[c(x')]) = \psi(\sigma_{\mathfrak{d}}(\pi(\tau_{x'}(\vec{q}_0, b))))$. Recall that $\beta_{x,y}$ was defined as $\beta_{u_n(i_n,j_n)} \circ \cdots \circ$

$\beta_{u_1(i_1,j_1)}$ for all $x' = y' = u_1 \dots u_n$ with $u_i \in [t]^k$ and $x^{(k+1)} = i_1 \dots i_n$, $y^{(k+1)} = j_1 \dots j_n$. Similarly, $\tau_{x'} = \tau_{u_n} \circ \cdots \circ \tau_{u_1}$. The results are established by applying the HOR Lemma with $\varphi = \psi \circ \sigma_{\mathfrak{d}} \circ \pi$ and $\Theta = [t]^k([t] \times [t])$, $\vartheta_{x \otimes y} = \beta_{x,y}$ in the first case, respectively with $\Theta = [t]^k$, $\vartheta_{x'} = \tau_{x'}$ in the second case.

□

In particular, the contraction of a morphic word wrt. any given lexicographic presentation and any given morphism into a finite monoid is an ultimately periodic sequence. This is already sufficient to yield MSO decidability of morphic words, and is essentially the proof given in [CT02]. Obviously, by iterating this contraction process starting from any given k-lex presentation of an ω-word we arrive after (at most) k contractions, at an ultimately periodic sequence. It is now easy to use this fact to prove MSO decidability of k-lexicographic words. However, we aim for the stronger result of MSO-friendliness.

Theorem 5.4.3 (MSO-friendliness of k-lex presentations).
All k-lexicographic presentations are MSO-friendly.

Proof. The proof is by induction on k, the base case being clear. For the induction step, we consider a $k+1$-lex presentation. Observe that if two $k+1$-lex presentations of the same ω-word are equivalent, then one is MSO-friendly iff the other one is. Therefore, by the Normal Form Lemma 5.3.5, it is sufficient to provide a proof for $k+1$-lex presentations in normal form. So let $\mathfrak{d} = (D, <_{k+1\text{-llex}}, \{P_a\}_{a \in \Sigma})$ be a $k+1$-lex presentation in normal form of an ω-word $w \in \Sigma^\omega$. Let a morphism $\psi \in \mathsf{Hom}(\Sigma^*, M)$ into a finite monoid M be given. We need to construct automata deciding, given words $x, y \in D$ with $x \leq_{k+1\text{-llex}} y$, whether $\psi(w[x,y]) = m$. There are two cases. If $x' = y'$ then we simply verify $(x,y) \in B'_m$ as in the Contraction Lemma. When on the other hand $x' <_{k\text{-llex}} y'$ then we partition the interval $x \leq_{k+1\text{-llex}} z \leq_{k+1\text{-llex}} y$ into three segments according to whether $x' = z'$, $x' <_{k\text{-llex}} z' <_{k\text{-llex}} y'$ or $z' = y'$, i.e. consider the factors $w[x, \hat{x}]$, $w[\{z \in D \mid x' <_{k\text{-llex}} z' <_{k\text{-llex}} y'\}]$ and $w[\hat{y}, y]$, where $\hat{x}$ is the greatest element of $c(x')$ with respect to $<_{k+1\text{-llex}}$ and similarly $\hat{y}$ is the least element of $c(y')$. Note that both $\hat{x}$ and $\hat{y}$ are first-order definable, hence automaton computable from x, respectively from y. We can therefore compute $\psi(w[x, \hat{x}])$ as well as $\psi(w[\hat{y}, y])$ by an automaton simultaneously verifying B'_m for both pairs $(x, \hat{x})$ and $(\hat{y}, y)$ for all $m \in M$.

It remains to show that the value of the central segment is also automaton computable. By the Contraction Lemma we know that $\mathfrak{d}' = (D', <_{k\text{-llex}}, \{P'_m\}_{m \in M})$ is a k-lex presentation of $c^{\psi}_{\mathfrak{d}}(w)$. Thus, by the induction hypothesis, $\mathfrak{d}'$ is MSO-friendly. We use this fact to compute the value of the central segment. To this end, we employ the *multiplier morphism* $\mu_M \in \mathsf{Hom}(M^*, M)$ defined by stipulating that $\mu_M(m) = m$ for all $m \in M$. Let ν' denote the co-ordinate mapping associated to $\mathfrak{d}'$. By definition of a contraction $\psi(w[\nu(c(z'))]) = c^{\psi}_{\mathfrak{d}}(w)[\nu'(z')]$, therefore the value of the central segment $\psi(w[\{z \in D \mid x' <_{k\text{-llex}} z' <_{k\text{-llex}} y'\}])$ can be written as $\mu_M(c^{\psi}_{\mathfrak{d}}(w)(x', y'))$, which is by MSO-friendliness of $\mathfrak{d}'$ automaton computable. □

Corollary 5.4.4 (MSO decidability).
The MSO*-theory of the word structure* $\mathfrak{W}_w$ *associated to a* k*-lex word* $w \in \mathcal{W}$ *is decidable.*

MSO interpretations are usually understood to be one-dimensional. We use the notation $\leq^{\mathcal{I}}_{\mathrm{mdMSO}}$ to stress that $\mathcal{I}$ might be multi-dimensional. Further, we say that a tuple $(\varphi(x), \{\varphi_b(x)\}_{b\in\Gamma})$ of MSO formulae, together with the formula $\varphi_<(x, y) = x < y$, form a *restricted* MSO *interpretation* $\mathcal{I}$ (the restriction being that $\mathcal{I}$ can only redefine the colouring, but not $<$) of a finite or infinite word structure $\mathfrak{W}_{w'} \leq^{\mathcal{I}}_{\mathrm{rMSO}} \mathfrak{W}_w$. From Theorem 5.4.3 and Theorem 3.1.3 we conclude the next corollaries.

Corollary 5.4.5 (Closure under MSO-interpretations).
Let w *be a* k*-lexicographic word. For every structure* $\mathfrak{A}$ *and word* w' *we have*

1. $\mathfrak{A} \leq_{\mathrm{mdMSO}} \mathfrak{W}_w \Longrightarrow \mathfrak{A}$ *is automatic,*
2. $\mathfrak{W}_{w'} \leq_{\mathrm{rMSO}} \mathfrak{W}_w \Longrightarrow \mathfrak{W}_{w'}$ *is* k*-lexicographic.*

Corollary 5.4.6 (Closure under DGSM mappings).
For each $k \in \mathbb{N}$ *the class* $\mathcal{W}_k$ *is closed under deterministic generalised sequential mappings.*

Proof. Every deterministic sequential transduction $S(w)$ of a word w can be obtained by a homomorphic mapping of the run of $S \times \mathcal{B}^1_\Sigma$ over w, where $\mathcal{B}^1_\Sigma$ is the De Bruin transition system with memory of the single last symbol of Σ read. The homomorphism is just the output function of the sequential transducer S. The run of S on w is of course rMSO interpretable in $\mathfrak{W}_w$, is thus in $\mathcal{W}_k$, and closure under homomorphic mappings holds by Proposition 5.3.4. □

As an example of what can be interpreted in a word consider the following.

Theorem 5.4.7 (Automatic equivalence structures). *Consider* $\mathfrak{A} = (A, E)$ *with* E *an equivalence relation on a countably infinite set* A *having only finite equivalence classes. Assume further that for each* n *there are* $f(n) \in \mathbb{N}$ *many equivalence classes of size* n.
Then $\mathfrak{A} \in \textsc{AutStr}$ *if and only if there is a 2-lex word* $w = 0^{m_0}10^{m_1}10^{m_2}1\ldots$ *such that* $f(n) = |\{i \mid m_i = n\}|$*, in which case* $\mathfrak{A} \leq^{\mathcal{I}}_{\mathrm{FO}} \mathfrak{W}_w$ *for a fixed one-dimensional* FO*-interpretation* $\mathcal{I}$*, also implying that* $\mathrm{Th}_{\mathrm{MSO}}(\mathfrak{A})$ *is decidable.*

Proof. For the "if" direction, the interpretation in question is $\mathcal{I} = (\varphi_A(x), \varphi_E(x, y))$ with $\varphi_A(x) = P_0(x)$ and $\varphi_E(x, y) = \varphi_A(x) \wedge \varphi_A(y) \wedge \forall z(x < z < y \vee y < z < x \rightarrow P_0(z))$. It is now easy to check that $\mathcal{I}(\mathfrak{W}_w)$ is indeed isomorphic to $\mathfrak{A}$ and is thus, by Theorem 3.1.3 or by Corollary 5.4.5, automatic.

For the "only if" direction we construct, given an a.p. (L_A, L_E) of $\mathfrak{A}$, an a.p. of a binary word with the claimed property. First observe that since all equivalence classes of $\mathfrak{A}$ are finite, there is a constant C such that for all $x, y \in L_A$ with $(x, y) \in L_E$ $||x| - |y|| < C$. We can therefore easily construct by padding an equivalent presentation of $\mathfrak{A}$ in which $|x| = |y|$ holds for all x,y representing equivalent elements. We shall now assume this holds. Let Γ be the alphabet of the presentation of $\mathfrak{A}$. Wlog. $\Gamma = \{0, \dots, s-1\}$. The alphabet of the presentation of w will be $\Gamma' = \{0, \dots, s-1, s\}$ ordered naturally. We set $P_0 = \{\otimes_2(x, y) \mid (x, y) \in L_E \wedge \forall (x, z) \in L_E \; x \leq_{\text{lex}} z\}$, $P_1 = \{\otimes_2(x, s^{|x|}) \mid \forall (x, z) \in L_E \; x <_{\text{lex}} z\}$, and $D = P_0 \cup P_1$. It is now clear that $(D, <_{\text{2-llex}}, P_0, P_1)$ is an a.p. as promised.

□

5.5 Hierarchy Theorem

It is readily seen, that $\mathcal{W}_k$ is included in $\mathcal{W}_{k+1}$ for each k. Next we show that each $\mathcal{W}_k$ is properly included in the next one by exhibiting ω-words $s_{k+1} \in \mathcal{W}_{k+1} \setminus \mathcal{W}_k$. We call the s_k *stuttering words*. Each s_k is a word over the $(k+1)$-letter alphabet $\{a_0, \dots, a_k\}$ and is defined as the infinite concatenation product $s_k = \prod_{n=0}^{\infty} s_{k,n}$, where $s_{0,n} = a_0$ and $s_{k+1,n} = (s_{k,n})^{2^n} a_{k+1}$ for every k and n. That is

$$s_k = \prod_{n=0}^{\infty} (\cdots(((a_0^{2^n})a_1)^{2^n})\cdots)^{2^n} a_k \quad .$$

To give an illustration, we write for convenience $a, b, c, d \dots$ instead of $a_0, a_1, a_2, a_3 \dots$ for small k. The first few stuttering words are

$$\begin{array}{lcl} s_0 & = & a^\omega \\ s_1 & = & abaabaaaaba^8ba^{16}b\dots \\ s_2 & = & abcaabaabc(aaaab)^4c(a^8b)^8c\dots \\ s_3 & = & abcd(aabaabc)^2d((aaaab)^4c)^4d((a^8b)^8c)^8d\dots \\ \vdots & & \end{array}$$

We remark, that s_2 is not a fixed point of any iterated DGSM mapping [AG89].

Theorem 5.5.1 (Hierarchy Theorem)**.**
For each $k \in \mathbb{N}$ we have $s_{k+1} \in \mathcal{W}_{k+1} \setminus \mathcal{W}_k$.

Proof. We leave it to the reader to give a k-lex presentation of s_k for every k.
To show that $s_{k+1} \notin \mathcal{W}_k$ we argue indirectly as follows. Assume that there is a k-lex presentation $(D, <_{k\text{-llex}}, \{P_{a_i}\}_{i \leq k+1})$ of s_{k+1}, and assume it to be in normal form, i.e. $D \subseteq (\{0,1\}^k)^*$. Consider for each $i \leq k+1$ the (regular) relations $S_i(x, y)$ consisting of pairs of consecutive words $x, y \in P_{a_i}$, i.e. such that there are no occurrences of a_i on intermediate positions. Let automata be given for D,P_{a_i}, and S_i for every $i \leq k+1$ and let C be greater than the maximum of the number of states of any of

these automata.

Claim 1. *Let x represent the position of the n^{th} occurrence of a_{k+1} in s_{k+1}. Then $(k+1)n < |x| \leq Cn$, i.e. $|x| = \Theta(n)$, and hence $n = \Theta(|x|)$.*
The upper bound $|x| \leq Cn$ is clear, and $(k+1)n \leq |x|$ follows from that there are more than $2^{(k+1)n}$ symbols preceding the n^{th} a_{k+1} in s_{k+1}.

Claim 2. *For every $i = 1, \ldots, k$ there is a t_i such that for all $N \in \mathbb{N}$ there are $x = \otimes_k(x^{(1)}, \ldots, x^{(k)})$, and $y = \otimes_k(y^{(1)}, \ldots, y^{(k)})$ with $|x| = |y| > N$ and such that $S_i(x, y)$ and $x \sim_{k-i} y$ (i.e. $x^{(j)} = y^{(j)}$ for all $j \leq k-i$) and that $x^{(k-i+1)}$ and $y^{(k-i+1)}$ differ only on their last t_i bits.*
For $i = 1$ we immediately get a contradiction since between consecutive a_1's represented by words x and y of length $> N$ there are $2^{\Omega(N)}$ many a_0's but by Claim 2 there are only 2^{t_1} words between x and y in the k-lexicographic ordering.
Proof of Claim 2. We start with $i = k$ and proceed inductively in descending order. Values of the t_i will be implicitly given during the proof.

From Claim 1 we know that $|v| < |u| + C$ for every $S_{k+1}(u, v)$, and that if u represents the position of the n^{th} a_{k+1} then $n = \Theta(|u|)$. Then there are 2^n many a_k's distributed evenly between u and v, therefore there must be some $|u| \leq l \leq |v|$ such that there are still at least $2^n/C$ many $u <_{k\text{-llex}} x <_{k\text{-llex}} v$, $|x| = l$, and $x \in P_{a_k}$. When $n > C\log C$ then $2^n/C > 2^C$, so we have more than 2^C many $|x| = l$, and $x \in P_{a_k}$.

We claim that there are $x = \otimes_k(x^{(1)}, \ldots, x^{(k)})$ and $y = \otimes_k(y^{(1)}, \ldots, y^{(k)})$ such that $S_k(x, y)$ and $x^{(1)}$ and $y^{(1)}$ agree on their first C symbols. In deed, the first C symbols of the 1^{st} component can be incremented at most 2^C times and by the choice of n and l there are more than 2^C occurrences of a_k on length l.

Let now $t_k = l - C$. By pumping into the initial segment of length kC of the pair (x, y) (note that this involves the first C symbols of each component) we obtain arbitrary long x', y' with $S_k(x', y')$ whose 1^{st} components only differ on their last t_i bits. Thus we have established the case $i = k$.

To advance from $i+1$ to i we do the same as above. By the induction hypothesis we have for arbitrary large n two words $u = \otimes_k(u^{(1)}, \ldots, u^{(k)})$ and $v = \otimes_k(v^{(1)}, \ldots, v^{(k)})$ both of length n such that $S_{i+1}(u, v)$ and having $u^{(j)} = v^{(j)}$ for all $j < k - i$ and $u^{(k-i)}$ and $v^{(k-i)}$ differing only on their last t_{i+1} bits. By Claim 1 there are $2^{\Theta(n)}$ occurrences of a_i in between these two positions. On the other hand the remaining last t_i bits of the $(k-i)^{th}$ components together with the first C bits of the $(k-i+1)^{th}$ components only allow for 2^{C+t_i} possibilities. Hence for large enough n we must have two consecutive a_i's on positions represented by x and y agreeing on their first $(k-i)$ components and on the first C bits of their $(k-i+1)^{th}$ components. Thus, by pumping into the initial segment of length kC of the pair (x, y) we obtain arbitrary long x', y' fulfilling the conditions of Claim 2 for i.

$\square$

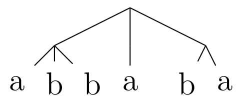


Figure 5.1: k-Stacks as depth k trees of unbounded branching.

5.6 k-morphic words

Let Γ be a finite, non-empty stack alphabet. A (level 1) stack is a finite sequence of symbols of Γ, and level $k+1$ stacks are sequences of level k stacks. Additionally, we shall call individual symbols of Γ level 0 stacks. Formally

$$\begin{aligned}\mathsf{Stack}_\Gamma^{(0)} &= \Gamma \\ \mathsf{Stack}_\Gamma^{(k+1)} &= [(\mathsf{Stack}_\Gamma^{(k)})^*]\end{aligned}$$

where '[' and ']' are used to identify the boundaries of lower-level stacks within higher-level ones. Outer most brackets will most often be omitted.

Level k stacks can be viewed as trees of height k having an unbounded number of ordered branches and leaves labelled by elements of Γ. See Figure 5.1 for an illustration. Each leaf, i.e. each level 0 element stored in a k-stack γ can be accessed by a vector of k indices $(i_0, \ldots, i_{k-1})$ leading to it. We denote the sequence of "leaves" of a $k+1$-stack γ, taken in the natural ordering, by $\mathsf{leaves}(\gamma)$. In other words, $\mathsf{leaves}(\gamma)$ is obtained from γ by forgetting the brackets.

The *concatenation* of two $(k+1)$-stacks $\gamma^{(k+1)} = [\gamma_1^{(k)} \ldots \gamma_s^{(k)}]$ and $\xi^{(k+1)} = [\xi_1^{(k)} \ldots \xi_t^{(k)}]$ is the $(k+1)$-stack $\gamma^{(k+1)} \cdot \xi^{(k+1)} = [\gamma_1^{(k)} \ldots \gamma_s^{(k)} \xi_1^{(k)} \ldots \xi_t^{(k)}]$. Concatenation can also be regarded as operations on trees. For $k > 0$ every k-stack $\gamma^{(k)} = [\gamma_0^{(k-1)} \ldots \gamma_{s-1}^{(k-1)}]$ can be written as the concatenation product $\prod_{i=0}^{s-1} [\gamma_i^{(k-1)}]$ and by propagating through all dimensions as

$$\gamma^{(k)} = \prod_{i_0} [\prod_{i_1} [\cdots \prod_{i_{k-1}} [\gamma^{(0)}_{(i_0,\ldots,i_{k-1})}] \cdots]] \tag{5.5}$$

where the index vector $(i_0, \ldots, i_{k-1})$ runs through all allowed tuples (all branches of length k) in a k-lexicographic fashion.

Definition 5.6.1 (Morphisms of k-stacks)**.** *Morphisms of k-stacks* over Γ are just k-stacks of actions of Γ. That is, $\mathsf{Hom}_\Gamma^{(k)} = \mathsf{Stack}_{\Gamma\to\Gamma}^{(k)}$, i.e. $\mathsf{Hom}_\Gamma^{(0)} = \Gamma \to \Gamma$ and $\mathsf{Hom}_\Gamma^{(k+1)} = [(\mathsf{Hom}_\Gamma^{(k)})^*]$. *Application* is defined inductively as follows.

