INAUGURAL DISSERTATION

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich-Mathematischen Gesamtfakultät

der

Ruprecht-Karls-Universität

Heidelberg

vorgelegt von

Diplom-Mathematiker Edgar Busse
(geb. Damir Serikovich Muldagaliev)

aus Almaty

Tag der mündlichen Prüfung: 27. April 2006

Thema

Finite-State Genericity

On the Diagonalization Strength of Finite Automata

Gutachter: Prof. Dr. Klaus Ambos-Spies

Prof. Dr. Frank Stephan

Zusammenfassung

Algorithmische Generizitätskonzepte spielen eine wichtige Rolle in der Berechenbarkeits- und Komplexitätstheorie. Diese Begriffe stehen in engem Zusammenhang mit grundlegenden Diagonalisierungstechniken, und sie wurden zur Erzielung starker Trennungen von Komplexitätsklassen verwendet. Da für jedes Generizitätskonzept die zugehörigen generischen Mengen eine co-magere Klasse bilden, ist die Analyse generischer Mengen ein wichtiges Hifsmittel für eine quantitative Analyse struktureller Phänomene. Typischerweise werden Generizitätskonzepte mit Hilfe von Erweiterungsfunktionen definiert, wobei die Stärke eines Konzepts von der Komplexität der zugelassenen Erwiterungsfunktionen abhängt. Hierbei erweisen sich die sog. schwachen Generizitätskonzepte, bei denen nur totale Erweiterungsfunktionen berücksichtigt werden, meist als wesentlich schwächer als die vergleichbaren allgemeinen Konzepte, bei denen auch partielle Funktionen zugelassen sind. Weiter sind die sog. beschränkten Generizitätskonzepte – basierend auf Erweiterungen konstanter Länge – besonders interessant, da hier die Klassen der zugehörigen generischen Mengen nicht nur co-mager sind sondern zusätzlich Maß 1 haben. Generische Mengen diesen Typs sind daher typisch sowohl im topologischen wie im maßtheoretischen Sinn.

In dieser Dissertation initiieren wir die Untersuchung von Generizität im Bereich der Theorie der Formalen Sprachen: Wir führen finite-state-Generizitätskonzepte ein und verwenden diese, um die Diagonalisierungsstärke endlicher Automaten zu erforschen.

Wir konzentrieren uns hierbei auf die beschränkte finite-state-Generizität und Spezialfälle hiervon, die wir durch die Beschränkung auf totale Erweiterungsfunktionen bzw. auf Erweiterungen konstanter Länge erhalten. Wir geben eine rein kombinatorische Charakterisierung der beschränkt finite-state-generischen Mengen: Diese sind gerade die Mengen, deren charakteristische Folge saturiert ist, d.h. jedes Binärwort als Teilwort enthält. Mit Hilfe dieser Charakterisierung bestimmen wir die Komplexität der beschränkt finite-state-generischen Mengen und zeigen, dass solch eine generische Menge nicht regulär sein kann es aber kontext-freie Sprachen mit dieser Generizitätseigenschaft gibt. Weiter untersuchen wir den Einfluss der Länge der Erweiterungen und der Beschränkung auf totale Erweiterungsfunktionen auf die Stärke der korrespondierenden Generizitätskonzepte. Die Untersuchung von eingeschränkten Erweiterungsfunktionen, deren Wert jeweils nur von der Eingabenlänge oder einem Endstück der Eingabe konstanter Länge abhängt, verdeutlicht weiter die geringe Diagonalisierungsstärke endlicher Automaten. Wir beenden unsere Untersuchung der beschränkten finite-state-Generizität damit, dass wir zeigen, dass die Stärke dieser Konzepte dramatisch erhöht wird, wenn wir Erweiterungsfunktionen zugrundelegen, deren Eingaben Anfangstücke in redundanter Darstellung sind. Auf diese Art erhalten wir beschränkt finite-state-generische Mengen, die REG-bi-immun sind, d.h. deren Erkennung die Kapazität eines endlichen Automaten nicht nur unendlich oft sondern fast überall überschreitet.

Die von uns betrachteten unbeschränkten finite-state-Generizitätskonzepte basieren auf Moore-Funktionen und auf Verallgemeinerungen dieser Funktionen. Auch hier vergleichen wir die Stärke der verschiedenen korrespondierenden Generizitätskonzepte und erörtern die Frage, inwieweit diese Konzepte mächtiger als die beschränkte finite-state-Generizität sind.

Unsere Untersuchungen der finite-state-Generizität beruhen zum Teil auf neuen Ergebnissen über Bi-Immunität in der Chomsky-Hierarchie, einer neuen Chomsky-Hierarchie für unendliche Folgen und einer gründlichen Untersuchung der saturierten Folgen. Diese Ergebnisse – die von unabhängigem Interesse sind – werden im ersten Teil der Dissertation vorgestellt. Sie können unabhängig von dem Hauptteil der Arbeit gelesen werden.

Abstract

Algorithmic genericity notions play a major role in computability theory and computational complexity theory. These notions are closely related to important diagonalization techniques and they can be used for obtaining strong separations of complexity classes. Moreover, since for any genericity concept, the class of the correspondent generic sets is comeager, the analysis of generic sets leads to a quantitative analysis of structural phenomena. Typically, genericity concepts are based on partial or total extension functions, where the strength of a concept is determined by the complexity of the admissible extension functions, where in general weak genericity notions based only on total extension functions are much weaker than the corresponding genericity notions allowing partial extension functions too. Moreover, so called bounded genericity concepts based on extensions of constant length are of particular interest since the classes of the corresponding generic sets are not only comeager but also have measure 1. So generic sets of these types are abundant in the topological and the measure theoretic sense.

In this thesis we initiate the investigation of genericity in the setting of formal language theory: We introduce finite-state genericity notions, i.e., genericity notions related to the lowest class in the Chomsky hierarchy and we apply these concepts to explore the diagonalization strength of finite automata.

We focus on bounded finite-state genericity and some special cases hereof allowing only total extensions and extensions of fixed length. We give a purely combinatorial characterization of bounded finite-state genericity by showing that a set A is bounded finite-state generic if and only if its characteristic sequence is saturated, i.e., contains any binary string as a subword. We use this characterization for determining the complexity of bounded finite-state generic sets. In particular we show that no bounded finite-state generic language is regular but that there are such languages which are context-free. Moreover, we explore the impact of the length of the admissible extensions and of the question whether we allow partial or only total extension functions. We further illustrate the limitations of the diagonalization strength of finite automata by considering some restricted types of extension strategies, namely length invariant and oblivious extensions. We complete our investigation of bounded finite-state genericity by showing that the strength of these concepts can be dramatically increased if we work with more redundant representations of initial segments: This way we obtain bounded finite-state generic sets which are REG-bi-immune, i.e., sets which exceed the capacity of finite automata not only infinitely often but almost everywhere.

The unbounded finite-state genericity concepts which we consider are based on Moore functions and various generalizations of these functions. Again we compare the strength of different concepts and discuss the question in which respect these concepts are more powerful than bounded finite-state genericity.

Our analysis of finite-state genericity is based in part on new results on bi-immunity in the Chomsky hierarchy, on a Chomsky hierarchy of sequences, and a thorrough analysis of saturated sequences. These results – which are of independent interest – are presented in the first part of the thesis and can be read independently.

Contents

CHAPTER 1

Introduction

Algorithmic genericity notions play a major role in computability theory and computational complexity theory. These notions are closely related to the finite extension method introduced by Kleene and Post (1954), the basic diagonalization technique in computability theory. In general a genericity notion is linked to a certain complexity class and the corresponding generic sets have all properties which can be forced by a finite extension argument where the complexity of the individual diagonalization strategies does not exceed the given level. Important examples of genericity notions in computability theory are arithmetic genericity (Feferman (1965)) and 1-genericity (Hinman (1969)). The former is based on diagonalization strategies definable in first order arithmetic while the latter is based on recursively enumerable strategies. In computational complexity theory various polynomial time bounded genericity notions have been introduced and successfully applied to the structural analysis of the exponential time classes (see Ambos-Spies (1996)). The goal of this thesis is to study genericity in the setting of formal language theory. To be more specific, we will introduce and study finite-state genericity, i.e., genericity notions related to the level of regular languages or finite automata.

The applications of generic sets are manyfold. First, by the dependence of genericity on the finite extension method, by analyzing the properties of generic sets we can illustrate the power and limitations of this important diagonalization method. Moreover, as Myhill (1961) has observed, the finite extension method is closely related to the topological concept of Baire category: The class of sets which share a certain property is comeager if and only if this property can be enforced by a finite extension argument. This easily implies that, for any genericity notion, the generic sets are abundant namely form a comeager class. So, by showing that a property is shared by all generic sets, we obtain strong existence results, namely we may deduce that not only sets with the desired property exist but that they are abundant. Yet often it is easier to show that all generic sets have a certain property than to show that there is a single set with this property. This is due to the fact that in the former case we can take a modular approach: If a property can be split into (finitely or countably many) simpler subproperties, it suffices to show that these individual subproperties are guaranteed by genericity.

Since the countable intersection of comeager classes is comeager again, the classical Baire category concept allows a similar modular approach. In contrast to the genericity approach, however, the Baire category approach does not give any information on complexity. Since all complexity classes (considered in computability theory, computational complexity theory, and formal language theory) are countable, hence meager, showing that the class of sets sharing a property $\mathcal{P}$ is comeager, does not tell us on what complexity levels we may find sets with this property. If we know, however, that all sets which are generic (relative to some complexity class) have property $\mathcal{P}$ then by analyzing the complexity of the generic sets we can obtain some positive results on the complexity of sets with property $\mathcal{P}$. We explain this difference by an example from computability theory. We can use the Baire category approach to show that there are incomparable degrees of unsolvability. To do so, obviously, it suffices to show that the class of sets A such that the even part $A_{even} = \{2n : n \geq 0\}$ and the odd part $A_{odd} = \{2n+1 : n \geq 0\}$ are Turing incomparable is comeager. The proof of the latter can easily be modified to show that any 1-generic sets A has this property. On the other hand, any 1-generic set is nonrecursive but there are 1-generic sets which are Δ^0_2, i.e., sets which are recursively approximable or – equivalently – recursive in the halting problem. So we may deduce that there are not only incomparable Turing degrees but that there are such degrees in the degrees below $\mathbf{0}'$, the degree of the halting problem. (Since the class of Δ^0_2 sets is countable hence meager, the latter does not follow by purely topological means.)

In general the above observation has been used to obtain so called strong separations of complexity classes via genericity: Given complexity classes C and C$'$ such that C $\subset$ C$'$ (like the classes of the recursive and the recursively approximable sets in the preceding example) one might try to design a genericity concept, say C-genericity, strong enough to capture diagonalizations over C but on the other hand not too strong so that there will be C-generic sets in C$'$ (1-genericity in the above example). Then C and C$'$ will be separated by any property $\mathcal{P}$ such that $\mathcal{P}$ is not compatible with membership in C but such that any C-generic set has property $\mathcal{P}$. An example of such a property $\mathcal{P}$ which is of great interest in computational complexity is C-bi-immunity. (A set A is C-bi-immune if both, A and the complement $\overline{A}$ of A do not contain any infinite set from C as a subset. The interest in this notion stems from the fact that, for a time (and, similarly, for a space) complexity class C $=$ DTIME$(t(n))$, the time complexity of a C-bi-immune set does not only exceed the time bound $t(n)$ infinitely often but for all but finitely many inputs.) In complexity theory various genericity concepts implying bi-immunity (and thereby giving some strong separations) have been introduced but in this setting it also became apparent that in general we can design different genericity notions related to a complexity class which are of quite different strength (see Ambos-Spies (1996)). In order to explain this we have to look at the finite extension method and some refinements hereof more closely.

In a finite extension argument a language (or, equivalently, a set of natural numbers) A with a certain property $\mathcal{P}$ is inductively defined by specifying longer and longer initial segments of the characteristic sequence α of A. The construction of A exploits the fact that the global property $\mathcal{P}$ can be obtained by satisfying countably many finitary conditions R_e, $e \geq 0$, called *requirements*. To be more precise, each of the requirements R_e has the property that, for any given finite initial segment (i.e., finite binary string) x there is a finite extension y of x *forcing* R_e, namely any set X such that the characteristic sequence of X extends y will meet R_e. So a strategy for meeting R_e can be described by an *extension function* $f_e : \Sigma^* \to \Sigma^*$ where, for any string x the extension $x f_e(x)$ of x forces R_e. In other words, if we say that A *meets* an extension function f at a number n if $(\alpha \upharpoonright n) f(\alpha \upharpoonright n) \sqsubset \alpha$ – where $\alpha \upharpoonright n$ denotes the initial segment of length n of the characteristic sequence α of A – then A will meet requirement R_e if A meets f_e (at some number n). So, in order to define a set A with property $\mathcal{P}$, it suffices to inductively define longer and longer initial segments $\alpha_{-1} = \varepsilon \sqsubset \alpha_0 \sqsubset \alpha_1 \sqsubset \alpha_2 \ldots$ of α by letting $\alpha_e = \alpha_{e-1} f_e(\alpha_{e-1})$. Then A meets f_e at $|\alpha_{e-1}|$ thereby ensuring that, in step e of the construction, A meets requirement R_e ($e \geq 0$).

By identifying strategies with extension functions we can define the complexity of a diagonalization strategy by the complexity of the corresponding extension function. Moreover, we get a very general approach for defining genericity notions: Given any countable class $\mathcal{F}$ of extension functions, we say that a set A is $\mathcal{F}$-*generic* if A meets all extension functions in $\mathcal{F}$. Many of the genericity concepts in the literature can be described this way by letting $\mathcal{F}$ be some of the common (functional) complexity classes. For instance $\mathcal{F}$-genericity coincides with Feferman's arithmetical genericity if we let $\mathcal{F}$ be the class of arithmetical functions and $\mathcal{F}$-genericity coincides with DTIME$(t(n))$-genericity in the sense of Lutz (1990) if we let $\mathcal{F}$ be the class of functions computable in time $t(n)$.

The above introduced concepts of $\mathcal{F}$-genericity, however, only capture such diagonalizations which can be phrased as finite extension arguments. Many proofs in computability and complexity theory, however, require more sophisticated diagonalization techniques like wait-and-see arguments (also called slow diagonalizations) or finite (or even infinite) injury priority arguments (see e.g. Soare (1987)). So for obtaining stronger genericity concepts we have to define genericity notions capturing these types of diagonalizations too. The additional power of these more sophisticated techniques stems from their higher efficiency. In general, when we apply such a technique our goal is not to obtain a set with a property which we cannot obtain by a standard finite extension argument but we want to decrease the complexity of the constructed set. So the priority method is the fundamental method for constructing recursively enumerable sets. Typically, a finite extension construction of a Δ^0_2 set with a certain property $\mathcal{P}$ can be turned into the construction of a recursively enumerable set with this property by using a finite injury priority argument. Genericity notions capturing the essence of the finite injury method have been introduced by Maass (1982) and Jockusch (1985).

In a constructive environment, however, injuries can be avoided and the priority method can be replaced by wait-and-see arguments. So, for our purposes, it suffices to consider this refinement of the finite extension method. While in a standard finite extension argument, in order to meet a requirement R, for any finite initial segment of the characteristic sequence of the set under construction we can find a finite extension forcing R, in a wait-and-see argument such extensions may exist only for some initial segments. Moreover, R will be met (for some trivial reason) if there are only finitely many initial segments of the set under construction which have extensions forcing R. (I.e., intuitively, diagonalization action for the sake of requirement R has to be taken only if there are infinitely many chances in the course of the construction to do so.) Correspondingly, here the strategy for meeting a requirement R is described by a *partial* extension function f. Moreover, in order to meet R it suffices to ensure that f is not dense along A or that A meets f (at some n) where we say that f is *dense* along A if f is defined on infinitely many initial segments of the characteristic sequence of A. As in the case of total extension functions, for any countable class $\mathcal{F}$ of partial extension functions, the class of sets A which meet every partial extension function $f \in \mathcal{F}$ which is dense along A is comeager. Sets with this property are just the sets generic relative to $\mathcal{F}$ and we call them $\mathcal{F}$-generic. (Note that if $\mathcal{F}$ consists only of total functions then this definition coincides with the previous definition of $\mathcal{F}$-genericity.) In order to distinguish between genericity based on total and partial extension functions we call the former weak genericity. In particular, we call a set A *weakly $\mathcal{F}$-generic* if A is $\hat{\mathcal{F}}$-generic where $\hat{\mathcal{F}}$ consists of all total extension functions in $\mathcal{F}$.

To illustrate the higher efficiency of wait-and-see arguments we compare the construction of a PTIME-bi-immune set A by a standard finite extension argument and by a wait-and-see argument. A typical requirement to be met is of the form

$$R: \quad \text{If } B \text{ is infinite then } B \cap A \neq \emptyset.$$

where B is a polynomial time computable set. In a finite extension argument we can meet R by meeting the total extension function

$$f(x) = \begin{cases} 1^{n_x} & \text{if } B \text{ is infinite} \\ \varepsilon & \text{otherwise.} \end{cases}$$

where n_x is the least number n such that the $(n + |x|)$th word is a member of B. In a wait-and-see argument we can work with the partial extension function

$$\hat{f}(x) = \begin{cases} 1 & \text{if the } |x|\text{th word is an element of } B \\ \uparrow & \text{otherwise.} \end{cases}$$

(Note that $\hat{f}$ is dense (along any set) if and only if B is infinite and by meeting $\hat{f}$ at n we insure that the nth word is an element of both, B and A.) The above strategies show that we can construct a PTIME-bi-immune set A using both techniques, a plain finite extension argument and a wait-and-see argument. The approaches, however, greatly differ in the complexity of the strategies needed to meet a single requirement: While the (partial) function $\hat{f}$ is polynomial (in fact linear) time computable, the complexity of f depends on the length of the gaps in the set B. Though, for a single PTIME set B, the length of these gaps is recursively bounded, there is no uniform recursive bounds for all PTIME sets. In particular, the finite extension construction yields a nonrecursive set A, and for any recursive time bound $t(n)$ there is a weakly DTIME$(t(n))$-generic set (i.e. a DTIME$(t(n))$-generic set in the sense of Lutz (1990)) which is not PTIME-bi-immune (Mayordomo (1994)). On the other hand, a wait-and-see construction based on the above partial extension functions $\hat{f}$ yields a recursive (in fact exponential time computable) PTIME-bi-immune set (Balcázar and Schöning (1985)), and any DTIME$(O(n))$-generic set is PTIME-bi-immune (now DTIME$(O(n))$-genericity in the strong sense, i.e., genericity based on partial extension functions computable in linear time).

In fact the above wait-and-see construction improves the plain finite extension construction not only with respect to complexity. In addition, in the case of partial extensions it suffices to consider extensions of length 1 while in the total case the length of the extensions depends on the inputs. This observation is of interest, since for any countable class $\mathcal{F}$ of (partial) bounded extension functions the class of $\mathcal{F}$-generic sets is not only comeager but it also has Lebesgue measure 1 (see e.g. Ambos-Spies (1996); here we call a function f *k-bounded* if $|f(x)| \leq k$ for all strings x, and *bounded* if f is k-bounded for some number k). In contrast, for any sufficiently closed family $\mathcal{F}$ containing unbounded functions, the class of (weakly) $\mathcal{F}$-generic sets has measure 0.

The above discussion of genericity in computability and computational complexity theory shows that the strength of a genericity concept does not only depend on the complexity of the extension functions it is based on but also on the question whether we admit partial or only total extension functions. Moreover, genericity notions based on bounded extension functions are of particular interest since they yield abundance results not only in the sense of category but also in the sense of measure.

Our analysis of finite-state genericity is guided by the above observations. Moreover, we focus on bounded genericity notions, i.e., our main goal is the investigation of the diagonalization power of finite-state transducers producing output of constant length. The outline of our thesis is as follows.

In Chapter 2 we present results on formal languages and infinite sequences which serve as the background of our investigations. After fixing some notation in Section 2.1 and shortly reviewing some basic notions and results from formal language theory and computational complexity in Section 2.2, in Section 2.3 we review various notions of regular functions which will serve as extension functions in our finite-state genericity notions. In particular we discuss some variants of Moore functions and introduce the notion of an (k-)bounded regular function and show how this concept is related to the Moore approach. In Section 2.4 we investigate (bi-)immunity for the Chomsky language classes. In particular, we observe that there is no strong separation of the classes of regular and context-free languages in terms of bi-immunity: no context-free language is REG-bi-immune. Since genericity actually is a property of the characteristic sequence of a set, in Section 2.5 we have a closer look at the Chomsky complexity of sequences. There we introduce a Chomsky hierarchy of sequences and show how the location of a language in the Chomsky hierarchy is related to the location of its characteristic sequence in this new hierarchy. In particular, we see that any set with regular characteristic sequence is regular too but that there are regular sets which have a nonregular characteristic sequence. The Chomsky complexity of a sequence is defined in terms of the complexity of the prefix set of the sequence. As a possible alternative we introduce such a hierarchy based on predictability by machines corresponding to the Chomsky language classes, and show that regularity of a sequence coincides with predictability by finite automata. We also show, however, that push down automata can predict sequences which are not context-free. The final section of Chapter 2 is devoted to saturated sequences, i.e., infinite sequences which contain any string as a substring. Since, as we will show later, saturation coincides with some of our finite-state genericity concepts we study this concept in great detail. Following some useful observations on invariance and closure properties of the saturated sequences we look at the complexity of these sequences. In particular we show that no regular languages is saturated (i.e., has a saturated characteristic sequence) while there are context-free – in fact – linear languages which are saturated. Moreover, we give a characterization of non-saturation in terms of partial finite-state predictability which will later lead to the relation between saturation and finite-state genericity and we will look at partial saturation properties.

Though the results in this chapter will be employed in our investigation of finite-state genericity some of our new results here are of independent interest. Hence we looked at bi-immunity (Section 2.4) and at the Chomsky complexity of sequences (Section 2.5) for all levels of the Chomsky hierarchy though in the sequel we will only need the results related to the regular and – in part – to the context-free languages. These more general results might be useful, however, for a forthcoming analysis of genericity on the other levels of the Chomsky hierarchy.

In Chapter 3 we shortly review the basic concepts and results on genericity, Baire category and diagonalization. In particular we introduce the framework in which our genericity notions are defined and make some general observations on the different genericity types like weak and bounded genericity.

In Chapter 4 – which is the core of this thesis – we introduce and analyze bounded finite-state genericity (bounded reg-genericity, for short). In Section 4.1

we show that for bounded finite-state genericity (based on partial regular bounded extension functions) the length of the admissible extensions does not matter whereas in the weak case (i.e. in case of total extension functions) the diagonalization power increases with the length of the extensions. Moreover, surprisingly and in contrast to corresponding results in computational complexity, bounded finite-state genericity and bounded weak genericity coincide. In other words, partial bounded regular extension functions can be simulated by total bounded regular extension functions, though the simulation in general will require an increase in the length of the extensions. Some of these results are obtained by the observation that bounded reg-genericity coincides with saturation. By our previous results on saturation the latter also illustrates the diagonalization strength of bounded-reg-genericity, namely no regular set is bounded reg-generic whereas there are linear hence context-free languages which are bounded reg-generic. Moreover, bounded reg-genericity in general does not imply REG-immunity.

The coincidence of bounded reg-genericity and saturations reveals the weakness of this concept. In Section 4.2 we further illustrate the low diagonalization power of bounded finite-state transducers by comparing bounded reg-genericity with apparently weaker concepts based on regular extension strategies which are given only partial information on the previously defined initial segment, namely strategies which depend only on the length of the initial segment and strategies which only use the last m bits of the initial segment (for some constant m).

We then discuss how the power of bounded finite-state genericity may be increased. For this sake in Section 4.3 we first discuss some direct Cantor-style diagonalization arguments in which the diagonalization only depends on the place where the action has to take place and not on the previously specified part of the set under construction. By formalizing this concept we introduce the concept of Cantor-style reg-genericity and show that this concept coincides with REG-bi-immunity, hence is not subsumed by bounded reg-genericity. The latter can be traced to the fact that a finite automaton which is given an initial segment of length n cannot extract the nth string from this information. This observation leads us to define finite-state extension functions which obtain as inputs finite initial segments in a more redundant representation– which allows to overcome the just described shortcoming – and to study the strength of the corresponding genericity notions (Section 4.4).

In Chapter 5 we start the investigation of unbounded finite-state genericity. Based on stronger and stronger notions of regular functions we introduce a hierarchy of corresponding genericity notions. We first consider Moore genericity based on partial Moore functions. We show that this concept strengthens bounded reg-genericity but that it does not suffice for forcing REG-bi-immunity. The latter can be achieved by using extension functions of generalized Moore type (where a generalized Moore function adds an arbitrary word to the output for every letter read – not a single letter as in case of a Moore function). We further discuss the strength of these genericity concepts if we replace deterministic Moore automata by nondeterministic ones and if we support the strategies by giving them the initial segments in the redundant form introduced in Section 4.4

Finally, in Chapter 6 we give some directions for further research in this area.

Acknowledgments

I am deeply grateful to my advisor, Klaus Ambos-Spies, for his intense and committed supervision. A large part of the results in this thesis emerged from numerous discussions with him, and his suggestions were crucial for developing the fundamental concepts treated in this thesis. Moreover, he patiently helped to improve the presentation of the material.

I am also grateful to the other members of the Heidelberg Logic Group: Paolo Di Muccio, Klaus Gloede, Piotr Grabowski, Felicitas Hirsch, Wolfgang Merkle, Nenad Mihailovic, Gert H. Müller and Jan Reimann, to the former members Frank Stephan, Hans-Christian Nies, and to the recent visitors Serikzhan Badaev and Bakhadyr Khoussainov.

I gratefully recognize all my friends, in particular, Paolo Di Muccio, Amaranta Melchor del Rio and Augusto Minatta for kind support during the study in Heidelberg.

Finally I am indebted to my parents and my fiancée Tanja Nikonova.

CHAPTER 2

Formal Languages and Infinite Sequences

In this chapter we will provide the concepts and results from formal language theory which we will need for introducing and analyzing our finite-state genericity notions. After fixing some notation in Section 2.1, in Section 2.2 we review some fundamental results on the Chomsky language classes and their relations to computability and computational complexity theory. In Section 2.3 we discuss various notions of regular functions. In particular we introduce a new notion of a regular function of type $f : \Sigma^* \rightarrow \Sigma^k$ ($k \geq 1$) and analyze the relation of this notion to some previously introduced concepts. In Section 2.4 we investigate immunity and bi-immunity for the Chomsky language classes and explore the question to what extend these notions yield strong separations among the Chomsky classes. In Section 2.5 we introduce a Chomsky hierarchy of sequences. This classification is based on the complexity of the prefix sets of sequences. We use this framework for comparing the Chomsky complexity of languages and their characteristic sequences. Moreover, we relate this hierarchy of sequences to a similar hierarchy based on predictability of sequences by machines corresponding to the levels of the Chomsky hierarchy. Finally, in Section 2.6 we study saturated 0-1-sequences, i.e., infinite binary sequences in which all binary words occur as subwords.

The results in Section 2.2 are standard and can be found in most textbooks (see e.g. Salomaa (1973), Hopcroft and Ullman (1979), Balcázar et al. (1990)) or in the handbook of formal languages by Rozenberg and Salomaa (1997). So in this section we omit references and proofs. The other sections obtain, besides results from the literature, a variety of new results. Here references to the previously known results and proofs of the new results are given.

Though the results of this chapter will be used in our analysis of finite-state genericity, many of the results will be of interest for themselves. So we will limit our investigation of (bi-)immunity and of the Chomsky complexity of sequences not only to the regular (or context-free) case - which will be used in the following - but we will carry out a more systematic analysis covering all levels of the Chomsky hierarchy.

2.1 Notation and Basic Concepts

Though we will mainly consider languages over the binary alphabet we will introduce the basic notions of formal language theory for arbitrary alphabets.

An *alphabet* is a finite nonempty ordered set. In the following we let $\Sigma_n = \{a_0, \ldots, a_{n-1}\}$ denote the n-ary alphabet where the elements are listed in order (i.e., $a_0 < a_1 < \cdots < a_{n-1}$). In particular, we let $\Sigma_2 = \{0,1\}$ and $\Sigma_1 = \{0\}$ be the binary alphabet and the unary alphabet, respectively. For simplicity, we usually denote the binary alphabet by Σ. The elements of an alphabet T are called *letters*, the elements of the binary alphabet are also called *bits*. Letters are usually denoted by lower case Latin letters from the beginning of the alphabet (a, b, c, ...).

A *word* over an alphabet T is a finite sequence of letters from T. The empty sequence is called the *empty word* and is denoted by ε. The set of all words over T is denoted by T^*, the set of the nonempty words is denoted by T^+. Words are usually denoted by lower case Latin letters from the end of the alphabet (u, v, w, x, y, z). The *length* of a word w is denoted by $|w|$ and we let

$$T^k = \{x \in T^* : |x| = k\} \quad \text{and} \quad T^{\leq k} = \{x \in T^* : |x| \leq k\}$$

be the sets of words of length k and of length at most k, respectively, over the alphabet T ($k \geq 0$). For a word x of length k we let $x = x(0)\ldots x(k-1)$ where $x(i)$ denotes the $(i+1)$th letter in x.

The ordering on an alphabet T is extended to the *length-lexicographical ordering* on T^* by letting v be less than w if the length of v is less than the length of w or if v and w have the same length and, for the least k such that $v(k)$ and $w(k)$ differ, the letter $v(k)$ precedes the letter $w(k)$ in the ordering of T. In general, we denote the ordering on an alphabet T and the induced length-lexicographical ordering on T^* both by $<$.

For the binary alphabet Σ, we let z_n denote the $(n+1)$th word with respect to the length-lexicographical ordering and we let z_n^k be the $(n+1)$th word of length k. Since there are 2^k binary words of length k, hence $2^{k+1}-1$ binary words of length at most k,

$$\Sigma^k = \{z_0^k, \ldots, z_{2^k-1}^k\} = \{z_{2^k-1}, \ldots, z_{2^{k+1}-2}\} \quad \text{and} \quad \Sigma^{\leq k} = \{z_n : n < 2^{k+1}-1\}.$$

A *language* A over the alphabet T is a set of words over T, i.e., $A \subseteq T^*$. In the following languages will be denoted by upper case Latin letters. A language over the binary alphabet Σ is also called a *binary language* or simply a *set*, while a set of binary languages is called a *class*. By identifying the $(n+1)$th binary word z_n with the number n, sometimes we interpret a binary language A as a set of natural numbers (i.e., we may write $n \in A$ in place of $z_n \in A$). Moreover, we often identify a set A with its *characteristic function* c_A. I.e., we write $A(x) = 1$ if $x \in A$ and $A(x) = 0$ if $x \notin A$.

The *characteristic sequence* of set A is denoted by $\chi(A)$. I.e., $\chi(A)$ is the infinite binary sequence defined by

$$\chi(A) = A(z_0)A(z_1)A(z_2)\ldots = A(0)A(1)A(2)\ldots$$

Conversely, for an infinite binary sequence $\alpha = \alpha(0)\alpha(1)\alpha(2)\ldots$ the *set* $S(\alpha) \subseteq \Sigma^*$ *corresponding to* α is defined by

$$S(\alpha) = \{z_n : \alpha(n) = 1\}$$

Note that $S(\chi(A)) = A$ and $\chi(S(\alpha)) = \alpha$, whence we may identify a set with its characteristic sequence and a sequence with the set corresponding to it. Usually we denote the characteristic sequence of a set $A, B, C, \ldots$ by the corresponding Greek letter $\alpha, \beta, \gamma, \ldots$.

The set of all infinite binary sequences is denoted by Σ^ω. This set (which is often denoted by 2^ω) is also called the *Cantor space*. Elements of Σ^ω will be shortly called *sequences* and will be denoted by lower case Greek letters.

For $s \in \Sigma^* \cup \Sigma^\omega$ and $v \in \Sigma^*$ we let vs denote the *concatenation* of v and s and we call v a *prefix* or *initial segment* of vs, s a *suffix* or *final segment* of vs. If $s \neq \varepsilon$, v is a *proper* prefix or initial segment, and if $v \neq \varepsilon$, s is a *proper* suffix or final segment. We write $v \sqsubseteq s$ if v is a prefix of s and $v \sqsubset s$ if the prefix v is proper. The prefix of a (finite or infinite) sequence s of length n is denoted by $s \upharpoonright n = s(0)\ldots s(n-1)$. We also write $A \upharpoonright z_n$ or $A \upharpoonright n$ in place of $\chi(A) \upharpoonright n$.

The *prefix set* $Prefix(\alpha) \subseteq \Sigma^*$ of an infinite sequence α is the set of all finite prefixes of α, i.e., $Prefix(A) = \{\alpha \upharpoonright n : n \geq 0\}$. In the following we will consider both representations of a sequence α by sets, namely the corresponding set $S(\alpha)$ and the prefix set $Prefix(\alpha)$. Note that any set corresponds to a sequence whereas the prefix set of a sequence is of some particular syntactic form. In particular, such a set contains just one word of any given length.

By x^n we denote the nth *iteration* of the word x, i.e., $x^0 = \varepsilon$ and $x^{n+1} = x^n x$. Similarly, the ω-*iteration* x^ω of a word x is the infinite sequence obtained by concatenating infinitely many copies of the word x. A sequence $\alpha = x^\omega$ is called *periodic*, a sequence $\alpha = xy^\omega$ is called *almost periodic*.

Finally, for $s, s' \in \Sigma^* \cup \Sigma^\omega$ and $v, w \in \Sigma^*$ we call w a *subword* or *infix* of s if $s = vws'$. If v is a subword of α we also say that v *occurs* in α. We say that v *occurs (at least) k times* in α if there are words $w_1 \sqsubset w_2 \sqsubset \ldots \sqsubset w_k$ such that $w_m v \sqsubset \alpha$ for $m = 1, \ldots, k$. A sequence α in which all words occur is called *saturated*.

2.2 Grammars and Automata

In this section we review some fundamental results on the Chomsky language classes and their relations to computability and computational complexity theory. The notions and results which we will present here can be found in the standard textbooks (see e.g. Salomaa (1973), Hopcroft and Ullman (1979), Balcázar et al. (1990)) or in the handbook of formal languages by Rozenberg and Salomaa (1997). So, in general, we will omit references and proofs. We will start by reviewing the different types of Chomsky grammars and the corresponding Chomsky hierarchy of languages.

2.2.1 Chomsky Grammars and the Chomsky Hierarchy

Definition 2.1 A *(Chomsky) grammar* is a quadrupel $G = (N,T,S,P)$, where N and T are disjoint alphabets, $S \in N$, and P is a finite subset of $(N \cup T)^* - T^* \times (N \cup T)^*$. The elements of N are called *nonterminal symbols* or *(syntactical) variables*, those of T are called *terminal symbols* (or *terminals* for short), S is the *start symbol* or *axiom*, and P the set of *(production) rules*.

For a rule $(u,v) \in P$ we usually write $u \to v$ and we call u the *premise* and v the *conclusion* of the rule. A word $w \in T^*$ is called a *terminal* word.

Next we define derivations in a grammar G and the language generated by G.

Definition 2.2 (a) Let $G = (N,T,S,P)$ be a grammar. For words $x,y \in (N \cup T)^*$ we say that y can be *derived* from x *in one step* - and write $x \Rightarrow_G y$ - if there is a rule $u \to v \in P$ and words $w_1, w_2 \in (N \cup T)^*$ such that $x = w_1 u w_2$ and $y = w_1 v w_2$. A *derivation* (*of length n*) of a word y from a word x is a sequence of words $x_0, \ldots, x_n \in (N \cup T)^*$ such that $x = x_0$, $x_i \Rightarrow_G x_{i+1}$ for $i = 0, \ldots, n-1$, and $x_n = y$. We say that y can be *derived* from x (*in n steps*) if there is a derivation of y from x (of length n) and we write $x \Rightarrow_G^* y$ ($x \Rightarrow_G^n y$). The *language* $L(G)$ *generated* by G consists of all terminal words which can be derived from the axiom S, i.e., $L(G) = \{w \in T^* : S \Rightarrow_G^* w\}$.

(b) Two grammars G_0 and G_1 are *equivalent* if they generate the same language, i.e., $L(G_0) = L(G_1)$.

If the grammar G is known from the context we write $\Rightarrow$ ($\Rightarrow^n$, $\Rightarrow^*$) in place of $\Rightarrow_G$ ($\Rightarrow_G^n$, $\Rightarrow_G^*$).

A language is called a *Chomsky language* if it is generated by a Chomsky grammar. Chomsky has shown that the Chomsky languages are just the recursively enumerable languages. So, in general, we cannot decide whether a word is generated by a given grammar. In order to get grammars with effectively or even efficiently decidable word problems, Chomsky has introduced special types of grammars obtained by restricting the admissible forms of rules. For describing these concepts we have to deal with rules with empty conclusions first.

Definition 2.3 Let $G = (N,T,S,P)$ be a grammar. A rule $u \to \varepsilon$ with empty conclusion is called an *ε-rule*. The grammar G is *ε-free*, if P does not contain any ε-rule. G is called *ε-honest* if G is either ε-free or $S \to \varepsilon$ is the only ε-rule in P and S does not occur in the conclusion of any rule in P.

Note that for ε-free G, $\varepsilon \notin L(G)$.

Definition 2.4 A rule $u \to v$ is

1. *length-increasing* if $|u| \leq |v|$;
2. *context-sensitive* if there are a variable $X \in N$, words $u_1, u_2 \in (N \cup T)^*$ and a word $y \in (N \cup T)^+$ such that $u = u_1 X u_2$ and $v = u_1 y u_2$;
3. *context-free* if $u \in N$;
4. *linear* if $u \in N$ and $v \in T^* \cup T^* N T^*$;
5. *right-linear* if $u \in N$ and $v \in T^* \cup T^* N$;

Definition 2.5 (a) A grammar $G = (N,T,S,P)$ is

1. *length-increasing* if G is ε-honest and all rules in P (with the possible exception of $S \to \varepsilon$) are length-increasing;
2. *context-sensitive* (or of *type-1*) if G is ε-honest and all rules in P (with the possible exception of $S \to \varepsilon$) are context-sensitive;
3. *context-free* (or of *type-2*) if all rules in P are context-free;
4. *linear* if all rules in P are linear;
5. *right-linear* (or of *type-3*) if all rules in P are right-linear;

Moreover, any grammar G is of *type-0*.

(b) A language A is *length-increasing* (*context-sensitive*, *context-free*, *linear*, *right-linear*, or *of type-i* (i=0,1,2,3)) if there is a grammar of the corresponding type which generates A.

A grammar or language of type-i is called a *Chomsky-i-grammar* or *Chomsky-i-language*, respectively, and right-linear languages are also called *regular*. The classes of the length-increasing, context-sensitive, context-free, linear, right-linear, type-i (i=0,1,2,3), and regular languages over alphabet T (i.e., T is the terminal alphabet) are denoted by LI_T, CS_T, CF_T, LIN_T, RLIN_T, $\mathrm{CH}i_T$, and REG_T, respectively. If T is the binary alphabet $\Sigma = \{0,1\}$ then we omit the subscript T.

Note that, by definition, $\mathrm{CH0}_T$ is the class of all Chomsky languages (over the alphabet T) while $\mathrm{CH1}_T = \mathrm{CS}_T$, $\mathrm{CH2}_T = \mathrm{CF}_T$, and $\mathrm{CH3}_T = \mathrm{RLIN}_T = \mathrm{REG}_T$. Moreover, any right-linear grammar is linear, any linear grammar is context-free, and any context-sensitive grammar is length-increasing. Finally, for ε-honest grammars obviously context-freenes implies context-sensitivity. Since, for any context-free grammar we can find an equivalent context-free grammar which is ε-honest the above relations among the different types of Chomsky grammars yield the following inclusions for the Chomsky language classes (over any fixed alphabet T):

$$\mathrm{REG}_T = \mathrm{RLIN}_T = \mathrm{CH3}_T \subseteq \mathrm{LIN}_T \subseteq \mathrm{CF}_T = \mathrm{CH2}_T \subseteq \mathrm{CS}_T = \mathrm{CH1}_T \subseteq \mathrm{LI}_T \subseteq \mathrm{CH0}_T$$

Chomsky has shown that the classes CS_T and LI_T coincide (for any alphabet T) but that the other inclusions are proper with the following exception: For the unary alphabet $\Sigma_1 = \{0\}$, the classes of the regular, linear and context-free languages coincide.

Theorem 2.6 *(Chomsky Hierarchy Theorem) For any alphabet T with $|T| \geq 2$*

$$\mathrm{REG}_T = \mathrm{RLIN}_T = \mathrm{CH3}_T \subset \mathrm{LIN}_T \subset \mathrm{CF}_T = \mathrm{CH2}_T \subset \mathrm{CS}_T = \mathrm{CH1}_T = \mathrm{LI}_T \subset \mathrm{CH0}_T \tag{2.1}$$

while for the unary alphabet $\Sigma_1 = \{0\}$,

$$\mathrm{REG}_{\Sigma_1} = \mathrm{RLIN}_{\Sigma_1} = \mathrm{CH3}_{\Sigma_1} = \mathrm{LIN}_{\Sigma_1} = \mathrm{CF}_{\Sigma_1} = \mathrm{CH2}_{\Sigma_1} \subset \mathrm{CS}_{\Sigma_1} = \mathrm{CH1}_{\Sigma_1} = \mathrm{LI}_{\Sigma_1} \subset \mathrm{CH0}_{\Sigma_1} \tag{2.2}$$

The classes of the Chomsky hierarchy have the following closure properties.

	Union	Intersection	Complement	Concatenation
CH0	*yes*	*yes*	*no*	*yes*
CS	*yes*	*yes*	*yes*	*yes*
CF	*yes*	*no*	*no*	*yes*
LIN	*yes*	*no*	*no*	*no*
REG	*yes*	*yes*	*yes*	*yes*

(2.3)

Moreover, all of the Chomsky classes are closed under finite variants, a fact we will (tacitly) use in the following quite frequently. Though the class of context-free languages is not closed under intersection, the following weaker closure property holds.

Lemma 2.7 *For any context-free language A and any regular language B, $A \cap B$ is context-free.*

In the next subsections, we look at the classes of the regular sets and the context-free sets, REG and CF, in more detail. In particular, we review the machine characterizations of the Chomsky classes and we summarize some facts about complexity.

2.2.2 Regular Languages and Finite Automata

Definition 2.8 A *deterministic finite automaton (DFA) M* is a quintuple $M = (T,S,\delta,s_0,F)$ where T is an alphabet, S is a finite set, $\delta : S \times T \to S$ is a total function, $s_0 \in S$, and $F \subseteq S$. T is called the *input alphabet*, S the set of *states*, δ the *transition function*, s_0 the *initial state*, and F the set of *final* or *accepting* states.

On an input $w \in T^*$ of length n, a *DFA* $M = (T,S,\delta,s_0,F)$ behaves as follows. Reading w letter by letter from left to right, M runs through a sequence of states $s_0, s_1, \ldots, s_{n-1}$ beginning with the initial state s_0 and going from state s to state $\delta(s,a)$ when a is the next letter read. M accepts the input w if the computation ends in a final state, i.e., if $s_{n-1} \in F$. In order to define this behaviour of M more formally, we extend the transition function $\delta : S \times T \to S$ to $\delta^* : S \times T^* \to S$ where $\delta^*(s,w)$ is the state reached by M after reading w when starting in state s.

Definition 2.9 Let $M = (T,S,\delta,s_0,F)$ be a deterministic finite automaton. The *generalized transition function* $\delta^* : S \times T^* \to S$ of M is inductively defined by

$$\delta^*(s,\varepsilon) = s$$
$$\delta^*(s,wa) = \delta(\delta^*(s,w),a),$$

where $s \in S, a \in T, w \in T^*$. A word $w \in T^*$ is *accepted by M* if $\delta^*(s_0,w) \in F$. The *language* $L(M) \subseteq T^*$ *accepted* by M is the set of all words over T accepted by M , i.e., $L(M) = \{w \in T^* : \delta^*(s_0,w) \in F\}$.

If M on input w, $|w| = n$, runs through the states $s_0, \ldots, s_{n-1}$ (i.e. $s_i = \delta^*(s_0, w \upharpoonright i+1)$ for $i = 0, \ldots, n-1$) then we call $s_0, \ldots, s_n$ the *run* or *computation* of M on input w. More generally, we call $s_0, \ldots, s_n$ a *run* of M if $s_0, \ldots, s_n$ is the run of M on some input w of length n. For a *DFA* $M = (T,S,\delta,s_0,F)$, every word $w \in T^*$ determines a unique run of M.

The following notion of a nondeterministic automaton in general allows more than one run on a given input w. Now, if the automaton is in a state s and reads a letter a, it may choose the next state from a finite transition set $\Delta(s,a)$. It may happen that $\Delta(s,a)$ is empty, i.e., that no transition is possible. If M cannot read the input completely, then w will be rejected. Moreover, the machine M will have a choice for the initial state in which the run states.

Definition 2.10 A *nondeterministic finite automaton (NFA)* M is a quintuple $M = (T,S,\Delta,S_0,F)$ where T is an alphabet, S is a finite set, $\Delta \subseteq (S \times T) \times S$ and S_0, $F \subseteq S$ where S_0 is nonempty. T is called the *input alphabet*, S the set of *states*, Δ the *transition relation*, S_0 the set of *initial states*, and F the set of $final$ or $accepting$ states.

For defining the language $L(M)$ accepted by M, it is useful to look at the transition relation as a function $\Delta : S \times T \to POWER(S)$, by letting $\Delta(s,a) = \{s' \in S : ((s,a),s') \in \Delta\}$. Moreover, for a set $\hat{S}$ of states we let $\Delta(\hat{S},a) = \{s' \in S : \exists s \in \hat{S}[((s,a),s') \in \Delta]\}$. Intuitively, $\Delta(s,a)$ is the set of all states s' into which M may move from state s when reading a, and $\Delta(\hat{S},a)$ is the set of all states s' into which M may move from a state in $\hat{S}$ when reading a.

Definition 2.11 Let $M = (T,S,\Delta,S_0,F)$ be a nondeterministic finite automaton. The *generalized transition relation* $\Delta^* : S \times T^* \to POWER(S)$ of M is inductively defined by

$$\Delta^*(s,\varepsilon) = \{s\}$$

and

$$\Delta^*(s,wa) = \bigcup_{s' \in \Delta^*(s,w)} \Delta(s',a),$$

where $s \in S, a \in T, w \in T^*$. Moreover, for $\hat{S} \subseteq S$ and $w \in T^*$, let $\Delta^*(\hat{S},w) = \bigcup_{s \in \hat{S}} \Delta^*(s,w)$.

Then a word $w \in T^*$ is *accepted by* M if there is a state $s_0 \in S_0$ such that $\Delta^*(s_0,w) \cap F \neq \emptyset$. The *language* $L(M) \subseteq T^*$ *accepted* by M is the set of all words over T accepted by M , i.e., $L(M) = \{w \in T^* : M \text{ accepts } w\} = \{w \in T^* : \Delta^*(S_0,w) \cap F \neq \emptyset\}$.