- $\varphi^{(0)}(\gamma^{(0)})$ is as given
- for $\varphi^{(k+1)} = [\varphi_1^{(k)} \ldots \varphi_s^{(k)}] \in \mathsf{Hom}_\Gamma^{(k+1)}$ and $\gamma^{(k+1)} = [\gamma_1^{(k)} \ldots \gamma_t^{(k)}] \in \mathsf{Stack}_\Gamma^{(k+1)}$ $\varphi^{(k+1)}(\gamma^{(k+1)}) = [\varphi_1^{(k)}(\gamma_1^{(k)}) \ldots \varphi_s^{(k)}(\gamma_1^{(k)}) \cdots \varphi_1^{(k)}(\gamma_t^{(k)}) \ldots \varphi_s^{(k)}(\gamma_t^{(k)})] \in \mathsf{Stack}_\Gamma^{(k+1)}$

Having defined morphisms of k stacks we can make use of them to generate infinite k-stacks, and by collecting leaves, infinite words in the very same way as morphisms are used to generate morphic words. We baptise the infinite words thus obtained as k-morphic.

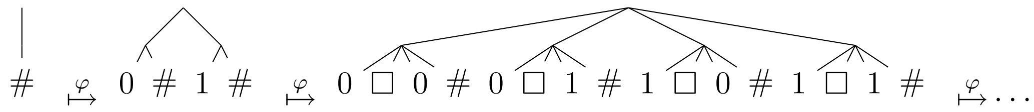


Figure 5.2: Iteratively applying $\varphi = [\tau\sigma]$ of Example 5.6.3 to $\gamma = [[\#]]$.

Definition 5.6.2 (k-morphic words)**.**
Let $k \in \mathbb{N}$. An infinite word $w \in \Sigma^\omega$ is *k-morphic* if there is a finite alphabet Γ, an initial k-stack $\gamma^{(k)} = [\cdots[\gamma_0^{(0)}]\cdots] \in \mathsf{Stack}_\Gamma^{(k)}$, a k-morphism $\varphi^{(k)} \in \mathsf{Hom}_\Gamma^{(k)}$ and a terminal homomorphism $h : \Gamma^* \to \Sigma^*$ such that

$$w = h\left(\prod_{n=0}^{\infty} \mathsf{leaves}(\varphi^n(\gamma))\right) \quad .$$

Note that our morphisms are *uniform*, e.g. $\mathsf{Hom}_\Gamma^{(1)}$ consists of the uniform homomorphisms of Γ^*. Please note that our definition is admittedly not streamlined for hands-on manipulation but much rather to be amenable to compact proofs. For a more "user-friendly" notation see the remarks at the end of this section. To illustrate the workings of morphisms of higher-level stacks consider the following level 2 example generating the binary Champernowne (cf. Example 5.3.3) word.

Example 5.6.3. Consider the initial 2-stack $\gamma = [[\#]]$ and the level 2 morphism $\varphi = [\tau\sigma]$ containing $\tau = [\tau_0\tau_1]$ and $\sigma = [\sigma_0\sigma_1]$ with

$$\tau : \left|\begin{array}{cccc} & & \tau_0 & \tau_1 \\ 0 & \mapsto & 0 & \square \\ 1 & \mapsto & 1 & \square \\ \# & \mapsto & 0 & \# \\ \square & \mapsto & \square & \square \end{array}\right. \qquad \sigma : \left|\begin{array}{cccc} & & \sigma_0 & \sigma_1 \\ 0 & \mapsto & 0 & \square \\ 1 & \mapsto & 1 & \square \\ \# & \mapsto & 1 & \# \\ \square & \mapsto & \square & \square \end{array}\right. .$$

Note that τ is just a complicated way of writing the morphism $(0 \mapsto 0, 1 \mapsto 1, \# \mapsto 0\#)$ in our framework as a sequence of 0-morphisms. Padding is needed to compensate for the inherent uniformity in our definition.

The stacks obtained in the first few iterations of φ on γ are depicted as trees in Figure 5.2. Let further h be the morphism erasing $\square$'s and $\#$'s. Then the 2-morphic word generated by φ on γ and filtered by h, is indeed the binary Champernowne word

$$0\,1\;00\,01\,10\,11\;000\,001\,010\,011\,100\,101\,110\,111\;\ldots$$

Clearly, an infinite word is 0-morphic iff it is *ultimately periodic*, and 1-morphic iff it is *morphic* in the customary sense despite the uniformity restriction on φ, which can be made up for by the choice of h. The next Lemma generalises (5.1).

Lemma 5.6.4 (Iteration Lemma)**.** Consider a k-stack $\gamma = [\cdots[\gamma_0]\cdots] \in \mathsf{Stack}_\Gamma^{(k)}$ and a morphism $\varphi = \varphi^{(k)} = \prod_{j_0}\left[\prod_{j_1}\left[\cdots\prod_{j_{k-1}}\left[\varphi^{(0)}_{j_0\cdots j_{k-1}}\right]\cdots\right]\right] \in \mathsf{Hom}_\Gamma^{(k)}$. Let B

be the set of those words $w = j_0 \dots j_{k-1}$ of length k corresponding to branches of the tree associated to φ, and let $\varphi_u^{(0)} = \varphi_{w_n}^{(0)} \circ \cdots \circ \varphi_{w_1}^{(0)}$ for all words $u = w_1 w_2 \cdots w_n \in B^*$. Then, applying φ n times to γ yields

$$\varphi^n(\gamma) = \underbrace{\prod_{u^{(1)}} [\prod_{u^{(2)}} [\cdots \prod_{u^{(k)}} [}_{u=\otimes_k(u^{(1)},\dots,u^{(k)})\in B^n} \quad \varphi_u^{(0)}(\gamma_0) \quad]\cdots]] \quad .$$

5.7 Equivalent characterisations

Consider a regular well-ordering $\prec$ of finite binary words and let $u_0 \prec u_1 \prec u_2 \prec \dots$ be the sequence of words in this ordering. We define the infinite word $w_\prec \in \{0,1,\#\}^\omega$ as the concatenation of the u_i in ascending order separated by # symbols: $w_\prec = u_0\#u_1\#u_2\#\cdots$. Let $w_{k-\text{llex}}$ be the word thus associated to $<_{k\text{-llex}}$ (restricted to words of length divisible by k). For instance,

$w_{1-\text{llex}} = \#0\#1\#00\#01\#10\#11\#000\#001\#010\#011\#100\#\dots$
$w_{2-\text{llex}} = \#00\#01\#10\#11\#0000\#0001\#0100\#0101\#0010\#0011\#0110\#0111\dots$

Further, let $w_{0-\text{llex}} = \#0\#00\#000\#\dots$. It is easy to see that $w_{k-\text{llex}} \in \mathcal{W}_{k+1}$ for all $k \in \mathbb{N}$. We say that a sequential transducer S with input alphabet $\{0,1,\#\}$ and output alphabet Σ is *#-driven* if it is deterministic and in each transition S produces either no output (i.e. the empty string ε) or a single letter output $a \in \Sigma$, but this only on reading a # on the input tape.

Theorem 5.7.1 (Equivalent Characterisations). *Let Σ be a finite alphabet. For every $k \in \mathbb{N}$ and every ω-word $w \in \Sigma^\omega$ the following are equivalent.*

(1) w is k-morphic

(2) w is k-lexicographic

(3) $w = S(w_{k-\text{llex}})$ for some #-driven sequential transduction S

(4) $\mathfrak{W}_w \leq^{\mathcal{I}}_{r\mathsf{MSO}} \mathfrak{W}_{w_{k\text{-llex}}}$ for an $\mathcal{I} = (\varphi_D, <, \{\varphi_a\}_{a\in\Sigma})$ s.t. $\models \forall x(\varphi_D(x) \to P_\#(x))$

Moreover, there are effective translations among these representations.

Proof.
(1)⇒(2):(For $k > 0$.) Let $w = h\left(\prod_{n=0}^{\infty} \mathsf{leaves}(\varphi^n(\gamma))\right)$ with $\gamma = [\cdots[\gamma_0]\cdots]$, φ and h as in the definition of k-morphic words. Consider the tree structure of φ, let l be the maximum of the number of children of any of the nodes, and let $B \subseteq [l]^k$ be the set of labels of ordered branches from the root to a leaf, using the natural ordering on $[l]$. We define the *index transition system* of φ as $\mathcal{I}_\varphi = (\Gamma, [l]^k, \delta)$ with $\delta(g,w) = \varphi_w^{(0)}(g)$ for each $g \in \Gamma$ and $w \in B$ and $\delta(g,w)$ undefined otherwise. Note

that for uniform morphism of words this definition is identical to that used in the proof of Proposition 5.1.4. By the Iteration Lemma

$$\mathsf{leaves}(\varphi^n(\gamma)) = \prod_{u \in B^n}^{<_{k\text{-llex}}} \varphi_u^{(0)}(\gamma_0)$$

and, since for each $g \in \Gamma$ the set $P_g = \{u \in B^* \mid \varphi_u^{(0)}(\gamma_0) = g\}$ is obviously accepted by $\mathcal{I}_\varphi$ with initial state γ_0 and single final state g, we can conclude that $(B^*, <_{k\text{-llex}}, \{P_g\}_{g\in\Gamma})$ is a k-lex presentation (in normal form) of $\hat{w} = \prod_{n=0}^{\infty} \mathsf{leaves}(\varphi^n(\gamma)) \in \Gamma^\omega$. By Proposition 5.3.4, $w = h(\hat{w})$ is also k-lex.

(2)⇒(1): (For $k > 0$.) By the Normal Form Lemma w has a k-lex presentation $(D, <_{k\text{-llex}}, \{P_a\}_{a\in\Sigma})$ in normal form over $\{0,1\}$, i.e. with D and each P_a being a regular subset of $(\{0,1\}^k)^*$. Recall $\mathcal{A}_\mathfrak{d}$, $\mathcal{T}\mathfrak{d}$, $\sigma_\mathfrak{d}$, etc. from Section 5.4. To provide a proof, we only need to adapt the notion of transition morphism to one over k-stacks. The stack alphabet will, of course, be $\Gamma = Q_\mathfrak{d}$. We define for each $l \le k$ and for every $u \in \{0,1\}^{k-l}$ a morphism $\tau_u^{(l)} \in \mathsf{Hom}_\Gamma^{(l)}$ recursively by setting $\tau_u^{(l+1)} = [\tau_{u0}^{(l)}\tau_{u1}^{(l)}]$ for each u of length $k-l-1$, $l < k$, and by setting $\tau_u^{(0)}(\vec{q}) = \delta_\mathfrak{d}^*(\vec{q}, u)$ for every $u \in \{0,1\}^k$. Finally, let $\varphi = \tau_\varepsilon^{(k)} = \prod_{j_0=0}^{1}\big[\prod_{j_1=0}^{1}\big[\cdots\prod_{j_{k-1}=0}^{1}\big[\tau_{j_0\cdots j_{k-1}}^{(0)}\big]\cdots\big]\big]$ and $\gamma = [..[\vec{q}_0]..] \in \mathsf{Stack}_\Gamma^{(k)}$. Observe that the structure of φ is the complete binary tree of depth k. Noting that $\tau_{w_n}^{(0)}(\ldots\tau_{w_2}^{(0)}(\tau_{w_1}^{(0)}(\vec{q}))\ldots) = \delta^*(\vec{q}, w_1w_2\ldots w_n)$ the Iteration Lemma yields

$$\varphi^n(\gamma) = \prod_{u^{(1)}=0^n}^{1^n}\Big[\prod_{u^{(2)}=0^n}^{1^n}\Big[\cdots\prod_{u^{(k)}=0^n}^{1^n}\big[\delta^*(\vec{q}_0, \otimes_k(u^{(1)},\ldots,u^{(k)}))\big]\cdots\Big]\Big]$$

and we can conclude that $w = \sigma_\mathfrak{d}(\prod_{n=0}^{\infty}\mathsf{leaves}(\varphi^n(\gamma)))$.

(2)⇒(3): (Hint) $\mathcal{S}$ simulates $\mathcal{A}_\mathfrak{d}$, restarting on every #.
(3)⇒(4): (Hint) The run of $\mathcal{S}$ is obviously restricted MSO-interpretable.
(4)⇒(2): There is a $k+1$-lex presentation $(\mathfrak{d}, \nu)$ of $w_{k\text{-llex}}$, similar to that given in Example 5.3.3, such that each maximal factor $u\#$ with $u \in \{0,1\}^*$ is represented on words $x \in D$ satisfying $x' = u$ and with the $k+1^{st}$ component telling the position within $u\#$. Let $\mathcal{I} = (\varphi_D, <, \{\varphi_a\}_{a\in\Sigma})$ be a restricted MSO- interpretation as in (4). By Theorem 5.2.5 each Colo-formula φ_a can be transformed into an equivalent automaton $\mathcal{A}_a$. Finally, to obtain a a k-lex presentation of $\mathcal{I}(\mathfrak{W}_{w_{k\text{-llex}}})$, we construct automata $\mathcal{A}'_a$ accepting those x' such that $x \in L(\mathcal{A}_a)$. □

5.8 Connection to the pushdown hierarchy

Given the fact that we have used morphisms of level k stacks to generate k-lex words and considering the nature of our decidability proof involving "higher-order" automata constructions a natural question to be asked is whether there is a connection to the hierarchy of graphs of higher-order pushdown automata (cf. Section 3.5.2). In this section we demonstrate that k-morphic words are on the $2k$-th level of the pushdown hierarchy of graphs. At this point we are not able to give lower bounds on their levels.

Note that it only makes sense to try to locate infinite words in the hierarchy of graphs, for unless a word is ultimately periodic it is not the unfolding of anything simpler than itself. Therefore we wish to view infinite words as graphs. To this end we identify each infinite word $a_1a_2a_3\dots$ with the edge-labelled successor graph

$$\bullet \xrightarrow{a_1} \bullet \xrightarrow{a_2} \bullet \xrightarrow{a_3} \cdots \quad .$$

Without doubt, the ω-words inhabiting the first level of the pushdown hierarchy are precisely the ultimately periodic ones. Indeed, by definition (cf. Section 3.5.2), the first level graphs are prefix-recognisable and those among them of finite degree are context-free (cf. Section 3.4.1) and as such, by a classical result of Muller and Schupp [MS83, MS85], have only finitely many *ends* up to isomorphism. For our word graphs this means precisely that they are ultimately periodic. The converse containment is obvious.

On the next level, Caucal [Cau02] has shown that morphic words, in the classical sense, are on the second level of the pushdown hierarchy. Whether they also exhaust the second level word graphs is, to the authors knowledge, not settled, though very plausible.

Starting with the third level, the pushdown hierarchy contains graphs of binary words of faster than exponential growth, which can hence not be automatic as can be verified by a standard pumping argument. An example of a fast growing sequence that is on the third level of the pushdown hierarchy [Bra06] is the characteristic sequence of the set of factorials, $01100010^{17}10^{95}10\dots$, also known as the Liouville word.

In order to place k-morphic words in the pushdown hierarchy, for each k we only need to locate a single tree $\mathfrak{T}_{<_{k\text{-llex}}}$, defined as follows. Let

$$T_{<_{k\text{-llex}}} = \{1^n\#w_1\#\otimes_2(w_1,w_2)\#\dots\#\otimes_k(w_1,w_2,\dots,w_k) \mid \forall i=1,..,k\ w_i \in \{0,1\}^n\}$$

and $\mathfrak{T}_{<_{k\text{-llex}}}$ be the tree $(\mathsf{Pref}(T_{<_{k\text{-llex}}}), succ_0, succ_1, succ_\#)$ illustrated on Figure 5.3. It has a single infinite branch 1^ω off of which at every position 1^n a finite subtree of depth $(n+1)k$ is hanging, the maximal paths of which are labelled by elements of $T_{<_{k\text{-llex}}}$. This set was designed so that the lexicographic ordering (for $\# < 0 < 1$) of these paths will correspond to the $<_{k\text{-llex}}$ ordering of their final segment below the last $\#$-edge.

We claim that an infinite word is k-lex iff its word graph is MSO-interpretable as a lexicographically ordered subset of the leaves of $\mathfrak{T}_{<_{k\text{-llex}}}$. Relying on the Normal Form Lemma 5.3.5 it is straightforward to give such an interpretation of a k-lex word. We defer the proof of the converse implication to Claim 6.2.4 of the next

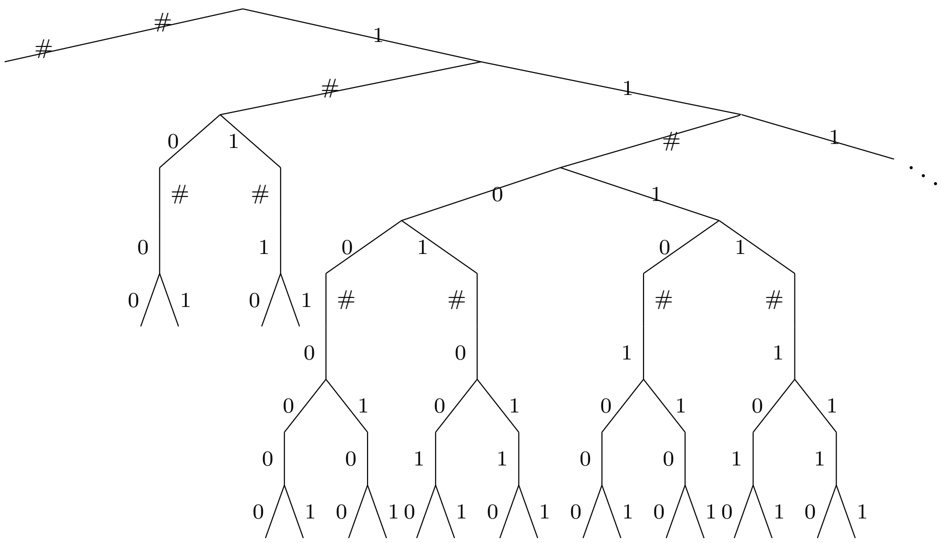


Figure 5.3: The tree $\mathfrak{T}_{<_{2\text{-llex}}}$ facilitating 2-lex words.

chapter. In the special case of unary relations the claim tells us that every such interpretation can be seen as a colouring of the leaves based on a regular condition on the path leading to each. It is then clear that the same colouring can be achieved by a regular condition on the path leading from the closest # to each leaf, since the path from the root to the leaf does not contain any information that could not be gathered from just the final segment.

Next, we show by induction that $\mathfrak{T}_{<_{k\text{-llex}}} \in \mathcal{G}raphs_{2k}$ for each $k > 0$ implying by our previous observation that k-morphic words are level $2k$ pushdown graphs.

Surely, $\mathfrak{T}_{<_{1\text{-llex}}}$ is an algebraic (level 2) tree as it is the unfolding of the graph of a one-counter automaton. This is essentially Caucal's argument [Cau02] showing that morphic words are on the second level of the pushdown hierarchy.

To proceed with the induction we give MSO-interpretations $\mathcal{I}, \mathcal{J}, \mathcal{K}$, such that $\mathfrak{T}_{<_{k+1\text{-llex}}} = \mathcal{K}(Unfold(\mathcal{J}(Unfold(\mathcal{I}(\mathfrak{T}_{<_{k\text{-llex}}})))))$ for each $k > 0$. This approach was first suggested to the author by Thomas Colcombet, the construction presented below was conceived during discussions with Arnaud Carayol and owes a lot to his assistance.

The first interpretation, $\mathcal{I}$, preserves the original structure while also introducing two kinds of new edges: 1) reflexive #-edges on all leaves; 2) $\bar{\sigma}$-labelled reversals of σ-edges, for $\sigma = 0, 1$, but only in "final segments": between nodes which do not have a #-edge in the subtree below them. Obviously, these definitions are MSO expressible.

It should be clear that the unfolding of $\mathcal{I}(\mathfrak{T}_{<_{k\text{-llex}}})$, let us denote this tree by T' for now, contains all branches of the form

$$1^n \# w_1 \# \otimes_2 (w_1, w_2) \# \ldots \# \otimes_k (w_1, w_2, \ldots, w_k) \# \overline{\otimes_k (w_1, w_2, \ldots, w_k)}^{rev} \qquad (5.6)$$

where $w_1, \ldots, w_k \in \{0,1\}^n$, and the last segment $\overline{\otimes_k(\ldots)}^{rev}$ denotes the reversal of $\otimes_k(\ldots)$ with barred symbols. This is precisely what we have intended. However,

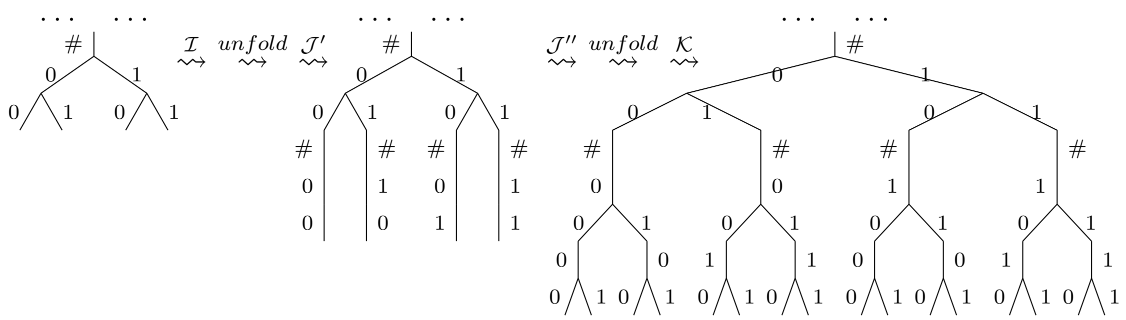


Figure 5.4: Constructing $\mathfrak{T}_{<_{2\text{-llex}}}$ from $\mathfrak{T}_{<_{1\text{-llex}}}$: illustration on a finite subtree.

aside of these, the unfolding produces an abundance of unwanted "junk" paths obtained by alternatingly traversing forward and backward edges and/or by passing through a reflexive edge more than once. The interpretation $\mathcal{J}$ is defined in order to achieve the following tasks.

- Restrict T' to nodes appearing on branches of type (5.6) above.
  This is done by forbidding unintended patterns, e.g. repeated reflexive edges, etc, as implicated above, on branches leading to a node from the root.

- Reversing the final $\overline{\otimes_k(w_1, w_2, \dots, w_k)}^{rev}$ segments of branches of type (5.6).
  This is a very simple operation, which can be done without producing any "junk": σ-labelled reversals of $\overline{\sigma}$-edges are added, while $\overline{\sigma}$-edges will be deleted, and those #-edges closest to a leaf are redirected to that leaf below them.

- Making room for w_{k+1} on every final segment: by introducing reflexive a- and b-labelled edges on nodes z from which the leaf below them is reachable on a #-free path of length divisible by k.

After the second unfolding we obtain a tree $T'' = Unfold(\mathcal{J}(Unfold(\mathcal{I}(\mathfrak{T}_{<_{k\text{-llex}}}))))$ which includes essentially $\mathfrak{T}_{<_{k+1\text{-llex}}}$ as an induced subtree (once a- and b-edges are renamed to 0 and 1 respectively), again, together with some unwanted branches arising from repeated traversals of reflexive a- or b-edges around the same node. The final clean-up needed is performed by the interpretation $\mathcal{K}$ by first restricting the domain to nodes reached from the root on a path avoiding immediate repetitions of a- or b-edges and finally renaming e.g. a-labels to 0 and b's to 1. The two-step construction is illustrated on Figure 5.4. [1] Thus we have established

Theorem 5.8.1. *For every k the word structure of every k-morphic word is on the $2k$-th level of the pushdown hierarchy: $\mathcal{W}_k \subset \mathcal{G}raphs_{2k}$.*

[1]Note that for the sake of a simpler illustration we decomposed $\mathcal{J}$ into two interpretations: $\mathcal{J}'$ purging unwanted branches produced by the previous unfolding and $\mathcal{J}''$ preparing ground for the second unfolding by the introduction of reverse edges and loops.

5.9 Remarks and questions

In this section we hint at some possible further generalisations of the results achieved in this chapter and raise a handful of related questions.

Variations on k-lex

Although it is unclear whether and how the results concerning MSO-friendliness, MSO-decidability, and -definability, as well as the embedding into the pushdown hierarchy of k-lex words carry over to all automatic presentations of ω-words, the stage is set for a simple generalisation.

To every finite word α over $\{l, r\}$ we associate a family of orderings, commonly denoted $<_\alpha$, over all ordered alphabets $\Sigma = \{a_0 < a_1 < \ldots < a_n\}$. In order to define $<_\alpha$ we introduce the notation $\overline{u}^l$ and $\overline{u}^r$ for every finite word u to stand for u, respectively, for the reversal of u, u^{rev}. Given an ordered alphabet Σ, and $\alpha \in \{l, r\}^k$ let

$$u <_\alpha v \overset{\text{def}}{\iff} |u| < |v| \vee \exists i < k : \; u =_i v \wedge \overline{u}^{\alpha_{i+1}} <_{\text{lex}} \overline{v}^{\alpha_{i+1}} \; .$$

Thus, $<_\alpha$ generalises $<_{k\text{-llex}}$ in that those components with an r in the respective position in α are compared not lexicographically but rather in reverse lexicographic order. The length comparison still remains prevailing to yield order type ω. Based on the given definition of α-lexicographic ordering we can introduce the following generalisation of k-lex presentations.

Definition 5.9.1 (α-lexicographic words)**.**
Let $\alpha \in \{l, r\}^k$ be given. An ω-word $w \in \Sigma^\omega$ is *α-lexicographic* (short: *α-lex*) if there is an automatic presentation $(D, <_\alpha, \{P_a\}_{a\in\Sigma})$ of the associated word structure $\mathfrak{W}_w$. For each α, the class of α-lexicographic words is denoted $\mathcal{W}_\alpha$, and we also let $\mathcal{W}^* = \bigcup_\alpha \mathcal{W}_\alpha$.

The classes $\mathcal{W}_\alpha$ form an infinite and possibly richer hierarchy as the classes of k-lex words. Let $\overline{l} = r$ and $\overline{r} = l$ and further extended to $\{l, r\}$-sequences. Clearly, $\mathcal{W}_\alpha = \mathcal{W}_{\overline{\alpha}}$ for each α since reversal of numerals assigned by the naming function in an α-lex presentation results in an $\overline{\alpha}$-lex presentation. Notice that the proof of the Hierarchy Theorem can be adapted to show that the $(k+1)$-st stuttering word s_{k+1} is not α-lex presentable for any $\alpha \in \{l, r\}^{\leq k}$. Also, if α is a proper subword (not necessarily a factor) of α' then $\mathcal{W}_\alpha \subsetneq \mathcal{W}'_\alpha$, where proper inclusion follows from the previous remark. But this is as far as such simple observations will lead us. A full comparison of the $\mathcal{W}_\alpha$ classes and a clear picture of the hierarchy remains open. It is for instance unclear how $\mathcal{W}_{lr}$ and $\mathcal{W}_{ll}$ are related.

We claim without giving a thorough proof that all α-lex presentations are MSO-friendly. This can be checked by adapting the proof of the Contraction Lemma heavily used in the inductive step in the proof of Theorem 5.4.3. One can argue that if the last symbol of α is l, i.e. if the last components are lexicographically ordered, then the proof goes through without any necessary adjustments. Furthermore, the

Contraction Lemma is invariant under reversal of all numerals of a presentation. Therefore, the Lemma holds for α iff it also holds for $\overline{\alpha}$, and obviously one of them ends with l. Thus, the MSO-decidability and -definability results extend to all α-lex words.

The embedding into the pushdown hierarchy is equally simple to adapt to α-lex words. Assuming the Normal Form Lemma, we can associate to each α the tree $\mathfrak{T}_\alpha$ to be constructed inside the pushdown hierarchy. Note, however, that we know of no better way of defining $\mathfrak{T}_{\alpha r}$ then via a construction from $\mathfrak{T}_{\overline{\alpha}l}$ involving unfolding. Also note that a single unfolding and MSO-interpretations suffice to build $\mathfrak{T}_{\overline{\alpha}l}$ from $\mathfrak{T}_\alpha$. For instance, $\mathfrak{T}_{rl} \in \mathcal{T}rees_3$.

Instead of dwelling on the technicalities of these constructions we would be eager to find the answer to the following more pressing questions. We conjecture that the answer to each of them is affirmative.

Question 5.9.2.

(1) Is every automatic presentation of an ω-word MSO*-friendly?*

(2) Is every automatic ω-word constructible in the pushdown hierarchy?

(3) Is every aut. pres. of an ω-word equivalent to an α-lex presentation?

Also note that we have thus far not found an equivalent way of generating those α-lex words with entangled l and r components as we have done with k-lex words. An effort would be worthwhile with the aim of finding a system of morphisms or the like generating all automatic words.

On uniformity of level k morphisms

Let us point out, that in the proof of (2)⇒(1) of Theorem 5.7.1 we made use of the Normal Form Lemma 5.3.5 to first uniformise the k-lexicographic presentation in preparation for turning it into a k-morphism generating the same word. This step was necessary due to the above hinted uniformity of our morphisms. Thus, Lemma 5.3.5 shows that this uniformity is really no restriction in terms of generating power as long as we allow ourselves to apply an arbitrary homomorphism h in the final step.

Nevertheless a formalism allowing description of non-uniform morphisms of higher levels would be of interest. To facilitate non-uniformity one can consider, for instance, tagged k-stacks instead of k-stacks and a kind of deterministic k-level indexed grammars... We shall not pursue formally defining these systems here but we give an illustrative example of what is meant.

Consider the level two rules

$$\begin{array}{rcl} S & \mapsto & SA_0 \\ A_\alpha & \mapsto & A_{\tau(\alpha)}B_{\sigma(\alpha)} \\ B_\beta & \mapsto & A_{\gamma(\beta)} \end{array}$$

where τ, σ, γ are level one rules, i.e. morphisms on words, e.g. $\tau : 0 \mapsto 00, 1 \mapsto 11$, $\sigma : 0 \mapsto 1, 1 \mapsto 0$, and $\gamma : 0 \mapsto 0, 1 \mapsto 10$. Then the first few derivations of S are

$$S \mapsto SA_0 \mapsto SA_0A_{00}B_1 \mapsto SA_0A_{00}B_1A_{0000}B_{11}A_{10} \mapsto \cdots$$

producing the word 000100001110 . . ..

It should be clear that the transformation from Theorem 5.7.1 of morphisms of k-stacks into k-lex presentations applies, with minor adjustments to these kinds of rules as well. In the above example, the first component in the 2-lex presentation would be a Fibonacci numeral corresponding to the derivation in the 2nd level rule $A \mapsto AB, B \mapsto A$ while the second component of the presentation would follow the derivation in the lower level rules.

Our definition of morphisms of k-stacks not only resembles that of morphisms of k-dimensional "pictures" (cf. [Mae99]), but is essentially identical with that up to a natural coding. Indeed, k-dimensional pictures are k-stacks satisfying the uniformity condition that every level $l + 1$ sub-stack consists of exactly the same number n_{l+1} of l-stacks, where $(n_1, \dots, n_k)$ are the dimensions of the picture. Due to their above mentioned uniformity our morphisms preserve uniformity of stacks. Hence, morphisms of k-stacks and morphisms of k-dimensional pictures are easily seen to be one and the same, up to this coding. However, while in [Mae99] morphisms of pictures were used to define relations of higher arity, we keep a linear structure that is *not* definable using the relations generated by associated morphisms using component-wise ordering.

Finite factors and combinatorics

Note that by Theorem 5.4.7 the (multi)sets of factors a^l of maximum sequences of consecutive a's of k-morphic words (or even automatic words for that) are already realised by 2-morphic words. Is this also true for sets of arbitrary finite factors?

We have seen that the Champernowne word having all finite words as factors, hence an exponential subword complexity, is 2-morphic. This is in contrast to the $\mathcal{O}(n^2)$ bound on the subword complexity of morphic words [AS03]. Analysing ω-regular sets using methods from descriptive set theory Staiger points out a key property of ω-words having maximal subword complexity, called *rich* in [Sta97]. We also note that the first-order theory of every rich ω-word is non-elementary, for it can interpret the finite satisfiability problem of FO on word structures.

We have mentioned that the Thue-Morse sequence of Example 5.1.2 has the uniform recurrence property. This means that every finite factor occurs infinitely often and that distances between consecutive occurrences of a factor of length n are bounded by some $c(n)$. In general we can only say, that in every automatic word the distances between consecutive occurrences of a given factor can not grow faster than exponentially.

We believe that these observations motivate a more thorough combinatorial analysis of higher-order morphic words.

Further questions

Interesting and difficult questions not considered here concern deciding the exact level of a given infinite word in our hierarchy, and deciding isomorphism of words on each level. Both of these problems have long been open for morphic words, that is for level one, having known solutions in very special cases only (see for instance [HR04] and the references therein).

Question 5.9.3.

(1) Is isomorphism of k-lexicographic words decidable?

(2) Let $k > k'$. Is it decidable whether a k-lex word is k'-lexicographic? In particular, is eventual periodicity of k-lex words decidable?

6 Regularity Preserving Transductions

When we think of an automatic structure, we frequently have a particular natural or canonical presentation in mind (for instance when the structure is defined over words to start with) or a typical presentation (e.g. the base k numeration system for Presburger arithmetic) if there is no apparent canonical one. That is to say we fix a, wlog. injective, presentation associated to a naming function f for each $\mathfrak{A} \in$ AutStr. Every other (injective) presentation of $\mathfrak{A}$ can then be seen as a (functional) transduction from the domain of the fixed presentation into a free monoid over a finite alphabet satisfying the constraint that the relations of $\mathfrak{A}$ have to remain regular after the transformation.

Often we have some additional constraints such as that some regular relations should be mapped to non-regular ones, while other non-regular relations should be transformed into regular ones. This is the case for instance when we wish to show that a certain relation is not intrinsically regular (cf. Chapter 7) with respect to a structure, or when we seek an automatic presentation of an expansion of $\mathfrak{A}$.

Working with injective presentations means that the transductions we are primarily interested in are bijective. Hence the notion of a translation as a bijection between regular sets (cf. Definition 4.2.2). Note that if f and g are injective naming functions corresponding to two automatic presentations of $\mathfrak{A}$ then $t = g^{-1} \circ f$ is a translation of names of elements of $\mathfrak{A}$ from one presentation into the other. A translation obtained this way preserves, by virtue of the presentations, regularity of (presentations of) all those sets and relations FO-definable in $\mathfrak{A}$. This motivates our interest in regularity preserving transductions.

Starting simple, in Section 4.2 we have shown that a translation preserves regularity and non-regularity of all relations iff it is computed by a semi-synchronous rational transducer. This provided us a particularly useful characterisation of equivalence of automatic presentations of arbitrary structures equally suited for the analysis of presentations of certain complete automatic structures. However, in every other case we are interested in more liberal transformations of presentations.

Following [PS05] we say that a transduction $\tau : M \to N$ between monoids M and N is *continuous* if τ^{-1} preserves recognisability of sets, i.e. if for every recognisable $R \subseteq N$, $\tau^{-1}(R)$ is recognisable. Note that both τ and τ^{-1} being continuous does not imply that they are semi-synchronous. For instance, τ mapping 0^n to 0^{2n} for all n while acting identically on all other words is continuous in both directions, however, not semi-synchronous.

6.1 MSO-definable string transductions

A transduction $T : \Sigma^* \to \mathcal{P}(\Gamma^*)$ is said to be MSO-definable if it is determined by an MSO-transduction $\mathcal{T}$ transforming word structures into word structures. We will only consider deterministic (i.e. parameterless) transductions, denoted DMSO, as we have defined them in Section 2.6.1.

Every DGSM mapping is MSO-definable, but the converse does not hold. The transduction $w \mapsto ww$ is an MSO-transduction, however, not computable by any DGSM. This example is obviously computable by a 2DGSM: a DGSM with a two-way read-only input tape and a one-way output tape. It is easy to see that 2DGSM mappings are MSO-definable string transductions. In [EH99, EH01] it is shown that DMSO string transductions are precisely those computable by 2DGSM's.

MSO-definable string transductions are closed under composition, they are continuous, though their inverses are generally not as witnessed by the mapping $w \mapsto ww$. Interestingly, it is decidable whether two 2DGSM's realise the same mapping (cf. [EH01].)

Recall our construction of Section 5.8 showing that k-lex words are on the $2k$-th level of the pushdown hierarchy. There we argued that k-lex words are MSO-definable in the tree $\mathfrak{T}_{<_{k\text{-llex}}} \in \mathcal{T}rees_{2k}$ (see Figure 5.3) whose set of leaves is

$$T_{<_{k\text{-llex}}} = \{1^n \# w_1 \# \otimes_2 (w_1, w_2) \# \ldots \# \otimes_k (w_1, w_2, \ldots, w_k) \mid \forall i \leq k \ w_i \in \{0,1\}^n\}$$

To look at this embedding from a different perspective consider the transduction $\tau_{<_{k\text{-llex}}}$ mapping

$$w \mapsto 1^{|w^{(1)}|} \# w^{(1)} \# \otimes_2 (w^{(1)}, w^{(2)}) \# \ldots \# \otimes_k (w^{(1)}, w^{(2)}, \ldots, w^{(k)}) \qquad (6.1)$$

where $(w^{(1)}, w^{(2)}, \ldots, w^{(k)})$ is the k-split of w as defined in Section 5.3. $\tau_{<_{k\text{-llex}}}$ is the mapping that embeds k-lexicographic presentations into the tree $\mathfrak{T}_{<_{k\text{-llex}}}$ by associating to every numeral the corresponding leaf of the tree.