We call $s_0,\ldots,s_{n-1}$ a *possible run* or a *possible computation* of M on input $w = a_0 \ldots a_{n-1}$ ($|w| = n$) if $s_0 \in S_0$ and $s_{i+1} \in \Delta(s_i,a_i)$ for $i = 0,\ldots,n-1$. Note that M accepts w if and only if there is a possible run of M on input w which ends in a final state.

We let DFA_T (NFA_T) denote the class of languages over the alphabet T which are accepted by a deterministic (nondeterministic) finite automaton (and we omit the subscript T if T is the binary alphabet Σ).

Theorem 2.12 *For any alphabet T,* $\mathrm{DFA_T} = \mathrm{NFA_T} = \mathrm{REG_T}$.

We get an alternative inductive characterization of $\mathrm{REG_T}$ by looking at regular expressions.

Definition 2.13 The *regular expressions* over T are inductively defined by:

(i) $\emptyset$ is a regular expression.

(ii) Any $a \in T$ is a regular expression.

(iii) If α, β are regular expressions then $(\alpha\beta)$ is a regular expression.

(iv) If α, β are regular expressions then $(\alpha \cup \beta)$ is a regular expression.

(v) If α is a regular expression then α^* is a regular expression.

Definition 2.14 *The language $L(\alpha) \subseteq T^*$ denoted by a regular expression* α over T is inductively defined by:

(i) $L(\emptyset) = \emptyset$.

(ii) $L(a) = \{a\}$ $(a \in T)$.

(iii) $L((\alpha\beta)) = L(\alpha)L(\beta)$, where $LL' = \{ww' : w \in L \;\&\; w' \in L'\}$ is the concatenation of the languages L, L'.

(iv) $L((\alpha \cup \beta)) = L(\alpha) \cup L(\beta)$.

(v) $L(\alpha^*) = L(\alpha)^*$, where $L^* = \{w_1 \ldots w_n : n \geq 1 \;\&\; w_1, \ldots, w_n \in L\} \cup \{\varepsilon\}$ is the iteration of L.

Theorem 2.15 *$L \subseteq T^*$ is regular iff there is a regular expression α over T such that $L = L(\alpha)$.*

The following lemma is useful for showing a language to be not regular.

Theorem 2.16 *(Pumping Lemma for Regular Languages) Let L be a regular language over the alphabet T. There are numbers $n, q \in \mathbb{N}$ such that for every word $z \in L$ with $|z| \geq n$ there is a partition $z = uvw$ of z into three words $u, v, w \in T^*$ such that*

1. $|uv| \leq q$,
2. $v \neq \varepsilon$, *and*
3. $uv^i w \in L$ *for all* $i \geq 0$.

2.2.3 Context Free Languages and Push Down Automata

We next turn to the context-free languages. The languages of this type are just the languages which are accepted by a nondeterministic pushdown automaton. We first introduce the deterministic variant of these machines before we give the general definition.

Definition 2.17 A *deterministic pushdown automaton (DPDA) M* is a 7-tupel $M = (T, \Gamma, S, \delta, s_0, b_0, F)$, where T and Γ are alphabets, S is a finite set, $s_0 \in S$, $b_0 \in \Gamma$, $F \subseteq S$, and

$$\delta : S \times (T \cup \{\varepsilon\}) \times \Gamma \rightarrow S \times \Gamma^*$$

is a partial function with the following property

$$\begin{array}{c}\text{For each } s \in S \text{ and } b \in \Gamma, \text{ whenever } \delta(s,\varepsilon,b) \text{ is defined,}\\ \text{then } \delta(s,a,b) \text{ is not defined for all } a \in T.\end{array} \tag{2.4}$$

T is called the *input alphabet*, Γ the *stack alphabet*, S the set of *states*, δ the *transition function*, s_0 the *initial state*, b_0 the *start symbol*, F the set of *final states*.

Intuitively, a *DPDA M* is the extension of a *DFA* by a stack as a storage device. A single move of M is as follows. Depending on the current state s, the next input letter a, and the top stack symbol b, M moves to a new state s' and replaces the top symbol b in the stack by a word w over the stack alphabet Γ. This move is expressed by the transition $\delta(s,a,b) = (s',w)$. In addition M may make some ε-move or spontaneous transition without reading the next input letter. Such a move is described by a transition $\delta(s,\varepsilon,b) = (s',w)$. Condition (2.4) guarantees that in any situation in which an ε-move is possible no regular move can be done thereby ensuring that the machine works deterministically. The content of the stack of M is represented by a word w over Γ where the rightmost symbol of w is the topmost symbol in the stack.

Just as in the case of finite automata we can easily generalize the concept of a *DPDA* to the nondeterministic case. We will do this next and then formally describe the behaviour of pushdown automata.

Definition 2.18 A *nondeterministic pushdown automata (NPDA) M* is a 7-tupel $M = (T, \Gamma, S, \Delta, s_0, b_0, F)$, where T and Γ are alphabets, S is a finite set, Δ is a relation of type

$$\Delta \subseteq [S \times (T \cup \{\varepsilon\}) \times \Gamma] \times (S \times \Gamma^*)$$

where, for any $(s,a,b) \in S \times (T \cup \{\varepsilon\}) \times \Gamma$ there are at most finitely many $(s',w) \in S \times \Gamma^*$ such that $(s,a,b,s',w) \in \Delta$, $s_0 \in S$, $b_0 \in \Gamma$, $F \subseteq S$. T is called the *input alphabet*, Γ the *stack alphabet*, S the set of *states*, Δ the *transition relation*, s_0 the *initial state*, b_0 the *start symbol*, F the set of *final states*.

Sometimes we will interpret the transition relation Δ as a function of type

$$\Delta : S \times (T \cup \{\varepsilon\}) \times \Gamma \to POWER(S \times \Gamma^*).$$

In particular, we let $\Delta(s,a,b) = \{(s',w) : (s,a,b,s',w) \in \Delta\}$.

We next formally describe the behaviour of an *NPDA*.

Definition 2.19 Let $M = (T,\Gamma,S,\Delta,s_0,b_0,F)$ be an *NPDA*.

An *instantaneous description (ID)* of M is a triple (s,w,v), where s is a state, w a string of input symbols, and v a string of stack symbols.

We write $(s,aw,bv) \vdash_M (s',w,uv)$ (and say that the ID (s,aw,bv) can be transformed into the ID (s',w,uv) in one move) if $(s',u) \in \Delta(s,a,b)$, where a may be ε or an input symbol.

$\vdash_M^*$ is the reflexive and transitive closure of $\vdash_M$, i.e., $I \vdash^* I$ for each *ID I*, and $I \vdash_M J$ and $J \vdash_M^* K$ imply $I \vdash_M^* K$. We say $I \vdash^i K$ if *ID I* can be transformed into K in exactly i moves.

The *language* $L(M) \subseteq T^*$ *accepted by final state* by the *NPDA M* is $L(M) = \{w | (s_0,w,b_0) \vdash^* (s,\varepsilon,v)$ for some $s \in F$ and $v \in \Gamma^*\}$.

The *language* $N(M) \subseteq T^*$ *accepted by empty stack (or null stack)* by M is $N(M) = \{w | (s_0,w,b_0) \vdash^* (s,\varepsilon,\varepsilon)$ for some $s \in S\}$.

In contrast to finite automata, for pushdown automata, the deterministic and nondeterministic models are not equivalent with respect to the languages accepted. Moreover, for *NPDAs* acceptance by state and acceptance by empty stack coincide, and the languages accepted by *NPDAs* are just the context-free languages.

Theorem 2.20 *For any language L the following are equivalent.*

1. *L is a context-free.*

2. *There exist an NPDA M such that* $L = L(M)$.

3. *There exist an NPDA M such that* $L = N(M)$.

A language L is called *deterministically context-free (dcf)* if it is accepted by states by a *DPDA M*, i.e., $L = L(M)$. The class of the dcf languages over T is denoted by DCF.

Theorem 2.21 DCF $\subset$ CF.

The following theorem is a useful tool for proving a variety of languages not to be context free (compare with the pumping lemma for regular languages).

Theorem 2.22 *(Pumping Lemma for Context-Free Languages) Let L be a context-free language over the alphabet T. There are numbers $n, q \in \mathbb{N}$ such that for every word $z \in L$ with $|z| \geq n$ there is a partition $z = uvwxy$ of z into five words $u, v, w, x, y \in T^*$ such that*

1. $|vwx| \leq q$,
2. $|vx| \neq \varepsilon$, *and*
3. $uv^i wx^i y \in L$ *for all* $i \geq 0$.

2.2.4 Turing Machines and Complexity

The stack of a pushdown automaton is not a general storage device. So, in order to define a general computing device, we have to replace the stack by a more flexible storage. In case of a Turing machine M the storage is a two-sided unbounded tape, partitioned into individual cells which can store one letter (of a given tape alphabet). The machine has a head which in one step can move one cell to the left (L) or one cell to the right (R) and which can read and rewrite the cell it is located on. A move of M is determined by the current state s of M and the letter a of the tape alphabet in the cell scanned by the head of M and it consists of a change of the state (s'), the rewriting of the cell currently scanned by a letter a', and a move of the head to the left (L) or right (R). In the following formal definition of the Turing machine M this move is describe by the transition $\delta(s,a) = (s', a', B)$ where $B = L$ or $B = R$.

Definition 2.23 A *deterministic Turing machine (TM) M* is a 7-tupel

$$M = (T, \Gamma, b, S, \delta, s_0, F)$$

as follows: T and Γ are alphabets where $b \in \Gamma$ and $T \subseteq \Gamma - \{b\}$, S is a finite set where S and Γ are disjoint, δ is a partial function of type

$$\delta : S \times \Gamma \to S \times \Gamma \times \{L, R\},$$

$s_0 \in S$, $F \subseteq S$. T is called the set of *input symbols*, Γ the set of *tape symbols*, b the *blank*, S the set of *states*, δ the *transition function*, s_0 the *initial state*, and F the set of *final states*.

In the above definition we use the blank symbol b to indicate that a cell is empty. If M processes an input word $w \in T^*$ then at the beginning of the computation w is written in the cells immediately to the right of the cell scanned by the head of M, all other cells are empty, and M is in the initial state s_0. The machine accepts the input w if it reaches a final state after finitely many moves.

An instantaneous description of M consists of the smallest finite sequence of cells (represented by a word w over the tape alphabet Γ) containing all nonempty cells and the cell scanned by the head, the position of the head on this sequence, and the current state s of M. We will code this information by the word w_0sw_1 where $w = w_0w_1$ and s is put to the right of the letter of w in the cell scanned by the head. (For convenience we will assume that w_0 and w_1 are nonempty. If necessary we achieve this by replacing the empty word ε by b.)

Using this coding we can define the behaviour of M formally as follows.

Definition 2.24 Let $M = (T,\Gamma,b,S,\delta,s_0,F)$ be a TM.

An *instantaneous description (ID)* of M is a word w_1sw_2, where $s \in S$ and $w_1,w_2 \in \Gamma^+$.

We write $w_1as\hat{a}w_2 \vdash_M w_1's'w_2'$ (and say that the ID $w_1as\hat{a}w_2$ can be transformed into the ID $w_1's'w_2'$ in one move) if either $\delta(s,a) = (s',a',R)$ and $w_1' = w_1a'\hat{a}$ and $w_2' = w_2$ or $\delta(s,a) = (s',a',L)$ and $w_1' = w_1$ and $w_2' = a'\hat{a}w_2$. (If w_1' (w_2') is empty, replace it by b.)

An *ID* vsw is called *terminal* if there is no *ID* $v's'w'$ such that $vsw \vdash_M v's'w'$ and vsw is called *final* if $s \in F$. (For technical convenience, in the following we will assume that any final *ID* will be terminal.)

$\vdash_M^*$ is the reflexive and transitive closure of $\vdash_M$, i.e., $I \vdash^* I$ for each *ID* I, and $I \vdash_M J$ and $J \vdash_M^* K$ imply $I \vdash_M^* K$. We say $I \vdash^i K$ if *ID* I can be transformed into K in exactly i moves.

The *language* $L(M) \subseteq T^*$ *accepted* by the Turing machine M is

$$L(M) = \{w \in T^* | \exists s \in F\ \exists v_1,v_2 \in \Gamma^+(bs_0w \vdash^* v_1sv_2)\}.$$

We say that M *converges* on input w if there is a terminal *ID* usv such that $bs_0w \vdash_M^* usv$; and M is called *total* if M converges on any input $w \in T^*$.

Definition 2.25 A language L is *recursively enumerable (r.e.)* if there is a Turing machine which accepts L, and L is *recursive* if there is a total Turing machine which accepts L.

Above we have introduced Turing machines as accepting devices for describing languages. Alternatively, we can use a Turing machine M as a device for computing a (partial) function $f : T^* \to T^*$, where $f(w)$ is defined if and only if M on input w terminates in a final state, and in this case $f(w)$ is the maximal word $v \in T^*$ written on the tape just to the right of the tape head.

Definition 2.26 A (partial) function $f : T^* \to T^*$ is *(partially) recursive* (or *(partially) Turing computable*) if there is a Turing machine which computes f.

By Church's Thesis, Turing machines are a universal computing device, i.e., a language is recursive (r.e.) if and only if it is decidable (effectively enumerable), and a (partial) function is (partially) recursive if and only if it is (partially) computable. So the standard extensions of Turing machines in the literature will not give stronger computing devices but possibly more efficient ones. For complexity matters we will consider the following two extensions of the above Turing machine model: nondeterministic machines and multi-tape machines. We do not define these models formally. The definition of a nondeterministic Turing machine is straightforward: The transition function δ has to be replaced by a transition relation Δ of type:

$$\Delta \subseteq (S \times \Gamma) \times (S \times \Gamma \times \{L,R\}).$$

A k-tape TM M ($k \geq 1$) has k tapes, each provided with a tape head, where the operations on the individual tapes are independent. Now a move of M is determined by the current state of M and the k letters scanned by the tape heads on the k tapes.

For the definition of the computational complexity classes based on the above Turing machine models we first introduce the running time and the space required by a TM M on input w, where we restrict ourselves to the case of a deterministic 1-tape TM.

Definition 2.27 Let M be a deterministic 1-tape TM.

1. If M on input w converges then the computation of M on input w is the unique finite sequence of IDs

$$Comp_M(w) = I_0, \dots, I_m$$

   where $I_0 = bs_0w$, $I_i \vdash_M I_{i+1}$ for $i < m$, and I_m is a terminal ID.

2. The *running time* of M is the partial function $time_M : T^* \to \mathbb{N}$ where

$$time_M(w) = \begin{cases} length(Comp_M(w)) & \text{if } M \text{ terminates on input } w \\ \uparrow & \text{otherwise.} \end{cases}$$

3. Let $t : \mathbb{N} \to \mathbb{N}$ be a total recursive function. Then M is $t(n)$*-time bounded* if

$$\forall w \in T^*(time_M(w) \leq t(|w|)).$$

4. The *space required* by M is the partial function $space_M : T^* \to \mathbb{N}$ where

$$space_M(w) = \begin{cases} max\{|I| : I \in Comp_M(w)\} & \text{if } M \text{ terminates on input } w \\ \uparrow & \text{otherwise.} \end{cases}$$

5. Let $s : \mathbb{N} \to \mathbb{N}$ be a total recursive function. Then M is $s(n)$-*space bounded*, if
$$\forall w \in T^*(space_M(w) \leq s(|w|)).$$

The following complexity classes are defined for multi-tape Turing machines. Let $t : \mathbb{N} \to \mathbb{N}$ be a total recursive function. Then the *deterministic time complexity class with bound* $t(n)$ (over the binary alphabet Σ) is defined by

$$\text{DTIME}(t(n)) = \{L \subseteq \Sigma^* : \exists \text{ det. } t(n)\text{-time-bounded TM } M(L(M) = L)\}.$$

The corresponding *nondeterministic time class* is defined by

$$\text{NTIME}(t(n)) = \{L \subseteq \Sigma^* : \exists \text{ nondet. } t(n)\text{-time-bounded TM } M(L(M) = L)\}.$$

Similarly, for a total recursive function $s : \mathbb{N} \to \mathbb{N}$, the *deterministic space complexity class with bound* $s(n)$ is defined by

$$\text{DSPACE}(s(n)) = \{L \subseteq \Sigma^* : \exists \text{ det. } s(n)\text{-space-bounded TM } M(L(M) = L)\}$$

and the corresponding *nondeterministic space class* is defined by

$$\text{NSPACE}(s(n)) = \{L \subseteq \Sigma^* : \exists \text{ nondet. } s(n)\text{-space-bounded TM } M(L(M) = L)\}.$$

If we admit only Turing machines with a fixed number of tapes then we add the number k of tapes as an index to the name of the class. For instance,

$$\text{DTIME}_1(t(n)) = \{L \subseteq \Sigma^* : \exists \text{ det. } t(n)\text{-time-bounded 1-tape-TM } M(L(M) = L)\}.$$

We will use the following results on the above Turing machine complexity classes: By the *linear-speed-up theorem*, the above classes are invariant under linear changes of the bound, i.e.,

$$\text{C}(f(n)) = \text{C}(O(f(n))$$

(for $\text{C} = \text{DTIME}, \text{DTIME}_k, \text{DSPACE},$ etc.). By the *tape-reduction theorem*,

$$\text{D(N)SPACE}_1(f(n)) = \text{D(N)SPACE}(f(n))$$

and
$$\text{D(N)TIME}(f(n)) \subseteq \text{D(N)SPACE}_1(O(f(n))^2).$$

For the comparison of the different complexity measures we have the following results:
$$\text{DTIME}(f(n)) \subseteq \text{NTIME}(f(n)),$$
$$\text{DTIME}(f(n)) \subseteq \text{DSPACE}(f(n)) \subseteq \text{NSPACE}(f(n)),$$

$$\text{NTIME}(f(n)) \subseteq \text{NSPACE}(f(n)),$$

and

$$\text{NSPACE}(f(n)) \subseteq \text{DTIME}(2^{O(f(n))}).$$

Finally, by *Savitch's Theorem*,

$$\text{NSPACE}(f(n)) \subseteq \text{DSPACE}(f(n)^2)$$

for space constructible bounds f, where a function f is space (time) constructible if there is a Turing machine M which on any input w of length n uses exactly space of size $f(n)$ (has running time $f(n)$). Finally we will use the fairly recent result that the nondeterministic space classes are closed under complement.

Theorem 2.28 *(Immerman-Szelepcsenyi) For space constructible bounds $s(n) \geq \log(n)$,*

$$\text{NSPACE}(s(n)) = \text{co} - \text{NSPACE}(s(n))$$

Here the co-class co-C of a complexity class C is the class of the complements of the languages in C, i.e., co-C $= \{\overline{A} : A \in \text{C}\}$. The Theorem of Immerman and Szelepcsenyi has been used to prove to the following hierarchy theorem for nondeterministic space classes.

Corollary 2.29 *(Nondeterministic Space Hierarchy Theorem) Let s and S be recursive functions such that $S(n) \notin O(s(n))$ and $S(n)$ is space constructible. Then* $\text{NSPACE}(S(n)) \not\subseteq \text{NSPACE}(s(n))$.

The Theorem of Immerman and Szelepcsenyi also gave the solution to a open problem of formal language theory namely the question whether the complement of every context-sensitive language is context-sensitive. An affirmative answer to this question follows from the coincidence of the class of the context-sensitive languages with the nondeterministic linear space class.

Theorem 2.30 $\text{CS} = \text{NSPACE}_1(n) = \text{NSPACE}(O(n))$

Sometimes this theorem is stated in a slightly different form using the notion of a (nondeterministic) *linearly bounded automaton* ((N)LBA). An LBA may be viewed as a 1-tape Turing machine which is not allowed to leave the cells bearing the input. For this sake the input is limited by end markers [and] and the head is not allowed to pass beyond these markers. The class of languages recognized by nondeterministic LBAs is denoted by NLBA. Obviously, $\text{NLBA} = \text{NSPACE}_1(n)$. Hence, by Theorem 2.30, a language is context-sensitive if and only if it is accepted by an NLBA, i.e.,

$$\text{CS} = \text{NLBA} \tag{2.5}$$

Note that any complexity class is countable and consists only of recursive languages. In fact the languages in such a class have a uniform recursive presentation. We will conclude this subsection by reviewing the important notion of uniform computability.

Definition 2.31 A countable class C of binary languages is *uniformly recursive* or *recursively presentable* (r.p. for short) if there is a binary recursive set $U \subseteq \mathbb{N} \times \Sigma^*$ such that $\mathrm{C} = \{U^{[n]} : n \geq 0\}$ where $U^{[n]} = \{x : (n,x) \in U\}$. U is called a *universal set* for C.

Theorem 2.32 *For any recursive function f, the complexity classes* $\mathrm{DTIME}(f(n))$, $\mathrm{NTIME}(f(n))$, $\mathrm{DSPACE}(f(n))$, *and* $\mathrm{NSPACE}(f(n))$ *are uniformly recursive. Moreover, the Chomsky classes* REG, LIN, CF *and* CS *are uniformly recursive.*

2.3 Regular Functions

In the preceding section we have reviewed the Chomsky language classes and the types of grammars and automata which characterize these classes. In case of Turing machines - which describe the most general Chomsky class - we have also indicated how this concept has been used to formalize the notion of a computable function. The finite-state genericity notions we will introduce in this thesis will be based on regular extension function. So, in order to introduce these notions, we have to review the approaches which have been used to define the concept of a regular function based on finite automata. We will restrict ourselves to the case of word functions over the binary alphabet Σ, since only functions of this type will be needed later. For references see e.g. Yu (1997).

The probably most common definition of a regular word function f of type $f: T_1^* \to T_2^*$ is due to Moore. It is based on an extension of the concept of a deterministic finite automaton where the states of the automaton are labeled by letters from the output alphabet T. Then the value of the computed function f on input x is the word $\lambda(s_0)...\lambda(s_n)$ where $s_0,...,s_n$ are the states visited by the automaton while reading input x. (In case of a partial function, this output is only given if the final state s_n is accepting.) We next formally introduce this concept (where, as remarked before, we limit ourselves to the case of the binary alphabet Σ).

Definition 2.33 A *Moore automaton* $M = (\Sigma, S, \delta, s_0, F, \lambda)$ is a deterministic finite automaton $M = (\Sigma, S, \delta, s_0, F)$ together with a *labelling function* $\lambda : S \to \Sigma$. The *(partial) Moore function* $f_M : \Sigma^* \to \Sigma^*$ computed by M is defined by

$$f_M(a_1...a_n) = \lambda(s_0)\lambda(\delta^*(s_0,a_1))\lambda(\delta^*(s_0,a_1a_2))...\lambda(\delta^*(s_0,a_1...a_n))$$

if $\delta^*(s_0,a_1...a_n) \in F$ (where $n \geq 0$), and $f_M(a_1...a_n) \uparrow$ otherwise. A (partial) function f is a *(partial) Moore function* if $f = f_M$ for some Moore automaton M.

If we consider total Moore functions then w.l.o.g. we may assume that all states are final or, in other words, we may omit the set of final states from the definition of the automaton. Note that, for a (partial) Moore function f, the length of the output is the length of the input increased by 1 (if defined) and f is monotonous in the sense that if y extends x and $f(x)$ and $f(y)$ are both defined then $f(y)$ extends $f(x)$.

Lemma 2.34 *Let f be a (partial) Moore function. Then, for $v, w \in \Sigma^*$,*

$$f(w)\downarrow \Rightarrow |f(w)| = |w| + 1 \tag{2.6}$$

and

$$(v \sqsubseteq w \;\&\; f(v)\downarrow \;\&\; f(w)\downarrow) \;\Rightarrow\; f(v) \sqsubseteq f(w). \tag{2.7}$$

PROOF. Immediate by definition. □

We can obtain a more general concept of a regular function by admitting labelling functions of a more general type. Instead of attaching a single letter to each state we now attach a word.

Definition 2.35 A *generalized Moore automaton* $M = (\Sigma, S, \delta, s_0, F, \lambda)$ is a deterministic finite automaton $M = (\Sigma, S, \delta, s_0, F)$ together with a *labelling function* $\lambda : S \to \Sigma^*$. The *(partial) generalized Moore function* $f_M : \Sigma^* \to \Sigma^*$ computed by M is defined by

$$f_M(a_1...a_n) = \lambda(s_0)\lambda(\delta^*(s_0,a_1))\lambda(\delta^*(s_0,a_1a_2))...\lambda(\delta^*(s_0,a_1...a_n))$$

if $\delta^*(s_0, a_1...a_n) \in F$ (where $n \geq 0$), and $f_M(a_1...a_n)\uparrow$ otherwise. A (partial) function f is a *(partial) generalized Moore function* if $f = f_M$ for some generalized Moore automaton M.

Again, if we are only interested in total generalized Moore functions we omit the set of final states from the definition. Obviously any (partial) Moore function is a (partial) generalized Moore function (but not vice versa). As in case of Moore functions, generalized Moore functions are monotonous but the length property (2.6) in general fails. Now we may only argue that the length of $f(x)$ (if defined) is linearly bounded in the length of x.

Lemma 2.36 *Let f be a (partial) generalized Moore function. Then* (2.7) *holds and there is a constant c such that, for any $w \in \Sigma^*$,*

$$f(w)\downarrow \Rightarrow \; |f(w)| \leq c|w| \tag{2.8}$$

PROOF. Immediate. □

Example 2.37 An example of a generalized Moore function $f : \Sigma^* \to \Sigma^*$ which is not a Moore function is the function $f : \Sigma^* \to \Sigma^*$ defined by $f(x) = 0^{2|x|+2}$. A generalized Moore automaton which computes f has only one state s_0 labelled by $\lambda(s_0) = 0^2$. f is not a Moore function since the length condition (2.6) is not satisfied.

We can further generalize the notion of a (generalized) Moore function by considering nondeterministic (generalized) Moore automata. In general such a nondeterministic automaton will not compute a function of type $\Sigma^* \to \Sigma^*$ since the different possible runs of a nondeterministic automaton on a given input may result in different function values.

Definition 2.38 A *nondeterministic (generalized) Moore automaton*

$$M = (\Sigma, S, \Delta, S_0, F, \lambda)$$

is a nondeterministic finite automaton $M = (\Sigma, S, \Delta, S_0, F)$ together with a *labelling function* $\lambda : S \to \Sigma$ ($\lambda : S \to \Sigma^*$). The automaton M is *complete* if for any $x \in \Sigma^*$ there is a run of M ending in a final state and M is *consistent* (or *single valued*) if for any input $x \in \Sigma^*$ and any two possible runs $s_0, \ldots, s_n$ and $s'_0, \ldots, s'_n$ of M on input x which end in final states, $\lambda(s_0) \ldots \lambda(s_n) = \lambda(s'_0) \ldots \lambda(s'_n)$. If M is consistent then the *(partial) nd. (generalized) Moore function* $f_M : \Sigma^* \to \Sigma^*$ computed by M is defined by

$$f_M(x) = \lambda(s_0)\lambda(s_1)\lambda(s_2)\ldots\lambda(s_n)$$

where $s_0, \ldots, s_n$ is any run of M on input x with $s_n \in F$ (if such a run exists, and $f_M(x) \uparrow$ otherwise). A (partial) function f is a *(partial) nd. (generalized) Moore function* if $f = f_M$ for some consistent nondeterministic (generalized) Moore automaton M.

As one can easily check, this definition is consistent, i.e., the (partial) function computed by a consistent n.d. (generalized) Moore automaton is well defined. Moreover, the function f_M computed by such an automaton is total if and only if M is complete. Moreover, for a partial n.d. (generalized) Moore function f, the corresponding length conditions from the deterministic case ((2.6) and (2.8), respectively) still hold but monotonicity (see (2.7)) in general fails.

Lemma 2.39 *For any (partial) n.d. Moore function f,* (2.6) *holds and, for any (partial) n.d. generalized Moore function f,* (2.8) *holds.*

PROOF. Immediate by definition. □

Example 2.40 The function $f : \Sigma^* \to \Sigma^*$ defined by

$$f(x) = \begin{cases} 0^{|x|+1} & \text{if } |x| \text{ is even} \\ 1^{|x|+1} & \text{if } |x| \text{ is odd} \end{cases}$$

is a n.d. Moore function. An n.d. Moore automaton $M = (\Sigma, S, \Delta, S_0, F, \lambda)$ which computes f can be defined as follows.

$$S = \{s_{0,0}, s_{0,1}, s_{1,0}, s_{1,1}\}$$

$$(s_{i,j}, k, s_{i',j'}) \in \Delta \Leftrightarrow i = i' \;\&\; j \neq j' \;\; (i, i', j, j', k \in \{0,1\})$$

$$S_0 = \{s_{0,0}, s_{1,0}\}$$

$$F = \{s_{0,0}, s_{1,1}\}$$

$$\lambda(s_{i,j}) = i \;\; (i, j \in \{0,1\})$$

Intuitively, M works as follows. On input x, M nondeterminstically guesses whether the length of x is even or odd. If M guesses that $|x|$ is even then it starts the computation in state $s_{0,0}$, otherwise in state $s_{1,0}$. If M starts in state $s_{0,0}$ then in the run of M the states $s_{0,0}$ and $s_{0,1}$ alternate where both states are assosciated with the label 0. Since the state $s_{0,0}$ is final but $s_{0,1}$ is not, a run beginning in state $s_{0,0}$ is accepting iff $|x|$ is even. Moreover, in this case, the output is $0^{|x|+1}$. Similarly a run beginning in state $s_{1,0}$ will be accepting iff $|x|$ is odd, and in this case the output will be $1^{|x|+1}$.

Note that f is not a (generalized) Moore function since f does not satisfy the extension property (2.7).

The above examples and lemmas show that the only relations among the different types of Moore functions are the trivial relations

$$\begin{array}{ccc} f \text{ (partial) Moore function} & \Rightarrow & f \text{ (partial) gen. Moore function} \\ \Downarrow & & \Downarrow \\ f \text{ (partial) n.d. Moore function} & \Rightarrow & f \text{ (partial) n.d. gen. Moore function} \end{array} \tag{2.9}$$

To see that no other relations hold it suffices to recall that the function of Example 2.37 is a generalized Moore function but, by failure of (2.6) and by Lemma 2.39, not a n.d. Moore function and that the function of Example 2.40 is a n.d. Moore function but not a generalized Moore function.

In the literature there are other approaches for defining regular functions of type $f : \Sigma^* \to \Sigma^*$: While in the Moore approach in any step of the computation the output is expanded by a letter (or – in the generalized case – by a word) where the chosen letter (word) depends on the current state, in the Mealy approach, the letter (word) appended depends not only the current state but on the current transition. Though aparently more flexible, it has been shown that the Mealy approach is not more powerful than the Moore approach and leads (essentially) to the same class of functions. (To be more precise, we can obtain any Mealy function f by dropping the first letter from the output of some Moore function f'.) The same observation applies to the corresponding nondeterministic functions. A further generalization of the Mealy approach has been proposed which is based on so-called generalized sequential machines: Here the machine is allowed to make ε-moves, i.e., to make a transition without reading a new letter, and to read more than one letter in one move. As one can easily show, the functions f obtained this way are essentially Mealy functions by differing from the latter only on the empty string. (See e.g. Yu (1997) for more details.) So for our applications it will suffice to deal with the above introduced variants of Moore functions.

Some of our finite-state genericity concepts will be based on extension functions of constant length, however, i.e. will be based on (partial) functions of type

$\Sigma^* \to \Sigma^k$ (for any given $k \geq 1$) which can be computed by a finite automaton. So in the remainder of this subsection we introduce a notion of regularity for (partial) regular functions of this type. This notion will be based on generalized Moore automata where each state is labeled with a word of length k. The value of the function will be defined iff the computation ends in a final state s and in this case the value will be the label of this state. Next we formally introduce this concept.

Definition 2.41 A *k-labelled automaton M* is a generalized Moore automaton $M = (\Sigma, S, \delta, s_0, F, \lambda)$ where $|\lambda(s)| = k$ for any $s \in S$ $(k \geq 1)$. The *(partial) function $f_M : \Sigma^* \to \Sigma^k$ computed by* the k-labelled automaton $M = (\Sigma, S, \delta, s_0, F, \lambda)$ is defined by

$$f_M(x) = \begin{cases} \lambda(\delta^*(s_0, x)) & \text{if } \delta^*(s_0, x) \in F \\ \uparrow & \text{otherwise.} \end{cases}$$

A (partial) function $f : \Sigma^* \to \Sigma^k$ is *regular* if f is computed by some deterministic k-labelled automaton.

We can extend this definitions to nondeterministic automata.

Definition 2.42 An *n.d. k-labelled automaton M* is a n.d. generalized Moore automaton $M = (\Sigma, S, \Delta, S_0, F, \lambda)$ where $|\lambda(x)| = k$ for any $s \in S$ $(k \geq 1)$. M is *consistent* (or *single valued*) if, for any word x and for any states s and s',

$$s, s' \in \Delta^*(S_0, x) \cap F \Rightarrow \lambda(s) = \lambda(s')$$

holds. The *(partial) function $f_M : \Sigma^* \to \Sigma^k$ computed by* the consistent n.d. k-labelled automaton $M = (\Sigma, S, \Delta, S_0, F, \lambda)$ is defined by $f_M(x) = \lambda(s)$ where s is any final state such that $s \in \Delta^*(S_0, x)$ if such a state s exists, and $f_M(x) \uparrow$ otherwise.

In contrast to the Moore function concept, however, here nondeterminism does not lead to a more powerful concept.

Lemma 2.43 *Let $f : \Sigma \to \Sigma^k$ be a partial function which is computed by a consistent n.d. k-labelled automaton. Then f is regular.*

PROOF. Fix a consistent n.d. k-labelled automaton $M = (\Sigma, S, \Delta, S_0, F, \lambda)$ such that $f = f_M$. We have to define a deterministic k-labelled automaton M' such that $f_M = f_{M'}$. The definition of $M' = (\Sigma, S', \delta', s_0{}', F', \lambda')$ is based on the standard power set construction for giving a deterministic automaton (without output) simulating a given n.d. automaton. To be more precise, we let

$$S' = \{\sigma : \sigma \subseteq S\}$$

$$\delta'(\sigma, a) = \Delta(\sigma, a)$$

$$s_0{}' = S_0$$

$$F' = \{\sigma : \sigma \cap F \neq \emptyset\}$$

Then, for any $x \in \Sigma^*$, x is accepted by M iff M is accepted by M' whence the domains of f_M and $f_{M'}$ agree. To ensure that $f_M(x) = f_{M'}(x)$ whenever defined, it remains to set $\lambda'(\sigma) = \lambda(s)$ for any $s \in \sigma \cap F$ if $\sigma \in F'$ and by letting $\lambda'(\sigma)$ have any value, say 0^k, otherwise. Note that, by consistency of M, λ' is well defined, and for any x such that $f_M(x)$ is defined, $f_{M'}(x) = f_M(x)$. □

Note that, by (2.6), no (partial) function of type $\Sigma^* \to \Sigma^k$ is a (partial) – deterministic or n.d. – Moore function. There are generalized Moore functions of this type, however, whence it is natural to ask what are the relations between regular functions and (deterministic or nondeterministic) generalized Moore functions of type $\Sigma^* \to \Sigma^k$. As we will show next the regular functions are just the n.d. generalized Moore functions.

Lemma 2.44 *Let $k \geq 1$ and let $f : \Sigma^* \to \Sigma^k$ be any (partial) function. The following are equivalent.*

(i) f is regular.

(ii) f is an n.d. generalized Moore function.

PROOF. For a proof of $(i) \Rightarrow (ii)$ let $M = (\Sigma, S, \delta, s_0, F, \lambda)$ be a deterministic k-labelled automaton which computes f. We have to define a consistent n.d. generalized Moore automaton $M' = (\Sigma, S', \Delta', S_0{}', F', \lambda')$ which is equivalent to M, i.e., such that $f_{M'} = f_M$. Note that the function value $f_M(x)$ computed by the k-labelled automaton M is the label $\lambda(s)$ of the state s reached by M after having read the input x. In contrast, $f_{M'}(x)$ is the concatenation of the labels of all states in the run of M' on input x. So, in order to simulate M, M' has to delete the labels of the intermediate states. This is achieved by having two copies s and $\bar{s}$ of any state s of M, where one copy ($\bar{s}$) has the same label attached as in M and one (s) has attached the empty word. Then M' copies the run of M by using the copies of the states with empty label. Only when M' guesses that the input is completely read, it switches to the copy with the original label.

Formally, M' is defined by

$$S' = S \cup \{\bar{s} : s \in S\}$$

$$\Delta'(s, a) = \{\delta(s, a), \overline{\delta(s, a)}\} \ \& \ \Delta'(\bar{s}, a) = \emptyset \quad (s \in S, a \in \Sigma)$$

$$S_0{}' = \{s_0, \overline{s_0}\}$$
$$F' = \{\bar{s} : s \in F\}$$
$$\lambda'(s) = \varepsilon \;\&\; \lambda'(\bar{s}) = \lambda(s) \quad (s \in S).$$

For a proof of $(ii) \Rightarrow (i)$ let $M = (\Sigma, S, \delta, s_0, F, \lambda)$ be a consistent n.d. generalized Moore automaton which computes f. By Lemma 2.43, it suffices to define a consistent n.d. k-labelled automaton $M' = (\Sigma, S', \Delta', S_0{}', F', \lambda')$ which is equivalent to M, i.e., such that $f_M = f_{M'}$. Intuitively, such an automaton M' simulates M step by step where M' remembers in his state the concatenation of the labels attached to the visited M-states up to this point. (This is possible since the generalized Moore function computed by M is k-bounded, i.e., it suffices to store a word of length at most k. If the length of the word exceeds k then the input will not be accepted and the computation can be aborted.) When M' guesses that the input is read completely, it outputs the sequence of these labels (by attaching it to the state reached in the end).

Formally, M' is defined by

$$S' = \{[s,x] : s \in S \;\&\; x \in \Sigma^{\leq k}\}$$
$$([s',y] \in \Delta'([s,x],a) \Leftrightarrow s' \in \Delta(s,a) \;\&\; y = x\lambda(s') \;\&\; |y| \leq k$$
$$S_0{}' = \{[s_0, \lambda(s_0)] : s_0 \in S_0\}$$
$$F' = \{[s,x] : s \in F \;\&\; |x| = k\}$$
$$\lambda'([s,x]) = x0^{k-|x|}.$$

□

The observation that not any n.d. generalized Moore function is a generalized Moore function can be extended to functions of type $\Sigma^* \to \Sigma^k$. So, by the preceding lemma, not every regular function $f : \Sigma^* \to \Sigma^k$ is a generalized Moore function.

Lemma 2.45 *For any $k \geq 1$ there is a regular function $f : \Sigma^* \to \Sigma^k$ such that f is not a generalized Moore function.*

PROOF. Define $f : \Sigma^* \to \Sigma^k$ by letting

$$f(x) = \begin{cases} 0^k & \text{if } |x| \text{ is even} \\ 1^k & \text{if } |x| \text{ is odd.} \end{cases}$$

Then f is regular. (A k-labelled automaton M which computes f has two states s_0 and s_1, where s_0 is the initial state, both states are final, and s_i is labelled with i^k. In any step M moves from the current state s_i to the other state s_{1-i}.) By Lemma

2.36, however, f is not a generalized Moore function since the extension property (2.7) is not satisfied. □

For later use we observe that the class of regular functions is closed under finite variants.

Lemma 2.46 *The class of the (partial) regular functions of type $\Sigma^* \to \Sigma^k$ ($k \geq 1$) is closed under finite variants.*

PROOF. We consider the case of total functions. (The case of partial functions is similar.) It suffices to consider a pair of functions which differs on a single argument. The general case follows by induction on the number of differences.

So assume that $f : \Sigma^* \to \Sigma^k$ is regular and $f' : \Sigma^* \to \Sigma^k$ differs from f on input x. We have to show that f' is regular too. Fix a k-labelled automaton $M = (\Sigma, S, \delta, s_0, F, \lambda)$ which computes f. Then a k-labelled automaton M' which computes f' can be obtained as follows. $M' = (\Sigma, S', \delta', s_0', F', \lambda')$ simulates M step by step. In addition, as long as the part y of the input word read so far is an initial segment of x, this information is stored in the state of M' and if $y = x$ then the label of the current state is replaced by $f'(x)$. Formally, for $s \in S$, $a \in \Sigma$, and $y \sqsubseteq x$,

$$S' = S \cup \{[s,y] : s \in S \;\&\; y \sqsubseteq x\}$$

$$\delta'(s,a) = \delta(s,a)$$

$$\delta'([s,y],a) = \begin{cases} [\delta(s,a), ya] & \text{if } ya \sqsubseteq x \\ \delta(s,a) & \text{otherwise} \end{cases}$$

$$s_0{}' = [s_0, \varepsilon]$$

$$F' = F \cup \{[s,y] : s \in F \;\&\; y \sqsubseteq x\}$$

$$\lambda'(s) = \lambda(s)$$

$$\lambda'([s,y]) = \begin{cases} \lambda(s) & \text{if } y \neq x \\ f'(x) & \text{if } y = x. \end{cases}$$

□

2.4 Strong Separations in the Chomsky Hierarchy

Immunity and bi-immunity are among the fundamental concepts in computability theory and computational complexity theory. An infinite language A is *immune* to a class C (C-*immune* for short) if it does not contain any infinite member of this class as a subset. If both, a language A and its complement $\overline{A}$ are immune to C then A is called C-*bi-immune*. Post (1944) introduced immunity and proved the existence of nonrecursive (many-one) incomplete recursively enumerable problems by constructing a *simple* set, i.e., an r.e. set whose complement is immune to the class of r.e. sets. Flajolet and Steyaert (1974) were probably the first who studied immunity in the context of formal language theory. For instance they have observed that the canonical examples $\{0^n1^n : n \geq 1\}$ and $\{0^n1^n0^n : n \geq 1\}$ of non-regular and non-context-free languages are in fact REG-immune and CF-immune respectively. Bi-immunity was introduced by Balcázar and Schöning (1985) who have also observed the close connection between bi-immunity and almost-everywhere complexity. In the sequel some hierarchy theorems for almost-everywhere complexity have been proven using this concept (see e.g. Geske et al. (1987)). Moreover, for any countable class C, the class of C-bi-immune sets has measure 0 and is comeager whence these concepts are of interest for the investigation of randomness and genericity notions.

In this subsection, by extending the work in computability and complexity theory and the work of Flajolet and Steyaert on immunity in formal language theory, we will present some fundamental properties and relations of the immunity and bi-immunity notions for the classes of the Chomsky hierarchy. We will proceed as follows. First we will show that the coincidence of almost-everywhere complexity and bi-immunity in complexity theory has its counterpart in formal language theory. Then we will present some general definitions and results before we will look at the individual levels of the Chomsky hierarchy where our focus will be on bi-immunity to the lower classes of the hierarchy, namely the classes of regular, linear, and context-free languages which haven't been considered in detail before.

2.4.1 Bi-Immunity and Almost Everywhere Complexity

In computational complexity theory bi-immunity has been extensively studied since it is closely related to almost-everywhere complexity (see Balcázar et al. (1990), Chapter 6 for details). For instance, if C is a determinstic time class DTIME$(t(n))$ and A is DTIME$(t(n))$-bi-immune then, for any Turing machine M which accepts A, $time_M(x) > t(|x|)$ for all but finitely many strings x. Alternatively, we can express this observation as follows: Call M an *extended* Turing machine if M has three types of states, accepting states ($+$-states), rejecting states ($-$-states) and undetermined states (?-states). Moreover, say that M *decides* (*accepts*; *rejects*) x if the computation of M on input x ends in a $+$-state or $-$-state ($+$-state; $-$-state) and say that M is *consistent* with A if M only accepts strings $x \in A$ and M only rejects strings $x \in \overline{A}$. Then A is DTIME$(t(n))$-bi-immune if and only if any deterministic $t(n)$-time bounded extended Turing machine M which is consistent with A decides only finitely many strings x.

For the classes C in the Chomsky hierarchy we can obtain similar characterizations of C-bi-immunity by considering the machine characterizations of these classes. We demonstrate this for the class REG of the regular languages here. We first formally define the notion of an extended deterministic finite automaton.

Definition 2.47 An *extended deterministic finite automaton* (EDFA) M is a 7-tuple $M = (\Sigma, S, \delta, s_0, S_+, S_-, S_?)$ where (Σ, S, δ, s_0) is a deterministic finite automaton without a distinguished set of final states and $(S_+, S_-, S_?)$ is a partition of the set S of states of M, called the sets of $+$-*states* (or *accepting states*), $-$-*states* (or *rejecting states*) and ?-*states* (or *undetermined states*), respectively. The EDFA M *accepts* (*rejects*) $w \in \Sigma^*$ if the computation of M on input w ends in a $+$-state ($-$-state), i.e., if $\delta^*(s_0, w) \in S_+$ ($\delta^*(s_0, w) \in S_-$). If M accepts or rejects w then we also say that M *decides* w. The EDFA M is *consistent* with a language A if M only accepts strings $x \in A$ and M only rejects strings $x \in \overline{A}$.

Theorem 2.48 *For any language A the following are equivalent.*

(i) A is REG-*bi-immune.*

(ii) For any extended deterministic finite automaton M which is consistent with A, M decides only a finite number of strings.

PROOF. The proofs of both implications are by contraposition. For a proof of $(i) \Rightarrow (ii)$ assume that M is an extended deterministic finite automaton which is consistent with A and which decides an infinite number of strings. Then, by symmetry, w.l.o.g. there are infinitely many strings which are accepted by M. So, if we convert M into a standard finite automaton M' by letting the $+$-states of M be the final states of M', then M' accepts an infinite set B and, by consistency of M with

A, B is contained in A. So there is an infinite regular subset B of A whence A is not REG-bi-immune.

For a proof of $(ii) \Rightarrow (i)$ assume that A is not REG-bi-immune. Then, by symmetry, w.l.o.g. we may assume that A contains an infinite regular set B as a subset. Given a deterministic finite automaton M which accepts B, convert M into an extended automaton M' by letting the final states of M be the $+$-states of M' and by letting the non-final states be ?-states. (There are no $-$-states.) Then, by $B \subseteq A$, M' is consistent with A and, by infinity of B, M' decides an infinite number of strings. □

By the above, REG-bi-immune sets are non-regular in a very strong sense. So, for investigating the power of diagonalizations over the class REG, REG-bi-immunity is a very interesting property.

2.4.2 Definitions and Basic Facts

We next look at some basic definitions and results on immunity and bi-immunity in a general setting. We first review the definitions of immunity, co-immunity and bi-immunity.

Definition 2.49 Let C be any class of sets. A set A is *immune to* C, C-*immune* for short, if A is infinite but no infinite subset of A is a member of C; A is C-*co-immune* if the complement $\overline{A}$ of A is C-immune; and A is C-*bi-immune* if A and $\overline{A}$ are C-immune.