It is a simple observation that $\tau_{<_{k\text{-llex}}}$ is realised by a 2DGSM. As such it is continuous, i.e. $\tau^{-1}_{<_{k\text{-llex}}}(L) = \{w \mid \tau_{<_{k\text{-llex}}}(w) \in L\}$ is regular whenever L is.

Most importantly, $\tau_{<_{k\text{-llex}}}$ transforms the k-length-lexicographic ordering into the lexicographic ordering, which is prefix-recognisable. It does this at the expense of appending redundant information to numerals of the presentation, thereby making the domain non-regular (but recognisable by a deterministic higher-order pushdown automaton (DHOPA) of level $2k$).

6.2 Translations mapping prefix-recognisable relations to regular ones

Let us fix a finite alphabet Σ for the rest of this section. Investigating possible enhancements of database query languages with string manipulating capability

Benedikt et al. have analysed subsystems of $\mathcal{S}_\Sigma$ from a model theoretic perspective [BLSS01, BLSS03]. One structure considered by Benedikt et al. is

$$\mathcal{S}_{\mathsf{Reg}(\Sigma)} = (\Sigma^*, \{\mathrm{succ}_a\}_{a\in\Sigma}, \preceq, \{P_L\}_{L\in\mathsf{Reg}})$$

where the relations P_L are defined as $\{(x, y) \mid \exists z(y = xz \wedge z \in L)\}$. The expansion of $\mathcal{S}_{\mathsf{Reg}(\Sigma)}$ with the greatest common prefix relation $\sqcap$ and the constant ε for the empty word allows quantifier elimination. This follows for instance from the following characterisation.

Proposition 6.2.1 (Läuchli and Savioz [LS87], see also [BLSS03])**.** The prefix-recognisable relations over the alphabet Σ are precisely those FO-definable in $\mathcal{S}_{\mathsf{Reg}(\Sigma)}$.

The strongest characterisation of prefix-recognisable relations in terms of logic is Theorem 3.4.3 (7) stating that prefix-recognisable relations are, *up to isomorphism*, FO-interpretable (in one-dimension) in Δ_2. However, Δ_2 does not allow a direct FO-definition of all PR relations, but typically some coding is necessary (cf. [Col07a, Lemma 5]). To overcome this we consider the structures

$$\mathcal{S}^0_{\mathsf{Reg}(\Sigma)} = (\Sigma^*, \{\mathrm{succ}_a\}_{a\in\Sigma}, \preceq, \{L\}_{L\in\mathsf{Reg}})$$

where each regular language L is identified with the unary predicate for membership in L. As a direct consequence of Theorem 3.4.4 we obtain the following strengthening of Proposition 6.2.1

Lemma 6.2.2. The prefix-recognisable relations over the alphabet Σ are precisely those FO-definable over $\mathcal{S}^0_{\mathsf{Reg}(\Sigma)}$

Proof.
From Theorem 3.4.3 and a successive remark concerning item (3) we know that every prefix-recognisable relation R over Σ is directly MSO-definable in $(\Sigma^*, \{\mathrm{succ}_a\}_{a\in\Sigma})$.

Let R be a relation defined by an MSO-formula $\phi(\vec{x})$ in the tree $(\Sigma^*, \preceq, \{\mathrm{succ}_a\}_{a\in\Gamma})$. According to Theorem 3.4.4 this definition decomposes into an MSO-marking $\mathcal{M}$ and an FO-interpretation $\mathcal{J}$ (which uses the prefix order $\preceq$ on the tree structure). Inspecting the proof of Theorem 3.4.4 in [Col07b] we see that $\mathcal{J}$ is (in this case) actually a single FO-formula ψ. (Subformulas of ψ compute, by aggregating relevant markings, predicates P_L for those $L \in \mathsf{Reg}$ involved in the prefix-recognisable expression for R, a' la Definition 3.3.10.) Since the markings produced by $\mathcal{M}$ are now regular, we can modify ψ by substituting its atomic relations referring to markings by corresponding predicates L of the signature, thus obtaining the required FO-definition of R in the tree $\mathcal{S}^0_{\mathsf{Reg}(\Sigma)}$. $\square$

Using the above lemma we can given give easy-to-check necessary and sufficient conditions for a translation to map all prefix-recognisable realtions to regular ones.

Theorem 6.2.3 (Transductions preserving regularity of all PR relations)**.**
Consider a bijection $t : D \to C$ between a regular $D \subseteq \Gamma^*$ and a prefix-closed $C \subseteq \Sigma^*$. Then, the following are equivalent

(i) *t is continuous and the inverse image of the prefix-relation under t is a regular relation, for short: $t^{-1}(\preceq) \in \mathsf{Reg}$;*

(ii) *t is the (injective) naming function of an automatic presentation of $\mathcal{S}^0_{\mathsf{Reg}(\Sigma)}\big|_C$;*

(iii) *the inverse image of every prefix-recognisable relation under t is regular, for short: $t^{-1}(\mathsf{PR}) \subseteq \mathsf{Reg}$.*

Proof. Statements (i) and (ii) are equivalent by definition, and (i) trivially follows from (iii). To check $(ii) \Rightarrow (iii)$ let $R \subseteq (\Sigma^*)^n$ be an arbitrary prefix-recognisable relation. By Lemma 6.2.2 R is FO-definable in $\mathcal{S}^0_{\mathsf{Reg}(\Sigma)}$, hence $R \cap C^n$ is FO-definable in $(\mathcal{S}^0_{\mathsf{Reg}(\Sigma)}, C)$, therefore also in $\mathcal{S}^0_{\mathsf{Reg}(\Sigma)}\big|_C$ since C is prefix-closed and since all subtrees of $\mathcal{S}^0_{\mathsf{Reg}(\Sigma)}$ disjoint from C are regular, hence there is no loss of information when disposing of them in $\mathcal{S}^0_{\mathsf{Reg}(\Sigma)}\big|_C$. Thus, by the Fundamental Theorem 3.1.3, R is regularly presented under t^{-1}. $\square$

We shall henceforth refer to transductions satisfying Theorem 6.2.3 as *PR-transductions*.

6.2.1 Alternative proof of MSO-friendliness of k-lex presentations

In Section 5.8 of the previous chapter we have shown that all k-lex words are constructible in the pushdown hierarchy by defining them on the leaves of the respective tree $\mathfrak{T}_{<_{k\text{-llex}}}$. Previously in this section we have argued that the embedding $\tau_{<_{k\text{-llex}}}$ from k-lex presentations in normal form to leaves of $\mathfrak{T}_{<_{k\text{-llex}}}$ was in fact continuous, being 2DGSM-computable. Although we have defined $\tau_{<_{k\text{-llex}}}$ to take on values only among the leaves of $\mathfrak{T}_{<_{k\text{-llex}}}$, this mapping can be easily modified to include all nodes of $\mathfrak{T}_{<_{k\text{-llex}}}$.

Every node of $\mathfrak{T}_{<_{k\text{-llex}}}$ below $1^n\#$ (i.e. belonging to the n-th finite subtree) is led to on a path marked out by some number of components $x_1 \ldots, x_i$, $i \leq k$, of which all but perhaps x_i is "complete", that is of length n. Therefore, a natural idea is to extend (6.1) to convolutions of such incomplete tuples. However, while every node of $\mathfrak{T}_{<_{k\text{-llex}}}$ uniquely determines an incomplete tuple $x_1 \ldots, x_i$, to each such tuple up to $i+1$ nodes may be associated. This can be made up for, say, by an additional component containing the finite bit of information necessary to uniquely determine a node. With this in mind we let $\vartheta_{<_{k\text{-llex}}}$ map $\otimes_{k+1}(j^n, x_1, \ldots, x_k)$ to

$$1^n \Big(\prod_{t=1}^{i-1} \# \otimes_t (x_1, \ldots, x_t)\Big) \Big((\# \otimes_i (x_1, \ldots, x_i)\#x_1[0] \cdots x_{i-1}[0])[0..i|x_i| + j)\Big) \quad (6.2)$$

for all words $\otimes_{k+1}(j^n, x_1, \ldots, x_k)$ such that $n = |x_1| = \cdots = |x_{i-1}| \geq |x_i| > 0 = |x_{i+1}| = \cdots = |x_k|$ for some $0 \leq i \leq k$ and either $i = 0$ and $j \in \{0, 1\}$ or $0 < i$ and $|x_i| < n$ and $1 \leq j \leq i$ or $i = k$ and $|x_i| = n$ and $1 \leq j \leq i+1$. In particular, for $j = i = 0$ the mapping is $(0\Box^k)^n \mapsto 1^n$, for $i = 0$, $j = 1$ it is $(1\Box^k)^n \mapsto 1^n\#$, and for $i > 0$, $j = 1$ and $|x_i| = n$ it essentially coincides with $\tau_{<_{i\text{-llex}}}$, except for the presence of the superfluous 1^n and empty input components. In accordance with our initial remarks, the refinements as to (6.1) are meant to ensure that the image of $\vartheta_{<_{k\text{-llex}}}$ is precisely the set of nodes of $\mathfrak{T}_{<_{k\text{-llex}}}$.

When restricted to words representing leaves, $\vartheta_{<_{k\text{-llex}}}$ is equivalent to $\tau_{<_{k\text{-llex}}}$ in the sense of Section 4.2. Indeed, $t = \vartheta^{-1}_{<_{k\text{-llex}}} \tau_{<_{k\text{-llex}}}$ is the $(k, k+1)$-synchronous translation $t : \otimes_k(x^{(1)}, \ldots, x^{(k)}) \mapsto \otimes_{k+1}(1^{|x^{(1)}|}, x^{(1)}, \ldots, x^{(k)})$.

Despite the awkwardness of its definition it should be clear that $\vartheta_{<_{k\text{-llex}}}$ is 2DGSM computable, indeed, in at most $k+1$ sweeps, hence it is continuous. The additional conditions on the input regarding n, $|x_i|$, i, and j are obviously regular. Furthermore, it poses no difficulty to check that $\vartheta^{-1}_{<_{k\text{-llex}}}(\preceq)$ is regular. Indeed, $\vartheta_{<_{k\text{-llex}}}(\otimes_{k+1}(0^n, \varepsilon, \ldots, \varepsilon)) \preceq \vartheta_{<_{k\text{-llex}}}(\otimes_{k+1}(l^m, y_1, \ldots, y_k))$ iff $n \leq m$, in every other case $\vartheta_{<_{k\text{-llex}}}(\otimes_{k+1}(j^n, x_1, \ldots, x_k)) \preceq \vartheta_{<_{k\text{-llex}}}(\otimes_{k+1}(l^m, y_1, \ldots, y_k))$ implies that $n = m$ and $x_l = y_l$ for all l such that $|x_l| = n$ and $x_i \preceq y_i$ for i such that $0 < |x_i| < n$, if it exists, and finally that $|x_i| + j \leq |y_i| + l$. Thus, we can conclude that $\vartheta_{<_{k\text{-llex}}}$ is a PR-transduction. In particular, it constitutes the naming function of an automatic presentation of $(\mathfrak{T}_{<_{k\text{-llex}}}, \preceq)$.

Let us now consider an arbitrary k-lex word $w \in \mathcal{W}_k$ with associated word structure $\mathfrak{W}_w$ and a k-lexicographic presentation $\mathfrak{d} = (D, <_{k\text{-llex}}, \{P_a\}_a)$. Wlog. $\mathfrak{d}$ is in normal form (cf. Lemma 5.2.4), therefore, it can be realised as an MSO-interpretation $\mathcal{I}$ in $\mathfrak{T}_{<_{k\text{-llex}}}$ such that

$$\mathfrak{W}_w \;\cong^{f^{-1}}\; (D, <_{k\text{-llex}}, \{P_a\}) \;\cong^{t}\; (D', <_{k+1\text{-llex}}, \{P'_a\}) \;\cong^{\vartheta_{<_{k\text{-llex}}}}\; \mathcal{I}(\mathfrak{T}_{<_{k\text{-llex}}}) \quad (6.3)$$

where f is the naming function of the presentation $\mathfrak{d}$.

By virtue of interpretations, to every MSO-formula $\varphi(\vec{x})$ defining some relation R in $\mathfrak{W}_w$ we can associate the formula $\varphi^{\mathcal{I}}$ defining $\vartheta_{<_{k\text{-llex}}}(t(f^{-1}(R)))$ in $\mathfrak{T}_{<_{k\text{-llex}}}$.

Claim 6.2.4. For every $\psi(\vec{x})$ there is a prefix-recognisable relation P such that restricted to leaves of $\mathfrak{T}_{<_{k\text{-llex}}}$ they coincide: $\psi^{\mathfrak{T}_{<_{k\text{-llex}}}} \cap T_{<_{k\text{-llex}}} = P \cap T_{<_{k\text{-llex}}}$.

Assuming this, $\vartheta_{<_{k\text{-llex}}}(t(f^{-1}(R)))$ is the restriction of a prefix-recognisable relation P to the leaves of $\mathfrak{T}_{<_{k\text{-llex}}}$. Thus, $t(f^{-1}(R))$ is the restriction of $\vartheta^{-1}_{<_{k\text{-llex}}}(P)$ to tuples of words of the form $\otimes_{k+1}(1^{|x^{(1)}|}, x^{(1)}, \ldots, x^{(k)})$. Since $\vartheta_{<_{k\text{-llex}}}$ is a PR-transduction this yields that $t(f^{-1}(R))$ is regular. Finally, by semi-synchronicity of t, $f^{-1}(R)$ is also regular. Since $\varphi(\vec{x})$ was arbitrary this proves that $\mathfrak{d}$ is MSO-friendly.

To prove Claim 6.2.4 we employ some very simple tree transducers, which are, analogously with rational transductions of words, known to be continuous. For the purposes of this proof an intuitive understanding of tree transducers suffices. As an exposition to the subject we recommend [FV98]. We shall not introduce tree transducers here formally, for only variants of the following transduction will be used.

$$T_{\text{blow}} \left| \begin{array}{lcl} q(a(x)) & \rightarrow & n(q(x), q(x)) \\ q(\#) & \rightarrow & \# \end{array} \right.$$

In plain words, T_{blow} maps each word $a^n\#$, seen as a monadic tree of a single branch, to the full binary tree of height n with its leaves labelled by $\#$. It does this with a single state, hence the transduction realised is called a tree homomorphism. We will use extensions and variants of this simple scheme which

- on a binary input word x produce the full binary tree of height $|x|$ with its branch labelled x specially marked;
- apply a copying rule only on every k-th position along every path;
- simultaneously simulate the run of a given DFA $\mathcal{A}$ along every path of the output tree, and labelling its leaves with the final state of the run of $\mathcal{A}$ along corresponding branches.

These aims can be achieved in general by rules of the form

$$T_{\mathcal{A},k} \left| \begin{array}{lcll} [q,j](\sigma(x)) & \to & \sigma([q',j+1 \mathrm{mod} k]) & \delta(q,\sigma)=q' \\ [q,0](\sigma(x)) & \to & n_\sigma([q_0,1](x),[q_1,1](x)) \quad i,\sigma\in\{0,1\} & \delta(q,i)=q_i \\ [q,j](\#) & \to & \#_q & \end{array} \right.$$

with states $[q,j]$ composed of a state q of the DFA $\mathcal{A}$ and $0 \leq j < k$ and with δ being the transition function of $\mathcal{A}$.

Fact 6.2.5. Each transduction $T_{\mathcal{A},k}$ is continuous, meaning that the inverse image of every regular set of trees is regular.

I thank Łukasz Kaiser for pointing out, that in this special case of word to tree transductions this can be shown e.g. by directly constructing for any given non-deterministic top-down tree automaton A on output trees an alternating automaton A' on input words accepting $T^{-1}_{\mathcal{A},k}(L(A))$. Intuitively, universal choices of A' on an input word simulate choices among branches of the output tree produced in a copying transition, while existential choices of A' correspond to non-deterministic transitions of A.

Fact 6.2.5 can be seen as the analogue of the Contraction Lemma 5.4.2. Moreover, in the proof of the latter we have used the morphism β to which the above $T_{\mathcal{A},k}$ bear great resemblance.

We shall also make use of the following well-known fact, which can be deduced either with the composition technique or with the aid of tree automata. The next claim constitutes an analogue of Lemma 5.3.5 for trees.

Fact 6.2.6. For every MSO-formula $\psi(\vec{x})$ on Σ-branching unlabelled trees there are MSO-definable markings $\{\varphi_c(x) \mid c \in \Gamma\}$ and a prefix-recognisable relation $P \in PR((\Sigma\times\Gamma)^*)$ such that in each of the formulae $\varphi_c(x)$ quantification is relativised to the subtree below x and such that over every tree T and every tuple of nodes $x_1,\dots,x_n$ of T

$$T \models \psi(\vec{x}) \iff \pi(\vec{x}) \in P$$

where $\pi(x)$ denotes for every node x the finite sequence of directions and labels along the path leading to x from the root as a word in $(\Sigma\times\Gamma)^*$.

We now have the essential ingredients to prove Claim 6.2.4.

Proof. of Claim 6.2.4
Towards a conclusion we take the markings $\{\varphi_c(x) \mid c \in \Gamma\}$ and the prefix-recognisable relation $\hat{P}$ associated to $\psi(\vec{x})$ by Fact 6.2.6, and prove, that restricted to leaves of $\mathfrak{T}_{<_{k\text{-llex}}}$, the mapping $x \mapsto \pi(x)$ as described there is in fact synchronised rational.

Although π fails to be regular over all internal nodes of $\mathfrak{T}_{<_{k\text{-llex}}}$, this observation is already sufficient to establish our claim for we consider only relations on leaves. Indeed, as $\pi(x \sqcap y) = \pi(x) \sqcap \pi(y)$ and given a DFA $\mathcal{P}$ recognising pairs $(x, \pi(x))$ we can convert every relation $P_L(\pi(x), \pi(y))$ into an equivalent prefix-recognisable form

$$\bigvee_q (\exists w\ \mathcal{P} \text{ accepts } (x, w) \text{ in final state } q) \wedge (\exists w \in L\ \mathcal{P} \text{ accepts } (x^{-1}y, w) \text{ from } q)$$

we can thus convert every prefix-recognisable relation $\hat{P}$ on π-values into an equivalent prefix-recognisable relation P on paths leading to leaves of $\mathfrak{T}_{<_{k\text{-llex}}}$. By construction P coincides with $\psi(\vec{x})$ on leaves as required.

It remains to be proved that π obtained from arbitrary MSO-definable markings $\{\varphi_c(x) \mid c \in \Gamma\}$, where each $\varphi_c(x)$ is relativised to the subtree rooted at x, is indeed synchronised rational when restricted to leaves. As a first step we show, using Fact 6.2.5, that

Lemma 6.2.7. Let $\varphi(x)$ be an MSO-formula in which all quantifiers are relativised to the subtree below the node represented by the single free variable x. Then a regular language L_φ can be constructed such that for every branch from the root of $\mathfrak{T}_{<_{k\text{-llex}}}$ labelled by a word

$$z \;=\; 1^n \# x_1 \# \otimes_2 (x_1, x_2) \# \ldots \# \otimes_i (x_1, \ldots, x_i) \#$$

the subtree rooted at the node z satisfies φ iff $z \in L_\varphi$.

Proof. Clearly, it is sufficient to consider only Hintikka formulas $\varphi(x)$ completely describing some r-theory of trees for a fixed quantifier rank r.

In comparison with our approach of Section 5.3 we note that each z as above represents a $=_i$ class of numerals sharing their first i components.

In much the same way as in Section 5.4.2 the construction of the L_φ's is achieved by iterated contractions, i.e. by inductively applying Fact 6.2.5 in each step. The induction base is the superfluous case of $i = k$, i.e. of leaves z. To proceed we apply a variant of the tree transductions $T_{\mathcal{A},i}$ to z to produce the incomplete subtree with leaves

$$z \;=\; 1^n \# x_1 \# \otimes_2 (x_1, x_2) \# \ldots \# \otimes_i (x_1, \ldots, x_i) \# \otimes_{i+1} (x_1, \ldots, x_i, y)$$

with y ranging over all words of length n. If $\mathcal{A}$ is chosen to be a DFA recognising, with different final states, L_φ for every Hintikka formula φ of some r-theory, then each leaf of the output tree $T_{\mathcal{A},i}(z)$, which is also an internal node of $\mathfrak{T}_{<_{k\text{-llex}}}$, is labelled with a state encoding the r-type of the subtree of $\mathfrak{T}_{<_{k\text{-llex}}}$ rooted in that node.

Hence, the r-type of the subtree rooted at z can computed by a bottom-up tree automaton working on $T_{\mathcal{A},i}(z)$. Because $T_{\mathcal{A},i}$ is continuous, the same r-type can be computed by a word automaton reading z. □

To finish the proof of the regularity of π we consider yet another variant of the tree transductions, $\tilde{T}_{\mathcal{A},i}$, applied this time to convolutions of pairs $x \otimes w$ with

$$x = 1^n \# x_1 \# \otimes_2 (x_1, x2) \# \dots \# \otimes_k (x_1, \dots, x_k)$$

a leaf of $\mathfrak{T}_{<_{k\text{-llex}}}$ and w a candidate for $\pi(x)$. Intuitively, $\tilde{T}_{\mathcal{A},i}$ acts much like $T_{\mathcal{A},k}$ on each segment of the input word between consecutive #'s by blowing up these word segments into completely branching trees at every k-th position and simultaneously simulating runs of $\mathcal{A}$ on each of the thus created branches. However, when a # is encountered in the input word, both the branching and the simulations terminate and a next cycle is started below the branch identical with the prefix of the input word. We illustrate this transduction in Figure 6.1.

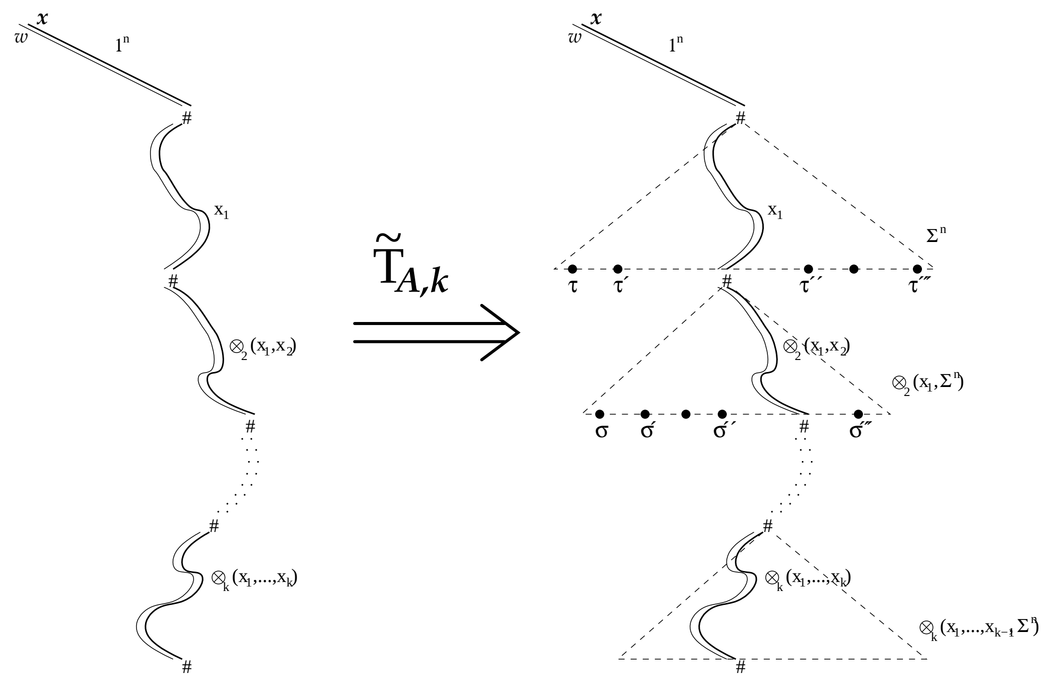


Figure 6.1: Tree transducer for type checking

Let now r be the quantifier rank of ψ (thereby also an upper bound on the quantifier rank of any of the markings $\varphi_c(x)$). If $\mathcal{A}$ is chosen so that it is able to recognise each of the L_φ's obtained in each of the contraction steps for each Hintikka formula describing an r-theory as described in Lemma 6.2.7 then the states of $\mathcal{A}$ assigned to the leaves of the output tree by $\tilde{T}_{\mathcal{A},i}$ directly determine the r-theories of the subtrees rooted in the corresponding nodes of $\mathfrak{T}_{<_{k\text{-llex}}}$ (and denoted as $\tau, \tau', \ldots\ \sigma, \sigma', \ldots$ in Figure 6.1).

Again, as in Lemma 6.2.7 we conclude that since checking that the types as given in the input by the w component are consistent with those assigned to the leaves by the transduction can be performed by, say, a bottom-up tree automaton, the set of correctly labelled pairs $(x, \pi(x))$ of inputs is regular as well. In other words, that π is synchronised rational on maximal paths x of $\mathfrak{T}_{<_{k\text{-llex}}}$.

This concludes the proof of Claim 6.2.4 $\square$

We have thus provided an alternative proof of Theorem 5.4.3 and its corollaries stated in Section 5.4.2. Roughly speaking, we have traded the framework of chains of homomorphisms for compositions of tree transducers, however, the two proofs bear some resemblance. The hope would be that the latter approach involving tree transducers might lend itself more easily to generalisations for automatic presentations of ω-words based on arbitrary implementation of the ordering relation. However, while we can do this by hand for particular automatic presentations, we still have not found a general construction achieving the same for arbitrary automatic orderings of type ω.

6.3 Representable transductions

It is well known that every rational transduction $\tau : \Sigma^* \to M$ admits a *linear representation* of the form (I, μ, T), where $\mu : \Sigma^* \to \mathsf{Rat}(M)^{n\times n}$ is a homomorphism and $I, T \in \mathsf{Rat}(M)^n$ for some $n > 0$ such that $\tau(x) = I^t\mu(x)T$ for every $x \in \Sigma^*$. Conversely, every such triple (I, μ, T) determines a rational transduction [Ber79].

In [PS85] Pin and Sakarovitch have proposed a very general framework for constructing continuous transductions by allowing far more liberal matrix presentations. *Representable transductions* are composed of a homomorphism $\mu : \Sigma^* \to \mathcal{P}(M)^{n\times n}$ and a mapping $\nu_U : \mathcal{P}(M)^{n\times n} \to \mathcal{P}(M)$ defined by substituting the matrix entries $L_{i,j}$ in place of corresponding $X_{i,j}$ for $1 \le i, j \le n$ into a fixed $U \subseteq (M \cup \{X_{i,j}\}_{1\le i,j\le n})^*$. Representable transductions are easily seen to be continuous by diagram chasing [PS85].

Clearly, for every I and T as in a linear representation one can take $U = \bigcup_i \bigcup_j I_i X_{i,j} T_j$ yielding $\tau(x) = I^t\mu(x)T = \nu_U(\mu(x))$. Thus, representable transductions do subsume rational transductions, but are not only more general due to the relaxation that μ may take non-rational entries in $\mathcal{P}(M)$. A more substantial increase in expressive power is achieved by specifying non-linear combinations of matrix entries using U.

We are of course interested in functional transductions. Note that if $\mu : \Sigma^* \to M^{n\times n}$ and $u \in (M \cup \{X_{i,j}\}_{1\le i,j\le n})^*$ then $\nu_u : M^{n\times n} \to M$ and $\tau = \nu_u \circ \mu$ are functional. This is already sufficient to see that e.g. the mapping $w \mapsto ww$ is representable. Also the mappings $\tau_{<_{k\text{-llex}}}$ and $\vartheta_{<_{k\text{-llex}}}$ from the previous subsections are representable for each k.

Despite the richness and robustness of the class of representable transductions, we suspect that they do not capture all continuous transductions. In particular, that inverses of run-length encodings of the next subsection are not representable.

6.4 Run-length encodings

Run-length encoding (RLE) is an extremely simple compression scheme that has actually found application in compressing bitmaps in the early days of a popular operating system. Run-length encoding works by removing "long" sequences of consecutive occurrences of a symbol in a data stream and replacing it with the length of the sequence as a binary numeral and a hint at the symbol repeated. It is perhaps best explained on an example, such as the following.

$$bbbbbaaabbbbaaaaaaaabbbbbb \quad \overset{\text{RLE}}{\mapsto} \quad b101a11b100a1000b1010\ .$$

We may observe, that thanks to the simplicity of this scheme, RLE has the notable feature that compressed words can be checked against a regular expression without decompression. In other words, the image of every regular set under the RLE encoding is again regular: RLE is regularity preserving. This is indeed very easy to see appealing to the fact that the behaviour of finite automata on a sequence of a's is periodic and modulo counting is automaton computable on binary numerals as well. We may also add that RLE preserves regularity of the prefix relation as well.

These observations call for the following generalisation. Every automatic presentation of an omega-word gives rise to a similar compression method, which preserves regularity of sets as well as of the prefix relation.

Let $\mathfrak{d} = (D, <, \{P_a\}_{a\in\Sigma})$ be an automatic presentation of an ω-word $w \in \Sigma^\omega$ with injective naming function $f : D \to \mathbb{N}$. Let Γ be the alphabet of this presentation and assume that it is disjoint from Σ. Associated to $\mathfrak{d}$ we define a mapping $\rho : \Sigma^* \to (\Sigma \cup \Gamma \cup \{[,]\})^*$ as follows. Every word $u \in \Sigma^*$ has a unique factorisation $u = xyz$ such that y is the left-most maximal factor of u that is a prefix of w. Given this factorisation of u its image is $\rho(u) = x[f^{-1}(|y|)]\rho(y)$. ρ is thus well defined.

Consider for example $w = a^\omega$ represented in the standard binary numeration system. In this case ρ behaves much like the by now well-known run-length encoding e.g.

$$bbbbbaaabbbbaaaaaaaabbbbbb \quad \overset{\rho_{a^\omega}}{\mapsto} \quad bbbbb[11]bbbb[1000]bbbbbb\ .$$

Actually, a composition of ρ_{a^ω} and ρ_{b^ω} is almost identical to RLE as introduced above. But let's take as a second example the compression according to the binary presentation of the Thue-Morse sequence $t = abbabaabbaababba\ldots$ illustrated on the following word (with the appropriate factorisation hinted):

$$bb\,abba\,abb\,b\,abbabaabb\,b\,ab\,a \quad \overset{\rho_t}{\mapsto} \quad bb[100][11]b[1001]b[10][1]\ .$$

So, in general, ρ_w – we write ρ_w when the underlying presentation of w is understood – compresses maximal prefixes of w occurring in the input word into the representations of their respective lengths.

The fact that ρ_w originated from an automatic presentation $\mathfrak{d}$ of w ensures that ρ_w preserves regularity of the prefix relation: if $v = uz$ then either $\rho_w(v) = \rho_w(u)\rho_w(z)$ or it is the case that u ends with a prefix of w continued in z and hence $\rho_w(v) = \rho_w(x)[f^{-1}(m)]\rho_w(y)$ and $\rho_w(u) = \rho_w(x)[f^{-1}(n)]$ for some x, y and $n < m$. These conditions can be checked by an automaton reading the compressed words $\rho_w(u)$ and $\rho_w(v)$ by invoking the automaton representing the ordering in $\mathfrak{d}$. Further, observing that a compressed word $\rho_w(u)$ either ends with a symbol $a \in \Sigma$ or with some $f^{-1}(n) \in D$ we see that the last symbol of u can in either case be recovered from $\rho_w(u)$ by an automaton using the automata of the presentation $\mathfrak{d}$. We may thus conclude that ρ_w preserves regularity of all star-free sets, for these are first-order definable from prefix-order and predicates for the terminal symbol.

How about preserving regularity of all sets? We have seen that ρ_{a^ω} associated to the simplest infinite word thinkable, a^ω, is regularity preserving. This remains true assuming an underlying presentation $\mathfrak{d}$ that is MSO-friendly. In fact, we have the following.

Proposition 6.4.1. Let $w \in \Sigma^\omega$ and an automatic presentation $\mathfrak{d}$ of its word structure $\mathfrak{W}_w$ be given. The associated compression scheme ρ_w is regularity preserving iff every MSO-definable subset of $\mathfrak{W}_w$ is regularly presented in $\mathfrak{d}$.

Indeed, both conditions require for every regular language L that the set $\{f^{-1}(n) \mid w[0,n] \in L\}$ be regular. By Theorem 3.4.4 this is already sufficient to conclude that every MSO-definable relation of $\mathfrak{W}_w$ is regularly presented in $\mathfrak{d}$, in other words that $\mathfrak{d}$ is MSO-friendly.

Corollary 6.4.2. *Let $\mathfrak{d}$ be an automatic presentation of w as above. Then the associated compression scheme ρ_w is regularity preserving iff $\mathfrak{d}$ is MSO-friendly.*

In particular, if ρ_w is based on a k-lexicographic presentation of some w, then it is regularity preserving. Another conclusion to be drawn from these considerations is, as we have promised, that infinite prefix-recognisable structures are not rigidly automatic.

Theorem 6.4.3. *Infinite prefix-recognisable structures are not rigidly automatic.*

Proof. Let $\mathfrak{A} = (D, \{R_i\})$ be a prefix-recognisable structure over an alphabet Σ. That is, $D \in \mathsf{Reg}(\Sigma^*)$ and each $R_i \in \mathsf{PR}(\Sigma^*)$. If D is infinite, then there are words $u, v, z \in \Sigma^*$, such that $uv^*z \in D$ for every n. Consider the run-length encoding scheme ρ_w associated to the binary presentation of the ultimately periodic word $w = uv^\omega \in \Sigma^\omega$.

By Corollary 6.4.2 ρ_w^{-1} is continuous, and we have already noted that $\rho_w(\preceq)$ is a regular relation for every automatic w. Therefore, by Theorem 6.2.3, ρ_w^{-1} maps every prefix-recognisable relation to a regular one. In other words, appropriately restricted, ρ_w^{-1} is the naming function of an automatic presentation of any prefix-recognisable structure over Σ, in particular of $\mathfrak{A}$.

Since ρ_w maps the non-regular subset $\{uv^{2^n}z \mid n \in \mathbb{N}\}$ of D onto a regular set, the automatic presentation of $\mathfrak{A}$ having ρ_w^{-1} as naming function is not equivalent to the natural presentation with the identity as naming function. $\square$

7 Definability and Intrinsic Regularity

Let $\mathfrak{A} = (A, \{R_i\}_i) \in$ AutStr and $\mathfrak{d}$ an injective automatic presentation of $\mathfrak{A}$ with naming function f. By definition f^{-1} maps every relation R_i of $\mathfrak{A}$ to a regular one. By the extension of the Fundamental Theorem to $\mathsf{FO}^{\infty,\mathrm{mod}}$ (Theorem 3.1.5 of Section 3.1.1) we know that f also maps all those relations to regular ones, which are $\mathsf{FO}^{\infty,\mathrm{mod}}$-definable in $\mathfrak{A}$. In other words, since in the case of automatic presentations injectivity can be assumed, $\mathsf{FO}^{\infty,\mathrm{mod}}$-definable relations are guaranteed to be regular in every automatic presentation. This property is captured by the notion of intrinsic regularity. Intrinsically regular relations of structures were introduced by Khoussainov, Rubin, and Stephan in [KRS03b, KRS04]. We shall also be concerned with the dual notion of intrinsic non-regularity.

Definition 7.0.4 (Intrinsic regularity)**.**
Let $\mathfrak{A}$ be automatic. The *intrinsically (non-)regular* relations of $\mathfrak{A}$ are those, which are (non-)regular in every automatic presentation of $\mathfrak{A}$. Formally,

$$\mathsf{IR}(\mathfrak{A}) = \{R \subseteq A^r \mid r \in \mathbb{N},\ \text{ for every a.p. } (\mathfrak{d}, f) \text{ of } \mathfrak{A} : f^{-1}(R) \text{is regular}\}$$

and dually

$$\mathsf{INR}(\mathfrak{A}) = \{R \subseteq A^r \mid r \in \mathbb{N},\ \text{ for every a.p. } (\mathfrak{d}, f) \text{ of } \mathfrak{A} : f^{-1}(R) \text{is } \textbf{not} \text{ regular}\}$$

Thus, by Theorem 3.1.5, we know that $\mathsf{FO}^{\infty,\mathrm{mod}}$-definable relations are intrinsically regular with respect to every automatic structure. One may ask how far can this extension be generalised.

Examples

Let Σ be a finite, non-unary alphabet. In Theorem 4.3.1 we have seen that all automatic presentations of the complete structure $\mathcal{S}_\Sigma$ are equivalent. This means that a relation is intrinsically (non-)regular with respect to $\mathcal{S}_Sigma$ if and only if it is (non-)regular.

Corollary 7.0.5 (of Theorem 4.3.1)**.** *Let Σ be a non-unary finite alphabet. Then* $\mathsf{IR}(\mathcal{S}_\Sigma)$ *is the set of regular relations over Σ, in particular,* $\mathsf{IR}(\mathcal{S}_\Sigma) = \mathsf{FO}(\mathcal{S}_\Sigma)$*. Moreover,* $\mathsf{INR}(\mathcal{S}_\Sigma)$ *is the set of non-regular relations over Σ.*

The circumstance that intrinsically non-regular relations complement those intrinsically regular over $\mathcal{S}_\Sigma$ is equivalent to saying that $\mathcal{S}_\Sigma$ is rigidly automatic.