The following observations are obvious.

Proposition 2.50 *(a) If A is* C*-immune then $A \notin$* C.

(b) If A is C*-co-immune and* C *is closed under complement then $A \notin$* C.

(c) If $C_0 \subseteq C_1$ *and A is* C_1*-(co/bi-)immune then A is* C_0*-(co/bi-)immune.*

(d) A is C*-bi-immune iff A is* C*-immune and* C*-co-immune.*

Proposition 2.51 *Let* C *be a class which contains all co-finite languages. Then the following are equivalent.*

(i) A is C*-bi-immune.*

(ii) Neither A nor $\overline{A}$ contains an infinite set $B \in$ C *as a subset.*

PROOF. By definition, (i) is a strengthening of (ii), namely (i) is obtained from (ii) by additionally requiring that A and $\overline{A}$ are infinite. So, given a set A satisfying (ii), it suffices to show that A and $\overline{A}$ are infinite. We do this for A. The proof for $\overline{A}$ is symmetric. For a contradiction assume that A is finite. Then $\overline{A}$ is co-finite. By choice of C this implies that that $\overline{A} \in$ C. So, for $B = \overline{A}$, B is infinite, $B \in$ C and $B \subseteq \overline{A}$. But this contradicts the assumption that A satisfies (ii). □

In the following we will tacitly use Propositions 2.50 and 2.51. We next turn to some general existence results for (bi-)immune sets. It has been shown, by diagonalization, that, for any countable class C, C-bi-immune sets exists. In fact, it is well known that for such classes C, typical sets are C-bi-immune. (See Chapter 3 below for more details. There also a proof of the following theorem can be found.)

Theorem 2.52 *For any countable class* C *there are* C*-bi-immune sets. In fact the class of* C*-bi-immune sets has Lebesgue measure 1 and is co-meager.*

For countable classes C which have an infinite and co-infinite member, there are C-immune sets which are not C-bi-immune. In order to show this we first prove the following fact.

Theorem 2.53 *Let* C *be a countable class and let B be an infinite set. There is a subset A of B which is* C*-immune.*

PROOF. Let C $= \{C_n : n \geq 0\}$ be a (possibly noneffective) enumeration of C. We define a set A with the required properties by a finite extension argument. I.e., we define A in stages $s \geq 0$ by simultaneously defining a strictly increasing function l, $l(s)$ beeing defined at stage s, such that the part A_s of A defined by the end of stage s will consist of all strings in A of length less than $l(s)$, i.e., $A_s = A \cap \Sigma^{<l(s)}$. At an even stage $2s$ we ensure that C_s is not contained in A unless C_s is finite; at an odd stage $2s+1$ we ensure that A contains a string of length at least $2s+1$ thereby ensuring that A is infinite.

Formally, A is defined as follows. Given A_{s-1} and $l(s-1)$ (where $A_{-1} = \emptyset$ and $l(-1) = 0$), A_s and $l(s)$ are defined as follows. If s is even, say $s = 2e$, then distinguish the following two cases. If C_e is infinite then let x_s be the least string x in C_e such that $|x| \geq l(s-1)$, let $l(s) = |x_s| + 1$ and set $A_s = A_{s-1}$. If C_s is finite then let $l(s) = l(s-1)+1$ and set $A_s = A_{s-1}$. Finally, if s is odd then let x_s be the least string x in B such that $|x| \geq l(s-1)$, let $l(s) = |x_s| + 1$ and set $A_s = A_{s-1} \cup \{x_s\}$.

The correctness of the construction easily follows from the remarks preceding the construction. □

Corollary 2.54 *Let* C *be a countable class which has an infinite and co-infinite member. There is a* C*-immune set A which is not* C*-bi-immune. In fact, if* C_0 *and* C_1 *are countable classes such that* $\mathrm{C}_0 \subseteq \mathrm{C}_1$ *and* C_0 *has an infinite and co-infinite member then there is a* C_1*-immune set A which is not* C_0*-bi-immune.*

PROOF. It suffices to prove the second part of the corollary. The first part follows by setting $C_0 = C_1 = C$. Fix $D \in C_0$ such that D and $\overline{D}$ are infinite. By Theorem 2.53 there is a C_1-immune set A contained in $\overline{D}$. $\overline{A}$ is not C_0-immune since the infinite set $D \in C_0$ is contained in $\overline{A}$. So A is not C_0-co-immune, hence not C_0-bi-immune. □

As pointed out above, in computational complexity theory many separations of complexity classes $C_0 \subset C_1$ can be extended to strong separations by showing that there is a C_0-(bi-)immune set in the class C_1 (see e.g. Geske et al. (1987) and Allender et al. (1993)). In the following we will look at the question what strong separations we can get for the Chomsky classes. We will use the following notation for strong separations.

$$C_0 <_0 C_1 :\Leftrightarrow C_0 \subset C_1$$

$$C_0 <_1 C_1 :\Leftrightarrow C_0 \subseteq C_1 \ \& \ \exists A \in C_1 (A\ C_0\text{-immune})$$

$$C_0 <_2 C_1 :\Leftrightarrow C_0 \subseteq C_1 \ \& \ \exists A \in C_1 (A\ C_0\text{-bi-immune})$$

Proposition 2.55 *Let* C_0 *and* C_1 *be any classes such that* C_0 *has an infinite member. Then*

$$C_0 <_2 C_1 \Rightarrow C_0 <_1 C_1 \Rightarrow C_0 <_0 C_1.$$

PROOF. Immediate by definition. □

By the preceding proposition, the following proposition establishes transitivity of the relations $<_0$, $<_1$ and $<_2$ in a strong sense.

Proposition 2.56 *Let* C_0, C_1, *and* C_2 *be classes which have infinite languages among their members and let* $i, j \in \{0,1,2\}$. *Then the following holds.*

$$C_0 <_i C_1 \ \& \ C_1 <_j C_2 \Rightarrow C_0 <_{max(i,j)} C_2.$$

PROOF. By Propositions 2.50 and 2.55. □

2.4.3 Immunity to the Class of Regular Languages

Now we will look at immunity and bi-immunity to the individual classes in the Chomsky hierarchy. We begin from the bottom of the hierarchy and start with the class REG of the regular sets.

As pointed out above, Flajolet and Steyaert (1974) have shown that the language $A = \{0^n 1^n : n \geq 1\}$ is immune to the class of regular languages. This easily follows from the pumping lemma for regular languages (see Theorem 2.16). For a contradiction assume that A is not REG-immune. Then there is an infinite regular subset B of A. By the pumping lemma there is a parameter $p \geq 0$ such that any word $w \in B$ with $|w| \geq p$ can be decomposed into $w = xyz$ such that $|xy| \leq p$, $y \neq \varepsilon$, and $w_n = xy^n z \in B$ for all $n \geq 0$. Now, by infinity of B and by $B \subseteq A$, there is a number $q \geq p$ such that $w = 0^q 1^q \in B$. Then, for the corresponding partition $w = xyz$, xy is a substring of 0^q whence, by $|y| > 0$, $w_0 = xz = 0^{q-|x|} 1^q \notin A$. So, by $B \subseteq A$, $w_0 \notin B$, a contradiction.

Since the language $A = \{0^n 1^n : n \geq 1\}$ is linear we obtain the following.

Theorem 2.57 *(Flajolet and Steyaert (1974)) There is a linear language A which is* REG-*immune.*

Corollary 2.58 REG $<_1$ LIN.

As we shall show next, however, the above theorem on REG-immune languages cannot be extended to REG-bi-immune languages.

Theorem 2.59 *No context-free language is* REG-*bi-immune.*

PROOF. Let A be context-free. We have to show that A or $\overline{A}$ is not REG-immune. Since, by Lemma 2.7, the intersection of any context-free language with a regular language is context-free again, $A \cap \{0\}^*$ is context-free. In fact, $A \cap \{0\}^*$ is regular since any unary context-free language is regular (see (2.2)). So, if $A \cap \{0\}^*$ is infinite, A is not REG-immune. Otherwise, the subset $\overline{A} \cap \{0\}^*$ of $\overline{A}$ is a finite variant of the infinite regular set $\{0\}^*$. Since the class of regular languages is closed under finite variants this implies that $\overline{A}$ is not REG-immune in this case. □

Corollary 2.60 REG $\not<_2$ CF.

Though REG-bi-immune sets are not context-free, hence not linear, in general REG-bi-immunity does not imply LIN-bi-immunity, in fact not even LIN(-co)-immunity.

Theorem 2.61 *There is a* REG-*bi-immune set A which is neither* LIN-*immune nor* LIN-*co-immune.*

PROOF. By Theorem 2.52, let A' be any CS-bi-immune set. Define A by letting

$$A = (A' \cup \{0^n 1^n : n \geq 1\}) - \{1^n 0^n : n \geq 1\}$$

Then A and $\overline{A}$ contain the infinite linear languages $\{0^n 1^n : n \geq 1\}$ and $\{1^n 0^n : n \geq 1\}$, respectively, whence neither A nor $\overline{A}$ is LIN-immune. So it suffices to show that A is REG-bi-immune.

For a contradiction assume that this is not the case. Then, by symmetry, w.l.o.g. we may assume that A contains an infinite regular set B as a subset. Split B into the two parts $B_0 = B - \{0^n 1^n : n \geq 1\}$ and $B_1 = B \cap \{0^n 1^n : n \geq 1\}$. Note that B_0 and B_1 are context-sensitive. (This follows from the facts that regular and linear languages are context-sensitive and that the class of context-sensitive languages is closed under the Boolean operations.) Moreover, B_0 is a subset of A' whence, by CS-bi-immunity of A', B_0 is finite. It follows that B_1 is a finite variant of the regular set B hence regular too. So B_1 is a regular subset of $\{0^n 1^n : n \geq 1\}$. Since, as shown above, $\{0^n 1^n : n \geq 1\}$ is REG-immune, it follows that B_1 is finite too. So $B = B_0 \cup B_1$ is finite contrary to assumption. This completes the proof. □

2.4.4 Immunity to the Classes of Linear and Context-Free Languages

Next we look at immunity and bi-immunity to the classes of the linear and context-free languages. We first observe that the immunity (hence bi-immunity) notions for these two language classes coincide.

Theorem 2.62 *Any infinite context-free language contains an infinite linear language as a sublanguage. Hence a set A is* CF-*(bi/co-)immune if and only if A is* LIN-*(bi/co-)immune.*

PROOF. Let A be context-free and infinite. By the pumping lemma for context-free languages (see Theorem 2.22), there is a word $z \in A$ and a partition $z = uvwxy$ of z such that vx is nonempty and, for any $n \geq 0$, the string $uv^n wx^n y$ is a member of A. So, for $B = \{uv^n wx^n y : n \geq 0\}$, B is an infinite subset of A and, as one can easily check, B is linear. □

Corollary 2.63 LIN $\not<_1$ CF.

Flajolet and Steyaert (1974) have shown that the context-sensitive language $A = \{0^n 1^n 0^n : n \geq 1\}$ is CF-immune. (This easily follows from the pumping lemma for context-free languages.) So in contrast to Corollary 2.63 we obtain a strong separation by immune sets on the next level of the Chomsky hierarchy, i.e., CF $<_1$ CS. In fact, as we will show next, this can be extended to a strong separation by bi-immune languages.

Theorem 2.64 *There is a context-sensitive language which is* CF-*bi-immune. Hence* CF $<_2$ CS.

PROOF. Let $A = \{x : \exists n(2^{2^{2n}} \leq |x| < 2^{2^{2n+1}})\}$. As one can easily check, A is in $DSPACE(O(n))$, hence context-sensitive. We will show that A is CF-bi-immune. By symmetry of the definition of A, it suffices to show that A is CF-immune. So, for a contradiction, assume that B is an infinite subset of A. Then, by the pumping lemma for context-free languages, there are words u, v, w, x, y such that $|vx| > 0$ and $z_n = uv^n wx^n y \in B$ for all $n \geq 0$. So, since B is contained in A, there are numbers $m \geq 0$ and $k \geq 1$ such that, for all $n \geq 0$ there is a word $z_n \in A$ such that $|z_n| = m + kn$. So A can have at most linear gaps, namely, for any number $n \geq m$, $A \cap \Sigma^{[n,n+k)} \neq \emptyset$. On the other hand, by definition of A, A has quadratic gaps, i.e., there are infinitely many numbers n such that $A \cap \Sigma^{[n,n^2)} = \emptyset$. Namely, for any $n \geq 0$,

$$\Sigma^{[2^{2^{2n+1}},(2^{2^{2n+1}})^2)} = \Sigma^{[2^{2^{2n+1}},2^{2^{2n+2}})} \subseteq \overline{A}.$$

This gives the desired contradiction. □

Theorem 2.64 implies that, in general, CF-bi-immunity does not imply CS-immunity.

Corollary 2.65 *There is a language A which is* CF-*bi-immune but neither* CS-*immune nor* CS-*co-immune.*

PROOF. Since the class of context-sensitive languages is closed under complement, this is immediate by Theorem 2.64. □

2.4.5 Immunity to the Higher Classes of the Chomsky Hierarchy

We now turn to the upper part of the Chomsky hierarchy. The basic results on immunity and bi-immunity to the classes of the context-sensitive languages, the recursive languages and the Comsky-0-languages, i.e., the recursively enumerable (r.e.) languages can be already found in the literature or can be easily derived from some general (bi-)immunity results there. So we only shortly review these results here.

We first observe that there is a strong separation of CS and REC by CS-bi-immune sets.

Theorem 2.66 *There is a recursive language which is* CS-*bi-immune.*

Theorem 2.66 is a special case of a quite general existence result for bi-immune sets: For any uniformly recursive class C there are recursive C-bi-immune languages (see Section 3.4 for details).

The following two corollaries are immediate by Theorem 2.66 (and the closure of REC under complement).

Corollary 2.67 CS $<_2$ REC.

Corollary 2.68 *There is a* CS-*bi-immune language which is neither* REC-*immune nor* REC-*co-immune.*

By the following well known observation in computability theory, there is no strong separation of REC and RE.

Theorem 2.69 *Every infinite recursively enumerable set contains an infinite recursive set as a subset. Hence* REC-*(bi/co-)immunity and* RE-*(bi/co-)immunity coincide.*

Corollary 2.70 REC $\not<_1$ RE.

2.4.6 Summary

By the above results on the immunity and bi-immunity notions for the individual classes of the Chomsky hierarchy, we can summarize the relations among these notions in the following table.

Theorem 2.71 *The following relations hold among the immunity and bi-immunity notions for the classes of the Chomsky hierarchy.*

$$\begin{array}{ccc}
\text{RE-}\textit{bi-immune} & \Rightarrow & \text{RE-}\textit{immune} \\
\Updownarrow & & \Updownarrow \\
\text{REC-}\textit{bi-immune} & \Rightarrow & \text{REC-}\textit{immune} \\
\Downarrow & & \Downarrow \\
\text{CS-}\textit{bi-immune} & \Rightarrow & \text{CS-}\textit{immune} \\
\Downarrow & & \Downarrow \\
\text{CF-}\textit{bi-immune} & \Rightarrow & \text{CF-}\textit{immune} \\
\Updownarrow & & \Updownarrow \\
\text{LIN-}\textit{bi-immune} & \Rightarrow & \text{LIN-}\textit{immune} \\
\Downarrow & & \Downarrow \\
\text{REG-}\textit{bi-immune} & \Rightarrow & \text{REG-}\textit{immune}
\end{array} \tag{2.10}$$

Moreover, these are the only implications (modulo transitive closure) which are valid in general.

PROOF. The implications from left to right are immediate by definition. Similarly, the downward implications are immediate by definition and the Chomsky hierarchy theorem. The two upper upwards implications hold by Theorem 2.69, the two lower upwards implications hold by Theorem 2.62. Corollary 2.54 implies that there is an RE-immune set which is not REG-bi-immune, whence none of the concepts in the right column implies any of the concepts in the left column, i.e., no implication from right to left holds. It remains to show that the concepts on levels 3, 4 and 6 do not imply any of the concepts on the next higher levels 2, 3 and 5, respectively. But this follows from Corollary 2.68, Corollary 2.65, and Corollary 2.61, respectively. □

The next theorem summarizes our answer to the question which strong separations can be achieved among the Chomsky classes.

Theorem 2.72 *The following strong separations hold among the Chomsky classes.*

$$\mathrm{REG} <_1 \mathrm{LIN} <_0 \mathrm{CF} <_2 \mathrm{CS} <_2 \mathrm{REC} <_0 \mathrm{RE}.$$

Moreover, these separations are optimal, since

$$\mathrm{REG} \not<_2 \mathrm{CF} \ \ \& \ \ \mathrm{LIN} \not<_1 \mathrm{CF} \ \ \& \ \ \mathrm{REC} \not<_1 \mathrm{RE}.$$

PROOF. The first part of the theorem follows from the Chomsky hierarchy theorem and Theorems 2.57, 2.64 and 2.67. The second part follows from Corollaries 2.60, 2.63 and 2.70. □

2.5 A Chomsky Hierarchy For Sequences

Formal language theory and computational complexity theory provide frame works for classifying languages according to their complexity. Since our genericity concepts will be based on characteristic sequences of languages not on the languages themselves it will be of interest to compare the complexity of a language with the complexity of its characteristic sequence. For this sake we will introduce an analogue of the Chomsky hierarchy for sequences in this section. There are two possible approaches for defining such a hierarchy. First, by identifying a sequence with its prefix set, we can transfer the Chomsky hierarchy for languages to a hierarchy for sequences. Alternatively, we can use the machine characterizations of the Chomsky language classes and consider the classes of sequences which can be predicted by the corresponding machines. Here we say that a machine M predicts (or computes) a sequence α if, given the first n bits of the sequence, M computes the $(n+1)$th bit.

Here we will use the first approach based on prefix sets but we will also show that the machine based approach is closely related to this though these two approaches do not always coincide. In particular, as we will show, the prediction model yields a larger class of context-free sequences.

2.5.1 Definitions and Basic Facts

Definition 2.73 A sequence α is *regular (linear, context-free, context-sensitive, recursive, recursively enumerable)* if the prefix set $Prefix(\alpha)$ of the sequence is regular (linear, context-free, context-sensitive, recursive, recursively enumerable, respectively).

The classes of regular, linear, context-free, context-sensitive, recursive, and recursively enumerable sequences are denoted by REG_S, LIN_S, CF_S, CS_S, REC_S, and RE_S, respectively. We call these classes the *Chomsky Hierarchy of Sequences*. Note that, by the coincidence of the Chomsky languages with the recursively enumerable languages, RE_S may be viewed as the class of all *Chomsky sequences*. The relations among the Chomsky language classes immediately imply the following.

Proposition 2.74 $\mathrm{REG}_S \subseteq \mathrm{LIN}_S \subseteq \mathrm{CF}_S \subseteq \mathrm{CS}_S \subseteq \mathrm{REC}_S \subseteq \mathrm{RE}_S$.

In the following we will analyze which of the above inclusions are proper. Moreover, we will compare the corresponding levels of the hierarchies of languages and sequences, i.e., the (language) classes C and

$$C'_S = \{A : \chi(A) \in C_S\}. \tag{2.11}$$

Note that, for any class C in the Chomsky hierarchy, C'_S is closed under complement.

Lemma 2.75 *For* $C \in \{REG, LIN, CF, CS, REC, RE\}$, C'_S *is closed under complement.*

PROOF. Note that for any language A, the characteristic sequence $\chi(\overline{A})$ of the complement of A is the dual sequence of the characteristic sequence $\chi(A)$ of A, i.e., $\chi(\overline{A})$ is obtained from $\chi(A)$ by flipping all bits. This easily implies that $Prefix(\chi(A))$ is regular (context-free, ...) if and only if $Prefix(\chi(\overline{A}))$ is regular (context-free, ...). □

This simple lemma shows, that for Chomsky classes C which are not closed under complement, i.e., for $C = LIN, CF, RE$, the classes C and C'_S do not coincide.

2.5.2 Regular and Context-Free Sequences

We now look at the Chomsky classes of sequences more closely starting from the bottom of the hierarchy. Our first observation is that, in contrast to the Chomsky hierarchy of languages, the lower levels of the Chomsky hierarchy of sequences collapse. This is an immediate consequence of the following observation of Calude and Yu on prefix sets.

Theorem 2.76 *(Calude and Yu (1997)) For an infinite sequence* $\alpha \in \Sigma^\omega$ *the following are equivalent.*

(i) α *is almost periodic, i.e., there are words* $v, w \in \Sigma^*$ *such that* $\alpha = vw^\omega$.

(ii) The prefix set $Prefix(\alpha)$ *of* α *is regular.*

(iii) The prefix set $Prefix(\alpha)$ *of* α *is context-free.*

(iv) The prefix set $Prefix(\alpha)$ *of* α *contains an infinite context-free subset, i.e.,* $Prefix(\alpha)$ *is not* CF-*immune.*

PROOF. The implications $(ii) \Rightarrow (iii) \Rightarrow (iv)$ are immediate by definition. The proof of the implication $(i) \Rightarrow (ii)$ is straightforward: Given an almost periodic sequence α fix $v, w \in \Sigma^*$ such that $\alpha = vw^\omega$. Then $Prefix(\alpha)$ is the finite union of the finite set $\{x : x \sqsubseteq v\}$ and the regular sets $A_y = \{vw^n y : n \geq 0\}$ (described by the regular expressions vw^*y) where $y \sqsubseteq w$ whence $Prefix(\alpha)$ is regular.

It remains to prove the implication $(iv) \Rightarrow (i)$. Assume that A is an infinite context-free subset of $Prefix(\alpha)$. Then, by the pumping lemma for context-free languages, there are words u, v, w, x, y such that $vx \neq \varepsilon$ and $z_n = uv^n wx^n y \in A$ for all $n \geq 0$. It follows that

$$uv^0wx^0y \sqsubset uv^1wx^1y \sqsubset \ldots \ uv^nwx^ny \sqsubset uv^{n+1}wx^{n+1}y \ \ldots \sqsubset \alpha.$$

So, if $v \neq \varepsilon$ then $\alpha = uv^\omega$, and if $v = \varepsilon$ then $\alpha = uwx^\omega$. □

Corollary 2.77 $\mathrm{REG}_S = \mathrm{LIN}_S = \mathrm{CF}_S$.

PROOF. Immediate by Proposition 2.74 and Theorem 2.76. □

Regularity of a set and regularity of its characteristic sequence are related as follows.

Theorem 2.78 *If the characteristic sequence $\chi(A)$ of a set A is regular then the set A is regular too.*

For the proof of this theorem we will need two observations: First, for any number m there is a finite automaton which, given a string x, computes the position of x in the length-lexicographical ordering modulo m. Second, any almost periodic sequence is a finite variant of a periodic sequence. We prove these two observations first.

Lemma 2.79 *For any number $m \geq 1$ there is a deterministic finite automaton $M_m = (\Sigma, S, \delta, s_0)$ (without distinguished final states) such that $S = \{s_0, \ldots, s_{m-1}\}$ and, for $n \geq 0$,*

$$\delta^*(s_0, z_n) = s_{n \bmod m}. \tag{2.12}$$

PROOF. By definition of the length-lexicographical ordering, for $n \geq 0$ and $i \leq 1$,

$$z_n i = z_{2n+2+i}. \tag{2.13}$$

So, since

$$(2n+2+i) \bmod m = (2(n \bmod m)+2+i) \bmod m, \tag{2.14}$$

we obtain the desired automaton $M_m = (\Sigma, \{s_0, \ldots, s_{m-1}\}, \delta, s_0)$ by letting

$$\delta(s_k, i) = s_{(2k+2+i) \bmod m} \quad (0 \leq k < m, 0 \leq i \leq 1). \tag{2.15}$$

The correctness of (2.12) is shown by induction on $|z_n|$. First assume that $|z_n| = 0$. Then $n = 0$ and, by $z_0 = \varepsilon$,

$$\delta^*(s_0, z_0) = \delta^*(s_0, \varepsilon) = s_0 = s_{0 \bmod m}.$$

For the inductive step, assume that $|z_n| > 0$ and that the claim is correct for all n' with $|z_{n'}| < |z_n|$. By $|z_n| > 0$ we may choose $n' \geq 0$ and $i \leq 1$ such that $z_n = z_{n'}i$. Then

$$\begin{aligned}
\delta^*(s_0, z_n) &= \delta^*(s_0, z_{n'}i) && \\
&= \delta(\delta^*(s_0, z_{n'}), i) && \\
&= \delta(s_{n' \bmod m}, i) && \text{(by inductive hypothesis)} \\
&= s_{[2(n' \bmod m)+2+i] \bmod m} && \text{(by (2.15))} \\
&= s_{(2n'+2+i) \bmod m} && \text{(by (2.14))} \\
&= s_{n \bmod m} && \text{(by } z_n = z_{n'}i \text{ and by (2.13))}
\end{aligned}$$

This completes the proof.

□

Lemma 2.80 *Let* α *be almost periodic. There is a periodic sequence* β *such that* $\alpha(n) = \beta(n)$ *for almost all numbers* $n \geq 0$.

PROOF. Fix $v, w \in \Sigma^*$ such that $\alpha = vw^\omega$ and let $k = |v|$ and $m = |w|$. W.l.o.g. we may assume that $k \leq m$. (Note that, for $m < k$, in the above presentation of α we may replace w by w^k to achieve this.) Define β by $\beta(n) = \alpha(m+n)$ for $n < k$ and $\beta(n) = \alpha(n)$ for $n \geq k$. Then β is a finite variant of α and $\beta = \hat{w}^\omega$ for $\hat{w} = w(m-k)...w(m-1)w(0)...w(m-k-1)$. □

PROOF OF THEOREM 2.78. Fix A and assume that $\alpha = \chi(A)$ is regular. We have to show that A is regular.

By Theorem 2.76, α is almost periodic. In fact, since the class of regular languages is closed under finite variants, by Lemma 2.80, w.l.o.g. we may assume that α is periodic. So fix $w \in \Sigma^*$ such that $\alpha = w^\omega$ and let m be the length of w. Then, for any $n \geq 0$, $z_n \in A$ if and only if $z_{n+m} \in A$, whence membership of a word z_n in A does not depend on n itself but only on n mod m:

$$\forall n \geq 0\,(A(z_n) = A(z_{n \bmod m})). \tag{2.16}$$

So if we extend the finite automaton M_m of Lemma 2.79 by letting

$$F = \{s_k : 0 \leq k < m \;\&\; z_k \in A\}$$

be the set of final states, then this automaton accepts A: For any $n \geq 0$,

$$\begin{aligned}
z_n \in A &\Leftrightarrow z_{n \bmod m} \in A && \text{(by (2.16))} \\
&\Leftrightarrow s_{n \bmod m} \in F && \text{(by definition of } F\text{)} \\
&\Leftrightarrow \delta^*(s_0, z_n) \in F && \text{(by (2.12))} \\
&\Leftrightarrow M \text{ accepts } z_n.
\end{aligned}$$

So A is regular. □

The converse of Theorem 2.78, however, fails.

Theorem 2.81 *There is a regular set A such that the characteristic sequence* $\chi(A)$ *of A is not regular.*

PROOF. An example of a regular set A with nonregular characteristic sequence is the set of all unary strings, i.e., $A = \{0^n : n \geq 0\}$. Obviously, A is regular but $\chi(A)$ is not almost periodic (note that the nth and $(n+1)$th 1 in $\chi(A)$ is separated by $2^{n-1} - 1$ many 0s), hence not regular by Theorem 2.76. □

Corollary 2.82 *(a)* $\{A : \chi(A)$ *regular*$\} \subset \{A : A$ *regular*$\}$
(b) $\{A : \chi(A)$ *context-free*$\} \subset \{A : A$ *context-free*$\}$

PROOF. The first part is immediate by Theorems 2.78 and 2.81. The second part follows from the first part since $\{A : \chi(A) \text{ regular}\} = \{A : \chi(A) \text{ context-free}\}$ by Corollary 2.77 and since any regular set is context-free. □

2.5.3 Context-Sensitive and Recursive Sequences

We now turn to the upper part of the Chomsky hierarchy of sequences. At the top we have a further collapse, namely a sequence is recursive if and only if it is recursively enumerable. This follows from the observation that any recursively enumerable prefix set is recursive. On the intermediate levels, however, the hierarchy of sequences is strict, i.e., $\mathrm{CF}_S \subset \mathrm{CS}_S \subset \mathrm{REC}_S$. To show this we will use the complexity theoretic characterization of the context-sensitive languages.

Theorem 2.83 *For any set A, the following are equivalent.*

(i) $A \in \mathrm{NSPACE}(2^n)$.

(ii) $Prefix(A) \in \mathrm{NSPACE}(n)$.

The key to the proof is the observation that the length of the nth binary word is logarithmic in n, i.e., that the number of the predecessors of a word w grows exponentially in the length $|w|$ of w. Further ingredients of the proof are the Linear-Compression Theorem for nondeterminstic space complexity which asserts that, for any space-constructible bound $s(n)$, $\mathrm{NSPACE}(s(n)) = \mathrm{NSPACE}(O(s(n)))$ and the Theorem of Immerman and Szelepcsenyi (Theorem 2.28) which asserts that, for constructible space bounds, the nondeterministic space classes are closed under complement.

PROOF. $(i) \Rightarrow (ii)$ Let A be given such that $A \in \text{NSPACE}(2^n)$. By linear compression it suffices to show that $Prefix(A) \in \text{NSPACE}(O(n))$. By closure under complement of the nondeterministic space classes with constructible bounds, $\overline{A} \in \text{NSPACE}(2^n)$ too whence we may fix 2^n-space bounded nondeterministic Turing machines N_1 and N_2 which accept A and $\overline{A}$, respectively.

Based on N_1 and N_2 we can define a nondeterministic machine N accepting $Prefix(A)$ as follows. On input $x = x(0)...x(|x|-1)$, N inductively (and nondeterministicly) computes $A(z_m)$ for $m = 0, ..., |x|-1$ and compares $A(z_m)$ with $x(m)$. If a string z_m with $A(z_m) \neq x(m)$ is found, N stops and rejects. If $A(z_m) = x(m)$ for all $m < |x|$ then N accepts x. For computing $A(z_m)$ for given m, N first (nondeterministicly) simulates N_1 on input z_m. If (the simulated computation path of) N_1 accepts then (the corresponding computation path of) N sets $A(z_m) = 1$. If (the simulated computation path of) N_1 rejects, then (the corresponding computation path of) N starts to simulate (a nondeterministicly chosen computation path) of N_2 on input z_m. If (the chosen computation path of) N_2 accepts then (the corresponding computation path of) N sets $A(z_m) = 0$. If (the simulated computation path of) N_2 also rejects, then (the corresponding computation path of) N stops and rejects (in particular, it does not output a value for $A(z_m)$).

Note that any computation path of N which assigns a value to $A(z_m)$ assigns the correct value. Moreover, there is at least one computation which assigns a value to $A(z_m)$. So the machine N accepts $Prefix(A)$.

It remains to show that N works within the space bound $O(n)$. Let $n = |x|$. Since the space, used in an individual cycle of the inductive procedure on which N is based, can be reused in the next cycle, it suffices to show that, for $m < n$, the length of z_m is bounded by $O(n)$ and that the computation of $A(z_m)$ can be done within the same space bound. Now, for $m \leq n$, $|z_m| \leq log(n) + O(1)$ whence the former is immediate. The latter follows from the space bounds on the machines N_1 and N_2: The space required for computing $A(z_m)$ is bounded by

$$\begin{aligned} max(space_{N_1}(z_m), space_{N_2}(z_m)) &\leq max(2^{|z_m|}, 2^{|z_m|}) \\ &= 2^{|z_m|} \\ &\leq 2^{log(n)+O(1)} \\ &= O(2^{log(n)}) \\ &= O(n) \end{aligned}$$

For a proof of the implication $(ii) \Rightarrow (i)$ assume that A is given such that $Prefix(A) \in \text{NSPACE}(n)$ and fix a nondeterministic n-space bounded machine N which accepts $Prefix(A)$.

Then a nondeterministic $O(2^n)$-space bounded machine N' which accepts A works as follows. On input w of length n, first N deterministicly computes the unique number $m+1$ (in unary representation) such that $w = z_m$. (Note that $m \leq$

2^{n+1} and that the unary representation 0^{m+1} of $m+1$ can be (deterministically) computed from w without using any additional space besides the space needed to hold 0^{m+1}.) Given 0^{m+1}, N' inductively (and nondeterministicly) simulates N on all words x of length $m+1$ (in lexicographical order starting with 0^{m+1}) until N accepts the first such x. Now if the accepted x ends with a 1 then N' accepts the input w, otherwise N' rejects. Note that the x accepted by N is a prefix of $\chi(A)$ of length $m+1$ whence, for the last bit $x(m)$ of x, $x(m) = A(z_m) = A(w)$. So N' accepts the language A.

It remains to show that N' is $O(2^n)$-space bounded. Now, on input w as above, the space needed by N' is bounded by the space needed for producing 0^{m+1} and by the space needed for simulating N on a word x of length $m+1$. Since N is n-space bounded and since the space required for producing 0^{m+1} is $m+1$, we have

$$space_{N'}(w) \leq m+1 \leq 2^{n+1}+1 \leq O(2^n).$$

This completes the proof. □

By CS = NSPACE(n), Theorem 2.83 implies that the characteristic sequence of any context-sensitive language is context-sensitive too but that the converse in general fails.

Corollary 2.84 $\{A : A \textit{ context-sensitive}\} \subset \{A : \chi(A) \textit{ context-sensitive}\}$.

PROOF. By CS = NSPACE(n) it suffices to show

$$\{A : A \in \mathrm{NSPACE}(n)\} \subset \{A : Prefix(A) \in \mathrm{NSPACE}(n)\}. \qquad (2.17)$$

Since, by Theorem 2.83,

$$\{A : Prefix(A) \in \mathrm{NSPACE}(n)\} = \{A : A \in \mathrm{NSPACE}(2^n)\},$$

this follows from the nondeterministic space hierarchy theorem (see Corollary 2.29). □

As another consequence of Theorem 2.83 we get the following strict inclusions among the classes of the contex-free, the context-sensitive, and the recursive sequences.

Corollary 2.85 $\mathrm{CF}_S \subset \mathrm{CS}_S \subset \mathrm{REC}_S$.

PROOF. By Proposition 2.74 it suffices to show that CS_S is not contained in CF_S and that REC_S is not contained in CS_S. For a proof of the former let A be any context-sensitive set which is not context-free. Then, by Corollary 2.84,

$\chi(A) \in \mathrm{CS}_S$ but, by Corollary 2.82, $\chi(A) \notin \mathrm{CF}_S$. For a proof of the latter let B be any recursive set such that $B \notin \mathrm{NSPACE}(2^n)$. (Note that such a set exists by the nondetermistic space hierarchy theorem.) Then, obviously, $Prefix(A)$ is recursive whence $\chi(A) \in \mathrm{REC}_S$ but, by Theorem 2.83, $Prefix(A) \notin \mathrm{NSPACE}(n) = \mathrm{CS}$ whence $\chi(A) \notin \mathrm{CS}_S$. □

For the proof of the second part of the preceding corollary we have used the observation that a set A is recursive if and only if its prefix set is recursive. This simple fact together with the observation that, for a prefix set, recursiveness and recursive enumerability coincide will give the still missing facts on the Chomsky hierarchy of sequences and its relation to the classical Chomsky hierarchy of languages.

Theorem 2.86 *(a) For any set A, A is recursive if and only if* $Prefix(A)$ *is recursive.*

(b) For any set A, $Prefix(A)$ *is recursive if and only if* $Prefix(A)$ *is recursively enumerable.*

PROOF. Part (a) is straightforward. For a proof of the nontrivial implication in (b) assume that $Prefix(A)$ is recursively enumerable, say $Prefix(A)$ is the range of the recursive function $f : \mathbb{N} \to \Sigma^*$. Then, given x, membership of x in $Prefix(A)$ can be decided as follows: Fix n minimal such that $|f(n)| \geq |x|$. Then $x \in Prefix(A)$ if and only if $x \sqsubseteq f(n)$. □

Corollary 2.87 $\mathrm{REC}_S = \mathrm{RE}_S$.

PROOF. This is immediate by Theorem 2.86 (b). □

Corollary 2.88 *(a)* $\{A : \chi(A) \textit{ recursive}\} = \{A : A \textit{ recursive}\}$.

(b) $\{A : \chi(A) \textit{ recursively enumerable}\} \subset \{A : A \textit{ recursively enumarable}\}$.

PROOF. The first part is immediate by Theorem 2.86 (a). Moreover, the first part together with Corollary 2.87 implies that

$$\{A : \chi(A) \text{ recursively enumerable}\} = \{A : A \text{ recursive}\}.$$

Since there are recursively enumerable sets which are not recursive this implies the second part of the corollary. □

2.5.4 The Chomsky Hierarchy Theorem For Sequences

The above results on the Chomsky classes of sequences lead to the following hierarchy theorem.

Theorem 2.89 $\mathrm{REG}_S = \mathrm{LIN}_S = \mathrm{CF}_S \subset \mathrm{CS}_S \subset \mathrm{REC}_S = \mathrm{RE}_S$.

PROOF. By Proposition 2.74 and Corollaries 2.77, 2.85 and 2.87. □

The relation between the location of a language A in the Chomsky hierarachy (of languages) and the location of its characteristic sequence $\chi(A)$ in the Chomsky hierarchy of sequences is given in the next theorem.

Theorem 2.90 *For any set A the following holds.*

$$\begin{array}{ccc}
A \textit{ regular} & \Leftarrow & \chi(A) \textit{ regular} \\
\Downarrow & & \Updownarrow \\
A \textit{ linear} & (\Leftarrow) & \chi(A) \textit{ linear} \\
\Downarrow & & \Updownarrow \\
A \textit{ context-free} & (\Leftarrow) & \chi(A) \textit{ context-free} \\
\Downarrow & & \Downarrow \\
A \textit{ context-sensitive} & \Rightarrow & \chi(A) \textit{ context-sensitive} \\
\Downarrow & & \Downarrow \\
A \textit{ recursive} & \Leftrightarrow & \chi(A) \textit{ recursive} \\
\Downarrow & & \Updownarrow \\
A \textit{ recursively enumerable} & (\Leftarrow) & \chi(A) \textit{ recursively enumerable}
\end{array} \tag{2.18}$$

Moreover, in general these are the only valid implications (modulo transitive closure).

Note that the implications marked by arrows in parantheses follow by transitive closure. We have only added these arrows to give the level-by-level relations between the hierarchy of languages and the hierarchy of sequences. By using the notation of (2.11), i.e., the classes

$$\mathrm{C}'_S = \{A : \chi(A) \in \mathrm{C}_S\} = \{S(\alpha) : \alpha \in \mathrm{C}_S\}$$

for any Chomsky class C, Theorem 2.90 is captured by the following relations among the Chomsky classes C and the corresponding classes C'_S:

$$\begin{aligned}
\mathrm{REG}'_S = \mathrm{LIN}'_S = \mathrm{CF}'_S \subset \mathrm{REG} \subset \mathrm{LIN} \subset \mathrm{CF} \subset \mathrm{CS} \\
\subset \mathrm{CS}'_S \subset \mathrm{REC} = \mathrm{REC}'_S = \mathrm{RE}'_S \subset \mathrm{RE}
\end{aligned} \tag{2.19}$$

PROOF (OF THEOREM 2.90). It suffices to show that (2.19) holds. The proper inclusions $\mathrm{REG} \subset \mathrm{LIN} \subset \mathrm{CF} \subset \mathrm{CS}$ and $\mathrm{REC} \subset \mathrm{RE}$ hold by the Chomsky hierarchy theorem for languages while the equalities $\mathrm{REG}'_S = \mathrm{LIN}'_S = \mathrm{CF}'_S$ and $\mathrm{REC}'_S = \mathrm{RE}'_S$ hold by the Chomsky hierarchy theorem for sequences (Theorem 2.89). The remaining proper inclusions $\mathrm{REG}'_S \subset \mathrm{REG}$ and $\mathrm{CS} \subset \mathrm{CS}'_S \subset \mathrm{REC}'_S$ hold by Corollary 2.82 and Corollary 2.85, respectively. □

2.5.5 Prediction Machines

In the remainder of this section we look at the Chomsky complexity of sequences in terms of prediction machines. This will yield alternative characterizations of the classes of regular, context-sensitive, and recursive sequences but it will also lead to a more general notion of a context-free sequence.

Intuitively, a machine M predicts a sequence α if, given the first n bits of the sequence, i.e., $\alpha \restriction n$, the machine outputs the $(n+1)$th bit $\alpha(n)$ of the sequence. An acceptor M can be used for modelling prediction in two somewhat different ways. First we can say that M on input $\alpha \restriction n$ predicts the next bit $\alpha(n)$ to be 1 if M accepts input $\alpha \restriction n$ and M predicts $\alpha(n)$ to be 0 otherwise. In this case we say that M *weakly predicts* α. Since for nondeterministic (or non-total) machines M, acceptance and rejection are not symmetric, this approach might lead to asymmetries in predicting a 0 or a 1, i.e., the fact that a sequence α can be predicted by a machine of a certain type in general will not imply that there is another machine of the same type predicting the dual sequence $\hat{\alpha}$ of α (which is obtained from α by interchanging zeroes and ones). We obtain a symmetric prediction model by considering extended machines M (see Definition 2.47). Such a machine M *strongly predicts* α, if on input $\alpha \restriction n$ the machine M predicts the next bit $\alpha(n)$ to be 1 if there is a computation ending in a $+$-state and 0 if there is a computation ending in a $-$-state. Here prediction in particular requires that M is *consistent along* α (i.e., on any input $\alpha \restriction n$ there can't be two computations one ending in a $+$-state and the other ending in a $-$-state) and *complete w.r.t.* α (i.e., on any input $\alpha \restriction n$ there is some computation ending in a $+$-state or $-$-state).

Note that any strong predictor M can be easily converted into a weak predictor M' of the same type by letting the accepting states of M' be the $+$-states of M. The converse is true for total determinstic machines: A total deterministic weak predictor M can be interpreted as a strong predictor M' by letting the $+$-states of M' be the accepting states of M, by letting the $-$-states of M' be the rejecting states of M, and by letting the set of the ?-states of M' be empty. So, for the standard classes of total deterministic machines, weak prediction and strong prediction will coincide. Consequently, in case of total deterministic machines we will denote weak prediction simply by *prediction*. For a non-total or non-deterministic weak predictor M, however, in general there is no trivial conversion into a strong predictor of the same type. So, for some of the standard classes of non-total or non-determinsitic machines, weak predictability might be more general than strong predictability.

We first observe that, for finite automata, the different types of predictability are equivalent and that a sequence α can be predicted by a finite automaton if and only if α is regular.

Theorem 2.91 *The following are equivalent.*

(i) α *is regular.*

(ii) α *is predictable by a deterministic finite automaton.*

(iii) α *is strongly predictable by a nondeterministic finite automaton.*

(iv) α *is weakly predictable by a nondeterministic finite automaton.*

PROOF. Note that, by definition, $(ii) \Rightarrow (iii) \Rightarrow (iv)$ holds. So it suffices to prove the implications $(i) \Rightarrow (ii)$, $(iv) \Rightarrow (ii)$, and $(ii) \Rightarrow (i)$.

For a proof of $(i) \Rightarrow (ii)$, assume that α is regular and let $M = (\Sigma, S, \delta, s_0, F)$ be a determinisitic finite automaton which accepts $Prefix(\alpha)$. Then a DFA M' which predicts α is obtained as follows. On any input x, $|x| = n$, M' simulates M on input $x1$ and M' accepts x if and only if M accepts $x1$. Formally, $M' = (\Sigma, S, \delta, s_0, F')$ where $F' = \{s \in S : \delta(s,1) \in F\}$.

For a proof of the implication $(iv) \Rightarrow (ii)$, assume that N is a nondeterministic finite automaton which weakly predicts α. Then $L(N)$ is regular whence there is a deterministic finite automaton M with $L(M) = L(N)$. It follows that M predicts α since, for any $n \geq 1$,

$$\begin{array}{rcll} N \text{ predicts } \alpha(n) = 1 & \Leftrightarrow & \alpha \restriction n \in L(N) & \text{(by definition)} \\ & \Leftrightarrow & \alpha \restriction n \in L(M) & \text{(by } L(N) = L(M)) \\ & \Leftrightarrow & M \text{ predicts } \alpha(n) = 1 & \text{(by definition).} \end{array}$$

Finally, for a proof of the implication $(ii) \Rightarrow (i)$, assume that $M = (\Sigma, S, \delta, s_0, F)$ is a deterministic finite automaton which predicts α. We will convert M into a DFA $\hat{M}$ which accepts $Prefix(\alpha)$. Note that the sequence α (and its initial segments $\alpha \restriction n$ of a given length n are uniquely determined by the predictor M: The prefix $\alpha \restriction n$ of α of length n is the unique string $x = x(0)...x(n-1)$ of length n satisfying

$$x(m) = 1 \Leftrightarrow M \text{ accepts } x \restriction m \tag{2.20}$$

for all $m < n$. The acceptor $\hat{M}$ of $Prefix(\alpha)$ is based on this observation. On input $x = x(0)...x(n-1)$ the automaton simulates M step by step as long as the input is consistent with (2.20) . If an inconsistency is found, $\hat{M}$ ends the simulation and goes into an absorbing rejecting state. Formally, $\hat{M} = (\Sigma, S \cup \{s_-\}, \hat{\delta}, s_0, S)$ where $s_- \notin S$ and $\hat{\delta}$ is defined as follows: $\hat{\delta}(s,i) = \delta(s,i)$ if $s \in F$ and $i = 1$ or $s \in S - F$ and $i = 0$; $\hat{\delta}(s,i) = s_-$ if $s \in F$ and $i = 0$ or $s \in S - F$ and $i = 1$; and $\hat{\delta}(s_-,i) = s_-$ for $i = 0,1$. □

The last part of the proof of Theorem 2.91 can be easily adapted to deterministic push down automata M and $\hat{M}$ in place of the corresponding deterministic finite automata. This yields the following lemma.

Lemma 2.92 *Assume that α can be predicted by a deterministic push down automaton. Then $Prefix(\alpha)$ is context-free.*

By coincidence of the almost periodic sequences, the regular sequences, and the context-free sequences, the above theorem and lemma imply

Corollary 2.93 *The following are equivalent.*
(i) α is almost periodic.
(ii) α is regular.
(iii) α is context-free.
(iv) α is predictable by a deterministic finite automaton.
(v) α is strongly predictable by a nondeterministic finite automaton.
(vi) α is weakly predictable by a nondeterministic finite automaton.
(vii) α is predictable by a deterministic push down automaton.

PROOF. By Theorems 2.76 and 2.91 and Lemma 2.92. □

As we will show next, however, prediction by nondeterministic push down automata is more powerful than prediction by finite automata or deterministic push down automata.