Recall the theorem of Cobham and Semenov from Section 4.3.3. It states that if a relation $R \subseteq \mathbb{N}^r$ regular in, say, both the binary and the ternary representation of naturals then it is already Presburger definable.

Corollary 7.0.6 (of Cobham-Semenov theorem). $\mathsf{IR}(\mathbb{N},+) = \mathsf{FO}(\mathbb{N},+)$.

Clearly, $\mathsf{FO}^{\infty,\mathrm{mod}}$ collapses to FO over both $(\mathbb{N},+)$ and, by completeness, over $\mathcal{S}_\Sigma$ provided Σ is not unary. In the unary case $\mathcal{S}_{[1]}$ degenerates to $(\mathbb{N},\mathrm{succ},<)$ and is naturally represented over the unary alphabet [1]. As noted in Section 3.3.1 regular relations over a unary alphabet are precisely those first-order definable in $\mathfrak{M} = (\mathbb{N},<,\{\equiv_{(\mathrm{mod}\, m)}\}_{m>1})$, or equivalently, those $\mathsf{FO}^{\mathrm{mod}}$-definable in $(\mathbb{N},<)$. Hence the following.

Corollary 7.0.7 (of 3.3.2). $\mathsf{IR}(\mathbb{N},<) = \mathsf{FO}^{\mathrm{mod}}(\mathbb{N},<)$.

Note that establishing a result of the above type concerning definability of intrinsically regular relations is still a far cry from understanding the automatic presentations of a structure. While Corollary 7.0.7 can be proved based solely on the unary presentation of $(\mathbb{N},<)$ we have devoted the whole of Chapter 5 to the investigation of the multitude of presentations of $(\mathbb{N},<)$, which is far from being complete.

In [KRS03b, KRS04] Khoussainov et al. also consider the successor structure $(\mathbb{N},\mathrm{succ})$, which is even more elusive when it comes to understanding its presentations up to equivalence. With regard to intrinsic regularity, however, Khoussainov et al. have been able to show that the modulo counting predicates $M^k = \{kn \mid n \in Nat\}$ are not in $\mathsf{IR}(\mathbb{N},\mathrm{succ})$ and furthermore that the natural order $<$ is not IR even in the structure $(\mathbb{N},\mathrm{succ},\{M^k\}_k)$. On the positive side, they show that with respect to $(\mathbb{N},\mathrm{succ})$ all intrinsically regular unary predicates are FO-definable, for short: $\mathsf{IR}_1(\mathbb{N},\mathrm{succ}) = \mathsf{FO}_1(\mathbb{N},\mathrm{succ})$.

7.1 Logical Extensions

For any given logic $\mathcal{L}$ extending FO let $\mathcal{L}(\mathfrak{A})$ denote the set of relations over $\mathsf{dom}(\mathfrak{A})$ definable by an $\mathcal{L}$-formula using a finite number of parameters. Khoussainov et al. asked whether there is a logic $\mathcal{L}$ capturing intrinsic regularity, i.e., such that $\mathcal{L}(\mathfrak{A}) = \mathsf{IR}(\mathfrak{A})$ for all $\mathfrak{A} \in \textsc{AutStr}$. We address this question in the current section. We enumerate partial results known to us, both positive and negative, however, the list is non conclusive.

The extension of the Fundamental Theorem to $\mathsf{FO}^{\infty,\mathrm{mod}}$ (Theorem 3.1.5) essentially states that given a regular relation $R(\vec{x},y)$ of arity $n+1$ the n-ary relations defined by the quantifiers $\exists^\infty y\ R$ and $\exists^{(r,m)} y\ R$, with y ranging over all words, are again regular. [1] In particular, it holds over every automatic structure $\mathfrak{A}$, that $\mathsf{FO}^{\infty,\mathrm{mod}}(\mathfrak{A}) \subseteq \mathsf{IR}(\mathfrak{A})$. For this reason we shall call $\exists^\infty$ and the $\exists^{(r,m)}$ *regularity preserving quantifiers*. Of course, key to the extension of the Fundamental Theorem to $\mathsf{FO}^{\infty,\mathrm{mod}}$ is the fact that the additional quantifiers preserve regularity *effectively*.

[1] Note that injectivity is implicit in the current formulation considering distinct words as distinct elements. Also, in injective automatic presentations quantification ranges over a subset D of words, which can be assumed to be enforced by the relation R.

Over possibly uncountable ω-(tree-)automatic structures it makes sense to consider cardinality quantifiers refining $\exists^\infty$. In [KL06] Kuske and Lohrey have shown that the cardinality quantifiers $\exists^\omega x.\varphi$ and $\exists^{\omega_1} x.\varphi$ meaning that there are countably, respectively, uncountably many ω-words satisfying φ preserve ω-regularity. Note that injectivity is again implicitly understood.

Throughout this chapter when speaking of logical constructs preserving regularity we will always consider an interpretation over the set of all (ω-)words with distinct words seen as distinct elements and with (ω-)regular atomic relations given.

Consider a logic $\mathcal{L}$ extending FO and such that $\mathcal{L}(\mathfrak{A}) \subseteq \mathsf{IR}(\mathfrak{A})$ holds for all $\mathfrak{A} \in$ AutStr. Then all $\mathcal{L}$-formulas are by assumption regularity preserving. A priori this is no longer clear if we only assume that AutStr is closed under $\mathcal{L}$-interpretations, for it is conceivable that while a relation R $\mathcal{L}$-definable in $\mathfrak{A} \in$ AutStr is not intrinsically regular wrt. $\mathfrak{A}$ the combined structure $(\mathfrak{A}, R)$ does have some automatic presentation. Nonetheless, using Theorem 4.3.1 we can prove that $\mathcal{L}$-formulas have to be regularity preserving even under this weaker assumption.

Theorem 7.1.1. *For every logic $\mathcal{L}$ extending FO and such that AutStr is closed under $\mathcal{L}$-interpretations it holds that all $\mathcal{L}$-formulas are regularity preserving.*

Proof. Let R be a relation $\mathcal{L}$-definable from regular relations $\{R_i\}$ over the alphabet Σ. Since $\mathcal{L}$ is an extension of FO each R_i is $\mathcal{L}$-definable in $\mathcal{S}_\Sigma$. The combined structure $(\mathcal{S}_\Sigma, R)$ is thus $\mathcal{L}$-interpretable in $\mathcal{S}_\Sigma$ and therefore, by assumption, automatic. By Corollary 7.0.5 we conclude that R is indeed a regular relation over Σ. ☐

Note that completeness alone was not sufficient above, the argument crucially relies on the fact that all automatic presentations of $\mathcal{S}_{\{0,1\}}$ are equivalent as established in Theorem 4.3.1.

Motivated by the above we turn our attention to regularity-preserving extensions of first-order logic. Just how far can we push these extension results? Is there a maximal regularity preserving extension of first-order logic? Is there an extension capturing intrinsic regularity? In the following subsections we address these questions discussing regularity-preserving generalised quantifiers as well as an "orthogonal" extension of the logic based on the notion of order-invariance for which we give a separation result.

7.1.1 Generalised Quantifiers

Consider the syntactic extension $\mathsf{FO}[\mathcal{Q}]$ of first-order logic by allowing *generalised quantifiers* in the construction of formulas. This extension is thoroughly explained in [Lib04, Chapter 8] in the context of finite model theory. Here we merely give the definitions of basic concepts.

A generalised quantifier is defined in terms of an isomorphism-closed class $\mathcal{K}$ of structures of a fixed (possibly infinite) relational signature τ. We associate a quantifier $Q_{\mathcal{K}}$ to this class. The *arity* of the quantifier $Q_{\mathcal{K}}$ is the supremum of the arities of relations in the signature of $\mathcal{K}$. Formulae of the logic $\mathsf{FO}[\mathcal{Q}]$ are built using, in addition to classical first-order constructs, also quantifications of the following from.

$$\psi(\vec{z}) = Q_{\mathcal{K}}\big[\{\vec{x}_R\}_{R\in\tau}.\{\phi_R(\vec{x}_R,\vec{z})\}_{R\in\tau}\big]$$

Naturally, $\vec{x}_R$ is required to have the same dimension as R for each $R \in \tau$. The formula $\psi(\vec{z})$ is true in a structure $\mathfrak{A}$ with universe A for given elements $\vec{a}$ as values for $\vec{z}$ if the interpreted structure

$$\big(A, \{\phi_R^{\mathfrak{A}}(\vec{x}_R,\vec{a})\}_{R\in\tau}\big)$$

belongs to the class $\mathcal{K}$, where

$$\phi_R^{\mathfrak{A}}(\vec{x}_R,\vec{a}) = \{\vec{b}\vec{\in}A \mid \mathfrak{A} \models \phi_R(\vec{b},\vec{a})\}$$

for each $R \in \tau$.

Let $\mathcal{Q}$ denote the class of all generalised quantifiers and $\mathcal{Q}_n$ the class of those of arity n for each n. We say that a generalised quantifier Q is *regularity preserving* if all $\mathsf{FO}[Q]$-formulas are regularity preserving. We denote the class of regularity preserving quantifiers by $\mathcal{Q}^{\mathrm{reg}}$ and respectively by $\mathcal{Q}_n^{\mathrm{reg}}$ to stress that the arity is restricted to n.

Observe that the first-order quantifiers $\exists$, $\forall$ as well as the modulo counting quantifiers $\exists^{r,m}$ and also $\exists^{\infty}$ are particular unary generalised quantifiers.

Unary or counting quantifiers

A unary generalised quantifier $Q_{\mathcal{K}}$ is one defined in terms of an isomorphism-closed class $\mathcal{K}$ of structures comprising a fixed (possibly infinite) number of unary relations. Let $\mathsf{FO}[\mathcal{Q}_1]$ ($\mathsf{FO}[\mathcal{Q}_1^{\mathrm{reg}}]$) stand for the syntactic extensions of FO via (regularity preserving) unary generalised quantifiers. For an exposé to $\mathsf{FO}[\mathcal{Q}_1]$ and its involvement in finite model theory we refer to [Lib04, Chapter 8]. Next we characterise $\mathsf{FO}[\mathcal{Q}_1^{\mathrm{reg}}]$ following [Rub07].

As an alternative to the above definition we can consider *counting quantifiers* Q_K each associated to a class $K \subseteq \mathbf{Card}^{\alpha}$ of α-tuples of cardinals, for a fixed ordinal α. The intended meaning of a formula

$$Q_K[\{x_\beta\}_{\beta<\alpha}.\{\phi_\beta(x_\beta,\vec{z})\}_{\beta<\alpha}]$$

over a structure $\mathfrak{A}$ with universe A is that for a given value $\vec{a}$ of the variables $\vec{z}$ the α-sequence

$$(|\phi_\beta^{(\mathfrak{A},\vec{a})}|)_{\beta<\alpha}$$

belongs to the class K where $\phi_\beta^{(\mathfrak{A},\vec{a})} = \{c \in A \mid \mathfrak{A} \models \phi_\beta(c,\vec{a})\}$ for each $\beta < \alpha$.

Clearly, every unary generalised quantifier of arity α is equivalent to a counting quantifier of arity 2^{α} for as far as an isomorphism-closed class $\mathcal{K}$ of unary structures is concerned, only the cardinalities of boolean combinations of the unary predicates are relevant.

Over automatic structures, being countable, only those counting quantifiers associated to some $K \subseteq (\omega+1)^\alpha$ have to be dealt with. In the following we only consider counting quantifiers of finite arity $\alpha = n \in \mathbb{N}$.

Appealing to unary automatic presentations it is easy to show that if Q_K is regularity preserving, then K is $\mathsf{FO}^{\mathrm{mod}}$-definable in $(\mathbb{N}, <)$. Indeed, $[Q_K \vec{x}.(\psi(x_i, y_i))_{x_i \in \vec{x}}]$ with $\psi(x, y) = x < y$ defines K itself in the unary presentation. However, not all these quantifiers are regularity preserving. For instance, the unary Härtig quantifier associated to $H = \{(n, n)\}$ tests whether two formulas (with parameters) have the same number of satisfying elements. Using Q_H one can thus define the non-regular language $\{x \mid |x|_0 = |x|_1\}$ in $\mathcal{S}_{[2]}$. A similar argument shows that assuming Q_K is regularity preserving, the unary coding of K is in fact recognisable. This establishes the following characterisation.

Proposition 7.1.2 ([Rub07])**.** The only regularity preserving unary generalised quantifiers are those definable in terms of $\exists^{(r,m)}$. In other words, $\mathsf{FO}[\mathcal{Q}_1^{\mathrm{reg}}] = \mathsf{FO}^{\mathrm{mod}}$.

Bijective Ehrenfeucht-Fraïssé games

We briefly recall some of the notions and results of [Hel89], which we will be using. Chapter 8 of [Lib04] is a handy reference on this subject as well.

Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures sharing a common signature. The *r-round bijective Ehrenfeucht-Fraïssé game* $\mathrm{BEF}_r(\mathfrak{A}, \mathfrak{B})$ is defined as follows. There are two players: I and II. The positions of the game are partial isomorphisms between the two structures, provided there are any, the initial position being the empty isomorphism. In case $\emptyset$ is not a partial isomorphism, the game is won by I up front without any moves having been made. In each round of the game, in position p, player II proposes a bijection $f : A \to B$ such that $p \cup (a, f(a))$ is again a partial isomorphism for every $a \in \mathsf{dom}(\mathfrak{A})$, or loses. Player I replies by choosing an element $a \in \mathsf{dom}(\mathfrak{A})$, thus determining the new position as $p \cup (a, f(a))$ (that is to say II fixed her reply $f(a)$ in advance). The game ends after at most r rounds. Player II wins if she does not lose in the mean time.

A strategy of player II in this game is captured by an *r-bijective back-and-forth system* consisting of a sequence $(I_i)_{i \le r}$ of sets of partial isomorphisms between $\mathfrak{A}$ and $\mathfrak{B}$, such that $\emptyset \in I_r$ and for every $k < r$ and $p \in I_{k+1}$ there is a bijection $f_p : A \to B$ for which $p \cup \{(a, f(a))\} \in I_k$ for every $a \in A$.

Theorem 7.1.3 ([Hel89],[Lib04, Chapter 8])**.** *Two structures $\mathfrak{A}$ and $\mathfrak{B}$ satisfy the same sentences of $\mathsf{FO}[\mathcal{Q}_1]$ of quantifier rank at most r iff player II has a winning strategy in the game $\mathrm{BEF}_r(\mathfrak{A}, \mathfrak{B})$ iff there is an r-bijective back-and-forth system $(I_i)_{i \le r} : \mathfrak{A} \sim^r \mathfrak{B}$.*

7.1.2 Order-Invariant Formulas

Let $\mathfrak{A}$ be a structure of signature τ. Assume that $<$ is a binary relation symbol not occurring in τ. A formula $\phi(\vec{x}) \in \mathsf{FO}[\tau,<]$ is *order-invariant over* $\mathfrak{A}$ if for any linear ordering $<_A$ of the elements of $\mathfrak{A}$, when $<$ is interpreted as $<_A$, $\phi(\vec{x})$ defines the same relation R over $\mathfrak{A}$. The relation R is in this case *order-invariantly definable.* We denote the set of order-invariantly definable relations over $\mathfrak{A}$ by $\mathsf{FO}_{<-\mathsf{inv}}(\mathfrak{A})$. Although it is only appropriate to speak of order-invariantly definable relations, rather than of relations definable in "order-invariant logic", we will tacitly use the latter term as well.

Order-invariant first-order logic has played an important role in finite model theory. It is well known that $\mathsf{FO}_{<-\mathsf{inv}}$ is strictly more expressive than FO on finite structures. Gurevich was the first to exhibit an order-invariantly definable class of finite structures, which is not first-order definable [Lib04, Sect. 5.2]. However, his class is $\mathsf{FO}^{\infty,\mathrm{mod}}$-definable. In [Ott00] Otto showed how to use order-invariance to express connectivity, which is not definable even in infinitary counting logic, in a particular class of finite graphs. Both constructions use order-invariance and some auxiliary structure to exploit the power of monadic second-order logic. We adopt Otto's technique to show that $\mathsf{FO}_{<-\mathsf{inv}}$ can be strictly more expressive than infinitary counting logic on automatic structures.

The fact that over any (ω-)(tree-)automatic $\mathfrak{A}$ order-invariantly definable relations are intrinsically regular is obvious. Indeed, given a particular automatic presentation of $\mathfrak{A}$ one just has to "plug in" any regular ordering (e.g. the lexicographic ordering, which does of course depend on the automatic presentation chosen) into the order-invariant formula defining a particular relation, thereby yielding a regular relation, which, by order-invariance, will always represent the same relation.

Observation 7.1.4. $\mathsf{FO}^{\infty,\mathrm{mod}}_{<-\mathsf{inv}}(\mathfrak{A}) \subseteq \mathsf{IR}(\mathfrak{A})$

7.1.3 Separating Example

In this subsection we present an automatic structure in which a certain relation is order-invariantly definable, but not by using only unary generalised quantifiers. The example is based on that of Otto [Ott00] and uses Hella's characterisation of expressibility of unary generalised quantifiers as summarised above in Theorem 7.1.3.

Consider the structure

$$\mathfrak{B} = (\mathbb{N} \uplus \mathcal{P}_{\mathrm{fin}}(4\mathbb{N} + \{2,3\}), S, \varepsilon, \iota, \subseteq)$$

illustrated in Figure 7.1, where $\mathcal{P}_{\mathrm{fin}}(H)$ consists of the finite subsets of H, S is the relation $\{(4n, 4n+4), (4n+1, 4n+5) \mid n \in \mathbb{N}\}$, ε is the equivalence relation consisting of classes $\{4n, 4n+1, 4n+2, 4n+3\}$ for each $n \in \mathbb{N}$, ι is the set of pairs $(n, \{n\})$ with $n \in 4\mathbb{N} + \{2,3\}$, and $\subseteq$ is the usual subset inclusion.

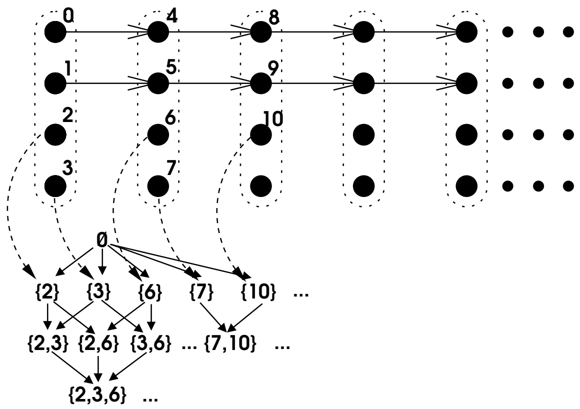


Figure 7.1: $\mathfrak{B}$, a separating example.