Theorem 2.94 *There is a sequence α which is not context-free and which can be strongly (hence weakly) predicted by a nondeterministic push down automaton.*

PROOF. Consider the sequence

$$\alpha = 1\,0\,1\,0^2\,1\,0^3\,\ldots$$

This sequence is not almost periodic hence, by Theorem 2.76, not context-free. It remains to show that there is an extended NPDA M which strongly predicts α. Such an automaton uses the following inductive characterization of $\alpha(n)$ in terms of $\alpha \restriction n$:

$$\alpha(0) = 1 \quad (2.21)$$

$$\alpha \restriction n = w10^m \Rightarrow [\alpha(n) = 1 \Leftrightarrow \#_1(w1) = m] \quad (2.22)$$

Now, intuitively, M works as follows. On input ε, M predicts the next bit to be a one. Given a nonempty input x, say $|x| = n \geq 1$, M reads the input and stores the 1s read in the stack. Moreover, whenever M has read a 1 it may guess that this was the last 1 in the input word x and may change its working mode as follows. For any 0 read in the sequel M pops a 1 from the stack. Moreover, as long as the stack still contains a 1, M predicts the next bit to be a 0 and if the stack is empty (i.e., only contains the start symbol) M predicts the next bit to be a 1 and stops. If M reads a 1 though the stack is not yet empty it aborts the computation and goes into an absorbing rejecting state thereby not making any more predictions (along this computation path). Formally, M is defined as follows. The states of M are s_0^1, s_1, s_2^0, s_3^1, where s_0^1 is the initial state, s_0^1 and s_3^1 are $+$-states (predicting the next bit to be a 1), s_2^0 is the only $-$-state (predicting the next bit to be a 0), and s_1 is a ?-state. The transition relation Δ is given in the following table.

S	$\times$	Σ	$\times$	Γ	$\times$	S	$\times$	Γ^*
s_0^1		1		b		s_1		$b1$
s_0^1		1		b		s_2^0		b
s_1		0		1		s_1		1
s_1		1		1		s_1		11
s_1		1		1		s_2^0		1
s_2^0		0		1		s_2^0		ε
s_2^0		0		b		s_3^1		b

(In a situation where no transition is specified and the input is not yet completely read, the automaton gets stuck, i.e., it ends in an implicitly given absorbing ?-state.) Note that M works as informally described above. In the initial state s_0^1 M nondeterministically decides whether, for an input $\alpha \upharpoonright n = 1x$, the suffix x is empty (s_2^0) or not (s_1). (If the first letter of the input is a 0 then M gets stuck.) In the latter case, i.e., in state s_1, M stores the 1s read in the stack until it may guess that the 1 just read is the last 1 in the input. If this happens M nondeterministically switches to the $-$-state s_2^0, in which M compares the number of 1s in the stack with the number of 0s in the not yet read part of the input. M accepts (s_3^1) if these numbers agree and M rejects (s_2^0) if the number of 0s is exceeded by the number of 1s. If the number of 0s exceeds the number of 1s or if the remainder of the input contains another 1 then this computation of M is aborted, i.e., ends in a ?-state. □

As one can esily show, any sequence α which can be (strongly or weakly) predicted by an NPDA is context-sensitive and there are context-sensitive sequences which cannot be predicted by an NPDA. So if we let NPDA_S^{sp} (NPDA_S^{wp}) denote the class of sequences which can be strongly (weakly) predicted by an NPDA then

$$\text{CF}_S \subset \text{NPDA}_S^{sp} \subseteq \text{NPDA}_S^{wp} \subset \text{CS}_S \tag{2.23}$$

It might be of interest to further investigate these intermediate prediction classes. In particular, it is natural to ask whether the class NPDA_S^{sp} is strictly contained in the class NPDA_S^{wp}. Moreover, what can we say about the complexity of the corresponding languages. Are there regular or context-free languages A such that $\chi(A) \in \text{NPDA}_S^{sp}$ or $\chi(A) \in \text{NPDA}_S^{wp}$? We leave these questions open and turn to context-sensitive prediction.

For context-sensitive sequences, prediction complexity and prefix complexity coincide. We show this by giving the following characterization of the context-sensitive sequences in terms of predictability by linear-space bounded nondeterministic Turing machines.

Theorem 2.95 *For any sequence* $\alpha \in \Sigma^\omega$ *the following are equivalent.*

(i) α *is context-sensitive.*

(ii) α *is strongly* $\text{NSPACE}(n)$*-predictable.*

(iii) α *is weakly* $\text{NSPACE}(n)$*-predictable.*

PROOF. For a proof of the implication $(i) \Rightarrow (ii)$ assume that α is context-sensitive. Then, by $\text{CS} = \text{NSPACE}(n)$, there is a nondeterministic n-space bounded Turing machine M which accepts $Prefix(\alpha)$ and, by closure of $\text{NSPACE}(n)$ under complement, there is a nondeterministic n-space bounded Turing machine M' which accepts $\overline{Prefix(\alpha)}$. Then an extended nondeterministic n-space bounded Turing machine M'' which strongly predicts α works as follows. On input $x = x_0 \dots x_{n-1}$, $|x| = n$, M'' first (nondeterministically) simulates M on input $x1$. If M accepts then M'' stops in a $+$-state. Otherwise, M'' next (nondeterministically) simulates M' on input $x1$. If M' accepts then M'' stops in a $-$-state. Otherwise, M'' stops in a ?-state.

The implication $(ii) \Rightarrow (iii)$ is immediate.

Finally, for a proof of the implication $(iii) \Rightarrow (i)$, assume that α is weakly $\text{NSPACE}(n)$-predicted by M and let $L(M)$ be the language accepted by M. By $\text{CS} = \text{NSPACE}(n) = \text{NSPACE}(O(n))$, it suffices to show that $Prefix(\alpha) \in \text{NSPACE}(O(n))$.

Since $L(M) \in \text{NSPACE}(n)$ and $\text{NSPACE}(n)$ is closed under complement, $\overline{L(M)} \in \text{NSPACE}(n)$ whence we may fix a nondeterministic n-space bounded Turing machine M' which accepts $\overline{L(M)}$. Moreover, for any number n, $\alpha \restriction n \in L(M)$ if and only if $\alpha(n) = 1$. We can use these observations for defining a nondeterministic $O(n)$-space bounded Turing machine M'' which, on input 0^n nondeterministicly computes $\alpha \restriction n$ (i.e., any accepting path yields $\alpha \restriction n$ as output and there is at least one accepting path). Obviously this suffices to prove the claim since M'' can be easily converted into a nondeterministic $O(n)$-space bounded Turing machine accepting $Prefix(\alpha)$ which works as follows: On input x, $|x| = n$, simulate M'' on input 0^n in order to compute $\alpha \restriction n$; accept if and only if $x = \alpha \restriction n$.

It remains to describe M''. M'' formalizes the following inductive procedure for computing $\alpha \restriction n$: First, $\alpha \restriction 0 = \varepsilon$. Second, given $\alpha \restriction m$, $\alpha \restriction (m+1) = (\alpha \restriction m)1$ if and only if $\alpha \restriction m \in L(M)$ and $\alpha \restriction (m+1) = (\alpha \restriction m)0$ if and only if $\alpha \restriction m \in L(M')$.

So, given $\alpha \upharpoonright m$, $\alpha \upharpoonright (m+1)$ can be (nondeterministicly) computed by simulating M and M' as follows. First simulate M on input $\alpha \upharpoonright m$. If (the nondeterministicly chosen computation of) M accepts then set $\alpha \upharpoonright (m+1) = (\alpha \upharpoonright m)1$. Otherwise, simulate M' on input $\alpha \upharpoonright m$. If (the nondeterministicly chosen computation of) M' accepts then set $\alpha \upharpoonright (m+1) = (\alpha \upharpoonright m)0$. Otherwise abort the computation by stopping in a rejecting state. □

We conclude our investigation of predictability by shortly commenting on predictability by Turing machines. It is easy to show that, for total machines, predictability coincides with recursiveness. (Here we call a nondeterministic machine *total* if, on any input, *all* possible computations are finite.) If we consider machines with divergent computations, then the strongly predictable sequences are recursive too. The weakly predictable sequences, however, in general are not recursive.

Theorem 2.96 *For any sequence* α*, the following are equivalent.*

(i) α *is recursive.*
(ii) α *is predictable by a total deterministic Turing machine.*
(iii) α *is strongly predictable by a total nondeterministic Turing machine.*
(iv) α *is weakly predictable by a total nondeterministic Turing machine.*
(v) α *is strongly predictable by a deterministic Turing machine.*
(vi) α *is strongly predictable by a nondeterministic Turing machine.*

PROOF (IDEA). Since the implications $(ii) \Rightarrow (iii) \Rightarrow (iv)$ and $(ii) \Rightarrow (v) \Rightarrow (vi)$ are immediate by definition, it suffices to show the implications $(i) \Rightarrow (ii)$, $(iv) \Rightarrow (i)$ and $(vi) \Rightarrow (i)$.

For a proof of $(i) \Rightarrow (ii)$ assume that α is recursive and fix a total, deterministic Turing machine M which accepts $Prefix(\alpha)$. Then the total, deterministic Turing machine $\hat{M}$ working as follows predicts α: On input x of length n, $\hat{M}$ simulates M on input $x1$. If M accepts then $\hat{M}$ predicts the next bit to be 1; if M rejects then $\hat{M}$ predicts the next bit to be 0.

For a proof of $(iv) \Rightarrow (i)$ assume that M is a total, nondeterministic Turing machine which weakly predicts α. Then $Prefix(\alpha)$ can be decided by the following inductive procedure. Let x be given, $|x| = n$. If $n = 0$, i.e. $x = \varepsilon$ then $x \in Prefix(\alpha)$. So assume that $n > 0$ and fix $y \in \Sigma^{n-1}$ and $i \in \Sigma$ such that $x = yi$. Then $x \in Prefix(\alpha)$ if and only if $y \in Prefix(\alpha)$ and either $i = 1$ and M accepts y (i.e., on input y, M predicts the next bit to be a 1) or $i = 0$ and M rejects y (i.e., on input y, M predicts the next bit to be a 0). Note that this allows us to decide $x \in Prefix(\alpha)$ since, by inductive hypothesis, we can decide $y \in Prefix(\alpha)$ and since, by totality of M, we can decide whether or not M accepts y.

The proof of $(vi) \Rightarrow (i)$ is similar. Fix a nondeterministic (not necessarily total) Turing machine M which strongly predicts α. Then, on input $\alpha \upharpoonright n$, there will be either a finite M-computation predicting $\alpha(n) = 1$ or a finite M-computation predicting $\alpha(n) = 0$. So, given $\alpha \upharpoonright n$, by a breadth-first search in the computation tree of M, we can compute $\alpha(n)$. Hence we can decide $Prefix(\alpha)$ as follows. Given a string x of length n, inductively compute $\alpha \upharpoonright m$ for $m \leq n$ and accept x if and only if $x = \alpha \upharpoonright n$. □

The following lemma shows that there are nonrecursive sequences which can be weakly predicted by a (nontotal) Turingmachine.

Lemma 2.97 *Let A be recursively enumerable and let α be the characteristic sequence of A. Then α is weakly predictable by a deterministic Turing machine.*

PROOF (IDEA). Fix a deterministic Turing machine M which accepts A. A weak deterministic Turing machine predictor $\hat{M}$ of α works as follows. Given a string x of length n, $\hat{M}$ simulates M on input z_n and accepts if and only if M accepts. So, in particular, $\hat{M}$ accepts the input $\alpha \upharpoonright n$ (i.e. predicts $\alpha(n) = 1$) if and only if $z_n \in A$. □

The converse of Lemma 2.97 in general fails. In fact, the class of sequences which are weakly predictable by Turing machines coincides with the class of sequences α which - if interpreted as the binary expansion $0.\alpha(0)\alpha(1)\alpha(2)...$ of a real - can be effectively approximated from below by rationals. Such sequences are called *left computable reals* or *computably enumerable reals* (see e.g. Ambos-Spies et al. (2000)).

Definition 2.98 A sequence α is a *left computable real* if there is a recursive sequence of words α_s $(s \geq 0)$ such that $|\alpha_s| = s$, $\alpha_s \leq \alpha_{s+1}$ and, for $n \geq 0$,

$$\alpha \upharpoonright n = \lim_{s \to \infty} \alpha_s \upharpoonright n.$$

Note that any recursive sequence is a left computable real. In fact, the characteristic sequence of any recursively enumerable set is left computable. As Jockusch has observed, there are left computable reals α, however, such that the corresponding set $S(\alpha)$ is not recursively enumerable (see e.g. Ambos-Spies et al. (2000)).

Theorem 2.99 *For any sequence α, the following are equivalent.*
(i) α is a left computable real.
(ii) α is weakly predictable by a deterministic Turing machine.
(iii) α is weakly predictable by a nondeterministic Turing machine.

PROOF (IDEA). Since the implication $(ii) \Rightarrow (iii)$ is obvious, it suffices to show $(i) \Rightarrow (ii)$ and $(iii) \Rightarrow (i)$.

For a proof of $(i) \Rightarrow (ii)$ assume that $(\alpha_s)_{s\geq 0}$ is a recursive approximation of α as in Definition 2.98. Then a Turing machine M which weakly predicts α works as follows. On input x, M enumerates the words α_s for $s \geq 0$. If for some s, $x1 \sqsubseteq \alpha_s$ then M accepts x (otherwise the computation of M will diverge thereby rejecting x).

For a proof of $(iii) \Rightarrow (i)$ assume that M is a nondeterministic Turing machine which weakly predicts α. Define a recursive sequence $(\alpha_s)_{s\geq 0}$ as follows. Given $s \geq 0$, the word $\alpha_s = \alpha_s(0)...\alpha_s(s-1)$ is inductively defined by $\alpha_s(k) = 1$ if and only if there is a computation of M of length at most s which accepts $\alpha_s \restriction k$. Then, as one can easily check, the sequence $(\alpha_s)_{s\geq 0}$ is witnessing left computability of α. □

2.6 Saturated Sequences

In the final section of this chapter we will look at infinite binary sequences which contain all (finite) binary words as subsequences. Such sequences are called saturated. As we will show later, the class of languages corresponding to such sequences will coincide with some of our genericity notions based on finite automata. Since these sequences are of independent interest, however, we will deal with them already in this part of our thesis.

2.6.1 Definitions and Basic Facts

Definition 2.100 A sequence α is *saturated* (or *disjunctive*) if every binary word occurs in α as a subsequence, i.e., if for every word $w \in \Sigma^*$ there is a number $n \geq 0$ such that $\alpha(n)...\alpha(n+|w|-1) = w$. A language A is *saturated* if its characteristic sequence $\chi(A)$ is saturated.

Saturated sequences have been studied in the literature under various names (rich, disjunctive, etc.). Jürgensen and Thierrin (1988) were probably the first who explicitly investigated these sequences. They introduced the term *disjunctive sequence* since they related these sequences to the so-called disjunctive languages. Disjunctivity of a language and disjunctivity of its characteristic sequence, however, are not equivalent (see Theorem 2.130 below). So, since we will often identify a language and its characteristic sequence, we prefer the term of a *saturated* sequence and language here though it might be less popular. For a recent survey on saturated sequences see Calude et al. (1997).

Saturated sequences are abundant as Staiger (see Staiger (1976), Staiger (1998), Staiger (2002)) has observed.

Theorem 2.101 *(Staiger) The class of saturated sequences is comeager and has measure 1.*

Simple examples of saturated sequences are the sequence obtained by concatenating all nonempty binary words in length-lexicographical order

$$\alpha_0 = z_1\ z_2\ z_3\ z_4\ z_5\ z_6\ z_7\ \dots = 0\ 1\ 00\ 01\ 10\ 11\ 000\ \dots$$

and the sequence obtained by concatenating the binary numbers in order

$$\alpha_1 = bin(0)\ bin(1)\ bin(2)\ bin(3)\ bin(4)\ bin(5)\ \dots = 0\ 1\ 10\ 11\ 100\ 101\ \dots$$

The latter sequence is known as the *Champernowne sequence* (for base 2).

It is well known that in a saturated sequence any word does not only occur just once but infinitely often.

Proposition 2.102 *Let α be saturated and let $w \in \Sigma^*$. Then w occurs in α infinitely often, i.e., there are infinitely many numbers n such that $\alpha(n)\ldots\alpha(n+|w|-1) = w$.*

PROOF. This immediately follows from the fact that, by saturation of α, w^n occurs in α for all $n \geq 0$. □

In general, however, we can't say anything about the relative frequency with which a word w occurs in a saturated sequence. A sequence α in which words of the same length occur with the same frequency is called *normal*.

Definition 2.103 (Hardy and Wright (1979)) A sequence α is *normal* if, for any words $v, w \in \Sigma^*$ where $|v| = |w| \geq 1$,

$$\lim_{n\to\infty} \frac{|\{m < n : \alpha(m)\ldots\alpha(m+|v|-1) = v\}|}{|\{m < n : \alpha(m)\ldots\alpha(m+|w|-1) = w\}|} = 1. \tag{2.24}$$

Note that a sequence α is normal if only if, for any word $w \in \Sigma^*$,

$$\lim_{n\to\infty} \frac{|\{m < n : \alpha(m)\ldots\alpha(m+|w|-1) = w\}|}{n} = 2^{-|w|}.$$

Obviously, any normal sequence is saturated but not vice versa. The saturated sequences α_0 and α_1 given above are in fact normal. An example of a saturated sequence which is not normal is the sequence

$$\alpha_2 = z_1\, 0^1\, z_2\, 0^2\, z_3\, 0^3\, z_4\, 0^4 \ldots = 0\ 0\ 1\ 00\ 00\ 000\ 01\ 0000 \ldots$$

In this sequence the frequency of zeroes is higher than the frequency of ones. To be more precise, by $|z_n| = O(\log(n))$,

$$\lim_{n\to\infty} \frac{|\{m < n : \alpha(m) = 1\}|}{|\{m < n : \alpha(m) = 0\}|} = 0$$

whence (2.24) fails for the words $v = 1$ and $w = 0$. By introducing longer blocks of zeroes we can modify this argument in order to get saturated sequences with very sparse corresponding sets.

Lemma 2.104 *Let $f : \mathbb{N} \to \mathbb{N}$ be nondecreasing and unbounded. There is a saturated sequence α such that $|S(\alpha) \upharpoonright n| \leq f(n)$ for all $n \geq 0$. (Here we interprete $S(\alpha) \upharpoonright n$ as a set, namely $S(\alpha) \upharpoonright n = \{z_m : m < n \,\&\, z_m \in S(\alpha)\}$.)*

PROOF. By a finite extension argument we define a sequence α with the required properties. I.e., simultaneously with α we define a strictly increasing function $l : \mathbb{N} \to \mathbb{N}$ where $l(n)$ and $\alpha \upharpoonright l(n)$ are defined at stage n of the construction. For $n = 0$ we let $l(0) = 0$, hence $\alpha \upharpoonright l(0) = \varepsilon$. Given $l(n)$ and $\alpha \upharpoonright l(n)$, $l(n+1)$ and $\alpha \upharpoonright l(n+1)$ are defined as follows. Fix $p > l(n)$ minimal such that $f(l(n)) + |z_{n+1}| < f(p)$, let $l(n+1) = p + |z_{n+1}|$ and define the extension $\alpha \upharpoonright l(n+1)$ of $\alpha \upharpoonright l(n)$ by letting $\alpha(m) = 0$ for $l(n) \leq m < p$ and $\alpha(p)...\alpha(l(n+1)-1) = z_{n+1}$. Then, by a straightforward induction on n, z_n occurs in $\alpha \upharpoonright l(n)$ and, for $m \leq l(n)$, $|S(\alpha) \upharpoonright m| \leq f(m)$. Obviously this implies that α has the required properties. □

We can also distinguish between saturated and normal sequences by looking at the size of the corresponding classes. As we have noted above, saturated sequences are abundant in the sense of both, measure and category. Though, as Hardy and Wright (1979) have shown, normal sequence are also abundant in the sense of measure this is not true for category.

Theorem 2.105 *1. (Hardy and Wright (1979)) The class of normal sequences has measure 1.*

2. The class of normal sequences is meager.

The second part of the theorem will be a direct consequence of one of or results on finite-state genericity given in Chapter 5 below (see Theorem 5.26).

2.6.2 Closure Properties and Some Technical Properties

For our investigation of the complexity of saturated sequences and languages and their relation to genericity it will be useful to have some closure properties and some further technical properties of saturated sequences. We start with some obvious closure properties.

Proposition 2.106 *(i) The class of saturated sequences is closed under finite variants. I.e., if α is saturated and $\beta(n) = \alpha(n)$ for almost all n then β is saturated too. (ii) The class of saturated sequences is closed under finite shifts. I.e. if α is saturated, $w \in \Sigma^*$ and $n \geq 1$ then the sequences $\beta = w\alpha$ and $\gamma = \alpha(n)\alpha(n+1)\alpha(n+2)...$ are saturated too. (iii) The class of saturated sequence is closed under duality. I.e., if α is saturated then the dual sequence $\hat{\alpha} = (1-\alpha(0))\,(1-\alpha(1))\,(1-\alpha(2))\,...$ is saturated too.*

PROOF. Parts (i) and (ii) are immediate by Proposition 2.102. Part (iii) is obvious. □

Proposition 2.106 (i) can be extended as follows. If a sequence β differs from a saturated sequence α at infinitely many places but the places at which the sequences differ are separated by longer and longer intervals then the sequence β is saturated too. In order to state this more formally we use the following notion of closeness.

Definition 2.107 A sequence β is *close* to a sequence α if there is a strictly increasing function $f : \mathbb{N} \to \mathbb{N}$ such that

$$\liminf_{n\to\infty} f(n+1) - f(n) = \infty \tag{2.25}$$

and

$$\forall\, n \geq 0\ (\alpha(n) \neq \beta(n) \Rightarrow n \in range(f)). \tag{2.26}$$

A set B is *close* to a set A if the characteristic sequence of B is close to the characteristic sequence of A.

Lemma 2.108 *Let α and β be infinite sequences such that α is saturated and β is close to α. Then β is saturated.*

PROOF. Given a word x, we have to show that x occurs in β. By closeness of β to α we may fix a strictly increasing function f such that (2.25) and (2.26) hold. Then, by (2.25), we may choose a number n_0 such that

$$\forall\, n \geq n_0\ (f(n+1) - f(n) > 2|x|).$$

It follows, by (2.26), that, for any $m \geq f(n_0)$, the words $\alpha(m)\ldots\alpha(m+2|x|-1)$ and $\beta(m)\ldots\beta(m+2|x|-1)$ differ at most one place. Hence $\alpha(m)\ldots\alpha(m+|x|-1) = \beta(m)\ldots\beta(m+|x|-1)$ or $\alpha(m+|x|)\ldots\alpha(m+2|x|-1) = \beta(m+|x|)\ldots\beta(m+2|x|-1)$. On the other hand, since α is saturated, it follows from Proposition 2.102 that the word xx occurs in α infinitely often. So there is a number $m \geq f(n_0)$ such that $\alpha(m)\ldots\alpha(m+2|x|-1) = xx$, i.e., $\alpha(m)\ldots\alpha(m+|x|-1) = x$ and $\alpha(m+|x|)\ldots\alpha(m+2|x|-1) = x$. So, by the above, $\beta(m)\ldots\beta(m+|x|-1) = x$ or $\beta(m+|x|)\ldots\beta(m+2|x|-1) = x$. Hence x occurs in β. □

We next look at saturated sets, i.e., sets corresponding to saturated sequences. Proposition 2.106 immediately implies the following closure properties.

Proposition 2.109 *The class of saturated languages is closed under finite variants and under complement.*

Note that by viewing a sequence α as the characteristic sequence of a language A we implicitly impose some additional structure on α. For instance, membership in A of the words of a given length k is determined by some interval of α, namely, for the 2^k words $z^k_0, \ldots, z^k_{2^k-1}$ of length k in lexicographical order, $A(z^k_0)\ldots A(z^k_{2^k-1}) = \alpha(2^k)\ldots\alpha(2^{k+1}-1)$. This gives a partition of the infinite sequence α in the finite subsequences $\alpha_k = \alpha(2^k)...\alpha(2^{k+1}-1)$, α_k determining membership of the words of length k in A ($k \geq 0$). These subsequences α_k can be further partitioned by considering membership in A of the extensions of fixed length of a given word w. To be more precise, given a number $k \geq 0$ and a word w of length n, membership in A of the words wx with $|x| = k$ is determined by a subinterval of α_{n+k}, i.e., $A(wz_0^k)...A(wz_{2^k-1}^k)$ is a subword of α_{n+k}. As we will show next, in a saturated sequence α, for any word x, we can find occurences of x which are compatible with the just described partitions of α.

Lemma 2.110 *Let α be a saturated sequence and let A be the set corresponding to α. Then the following holds.*

(a) For any word $x \in \Sigma^*$ there are infinitely many numbers $n \geq 0$ such that

$$\exists i(0 \leq i \leq 2^n - (|x|-1) \text{ and } A(z_i^n)...A(z_{i+|x|-1}^n) = x) \tag{2.27}$$

(b) For any word $x \in \Sigma^*$ such that $|x| = 2^m$ for some number $m > 0$ there are infinitely many words w such that

$$A(wz_0^m)...A(wz_{2^m-1}^m) = x. \tag{2.28}$$

PROOF. For a proof of part (a) fix a word x and a number n_0. We have to show that (2.27) holds for some $n \geq n_0$. W.l.o.g. we may assume that $|x| < 2^{n_0}$. Now, by Proposition 2.102, the word xx occurs in α infinitely often. So we may fix $m > 2^{n_0}$ such that

$$A(z_m)...A(z_{m+2|x|-1}) = \alpha(m)...\alpha(m+2|x|-1) = xx.$$

Then, by $m > 2^{n_0}$, $|z_m| \geq n_0$ and, by $|x| < 2^{n_0}$, $|z_{m+2|x|-1}| \leq |z_m| + 1$. So $|z_m| = ... = |z_{m+|x|-1}|$ or $|z_{m+|x|}| = ... = |z_{m+2|x|-1}|$. Since

$$A(z_m)...A(z_{m+|x|-1}) = A(z_{m+|x|})...A(z_{m+2|x|-1}) = x$$

this implies that (2.27) holds for $n = |z_m| \geq n_0$ or $n = |z_m| + 1 \geq n_0$.

For a proof of part (b) fix a word x of exponential length, say $|x| = 2^m$, and let n_0 be given. It suffices to show that there is a word w of length at least n_0 such that (2.28) holds.

Let $\hat{x} = x(0x)^{2^m}$. Note that $\hat{x}$ consists of $2^m + 1$ copies of the word x each copy separated by a 0 from the next copy. Hence for any $k < 2^m$ there is a number $j_k < 2^m \cdot (2^m + 1)$ such that

$$j_k = k \mod 2^m \ \ \& \ \ \hat{x}(j_k)...\hat{x}(j_k + 2^m - 1) = x \tag{2.29}$$

Now, by the first part of the lemma, we may fix $n \geq n_0 + m$ such that

$$A(z_i^n)...A(z_{i+|\hat{x}|-1}^n) = \hat{x}$$

for some $i \leq 2^n - (|\hat{x}| - 1)$. It follows with (2.29) that there is a number p such that $p \leq 2^n - (2^m - 1)$ and

$$p = 0 \mod 2^m \ \& \ A(z_p^n)\ldots A(z_{p+2^m-1}^n) = x. \tag{2.30}$$

Fix such a number p and fix q correspondingly such that $p = q \cdot 2^m$. Then, by definition of the length-lexicographical ordering,

$$z_p^n \ldots z_{p+2^m-1}^n = z_q^{n-m} z_0^m \ldots z_q^{n-m} z_{2^m-1}^m.$$

So (2.30) implies that $A(wz_0^m)\ldots A(wz_{2^m-1}^m) = x$ for the word $w = z_q^{n-m}$. This completes the proof. □

We conclude our investigation of the closure properties of the saturated sets by the observation that, for a saturated language A and for any word w, the language $wA = \{wv : v \in A\}$ is saturated too.

Lemma 2.111 *Let A be saturated.*

1. *For any word $w \in \Sigma^*$, wA is saturated too. In fact, any set B such that $B \cap w\Sigma^* = wA$ is saturated.*

2. *For any set B, the effective disjoint union of A and B, $A \oplus B = 0A \cup 1B = \{0v : v \in A\} \cup \{1w : w \in B\}$, is saturated.*

PROOF. For a proof of the first part, fix a word $w \in \Sigma^*$ and a language B such that $B \cap w\Sigma^* = wA$. Then, given $x \in \Sigma^*$, we have to show that x occurs in the characteristic sequence of B. By the first part of Proposition 2.110 there are numbers n and i such that $A(z_i^n)\ldots A(z_{i+|x|-1}^n) = x$. So, by choice of B, $B(wz_i^n)\ldots B(wz_{i+|x|-1}^n) = x$. Since $wz_i^n, \ldots, wz_{i+|x|-1}^n$ are consecutive words with respect to the length-lexicographical ordering this implies that x occurs in $\chi(B)$.

For a proof of the second part it suffices to note that $A \oplus B$ and $0A$ agree on $0\Sigma^*$. So the claim follows from the first part. □

2.6.3 Saturated Sequences and Languages and the Chomsky Hierarchy

We now will measure the complexity of saturated sequences and languages in terms of the Chomsky hierarchy. Calude and Yu (1997) have investigated the Chomsky complexity of the prefix sets of saturated sequences thereby classifying the Chomsky classes of sequences (in the sense of Section 2.5) which contain saturated sequences. Their negative results are based on the following observation.

Lemma 2.112 *(Calude and Yu (1997)) Let α be almost periodic. Then α is not saturated.*

PROOF (IDEA). Fix $v,w \in \Sigma^*$ such that $\alpha = vw^\omega$ and let $n = |v|+|w|$. Then 0^n or 1^n does not occur in α. Namely, if the letter 1 occurs in w then 0^n does not occur in α, and 1^n does not occur in α otherwise. □

Theorem 2.113 *(Calude and Yu (1997)) There is a saturated sequence α such that $Prefix(\alpha)$ is context-sensitive but there is no saturated sequence with context-free prefix set.*

PROOF (IDEA). For the first part of the theorem it suffices to observe that, for the canonical saturated sequence $\alpha = z_0\, z_1\, z_2\, z_3\, \dots$, $Prefix(\alpha)$ can be recognized by a linear-space bounded Turing machine whence $Prefix(\alpha)$ is context-sensitive. The second part of the theorem follows from Lemma 2.112 since, by Theorem 2.76, a sequence is almost periodic if and only if its prefix set is context-free. □

Corollary 2.114 *There is a context-sensitive sequence which is saturated but no context-free sequence is saturated.*

PROOF. This is immediate by Theorem 2.113 and Definition 2.73. □

By the coincidence of the regular sequences with the context-free sequences (Theorem 2.76) the classification of the saturated sequences in the Chomsky hierarchy of sequences is rather coarse. We get a better lower and upper bounds on the complexity of saturated sequences if we look at the Chomsky language classes which contain saturated languages, i.e., languages corresponding to saturated sequences. As we will show next, no regular language is saturated but there are context-free, in fact linear, languages which are saturated.

Theorem 2.115 *Let α be a saturated sequence and let A be the set corresponding to α. Then A is not regular.*

PROOF. For a contradiction assume that A is regular. Fix a deterministic finite automaton M which accepts A and let p be the number of states of M. Finally, fix m such that $p < 2^m$.

Then for any set S of 2^m words there are at least two words u and u' in S such that M is in the same state after reading u and u', hence for any word v the extension of u by v is in A if and only if the corresponding extension of u' is in A. I.e., for any set S,

$$|S| = 2^m \Rightarrow \exists u, u' \in S(u \neq u' \ \& \ \forall v(uv \in A \Leftrightarrow u'v \in A)). \tag{2.31}$$

Now, in order to get the desired contradiction, in the following we will produce a counterexample to (2.31). For this sake we first consider the 2^m words of length 2^m which contain a unique 1 and take their concatenation:

$$x_i = 0^i 1 0^{2^m-(i+1)} (0 \leq i < 2^m) \ \& \ x = x_0 \ldots x_{2^m-1}. \tag{2.32}$$

Note that $|x| = 2^m 2^m = 2^{2m}$. So, by saturation of α and by Lemma 2.110, there is a word w such that

$$A(wz_0^{2m}) \ldots A(wz_{2^{2m}-1}^{2m}) = x. \tag{2.33}$$

Since

$$wz_0^{2m} \ldots wz_{2^{2m}-1}^{2m} = wz_0^m z_0^m \ldots wz_0^m z_{2^m-1}^m \ldots wz_{2^m-1}^m z_0^m \ldots wz_0^m 2^m - 1_{2^m-1}^m$$

it follows, by choice of x, that

$$A(wz_i^m z_0^m) \ldots A(wz_i^m z_{2^m-1}^m) = x_i$$

for $i < 2^m$. By choice of the strings x_i this implies

$$A(wz_i^m z_j^m) = 1 \Leftrightarrow i = j \ (0 \leq i, j < 2^m). \tag{2.34}$$

So, for $S = \{wz_i^m : i < 2^m\}$, $|S| = 2^m$. Moreover, by (2.34), for any words $u \neq u' \in S$, say $u = wz_i^m$ and $u' = wz_{i'}^m$ where $i \neq i'$, there is a word v, namely $v = z_i^m$, such that $uv \in A$ but $u'v \notin A$. But this contradicts (2.31). □

Theorem 2.116 *There is a saturated sequence α such that the language A corresponding to α is linear.*

PROOF. For a nonempty word $x = a_1 \ldots a_n \in \Sigma^n$ let $x^D = a_1 a_1 \ldots a_n a_n$ be the *duplication* of x and let $x^R = a_n \ldots a_1$ be the *reversal* of x ($\varepsilon^D = \varepsilon^R = \varepsilon$). Then the required language A is defined by

$$A = \{x_1^D 01 x_2^D 01 \ldots x_n^D 10 x_m^R : 1 \leq m \leq n \ \& \ x_1, \ldots, x_n \in \Sigma^+\}.$$

Intuitively, A is the set of all words $w = \langle x_1, \ldots, x_n \rangle x_{n+1}$ where the first part of w codes a nonempty finite sequence of words $x_1, \ldots, x_n$ (namely $\langle x_1, \ldots, x_n \rangle =$

$(x_1^R)^D 01 \ldots (x_n^R)^D 10)$ and the second part x_{n+1} of w coincides with one of the members of this sequence (note that $x^{RR} = x$).

The coding is chosen in such a way that the language A is linear. For instance, as one can easily check, the linear grammar G with the following rules generates A.

$$\begin{array}{lcl} S & \rightarrow & 00V \mid 11V \mid T \\ V & \rightarrow & 00V \mid 11V \mid 01S \\ T & \rightarrow & 00T0 \mid 11T1 \mid 00U0 \mid 11U1 \\ U & \rightarrow & 01W \mid 10 \\ W & \rightarrow & 00W \mid 11W \mid 00U \mid 11U \end{array}$$

Here S is the axiom of the grammar G and T, U, V, W are the other variables of G. Note that the S- and V-rules can produce deductions $S \overset{*}{\Rightarrow} wT$ where $w = \varepsilon$ or $w = x_1^D 01 \ldots x_n^D 01$ for some $n \geq 1$, the T-rules give deductions $T \overset{*}{\Rightarrow} x^D U x^R$, and the U- and W-rules allow deductions of the form $U \overset{*}{\Rightarrow} 10$ or $U \overset{*}{\Rightarrow} 01x_1^D \ldots 01x_n^D$ for $n \geq 1$.

It remains to show that A is saturated, i.e., that any given string x occurs in the characteristic sequence α of A. So fix x where w.l.o.g. we may assume that $|x| = 2^n$ for some $n \geq 1$ and that at least one 1 occurs in x (otherwise consider an extension of x with these properties). Then $x = a_0 \ldots a_{2^n-1}$ and $P = \{j : 0 \leq j < 2^n \,\&\, a_j = 1\} \neq \emptyset$. Fix $0 \leq j_1 < j_2 < \cdots < j_l < 2^n$ such that $P = \{j_1, \ldots, j_l\}$ and let $w = ((z_{j_1}^n)^R)^D 01((z_{j_2}^n)^R)^D 01 \ldots ((z_{j_l}^n)^R)^D$ and $w_j = w10z_j^n$ $(0 \leq j < 2^n)$. Then $w_0, \ldots, w_{2^n-1}$ are consecutive words, whence the sequence $A(w_0) \ldots A(w_{2^n-1})$ occurs in α. On the other hand, by definition, $w_j \in A$ iff $j \in P$ whence

$$x = A(w_0) \ldots A(w_{2^n-1}).$$

So x occurs in α which completes the proof. □

We conclude this subsection by some observations on the relation between immunity and saturation. In Section 2.4 we have studied immunity notions as a means for obtaining strong separations between complexity classes. There we have shown that any language which is bi-immune to the class of regular languages cannot be context-free but that there are linear languages which are immune to REG. The former observation together with Theorem 2.116 shows that there is a saturated sequence such that the corresponding set is not bi-immune to REG. Next we will extend this result by showing that there is a saturated language A such that neither A nor the complement of A is immune to REG.

Theorem 2.117 *There is a saturated sequence* α *such that neither* $S(\alpha)$ *nor* $\overline{S(\alpha)}$ *is* REG-*immune.*

PROOF. Let B be any saturated set and define A by letting

$$A = (B \cup \{0^n : n \geq 1\}) \setminus \{1^n : n \geq 1\}.$$

Then neither A nor $\overline{A}$ is REG-immune since the infinite regular set $\{0^n : n \geq 1\}$ is contained in A and the infinite regular set $\{1^n : n \geq 1\}$ is contained in $\overline{A}$. It remains to show that A is saturated, i.e., given any word x we have to show that $x = A(z_n) \ldots A(z_{n+|x|-1})$ for some number n. Now, by saturation of B and by Lemma 2.110, there are numbers m and k such that $B(z^k_m) \ldots B(z^k_{m+|x|+1}) = 0x0$. By choice of A this implies that $A(z^k_{m+1}) \ldots A(z^k_{m+|x|}) = x$. This completes the proof. (Alternatively, saturation of A can be deduced from Lemma 2.108 since $B \cup \{0^n : n \geq 1\}$ is close to B and A is close to $B \cup \{0^n : n \geq 1\}$.) □

The above shows that saturation does not imply REG-immunity. The converse is also true as the following lemma shows, i.e., saturation and immunity are independent concepts.

Lemma 2.118 *There is a* REG-*bi-immune language A which is not saturated.*

PROOF. By Theorem 2.52 let A' be any REG-bi-immune language. Define A by letting $A = \{wi : w \in A' \ \& \ i = 0,1\}$. Then, as one can easily check, A is REG-bi-immune too. On the other hand, by choice of A, the characteristic sequence of A is the duplication of the charateristic sequence of A' whence the word 010 does not occur in $\chi(A)$. So A is not saturated. □

2.6.4 Saturation and Predictability

In Section 2.5.5 we have shown that the Chomsky complexity of the prefix set of a sequence is closely related to predictability. For instance we have seen that that a sequence is regular if and only if the sequence can be predicted by a (deterministic or nondeterministic) finite automata or by a deterministic push-down automaton. So, by Theorem 2.113, a saturated sequence cannot be predicted by such an automaton. In Section 2.5.5 we have also shown, however, that there are nonregular sequences which can be predicted by a nondeterministic push-down automaton. Our above results on the Chomsky complexity of saturated sequences do not settle the question whether there are saturated sequences which can be predicted by a nondeterministic push-down automaton and we leave this as an open problem. In the following we will show, however, that the saturated sequences can be characterized in terms of partial predictability by finite automata. This characterization will be used for establishing the relations between saturation and finite-state genericity in Section 4.1.

Definition 2.119 Let M be an extended deterministic finite automaton and let $f_M : \Sigma^* \to \Sigma$ be the partial function computed by M. Then M *partially predicts* (or *infinitely often predicts*) the sequence α if M is consistent with α, i.e.,

$$\forall\, n \geq 0\ (f_M(\alpha \restriction n) \downarrow\ \Rightarrow\ \ f_M(\alpha \restriction n) = \alpha(n)) \tag{2.35}$$

and M makes infinitely predictions about α, i.e.,

$$\exists^\infty\, n \geq 0\ (f_M(\alpha \restriction n) \downarrow). \tag{2.36}$$

Theorem 2.120 *For any sequence* α *the following are equivalent.*

1. α *is saturated.*
2. *There is no deterministic finite automaton which partially predicts* α*.*

PROOF. For a proof of the implication $1 \Rightarrow 2$ assume that α is saturated and let M be an EDFA such that the function f_M computed by M satisfies (2.36). We have to show that (2.35) fails, i.e., that there is a number n such that $f_M(\alpha \restriction n)$ is defined and $f_M(\alpha \restriction n) \neq \alpha(n)$.

Let M be the automaton $M = (\Sigma, S, \delta, s_0, \lambda)$ where the partial state labeling function $\lambda : S \to \Sigma$ describes the different types of states of M as follows. For a state s, $f(s) = 1$ if s is an acccepting state, $f(s) = 0$ if s is a rejecting state, and $f(s)$ is undefined if s is an undetermined state. Then the function f_M computed by M is defined by

$$f_M(x) = \lambda(\delta^*(s_0, x)). \tag{2.37}$$

Let $s_1, \ldots s_m$ be the set of states s visited by M infinitely often when reading α, i.e., the set of states s such that $\delta^*(s_0, \alpha \restriction n) = s$ for infinitely many numbers n. Then, by (2.36) and by (2.37), there is a state s_k ($1 \leq k \leq m$) such that $\lambda(s_k)$ is defined. By symmetry, w.l.o.g. we may assume that $k = 1$ and $\lambda(s_1) = 1$. So in order to show that (2.35) fails it suffices to show that there is a number n such that

$$\delta^*(s_0, \alpha \restriction n) = s_1\ \ \&\ \ \alpha(n) = 0. \tag{2.38}$$

To show this we consider the string $x \in \Sigma^*$ defined as follows. Let

$$x = y_1 0 y_2 0 \ldots y_m 0$$

where the substrings y_k are defined by induction on k by letting $y_1 = \lambda$ and by letting y_k for $2 \leq k \leq m$ be the least string y such that

$$\delta^*(s_k, y_1 0 \ldots y_{k-1} 0 y) = s_1$$

if such a string y exists and by letting $y_k = \lambda$ otherwise.

Now, by saturation of α and by Proposition 2.102, the string x occurs in α infinitely often. So we may fix $n_0 \geq 0$ and $1 \leq k \leq m$ such that

$$\delta^*(s_0, \alpha \upharpoonright n_0) = s_k \tag{2.39}$$

and

$$(\alpha \upharpoonright n_0)\, x \sqsubseteq \alpha. \tag{2.40}$$

hold. Now, if $k = 1$ then, by definition of x, $y_1 = \lambda$ hence $x(0) = 0$. By (2.39) and (2.40) this implies that (2.38) holds for $n = n_0$. So in the following we may assume that $2 \leq k \leq m$. Then, by definition of x, $x_{k-1} = y_1 0 y_2 0 \ldots y_{k-1} 0$ is an initial segment of x. So, by (2.40), for $n_1 = |x_{k-1}|$, $\alpha \upharpoonright (n_0 + n_1) = (\alpha \upharpoonright n_0)\, x_{k-1}$. Moreover, since M runs through the state s_1 infinitely often when reading α, there is a number $n_2 \geq n_0 + n_1$ such that $\delta^*(s_0, \alpha \upharpoonright n_2) = s_1$. By (2.39) this implies that, for the string $y = \alpha(n_0 + n_1) \ldots \alpha(n_2 - 1)$,

$$\delta^*(s_k, x_{k-1} y) = \delta^*(s_0, (\alpha \upharpoonright n_0) x_{k-1} y) = \delta^*(s_0, \alpha \upharpoonright n_2) = s_1$$

So, by definition of x, $\delta^*(s_k, y_1 0 \ldots y_{k-1} 0 y_k) = s_1$. Since $y_1 0 \ldots y_{k-1} 0 y_k 0$ is an initial segment of x it follows, by (2.39) and (2.40), that (2.38) holds for $n = n_0 + |y_1 0 \ldots y_{k-1} 0 y_k|$.

The proof of the implication $2 \Rightarrow 1$ is by contraposition. Assume that the sequence α is not saturated. We will show that there is an extended deterministic finite automaton M which partially predicts α. By assumption we may fix a string x of minimal length such that x occurs in the sequence α at most finitely often, say $(\alpha \upharpoonright n) x \not\sqsubseteq \alpha$ for all $n \geq n_0$. Then, obviously, $|x| > 0$ whence we may fix $i \in \Sigma$ and $y \in \Sigma^*$ such that $x = yi$. Define the partial function $f : \Sigma^* \to \Sigma$ by

$$f(z) = \begin{cases} 1 - i & \text{if } z = vy \text{ for some string } v \in \Sigma^{\geq n_0} \\ \uparrow & \text{otherwise.} \end{cases}$$

Note that, by minimality of x, the string y occurs in α infinitely often whence $f(\alpha \upharpoonright n) \downarrow$ for infinitely many n. Moreover, for any m such that $f(\alpha \upharpoonright m)$ is defined, $f(\alpha \upharpoonright m) = \alpha(m)$. (Namely, if $f(\alpha \upharpoonright m)$ is defined then there is a number $n \geq n_0$ such that $\alpha \upharpoonright m = (\alpha \upharpoonright n) y$ whence, by choice of n_0, $\alpha \upharpoonright (m+1) \neq (\alpha \upharpoonright n) x$. By $x = yi$, however, this implies that $\alpha(m) \neq i$, i.e., $\alpha(m) = 1 - i$.) Hence (2.35) and (2.36) hold if we replace f_M by f. This completes the proof since, as one can easily show, f can be computed by an EDFA M. □

In Section 2.5.5 we have shown that total predictability by deterministic pushdown automata coincides with total predictability by deterministic finite automata. For partial predictability, however, these devices are not equivalent. By Theorem 2.120 this follows from the following result of Merkle and Reimann.

Theorem 2.121 *(Merkle and Reimann (2003)) There is a saturated (in fact normal) sequence α which can be partially predicted by a deterministic pushdown automaton.*

PROOF (IDEA). The following example of a sequence α with the required properties is somewhat simpler than the examples given in Merkle and Reimann (2003). Let α be the saturated (in fact normal) sequence obtained by concatenating all binary words in order, i.e., $\alpha = z_0 z_1 z_2 \ldots$. Then, for any number n, the initial segment $\alpha \upharpoonright n$ contains at least as many occurrences of the bit 0 as of the bit 1. Moreover, the number of occurences is equal if and only if $\alpha \upharpoonright n$ consists of all words up to a given length, i.e., $\alpha \upharpoonright n = z_0^0 \ldots z_{2^m-1}^m$ for some number m. So, for such an n, $\alpha(n)$ will be the first bit of the word z_0^{m+1} which is a 0. A deterministic pushdown automaton M which correctly predicts these occurences of zeroes in α, pushes a 0 on its stack when it reads a 0, pops a 0 from the stack when it reads a 1, and predicts the next bit to be a 0 if the stack is empty. (Note that M works with the unary alphabet $\{0\}$ as its stack alphabet. DPDAs with this additional property are also called deterministic 1-counter automata.) □

2.6.5 Computational Complexity of Saturated Sequences and Languages

Following our investigations of the complexity of saturated sequences in the sense of formal language and automata theory, in this subsection we shortly discuss the computational complexity of saturated sequences. We show that for any set A there is a saturated sequence α such that the set $S(\alpha)$ corresponding to α is equivalent to A under linear-time many-one reducibility. Roughly speaking, this says that there are saturated languages of any given time complexity. As we will also note, however, the corresponding fact for the prefix sets of saturated sequences fails.