To give an automatic presentation of $\mathfrak{B}$ over the alphabet $\{b, 0, 1\}$ we represent $(\mathbb{N}, S, \varepsilon)$ in the unary encoding using the symbol b, and the finite sets by their (shortest) characteristic words over $\{0, 1\}$. Regularity of ι and $\subseteq$ is obvious.

Proposition 7.1.5 ([Bár06b]). The transitive closure $S^\star$ of S is order-invariantly definable, hence intrinsically regular, but not $\mathsf{FO}[\mathcal{Q}_1]$-definable in $\mathfrak{B}$.

Proof. The proof is an adaptation of the one presented in [Ott00].
$S^\star \in \mathsf{FO}_{<-\mathsf{inv}}(\mathfrak{B})$: Given any ordering $\prec$ of the universe of $\mathfrak{B}$ we can first-order define a bijection $\nu = \nu_\prec : 4\mathbb{N} \cup 4\mathbb{N}+1 \to 4\mathbb{N}+2 \cup 4\mathbb{N}+3$ as follows. Each ε-class contains two isolated points $4n+2$ and $4n+3$ and two points $4n$ and $4n+1$ having an S-successor for some n. Using $\prec$ we can map e.g. the smaller (larger) of the latter to the smaller (larger) of the former. This bijection, regardless of the actual mapping, provides access to the subset structure. Take any wMSO formula defining transitive closure and translate it using ν and the built-in subset structure to express $S^\star$.
$S^\star \notin \mathsf{FO}[\mathcal{Q}_1](\mathfrak{B})$: Let $\mathfrak{B}_n = (\mathfrak{B}, 0, 4n)$ and $\mathfrak{B}'_n = (\mathfrak{B}, 0, 4n+1)$. It is sufficient to show that for large enough n player II wins $\mathrm{BEF}_r(\mathfrak{B}_n, \mathfrak{B}'_n)$.

Let $B = \mathsf{dom}(\mathfrak{B})$ and $D = \mathsf{dom}(S)$. Considering the reducts $\mathfrak{S}_n = \mathfrak{B}_n|_D$ and $\mathfrak{S}'_n = \mathfrak{B}'_n|_D$ it should be clear that player II has a winning strategy in the r-round bijective game $BEF_r(\mathfrak{S}_n, \mathfrak{S}'_n)$ for some $n \in 2^{O(r)}$. Moreover, there is an r-bijective back-and-forth system $(I_i)_{i \leq r} : \mathfrak{S}_n \sim^r \mathfrak{S}'_n$, such that for every $k \leq r$ each $p \in I_k$ maps $\{4m, 4m+1\}$ into itself for every m (*), i.e. ε-classes are preserved throughout any play consistent with this strategy.

We extend this strategy to one in $\mathrm{BEF}_r(\mathfrak{B}_n, \mathfrak{B}'_n)$ by extending the bijections given by the former strategy identically onto all elements outside of the domain of S. Equivalently, we claim that $(J_i)_{i \leq r} : \mathfrak{B}_n \sim^r \mathfrak{B}'_n$, where $J_k = \{p \cup q \mid p \in I_k, q \subset \mathsf{id}|_{B \setminus D}\}$ for each $k \leq r$. Each such $p \cup q$ is indeed a partial isomorphism, because both p and q are on the respective "halves" of the structures and p also satisfies (*). Further, for any $p \cup q \in J_{k+1}$, thus $p \in I_{k+1}$, there is by definition a bijection $f_p : D \to D$ such that $p \cup (a, f_p(a))$ is in I_k for any $a \in D$. Hence, with

$g_p = f_p \cup \mathsf{id}|_{B \setminus D}$ it holds that $p \cup q \cup (a, g_p(a)) \in J_k$ for any $a \in B$. This concludes the proof. □

Corollary 7.1.6. *No extension of* FO *with unary generalised quantifiers is capable of capturing intrinsic regularity over all automatic structures.*

7.1.4 The hierarchy of regularity preserving quantifiers

Motivated by the examples of modulo counting quantifiers we have introduced regularity preserving generalised quantifiers. We have seen that the modulo counting quantifiers suffice to define the every regularity preserving unary quantifier. We have also introduced order-invariant extension of first-order logic, argued that it is regularity preserving and showed that there are order-invariantly definable relations that are not definable by unary quantifiers.

In [Rub07] Rubin gives the following (effectively) regularity-preserving quantifiers of arity two.

- The Ramsey quantifier $\exists^{\mathcal{R}} xy.\varphi(x, y, \vec{z})$ expresses that the (undirected) graph defined by φ parameterised by $\vec{z}$ contains an infinite clique.
- The quantifier $\exists^{\rho} xy.\varphi(x, y, \vec{z})$ expressing that with parameters $\vec{z}$ the graph defined by φ is an equivalence relation having infinitely many infinite equivalence classes (cliques).

Observe that the latter quantifier is definable in $\mathsf{FO}^{\infty}_{<-\mathsf{inv}}$, and is thus effectively regularity preserving. However, it is not definable in $\mathsf{FO}^{\infty,\mathrm{mod}}$ as can be checked by an argument involving bijective Ehrenfeucht-Fraïssé games as in Proposition 7.1.5. To show that the Ramsey quantifier is regularity preserving requires a bit more effort, in [Rub07] Rubin makes a detour through an ω-regular representation to conclude.

As it turns out a generalisation of the bijective Ehrenfeucht-Fraïssé games along the lines of [Hel89, Hel96] can be used to prove separation results of the $\mathsf{FO}[\mathcal{Q}^{\mathrm{reg}}_n]$-definability classes analogous to our argument that $\exists^{\rho}$ is not $\mathsf{FO}[\mathcal{Q}^{\mathrm{reg}}_1]$-definable. Thus establishing that the logics $\mathsf{FO}[\mathcal{Q}^{\mathrm{reg}}_n]$ form an infinite hierarchy in terms of expressiveness [Rub07].

Proposition 7.1.7 (In [Rub07] contributed to Hella, cf. also [Hel96])**.** For every n there is a quantifier $Q \in \mathcal{Q}^{\mathrm{reg}}_{n+1}$, which is definable in the ω-order-invariant fragment of $\mathsf{FO}[\exists^{(0,2)}]$ but not definable in $\mathsf{FO}[\mathcal{Q}^{\mathrm{reg}}_n]$.

In light of these results it is natural to ask whether $\mathsf{FO}_{<-\mathsf{inv}}[\mathcal{Q}^{\mathrm{reg}}]$ captures intrinsic regularity. We have to note that even a positive answer would not be satisfactory as order-invariance is a semantic property that is undecidable even over finite structures. $\mathsf{FO}_{<-\mathsf{inv}}[\mathcal{Q}^{\mathrm{reg}}]$ is therefore *not* a logic in the sense that it does not come with a recursive syntax. On the other hand, if it is not even semi-decidable whether a given regular relation is intrinsically regular with respect to a structure given by an automatic presentation then there is no logic of recursive syntax and capturing intrinsic regularity. These questions are left open.

7.2 More examples, remarks and questions

7.2.1 Structures of Bounded Degree

A relational structure is said to be of *bounded degree* if its Gaifman graph is of bounded degree. The *Gaifman graph* $G(\mathfrak{A})$ of a structure $\mathfrak{A}$ consists of the elements of the universe of $\mathfrak{A}$ as its vertices, and with an (undirected) edge defined between any two elements $a, b \in \mathfrak{A}$ iff there is a tuple $(a_1, \ldots, a_n)$ of elements in some relation $R^{\mathfrak{A}}$ of $\mathfrak{A}$ such that $a = a_i$ and $b = b_j$ for some $1 \leq i, j \leq n$. Observe that $G(\mathfrak{A})$ is first-order definable in $\mathfrak{A}$. The distance $d(x, y)$ of two elements x, y of a structure is defined as their distance in the Gaifman graph, i.e. as the length of the shortest path leading from x to y, or infinite if they are not connected. The d-neighbourhood of an element x is the substructure induced by those elements having a distance at most d from x. Note that k quantifiers suffice to express in first-order logic that $d(x, y) \leq 2^k$. Conversely, the locality theorems of Hanf and Gaifman tell us, intuitively, that first-order logic can only express a boolean combination of statements depending on the existence of neighbourhoods of a certain type. For a precise formulation the reader should consult either one of the books [EF95, Lib04].

Locality is a particularly useful concept when studying structures of bounded degree. In [Loh03], using locality as a key ingredient, Lohrey gave an elementary bound on the complexity of first-order theories of automatic structures of bounded degree. Locality also plays a vital role in the result of Carayol and Morvan establishing decidability of first-order theories of rational trees [CM06].

Let us note that both $\mathsf{FO}^{\infty,\mathrm{mod}}$ and $\mathsf{FO}_{<-\mathsf{inv}}$ have this locality property (cf. [Lib04]), but $\mathsf{FO}^{\infty,\mathrm{mod}}_{<-\mathsf{inv}}$ does *not*. On inquiry of the author Hannu Niemistö gave a simple example of a non-local property over a class of finite structures of bounded degree expressible in order-invariant $\mathsf{FO} + \exists^{(0,2)}$ [Nie]. However, to adapt Niemistö's example to infinite automatic structures would require the introduction of additional predicates causing the Gaifman graph to be of unbounded degree. Therefore we pose the following question.

Question 7.2.1. *Consider an arbitrary* $\mathfrak{A} \in \textsc{AutStr}$ *of bounded degree. Is every intrinsically regular relation over* $\mathfrak{A}$ *Gaifman local?*

Note that over structures of bounded degree, Gaifman locality means that there is a constant d such that membership of a tuple $\vec{a}$ in R solely depends on the isomorphism type of the d-neighbourhood of $\vec{a}$. Gaifman local relations are thus first-order definable over structures of bounded degree. So the above question asks whether FO suffices to capture IR on automatic structures of bounded degree.

7.2.2 Prefix-ordered trees

Consider an automatic presentation of $\mathfrak{T}_\Sigma = (\Sigma^*, \{\mathrm{succ}_a\}_{a \in \Sigma}, \preceq)$, that is of the prefix-ordered infinite Σ-branching tree. If the presentation preserves regularity of the equal-length relation, e.g. if it preserves length, then we are talking a presentation of $\mathcal{S}_\Sigma$, which is, by Theorem 4.3.1, equivalent to the natural one having the identity naming function. Therefore, we need to consider presentations which "compress" some words while "stretching" others in a non-trivial manner so as not to let equal-length be regular. How can this be achieved?

Recall the generalised run-length encoding ρ_w associated to an automatic presentation of an ω-word $w \in \Sigma^\omega$. In Section 6.4 we have shown how automaticity of w implies that ρ_w preserves regularity of the prefix relation and of the successors succ_a for every $a \in \Sigma$. In other words, that ρ_w determines (is the naming function of) an automatic presentation of $\mathfrak{T}_\Sigma$.

Unless the underlying presentation $\mathfrak{d}$ of w is unary, ρ_w does have the desired property of compressing some branches of $\mathfrak{T}_\Sigma$ while stretching others in a non-regular fashion. For instance, when as underlying presentation of a^ω the binary numeration system is chosen, then $\rho_{a^\omega}(a^{2^n}) = 10^n$ while $\rho_{a^\omega}(b^{2^n}) = b^{2^n}$, whence it is routine to prove that $\rho_{a^\omega}(\mathrm{el})$ is not regular.

According to Corollary 6.4.2 if $\mathfrak{d}$ is an MSO-friendly presentation of some word w, e.g. a k-lexicographic one, then the associated run-length compression scheme ρ_w is regularity preserving. In other words, ρ_w^{-1} is continuous, hence, by Theorem 6.2.3, ρ_w maps every prefix-recognisable relation over Σ to a regular relation. We used this fact in Theorem 6.4.3 to show that ρ_w^{-1} can be used to give an automatic presentation of any prefix-recognisable structure.

These considerations prompt us to generalise the notion of MSO-friendly presentations from word structures to prefix-ordered trees. Note that by Theorem 6.2.3 it is sufficient to require MSO-definable sets to be regularly represented. Hence the following definition (cf. Definition 5.2.1).

Definition 7.2.2. An automatic presentation $\mathfrak{d}$ of a tree $\mathfrak{T} = (T, \{succ_i\}_{i<r}, \preceq)$ is *MSO-friendly* if for every MSO formula $\varphi(x)$ (equivalently regular language L) the set of nodes of $\mathfrak{T}$ satisfying φ (equivalently, which are lead to from the root on a path labelled by a word belonging to L) is represented on a regular set. In short, if all MSO-definable sets of nodes are regularly represented.

Question 7.2.3. *Is every* MSO*-definable relation in $\mathfrak{T}_\Sigma$ intrinsically regular with respect to $\mathfrak{T}_\Sigma$?*

Note that by Theorem 6.2.3 it is sufficient to confirm this for MSO-definable, i.e. regular, sets. We remark that those sets $\mathsf{FO}^{\infty,\mathrm{mod}}$-definable, equivalently, recognised by a solvable monoid [Str94], are bound to be intrinsically regular. A counterexample would therefore have to be based on a non-solvable monoid. A positive answer to Question 7.2.3, on the other hand, would also answer Question 5.9.2(1) in the affirmative, and would drive the quest for a logical characterisation of intrinsic regularity over all automatic structures in a new direction.

From trees to branches

Automatically presentable prefix-ordered trees were first studied by Khoussainov, Rubin and Stephan in [KRS03a, KRS05] from a structural point of view using model-theoretic notions of condensation rank and Cantor-Bendixon rank. They have shown that automatic linear orderings have a finite condensation rank, and similarly, that the Cantor-Bendixon rank of every automatic prefix-ordered tree is finite. Concerning regularly represented paths, Khoussainov et al. show the following, among them an automatic version of Kőnig's Lemma.

Theorem 7.2.4 ([KRS05]).
In every automatic presentation of a prefix-ordered tree $\mathfrak{T}$ having at least one infinite branch there exists an infinite branch, which is regularly represented.
Moreover, if $\mathfrak{T}$ has only countably many infinite paths, then every infinite path is regularly presented in every automatic presentation of $\mathfrak{T}$.

These results are obtained by showing that under the stated assumptions an infinite branch, respectively, every infinite branch is FO^{∞}-definable. For the latter, one makes use of the finiteness of the Cantor-Bendixon rank of the tree.

Given an automatic presentation of some word $w \in \Sigma^\omega$, in the presentation of $\mathfrak{T}_\Sigma$ having the associated ρ_w as naming function the infinite branch corresponding to w is regularly represented. And if the presentation of w is MSO-friendly, then so is the associated presentation of $\mathfrak{T}_\Sigma$. Conversely, automatic presentations of $\mathfrak{T}_\Sigma$ induce automatic presentations of regularly presented branches. That is, if the set of nodes of a branch π of $\mathfrak{T}_\Sigma$ is represented by a regular language P in some automatic presentation of $\mathfrak{T}_\Sigma$ then $(P, \preceq, \{S_a\}_{a\in\Sigma})$ is an automatic presentation of $\pi \in \Sigma^\omega$ where S_a is defined as the set of nodes on π with an incoming a-edge. If a presentation of $\mathfrak{T}_\Sigma$ is MSO-friendly, then so are the presentations of its regular paths thus obtained.

Bibliography

[AdMO05] K. Aehlig, J. G. de Miranda, and C.-H. L. Ong. Safety is not a restriction at level 2 for string languages. In *FoSSaCS*, pages 490–504, 2005.

[AG89] J.-M. Autebert and J. Gabarró. Iterated GSMs and Co-CFL. *Acta Informatica*, 26:749–769, 1989.

[AS99] J.-P. Allouche and J. O. Shallit. The Ubiquitous Prouhet-Thue-Morse Sequence. In C. Ding, T. Helleseth, and H. Niederreiter, editors, *Sequences and Their Applications: Proceedings of SETA '98*, pages 1–16. Springer-Verlag, 1999.

[AS03] J.-P. Allouche and J. Shallit. *Automatic Sequences, Theory, Applications, Generalizations.* Cambridge University Press, 2003.

[Bar98] K. Barthelmann. When Can an Equational Simple Graph Be Generated by Hyperedge Replacement? In *MFCS*, pages 543–552, 1998.

[Bár06a] V. Bárány. A hierarchy of automatic ω-words having a decidable MSO theory. Journées Montoises '06, Rennes, 2006.

[Bár06b] V. Bárány. Invariants of automatic presentations and semi-synchronous transductions. In *STACS '06*, volume 3884 of *LNCS*, pages 289–300, 2006.

[Ber79] J. Berstel. *Transductions and Context-Free Languages.* Teubner, Stuttgart, 1979.

[Bés00] A. Bés. An Extension of the Cobham-Semënov Theorem. *J. of Symb. Logic*, 65(1):201–211, 2000.

[BG00] A. Blumensath and E. Grädel. Automatic structures. In *Proceedings of 15th IEEE Symposium on Logic in Computer Science LICS 2000*, pages 51–62, 2000.

[BG04] A. Blumensath and E. Grädel. Finite presentations of infinite structures: Automata and interpretations. *Theory of Comp. Sys.*, 37:641 – 674, 2004.

[BHMV94] V. Bruyère, G. Hansel, Ch. Michaux, and R. Villemaire. Logic and p-recognizable sets of integers. *Bull. Belg. Math. Soc.*, 1:191 – 238, 1994.

[BKR07] V. Bárány, Ł. Kaiser, and S. Rubin. Cardinality and counting quantifiers on ω-automatic structures. Manuscript, 2007.

[BLS06] V. Bárány, Ch. Löding, and O. Serre. Regularity problems for visibly pushdown languages. In *STACS '06*, volume 3884 of *LNCS*, pages 420–431, 2006.

[BLSS01] M. Benedikt, L. Libkin, Th. Schwentick, and L. Segoufin. A model-theoretic approach to regular string relations. In Joseph Halpern, editor, *Proceedings of the Sixteenth Annual IEEE Symp. on Logic in Computer Science, LICS 2001*, pages 431–440. IEEE Computer Society Press, June 2001.

[BLSS03] M. Benedikt, L. Libkin, Th. Schwentick, and L. Segoufin. Definable relations and first-order query languages over strings. *J. ACM*, 50(5):694–751, 2003.

[Blu99] A. Blumensath. Automatic Structures. Diploma thesis, RWTH-Aachen, 1999.

[Blu01] A. Blumensath. Prefix-Recognisable Graphs and Monadic Second-Order Logic. Technical report AIB-2001-06, RWTH Aachen, 2001.

[Blu02] A. Blumensath. Axiomatising Tree-interpretable Structures. In *STACS*, volume 2285 of *LNCS*, pages 596–607. Springer-Verlag, 2002.

[BP97] M.-P. Béal and D. Perrin. Symbolic Dynamics and Finite Automata. In A. Salomaa and G. Rosenberg, editors, *Handbook of Formal Languages, Vol. 2*, pages 463–503. Springer Verlag, 1997.

[Bra06] L. Braud. Higher-order schemes and morphic words. Journées Montoises, Rennes, 2006.

[Büc60] J. R. Büchi. Weak second-order arithmetic and finite automata. *Zeit. Math. Logih Grund. Math.*, 6:66–92, 1960.