We start with the observation that there is a saturated sequence such that the corresponding set is linear time computable.

Lemma 2.122 *Let $D = \{z_i^n : z_n(i) = 1\}$. Then the characteristic sequence $\chi(D)$ of D is saturated and $D \in \mathrm{DTIME}(O(n))$.*

PROOF (IDEA). Note that for the nth word z_n occurrence of z_n in $\chi(D)$ is guaranteed by the first $|z_n|$ words of length n, namely $D(z_0^n) \ldots D(z_{|z_n|-1}^n) = z_n$. We omit the straightforward but somewhat tedious proof for $D \in \mathrm{DTIME}(O(n))$. □

Recall that $A \leq_m^{lin} B$ (A *is many-one reducible to B in linear time*) if there is a linear-time computable function $f : \Sigma^* \to \Sigma^*$ such that $x \in A$ if and only if $f(x) \in B$ (for all $x \in \Sigma^*$); and that $A =_m^{lin} B$ (A *is many-one equivalent to B in linear time*) if $A \leq_m^{lin} B$ and $B \leq_m^{lin} A$.

Theorem 2.123 *For any set $A \neq \emptyset, \Sigma^*$ there is a saturated set B such that $A =_m^{lin} B$.*

PROOF. Fix $A \neq \emptyset, \Sigma^*$, choose D as in Lemma 2.122, and let $B = D \oplus A = \{0x : x \in D\} \cup \{1x : x \in A\}$ be the effective disjoint union of D and A. Then B is saturated by Lemma 2.111 and by saturation of D. $A =_m^{lin} B$ follows from $D \in \mathrm{DTIME}(O(n))$ as follows. Obviously, $A \leq_m^{lin} B$ via $f(x) = 1x$. For a proof of $B \leq_m^{lin} A$ fix words x_0 and x_1 such that $x_0 \in A$ and $x_1 \notin A$. Then $B \leq_m^{lin} A$ via g where

$$g(0x) = \begin{cases} x_0 & \text{if } x \in D \\ x_1 & \text{otherwise.} \end{cases}$$

and $g(1x) = x$. Note that $g(0x)$ can be computed in linear time since D is linear-time computable. □

For space complexity we easily obtain the corresponding results for logarithmic space in place of linear time. In fact in Theorem 2.123 we may replace *lin-m* reducibility by simultaneously linear-time and logarithmic-space bounded many-one reducibility. Calude and Yu (1997) have shown that there is a saturated sequence with prefix set in $\mathrm{DSPACE}(O(\log n))$. In fact, as one can easily check, for D as in Lemma 2.122, $Prefix(\chi(D)) \in \mathrm{DTIME}(O(n)) \cap \mathrm{DSPACE}(O(\log n))$.

Lemma 2.124 *There is a saturated sequence α such that*

$$Prefix(\alpha) \in \mathrm{DTIME}(O(n)) \cap \mathrm{DSPACE}(O(\log n)).$$

In contrast to Theorem 2.123, however, there are *lin-m* equivalence classes - in fact polynomial-time Turing equivalence classes - which do not contain the prefix set of any saturated sequence (in fact no prefix set at all). This follows from some general results on sets of low nonuniform complexity. Note that any prefix set A contains just one word of each length whence A is *sparse*. Any sparse set A possesses polynomial-time circuits, i.e., is a member of the class $\mathrm{P}/poly$ and it has been shown that there are polynomial-time Turing equivalence classes which do not intersect $\mathrm{P}/poly$ (see e.g. Balcázar et al. (1995), Chapter 5). So, in order to obtain an analog of Theorem 2.123 for prefix sets we have to work with some weaker reducibilities. E.g., we can show that, for any set A, there is the prefix set $Prefix(\alpha)$ of a saturated sequence α which is exponential-time Turing equivalent to A. In the following let $\leq_T^e$ denote exponential-time (i.e. $O(2^{O(n)})$ bounded Turing reducibility. Similarly, let $\leq_{tt}^e$ denote exponential-time bounded truth-table reducibility and $\leq_{tt}^p$ denote polynomial-time bounded truth-table reducibility

Theorem 2.125 *For any set A there is a saturated sequence α such that $A =_T^e Prefix(\alpha)$.*

This easily follows from the following lemma and Theorem 2.123.

Lemma 2.126 *For any sequence* α, $S(\alpha) =_T^e Prefix(\alpha)$. *In fact,* $S(\alpha) \leq_T^e Prefix(\alpha)$ *and* $Prefix(\alpha) \leq_{tt}^p S(\alpha)$.

PROOF. In order to show $S(\alpha) \leq_T^e Prefix(\alpha)$ consider the following reduction from $S(\alpha)$ to $Prefix(\alpha)$: Given x, $|x| = n$, compute the unique m such that $x = z_m$. Then, by m adaptive queries to $Prefix(\alpha)$, inductively compute $\alpha \upharpoonright (m+1)$ and accept x iff $(\alpha \upharpoonright (m+1))(m) = 1$. Since $m \leq 2^{n+1}$, this reduction can be done in time $O(2^{O(n)})$. For a proof of $Prefix(\alpha) \leq_{tt}^m S(\alpha)$, consider the following reduction from $Prefix(\alpha)$ to $S(\alpha)$: Given x, $|x| = n$, compute $\alpha(0), \ldots, \alpha(n-1)$ by letting the oracle $S(\alpha)$ answer the n queries $z_0, \ldots, z_{n-1}$. Accept the input x iff $x = \alpha(0)\ldots\alpha(n-1)$. This reduction is polynomial-time bounded and requests only a linear number of nonadaptive queries. □

2.6.6 Saturated and Disjunctive Languages

As pointed out before, saturated sequences are also called disjunctive sequences since they are closely related to disjunctive languages. Though we will not need the latter concept later, we will shortly describe the relations between saturated languages and disjunctive languages here.

Definition 2.127 (Shyr (1977)) A language A is *disjunctive* if any two words x and y can be distinguished by the context of their occurences in words in the language A, i.e., if

$$\forall x, y \in \Sigma^* \ (x \neq y \Rightarrow \exists u, v \in \Sigma^* (uxv \in A \Leftrightarrow uyv \notin A)) \tag{2.41}$$

holds.

For a disjunctive language A, any word z occurs in some element of A as a subword (namely, otherwise, (2.41) will fail for the words $x = z0$ and $y = z1$ since $uxv, uyv \notin A$ for all words u, v). The converse, however, is not true. For example any word occurs as a subword of a word in Σ^*, but all words occur in the same context (namely, for any x, $uxv \in \Sigma^*$ for all words u, v) whence Σ^* is not disjunctive. As Calude et al. (1997) have shown, however, for a prefix set A, A is disjunctive if every word occurs as a subword of a word in A, whence a sequence α is disjunctive (i.e. saturated) if and only if its prefix set $Prefix(\alpha)$ is disjunctive. This easily follows from the observation by Jürgensen and Thierrin (1988) that in the definition of a disjunctive language in (2.41) it suffices to consider words x and y of the same length.

Lemma 2.128 *(Jürgensen and Thierrin (1988)) A language A is disjunctive if and only if*

$$\forall n\,\forall x,y \in \Sigma^n (x \neq y \Rightarrow \exists u,v \in \Sigma^*(uxv \in A \Leftrightarrow uyv \notin A)) \tag{2.42}$$

holds.

Theorem 2.129 *(Calude et al. (1997)) For any infinite sequence* α *the following are equivalent.*

1. α *is saturated.*

2. *Every word* $x \in \Sigma^*$ *occurs as a subword in an element of* $Prefix(\alpha)$.

3. $Prefix(\alpha)$ *is disjunctive.*

PROOF. The implications $1 \Leftrightarrow 2$ and $3 \Rightarrow 2$ are straightforward. So it suffices to show the implication $1 \Rightarrow 3$. Assume that α is saturated. By Lemma 2.128 it suffices to show that, given x and y such that $|x| = |y|$ and $x \neq y$, there are words u and v such that uxv is a prefix of α but uyv is not a prefix of α. Now, since α is saturated, we may fix u such that ux is a prefix of α and let v be the empty string. Then uxv is a prefix of α. On the other hand, however, uyv is not a prefix of α since $|uxv| = |uyv|$ but $uxv \neq uyv$ and since the prefix of α of a given length is uniquely determined. □

The relation between saturation (disjunctivity) of a sequence and disjunctivity of its corresponding set is as follows.

Theorem 2.130 *(a) Let* α *be a saturated sequence. Then the set* $S(\alpha)$ *corresponding to* α *is disjunctive.*

(b) There is a disjunctive language A such that the characteristic sequence $\chi(A)$ *of A is not saturated.*

PROOF. *(a)* To show that $S(\alpha)$ is disjunctive it suffices to establish (2.42). So let $n \geq 0$ and words x and y with $x \neq y$ and $|x| = |y| = n$ be given. Fix j and k such that $x = z^n_j$ and $y = z^n_k$ and define $z \in \Sigma^{2^n}$ by $z(j) = 1$ and $z(i) = 0$ for all $i < 2^n$ with $i \neq j$. Then, by Lemma 2.110, there is a word w such that $A(wz^n_0)...A(wz^n_{2^n-1}) = z$. So, in particular, $A(wx) = A(wz^n_j) = z(j) = 1$ and $A(wy) = A(wz^n_k) = x(k) = 0$, whence $wx \in A$ and $wy \notin A$. So (2.42) holds for $u = w$ and $v = \lambda$.

(b) Let A be the language $A = \{ww^R : w \in \Sigma^*\}$ (where w^R is the reversal of w). Then, for any words x and y such that $|x| = |y|$ and $x \neq y$, $x^R x \in A$ whereas $x^R y \notin A$. So, by Lemma 2.128, A is disjunctive. As one can easily check, however, the word 11 does not occur in $\chi(A)$. So $\chi(A)$ is not saturated. □

Corollary 2.131 *Every saturated language is disjunctive but there are disjunctive languages which are not saturated.*

PROOF. Since, by definition, a language is saturated if and only if its characteristic sequence is saturated, this is immediate by Theorem 2.130. □

By Corollary 2.131, negative results on disjunctive languages carry over to saturated languages and positive results on saturated languages carry over to disjunctive languages. For instance, Shyr (1977) have shown that no disjunctive language is regular. So, by Corollary 2.131, no saturated language is regular. This gives an alternative proof of Theorem 2.115. Conversely, Theorem 2.116 and Corollary 2.131 imply that there are linear languages which are disjunctive. Here, however, we get simpler examples by a direct argument as the language $A = \{ww^R : w \in \Sigma^*\}$ in the proof of Theorem 2.130 shows. In fact, as one can easily show, the language $A = \{w^D 01 w^R : w \in \Sigma^*\}$ (where w^D is the duplication of w) is disjunctive too. On the other hand, A is deterministic context-free and linear. So there is a disjunctive language which is both, deterministic context-free and linear. We do not know, however, whether there are saturated languages which are deterministic context-free (or even both, deterministic-context free and linear).

2.6.7 Partial Saturation

We conclude this chapter on saturated sequences by some observations on partially saturated sequences. These results will be used later for separating some of our finite-state genericity notions. We begin with some definitions.

Definition 2.132 A sequence α is *k-n-saturated* if every word of length k occurs in α at least n times ($k,n \geq 1$); α is *ω-n-saturated*, if every word occurs in α at least n times, i.e., if α is k-n-saturated, for all $k \in \mathbb{N}_+$; α is *k-ω-saturated* if every word of length k occurs in α infinitely often, i.e., if α is k-n-saturated for all $n \in \mathbb{N}_+$; α is *ω-ω-saturated* if every word occurs infinitely often in α, i.e., if α is k-n-saturated for all $k,n \in \mathbb{N}_+$.

A set A is *k-n-saturated* ($k,n \in \mathbb{N}_+ \cup \{\omega\}$) if its characteristic sequence $\chi(A)$ is k-n-saturated.

The following relations among these notions are immediate by definition.

Proposition 2.133 *If $k \leq k'$ and $n \leq n'$ $(k,k',n,n' \in \mathbb{N}_+ \cup \{\omega\})$ then every k'-n'-saturated sequence is also k-n-saturated.*

Moreover, by definition, a sequence α is saturated if and only if α is ω-1-saturated and, by Proposition 2.102, saturation and ω-ω-saturation coincide. This immediately yields the following.

Proposition 2.134 *For any sequence* α *the following are equivalent.*

1. α *is saturated.*

2. α *is* ω-1-*saturated.*

3. α *is* ω-n-*saturated (for any fixed* $n \geq 1$*)* .

4. α *is* ω-ω-*saturated.*

For $1 \leq k, n < \omega$ there are finite k-n-saturated sets, namely the sequence $\alpha = (z_0^k \ldots z_{2^k-1}^k)^n 0^\omega$ is k-n-saturated and the corresponding set is finite. Since we will be only interested in infinite sets, in the following we will focus on k-n-saturated sequences where $k = \omega$ or $n = \omega$. Obviously, the set corresponding to such a sequence cannot be finite.

Proposition 2.135 *Let* α *be k-n-saturated where* $k = \omega$ *or* $n = \omega$*. Then the corresponding set* $S(\alpha)$ *is infinite and co-infinite.*

The following simple separation lemma will become useful later.

Lemma 2.136 *For any* $k \in \mathbb{N}_+$ *there is a* k-ω-*saturated sequence* α *which is not* $(k+1)$-1-*saturated, hence not* $(k+1)$-ω-*saturated.*

PROOF. Let $\alpha = (z_0^k \, 0 \, z_1^k \, 0 \, \ldots \, z_{2^k-1}^k \, 0)^\omega$. Then every word of length k occurs in α infinitely often whence α is k-ω-saturated. The word 1^{k+1} of length $k+1$, however, does not occur in α whence α is not $(k+1)$-1-saturated. □

The above observations on partial saturation give the following relations among the infinitary saturation notions where Lemma 2.136 implies that no other relations hold (for $k \geq 2, n \geq 1$).

$$\begin{array}{ccccc}
\alpha\ \omega\text{-}\omega\text{-saturated} & \Leftrightarrow & \alpha\ \omega\text{-}n\text{-saturated} & \Leftrightarrow & \alpha \text{ saturated} \\
\Downarrow & & & & \\
\alpha\ (k+1)\text{-}\omega\text{-saturated} & & & & \\
\Downarrow & & & & \\
\alpha\ k\text{-}\omega\text{-saturated} & & & & \\
\Downarrow & & & & \\
\alpha\ 1\text{-}\omega\text{-saturated} & & & &
\end{array}$$

Above we have shown that saturated sequences and languages are not regular. This is contrasted by the following.

Lemma 2.137 *For any* $k \geq 1$ *there is an* k-ω-*saturated sequence* α *which is regular.*

PROOF. The k-ω-saturated sequence α defined in the proof of Lemma 2.136 is periodic hence regular. □

For some recent results on representability and decidability questions for saturated sequences see Ambos-Spies and Busse (2004).

CHAPTER 3

Baire Category, Forcing, Genericity

In this chapter we discuss some of the fundamental relations among genericity, Baire category and some of the fundamental diagonalization techniques in computability and computational complexity theory. Our presentation is based on papers and lectures by Ambos-Spies on this topic, in particular on Ambos-Spies (1996). A more detailed treatment of classical Baire category theory can be found in Oxtoby (1980). More information on the role played by Baire category in computability theory is given in Odifreddi (1989).

In Section 3.1 we shortly review classical Baire category for the Cantor space. Then, in Section 3.2, we give alternative characterizations of this concept based on total respectively partial extension functions. In Section 3.3 we discuss the relations between category and (Lebesgue) measure. It is well known that in general the category and measure approaches for defining large classes are incompatible. As we will show here, however, any comeager class defined in terms of *bounded* extension functions has measure 1 too, i.e., is large in both senses. In Section 3.4 we review the finite extension method and an important refinement of this technique, namely the wait-and-see technique or slow-diagonalization technique. Moreover, we show how these techniques can be related to Baire category by exploiting the characterization of the latter in terms of total extension functions and partial extension functions, respectively. Finally, in Section 3.5 we introduce a general framework for genericity notions by attaching a genericity concept to any countable class $\mathcal{F}$ of (total or partial) extension functions. Most of the common genericity concepts in computability and complexity theory in the literature – as well as the finite-state genericity concepts introduced in this thesis – can be described this way by appropriately choosing the class $\mathcal{F}$. We will distinguish some special types of genericity notions – namely weak genericity based on total extension functions and bounded genericity based on bounded extension functions – and we will point out some limitations of these restricted concepts.

3.1 Baire Category and the Cantor Space

Using the concept of classical Baire category we can classify the subclasses of the Cantor space Σ^ω according to their size. In order to introduce this concept, we first have to define an appropriate topology on the Cantor space.

Definition 3.1 (i) For any string x, the class $\mathrm{B}_x = \{A : x \sqsubset \chi(A)\}$ is *basic open*.

(ii) A class C is *open* if it is the union of basic open classes or empty.

It is easy to see that this defines a topology on POWER(Σ^ω), i. e., $\emptyset$ and Σ^ω are open, the union of open classes is open again, and the finite intersection of open classes is open again. The latter follows from the fact that, for any strings x and y, $\mathrm{B}_x \cap \mathrm{B}_y$ is either empty (namely if x and y are incomparable) or a basic open class again (namely $\mathrm{B}_x \cap \mathrm{B}_y = \mathrm{B}_x$ if $y \sqsubseteq x$ and $\mathrm{B}_x \cap \mathrm{B}_y = \mathrm{B}_y$ if $x \sqsubseteq y$).

Definition 3.2 (Baire Category) (i) A class C is *dense* if it intersects all open classes.

(ii) A class C is *nowhere dense* if C is contained in the complement of an open and dense class.

(iii) A class C is *meager* if C is the countable union of nowhere dense classes.

(iv) A class C is *comeager* if C is the complement of a meager class.

Intuitively, we can interpret meager classes as small and comeager classes as large. For more details see Odifreddi (1989).

The following observations are easy consequences of Definition 3.2.

Proposition 3.3 *A class* C *is comeager if and only if there are countably many open and dense classes* C_n, $n \geq 0$, *such that*

$$\bigcap_{n\geq 0} \mathrm{C}_n \subseteq \mathrm{C}.$$

Proposition 3.4 *(i) For any set A, the singleton $\{A\}$ is nowhere dense.*

(ii) The countable union of meager classes is meager.

(iii) Any subclass of a meager class is meager.

(iv) Any countable class is meager.

Proposition 3.5 *(i) The countable intersection of comeager classes is comeager.*

(ii) Any superclass of a comeager class is comeager.

(iii) Any class C *with countable complement is comeager.*

In particular, Σ^ω is comeager. The non-triviality of the Baire category concept, i.e., the fact that there is no class which is both meager and comeager follows from Baire's Theorem.

Theorem 3.6 (Baire) Σ^ω *is not meager.*

Corollary 3.7 *If* C *is comeager then* C *is not meager.*

PROOF. For a contradiction assume that C is comeager and meager. Then, by the former, $\overline{\mathrm{C}}$ is meager, hence Σ^ω is the union of the meager classes C and $\overline{\mathrm{C}}$. So, by Proposition 3.4(ii), Σ^ω is meager contrary to Baire's Theorem. □

3.2 Extension Functions

The Baire category concept has been alternatively described in terms of (total) extension functions. This characterization shows the close relation between this topological concept and one of the most fundamental diagonalization techniques in computability theory, namely the finite-extension method. Similarly, a somewhat more sophisticated diagonalization method, namely the so-called wait-and-see or slow-diagonalization technique, can be linked to Baire category by using partial extension functions. In this section we review the characterizations of the comeager classes in terms of total and partial extension functions. Then, after some remarks on relations between category and measure in the next section, we discuss the relations to the above mentioned diagonalization techniques.

We first consider the case of total extension functions and define the required concepts.

Definition 3.8 (i) A *total extension function* f is a total function $f : \Sigma^* \to \Sigma^*$.

(ii) A set A *meets* f at n if $(\alpha \upharpoonright n) f(\alpha \upharpoonright n) \sqsubset \alpha$, where α is the characteristic sequence of A. A *meets* f if A meets f at some n.

Intuitively, an extension function f may be viewed as an instruction for finitely extending a given finite initial segment of an infinite sequence under construction. Then A meets f at n if the initial segment $\chi(A) \upharpoonright n$ of length n of the characteristic sequence of A is extended according to this instruction. The following theorem gives the characterization of open and dense classes in terms of extension functions.

Theorem 3.9 *For a class* $\mathrm{C} \subseteq \Sigma^\omega$ *the following are equivalent.*

(i) C *contains an open and dense class.*

(ii) $\forall x\, \exists y \sqsupseteq x\, (\mathrm{B}_y \subseteq \mathrm{C})$

(iii) There is a total extension function f *such that* $\{A : A \text{ meets } f\}$ *is contained in* C.

PROOF. We will prove the implications $(i) \Rightarrow (ii) \Rightarrow (iii) \Rightarrow (i)$.

For a proof of the implication $(i) \Rightarrow (ii)$ assume that the open and dense class C' is contained in C. Then, given any string x, it suffices to show that there is an extension y of x such that B_y is contained in C'. By density of C', the intersection of C' and B_x is not empty. Hence we may fix an infinite sequence α in C' which extends x. Since C' is open, it follows that there is a neighbourhood of α which is completely contained in C', i.e., $\mathrm{B}_{\alpha \upharpoonright n} \subseteq \mathrm{C}'$ for some number $n \geq 0$. Since, for

$n' \geq n$, $B_{\alpha \restriction n'} \subseteq B_{\alpha \restriction n}$, w.l.o.g. we may assume that $n \geq |x|$, i.e., that $x \sqsubseteq \alpha \restriction n$. So, $y = \alpha \restriction n$ has the required properties.

For a proof of $(ii) \Rightarrow (iii)$ assume that C satisfies (ii). We have to define a total extension function f such that (the characteristic sequence of) any set A which meets f is a member of C, i.e., such that

$$\forall \alpha \, (\exists n((\alpha \restriction n) f(\alpha \restriction n) \sqsubset \alpha) \Rightarrow \alpha \in \mathrm{C}) \tag{3.1}$$

holds. Define f as follows. Given x, by assumption (ii), fix the least string y extending x such that $B_y \subseteq C$ and let $f(x) = z$ for the unique string z such that $y = xz$. Then, for any string x, $B_{xf(x)} \subseteq C$. Obviously this implies that (3.1) holds.

The remaining implication $(iii) \Rightarrow (i)$ is an immediate consequence of the following somewhat more general lemma by considering the case $n_0 = 0$. □

Lemma 3.10 *For any total extension function f and any number n_0, the class $\{A : A$ meets f at some number $n \geq n_0\}$ is open and dense.*

PROOF. Fix a total extension function f and a number n_0, and let $D = \{A : A$ meets f at some number $n \geq n_0\}$. Note that, by our identification of a set with its characteristic function, the class D can be restated as

$$\mathrm{D} = \{\alpha : \exists n \geq n_0((\alpha \restriction n) f(\alpha \restriction n) \sqsubset \alpha)\}.$$

Now to show that D is open, fix $\alpha \in D$. It suffices to show that, for some number m, $B_{\alpha \restriction m}$ is contained in D. By $\alpha \in D$ we may fix $n \geq n_0$ such that $(\alpha \restriction n) f(\alpha \restriction n) \sqsubset \alpha$ holds. Then $m = n + |f(\alpha \restriction n)|$ has the required properties. Namely, for any $\beta \in B_{\alpha \restriction m}$, the set corresponding to β meets f at n. It remains to show that D is dense, i.e., that for any string x there is a sequence $\alpha \in D$ extending x, where w.l.o.g. we may assume that $|x| \geq n_0$. Obviously, the sequence $\alpha = xf(x)0^\omega$ will do.

□

Theorem 3.9 and Lemma 3.10 yield the following characterization of comeager classes.

Corollary 3.11 *The following are equivalent.*

(i) C *is comeager.*

(ii) There is a countable class $\mathcal{F} = \{f_n : n \in \mathbb{N}\}$ of total extension functions such that the class

$$\mathrm{M}_{\mathcal{F}} = \{A : \forall n \in \mathbb{N} \, (A \text{ meets } f_n)\}$$

is contained in C.

(iii) *There is a countable class* $\mathcal{F} = \{f_n : n \in \mathbb{N}\}$ *of total extension functions such that the class*

$$\mathrm{M}^{\infty}_{\mathcal{F}} = \{A : \forall n \in \mathbb{N} \text{ (A meets } f_n \text{ infinitely often)}\}$$

is contained in C*.*

PROOF. We prove the implications $(i) \Rightarrow (ii) \Rightarrow (iii) \Rightarrow (i)$

$(i) \Rightarrow (ii)$. Let C be comeager. By definition, there is a countable family of open dense classes C_n $(n \geq 0)$ such that the intersection of these classes is contained in C. So, given $n \geq 0$, it suffices to show that there is an extension function f_n such that the class $\mathrm{M}_{f_n} = \{A : A \text{ meets } f_n\}$ is contained in C_n. But this is immediate by Theorem 3.9.

$(ii) \Rightarrow (iii)$ is immediate, since for any $\mathcal{F}$, $\mathrm{M}^{\infty}_{\mathcal{F}} \subseteq \mathrm{M}_{\mathcal{F}}$.

$(iii) \Rightarrow (i)$. Note that $\mathrm{M}^{\infty}_{\mathcal{F}}$ is the intersection of the countably many classes $\mathrm{D}_{n,m} = \{A : A \text{ meets } f_n \text{ at some number} \geq m\}$. By Lemma 3.10, the classes $\mathrm{D}_{n,m}$ are open and dense. So $\mathrm{M}^{\infty}_{\mathcal{F}}$ and any superclass C of $\mathrm{M}^{\infty}_{\mathcal{F}}$ is comeagar.
This completes the proof.

□

We now turn to partial extension functions and give an alternative characterization of comeagerness in these terms. We start with the fundamental definitions.

Definition 3.12 (i) A *partial extension function* f is a partial function $f : \Sigma^* \to \Sigma^*$.

(ii) A partial extension function f is *dense along* a set A if $f(\chi(A) \upharpoonright n)$ is defined for infinitely many $n \in \mathbb{N}$.

(iii) *A meets* f at n if $f(\alpha \upharpoonright n)\downarrow$ and $(\alpha \upharpoonright n) f(\alpha \upharpoonright n) \sqsubset \alpha$ where α is the characteristic sequence of A. *A meets* f if A meets f at some n.

We use the above notions for infinite sequences in place of sets too. I.e., if α is the characteristic sequence of A and f is dense along A or A meets f then we also say that f is dense along α and α meets f, respectively.

Again, intuitively, a partial extension function f may be viewed as an instruction for finitely extending a given finite initial segment of an infinite sequence under construction. As before, A meets f at n if the initial segment $\chi(A) \upharpoonright n$ of length n of the characteristic sequence of A is extended according to this instruction. Now, however, the instruction can be followed only for certain initial segments. Density of A along f expresses that there are infinitely many chances for following the instruction.

Note that a total extension function f is dense along all sets. As we will show next, for any partial extension function f, the class of sets A such that f is not dense along A or A meets f contains an open dense class. This follows from the next lemma by letting $n_0 = 0$.

Lemma 3.13 *Let f be any partial extension function, let n_0 be any number, and let* D *be the class $\{A : f$ is not dense along A or A meets f at some number $n \geq n_0\}$. Then* D *contains an open and dense subclass.*

PROOF. Given a string x it suffices to show that there is an extension y of x such that the basic open class B_y is contained in D. To show this we first observe that, by our identification of a set with its characteristic function, the class D can be restated as

$$\mathrm{D} = \{\alpha : \exists^\infty n (f(\alpha \upharpoonright n)\downarrow) \Rightarrow \exists n \geq n_0 (f(\alpha \upharpoonright n)\downarrow \ \& \ (\alpha \upharpoonright n) f(\alpha \upharpoonright n) \sqsubset \alpha)\}.$$

Now, in order to get the desired extension y of x distinguish the following two cases. First assume that there is an extension $\hat{x}$ of x such that $|\hat{x}| \geq n_0$ and $f(\hat{x})$ is defined. Then, for the extension $y = \hat{x} f(\hat{x})$ of x, B_y is contained D, since any $\alpha \in \mathrm{B}_y$ meets f at $|\hat{x}|$. Now assume that there is no extension $\hat{x}$ of x as above. Then, for any extension y of x of length $\geq n_0$, $f(z)$ is undefined for all extension z of y. So, for any $\alpha \in \mathrm{B}_y$, f is not dense along α. Hence B_y is contained in D. □

The preceding lemma implies the following analog of Theorem 3.9 for partial extension functions.

Theorem 3.14 *For a class $\mathrm{C} \subseteq \Sigma^\omega$ the following are equivalent:*

(i) C *contains an open and dense class.*

(ii) $\forall x \, \exists y \sqsupseteq x \, (\mathrm{B}_y \subseteq \mathrm{C})$

(iii) There is a partial extension function f such that

$$\{A : f \text{ is not dense along } A \text{ or } A \text{ meets } f\}$$

is contained in C*.*

PROOF. Note that the implication $(i) \Rightarrow (ii)$ is immediate by Theorem 3.9 and the implication $(ii) \Rightarrow (iii)$ follows from the corresponding implication in Theorem 3.9 since a total extension function f is dense along any set, i.e., for total f the classes $\{A : A$ meets $f\}$ and $\{A : f$ is not dense along A or A meets $f\}$ coincide. So it only remains to prove the implication $(iii) \Rightarrow (i)$. But this is an immediate consequence of Lemma 3.13. □

Now, by applying Theorem 3.14 and Lemma 3.13 in place of Theorem 3.9 and Lemma 3.10, respectively, in analogy to Corollary 3.11 we obtain the following characterization of comeagerness in terms of partial extension functions.

Corollary 3.15 *The following are equivalent:*

(i) C *is comeager.*

(ii) *There is a countable class* $\mathcal{F} = \{f_n : n \in \mathbb{N}\}$ *of partial extension functions such that the class*

$$\mathrm{M}_{\mathcal{F}} = \{A : \forall n \in \mathbb{N}\,(f_n \text{ is not dense along } A \text{ or } A \text{ meets } f_n)\}$$

is contained in C*.*

(iii) *There is a countable class* $\mathcal{F} = \{f_n : n \in \mathbb{N}\}$ *of partial extension functions such that the class*

$$\mathrm{M}^{\infty}_{\mathcal{F}} = \{A : \forall n \in \mathbb{N}\,(f_n \text{ is not dense along } A \text{ or } A \text{ meets } f_n \text{ infinitely often})\}$$

is contained in C*.*

PROOF. This follows from Theorem 3.14 and Lemma 3.13 just as Corollary 3.11 follows from Theorem 3.9 and Lemma 3.10, respectively. □

3.3 Baire Category and Lebesgue Measure

An alternative to the Baire category concept for classifying the subclasses of the Cantor space Σ^ω according to their size is Lebesgue measure. Here the measure-0 classes are the small classes and the measure-1 classes are large. I.e. meagerness (comeagerness) in the setting of category corresponds to measure 0 (measure 1) in the setting of measure. These two classifications, however, are incompatible in general. There are classes which are large in one setting but small in the other. To be more precise, there are classes C such that C is comeager and has measure 0 (hence $\overline{\mathrm{C}}$ is meager and has measure 1). We can use the description of Baire category in terms of (partial) extension functions, however, to give a sufficient condition for a class C to be both comeager and of measure 1.

In the following we will shortly develop the basic concepts of Lebesgue measure which we will need later.

The Lebesgue measure μ on the Cantor space is the product measure induced by the equiprobable measure on the finite space $\{0,1\}$ which assigns to both 0 and 1 the probability 2^{-1}. So, in particular, for a basic open class B_x,

$$\mu(\mathrm{B}_x) = 2^{-|x|}.$$

In order to obtain the notion of a measure-0 class we have to consider coverings by basic open sets.

Definition 3.16 (i) Let C be a class, let $\rho > 0$ be a real number, and let $\mathcal{B} = \{B_{x_n} : n \geq 0\}$ be a countable sequence of basic open classes. $\mathcal{B}$ is an ρ *-cover* of C if

$$\mathrm{C} \subseteq \bigcup_{n\geq 0} \mathrm{B}_{x_n} \ \ \& \ \ \sum_{n\geq 0} 2^{-|x_n|} < \rho.$$

(ii) A class C *has (Lebesgue) measure* 0, $\mu(\mathrm{C}) = 0$, if for all $n \geq 0$ there is a 2^{-n}-cover of C.

(iii) A class C *has (Lebesgue) measure* 1, $\mu(\mathrm{C}) = 1$, if the complement of C has measure 0, i.e., if $\mu(\overline{\mathrm{C}}) = 0$.

For measure-0 and measure-1 classes we get the following properties corresponding to the properties of meager and comeager classes given in Propositions 3.4 and 3.5.

Proposition 3.17 *(i) Any countable class has measure 0.*

(ii) The countable union of measure-0 classes has measure 0.

(iii) Any subclass of a measure-0 class has measure 0.

(iv) Any co-countable class has measure 1.

(v) The countable intersection of measure-1 classes has measure 1.

(vi) Any superclass of a measure-1 class has measure 1.

In particular, $\mu(\Sigma^\omega) = 1$. Moreover, the measure concept is nontrivial, i.e. a measure-1 class does not have measure 0.

Proposition 3.18 Σ^ω *is not a measure-0 class. More generally, no measure-1 class has measure 0.*

As mentioned before, in general Baire category and Lebesgue measure are not compatible. We demonstrate this by giving an example of a comeager class C with $\mu(\mathrm{C}) = 0$.

Lemma 3.19 *Let* C *be the class*

$$\mathrm{C} = \{A : \exists^\infty n\ (A \cap \Sigma^n = \emptyset)\}.$$

Then C *is comeager and* $\mu(\mathrm{C}) = 0$.

PROOF. To show that C is comeager, by Corollary 3.11, it suffices to observe that any set A which infinitely often meets the extension function f defined by $f(x) = 0^{3|x|}$ is a member of C. To show that $\mu(\mathrm{C}) = 0$, we first observe that, for any $n \geq 1$, the class $\mathrm{C}_n = \{A : A \cap \Sigma^n = \emptyset\}$ is covered by the finite 2^{-n}-cover $\mathcal{B}_n = \{\mathrm{B}_{x0^{2^n}} : |x| = 2^n - 1\}$. It follows that there is a 2^{-n}-cover of $\hat{\mathrm{C}}_n = \bigcup_{m>n} C_m$. Since C is contained in $\hat{\mathrm{C}}_n$ for all $n \geq 0$ it follows that $\mu(\mathrm{C}) = 0$. □

We obtain a sufficient condition for a class to be both comeager and of measure 1 by considering bounded extension functions.

Definition 3.20 A (partial) extension function f is *k-bounded* if $|f(w)| = k$ whenever $f(w)$ is defined, i.e., if f is a (partial) function $f : \Sigma^* \to \Sigma^k$. f is *bounded* if f is k-bounded for some k.

(Sometimes we will also call a function $f : \Sigma^* \to \Sigma^{\leq k}$ k-bounded. Then, formally, f should be considered to be the function $\hat{f} : \Sigma^* \to \Sigma^k$ which is obtained from f by extending any value $f(x)$ to a string of length k by adding zeroes at the end. I.e. f and $\hat{f}$ have the same domain and if $f(x)$ is defined then $\hat{f}(x) = f(x)0^{k-|f(x)|}$.)

Theorem 3.21 *(a) Let $\mathcal{F} = \{f_n : n \geq 0\}$ be a countable class of bounded total extension functions. Then*

$$\mathrm{M}_{\mathcal{F}} = \{A : \forall n \in \mathbb{N}\ (A \text{ meets } f_n)\}$$

and

$$\mathrm{M}^{\infty}_{\mathcal{F}} = \{A : \forall n \in \mathbb{N}\ (A \text{ meets } f_n \text{ infinitely often})\}$$

are comeager and have measure 1.

(b) Let $\mathcal{F} = \{f_n : n \geq 0\}$ be a countable class of bounded partial extension functions. Then

$$\mathrm{M}_{\mathcal{F}} = \{A : \forall n \in \mathbb{N}\ (f_n \text{ is not dense along } A \text{ or } A \text{ meets } f_n)\}$$

and

$$\mathrm{M}^{\infty}_{\mathcal{F}} = \{A : \forall n \in \mathbb{N}\ (f_n \text{ is not dense along } A \text{ or } A \text{ meets } f_n \text{ infinitely often})\}$$

are comeager and have measure 1.

PROOF. Since any total extension function is dense along any set, part (a) is a special case of part (b). So it suffices to prove (b). Moreover, the claim about comeagerness has been shown in Corollary 3.15 already. So it suffices to show that $\mu(\mathrm{M}_{\mathcal{F}}) = 1$ and $\mu(\mathrm{M}^{\infty}_{\mathcal{F}}) = 1$. In fact, since $\mathrm{M}^{\infty}_{\mathcal{F}}$ is contained in $\mathrm{M}_{\mathcal{F}}$, it suffices to prove the latter. Since the countable intersection of measure-1 classes is a measure-1 class again this task can be reduced to showing that, for a given bounded extension function f, the class

$$\mathrm{M}^{\infty}_{f} = \{A : f \text{ is not dense along } A \text{ or } A \text{ meets } f \text{ infinitely often}\}$$

has measure 1 or, equivalently, the complement $\overline{\mathrm{M}^{\infty}_{f}}$ of M_f has measure 0.

Note that, for

$$\mathrm{D} = \{A : \exists^{\infty} n\ (f(\chi(A) \restriction n) \downarrow)\}$$

and

$$\mathrm{N}_m = \{A : \forall n \geq m\ (f(\chi(A) \restriction m) \downarrow \Rightarrow (\chi(A) \restriction m) f(\chi(A) \restriction m) \not\sqsubseteq \chi(A))\},$$

$$\overline{\mathrm{M}^{\infty}_{f}} = \mathrm{D} \cap (\bigcup_{m \geq 0} \mathrm{N}_m).$$

So in order to show that $\mu(\overline{\mathrm{M}^{\infty}_{f}}) = 0$ it suffices to show that, for any given $m \geq 0$, $\mu(\mathrm{D} \cap \mathrm{N}_m) = 0$. In other words, given $p \geq 1$, we have to show that there is a 2^{-p}-cover of $\mathrm{D} \cap \mathrm{N}_m$. So fix m and p and fix k such that f is k-bounded. Moreover, for $q \geq 1$ let $\mathrm{C}_{m,q}$ be the class consisting of all sets A such that there are at least q

numbers $n > m$ such that $f(A \upharpoonright n)$ is defined and A does not meet f at the first q such numbers n. Then $\mathrm{D} \cap \mathrm{N}_m$ is contained in $\mathrm{C}_{m,q}$ for all $q \geq 1$. Hence it suffices to find a 2^{-p}-cover $\mathcal{B}$ of $\mathrm{C}_{m,q}$ for some $q \geq 1$. Note that given any string x such that $f(x)$ is defined, there is one extension y of x of length $|x| + k$, namely $y = xf(x)$, such that all sets A in B_y meet f at $|x|$. Since there are 2^k extensions of x of length $|x| + k$ this implies that there is a $(2^k - 1)2^{-k}$-cover of $\mathrm{C}_{m,1}$. So, by induction, we can argue that there is an ρ-cover of $\mathrm{C}_{m,q}$ where $\rho = [(2^k - 1)2^{-k}]^q$. Since

$$lim_{q \to \infty}[(2^k - 1)2^{-k}]^q = 0$$

it follows that, for sufficiently large q we obtain the desired 2^{-p}-cover $\mathcal{B}$ of $\mathrm{C}_{m,q}$.
□

3.4 Finite Extension Arguments

We now will use the finite-extension-function characterization of Baire category in order to demonstrate the relations between this concept and the finite-extension method. Most of the diagonalization techniques in computability theory and computational complexity have been obtained by refining the finite-extension method. The finite-extension method itself is an extension of Cantor's diagonalization technique, i.e., of a direct diagonalization. Roughly speaking, in a direct diagonalization argument we define a set A which is not a member of a countable class C as follows: we fix an enumeration $\{C_n : n \geq 0\}$ of C and ensure that A differs from the nth set C_n in this list by making A and C_n differ on the nth string, i.e., by letting $A(z_n) = 1 - C_n(z_n)$. More generally we can say that a set A will not be a member of C if it meets the *requirements*

$$\mathcal{R}_n : \exists x \, (A(x) \neq C_n(x))$$

for all numbers $n \geq 0$. I.e., the global, infinitary task of ensuring that A is not a member of C is split into an infinite sequence of finitary requirements. Here the requirements are of a particularly simple form, namely in order to meet a single requirement it suffices to appropriately define A on any given string.

In a finite-extension argument we also decompose a global task into infinitely many finitary requirements but the requirements are of a more general nature. Here, in order to meet a single requirement $\mathcal{R}$, given any finite initial segment $\alpha \restriction n$ of the characteristic sequence α of the set A under construction, there will be a possible finite extension $\alpha \restriction m$, $m \geq n$, of $\alpha \restriction n$ such that this extension will ensure that the requirement is met no matter how we will define A on the remaining inputs. The desired set A is inductively defined in stages $s \geq 0$ by specifying longer and longer initial segments $\alpha \restriction l(s)$, where $l(s) < l(s+1)$ and where at stage s, given the part $\alpha \restriction l(s-1)$ of A defined at the previous stages, the string $l(s) > l(s-1)$ and the extension $\alpha \restriction l(s)$ of $\alpha \restriction l(s-1)$ is chosen so that the sth requirement will be met by this extension. So, in contrast to a direct diagonalization, for meeting a single requirement in general we have to appropriately fix A not only on a single string but on a finite number of strings, and the way we can meet the requirement may depend on the finite part of A previously specified.

We explain this method by giving an example from computability theory. A set which is many-one reducible to its complement is called *self-dual*. So, given a list $\{f_n : n \geq 0\}$ of the total recursive functions, a non-self-dual set A has to meet the requirements

$$\mathcal{R}_n : \exists x \, (A(x) = A(f_n(x)))$$

for all numbers $n \geq 0$. In order to meet $\mathcal{R}_n$, given a string x on which A is not yet defined we have to make sure that $A(x)$ and $A(f_n(x))$ agree. There are two cases: If A has been defined on $f_n(x)$ before then it suffices to let $A(x)$ have the value $A(f_n(x))$. If A has not yet been defined on $f_n(x)$ then we have to fix A on x and $f_n(x)$. (So the action for meeting the requirement may depend on the previous action or may require to determine A on more than one string.) Now, a finite-extension construction of (the characteristic sequence α of) a non-self-dual set is as follows. Given the finite initial segment $\alpha \upharpoonright l(s-1)$ of α specified prior to stage s (where $l(-1) = 0$, i.e., $\alpha \upharpoonright l(-1) = \varepsilon$) we define $l(s)$ and $\alpha \upharpoonright l(s)$ as follows. Let $x = l(s-1)$. If $f_s(x) < x$ then let $l(s) = l(s-1)+1$ and let $\alpha \upharpoonright l(s) = (\alpha \upharpoonright l(s-1))\alpha(f_s(x))$. Otherwise, let $l(s) = max(x, f_s(x))+1$ and set $\alpha \upharpoonright l(s) = (\alpha \upharpoonright l(s-1))0^{l(s)-l(s-1)}$ thereby ensuring that $\alpha(x) = \alpha(f_s(x)) = 0$. Then, in either case, the extension $\alpha \upharpoonright l(s)$ of $\alpha \upharpoonright l(s-1)$ guarantees that $\mathcal{R}_s$ is met.

Now, in order to relate the finite extension method to Baire category, based on the above intuitive remarks and observations we first formally define what we mean by saying that a property can be ensured by the finite-extension method.

Definition 3.22 A property $\mathcal{P}$ can be *ensured by a finite-extension argument* if there is a sequence $\{\mathcal{R}_n : n \geq 0\}$ of finitary requirements such that any set A which meets all requirements $\mathcal{R}_n$ has property $\mathcal{P}$. Here a requirement $\mathcal{R}$ is *finitary*, if for any string x there is a string y extending x such that any set A with $y \sqsubset \chi(A)$ meets $\mathcal{R}$. For a string y such that all sets A with $y \sqsubset \chi(A)$ meet $\mathcal{R}$ we say that y *forces* $\mathcal{R}$.

Note that the formal definition of a finitary requirement $\mathcal{R}$ reflects the fact, that having specified a finite initial segment of a set A under construction we can finitely extend this initial segment in such a way that this extension will ensure that A will meet $\mathcal{R}$ no matter how we will define A on all larger inputs not specified by this extension. Next we will observe that, for any finitary requirement $\mathcal{R}$, we can attach an extension function f to $\mathcal{R}$ such that a set A meets $\mathcal{R}$ if A meets f. In this case we say that the extension function f corresponds to the requirement $\mathcal{R}$ or that f is a strategy for meeting $\mathcal{R}$.

Definition 3.23 Let $\mathcal{R}$ be a finitary requirement and let f be a total extension function. Then f *corresponds* to $\mathcal{R}$ or f is a *strategy* for $\mathcal{R}$ if, for any set A which meets f, A meets $\mathcal{R}$.

Lemma 3.24 *Let $\mathcal{R}$ be a finitary requirement. There is an extension function f corresponding to $\mathcal{R}$.*

PROOF. We obtain the required extension function f by letting $f(x)$ be the least string z such that xz forces $\mathcal{R}$. □

Note that for a finitary requirement $\mathcal{R}$ there may be many strategies for $\mathcal{R}$ not just one. In a finite-extension construction of a set A we meet the individual requirements by using some given strategies. Formally we can express this as follows.

Definition 3.25 A *finite-extension construction* $\mathcal{C}$ of a set A with property $\mathcal{P}$ is given by a sequence of finitary requirements $\{\mathcal{R}_n : n \geq 0\}$ ensuring $\mathcal{P}$ together with a sequence of corresponding extension functions $\{f_n : n \geq 0\}$. The set A defined by $\mathcal{C}$ is inductively defined by $\chi(A) \upharpoonright l(-1) = \chi(A) \upharpoonright 0 = \varepsilon$ and $\chi(A) \upharpoonright l(n) = (\chi(A) \upharpoonright l(n-1)) f_n(\chi(A) \upharpoonright l(n-1))$ for $n \geq 0$.

Lemma 3.24 gives the desired relation between the finite-extension method and Baire category by using the characterization of the latter in terms of extension functions.