[Car06] O. Carton. The growth ratio of synchronous rational relations is unique. In *Developments in Language Theory*, volume 4036 of *LNCS*, pages 270–279, 2006.

[Cau92a] D. Caucal. Monadic theory of term rewritings. In *LICS*, pages 266–273. IEEE Computer Society, 1992.

[Cau92b] D. Caucal. On the regular structure of prefix rewriting. *Theor. Comput. Sci.*, 106(1):61–86, 1992.

[Cau96] D. Caucal. On infinite transition graphs having a decidable monadic theory. In *ICALP'96*, volume 1099 of *LNCS*, pages 194–205, 1996.

[Cau02] D. Caucal. On infinite terms having a decidable monadic theory. In *MFCS*, pages 165–176, 2002.

[Cau03] D. Caucal. On infinite transition graphs having a decidable monadic theory. *Theor. Comput. Sci.*, 290(1):79–115, 2003.

[CC03] A. Carayol and Th. Colcombet. On equivalent representations of infinite structures. In *ICALP*, volume 2719 of *LNCS*, pages 599–610. Springer, 2003.

[CDG+] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree Automata Techniques and Applications. In preparation, draft available online at http://www.grappa.univ-lille3.fr/tata/.

[CEH+92] J.W. Cannon, D.B.A. Epstein, D.F. Holt, S.V.F. Levy, M.S. Paterson, and W.P. Thurston. *Word processing in groups*. Jones and Barlett Publ., Boston, MA, 1992.

[Cho02] Ch. Choffrut. A short introduction to automatic group theory, 2002.

[CK02] B. Courcelle and T. Knapik. The evaluation of first-order substitution is monadic second-order compatible. *TCS*, 281:177–206, 2002. Special issue offered to Maurice Nivat.

[CL06] Th. Colcombet and Ch. Löding. Transforming structures by set interpretations. Technical Report AIB-2006-07, RWTH Aachen, 2006.

[CL07] A. Caryol and Ch. Löding. MSO on the Infinite Binary Tree: Choice and Order. Submitted to CSL'07., 2007.

[CM02] B. Courcelle and J. A. Makowsky. Fusion in Relational Structures and the Verification of Monadic Second-Order Properties. *Mathematical Structures in Computer Science*, 12(2):203–235, 2002.

[CM05] A. Carayol and A. Meyer. Context-sensitive languages, rational graphs and determinism, 2005.

[CM06] A. Carayol and C. Morvan. On rational trees. In Z. Ésik, editor, *CSL 06*, volume 4207 of *LNCS*, pages 225–239, 2006.

[Col02] Th. Colcombet. On families of graphs having a decidable first order theory with reachability. In *ICALP*, volume 2380 of *LNCS*, pages 98–109. Springer, 2002.

[Col04a] Th. Colcombet. Equational presentations of tree-automatic structures. In Workshop on Automata, Structures and Logic, Auckland, NZ, 2004.

[Col04b] Th. Colcombet. *Propriétés et représentation de structures infinies.* Thèse de doctorat, PhD Thesis, Université Rennes I, 2004.

[Col07a] Th. Colcombet. A combinatorial theorem for trees. In *ICALP*, 2007.

[Col07b] Th. Colcombet. On factorisation forests and some applications. manuscript, arXiv:cs.LO/0701113v1, Jan. 2007.

[Cou90a] B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics*, pages 193–242. Elsevier and MIT Press, 1990.

[Cou90b] B. Courcelle. Recursive applicative program schemes. In J. v.d. Leeuwen, editor, *Handbook of Theoretical Computer Science, Vol. B*, pages 459–492. Elsevier, 1990.

[Cou94] B. Courcelle. Monadic secon-order definable graph transduction: a survey. *Theoretical Computer Science*, 126:53–75, 1994.

[CRRT01] C. M. Campbell, E. F. Robertson, N. Ruskuc, and R. M. Thomas. Automatic semigroups. *Theor. Comput. Sci.*, 250(1-2):365–391, (2001).

[CT02] O. Carton and W. Thomas. The monadic theory of morphic infinite words and generalizations. *Information and Computation*, 176(1):51–65, 2002.

[CW98] B. Courcelle and I. Walukiewicz. Monadic second-order logic, graph coverings and unfoldings of transition systems. *Annals of Pure and Applied Logic*, 92:35–62, 1998.

[CW03] A. Carayol and S. Wöhrle. The Caucal hierarchy of infinite graphs in terms of logic and higher-order pushdown automata. In *FSTTCS*, volume 2914 of *LNCS*, pages 112–123. Springer, 2003.

[Dam82] W. Damm. The IO- and OI hierarchies. *Theoretical Computer Science*, 20(2):95–208, 1982.

[DGK] Ch. Delhomme, V. Goranko, and T. Knapik. Automatic linear orderings. Unpublished.

[DT90] M. Dauchet and S. Tison. The theory of ground rewrite systems is decidable. In *LICS'90*, pages 242–248. IEEE, 1990.

[EF95] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Perspectives in Mathematical Logic. Springer, 1995.

[EH99] J. Engelfriet and H. J. Hoogeboom. Two-way finite-state transducers and monadic second-order logic. In *ICALP*, pages 311–320, 1999.

[EH01] J. Engelfriet and H. J. Hoogeboom. MSO definable string transductions and two-way finite-state transducers. *ACM Transactions on Computational Logic*, 2(2):216–254, 2001.

[Elg61] C.C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Amer. Math. Soc.*, 98:21–51, 1961.

[EM65] C. C. Elgot and J. E. Mezei. On relations defined by generalized finite automata. *IBM J. Research and Development*, 9:47 – 68, 1965.

[Eng97] J. Engelfriet. Context-free graph grammars. In *Handbook of formal languages, vol. III*, pages 125–213. Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[ER66] C. C. Elgot and M. O. Rabin. Decidability and undecidability of extensions of second (first) order theory of (generalized) successor. *Journal of Symbolic Logic*, 31(2):169–181, 1966.

[Far92] B. Farb. Automatic Groups: A Guided Tour. *L'Enseignment Math.*, 38:291–313, 1992.

[Fro02] Ch. Frougny. Numeration systems. In M. Lothaire, editor, *Algebraic Combinatorics on Words*. Cambridge University Press, 2002.

[FS93] Ch. Frougny and J. Sakarovitch. Synchronized rational relations of finite and infinite words. *Theoretical Computer Science*, 108:45–82, 1993.

[FV98] Z. Fülöp and H. Vogler. *Syntax-Directed Semantics: Formal Models Based on Tree Transducers*. Springer-Verlag Berlin Heidelberg, 1998.

[GKL+07] E. Grädel, P.G. Kolaitis, L. Libkin, M. Marx, Spencer J.H, M.Y. Vardi, Y. Venema, and S. Weinstein. *Finite Model Theory and Its Applications*. Number XIII in Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, 2007.

[Grä90] E. Grädel. Simple interpretations among complicated theories. *Information Processing Letters*, 35:235–238, 1990.

[GS84] F. Gécseg and M. Steinby. *Tree Automata*. Akadémiai kiadó, Budapest, 1984.

[GS97] F. Gécseg and M. Steinby. Tree languages. In *Handbook of formal languages, volume III*, pages 1–68. Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[GTW02] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games*, volume 2500 of *LNCS*. Springer-Verlag, 2002.

[Gur85] Y. Gurevich. Monadic Second-Order Theories. In J. Barwise and S. Feferman, editors, *Model-Theoretic Logics*, Perspectives in Mathematical Logic, pages 479–506. Springer-Verlag, 1985.

[Hel89] L. Hella. Definability hierarchies of generalized quantifiers. *Ann. Pure Appl. Logic*, 43:235 – 271, 1989.

[Hel96] L. Hella. Logical hierarchies in PTIME. *Information and Computation*, 129(1):1–19, 1996.

[HKOT02] M. Hoffmann, D. Kuske, F. Otto, and R. M. Thomas. Some relatives of automatic and hyperbolic groups, 2002.

[HMOS] M. Hague, A.S. Murawski, C.-H. L. Ong, and O. Serre. Collapsible pushdown automata and recursion schemes. (extended abstract), Nov. 2006.

[Hod82] B.R. Hodgson. On direct products of automaton decidable theories. *Theoretical Computer Science*, 19:331–335, 1982.

[Hod83] B.R. Hodgson. Décidabilité par automate fini. *Ann. sc. math. Québec*, 7(1):39–57, 1983.

[HR04] J. Honkala and M. Rigo. A note on decidability questions related to abstract numeration systems. *Discrete Math.*, 285:329–333, 2004.

[Imm99] N. Immerman. *Descriptive Complexity*. Graduate Texts in Computer Science. Springer, 1999.

[KL06] D. Kuske and M. Lohrey. First-order and counting theories of ω-automatic structures. In *FoSSaCS*, pages 322–336, 2006.

[KLP00] G. Kuper, L. Libkin, and J. Paredaens, editors. *Constraint Databases*. Springer-Verlag, 2000.

[KN95] B. Khoussainov and A. Nerode. Automatic presentations of structures. In *LCC '94*, volume 960 of *LNCS*, pages 367–392. Springer-Verlag, 1995.

[KNRS04] B. Khoussainov, A. Nies, S. Rubin, and F. Stephan. Automatic structures: Richness and limitations. In *LICS*, pages 44–53, 2004.

[KNU02] T. Knapik, D. Niwinski, and P. Urzyczyn. Higher-order pushdown trees are easy. In *FoSSaCS'02*, volume 2303 of *LNCS*, pages 205–222, 2002.

[KR01] B. Khoussainov and S. Rubin. Graphs with automatic presentations over a unary alphabet. *Journal of Automata, Languages and Combinatorics*, 6(4):467–480, 2001.

[KR03] B. Khoussainov and S. Rubin. Automatic structures: Overview and future directions. *J. Aut., Lang. and Comb.*, 8(2):287–301, 2003.

[KRS03a] B. Khoussainov, S. Rubin, and F. Stephan. Automatic linear orders and trees (revised). Technical Report CDMTCS-208, Univ. Auckland, NZ, 2003.

[KRS03b] B. Khoussainov, S. Rubin, and F. Stephan. Definability and regularity in automatic presentations of subsystems of arithmetic. Technical Report CDMTCS-209, Univ. Auckland, NZ, 2003.

[KRS03c] B. Khoussainov, S. Rubin, and F. Stephan. On automatic partial orders. In *LICS*, pages 168–177, 2003.

[KRS04] B. Khoussainov, S. Rubin, and F. Stephan. Definability and regularity in automatic structures. In *STACS '04*, volume 2996 of *LNCS*, pages 440–451, 2004.

[KRS05] B. Khoussainov, S. Rubin, and F. Stephan. Automatic linear orders and trees. *ACM Transactions on Computational Logic*, 6(4):675–700, 2005.

[Kus03] D. Kuske. Is cantor's theorem automatic? In *LPAR*, volume 2850 of *LNCS*, pages 332–345. Springer, 2003.

[Lib04] L. Libkin. *Elements of Finite Model Theory.* Texts in Theoretical Computer Science. Springer, 2004.

[Löd03] Ch. Löding. *Infinite Graphs Generated by Tree Rewriting.* Doctoral thesis, RWTH Aachen, 2003.

[Loh03] M. Lohrey. Automatic structures of bounded degree. In *LPAR*, volume 2850 of *LNCS*, pages 346–360. Springer, 2003.

[Lot83] M. Lothaire. Combinatorics on Words. Vol. 17 of Encyclopedia of Mathematics, Addison-Wesley. Reprinted in the Cambridge Mathematical Library, Cambridge University Press, 1997., 1983.

[Lot02] M. Lothaire. *Algebraic Combinatorics on Words.* Cambridge University Press, 2002.

[LS87] H. Lauchli and Ch. Savioz. Monadic Second Order Definable Relations on the Binary Tree. *J. of Symbolic Logic*, 52(1):219–226, 1987.

[Mae99] A. Maes. An automata theoretic decidability proof for first-order theory of $\langle \mathbb{N}, <, P\rangle$ with morphic predicate P. *J. Automata, Languages and Comb.*, 4:229–245, 1999.

[May98] R. Mayr. *Decidability and Complexity of Model Checking Problems for Infinite-State Systems.* Doctoral thesis, TU Mïchen, 1998.

[May00] R. Mayr. Process rewrite systems. *Information and Computation*, 156(1-2):264–286, 2000.

[Mey05] A. Meyer. *Finitely presented infinite graphs.* Thèse de doctorat, Université Rennes I, 2005.

[MN80] H. A. Mauer and M. Nivat. Rational bijection of rational sets. *Acta Informatica*, 13:365–378, 1980.

[Mor01] Ch. Morvan. *Les graphes rationnels.* Thèse de doctorat, Université de Rennes 1, Novembre 2001.

[Mor06] Ch. Morvan. Classes of rational graphs. Journées Montoises '06, Rennes, 2006.

[MR05] Ch. Morvan and Ch. Rispal. Families of automata characterizing context-sensitive languages. *Acta Informatica*, 41(4-5):293–314, 2005.

[MS83] D. E. Muller and P. E. Schupp. Groups, the theory of ends, and context-free languages. *J. Comput. Syst. Sci.*, 26(3):295–310, 1983.

[MS85] D. E. Muller and P. E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theor. Comput. Sci.*, 37:51–75, 1985.

[MS01] Ch. Morvan and C. Stirling. Rational graphs trace context-sensitive languages. In A. Pultr and J. Sgall, editors, *MFCS 01*, volume 2136 of *LNCS*, pages 548–559, 2001.

[Muc03] A. A. Muchnik. The definable criterion for definability in Presburger arithmetic and its applications. *Theor. Comput. Sci.*, 290(3):1433–1444, 2003.

[Nab77] A. A. Nabebin. Expressibility in a restricted second-order arithmetic. *Siberian Mathematical Journal*, 18(4):588–593, 1977.

[Nie] H. Niemistö. Private communication.

[Ong06] C.-H. L. Ong. On model-checking trees generated by higher-order recursion schemes. In *LICS*, pages 81–90. IEEE Computer Society, 2006.

[OT05] G. P. Oliver and R. M. Thomas. Finitely generated groups with automatic presentations. In *STACS 2005*, volume 3404 of *LNCS*, pages 693–704. Springer, 2005.

[Ott00] M. Otto. Epsilon-logic is more expressive than first-order logic over finite structures. *J. Symb. Logic*, 65:1749 – 1757, 2000.

[Ott01] M. Otto. Logics - Invariances - Games : domain specific algorithmic model theory. Survey Talk, held at the University of Edinburgh, 2001.

[Pan84] J.-J. Pansiot. On various classes of infinite words obtained by iterated mappings. In *Automata on Infinite Words*, pages 188–197, 1984.

[PP95] D. Perrin and J.-E. Pin. Semigroups and automata on infinite words. In J. Fountain, editor, *Semigroups, Formal Languages and Groups*, NATO Advanced Study Institute, pages 49–72. Kluwer academic publishers, 1995.

[PP04] D. Perrin and J.-E. Pin. *Infinite Words – Automata, Semigroups, Logic and Games*, volume 141 of *Pure and Applied Mathematics*. Elsevier, 2004.

[PS85] J.-E. Pin and J. Sakarovitch. Une application de la representation matricielle des transductions. *Theor. Comput. Sci.*, 35:271–293, 1985.

[PS95] G. Păun and A. Salomaa. Thin and slender languages. *Discrete Applied Mathematics*, 61(3):257–270, 1995.

[PS05] J.-E. Pin and P. V. Silva. A topological approach to transductions. *Theoretical Computer Science*, 340(1):443–456, 2005.

[Rab05] A. Rabinovich. On decidability of monadic logic of order over the naturals extended by monadic predicates. Unpublished note, 2005.

[Rig01] M. Rigo. Numeration systems on a regular language: Arithmetic operations, recognizability and formal power series. *Theoretical Computer Science*, 269:469, 2001.

[Ris02] Ch. Rispal. The synchronized graphs trace the context-sensisitive languages. *Electronic Notes in Theor. Comp. Sci.*, 68(6), 2002.

[RM02] M. Rigo and A. Maes. More on generalized automatic sequences. *J. of Automata, Languages and Combinatorics*, 7(3):351–376, 2002.

[RT06] A. Rabinovich and W. Thomas. Decidable theories of the ordering of natural numbers with unary predicates. Submitted, 2006.

[Rub04] S. Rubin. *Automatic Structures*. Ph.d. thesis, University of Auckland, NZ, 2004.

[Rub07] S. Rubin. Automata presenting structures: A survey of the finite-string case. Manuscript, to appear in Bulletin of Symbolic Logic, 2007.

[Sak03] J. Sakarovitch. *Éléments de théorie des automates*. Vuibert, 2003.

[See91] D. Seese. The structure of the models of decidable monadic theories of graphs. *Annals of Pure and Applied Logic*, 53:169–195, 1991.

[SS04] P. V. Silva and B. Steinberg. A geometric characterization of automatic monoids. *The Quarterly Journal of Mathematics*, 55:333–356, 2004.

[Sta97] L. Staiger. Rich omega-words and monadic second-order arithmetic. In *CSL*, pages 478–490, 1997.

[Str94] H. Straubing. *Finite Automata, Formal Logic, and Circuit Complexity.* Birkhäuser, Boston, 1994.

[SYZS92] A. Szilard, Sh. Yu, K. Zhang, and J. Shallit. Characterizing regular languages with polynomial densities. In *MFCS*, pages 494–503, 1992.

[Tho97] 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.

[TL94] Wolfgang Thomas and Helmut Lescow. Logical specifications of infinite computations. In *A Decade of Concurrency, Reflections and Perspectives, REX School/Symposium*, volume 803 of *LNCS*, pages 583–621. Springer-Verlag, 1994.

[Wal02] I. Walukiewicz. Monadic second-order logic on tree-like structures. *Theoretical Computer Science*, 275:311–346, 2002.

Index

Lebenslauf

Zur Person

Name:	Vince Bárány
Geboren:	8. Dezember 1976, Budapest, Ungarn
Staatsangehörigkeit:	ungarisch

Bildungsgang

1982 - 1990	Grundschule in Budapest
1991 - 1993	Gymnasium in Los Angeles
1993 - 1995	Gymnasium in Budapest
1995	Abitur
1997 - 2003	Studium der Informatik an der Eötvös Loránd Universität (ELTE), Budapest
07/2001	B.Sc. in Informatik (ELTE)
2001 - 2002	'Master's Programme' an der Vrije Universiteit Amsterdam (VU)
08/2002	M.Sc. in Informatik (VU)
SS 2003	Teilnahme an der 'Master Class' des Mathematical Research Institutes der Niederlande.
07/2003	M.Sc. in Informatik (ELTE)
seit 10/2003	Wissenschaftlicher Mitarbeiter an der Rheinisch-Westfälischen Technischen Hochschule Aachen, wo unter der Leitung von Prof. Dr. Erich Grädel die vorliegende Dissertation entstand.

September 26, 2007