Theorem 3.26 *Let $\mathcal{P}$ be a property and let $\mathrm{C}_{\mathcal{P}}$ be the class of sets with property $\mathcal{P}$. Then the following are equivalent.*

(i) $\mathcal{P}$ can be ensured by a finite-extension argument.

(ii) $\mathrm{C}_{\mathcal{P}}$ is comeager.

PROOF. First assume that $\mathcal{P}$ can be ensured by a finite-extension argument and let $\{\mathcal{R}_n : n \geq 0\}$ be a sequence of finitary requirements ensuring $\mathcal{P}$. By Lemma 3.24 fix a sequence of extension functions, $\mathcal{F} = \{f_n : n \geq 0\}$, corresponding to $\{\mathcal{R}_n : n \geq 0\}$ and let $\mathrm{M}_{\mathcal{F}}$ be the class of languages A which meet all extension functions f_n ($n \geq 0$). Then, by choice of the functions f_n, any set A in $\mathrm{M}_{\mathcal{F}}$ has property $\mathcal{P}$ and, by Corollary 3.11, $\mathrm{M}_{\mathcal{F}}$ is comeager.

For a proof of the other direction assume that $\mathcal{P}$ is a property such that $\mathrm{C}_{\mathcal{P}}$ is comeager. By Corollary 3.11 fix a sequence of extension functions, $\mathcal{F} = \{f_n : n \geq 0\}$, such that $\mathrm{M}_{\mathcal{F}}$ is contained in $\mathrm{C}_{\mathcal{P}}$. Define requirements $\mathcal{R}_n$ ($n \geq 0$) by

$$\mathcal{R}_n : A \text{ meets } f_n.$$

Obviously, $\mathcal{R}_n$ is finitary and f_n corresponds to $\mathcal{R}_n$. Moreover, the requirements $\{\mathcal{R}_n : n \geq 0\}$ imply $\mathcal{P}$. So $\mathcal{P}$ can be ensured by a finite-extension argument.

□

The above close relation between the finite-extension method and the Baire category concept sheds some more light on the finite-extension method. For instance, since the countable intersection of comeager classes is comeager again, any two - in fact any countably many - finite extension constructions can be combined. I.e. if $\mathcal{P}_n$ ($n \geq 0$) are properties which can be ensured by finite-extension constructions then, by such a construction, we can construct a single set A which has all of these properties. On the other hand, the finite-extension approach can give us some insight on the complexity of the members of a comeager class. Since any countable class is meager and any complexity class is countable, the fact that the class of sets with a certain property $\mathcal{P}$ is comeager does not tell us whether there is a set of a given complexity with this property. Since in a finite extension-construction the complexity of the constructed set A is explicitly determined by the complexity of the strategies used for meeting the individual finitary requirements (see Definition 3.25), an analysis of the complexity of the required strategies will yield complexity results along these lines.

We now turn to an important refinement of the finite-extension-method, the so-called *wait-and-see* arguments or *slow diagonalizations* which can be linked to Baire category too, now by using the description of comeager classes in terms of partial extension functions.

We first describe this technique by giving a simple example. Recall that a set A is bi-immune to a countable class C if neither A nor its complement contains an infinite member of C as a subset. Given an enumeration $\{C_n : n \geq 0\}$ of C we can define a sequence of requirements, $\{\mathcal{R}_n : n \geq 0\}$, ensuring C-bi-immunity as follows.

$$\mathcal{R}_{2n} : |C_n| = \infty \Rightarrow \exists x \in C_n \ (x \notin A)$$

$$\mathcal{R}_{2n+1} : |C_n| = \infty \Rightarrow \exists x \in C_n \ (x \in A)$$

(Note that the requirements with even index ensure that no infinite member of C is a subset of A and the requirements with odd index ensure the corresponding fact for $\overline{A}$.) Now, these requirements are finitary. For instance, we get an extension function f_{2n} corresponding to $\mathcal{R}_{2n}$ as follows. If C_n is finite then requirement $\mathcal{R}_{2n}$ is vacuously met and we can let f_{2n} be any extension function, e.g., $f_{2n}(x) = 0$ for all x. If C_n is infinite, then f_{2n} can be defined as follows. Given x, let $p = |x|$ and, by infinity of C_n choose $q > p$ minimal such that $z_q \in C_n$. Then $f_{2n}(x) = 0^{q+1-p}$ will do since this ensures that if a set A meets f_{2n} at p then $A(z_q) = 0$, i.e., $z_q \in C_n \setminus A$.

The above shows that, for any countable class C, C-bi-immunity can be ensured by a finite-extension argument. Even, for uniformly computable classes C, however, the above argument may not yield a recursive set. This follows from the fact, that in general the infinity problem for such classes can be undecidable. Moreover the above finite-extension argument is not bounded since in general a class C will have an infinite member C_n with unbounded gaps (i.e. there will be blocks 0^m for any $m \geq 0$ in the characteristic sequence of C_n). (This may be considered unsatisfying since for meeting a requirement $\mathcal{R}_{2n}$, just as in a simple Cantor diagonalization, it suffices to appropriately fix $A(x)$ on a single string x and the value to be given to $A(x)$ does not depend on the other values of A. The difference to a Cantor diagonalization is that we cannot choose any string x for diagonalization but we are limited to the infinitely many strings x in C_n.)

These shortcomings of a finite-extension construction of a C-bi-immune set are overcome in a slow diagonalization of such a set. Here the requirements are as before but the strategy for meeting the requirements is a different one. While in a finite-extension argument the requirements are met in order, in fact at stage s of the construction the sth requirement is met, here the order in which the requirements are met dynamically depends on the construction. Now, at stage s of the construction of A, we determine A on the sth string $z_s = s$. I.e., we extend the previously given initial segment $\alpha \upharpoonright s$ of the characteristic sequence α of A by one bit. This extension is chosen in such a way that we will meet the requirement $\mathcal{R}_{2n+i}$ with the least index $2n+i \leq s$ which has not been met before and which can be met now. (Note that the latter will be the case if and only if $C_n(s) = 1$.)

This idea is made more precise by introducing the following notions. We say that a requirement $\mathcal{R}_{2n+i}$ is *satisfied at (the end of) stage s* if there is a string $x \leq s$ such that $x \in C_n$ and $A(x) = i$ and we say that $\mathcal{R}_{2n+i}$ *requires attention at stage s* if $2n+i \leq s$, $\mathcal{R}_{2n+i}$ is not satisfied at stage $s-1$, and $s \in C_n$.

Then, at stage s of the construction, given $\alpha \upharpoonright s$ we fix $2n+i$ minimal such that the requirement $\mathcal{R}_{2n+i}$ requires attention (if there is any), let $A(s) = i$, and say that $\mathcal{R}_{2n+i}$ *receives attention* or is *active* at stage s.

Now to show that the constructed set A meets all requirements, hence is C-bi-immune, we start with two easy observations. First, note that a requirement $\mathcal{R}_{2n+i}$ which is active at stage s is satisfied at stage s. Second, if $\mathcal{R}_{2n+i}$ is satisfied at some stage s, then $\mathcal{R}_{2n+i}$ is satisfied at all later stages, hence will not require or receive attention at any stage $> s$, and $\mathcal{R}_{2n+i}$ is actually met. In particular it follows that any requirement becomes active at most once. Now, for a contradiction, assume that the requirement $\mathcal{R}_{2n+i}$ is not met. Then, by the above observations, there is no stage s such that $\mathcal{R}_{2n+i}$ is satisfied at stage s or such that $\mathcal{R}_{2n+i}$ becomes active at stage s. On the other hand, there are infinitely many stages s with $s \in C_n$ since otherwise $\mathcal{R}_{2n+i}$ trivially holds. Since any requirement becomes active at most once, by the latter we may choose $s > 2n+i$ such that $s \in C_n$ and no requirement R_m with $m < 2n+i$ becomes active at stage s. It follows that $\mathcal{R}_{2n+i}$ requires and receives attention at stage s. So, by the above, $\mathcal{R}_{2n+i}$ is satisfied at stage s hence met contrary to assumption.

Note that the above construction of A is recursive in $\{C_n : n \geq 0\}$. So, for any given uniformly recursive class C the construction yields a recursive C-bi-immune set. In particular, this implies Theorem 2.66.

In the above construction of a C-bi-immune set the strategy for meeting a single requirement can be described by a partial extension function. To be more precise, the partial extension function f_{2n+i} defined by $f_{2n+i}(x) = i$ if $z_{|x|} \in C_n$ and $f_{2n+i}(x) \uparrow$

otherwise corresponds to the requirement $\mathcal{R}_{2n+i}$ as follows. If f_{2n+i} is not dense along the constructed set A then C_n is finite whence the requirement $\mathcal{R}_{2n+i}$ is trivially met. On the other hand, if A meets f_{2n+i} at some number m then $z_m \in C_n$ and $A(z_m) = i$ whence $\mathcal{R}_{2n+i}$ is met. So in order that A meets requirement $\mathcal{R}_{2n+i}$ it suffices that either f_{2n+i} is not dense along A or A meets f_{2n+i}.

The construction of a C-bi-immune differs from a general wait-and-see argument in one respect, namely, as pointed out above already, the diagonalization depends only a single input. This is reflected by the fact that the partial extension functions attached to the individual requirements are 1-bounded. In a general argument, such a constant bound on the required extension will not exist. In analogy to the formalization of finite- extension arguments in Definitions 3.22 and 3.23 above, we can formalize the wait-an-see method as follows.

Definition 3.27 A property $\mathcal{P}$ can be *ensured by a wait-and-see argument* if there is a sequence $\{\mathcal{R}_n : n \geq 0\}$ of quasi-finitary requirements such that any set A which meets all requirements $\mathcal{R}_n$ has property $\mathcal{P}$.

Here a requirement $\mathcal{R}$ is *quasi-finitary*, if there is a set $D_{\mathcal{R}}$ of strings such that (1) for any string $x \in D_{\mathcal{R}}$ there is a string y extending x such that any set A with $y \sqsubset \chi(A)$ meets $\mathcal{R}$ (we say that *y forces* $\mathcal{R}$) and (2) any set A such $D_{\mathcal{R}}$ contains at most finitely many initial segments of A meets $\mathcal{R}$ too.

Definition 3.28 Let $\mathcal{R}$ be a quasi-finitary requirement and let f be a partial extension function. Then f *corresponds* to $\mathcal{R}$ or f is a *strategy* for $\mathcal{R}$ if, for any set A such that f is not dense along A or A which meets f, A meets $\mathcal{R}$.

Again one can easily show that there is a strategy for any quasi-finitary requirement.

Lemma 3.29 *Let $\mathcal{R}$ be a quasi-finitary requirement. There is a partial extension function f corresponding to $\mathcal{R}$.*

PROOF. Fix $D_{\mathcal{R}}$ as in Definition 3.27. We obtain the required extension function f by letting $f(x)$ be defined if and only if x $D_{\mathcal{R}}$ and by letting $f(x)$ be the least string z such that xz forces $\mathcal{R}$ if $f(x)$ is defined. □

The above together with the characterization of the comeager classes in terms of partial extension functions shows that a property $\mathcal{P}$ can be ensured by a wait-and-see argument if and only if the corresponding class $C_{\mathcal{P}}$ of sets with property $\mathcal{P}$ is comeager. Together with the similar result for the finite-extension method we obtain the following equivalence theorem.

Theorem 3.30 *Let $\mathcal{P}$ be a property and let* $C_{\mathcal{P}}$ *be the class of sets with property $\mathcal{P}$. Then the following are equivalent.*

(i) *$\mathcal{P}$ can be ensured by a wait-and-see argument.*

(ii) *$\mathcal{P}$ can be ensured by a finite-extension argument.*

(iii) $C_{\mathcal{P}}$ *is comeager.*

PROOF. This easily follows from Lemma 3.29 and Corollary 3.15 together with Theorem 3.26 □

The above theorem shows that in principle the wait-and-see method is not more powerful than the finite-extension method, i.e., there is no property $\mathcal{P}$ which can be ensured by a wait-and-see argument but not by a finite-extension argument. In fact one can easily show that the notion of a finitary requirement and the notion of a quasi-finite requirement used in the formal characterizations of the finite-extend method and the wait-and-see method, respectively, coincide.

The advantage of the wait-and-see arguments stems from the fact that this method admits simpler strategies, hence yields witnesses for a property $\mathcal{P}$ of lower complexity. We have exemplified this abov by looking at the construction of a C-bi-immune set using both, a finite extension argument and a wait-and-see argument. While the former construction was noneffective hence (in general) gave a nonrecursive set, for uniformly recursive C the latter proof was effective thereby yielding a recursive C-bi-immune set. This phenomenon has been closer analyzed in the literature. For instance, if we let C be the important class P of the polynomial-time computable sets, then Mayordomo (1994) has shown that there is no uniformly recursive set of total extension functions ensuring P-bi-immunity (whence any finite-extension construction yields only a nonrecursive P-bi-immune set) while Ambos-Spies (1996) has shown that there is a uniformly recursive class of partial extension functions of time complexity $O(n^2)$ ensuring P-bi-immunity whence a P-bi-immune set in $\mathrm{DTIME}(2^{3n})$ can be constructed by a wait-and-see argument. Moreover, as pointed out above already, the finite-extension construction of a C-bi-immune set requires unbounded extensions whereas for the wait-and-see construction 1-bounded partial extension functions suffice. So, by Theorem 3.21, the wait-and-see approach shows that (for any countable class C the class of C-bi-immune sets has measure 1, an observation we cannot make based on the finite-extension approach.

In order to analyze these complexity-issues more closely, in the next section we introduce generic sets. Intuitively speaking, this will allow us to define for any complexity level a notion of genericity and weak genericity such that the corresponding generic sets will have all properties which can be ensured by wait-and-see arguments based on strategies of the corresponding complexity level while the weakly generic sets will have all properties which can be ensured by finite-extension arguments based on strategies of the corresponding complexity level.

3.5 Generic Sets

The finite-extension method and, similarly, the wait-and-see method have been introduced as diagonalization techniques for constructing sets with a certain property $\mathcal{P}$. Starting from this property $\mathcal{P}$, first an infinite list of (quasi-)finitary requirements $\mathcal{R}_n$, $n \geq 0$, is given such that these requirements together ensure $\mathcal{P}$. Then strategies f_n (i.e., partial or total extension functions) corresponding to the requirements $\mathcal{R}_n$ are designed and, finally, these strategies are used in a canonical way for defining a set A with property $\mathcal{P}$. So the complexity of A will depend on the complexity of the strategies f_n, hence, by analyzing the complexity of these strategies, we can analyze some complexity issues of sets with property $\mathcal{P}$.

Now we proceed in the opposite direction. We start with an arbitrary countable family $\mathcal{F}$ of strategies and look at the property ensured by these strategies. We call this property $\mathcal{F}$-genericity and sets with this property $\mathcal{F}$-generic. Typically, $\mathcal{F}$ will consist of all the strategies of a certain complexity level, for example the recursive or the polynomial-time computable strategies. (Note that all complexity classes are countable, hence the corresponding family $\mathcal{F}$ of strategies will be countable too.) By the relations between the finite extension-method and Baire category, generic sets will always exist, in fact the class of $\mathcal{F}$-generic sets will be comeager.

Note that $\mathcal{F}$-genericity is the strongest property which can be ensured by families of strategies which are members of $\mathcal{F}$. So if $\mathcal{F}$ consists of all strategies of a certain complexity level, $\mathcal{F}$-genericity will be the strongest property we can obtain by strategies whose complexities do not exceed this level.

So, in order to show that strategies of a certain complexity level do not suffice for ensuring a certain property $\mathcal{P}$, it suffices to show that there is a set which is generic for this family of strategies but which does not have property $\mathcal{P}$. This shows that generic sets are of great interest for the formal analysis of the strength of resource-bounded diagonalization techniques (see Ambos-Spies (1996) for more details).

Generic sets play another important role in structural investigations in computability theory and computational complexity theory for obtaining strong separations. If there are complexity classes $C_1 \subset C_2$ such that there is a set G in C_2 which is generic for C_1 then we may deduce that all diagonalization arguments based on strategies from C_1 can be carried out inside of C_2. So, roughly speaking, the genericity concepts will combine the advantages of Baire category, namely combinability and modularity, with a way to control the complexity of the witness sets.

After these intuitive remarks pointing out some of the important aspects and applications of genericity concepts, we will now formally introduce genericity. We will distinguish between (general) genericity based on partial extension functions and *weak genericity* based on total extension functions. Moreover, we will call a genericity concept a *bounded genericity* concept if it is based on bounded (total or partial) extension functions.

Definition 3.31 Let $\mathcal{F}$ be a countable class of partial extension functions and let $\mathcal{F}[tot]$ be the class of total extension functions in $\mathcal{F}$. A set G is *$\mathcal{F}$-generic* if G meets all partial extension functions in $\mathcal{F}$ which are dense along G, i.e., if G is a member of the class

$$\mathrm{M}_{\mathcal{F}} = \{A : \forall f \in \mathcal{F}\ (f \text{ is not dense along } A \text{ or } A \text{ meets } f)\}.$$

A set G is *weakly $\mathcal{F}$-generic* if G meets all total extension functions in $\mathcal{F}$, i.e., if G is a member of the class

$$\mathrm{M}_{\mathcal{F}[tot]} = \{A : \forall f \in \mathcal{F}[tot]\ (A \text{ meets } f)\}.$$

The relations between $\mathcal{F}$-genericity and weak $\mathcal{F}$-genericity are as follows.

Proposition 3.32 *Let $\mathcal{F}$ be any countable class of partial extension functions. Then any $\mathcal{F}$-generic set is weakly $\mathcal{F}$-generic. Moreover, if $\mathcal{F}$ contains only total functions then $\mathcal{F}$-genericity and weak-$\mathcal{F}$-genericity coincide.*

PROOF. This is immediate by definition and by the fact that any total extension function is dense along all sets. □

Some examples of genericity notions which have been studied in the literature are as follows: By letting $\mathcal{F}$ be the class of all arithmetically definable partial extension functions we obtain arithmetical genericity introduced by Feferman (1965) and by considering the class of partial recursive extension functions we obtain the notion of 1-genericity introduced by Hinman (1969) which plays a major role in the degrees of unsolvability (see e.g. Jockusch (1980)). By considering the class of total recursive functions we obtain the genericity concept related to the effective Baire category concept of Mehlhorn (1973). Moreover various of the resource bounded genericity concepts introduced in complexity theory can be obtained by letting $\mathcal{F}$ be the class of (partial or total, bounded or unbounded) extension functions computable within some give time or space bounds where in some cases the representation of the input or the output has to be modified (see e.g. Ambos-Spies (1996), Ambos-Spies et al. (1987), Ambos-Spies et al. (1988), Fenner (1991), Fenner (1995), Fleischhack (1985), Fleischhack (1986), and Lutz (1990)). The finite-state genericity concepts introduced and discussed in this thesis will be obtained by considering classes $\mathcal{F}$ of different types of extension functions all of which are computable by finite automata.

We next turn to the existence of $\mathcal{F}$-generic sets.

Theorem 3.33 *Let $\mathcal{F}$ be any countable class of partial extension functions. Then the classes of the $\mathcal{F}$-generic sets and the weakly $\mathcal{F}$-generic sets are comeager.*

PROOF. This is immediate by Corollary 3.15 and Proposition 3.32. □

If the class $\mathcal{F}$ consists only of bounded extension functions (see Definition 3.20) then we can strengthen the previous theorem.

Definition 3.34 Let $\mathcal{F}$ be a countable class of bounded partial extension functions. Then (weak) $\mathcal{F}$-genericity is called a *bounded* genericity concept.

Theorem 3.35 *For any bounded genericity concept, the class of generic sets is comeager and has measure 1. I.e., if $\mathcal{F}$ is a countable class of bounded partial extension functions then the classes of the $\mathcal{F}$-generic sets and of the weakly $\mathcal{F}$-generic sets are comeager and have Lebesgue measure 1.*

PROOF. By Theorem 3.21. □

In our definition of an $\mathcal{F}$-generic set G we require that G meets all partial extension functions in $\mathcal{F}$ which are dense along G. For sufficiently closed function classes $\mathcal{F}$, however, G will meet such a partial function f not just once but infinitely often. Since this observation will be very useful in the following, we will address this matter more formally and will introduce some related notation.

Definition 3.36 Let $\mathcal{F}$ be a countable class of partial extension functions and let $\mathcal{F}[tot]$ be the class of total extension functions in $\mathcal{F}$. A set G is *i.o.-$\mathcal{F}$-generic* if G infinitely often meets all partial extension functions in $\mathcal{F}$ which are dense along G, i.e., if G is a member of the class

$$\mathrm{M}^{\infty}_{\mathcal{F}} = \{A : \forall f \in \mathcal{F}\ (f \text{ is not dense along } A \text{ or } A \text{ infinitely often meets } f)\}.$$

A set G is *weakly i.o.-$\mathcal{F}$-generic* if G infinitely often meets all total extension functions in $\mathcal{F}$, i.e., if G is a member of the class

$$\mathrm{M}^{\infty}_{\mathcal{F}[tot]} = \{A : \forall f \in \mathcal{F}[tot]\ (A \text{ infinitely often meets } f)\}.$$

Lemma 3.37 *Let $\mathcal{F}$ be a class of (partial) extension functions which is closed under finite variants. Then any $\mathcal{F}$-generic set is i.o.-$\mathcal{F}$-generic.*

If we say that a class $\mathcal{F}$ of (partial) extension functions is closed under finite variants (c.f.v.) then this means that all extension functions of the same type which are a finite variant of a function in $\mathcal{F}$ are a member of $\mathcal{F}$ again. So, for example, if $\mathcal{F}$ is a class of total functions then we only consider finite variants which are total too and if $\mathcal{F}$ is a class of k-bounded functions then we only consider finite variants which are k-bounded again. Though this use of the term of closure under finite variants might be somewhat ambiguous since in general the type we have in mind will not be mentioned explicitly, the intended type should be obvious from the context so that no confusion should arise.

PROOF OF LEMMA 3.37. Assume that G is $\mathcal{F}$-generic and fix $f \in \mathcal{F}$ such that f is dense along G. We have to show that G infinitely often meets f. For a contradiction assume that this is not the case. Then we may fix $m \geq 0$ such that, for all $n \geq m$ such that $f(\gamma \upharpoonright n)\downarrow$, $(\gamma \upharpoonright n) f(\gamma \upharpoonright n) \not\sqsubseteq \gamma$, where γ is the characteristic sequence of G. Consider the following finite variant g of f: Given x such that $|x| \geq m$, $g(x) = f(x)$. Otherwise, let $g(x) = 1 - G(z_{|x|})$. (Note that, for total or k-bounded f, g is total or k-bounded again. I.e., g is of the same type as f.) Now, by closure of $\mathcal{F}$ under finite variants, $g \in \mathcal{F}$. Moreover, since f is dense along G, g is dense along G too. So, by $\mathcal{F}$-genericity of G, G meets g at some number n. By definition of g, however, G does not meet g at any number less than m. So $n \geq m$. Since f and g agree on all inputs of length at least m it follows that G meets f at n. By $n \geq m$ this contradicts the choice of m. □

We close our discussion of genericity by giving some examples showing how closure properties of a function class $\mathcal{F}$ carry over to the corresponding class of $\mathcal{F}$-generic sets.

Definition 3.38 A class $\mathcal{F}$ of (partial) extension functions is closed under *finite replacement* if for any words x and x' with $|x| = |x'|$ the following holds. For any function $f \in \mathcal{F}$ there is a function $f' \in \mathcal{F}$ such that $f'(x'y) = f(xy)$ for all $y \in \Sigma^*$.

Lemma 3.39 *Let $\mathcal{F}$ be a class of (partial) extension functions which is closed under finite replacement. Then the class of the i.o.-$\mathcal{F}$-generic sets is closed under finite variants.*

PROOF. Assume that G is i.o.-$\mathcal{F}$-generic and that G' is a finite variant of G. We have to show that G' is i.o.-$\mathcal{F}$-generic too. So fix $f \in \mathcal{F}$. It suffices to show that G' meets f infinitely often. Since G' is a finite variant of G we may fix a number n such that $G(z_m) = G'(z_m)$ for all $m \geq n$. Now let $x = G' \upharpoonright n$ and $x' = G \upharpoonright n$. Then, by closure of $\mathcal{F}$ under finite replacement, there is a function $f' \in \mathcal{F}$ such that $f'(x'y) = f(xy)$ for all $y \in \Sigma^*$. By choice of n, x and x', G' meets f at $m \geq n$ iff

G meets f' at m. Since, by i.o.-$\mathcal{F}$-genericity, G meets f' infinitely often, it follows that G' meets f infinitely often. □

Note that, by Lemmas 3.37 and 3.39, for any countable class $\mathcal{F}$ of (partial) extension functions which is closed under finite variants and finite replacement the class of $\mathcal{F}$-generic sets is closed under finite variants.

Recall that for a partial function $f : \Sigma^* \to \Sigma^*$ the *dual* function $\hat{f}$ is defined by $\hat{f}(\overline{x}) = \overline{f(x)}$ if $f(x)$ is defined and $\hat{f}(\overline{x}) \uparrow$ otherwise, where $\overline{x}$ is the dual string of x, i.e., $\overline{\varepsilon} = \varepsilon$ and $\overline{(a_1 \dots a_n)} = \overline{a_1} \dots \overline{a_n} = (1 - a_1) \dots (1 - a_n)$.

Lemma 3.40 *Let $\mathcal{F}$ be a class of (partial) extension functions such that, for any (partial) function $f \in \mathcal{F}$, $\hat{f} \in \mathcal{F}$. Then the class of (i.o.-) $\mathcal{F}$-generic sets is closed under complement.*

PROOF. This easily follows from the observation that, for any set A and any partial extension function f, f is dense along A iff $\hat{f}$ is dense along $\overline{A}$, and A meets f at n iff $\overline{A}$ meets $\hat{f}$ at n. □

CHAPTER 4

Bounded Finite-State Genericity

In this chapter – which is the core of this thesis - we introduce and analyze bounded finite-state genericity, i.e., genericity notions based on bounded extension functions computable by finite automata. By analyzing the properties which all of the corresponding generic sets have in common we can decide which properties can be forced by finite extension or wait-and-see arguments where the strategy for meeting a single requirement is bounded and computable by a finite automaton.

In Section 4.1.1 we introduce the basic concepts: k-reg-genericity capturing regular partial extensions of length k, ω-reg-genericity or bounded reg-genericity capturing regular partially defined extensions of arbitrary constant length, and the corresponding weak genericity notions based on extensions which are defined everywhere. We also show that k-reg-genericity and ω-reg-genericity coincide (for any k), i.e., that the power of partially defined bounded extension strategies computable by finite automata does not depend on the length k of the extension. In order to show that, in contrast to this observation, the power of totally defined bounded finite-state extension strategies depends on the length of the admissible extension, in Section 4.1.2 we explore the saturation properties of the different types of reg-generic sets. Our main result here is that both, the bounded reg-generic sets and the weakly ω-reg-generic sets coincide with the saturated sets. This result can be viewed as Baire category counter part to the result of Schnorr and Stimm (1971/72) in the setting of measure which asserts that the finite-state random sets are just the sets with normal characteristic sequence. Our result also shows that if we consider extension strategies of constant but arbitrary length then partially defined finite-state strategies are not more powerful than totally defined finite-state strategies. This surprising result contrasts results on genericity and weak genericity in the setting of complexity theory (see the results on P-immunity in Section 3.4). After discussing some closure properties of the bounded finite-state genericity notions in Section 4.1.3, we then analyze the diagonalization strength of these genericity notions (Section 4.1.4). In particular we show that no bounded reg-generic set is regular but that there are context-free – in fact linear – languages which are bounded reg-generic. So this genericity notion provides a strong separation of the classes of regular and context-free (or linear) languages at the bottom of the Chomsky hierarchy. We may also conclude that these genericity notions do not imply REG-bi-immunity since, as we have shown in Section 2.4, no context-free language is REG-bi-immune. Finally, we illustrate the difference in power of the weak and general bounded finite-state genericity concepts if we fix the length of the extension by showing that, in contrast to the above, there are regular weakly 1-reg-generic sets.

In Section 4.2 we discuss some variants of the bounded finite-state genericity concepts which are based on extension strategies working with partial information on the initial segment specified previously. We consider the following two limitations: First (in Section 4.2.1 we consider length invariant extension strategies, i.e., extension strategies which are not given the initial segment itself but only its length. Second (in Section 4.2.2) we look at oblivious extension strategies, i.e., strategies which remember the last k bits of the given initial segment (for some constant k). By comparing the strength of the corresponding (apparently weaker) finite-state genericity concepts with the previously introduced concepts we can see what information on a given initial segment can be extracted and used by a finite automaton. While for the common bounded genericity concepts in computational complexity theory the corresponding length invariant or oblivious genericity concepts are strictly weaker, here we show that some of the length invariant and oblivious bounded finite-state genericity notions coincide with bounded reg-genericity thereby demonstrating the low computational power of finite automata.

In the remainder of this chapter we discuss the question whether we can introduce some stronger bounded finite-state genericity concepts which force REG-bi-immunity. First, in Section 4.3 we formalize finite-state Cantor style diagonalization arguments. In such a diagonalization argument, the diagonalization at a string x does not depend on the values of the constructed set A on the previous strings (as in a finite extension argument) but is independent of the previously specified part of A. By formalizing these arguments in terms of diagonalization functions and by introducing corresponding genericity notions we show that total finite-state Cantor diagonalization functions can force nonregularity (but not more) and partial finite-state Cantor diagonalization functions can force REG-bi-immunity (but not more), namely the finite-state Cantor style generic sets are just the REG-bi-immune sets and the weakly finite-state Cantor style generic sets are just the nonregular languages. Based on these observations, in Section 4.4 we introduce the desired stronger bounded finite-state genericity concepts subsuming both, bounded reg-genericity and Cantor style reg-genericity, by considering regular extension functions which obtain as their input the given finite initial segment in a redundant form allowing a finite automaton to extract both, the standard representation of this initial segment and the string at which the diagonalization takes place.

4.1 Bounded reg-Genericity

In this section we introduce and investigate bounded genericity concepts based on regular extension functions. By considering total extension functions and partial extension functions and by considering extensions of arbitrary constant length and extensions of given constant length we get a variety of concepts. Recall that we have introduced (partial) regular functions f of type $f:\Sigma^* \to \Sigma^k$ in Section 2.3 and the notion of bounded and k-bounded extension function in Section 3.3. There we have also explained what it means that a partial function is dense along a set and that a set meets an extension function.

4.1.1 Definitions and Basic Facts

Definition 4.1 A set G is *k-reg-generic* if it meets all regular partial k-bounded extension functions which are dense along G; G is *weakly k-reg-generic* if G meets all regular total k-bounded extension functions; G is *ω-reg-generic* or *bounded reg-generic* if G is k-reg-generic for all $k \geq 1$, i.e., if G meets all regular partial bounded extension functions which are dense along G; G is *weakly ω-reg-generic* if G is weakly k-reg-generic for all $k \geq 1$, i.e., if G meets all regular total bounded extension functions.

We apply the above notions to infinite sequences as well as to sets. E.g. we call a sequence α k-reg-generic if the set $S(\alpha)$ corresponding to α is k-reg-generic.

The following relations among the bounded finite-state genericity concepts are immediate by definition (where $k \geq 2$).

$$\begin{array}{ccc}
\omega\text{-reg-generic} & \Longrightarrow & \text{weakly } \omega\text{-reg-generic} \\
\Downarrow & & \Downarrow \\
(k+1)\text{-reg-generic} & \Longrightarrow & \text{weakly } (k+1)\text{-reg-generic} \\
\Downarrow & & \Downarrow \\
k\text{-reg-generic} & \Longrightarrow & \text{weakly } k\text{-reg-generic} \\
\Downarrow & & \Downarrow \\
1\text{-reg-generic} & \Longrightarrow & \text{weakly } 1\text{-reg-generic}
\end{array} \tag{4.1}$$

Note that the finite-state genericity concepts above are genericity notions in the sense of Definition 3.31. For instance, (weak) ω-reg-genericity coincides with (weak) $\mathcal{F}$-genericity if we let $\mathcal{F}$ be the class of partial regular extension functions of type $f:\Sigma^* \to \Sigma^k$ for $k \geq 1$. Moreover, all of the above genericity concepts are bounded in the sense of Definition 3.34 whence, by Theorem 3.35, the corresponding generic sets are abundant in the sense of both, category and measure. By (4.1) it suffices to state this observation for bounded reg-genericity.

Theorem 4.2 *The class of bounded reg-generic sets (i.e.,* ω*-reg-generic sets) is comeager and has measure 1.*

PROOF. By Theorem 3.35. □

Before we address the question which of the implications in (4.1) are strict, it will be useful to note that, for the above genericity notions, a generic set meets a corresponding extension function not just once but infinitely often. This technical fact will be applied in the proofs of many of our results.

Lemma 4.3 *Let A be k-reg-generic. Then A infinitely often meets every regular partial k-bounded extension function f which is dense along A, i.e.,* $(A \upharpoonright n) f(A \upharpoonright n) \sqsubset A$ *for infinitely many n. Similarly, any weakly k-reg-generic set A meets every regular total k-bounded extension function infinitely often.*

PROOF. Since, by Lemma 2.46, the class of regular (partial) k-bounded extension functions is closed under finite variants, this is immediate by Lemma 3.37. □

Next we use Lemma 4.3 to show that from the above genericity concepts the concepts based on partial extension functions coincide.

Theorem 4.4 *For any set A the following are equivalent.*

1. *A is bounded reg-generic, i.e.,* ω*-reg-generic.*

2. *A is* 1*-reg-generic.*

PROOF. By (4.1) it suffices to show that any 1-reg-generic set is ω-reg-generic. So let A be 1-reg-generic. In order to show that A is ω-reg-generic, by definition, it suffices to show that A is k-reg-generic for all numbers $k \geq 1$. We proceed by induction. The case $k = 1$ holds by assumption. For the inductive step fix $k \geq 1$ and, by inductive hypothesis, assume that A is k-reg-generic. We have to show that A is $(k+1)$-reg-generic. Let f be any regular partial $(k+1)$-bounded extension functions which is dense along A. It suffices to show that A meets f. For x such that $f(x)$ is defined, let $f(x)^-$ be the first k bits of $f(x)$, and, for $i \leq 1$, define the partial k-bounded extension function f_i by letting $f_i(x) = f(x)^-$ if $f(x)$ is defined and $f(x) = f(x)^- i$ and by letting $f_i(x)$ be undefined otherwise. Then, as one can easily check, the partial functions f_0 and f_1 are regular and k-bounded. Moreover, for any x such that $f(x)$ is defined, either $f_0(x)$ or $f_1(x)$ is defined. Hence, by density of f along A, f_0 or f_1 is dense along A too. For the remainder of the proof fix $i \leq 1$ such that the regular partial k-bounded extension function f_i is dense along A.

Then, by inductive hypothesis and by Lemma 4.3, A meets f_i infinitely often, i.e.,

$$\exists^\infty n \geq 1 \, ((\alpha \restriction n) f_i(\alpha \restriction n) \sqsubset \alpha) \tag{4.2}$$

where α is the characteristic sequence of A. Define the partial 1-bounded extension function $g : \Sigma^* \to \Sigma$ by letting

$$g(w) = \begin{cases} i & \text{if } \exists x \sqsubset w \, (f_i(x) \downarrow \;\; \& \;\; w = x f_i(x)) \\ \uparrow & \text{otherwise.} \end{cases}$$

Then, as one can easily check, g is regular and, by (4.2), g is dense along A. So, by 1-reg-genericity of A, A meets g at some number n, i.e., $g(\alpha \restriction n)$ is defined and

$$(\alpha \restriction n) g(\alpha \restriction n) \sqsubset \alpha. \tag{4.3}$$

By definition of f_i and g, the former implies that $f(\alpha \restriction (n-k))$ and $f_i(\alpha \restriction (n-k))$ are defined and

$$(\alpha \restriction n) g(\alpha \restriction n) = (\alpha \restriction n) i = (\alpha \restriction (n-k)) f_i(\alpha \restriction (n-k)) i = (\alpha \restriction (n-k)) f(\alpha \restriction (n-k)).$$

By (4.3) this implies that $(\alpha \restriction (n-k)) f(\alpha \restriction (n-k)) \sqsubset \alpha$, i.e, that A meets f at $n-k$. □

In order to determine which of the other implications in (4.1) are strict, next we explore the saturation properties of the bounded finite-state generic sets.

4.1.2 Finite-State Genericity vs. Saturation

We will now show that some of the genericity concepts in (4.1) coincide with saturation. To establish this we will need the following two relations between the bounded finite-state genericity notions in (4.1) and (partial) saturation.

Lemma 4.5 *Let A be weakly k-reg-generic ($k \geq 1$). Then A is k-ω-saturated.*

PROOF. Given any string x of length k we have to show that x occurs infinitely often in the characteristic sequence α of A, i.e., there are infinitely many n such that $(A \restriction n)x \sqsubset \alpha$. Define the k-bounded regular extension function f by $f(y) = x$ for all $y \in \Sigma^*$. Then, by weak k-reg-genericity of A and by Lemma 4.3, $(A \restriction n)x = (A \restriction n) f(A \restriction n) \sqsubset \alpha$ for infinitely many n. □

Lemma 4.6 *Let A be saturated. Then A is 1-reg-generic.*

The proof of this lemma uses the characterization of saturated sequences in terms of regular partial prediction functions together with the observation that partial 1-bounded extension functions may alternatively be interpreted as partial prediction functions.

PROOF. Let $f : \Sigma^* \to \Sigma$ be a regular partial 1-bounded extension function which is dense along A. We have to show that A meets f at some n. For a contradiction assume that this is not the case, and let α be the characteristic sequence of A. Then, by density of f along A,

$$\exists^\infty n\, (f(\alpha \upharpoonright n)\downarrow) \tag{4.4}$$

and, by failure of A to meet f,

$$\forall n\, (f(\alpha \upharpoonright n)\downarrow \Rightarrow (\alpha \upharpoonright n)(1 - f(\alpha \upharpoonright n)) \sqsubset \alpha) \tag{4.5}$$

Now let $\overline{f}$ be the negation of f, i.e., to be more precise,

$$\overline{f}(x) = \begin{cases} 1 - f(x) & \text{if } f(x)\downarrow \\ \uparrow & \text{otherwise.} \end{cases}$$

Then, obviously, $\overline{f}$ is regular and, by (4.4) and (4.5),

$$\exists^\infty n\, (\overline{f}(\alpha \upharpoonright n)\downarrow)$$

and

$$\forall n\, (\overline{f}(\alpha \upharpoonright n)\downarrow \Rightarrow (\alpha \upharpoonright n)\overline{f}(\alpha \upharpoonright n) \sqsubset \alpha).$$

So, if we view $\overline{f}$ as a partial prediction function, then, by the former, $\overline{f}$ makes infinitely many predictions about α and, by the latter, all predictions about α made by $\overline{f}$ are correct. It follows by Theorem 2.120 that α is not saturated which gives the desired contradiction. □

We are now ready to state the following equivalence theorem.

Theorem 4.7 *The following are equivalent.*

(i) A is saturated.

(ii) A is 1-reg-generic.

(iii) A is bounded reg-generic, i.e., ω*-reg-generic.*

(iv) A is weakly ω*-reg-generic.*

The equivalence of saturation and bounded reg-genericity can be viewed as an effectivization of Staiger's observation that the class of saturated sequences is comeager and has measure 1 (see Theorem 2.101): Since bounded reg-genericity is a bounded genericity concept, the class of bounded reg-generic sets is comeager and has measure 1 (see Theorem 4.2).

PROOF OF THEOREM 4.7. It suffices to show the implications $(i) \Rightarrow (ii) \Rightarrow (iii) \Rightarrow (iv) \Rightarrow (i)$. The implication $(i) \Rightarrow (ii)$ holds by Lemma 4.6, the implication $(ii) \Rightarrow (iii)$ holds by Theorem 4.4, and the implication $(iii) \Rightarrow (iv)$ is immediate by definition (see (4.1)). Finally, for a proof of the implication $(iv) \Rightarrow (i)$ note that, by Lemma 4.5, any weakly k-reg-generic set is k-ω-saturated ($k \geq 1$). Since, by definition, a weakly ω-reg-generic set A is weakly k-reg-generic for all $k \geq 1$, it follows that A is k-ω-saturated for all $k \geq 1$, hence saturated. □

In contrast to Theorem 4.7, the weak k-reg-genericity notions lead to a proper hierarchy for growing k. This also follows from the partial saturation properties of the weakly k-reg-generic sets. It suffices to complement the positive saturation property of these sets in Lemma 4.5 by the following negative result.

Lemma 4.8 *For any $k \geq 1$ there is a weakly k-reg-generic set A which is not $(k+1)$-1-saturated.*

PROOF. It suffices to construct a weakly k-reg-generic set A such that $(A \upharpoonright n)1^{k+1} \not\sqsubseteq \alpha$ for all $n \geq 0$, where α is the characteristic sequence of A. We do this by a finite extension argument. Fix an enumeration $\{f_e : e \geq 0\}$ of the total regular k-bounded extension functions. Define $n_0 < n_1 < n_2 \ldots$ and $A \upharpoonright n_e$ by induction on $e \geq 0$ by letting $n_0 = 0$ and $A \upharpoonright n_{e+1} = (A \upharpoonright n_e) f_e(A \upharpoonright n_e)0$. Obviously this ensures that A has the required properties. □

Theorem 4.9 *For any $k \geq 1$ there is a weakly k-reg-generic set which is not weakly $(k+1)$-reg-generic.*

PROOF. This is immediate by Lemmas 4.5 and 4.8. □

Theorems 4.7 and 4.9 together with (4.1) give the desired complete characterization of the relations among the bounded finite-state genericity notions.

Theorem 4.10 *For $k \geq 2$ the following and only the following implications hold* *(up to transitive closure).*

$$\begin{array}{ccc} \omega\text{-}\mathit{reg}\text{-}\mathit{generic} & \Longleftrightarrow & \mathit{weakly}\ \omega\text{-}\mathit{reg}\text{-}\mathit{generic} \\ \Updownarrow & & \Downarrow \\ (k+1)\text{-}\mathit{reg}\text{-}\mathit{generic} & & \mathit{weakly}\ (k+1)\text{-}\mathit{reg}\text{-}\mathit{generic} \\ \Updownarrow & & \Downarrow \\ k\text{-}\mathit{reg}\text{-}\mathit{generic} & & \mathit{weakly}\ k\text{-}\mathit{reg}\text{-}\mathit{generic} \\ \Updownarrow & & \Downarrow \\ 1\text{-}\mathit{reg}\text{-}\mathit{generic} & & \mathit{weakly}\ 1\text{-}\mathit{reg}\text{-}\mathit{generic} \end{array} \tag{4.6}$$

By our detailed analysis of the saturated sets and sequences in Section 2.6, we can exploit the relations between saturation and bounded finite-state genericity in order to obtain a series of interesting results on the latter notion. In the next two subsections we will use this approach in order to give some closure properties and to analyze the diagonalization strength of the bounded finite-state genericity notions.

4.1.3 Closure Properties

By Theorem 4.7 the closure properties of the saturated sets and sequences obtained in Section 2.6 directly carry over to the bounded reg-generic sets and sequences. In particular we obtain the following.

Lemma 4.11 *The class of the bounded reg-generic sets is closed under finite variants and under complement. Moreover, the class of the bounded reg-generic sequences is closed under closeness.*

PROOF. By Theorem 4.7 this immediately follows from Proposition 2.109 and Lemma 2.108. □

Lemma 4.12 *Let A be bounded reg-generic.*

1. *For any word $w \in \Sigma^*$, wA is bounded reg-generic too. In fact, any set B such that $B \cap w\Sigma^* = wA$ is bounded reg-generic.*

2. *For any set B, the effective disjoint union of A and B, $A \oplus B = 0A \cup 1B = \{0v : v \in A\} \cup \{1w : w \in B\}$, is bounded reg-generic.*

PROOF. By Theorem 4.7 this immediately follows from Lemma 2.111. □

In order to explore the closure properties of the weakly k-reg-generic sets we have to use some more direct arguments. Next we will discuss the results corresponding to Lemma 4.11 in this setting.

Lemma 4.13 *For $k \geq 1$, the class of the weakly k-reg-generic sets is closed under finite variants and under complement.*

PROOF. As one can easily show, the class of total regular k-bounded extension functions is closed under finite replacement and under dual functions. So the closure of the class of the weakly k-reg-generic sets under finite variants and under complement follows from Lemmas 3.39, 3.40 and 4.3. □

In contrast to Lemma 4.11, however, the classes of the weakly k-reg-generic sequences ($k \geq 1$) are not closed under closeness.

Lemma 4.14 *For any $k \geq 1$ there are sequences α and β such that α is weakly k-reg-generic, β is close to α and β is not weakly k-reg-generic.*

PROOF. Let $k \geq 1$ be given. By a finite extension argument we construct a weakly k-reg-generic set A such that $A \setminus \{0\}^*$ is not weakly k-reg-generic. Then the characteristic sequences α and β of the sets A and $A \setminus \{0\}^*$, respectively, have the required properties.

Let $\mathrm{F} = \{f_n : n \geq 0\}$ be an enumeration of the regular total k-bounded extension functions and fix n_0 such that $k < 2^{n_0}$. Then $A \upharpoonright 0^{n_0+e}$ is defined by induction on e as follows. Let $A \upharpoonright 0^{n_0}$ be the empty set, i.e., the sequence $0^{2^{n_0}-1}$. Then, given $A \upharpoonright 0^{n_0+e}$, let $A \upharpoonright 0^{n_0+e+1} = A \upharpoonright 0^{n_0+e} f_e(A \upharpoonright 0^{n_0+e}) 0^{2^{n_0+e+1}-k}$. So the extension $A \upharpoonright 0^{n_0+e+1}$ of $A \upharpoonright 0^{n_0+e}$ ensures that A meets the e-th regular k-bounded extension function f_e. The block of zeroes following $A \upharpoonright 0^{n_0+e} f_e(A \upharpoonright 0^{n_0+e})$ in the definition of $A \upharpoonright 0^{n_0+e+1}$ ensures that any block 1^m of ones in α has length at most k. Moreover, such a block $A(z_p)...A(z_{p+k-1}) = 1^k$ of maximum length k begins with position $z_p = 0^{n_0+e}$ for some $e \geq 0$. So the word 1^k does not occur in the characteristic sequence β of $A \setminus \{0\}^*$. By Lemma 4.5 this implies that $A \setminus \{0\}^*$ is not weakly k-reg-generic. □

4.1.4 On the Diagonalization Strength of Bounded reg-Genericity

We will now look at the diagonalization strength of the bounded finite-state genericity concepts where, by (4.6), it suffices to consider bounded reg-genericity and weak k-reg-genericity for $k \geq 1$. The first question to ask here is of course whether these concepts are strong enough to diagonalize over all regular sequences and all regular sets, i.e., whether the characteristic sequence of any generic set (of a given type) is nonregular and whether any generic set is nonregular. Recall that in Section 2.5.2 we have shown that any language which has a regular characteristic sequence is regular but that there are regular languages with nonregular characteristic sequences.

Our first observation is that any of the bounded finite-state genericity concepts forces nonregularity of the characteristic sequence. In fact, for our weakest reg-genericity concept, the generic sets are just the sets with nonregular characteristic sequence.

Theorem 4.15 *A set A is weakly 1-reg-generic if and only if the characteristic sequence α of A is not regular.*

PROOF. The proof is by contraposition. First assume that α is regular. Then, by Theorem 2.91, α can be predicted by a finite automaton, i.e., the function $f : \Sigma^* \to \Sigma$ defined by $f(x) = \alpha(|x|)$ is regular. It follows that the function $\overline{f}$ defined by $\overline{f}(x) = 1 - f(x)$ is a regular 1-bounded extension function and that A does not meet $\overline{f}$. So A is not weakly 1-reg-generic.

For a proof of the other direction assume that α is not weakly 1-reg-generic. Then there is a total regular 1-bounded extension function f such that $f(\alpha \restriction n) \neq \alpha(n)$ for all n. Hence, for $\overline{f}$ defined by $\overline{f}(x) = 1 - f(x)$, $\overline{f}$ is regular again and $\overline{f}(\alpha \restriction n) = \alpha(n)$ for all $n \geq 0$ whence α is regular by Theorem 2.91. □

This characterization of weak 1-reg-genericity has some interesting consequences. First we deduce that, in contrast to bounded reg-genericity, weak k-reg-genericity cannot be characterized solely in terms of (partial) saturation. Since weak k-reg-genericity does not coincide with saturation it suffices to show that none of the partial saturation properties implies weak k-reg-genericity.

Corollary 4.16 *For any $k, k' \geq 1$ there is a k-ω-saturated set A which is not weakly k'-reg-generic.*

PROOF. By Lemma 2.137 there is a regular k-ω-saturated sequence α. So, for $A = S(\alpha)$, A is k-ω-saturated but A is not weakly 1-reg-generic by Theorem 4.15, hence not weakly k'-reg-generic by (4.1). □

Next we observe that forcing with regular total 1-bounded extensions functions is not strong enough to force nonregularity in the sense of languages. Since, by Theorem 2.81, there are regular languages with nonregular characteristic sequences, this is immediate by the preceding theorem.

Corollary 4.17 *There is a weakly* 1*-reg-generic set which is regular.*

As observed in the proof of Theorem 2.81, the unary language $\{0\}^*$ is regular but has a non-regular characteristic sequence. So $\{0\}^*$ is a natural example of a regular language which is weakly 1-reg-generic. This observation can be extended as follows. In general, no infinite unary language and no infinite length-language has a regular characteristic sequence. (Here we call A a *length-language* if A is length invariant, i.e., if, for any words x and y with $|x| = |y|$, $x \in A$ if and only if $y \in A$.) Hence all these languages are weakly 1-reg-generic. In contrast, however, none of these languages is weakly 2-generic.

Lemma 4.18 *Let A be an infinite unary language or length language. Then A is weakly 1-reg generic but not weakly 2-reg-generic.*

PROOF. First assume that $A \subseteq \{0\}^*$ is infinite and let α be the characteristic sequence of A. Then, as one can easily check, α is not almost periodic, hence not regular by Theorem 2.76. So, by Theorem 4.15, A is weakly 1-reg-generic. To show that A is not weakly 2-reg-generic it suffices to note that the word 11 occurs in the characteristic sequence of any unary language at most once. (Namely, the only possible occurrence of 11 may be at the first two bits of the sequence if $0^0 = z_0$ and $0^1 = z_1$ are both members of A.) So A is not 2-2-saturated, hence not weakly 2-reg-generic by Lemma 4.5.

Now assume that A is an infinite length language, i.e., that there is an infinite set D of numbers such that $A = \{x : |x| \in D\}$. Then, as in the case of unary languages, α is not almost periodic, hence A is weakly 1-reg-generic. To show that A is not weakly 2-reg-generic, however, we cannot apply Lemma 4.5 since, for infinite and co-infinite D, the set A is 2-ω-saturated. We observe, however, that the words 010 and 101 do not occur in the characteristic sequence of any length language. So the language A will meet the total regular extension function $f : \Sigma^* \to \Sigma^2$ defined by $f(\varepsilon) = 00$, $f(x0) = 10$ and $f(x1) = 01$ ($x \in \Sigma^*$) at most once (namely at $n = 0$). By Lemma 4.3 this implies that A is not weakly 2-reg-generic. □

As we will show next, bounded reg-genericity does not only force nonregularity of the characteristic sequence but nonregularity of the language itself too. We obtain this result by the coincidence of this genericity notion with saturation and our analysis of the complexity of saturated sets in Section 2.6.

Theorem 4.19 *Let A be bounded reg-generic. Then A is not regular.*

PROOF. By Theorem 4.7, A is saturated and, by Theorem 2.115, no saturated language is regular. □

We do not know, whether this theorem can be extended to weak k-genericity for any $k \geq 2$ or whether there are regular weakly k-reg-generic sets. Again by using the coincidence of saturation and bounded reg-genericity, however, we can show that there are context-free languages - in fact linear languages - which are bounded reg-generic.

Theorem 4.20 *There is a linear language A which is bounded reg-generic.*

PROOF. By Theorems 4.7 and 2.116. □

The preceding two theorems show that we may say that bounded reg-genericity is a genericity concept pertaining to the class REG of regular languages since the concept is strong enough to allow us to diagonalize over this class but, on the other hand, it is not too strong, so that we can obtain these diagonalizations inside the next bigger classes in the Chomsky hierarchy, namely the classes LIN and CF of the linear and context-free languages.

Of course we may ask what stronger properties related to REG can be forced by bounded reg-genericity. The probably most important properties here are REG-immunity and REG-bi-immunity. In Section 2.4.3 we have shown that no context-free language is REG-bi-immune though there are context-free REG-immune sets. By the former, Theorem 4.20 implies that there are bounded reg-generic sets which are not REG-bi-immune. We next extend this observation to REG-immunity.

Theorem 4.21 *There is a bounded reg-generic set A such that neither A nor $\overline{A}$ is* REG-*immune.*

PROOF. By the coincidence of bounded reg-genericity and saturation this is immediate by Theorem 2.117. □

Below we will introduce some stronger genericity concepts for REG which force REG-bi-immunity. As observed above, such a concept must entail certain diagonalizations over the class CF of context-free languages since REG-bi-immune sets cannot be context free. So, in contrast to bounded reg-genericity, such a genericity concepts will not pertain to REG in the sense discussed above.

In the following sections we will consider some variants of finite-state genericity concepts based on bounded extension functions. First, we consider extension functions which may use the information on the given initial segment only in part thereby leading to some apparently weaker bounded reg-genericity notions. Second, we will give the extension functions the initial segment in some enriched form as an input which will yield stronger bounded reg-generic notions. For preparing the latter concept we will also discuss the power of finite-state Cantor style diagonalizations were the diagonalization at a string x does not depend on the previously defined initial segment of the set under construction but only on the string x itself.

4.2 Extensions Based on Partial Information

In the following we discuss some bounded regular genericity concepts which are based on extension functions which obtain as their input not a finite initial segment of a sequence but only some partial information on the initial segment. We consider two cases: In the first case the extension function is given the length of the initial segment (in unary notation). In the second case, only the last m bits of the initial segment (for some constant $m \geq 1$) are given. By comparing the bounded finite-state genericity notions based on these limitations with the standard bounded finite-state genericity notions introduced in the preceding section we can analyse the type of information which a finite automaton can extract from a given initial segment.

4.2.1 Length Invariant Extension Functions

An extension function which is given only the length of the current initial segment not the initial segment itself may be described as a length invariant extension function. In case of regular bounded extension functions this leads to the following definition.

Definition 4.22 Let $f : \Sigma^* \to \Sigma^k$ be a (partial) k-bounded extension function. f is *length invariant* if $f(w) = f(w')$ for all words w and w' with $|w| = |w'|$. A set G is *li-k-reg-generic* if it meets all regular partial k-bounded extension functions which are length invariant and dense along G; and G is *weakly li-k-reg-generic* if G meets all regular total k-bounded extension functions which are length invariant. G is *(weakly) li-ω-reg-generic* if G is (weakly) li-k-reg-generic for all $k \geq 1$.

Note that a partial length invariant extension function f which is defined infinitely often is dense along all sets. So we call such a function f *dense* if the domain of f is infinite or, equivalently, if $f(0^n)\downarrow$ for infinitely many numbers n. Alternatively we can describe a length invariant k-bounded extension function f by a function $\hat{f} : \{0\}^* \to \Sigma^k$. We say that such a function $\hat{f}$ is *dense* if the domain of $\hat{f}$ is infinite, and we say that a set A *meets* $\hat{f}$ *at some number* n if $\hat{f}(0^n)$ is defined and $(A \restriction n)\hat{f}(0^n) \sqsubset \chi(A)$ and that A *meets* $\hat{f}$ if A meets $\hat{f}$ at some n.

Proposition 4.23 *A set G is weakly li-k-reg-generic if G meets all total regular functions $\hat{f} : \{0\}^* \to \Sigma^k$. G is li-k-reg-generic if and only if G meets all partial regular functions $\hat{f} : \{0\}^* \to \Sigma^k$ which are dense.*

PROOF. Consider the following correspondence between length invariant k-bounded extension functions $f : \Sigma^* \to \Sigma^k$ and functions $\hat{f} : \{0\}^* \to \Sigma^k$: Given f let $\hat{f}$ be defined by $\hat{f}(0^n) = f(0^n)$. Conversely, given $\hat{f}$ let f be defined by $f(x) = \hat{f}(0^{|x|})$.

Then f is regular if and only if $\hat{f}$ is regular, f is total if and only if $\hat{f}$ is total, f is dense along a given set A if and only if $\hat{f}$ is dense, and A meets f at n if and only if A meets $\hat{f}$ at n. By definition, these observations easily imply the claims. □

Just as in case of the standard bounded finite-state generic sets, a length invariant finite-state generic set meets a corresponding extension function not just once but infinitely often, and the length invariant finite-state genericity notions are closed under complement.

Lemma 4.24 *Let A be li-k-reg-generic. Then A infinitely often meets every regular partial length invariant k-bounded extension function f which is dense, i.e., $(A \upharpoonright n) f(A \upharpoonright n) \sqsubset A$ for infinitely many n. Similarly, any weakly li-k-reg-generic set A meets every regular total length invariant k-bounded extension function infinitely often.*

Lemma 4.25 *Let A be (weakly) li-k-reg-generic. Then $\overline{A}$ is (weakly) li-k-reg-generic to ($k \geq 1$ or $k = \omega$).*

We omit the straightforward proofs of the preceding two lemmas and turn to the comparison of the length invariant bounded finite-state genericity concepts. The following relations are immediate by definition (for $k \geq 1$).

$$\begin{aligned} \text{(weakly) li-}\omega\text{-reg-generic} &\Rightarrow \text{(weakly) li-}(k+1)\text{-reg-generic} \\ &\Rightarrow \text{(weakly) li-}k\text{-reg-generic} \end{aligned} \tag{4.7}$$

If we compare the strength of the genericity concepts based on regular length-invariant extension functions with that of the standard bounded reg-genericity concepts, the following relations are immediate by definition.

$$\begin{array}{ccc} k\text{-reg-generic} & \Rightarrow & \text{li-}k\text{-reg-generic} \\ \Downarrow & & \Downarrow \\ \text{weakly } k\text{-reg-generic} & \Rightarrow & \text{weakly li-}k\text{-reg-generic} \end{array} \tag{4.8}$$

We can combine the above relations in the following table where, by the equivalences in (4.6), we may omit reference to k-reg-genericity (for $k \in \mathbb{N} \cup \{\omega\}$).

$$\begin{array}{ccccc}
\text{li-}\omega\text{-reg-generic} & \Rightarrow & \text{weakly li-}\omega\text{-reg-generic} & \Leftarrow & \text{weakly }\omega\text{-reg-generic} \\
\Downarrow & & \Downarrow & & \Downarrow \\
\text{li-}(k+1)\text{-reg-generic} & \Rightarrow & \text{weakly li-}(k+1)\text{-reg-generic} & \Leftarrow & \text{weakly }(k+1)\text{-reg-generic} \\
\Downarrow & & \Downarrow & & \Downarrow \\
\text{li-}k\text{-reg-generic} & \Rightarrow & \text{weakly li-}k\text{-reg-generic} & \Leftarrow & \text{weakly }k\text{-reg-generic} \\
\Downarrow & & \Downarrow & & \Downarrow \\
\text{li-1-reg-generic} & \Rightarrow & \text{weakly li-1-reg-generic} & \Leftarrow & \text{weakly 1-reg-generic}
\end{array} \tag{4.9}$$

In the following we will determine which of these implications are strict. This will require to prove a series of facts which will also illustrate some of the differences between these concepts.

We first look at the saturation properties of the various length invariant finite-state genericity notions.

Lemma 4.26 *Let A be weakly li-k-reg-generic ($k \geq 1$). Then A is k-ω-saturated.*

PROOF. Given $x \in \Sigma^k$ it suffices to show that x occurs in the characteristic sequence α of A infinitely often. I.e., given $n \geq 1$ we have to show that there is a number $m \geq n$ such that $\alpha(m)\dots\alpha(m+k-1) = x$. Consider the total k-bounded extension function f defined by $f(y) = x$ for all $y \in \Sigma^*$ with $|y| \geq n$ and $f(y) = (1-A(z_{|y|}))^k$ for strings y with $|y| < n$. Then f is regular and length invariant. So, by assumption, A meets f at some m. By choice of f this implies that $m \geq n$ and

$$(A \upharpoonright m)x = (A \upharpoonright m)f(A \upharpoonright m) \sqsubset \alpha.$$

□

The preceding lemma shows that any (weakly) li-ω-reg-generic set is saturated. By Theorem 4.7 this implies the following equivalence theorem.

Theorem 4.27 *The following are equivalent.*

(i) A is saturated.

(ii) A is 1-reg-generic.

(iii) A is bounded reg-generic, i.e., ω-reg-generic.

(iv) A is weakly ω*-reg-generic.*

(v) A is li-ω*-reg-generic.*

(vi) A is weakly li-ω*-reg-generic.*

Theorem 4.27 shows that the diagonalization strength of bounded regular extension functions is not decreased if we limit ourselves to total functions or length invariant functions (or functions which are both, total and length invariant). This observation, however, is based on the assumption that the length of the extensions is not fixed. I.e., replacing an extension function by (a set of) equivalent total or length invariant extension functions may lead to functions of higher norm. For moving from partial to total functions we already have observed this in the preceding section. Next we will make similar observations for the length invariant case. In particular we will show that, in contrast to k-reg-genericity, the strength of li-k-reg-genericity depends on the norm k. This will be established by the following negative saturation result for li-k-reg-genericity which, for later use, we will state not only for length invariant finite-state genericity but for length invariant genericity related to any countable class.

Lemma 4.28 *Let $k \geq 1$ and let $\mathcal{F}$ be any countable set of partial length invariant k-bounded extension functions. Then there is an $\mathcal{F}$-generic set A such that the word 1^{k+1} does not occur in the characteristic sequence $\chi(A)$ of A. Hence, in particular, there is an li-k-reg-generic set A such that the word 1^{k+1} does not occur in the characteristic sequence $\chi(A)$ of A.*

PROOF. We construct a set A with the required properties by a finite extension argument. Fix an enumeration $\{f_n : n \geq 0\}$ of the class $\mathcal{F}$, i.e., any function f_n is a partial length invariant function of type $f : \Sigma^* \to \Sigma^k$. Then in order to make A $\mathcal{F}$-generic it suffices to meet the requirements

$$R_n : \; f_n \text{ dense along } A \Rightarrow \exists\, m\, (f_n(\alpha \upharpoonright m)\downarrow \;\&\; (\alpha \upharpoonright m) f_n(\alpha \upharpoonright m) \sqsubset \alpha)$$

for $n \geq 0$ where α denotes the characteristic sequence of A. In fact since, by length invariance of f_n, $f_n(\alpha \upharpoonright m) = f(0^m)$ we may restate requirement R_n as follows.

$$R_n : \; \exists^\infty m\, (f_n(0^m)\downarrow) \Rightarrow \exists\, m\, (f_n(0^m)\downarrow \;\&\; (\alpha \upharpoonright m) f_n(0^m) \sqsubset \alpha)$$

Simultaneously with A we define an increasing function $l : \mathbb{N} \to \mathbb{N}$ where $l(s)$ and $A_s = A \upharpoonright l(s)$ are defined at stage s of the construction ($l(-1) = 1$ and $A_{-1} = \emptyset$). At stage s of the construction we will ensure that requirement R_s is met. In addition we will guarantee that 1^{k+1} does not occur in α. For the latter, we will ensure that, for any $s \geq 0$, the finite characteristic string α_{s-1} of A_{s-1} (i.e., $\alpha_{s-1} =$

$A(0)\dots A(l(s-1)-1))$ ends with a 0 and that the extension α_s of α_{s-1} contains at most k additional occurences of the letter 1.

Now stage s of the construction is as follows. Given $l(s-1)$ and $A_{s-1} = A \upharpoonright l(s-1)$ distinguish the following two cases. If there is a number $m \geq l(s-1)$ such that $f_s(0^m)$ is defined then, for the least such m let $l(s) = m+k+1$ and define A_{s-1} by letting $\alpha_s = \alpha_{s-1}0^{m-l(s-1)} f_s(0^m)0$. Otherwise, let $l(s) = l(s-1)+1$ and define A_{s-1} by letting $\alpha_s = \alpha_{s-1}0$.

As one can easily check, the definition of A_s ensures that A meets f_s if $f_s(0^m)$ is defined for infinitely many numbers m. So all requirements are met, hence A is $\mathcal{F}$-generic. Moreover, since $|f_s(0^m)| = k$ (if $f_s(0^m)$ is defined) the construction obviously ensures that 1^{k+1} does not occur in α. □

The preceding lemma in particular shows that there are li-k-reg-generic sets which are not $(k+1)$-1-saturated. By Lemma 4.26 this implies the strictness of the (weak) li-k-reg-genericity hierarchy.

Theorem 4.29 *For any $k \geq 1$ there is an li-k-reg-generic set A which is not weakly li-$(k+1)$-reg-generic.*

Next we will turn to the relations between length invariant and standard bounded reg-genericity of fixed norm. We first observe a coincidence on level 1.

Lemma 4.30 *The following are equivalent.*

(i) A is weakly li-1-reg-generic.

(ii) A is weakly 1-reg-generic.

(iii) $\chi(A)$ is not regular, i.e., not almost periodic.

PROOF. By Theorem 4.15 and by (4.9) it suffices to show that for any given weakly li-1-reg-generic set A the characteristic sequence α of A is not almost periodic. For a contradiction assume that α is almost periodic, say $\alpha = vw^\omega$. Let $p = |v|$ and $q = |w|$ and define the total 1-bounded extension function f by letting $f(x) = 1 - v(|x|)$ if $|x| < p$ and $f(x) = 1 - v(m)$ if $|x| \geq p$ and $|x| - p = m \mod q$. Then f is regular and length invariant. Moreover, $f(\alpha \upharpoonright n) = 1 - \alpha(n)$ for all $n \geq 0$ whence A does not meet f. But this contradicts the assumption that A is weakly li-1-reg-generic. □

For $k \geq 2$ we do not encounter any equivalences as in Lemma 4.30 but get the following two independence results.

Lemma 4.31 *Let $k \geq 1$. There is a weakly k-reg-generic set A which is not li-1-reg-generic.*

PROOF. We first observe that for any li-1-reg-generic set A there is a number n such that $\chi(A)((k+1)n) = 1$. This follows from 1) the fact that any set A has this property if A meets the partial 1-bounded extension function f defined by $f(x) = 1$ if $|x| = 0 \mod (k+1)$ and $f(x) \uparrow$ otherwise and 2) the fact that this function f is regular, length invariant and dense along any set.

So it suffices to construct a weakly k-reg-generic set A such that

$$\forall\, n \geq 0\ (\chi(A)((k+1)n) = 0). \tag{4.10}$$

Fix an enumeration $\{f_n : n \geq 1\}$ of the total regular 1-bounded extension functions and let A be defined by

$$\chi(A) = 0x_0 0x_1 0x_2 0x_3 0... \tag{4.11}$$

where x_n is inductively defined by $x_0 = 0^k$ and

$$x_n = f_n(0x_0 0x_1 0...x_{n-1}0)$$

for $n \geq 1$. Then, by construction, A is weakly k-reg-generic. On the other hand, since $|x_n| = k$ for $n \geq 0$, (4.10) follows from (4.11) whence A is not li-1-reg-generic.
□

For later use we will state the next lemma not only for length invariant finite-state genericity but for length invariant genericity related to any countable class.

Lemma 4.32 *Let $k \geq 1$ and let $\mathcal{F}$ be any countable set of partial length invariant k-bounded extension functions. Then there is an $\mathcal{F}$-generic set A such that A is not weakly 2-reg-generic. In particular, there is an li-k-reg-generic set A such that A is not weakly 2-reg-generic.*

PROOF. We will construct a set A with the required properties by a finite extension argument. We let α be the characteristic sequence of A, denote the initial segment of A determined by the end of stage s of the construction by A_s, let $l(s)$ be the length of this initial segment, i.e., $A_s = A \upharpoonright l(s)$, and denote the initial segment of α corresponding to A_s by α_s, i.e. $\alpha_s = A(0)...A(l(s)-1)$. Moreover, by convention, $A_{-1} = \emptyset$, $l(-1) = 0$ and $\alpha_{-1} = \varepsilon$.

In order to make A $\mathcal{F}$-generic we will basically use the standard approach. Given an enumeration $\{\hat{f}_n : n \geq 0\}$ of the partial functions of type $\hat{f} : \{0\}^* \to \Sigma^k$ corresponding to an enumeration $\{f_n : n \geq 0\}$ of $\mathcal{F}$ (see the paragraph following Definition 4.22) we will ensure that the requirements

$$R_n : \hat{f}_n \text{ dense} \Rightarrow A \text{ meets } \hat{f}_n$$

are met ($n \geq 0$). This will be sufficient by an obvious generalization of Proposition 4.23.

As usual, at stage s of the construction we take action to ensure that requirement R_s is met. Given $l(s-1)$, $A_s = A \upharpoonright l(s-1)$ and $\alpha_{s-1} = A(0)...A(l(s-1)-1)$ this is achieved as follows. If there is a number $m > l(s-1)$ such that $\hat{f}_s(0^m)$ is defined then we define $l(s)$ and A_s by letting $l(s) = m+k$ and $\alpha_s = \alpha_{s-1}\beta_s\hat{f}_s(0^m)$ for such an m where β_s can be any string of length $m - l(s-1)$. If there is no such number m then we let $l(s) = l(s-1)$ and $A_s = A_{s-1}$. Note that in the latter case $\hat{f}_s$ is not dense whence R_s is trivially met, while in the former case the construction ensures that A meets $\hat{f}_s$ at m. So in either case R_s is met.

Our first goal of making A $\mathcal{F}$-generic is complemented by the second goal of making sure that A is not weakly 2-reg-generic. Here we have to show that there is a total regular 2-bounded extension function $f : \Sigma^* \to \Sigma^2$ which is not met by A, i.e., for which

$$\forall n\,(f(\alpha \upharpoonright n) \neq \alpha(n)\alpha(n+1)) \tag{4.12}$$

holds. Intuitively, we have to show that there is a finite automaton M which on input $\alpha \upharpoonright n$ can rule out one of the four possible values $00, 01, 10, 11$ of the next pair of bits $\alpha(n)\alpha(n+1)$ in α. For this sake we will make sure that in the extension $\alpha_s = \alpha_{s-1}\beta_s\hat{f}_s(0^m)$ of α_{s-1} the string β_s is chosen so that it encodes information on the final part $\hat{f}_s(0^m)$ of α_s. Note that $|\hat{f}_s(0^m)| = k$. So, given the 2^k binary strings z^k_0, ..., $z^k_{2^k-1}$ of length k in lexicographical order, $\hat{f}_s(0^m) = z^k_p$ for some number $p < 2^k$, whence it suffices to code p into β_s. Also note that the length of β_s depends on m. So in order to make sure that β_s provides enough space for coding p, in general we will not take the least $m > l(s-1)$ such that $\hat{f}_s(0^m)$ is defined but will impose some higher lower bound on m. Since action for meeting requirement R_s has only to be taken if $\hat{f}_s(0^m)$ is defined for infinitely many m this will not interfere with our strategy for making A $\mathcal{F}$-generic.

We now formally describe stage s of the construction of A. Given $l(s-1)$, $A_s = A \upharpoonright l(s-1)$ and $\alpha_{s-1} = A(0)...A(l(s-1)-1)$, distinguish the following two cases. If there is a number $m \geq l(s-1)+2k+2^{k+1}+6$ such that $\hat{f}_s(0^m)$ is defined fix the least number m with these properties, let $l(s) = m+k$ and define A_s and α_s by letting

$$\alpha_s = \alpha_{s-1}\beta_s\gamma_s$$

where, for the unique $p < 2^k$ with $z^k_p = \hat{f}_s(0^m)$ and for $q = m - (l(s-1)+2k+2(p+1)+6)$,

$$\beta_s = 1^q 11(01)^{k+p+1}11 \;\;\&\;\; \gamma_s = \hat{f}_s(0^m) = z^k_p.$$

If there is no such number m then let $l(s) = l(s-1)$, $A_s = A_{s-1}$ and $\alpha_s = \alpha_{s-1}$. This completes the construction.

In order to show that the constructed set A has the requested properties, first, by a straightforward induction, we observe that all requirements are met, whence A is

$\mathcal{F}$-generic. (Note that in the first case of the construction above, $\alpha_s = \alpha_{s-1}\beta_s \hat{f}_s(0^m)$ for some m such that $\hat{f}_s(0^m)$ is defined and $|\alpha_{s-1}\beta_s| = m$ whence A meets $\hat{f}_s$ at m.)

It remains to show that A is not weakly 2-reg-generic, i.e., to show that there is a regular total function $f : \Sigma^* \to \Sigma^2$ such that (4.12) holds.

For defining such a function f we start with some observations. First note that there are stages $s_0 < s_1 < s_2 \ldots$ and numbers $p_n < 2^k$ and $q_n \geq 0$ ($n \geq 0$) such that

$$\begin{aligned} \alpha &= \beta_{s_0}\gamma_{s_0}\beta_{s_1}\gamma_{s_1}\beta_{s_2}\gamma_{s_2}\ldots \\ &= 1^{q_0}11(01)^{k+p_0+1}11z^k_{p_0}1^{q_1}11(01)^{k+p_1+1}11z^k_{p_1}1^{q_2}11(01)^{k+p_2+1}11z^k_{p_2}\ldots \end{aligned} \tag{4.13}$$

Note that the only occurrences of two consecutive zeroes 00 in α can occur in the subwords $\gamma_s = z^k_p$ and that any such subword z^k_p is followed by the word 11 and preceded by the word $11(01)^{2^k+p+1}11$. We call a string w a p-string of rank r ($p < 2^k$, $r < k$) if there are strings x and y such that $|y| = r$ and

$$w = x11(01)^{k+p+1}11y.$$

Then, by (4.13), for any initial segment $\alpha \upharpoonright n$ of α, $\alpha \upharpoonright n$ is a p-string of rank 0 if and only if there is some s such that $\alpha \upharpoonright n = \alpha_{s-1}\beta_s$ where $\gamma_s = z^k_p$. It follows that

$$\alpha \upharpoonright n \ p\text{-string of rank } r \ \Rightarrow \alpha(n)\alpha(n+1) = z^k_p(r)z^k_p(r+1)$$

and

$$\forall\, p < 2^k\ \forall\, r < k\ (\alpha \upharpoonright n \text{ is not a } p\text{-string of rank } r\ ) \Rightarrow \alpha(n)\alpha(n+1) \neq 00.$$

So A does not meet the 2-bounded function extension function f defined by

$$f(w) = \begin{cases} (1-z^k_p(r))(1-z^k_p(r+1)) & \text{if } w \text{ is a } p\text{-string of rank } r \\ 00 & \text{otherwise.} \end{cases}$$

Moreover, a finite automaton can recognize whether a string w is a p-string of rank r (and, if so, can store p, r, and z^k_p in its state). So f is regular.

This completes the proof. □

By combining the above results, we can now completely determine the relations among the various standard and length invariant bounded finite-state genericity concepts. Note that, by (4.6), in case of the standard notions it suffices to consider weak genericity.

Theorem 4.33 *For $k \geq 2$ the following and - up to transitive closure - only the following implications hold among the (weak) length invariant bounded regular genericity concepts and the weak general bounded regular genericity concepts.*

$$\begin{array}{ccccc}
\textit{li-}\omega\textit{-reg-generic} & \Leftrightarrow & \textit{weakly li-}\omega\textit{-reg-generic} & \Leftrightarrow & \textit{weakly }\omega\textit{-reg-generic} \\
\Downarrow & & \Downarrow & & \Downarrow \\
\textit{li-}(k+1)\textit{-reg-generic} & \Rightarrow & \textit{weakly li-}(k+1)\textit{-reg-generic} & \Leftarrow & \textit{weakly }(k+1)\textit{-reg-generic} \\
\Downarrow & & \Downarrow & & \Downarrow \\
\textit{li-}k\textit{-reg-generic} & \Rightarrow & \textit{weakly li-}k\textit{-reg-generic} & \Leftarrow & \textit{weakly }k\textit{-reg-generic} \\
\Downarrow & & \Downarrow & & \Downarrow \\
\textit{li-}1\textit{-reg-generic} & \Rightarrow & \textit{weakly li-}1\textit{-reg-generic} & \Leftrightarrow & \textit{weakly }1\textit{-reg-generic}
\end{array} \tag{4.14}$$

PROOF. Correctness of the stated implications follows from (4.9) together with Theorem 4.27 and Lemma 4.30. The fact that only the indicated implications are valid in general is established as follows where it suffices to consider the concepts in lines 2 - 4.

First we observe that no concept on a lower level implies any concept on a higher level. This follows from the saturation properties of the considered genericity concepts. By Lemma 4.26, any weakly li-k-reg-generic set A (hence any weakly k-reg-generic set and any li-k-reg-generic set) is k-ω-saturated, but by Lemma 4.28 and Theorem 4.9 there are li-k-reg-generic sets and weakly k-reg-generic sets (hence weakly li-k-reg-generic sets) which are not $(k+1)$-1-saturated.

It remains to show that none of the concepts in column 1 implies any of the concepts in column 3 with the exception of weak 1-reg-genericity, and that, conversely, none of the concepts in column 3 implies any of the concepts in column 1. But this is immediate by Lemma 4.32 and Lemma 4.31, respectively. □

By the coincidence of weak ω-reg-genericity with saturation, Theorem 4.33 shows that (weakly) li-ω-reg-generic sets are saturated, hence not regular. An interesting question on the power of length invariant finite-state genericity left open by the above theorem is the question, whether, for fixed $k \geq 1$, (weakly) li-k-reg-generic sets are non-regular. We will conclude this subsection by giving a negative answer to this question. Before considering the general case, we will present the case of $k = 1$.

Lemma 4.34 *The set* $0\Sigma^* = \{0w : w \in \Sigma^*\}$ *is li-1-reg-generic.*

PROOF. Let $A = 0\Sigma^*$ and let $f : \Sigma^* \to \Sigma$ be a regular partial length invariant 1-bounded extension function which is dense along A. It suffices to show that A

meets f. By length invariance of f, density of f along A implies that

$$\exists^{\infty} n\ (f(0^n)\downarrow) \tag{4.15}$$

and in order to show that A meets f it suffices to show that

$$\exists n\ (f(0^n)\downarrow\ \&\ A(z_n) = f(0^n)). \tag{4.16}$$

Let $M = (\Sigma, S, \delta, s_0, F, \lambda)$ be a 1-labelled finite automaton which computes f. Since $f(0^n) = \lambda(\delta^*(s_0, 0^n))$ if $\delta^*(s_0, 0^n) \in F$ and $f(0^n)\uparrow$ otherwise, by (4.15) we may fix a state $s \in F$ such that

$$\exists^{\infty} n\ (\delta^*(s_0, 0^n) = s). \tag{4.17}$$

So, for the least n_0 and least $n_1 > n_0$ such that (4.17) holds for n_0 and n_1 in place of n and for $p = n_0$ and $q = n_1 - n_0$,

$$\forall n\ (\delta^*(s_0, 0^{p+nq}) = s).$$

It follows that, for $i = \lambda(s)$,

$$\forall n\ (f(0^{p+nq}) = i).$$

So, in order to satisfy (4.16), it suffices to show

$$\exists n\ (A(z_{p+nq}) = i). \tag{4.18}$$

For a proof of (4.18), by symmetry, w.l.o.g. we may assume that $i = 0$. Fix k such that $2^k > p + q$ and consider the sequence $1z_0^k, \ldots, 1z_{2^k-1}^k$ of the 2^k words of length $k+1$ in $1\Sigma^k$. Note that these words are consecutive words with respect to the length-lexicographical ordering, i.e.

$$1z_0^k, \ldots, 1z_{2^k-1}^k = z_r, \ldots, z_{r+2^k-1}$$

for some number r. By choice of k this implies that $1z_j^k = z_{p+nq}$ for some $j < 2^k - 1$ and $n \geq 0$. Since, by definition of A, $1z_j^k \notin A$, it follows that $A(z_{p+nq}) = A(1z_j^k) = 0 = i$. So (4.18) holds. This completes the proof. □

Lemma 4.34 in particular shows that there is a regular li-1-reg-generic set. By refining the proof of this lemma, we can extend this observation to li-k-reg-generic sets for any $k \geq 1$.

Theorem 4.35 *For any $k \geq 1$ there is a regular li-k-reg-generic set.*

For the proof of this theorem we will need the following observation.

Proposition 4.36 *Let $p \geq 0$ and $k' \geq 1$ be given and let $k = 2^{k'}$. There is a number $r < k$ such that*

$$\forall m > max(p,k')\ \forall n\ \forall s < 2^m\ (z_s^m = z_{p+nk} \Rightarrow r = s\ mod\ k). \tag{4.19}$$

PROOF. Let $m_0 = max(p,k')$. Since there are 2^{m_0} strings of length m_0 and since $p,k \leq 2^{m_0}$ there are numbers n_0 and $s_0 < 2^{m_0}$ such that $z_{s_0}^{m_0} = z_{p+n_0k}$. Fix the least such numbers and let r be the unique number $< k$ such that $r = s_0 \bmod k$. Then it suffices to show that for all numbers $n \geq n_0$, m, and $s < 2^m$ the matrix of (4.19) holds. We proceed by induction on $n \geq n_0$, where for $n = n_0$ the claim is immediate by choice of s_0 and definition of r. For the inductive step we have to establish the claim for $n+1 > n_0$ assuming the claim for n. So fix m, $s < 2^m$, m', and $s' < 2^{m'}$ such that $z_{p+nk} = z_s^m$ and $z_{p+(n+1)k} = z_{s'}^{m'}$ hold. By inductive hypothesis, $r = s \bmod k$. To show that $r = s' \bmod k$ we distinguish the following two cases. If $m = m'$ then $s' = s+k$. It follows that $s' \bmod k = s \bmod k = r$. Otherwise, $m' = m+1$. Moreover, since there are 2^m words of length m, $s+k = 2^m + s'$. Since $k = 2^{k'}$ is a factor of 2^m, it follows that

$$s' \bmod k = (2^m + s') \bmod k = (s+k) \bmod k = s \bmod k = r.$$

This completes the proof. □

PROOF OF THEOREM 4.35. Fix $k \geq 1$ where, by (4.7), w.l.o.g. we may assume that $k = 2^{k'}$ for some number $k' \geq 1$. Define A by specifying the slices $A^{=m} = A \cap \Sigma^m$ of A as follows. For $m < 2^k$ let $A^{=m} = \emptyset$ while, for $m \geq 2^k$,

$$m = i \bmod 2^k\ (0 \leq i < 2^k)\ \Rightarrow\ A(z_0^m)\ldots A(z_{2^m-1}^m) = (z_i^k)^{2^{m-k'}} \tag{4.20}$$

Note that, for a word z of length $m \geq 2^k$, membership of z in A depends only $m \bmod 2^k$ and the last k bits of z. This easily implies that A is regular.

It remains to show that A is li-k-reg-generic. So let $f : \Sigma^* \to \Sigma^k$ be a regular partial length invariant k-bounded extension function which is dense along A. It suffices to show that A meets f. To show this, as in the proof of Lemma 4.34 we can argue that there are numbers $p \geq 0$ and $q \geq 1$ and a word z_i^k of length k such that

$$\forall n\ (f(0^{p+nq}) = z_i^k) \tag{4.21}$$

whence, by length invariance of f, it suffices to show that

$$\exists n\ (A(z_{p+nq})\ldots A(z_{p+nq+k-1}) = z_i^k). \tag{4.22}$$

For a proof of (4.22) fix $r < k$ as in Proposition 4.36 and let m_0 be the least number m such that $m_0 > max(p,k')$ and $2^{m_0} > kq + k$. Then, for any $m \geq m_0$, we may fix numbers s_m and n_m such that

$$s_m < 2^m - k\ \&\ z_s^m = z_{p+n_m \cdot k \cdot q}. \tag{4.23}$$

Note that, by Proposition 4.36,

$$r = s_m \bmod k. \tag{4.24}$$

Finally, fix $j < 2^k$ such that

$$z_j^k(r)\, z_j^k(r+1)\ \ldots\ z_j^k(k)\, z_j^k(k+1)\ \ldots\ z_j^k(k+r-1) = z_i^k \tag{4.25}$$

and choose $m \geq m_0$ minimal such that $m > 2^k$ and $m = j \bmod 2^k$. Then, by (4.20) and (4.25),

$$A(z_0^m)\ldots A(z_{2^m-1}^m) = (z_j^k)^{2^{m-k'}} = z_j^k(0)\ ...\ z_j^k(r-1)\ (z_i^k)^{2^{m-k'}-1}\ z_j^k(r)\ ...\ z_j^k(k-1).$$

Obviously, this implies (for any number s)

$$s < 2^m - k\ \ \&\ \ r = s \bmod k \Rightarrow A(z_s^m)...A(z_{s+k-1}^m) = z_i^k.$$

Hence, by (4.23) and (4.24),

$$A(z_{p+n_m\cdot k\cdot q})\ldots A(z_{p+n_m\cdot k\cdot q+k-1}) = A(z_s^m)\ldots A(z_{s+k-1}^m) = z_i^k.$$

So (4.22) holds for $n = n_m \cdot k$. This completes the proof. □

4.2.2 Oblivious Extension Functions

Next we consider bounded finite-state genericity concepts based on extension functions which, for a constant $m \geq 1$, remember only the last m bits of the initial segments given to them as inputs. We first formalize this concept by introducing oblivious extension functions.

Definition 4.37 Let $f : \Sigma^* \to \Sigma^k$ be a (partial) k-bounded extension function. f is *m-oblivious* if $f(wx) = f(w'x)$ for all words w, w', and x with $|x| = m$. A set G is *$[m,k]$-reg-generic* if it meets all regular partial k-bounded extension functions which are m-oblivious and dense along G; and G is *weakly $[m,k]$-reg-generic* if G meets all regular total k-bounded extension functions which are m-oblivious. G is *(weakly) $[m,\omega]$-reg-generic* if G is (weakly) $[m,k]$-reg-generic for all $k \geq 1$; G is *(weakly) $[\omega,k]$-reg-generic* if G is (weakly) $[m,k]$-reg-generic for all $m \geq 1$; and G is *(weakly) $[\omega,\omega]$-reg-generic* if G is (weakly) $[m,k]$-reg-generic for all $m,k \geq 1$.

Note that any (partial) m-oblivious k-bounded extension function is regular. Alternatively we can describe an m-oblivious k-bounded extension function f by a function $\hat{f} : \Sigma^m \to \Sigma^k$. We say that such a function $\hat{f}$ is *dense along a set A* if $\hat{f}(x)$ is defined for some word x of length m such that $(A \restriction n)x \sqsubset \chi(A)$ for infinitely many numbers n, and we say that *A meets $\hat{f}$ at some number n* if $n \geq m$ and, for the unique strings $x \in \Sigma^m$ and $y \in \Sigma^*$ such that $A \restriction n = yx$, $\hat{f}(x)\downarrow$ and $(A \restriction n)\hat{f}(x) \sqsubset \chi(A)$. Finally, we say that *A meets $\hat{f}$* if A meets $\hat{f}$ at some n.

Proposition 4.38 *A set G is weakly $[m,k]$-reg-generic if and only if G meets all total functions $\hat{f}:\Sigma^m \to \Sigma^k$. G is $[m,k]$-reg-generic if and only if G meets all partial functions $\hat{f}:\Sigma^m \to \Sigma^k$ which are dense along G.*

PROOF. We prove the second part of the proposition. The proof of the first part is similar.

First assume that G is $[m,k]$-reg-generic and that the partial functions $\hat{f}:\Sigma^m \to \Sigma^k$ is dense along G. We have to show that G meets $\hat{f}$ at some n. Consider the partial k-bounded extension function f defined by $f(yx) = \hat{f}(x)$ for all strings $y \in \Sigma^*$ and $x \in \Sigma^m$ and $f(z)\uparrow$ for all strings $z \in \Sigma^{<m}$. Then f is m-oblivious. Moreover, density of $\hat{f}$ along G implies that f is dense along G. So, by $[m,k]$-reg-genericity of G, G meets f at some number n. By definition of f it follows that $n \geq m$ and G meets $\hat{f}$ at n.

Now assume that G meets all partial functions $\hat{f}:\Sigma^m \to \Sigma^k$ which are dense along G and assume that f is an m-oblivious k-bounded extension function which is dense along G. We have to show that G meets f at some number n. Define $\hat{f}:\Sigma^m \to \Sigma^k$ by letting $\hat{f}(x) = f(x)$ for all $x \in \Sigma^m$. Then $\hat{f}$ is dense along G whence, by assumption, there is a number n such that G meets $\hat{f}$ at n. By definition of $\hat{f}$ this implies that G meets f at n. □

As Proposition 4.38 shows, for fixed numbers $m,k \geq 1$, (weak) $[m,k]$-reg-genericity is rather a Boolean genericity concept than a regular genericity concept. This is also demonstrated by the following observations: In contrast to the previously introduced regular genericity concepts there are finite generic sets of these types and a generic set may meet an extension function it has to meet just once not infinitely often.

Lemma 4.39 *Let $m,k \geq 1$. There is a finite $[m,k]$-reg-generic set A. Moreover, A can be chosen such that A meets the total m-oblivious k-bounded extension function f defined by $f(x) = 1^k$ (for $x \in \Sigma^*$) just once.*

PROOF. Consider the sequence $\alpha = \beta_{[m,k]}0^\omega$ where

$$\beta_{[m,k]} = 0^m z_0^k 0^m z_1^k \ldots 0^m z^k \tag{4.26}$$

and let A be the set corresponding to α. Obviously, A is finite and the string 1^k occurs in α only once. It remains to show that A is $[m,k]$-reg-generic. Given a partial function $\hat{f}:\Sigma^m \to \Sigma^k$ which is dense along A, by Proposition 4.38, it suffices to show that A meets $\hat{f}$, i.e., that there is a string x of length m such that $\hat{f}(x)$ is defined and $x\hat{f}(x)$ occurs in α. Since 0^m is the only string of length m which occurs in α infinitely often, density of $\hat{f}$ along A implies that $\hat{f}(0^m)$ is defined (and has length k), say $\hat{f}(0^m) = z_p^k$ where $p < 2^k$. So, for $x = 0^m$, $x\hat{f}(x) = 0^m z_p^k$ occurs in $\beta_{[m,k]}$ hence in α. □

We next present some saturation properties of the (weakly) $[m,k]$-reg-generic sets and some relations among these concepts. The preceding lemma already implicitly gives the basic saturation properties of (weakly) $[m,k]$-reg-generic sets which we explicitly state in the following lemma.

Lemma 4.40 *Let $m,k \geq 1$. Any weakly $[m,k]$-reg-generic set is k-1-saturated but there is an $[m,k]$-reg-generic set A which is not k-2-saturated, hence not $(k+1)$-1-saturated.*

PROOF. The $[m,k]$-reg-generic set A of Lemma 4.39 is not k-2-saturated. On the other hand, if A' is weakly $[m,k]$-reg-generic then, by Proposition 4.38, A' meets any total function $\hat{f} : \Sigma^m \to \Sigma^k$. In particular, given $y \in \Sigma^k$, A' meets the constant function $\hat{f}(x) = y$ (for all $x \in \Sigma^m$) whence y occurs in the characteristic sequence of A'. So A' is k-1-saturated. □

For weak genericity we can improve Lemma 4.40 by the following combinatorial characterization of the weakly $[m,k]$-reg-generic sets.

Lemma 4.41 *Let A be a language, let α be the characteristic sequence of A and let m and k be any numbers ≥ 1. The following are equivalent.*

(i) A is weakly $[m,k]$-reg-generic.

(ii) There is a string $x \in \Sigma^m$ such that xy occurs in α for all strings $y \in \Sigma^k$.

PROOF. The proof of the implication $(i) \Rightarrow (ii)$ is by contraposition. Assume that (ii) fails. For any $x \in \Sigma^m$ fix $y_x \in \Sigma^k$ minimal such that xy_x does not occur in α. Define $\hat{f} : \Sigma^m \to \Sigma^k$ by $\hat{f}(x) = y_x$. Then A does not meet A. Hence, by Proposition 4.38, A is not weakly $[m,k]$-reg-generic whence (i) fails.

For a proof of the implication $(ii) \Rightarrow (i)$ fix $x \in \Sigma^m$ such that xy occurs in α for all $y \in \Sigma^k$and let $\hat{f}$ be a total function $\hat{f} : \Sigma^m \to \Sigma^k$. By Proposition 4.38 it suffices to show that A meets $\hat{f}$. Since $|\hat{f}(x)| = k$, $x\hat{f}(x)$ occurs in α (by choice of x). So there is a number n such that $(\alpha \upharpoonright n)x\hat{f}(x) \sqsubset \alpha$. Hence A meets $\hat{f}$ at $n+m$. □

As the next lemma shows a sufficient level of saturation suffices for guaranteeing $[m,k]$-reg-genericity.

Lemma 4.42 *Let $k,m,p \geq 1$ be given such that $m+k \leq p$. Then any p-1-saturated set A is $[m,k]$-reg-generic.*

PROOF. Assume that the characteristic sequence α of A is p-1-saturated and assume that the partial function $\hat{f} : \Sigma^m \to \Sigma^k$ is dense along A. By Proposition 4.38 it suffices to show that A meets $\hat{f}$, i.e., that there is a string x of length m such that $\hat{f}(x)$ is defined and $x\hat{f}(x)$ occurs in α. By density of $\hat{f}$ along A we may fix $x \in \Sigma^m$ such that $\hat{f}(x)$ is defined. Then $x\hat{f}(x) \in \Sigma^{m+k}$ whence, by $m+k \leq p$ and p-1-saturation of α, $x\hat{f}(x)$ occurs in α. □

The above relations among the oblivious genericity notions and levels of saturation show that these concepts are intertwined as follows.

$$\begin{array}{rcl} (m+k)\text{-1-saturated} & \Rightarrow & [m,k]\text{-reg-generic} \\ & \Rightarrow & \text{weakly } [m,k]\text{-reg-generic} \\ & \Rightarrow & k\text{-1-saturated.} \end{array}$$

We now leave the Boolean type oblivious finite-state genericity notions, i.e., (weak) $[m,k]$-reg-genericity where $m,k \in \mathbb{N}$ and turn to the more powerful concepts where one of these parameters is unbounded (i.e., $m = \omega$ or $k = \omega$) and show how these concepts are related to the standard bounded finite-state genericity concepts. For this sake we analyse the saturation properties of these concepts.

Lemma 4.43 *(a) Any (weakly) $[m,\omega]$-reg-generic set A is saturated ($m \geq 1$).*

(b) Any $[\omega,k]$-reg-generic set A is saturated ($k \geq 1$).

(c) For any number $k \geq 1$ there is a weakly $[\omega,k]$-reg-generic set A which is not $(k+1)$-1-saturated.

PROOF. Part (a) is immediate by the first part of Lemma 4.40.

For a proof of part (b) assume that A is $[\omega,k]$-reg-generic and let α be the characteristic sequence of A. It suffices to show that any word x occurs in α infinitely often. We proceed by induction on the length of x. For $|x| = 0$ the claim is trivial. So assume that $|x| > 0$, say $|x| = m+1$ and $x = x'a$ where $x' \in \Sigma^m$ and $a \in \Sigma$. By inductive hypothesis, x' occurs in α infinitely often. So the partial function $\hat{f} : \Sigma^m \to \Sigma^k$ defined by $\hat{f}(x') = a^k$ and $\hat{f}(y)\uparrow$ for $y \neq x'$ is dense along A. Hence, by $[m,k]$-reg-genericity of A and by Proposition 4.38, A meets $\hat{f}$ at some n. By definition of $\hat{f}$ this implies that $x'a^k$ - hence x - occurs in α.

Finally, for a proof of part (c) consider the sequence

$$\alpha = \beta_{[1,k]}\beta_{[2,k]}\beta_{[3,k]}\beta_{[4,k]}\cdots$$

where $\beta_{[m,k]}$ is defined as in (4.26) and let A be the set corresponding to α. Then $0^m x$ occurs in α for all $m \geq 1$ and all $x \in \Sigma^k$ whence, by Lemma 4.41, A is weakly $[\omega,k]$-reg-generic. On the other hand, however, 1^{k+1} does not occur in α whence A is not $(k+1)$-1-saturated. □

Since the saturated sets and the bounded reg-generic sets coincide, Lemma 4.43 implies the following equivalence theorem.

Theorem 4.44 *For any numbers $k, m \geq 1$ the following are equivalent.*

(i) A is weakly $[m, \omega]$-reg-generic.

(ii) A is $[m, \omega]$-reg-generic.

(iii) A is $[\omega, k]$-reg-generic.

(iv) A is bounded reg-generic.

(v) A is saturated.

PROOF. By coincidence of bounded reg-genericity and saturation (see Theorem 4.7), it suffices to show the implications $(iv) \Rightarrow (x) \Rightarrow (v)$ for $(x) = (i), (ii), (iii)$. But the first implication is immediate by definition while the second implication follows from Lemma 4.43. □

In contrast to the preceding theorem, weak $[\omega, k]$-reg-genericity is weaker than bounded reg-genericity and the strength of weak $[\omega, k]$-reg-genericity depends on the parameter k. In the following theorem we summarize some basic observations on complexity and strength of weak $[\omega, k]$-reg-genericity.

Lemma 4.45 *(a) Any weakly $[\omega, k+1]$-reg-generic set is weakly $[\omega, k]$-reg-generic but there is a weakly $[\omega, k]$-reg-generic set which is not weakly $[\omega, k+1]$-reg-generic.*

(b) A set A is weakly $[\omega, 1]$-reg-generic if and only if $\chi(A)$ is not regular.

(c) For $k \geq 2$, any weakly $[\omega, k]$-reg-generic A has a non-regular characteristic sequence but there are sets with non-regular characteristic sequence which are not weakly $[\omega, k]$-reg-generic.

(d) For $k \geq 1$ there is a weakly $[\omega, k]$-reg-generic set which is regular.

PROOF. The first part of (a) is immediate by definition. So it suffices to show that there is a weakly $[\omega, k]$-reg-generic set which is not weakly $[\omega, k+1]$-reg-generic. Now, by Lemma 4.43, there is a weakly $[\omega, k]$-reg-generic set A which is not $(k+1)$-1-saturated. So, by Lemma 4.40, A is not weakly $[\omega, k+1]$-reg-generic.

For a proof of (b), first assume that A is weakly $[\omega, 1]$-reg-generic and let α be the characteristic sequence of A. Since a sequence is regular if and only if it is almost periodic (see Theorem 2.76), it suffices to show that α is no almost periodic.

For a contradiction assume that α is almost periodic, say $\alpha = vw^{\omega}$ where $|v| = p$ and $|w| = q \geq 1$, and let $m = p + q$. Note that, for any $n \geq p$, $\alpha(n) = \alpha(n+q)$.

Now to get the desired contradiction, define the function $\hat{f} : \Sigma^m \to \Sigma^1$ by letting $\hat{f}(x) = 1 - x(p)$. By $[\omega, 1]$-reg-genericity of A and by Proposition 4.38, A meets $\hat{f}$, i.e., there is a string x of length $m = p + q$ such that $x\hat{f}(x) = x(1 - x(p))$ occurs in α. It follows that there is a number $n \geq p$ such that

$$\begin{gathered}\alpha(n-p)\ldots\alpha(n-1)\alpha(n)\ldots\alpha(n+q-1)\alpha(n+q)\\ =\\ x(0)\ldots x(p-1)x(p)\ldots x(p+q-1)(1-x(p)).\end{gathered}$$

In particular, $\alpha(n) = x(p)$ whereas $\alpha(n+q) = 1 - x(p)$, whence $\alpha(n) \neq \alpha(n+q)$. But this is impossible as shown above.

In order to complete the proof of (b) we have to show that any set A with non-regular characteristic sequence is weakly $[\omega, 1]$-reg-generic. Since, by definition, any weakly 1-reg-generic set is weakly $[\omega, 1]$-reg-generic, this follows from Theorem 4.15.

Part (c) is immediate by parts (a) and (b).

For a proof of (d) fix $k \geq 1$ and let

$$A = \{z_q^p : \exists\, \ell < 2^k\ (p \geq k\ \&\ p = \ell \bmod 2^k\ \&\ q < k\ \&\ z_\ell^k(q) = 1)\}.$$

Note that for each number $p \geq k$ at most the first k strings of length p can be elements of A. Moreover, which of these strings are elements of A is determined by the value ℓ of p modulo 2^k. Namely, the qth string of length p is element of A if the qth bit of the ℓth word of length k is a one, i.e., $A(z_0^p)\ldots A(z_{k-1}^p) = z_\ell^k$.

As one can easily check, A is regular. In order to show that A is weakly $[\omega, k]$-reg-generic, by Proposition 4.38, it suffices to show that for any given number $m \geq 1$ and for any given total function $\hat{f} : \Sigma^m \to \Sigma^k$ there is a string x of length m such that $x\hat{f}(x)$ occurs in the characteristic sequence α of A. So fix such m and $\hat{f}$, let ℓ be the unique number $\ell < 2^k$ such that $\hat{f}(0^m) = z_\ell^k$, and choose $p > k + m$ such that $p = \ell \mod 2^k$. Then, by definition of A, for the last m strings $z^{p-1}_{2^{p-1}-m-1}, \ldots, z^{p-1}_{2^{p-1}-1}$ of length $p - 1$,

$$A(z^{p-1}_{2^{p-1}-m-1})\ldots A(z^{p-1}_{2^{p-1}-1}) = 0^m$$

(since any elements x of A of length $p - 1$ are among the first k strings of length $p - 1$ and, by choice of p there are at least $2^{k+m} > k + m$ strings of length $p - 1$) while for the first k strings $z_0^p, \ldots, z_{k-1}^p$ of length p

$$A(z_0^p)\ldots A(z_{k-1}^p) = z_\ell^k.$$

So $0^m\hat{f}(0^m) = 0^m z_\ell^k$ occurs in α. □

4.3 Cantor-Style Finite-State Diagonalization

In the next section we will introduce some stronger bounded finite-state genericity concepts which are based on extension functions which obtain initial segments in a more redundant presentation as their inputs. This will allow a finite automaton to extract more information from the initial segment than in the case of the standard presentation. In particular these stronger concepts will subsume finite-state Cantor style diagonalizations whence we will look at this type of diagonalizations here first.

If a set A is constructed by a Cantor style diagonalization then the diagonalization step at some string x does not depend on the earlier construction, i.e., on $A \upharpoonright x$, but only on the diagonalization location x. So the individual diagonalization requirements are not described by extension functions but by diagonalization functions $f : \Sigma^* \to \Sigma$ where f is given the place x for the diagonalization as its input and $A(x) = f(x)$ will ensure that the requirement is met by A at the string x. We generalize this concept by also considering diagonalizations requiring to fix A not only on a single string x but on k consecutive strings $x, \dots, x+k-1$ which we will formalize by k-diagonalization functions f of type $f : \Sigma^* \to \Sigma^k$. Here A will meet f at x if $A(x)...A(x+k-1) = f(x)$. Finally, we will also formalize *slow Cantor style diagonalizations*, i.e., those diagonalizations were the diagonalization cannot take place at any string but only at selected places. These more powerful diagonalizations relate to the classical Cantor style diagonalizations just as the wait-and-see arguments relate to the standard finite-extension arguments. A typical example of a slow Cantor style diagonalization is the construction of a bi-immune set described in Section 3.4. Of course this type of diagonalizations is formalized by *partial* diagonalization functions.

Definition 4.46 A (partial) *k-bounded diagonalization function* f, – or a (partial) *k-diagonalization function* for short – is a (partial) function $f : \Sigma^* \to \Sigma^k$. A (partial) 1-diagonalization function is also simply called a (partial) *diagonalization function*. A partial k-diagonalization function f is *dense* if the domain of f is infinite. A set A *meets* a k-diagonalization function f *at* a string x if $f(x)$ is defined and $f(x) = A(x)...A(x+k-1)$, and A *meets* f if A meets f at some string. A is *(weakly) k-C-reg-generic* if A meets every dense (total) regular k-diagonalization function. A is *(weakly) ω-C-reg-generic* if A is (weakly) k-C-reg-generic for all $k \geq 1$.

Note that the following relations among the Cantor style finite-state genericity notions are immediate by definition (where $k \geq 2$; compare with (4.1)).

$$\begin{array}{ccc}
\omega\text{-C-reg-generic} & \Rightarrow & \text{weakly } \omega\text{-C-reg-generic} \\
\Downarrow & & \Downarrow \\
(k+1)\text{-C-reg-generic} & \Rightarrow & \text{weakly } (k+1)\text{-C-reg-generic} \\
\Downarrow & & \Downarrow \\
k\text{-C-reg-generic} & \Rightarrow & \text{weakly } k\text{-C-reg-generic} \\
\Downarrow & & \Downarrow \\
1\text{-C-reg-generic} & \Rightarrow & \text{weakly } 1\text{-C-reg-generic}
\end{array} \tag{4.27}$$

Moreover, by the closure of the class of regular languages under complement, we easily obtain the corresponding closure property for the finite-state Cantor style genericity notions. Furthermore, as in case of the standard bounded finite-genericity concepts, infinitely-often genericity and genericity coincides for the finite-state Cantor style genericity notions too, i.e., a generic set will meet any diagonalization function it has to meet not just once but infinitely often.

Proposition 4.47 *For any (weakly) k-C-reg-generic set A, the complement $\overline{A}$ of A is (weakly) k-C-reg-generic too ($k \geq 1$ or $k = \omega$).*

Proposition 4.48 *Let A be k-C-reg-generic. Then A infinitely often meets every regular partial k-diagonalization function f which is dense, i.e.,*

$$f(z_n) = A(z_n)\ldots A(z_{n+k-1})$$

for infinitely many n. Similarly, any weakly k-C-reg-generic set A meets every regular total k-diagonalization function infinitely often.

We omit the straightforward proofs of the two preceding propositions and turn to a closer analysis of the Cantor style genericity concepts. For this sake it is important to observe the relations between Cantor style diagonalizations and finite extension arguments based on length invariant extension functions.

Note that, formally, k-bounded diagonalization functions and k-bounded extension functions are functions f of the same type, namely $f : \Sigma^* \to \Sigma^k$. For a k-bounded diagonalization function, however, the input x is interpreted as the string at which the diagonalization takes place and $f(x)$ gives us the values the characteristic function c_A of a set A has to assume at x and the $k-1$ following strings, i.e. $c_A(x)...c_A(x+(k-1)) = f(x)$, in order to perform the diagonalization. In particular, the diagonalization action to be taken does not depend on any previous values of A. In contrast, if f is an k-extension function then the input x is interpreted as an initial segment of the set A for which the diagonalization is to be performed. Here, the diagonalization step is carried out at $z_{|x|}$ (not at x), i.e., at the first string whose membership in A is not yet determined by the initial segment x of $\chi(A)$ and the action required may depend on the previous values for A.

Diagonalization functions, however, can be interpreted as length invariant extensions functions and vice versa: Namely, in order to simulate a k-diagonalization function f consider the length invariant k-bounded extension function f' which on any input of length n produces the value $f(z_n)$. Conversely, a length invariant k-bounded extension function f' can be simulated by the k-diagonalization function f defined by $f(z_n) = f(0^n)$.

Definition 4.49 Let f be a partial k-bounded diagonalization function and let f' be a partial length invariant k-bounded extension function. We say that f and f' are *equivalent* or that f' is the *length invariant k-extension function corresponding to* f and f is the *k-diagonalization function corresponding to* f' if, for all numbers $n \geq 0$, $f(z_n)\downarrow$ if and only if $f'(0^n)\downarrow$ and, if defined, $f(z_n) = f'(0^n)$.

Note that, for any partial k-bounded diagonalization function f there is a unique length invariant k-extension function f' corresponding to f and vice versa.

Lemma 4.50 *Let f be a partial k-diagonalization function and let f' be the corresponding length invariant extension function f'. Then f is dense (total) iff f' is dense along all sets (total) and, for any set A, A meets f at z_n iff A meets f' at n.*

PROOF. Straightforward. □

This correspondence between diagonalization functions and length invariant extension functions shows that any Cantor-style genericity concept coincides with a bounded genericity concept based on length invariant extension functions. In particular, we get the following.

Lemma 4.51 *Let $k \geq 1$. There is a countable class $\mathcal{F} = \{f_n : n \geq 1\}$ of length invariant partial k-bounded extension functions such that, for any set A, A is weakly k-C-reg-generic if and only if A is weakly $\mathcal{F}$-generic and A is k-C-reg-generic if and only if A is $\mathcal{F}$-generic.*

PROOF. This follows from Lemma 4.50 by letting $\mathcal{F}$ be the class of the length invariant extension functions f' corresponding to the regular partial k-bounded diagonalization functions f. □

In particular this shows that any bounded finite-state Cantor-style genericity concept is a bounded genericity concept in the sense of Definition 3.34 whence, by Theorem 3.35, the classes of these generic sets have measure 1 and are comeager.

The equivalence of diagonalization functions and length invariant extension functions, however, does not preserve the complexity. In the following we will show that, for a (partial) k-diagonalization function f and the corresponding (partial) length invariant k-extension function f', regularity of f' implies regularity of f but, in general, not vice versa. We may conclude from this that the finite-state Cantor-style genericity concepts are stronger than the corresponding length invariant genericity concepts.

Lemma 4.52 *Let f be any regular partial length invariant k-extension function and let $\tilde{f}$ be the corresponding partial k-bounded diagonalization function. Then $\tilde{f}$ is regular too.*

PROOF. Recall that $\tilde{f}$ is defined by $\tilde{f}(z_n) = f(0^n)$ if $f(0^n)\downarrow$ and $\tilde{f}(z_n)\uparrow$ otherwise. Since, by Lemma 2.46, the class of regular partial functions of type $\Sigma^* \to \Sigma^k$ is closed under finite variants, it suffices to show that there is a finite variant of $\tilde{f}$ which is regular.

Fix a k-labelled finite automaton $M = (\Sigma, S, \delta, s_0, F, \lambda)$ which computes f, i.e., such that $f = f_M$, and let $S = \{s_0, ... s_{p-1}\}$. Then, for $s_{i(n)} = \delta^*(s_0, 0^n)$, $f(0^n) = \lambda(s_{i(n)})$ if $s_{i(n)} \in F$ and $f(0^n)\uparrow$ otherwise. Moreover, the sequence $s_{i(0)}s_{i(1)}s_{i(2)}\dots$ is almost periodic. So we may choose $q \geq 0$ and $r \geq 1$ – where w.l.o.g. $q \leq r$ – such that

$$\forall n \geq q \ (s_{i(n+r)} = s_{i(n)}),$$

hence

$$\forall n \geq q \ (s_{i(n)} = s_{i([n \bmod r]+r)}). \tag{4.28}$$

On the other hand, by Lemma 2.79, there is a deterministic finite automaton $M' = (\Sigma, S', \delta', s'_0)$ (without a distinguished set of final states) such that $S' = \{s'_0, ..., s'_{r-1}\}$ and, for $n \geq 0$,

$$\delta'^*(s'_0, z_n) = s'_{n \bmod r}. \tag{4.29}$$

We can extend M' to a k-labelled finite automaton $M'' = (\Sigma, S', \delta', s'_0, F', \lambda')$ which computes $\tilde{f}$ on all inputs z_n with $n \geq q$ by letting $F' = \{s'_j : s_{i(j+r)} \in F\}$ and $\lambda'(s'_j) = \lambda(s_{i(j+r)})$.

It remains to show that the function $f_{M''}$ computed by M'' coincides with $\tilde{f}$ on all inputs z_n with $n \geq q$, i.e., that $f_{M''}(z_n)\downarrow$ if and only if $f(0^n)\downarrow$ and, if defined, $f_{M''}(z_n) = f(0^n)$. So fix $n \geq q$.

To show that $f_{M''}(z_n)\downarrow$ if and only if $f(0^n)\downarrow$ it suffices to show

$$\delta'^*(s'_0, z_n) \in F' \Leftrightarrow \delta^*(s_0, 0^n) \in F. \tag{4.30}$$

This follows from the following observations: By (4.29),

$$\delta'^*(s'_0, z_n) = s'_{n \bmod r}$$

and, by definition of $s_{i(n)}$ and by (4.28),

$$\delta^*(s_0, 0^n) = s_{i(n)} = s_{i([n \bmod r]+r)}.$$

Since, by definition of F',

$$s'_{n \bmod r} \in F' \Leftrightarrow s_{i([n \bmod r]+r)} \in F,$$

(4.30) follows.

Finally, it remains to show that – assuming $f_{M''}(z_n)\downarrow$ – $f_{M''}(z_n) = f(0^n)$. By (4.30) it suffices to show that

$$\lambda'(\delta'^*(s'_0, z_n)) = \lambda(\delta^*(s_0, 0^n)). \tag{4.31}$$

Since, as observed above,

$$\delta'^*(s'_0, z_n) = s'_{n \bmod r} \ \& \ \delta^*(s_0, 0^n) = s_{i([n \bmod r]+r)},$$

this follows by definition of λ'.

□

Theorem 4.53 *Any (weakly) k-C-reg-generic set A is (weakly) li-k-reg-generic.*

PROOF. This is immediate by Lemma 4.52. □

In order to show that the converse of Lemma 4.52 fails, we next analyze the strength of total and partial regular 1-diagonalization functions. As one might expect, the corresponding genericity concepts coincide with nonregularity and REG-bi-immunity.

Theorem 4.54 *(a) A set A is weakly* 1*-C-reg-generic if and only if* $A \notin$ REG.

(b) A set A is 1*-C-reg-generic if and only if A is* REG*-bi-immune.*

PROOF. *(a)* The proof is by contraposition. First assume that A is regular. In order to show that A is not weakly 1-C-reg-generic it suffices to give a regular total diagonalization function f such that A does not meet f, i.e., such that $A(x) = 1 - f(x)$ for all strings x. By regularity of A and closure of REG under complement, the function f defined by $f(x) = 1 - A(x)$ is regular and has this property. Now assume that A is not weakly 1-C-reg-generic, i.e., that there is a regular diagonalization function f such that $A(x) = 1 - f(x)$ holds for all strings x. We have to show that

A is regular. But this is obvious since the regular function f is the characteristic function of the complement of A and the class of regular languages is closed under complement.

(b) First assume that A is 1-C-reg-generic. We have to show that A is REG-bi-immune. In fact, by Proposition 4.47, it suffices to show that A is REG-co-immune, i.e., that, for any infinite regular set B, $A \cap B$ is not empty. Define the partial diagonalization function f by letting

$$f(x) = \begin{cases} 1 & \text{if } x \in B \\ \uparrow & \text{otherwise.} \end{cases}$$

Then, by infinity of B, f is dense and, by regularity of B, f is regular. So, by 1-C-reg-genericity, A meets f at some string x. Obviously, this implies that $x \in A \cap B$.

For a proof of the other direction assume that A is REG-bi-immune. We have to show that A is 1-C-reg-generic, i.e., that for a given dense regular diagonalization function f there is a string x in the domain of f such that $A(x) = f(x)$. Note that, by density of f, the domain $D(f)$ of f is infinite and, by regularity of f, $D(f)$ is regular. It follows that, for some $i \leq 1$, $\{x : f(x) = i\}$ is infinite and regular. By symmetry, w.l.o.g. we may assume that this is true for $i = 1$. So, by REG-co-immunity of A, $A \cap \{x : f(x) = 1\} \neq \emptyset$, i.e., there is a string $x \in A$ such that $f(x) = 1$. Hence A meets f at x. □

By combining this theorem with some previous results we can establish a number of relations between finite-state Cantor style genericity and some of the previously introduced genericity notions. We first observe that the converse of Theorem 4.53 fails.

Corollary 4.55 *For any $k \geq 1$ there is an li-k-generic set which is not weakly 1-C-reg-generic.*

PROOF. This follows from the first part of Theorem 4.54 since, by Theorem 4.35, there are regular li-k-generic sets. □

Corollary 4.55 immediately implies:

Corollary 4.56 *For any $k \geq 1$ there is a regular k-diagonalization function f such that the corresponding length invariant k-extension function f' is not regular.*

Another consequence of the first part of Theorem 4.54 is the following distinction between weak 1-C-reg-genericity and weak 1-reg-genericity.

Corollary 4.57 *Any weakly 1-C-reg-generic set is weakly 1-reg-generic but there is a weakly 1-reg-generic set which is not weakly 1-C-reg-generic.*

PROOF. By Theorem 4.15, a set A is weakly 1-reg-generic iff the characteristic sequence of A is regular, and in Section 2.5 we have shown that regularity of the characteristic sequence of a set implies regularity of the set but in general not vice versa. □

Similarly, it follows from the second part of Theorem 4.54 that there is a 1-reg-generic set A which is not 1-C-reg-generic: In Section 4.1 we have shown that there are context-free 1-reg-generic sets (Theorem 4.20) whereas in Section 2.4 we have shown that no context-free set is REG-bi-immune (Theorem 2.59). So, by Theorem 4.54, no 1-C-reg-generic is context-free.

Corollary 4.58 *There is a* 1*-reg-generic set which is not* 1*-C-reg-generic.*

Despite this observation the full analog of Corollary 4.57 for 1-reg-genericity fails since 1-C-reg-genericity in general does not imply 1-reg-genericity. In order to show this we will look at the saturation properties of the C-reg-genericity notions. These saturation properties will also help us to decide the relations among the different Cantor style finite-state genericity concepts. We will show first that k-diagonalization functions can force k-saturation but not $(k+1)$-saturation.

Lemma 4.59 *Any weakly k-C-reg-generic set is k-ω-saturated ($k \geq 1$).*

PROOF. By Theorem 4.53, any weakly k-C-reg-generic set is weakly l.i. k-reg-generic and, by Lemma 4.26, any weakly l.i. k-reg-generic set is k-ω-saturated. □

Lemma 4.60 *There is a k-C-reg-generic set A which is not $(k+1)$-1-saturated ($k \geq 1$).*

PROOF. This is immediate by Lemmas 4.51 and 4.28. □

Lemma 4.59 implies that weakly ω-C-reg-generic sets are saturated. As we will show next, the converse is true too.

Theorem 4.61 *A set A is weakly ω-C-reg-generic if and only if A is saturated.*

PROOF. By Lemma 4.59 it suffices to show that any saturated set is ω-C-reg-generic. So assume that A is saturated and fix a total regular k-bounded diagonalization function $f : \Sigma^* \to \Sigma^k$ where w.l.o.g. $k = 2^{k'}$ for some $k' \geq 1$. We have to show that A meets f, i.e., that there is a string z_n such that

$$f(z_n) = A(z_n)\dots A(z_{n+k-1}) \tag{4.32}$$

Fix a k-labelled finite automaton $M = (\Sigma, S, \delta, s_0, F, \lambda)$ which computes f, i.e., such that $f = f_M$, and let $S = \{s_0, ... s_{p-1}\}$ where w.l.o.g. $p = 2^{p'}$ for some $p' \geq 1$.

Define a word x of length $p \cdot k = 2^{p'+k'}$ by letting

$$x = x_0 \cdots x_{p-1}$$

where the words $x_i \in \Sigma^k$ $(0 \leq i < p)$ are defined by

$$x_i = \lambda(\delta^*(s_i, z_i^{p'} z_0^{k'})). \tag{4.33}$$

Then, by saturation of A and by Lemma 2.110 (b), there is a string z_m such that

$$A(z_m z_0^{p'+k'}) \ldots A(z_m z_{pk-1}^{p'+k'}) = x$$

Since

$$A(z_m z_0^{p'+k'}) \ldots A(z_m z_{pk-1}^{p'+k'}) =$$

$$[A(z_m z_0^{p'} z_0^{k'}) \ldots A(z_m z_0^{p'} z_{k-1}^{k'})] \cdots [A(z_m z_{p-1}^{p'} z_0^{k'}) \ldots A(z_m z_{p-1}^{p'} z_{k-1}^{k'})]$$

it follows, by definition of x, that, for $0 \leq i < p$,

$$A(z_m z_i^{p'} z_0^{k'}) \ldots A(z_m z_i^{p'} z_{k-1}^{k'}) = x_i. \tag{4.34}$$

Now fix i such that $\delta^*(s_0, z_m) = s_i$ and let $z_n = z_m z_i^{p'} z_0^{k'}$. Then, by (4.34),

$$A(z_n) \ldots A(z_{n+k-1}) = A(z_m z_i^{p'} z_0^{k'}) \ldots A(z_m z_i^{p'} z_{k-1}^{k'}) = x_i \tag{4.35}$$

while, by choice of M, z_n and i and by (4.33),

$$\begin{aligned} f(z_n) &= \lambda(\delta^*(s_0, z_n)) \\ &= \lambda(\delta^*(s_0, z_m z_i^{p'} z_0^{k'})) \\ &= \lambda(\delta^*(\delta^*(s_0, z_m), z_i^{p'} z_0^{k'})) \\ &= \lambda(\delta^*(s_i, z_i^{p'} z_0^{k'})) \\ &= x_i. \end{aligned}$$

By (4.35) this implies (4.32). □

By combining the above observations we can now show which of the implications in (4.27) are strict and we can further illustrate the power of the various finite-sate Cantor-style genericity concepts by specifying their relations to some fundamental concepts such as nonregularity, saturation and REG-bi-immunity.

Theorem 4.62 *The following and – up to transitive closure – only the following implications hold in general ($k \geq 2$).*

$$\begin{array}{ccc}
 & & \textit{saturated} \\
 & & \Updownarrow \\
\omega\text{-}\textit{C-reg-generic} & \Rightarrow & \textit{weakly } \omega\text{-}\textit{C-reg-generic} \\
\Downarrow & & \Downarrow \\
(k+1)\text{-}\textit{C-reg-generic} & \Rightarrow & \textit{weakly } (k+1)\text{-}\textit{C-reg-generic} \\
\Downarrow & & \Downarrow \\
k\text{-}\textit{C-reg-generic} & \Rightarrow & \textit{weakly } k\text{-}\textit{C-reg-generic} \\
\Downarrow & & \Downarrow \\
1\text{-}\textit{C-reg-generic} & \Rightarrow & \textit{weakly } 1\text{-}\textit{C-reg-generic} \\
\Updownarrow & & \Updownarrow \\
\textit{REG-bi-immune} & & \notin \text{REG}
\end{array} \tag{4.36}$$

PROOF. We first observe that the one-sided implications in (4.36) hold by (4.27) while the three equivalences hold by Theorem 4.54 (a) and (b) and by Theorem 4.61. So it suffice to show that only the given implications hold. This follows from the following two observations.

First, no genericity concept on a lower level implies any of the genericity on a higher level. This follows from the fact that any (weakly) $(k+1)$-C-reg-generic set is $(k+1)$-ω-saturated (Lemma 4.59) but that there are (weakly) k-C-reg-generic sets which are not $(k+1)$-1-saturated (Lemma 4.60).

Second, no genericity concept on the right hand side implies any of the genericity concepts on the left hand side. To show this we first observe that, by the positive relations in (4.36) established above, any saturated set A has all the genericity properties on the right hand side whereas any set with any of the genericity properties of the left hand side is REG-bi-immune. So it suffices to show that there is a saturated set which is not REG-bi-immune. But this has been shown in Theorem 2.117. □

By the coincidence of weak ω-C-reg-genericity with saturation and of 1-C-reg-genericity with REG-bi-immunity, ω-C-reg-genericity implies saturation and REG-bi-immunity. This might lead one to conjecture that the ω-C-reg-generic sets are just the sets with these two properties. But this is not the case as the following lemma shows.

Lemma 4.63 *There is a set A which is both saturated and REG-bi-immune but not 2-C-reg-generic.*

PROOF. We only sketch the proof. It suffices to show that there is a saturated and REG-bi-immune set A such that, for all $n \geq 2$, $|\{0^n, 0^n+1\} \cap A| \leq 1$. The latter implies that A does not meet the partial 2-diagonalization function f where $f(0^n) = 11$ for $n \geq 2$ and $f(x)\uparrow$ otherwise. Obviously f is regular and dense whence we may conclude that A is not 2-C-reg-generic. A set A with the desired properties is constructed by a slow diagonalization: The standard bi-immunity construction as described in Section 3.4 can be easily modified to make A REG-bi-immune and at the same time make sure that for any number n at most one string of length n is put into A and at most one string of length n is restrained from A. So, by the former, we may ensure that $|A \cap 0\Sigma^n| \leq 1$ for all $n \geq 1$ while, by the latter, we can use the part $A \cap 1\Sigma^*$ for making A saturated without interfering with the bi-immunity requirements. □

Since the saturated sets coincide with many of the bounded reg-genericity concepts investigated in the preceding sections – namely, in particular, with bounded reg-genericity (i.e., ω-reg-genericity), k-reg-genericity for any $k \geq 1$, weak ω-reg-genericity, li-ω-reg-genericity, and weak li-ω-reg-genericity – Theorem 4.62 also clarifies the relations between the Cantor-style finite-state genericity concepts with many of the previously discussed genericity notions. In the remainder of this section we will discuss the relations between Cantor-style genericity and those previously considered genericity notions of standard type and of length invariant type which are weaker than saturation.

We first address the question which of the previously introduced genericity notions are implied by (weak) k-C-genericity (for fixed $k \geq 1$).

If we consider the standard bounded reg-genericity notions then the only positive positive results are the ones following from

$$\text{weakly 1-C-reg-generic} \Rightarrow \text{weakly 1-reg-generic} \tag{4.37}$$

by Theorem 4.62. Note that (4.37) holds since the weakly 1-C-reg-generic sets are just the non-regular sets while the weakly 1-reg-generic sets are just the sets with non-regular characteristic sequence. The fact that we do not get any other implications follows from the next lemma.

Lemma 4.64 *Let $k \geq 1$. There is a k-C-reg-generic set A such that A is not weakly 2-reg-generic.*

PROOF. This is immediate by Lemmas 4.32 and 4.51. □

If we consider length invariant genericity in place of standard genericity then, by Theorem 4.53,

$$\text{(weakly) } k\text{-C-reg-generic} \Rightarrow \text{weakly li-}k\text{-reg-generic} \tag{4.38}$$

and

$$k\text{-C-reg-generic} \Rightarrow \text{(weakly) li-}k\text{-reg-generic} \qquad (4.39)$$

hold for $k \geq 1$.

To show that these are the only valid implications we first observe that, by the saturation properties established for the various genericity notions, k-C-reg-genericity does not imply weak li-k'-reg-genericity for any $k' > k$ since there are k-C-reg-generic sets which are not k'-1-saturated whereas every weakly li-k'-reg-generic set has this property. So the optimality of (4.38) and (4.39) follows from the next lemma.

Lemma 4.65 *Let $k \geq 1$. There is a weakly k-C-reg-generic set A such that A is not li-1-reg-generic.*

PROOF. This is shown by a straightforward modification of the proof of Lemma 4.31. □

Now we address the question which of the (weak) k-C-genericity concepts are implied by the previously introduced genericity notions weaker than saturation.

For the length invariant genericity notions we obtain a complete negative answer: By Corollary 4.55, there is no number k such that (weak) li-k-reg-genericity implies any of the Cantor style regular genericity concepts.

For the standard genericity notions the situation is somewhat more complex. Here it suffices to consider weak k-reg-genericity for $k \geq 1$ (since the other concepts coincide with saturation). Since any saturated set is weakly k-reg-generic and since there are saturated sets which are not REG-bi-immune, it follows from (4.36) that weak k-reg-genericity in general does not imply 1-C-reg-genericity. So it only remains to consider the question for which numbers k and k'

$$\text{weakly } k\text{-reg-generic} \Rightarrow \text{weakly } k'\text{-C-reg-generic} \qquad (4.40)$$

holds. Again, by the established partial saturation properties of the considered genericity notions, (4.40) fails for all numbers $k, k' \geq 1$ with $k < k'$. Moreover, (4.40) fails for $k = k' = 1$ by Corollary 4.57.

By the following lemma, (4.40) also fails for $k = k' \geq 2$.

Lemma 4.66 *For $k \geq 2$ there is a weakly k-reg-generic set A which is not weakly k-C-reg-generic.*

PROOF. Given $k \geq 2$, by a finite extension argument we construct a set A such that A is weakly k-reg-generic but not weakly k-C-reg-generic.

Fix a recursive enumeration $\{f_e : e \geq 0\}$ of the total regular k-bounded extension functions. Then, in order to make A weakly k-reg-generic, it suffices to meet the requirements

$$\mathcal{R}_e:\ \exists n\,((\alpha \restriction n)\,f_e(\alpha \restriction n) \sqsubseteq \alpha)$$

where α is the characteristic sequence of A.

In order to ensure that A is not weakly k-C-reg-generic we will ensure that

$$\forall x \in 0\Sigma^*\ (A(x)\ldots A(x+k-1) \neq 1^k) \tag{4.41}$$

and

$$\forall x \in 1\Sigma^*\ (A(x) = 1). \tag{4.42}$$

This will ensure that A does not meet the k-bounded diagonalization function f defined by

$$f(x) = \begin{cases} 1^k & \text{if } x \in 0\Sigma^* \\ 0^k & \text{otherwise.} \end{cases}$$

at any string $x \neq \lambda$. Since, obviously, f is regular, by Proposition 4.48, this will guarantee that A is not weakly k-C-reg-generic.

We now describe the construction of A. At stage s, given an initial segment α_{s-1} of α of the form $\alpha_{s-1} = \alpha \restriction 0^{l(s-1)}$ (where $l(-1) = 0$ and $\alpha_{-1} = \varepsilon$), we define $l(s) > l(s-1)$ and the extension $\alpha_s = \alpha \restriction 0^{l(s)}$ of α_{s-1} in such a way that requirement $\mathcal{R}_s$ will be met and such that the definition of α_s is consistent with (4.41) and (4.42).

For the definition of $l(s)$ and α_s we proceed as follows. Let M be an automaton which computes f_s and let p be the number of states of M. Then fix $m \geq l(s-1)$ minimal such that $p+k < 2^{m-1}$ and set $l(s) = m+1$. For strings x with $l(s-1) \leq |x| < m$ define $A(x)$ by letting $A(x) = 0$ if $x \in 0\Sigma^*$ and by letting $A(x) = 1$ otherwise. Note that this defines $\alpha \restriction m$ in a way consistent with (4.41) and (4.42). For the definition of $A(x)$ for the strings x of length m we distinguish two cases.

First assume that

$$\exists q \leq p\ (f_s((\alpha \restriction m)0^q) \neq 1^k) \tag{4.43}$$

holds. Then, for the least such q let

$$A(z_0^m)\ldots A(z_{2^m-1}^m) = 0^q\ f_s((\alpha \restriction m)0^q)\ 0^{2^{m-1}-(q+k)}\ 1^{2^{m-1}}$$

(note that $q+k \leq p+k < 2^{m-1}$). Note that this is consistent with (4.41) and (4.42) and ensures that $\mathcal{R}_s$ is met.

If (4.43) fails then

$$\forall q \leq p\ (f_s((\alpha \restriction m)0^q) = 1^k).$$

Since the automaton M computing f_s has p states this implies that

$$\forall q \geq 0 \; (f_s((\alpha \restriction m)0^q) = 1^k).$$

So, by letting

$$A(z_0^m)\dots A(z_{2^m-1}^m) = 0^{2^{m-1}} \; 1^{2^{m-1}},$$

we ensure that A meets f_s (since $(\alpha \restriction 10^{m-1}) f_s(\alpha \restriction 10^{m-1}) \sqsubseteq \alpha$) and this definition is consistent with (4.41) and (4.42).

This completes the proof. □

It remains the question whether (4.40) may hold for some numbers k and k' with $k' < k$. We leave this as an open question. We only remark that, by the coincidence of weak 1-C-genericity and nonregularity, the question whether (4.40) holds for $k \geq 2$ and $k' = 1$ is equivalent to the question whether there are regular weakly k-reg-generic sets for $k \geq 2$.

4.4 Enriched Encodings of Initial Segments

We now introduce stronger bounded finite-state genericity concepts which are based on extension functions which obtain initial segments in a more redundant presentation as their inputs. These notions will combine the power of the standard bounded finite-state genericity concepts with that of the Cantor style genericity notions introduced in the preceding section.

Here we consider the following redundant presentation $A \upharpoonright_r z_n$ of the initial segment of a set A of length n defined by

$$A \upharpoonright_r z_n = z_0\#A(z_0)\#z_1\#A(z_1)\#\dots z_{n-1}\#A(z_{n-1}). \tag{4.44}$$

We will use the following notation. We call $A \upharpoonright_r z_n$ the *redundant initial segment* of A of length n and we let

$$Prefix_r(A) = \{A \upharpoonright_r z_n : n \geq 0\}$$

be the prefix set of A with respect to redundant presentation. The set of all redundant initial segments is denoted by I_r:

$$I_r = \{z_0\#i_0\#\dots\#z_{n-1}\#i_{n-1} : n \geq 0 \ \&\ i_0,\dots,i_{n-1} \in \{0,1\}\}.$$

Then we can define extension functions operating on redundant initial segments and corresponding (bounded) genericity notions in the canonical way.

Definition 4.67 A (partial) *red-extension function* f is a (partial) function $f : I_r \to \Sigma^*$. If $f : I_r \to \Sigma^k$ then f is a *k-bounded red-extension function* ($k \geq 1$), and f is *bounded* if f is k-bounded for some $k \geq 1$.

Definition 4.68 The partial red-extension function f is *dense along* the set A if $f(A \upharpoonright_r z_n) \downarrow$ for infinitely many numbers n. *A meets f at n* if $f(A \upharpoonright_r z_n)$ is defined and $(A \upharpoonright z_n) f(A \upharpoonright_r z_n) \sqsubset \chi(A)$, and *A meets f* if A meets f at some n.

Note that the set I_r is not regular. So in order to define regular red-extension functions f we have to consider extensions of f.

Definition 4.69 A (partial) k-bounded red-extension function f is *regular* if there is a (partial) regular function $f' : (\Sigma \cup \{\#\})^* \to \Sigma^k$ such that f is the restriction of f' to I_r.

Note that a regular function $f' : (\Sigma \cup \{\#\})^* \to \Sigma^k$ inducing a total regular k-bounded red-extension function f may be partial. We only request that the set I_r

is contained in the domain of f'. Sometimes it will be convenient to define the extension f' on the regular superset

$$\Sigma^{\#} = \{x_0 \# i_0 \# \dots \# x_{n-1} \# i_{n-1} : n \geq 0 \ \& \ x_0, \dots, x_{n-1} \in \Sigma^* \ \& \ i_0, \dots, i_{n-1} \in \Sigma\}$$

of I_r.

Based on these definitions we obtain the following bounded finite-state genericity notions.

Definition 4.70 A set G is *red-k-reg-generic* if G meets all regular partial k-bounded red-extension functions which are dense along G; and G is *weakly red-k-reg-generic* if G meets all regular total k-bounded red-extension functions. G is *(weakly) red-ω-reg-generic* if G is (weakly) red-k-reg-generic for all $k \geq 1$.

Next we will show that these genericity notions subsume the corresponding standard genericity notions (introduced in Section 4.1) and the corresponding Cantor style genericity notions (introduced in the preceding section). For this sake we have to show that regular (partial) k-bounded extension functions and regular (partial) k-bounded diagonalization functions can be simulated by regular (partial) k-bounded red-extension functions.

Lemma 4.71 *Let f be a regular partial k-extension function ($k \geq 1$). There is a regular partial k-red-extension function f' such that, for any set A and any number n, $f'(A \restriction_r z_n)$ is defined if and only if $f(A \restriction z_n)$ is defined; and $f'(A \restriction_r z_n) = f(A \restriction z_n)$ if defined.*

PROOF. Given an automaton M which computes f, an automaton M' which computes the desired function f' works as follows. On input $x_0 \# i_0 \# x_1 \# i_1 \# \dots \# x_{n-1} \# i_{n-1}$ M' skips (i.e. reads without changing its state) the parts $x_0\#$, $x_1\#$, ..., $x_{n-1}\#$ and simulates M on the remaining part $i_0 \dots i_{n-1}$. □

Theorem 4.72 *Let A be (weakly) red-k-generic. Then A is (weakly) k-reg-generic ($k \in \mathbb{N} \cup \{\omega\}$).*

PROOF. By Lemma 4.71 and definition. □

For the simulation of diagonalization functions by red-extension functions we will need the following lemma.

Lemma 4.73 *Let $f : \Sigma^* \to \Sigma^k$ be a regular partial function ($k \geq 1$). There is a regular partial function $f_{\leftarrow} : \Sigma^* \to \Sigma^k$ such that, for any $n \geq 1$, $f_{\leftarrow}(z_{n-1})$ is defined if and only if $f(z_n)$ is defined, and – if defined – $f_{\leftarrow}(z_{n-1}) = f(z_n)$.*

PROOF. Fix a deterministic finite automaton $M = (\Sigma, S, \delta, s_0, F, \lambda)$ which computes f. By Lemma 2.43, it suffices to give a nondeterministic (consistent) finite automaton $M' = (\Sigma, S', \Delta', S'_0, F', \lambda')$ which computes $f_{\leftarrow}$.

The automaton M' on input z_n will simulate the automaton M on input z_{n+1}. To be more precise, if M accepts z_{n+1} and the computation of M ends in the accepting state s, i.e., if $f(z_{n+1})$ is defined and $f(z_{n+1}) = \lambda(s)$, then there will be a unique accepting computation of M' on input z_n and this computation will end in a state s' with $\lambda'(s') = \lambda(s)$ whence $f_{\leftarrow}(z_n) = f(z_{n+1})$. If M rejects z_{n+1}, i.e., if $f(z_{n+1})$ is undefined, then there will be no accepting computation of M' on input z_n whence $f_{\leftarrow}(z_n)$ will be undefined too.

We will describe the automaton M' only informally. Note that, for a string $x \in \{1\}^*$, $x+1 = 0^{|x|+1}$ while for a string $x \notin \{1\}^*$, x is of the form $x = u01^m$ and $x+1 = u10^m$. So on input x the automaton M' first makes a guess whether or not $x \in \{1\}^*$.

The computation guessing that this is the case simulates M on input $0^{|x|+1}$. I.e., the initial state is (a copy of) the state M enters after reading a single 0 and then, for any 1 read, M' performs the transition performed by M when reading a 0. If M' reads a 0, thereby realizing that its guess was wrong, it stops the simulation of M and goes into a rejecting state which it will never leave.

The computation guessing that x contains at least one 0 is nondeterministic depending on an additional guess: when a 0 is read M' has to guess whether or not this will be the last 0 in x. M' starts to simulate M on input x until a 0 is a read. Then M may decide either to continue the simulation of M on x (guessing that there will be another 0 in the not yet read part of x) or (guessing that this will be the last 0 in x) it simulates the transition of M when reading a 1 in this step, and in all consecutive steps, when reading a 1, M' will simulate the transition of M when reading a 0. In the latter case, when M' will later see a 0 (thereby realizing that its guess was wrong) it will stop the simulation of M and it will go into a rejecting state which it will never leave. Moreover, before M' guessed that a 0 it has seen is the last 0 in x it will always be rejecting though it may be in (the copy of) an accepting state of M. (So if the guess that there is a 0 in x was wrong or the chosen computation of M' fails to make a guess about the last 0 then this computation will be rejecting.) □

Lemma 4.74 *Let f be a regular partial k-diagonalization function ($k \geq 1$). There is a regular partial k-red-extension function f' such that, for any set A and any number n, $f'(A \upharpoonright_r z_n)$ is defined if and only if $f(z_n)$ is defined, and – if defined – $f'(A \upharpoonright_r z_n) = f(z_n)$.*

PROOF. By Lemma 4.73 we may fix a finite automaton M which computes $f_{\leftarrow}$.

Then a finite automaton M' which computes the desired function f' works as follows. On an input $w = x_0\#i_0\#x_1\#i_1\#\ldots\#x_{n-1}\#i_{n-1}$, M' skips (i.e. reads without changing its state) the parts $\#i_0$, $\#i_1$, ..., $\#i_{n-1}$ while on each x_m M' simulates M (beginning each of the simulations in the initial state of M). So, after reading w, M' is in the same state as M after reading x_{n-1}. Hence if w is a string $A \upharpoonright_r z_n$ then $f_{M'}(A \upharpoonright_r z_n) = f_M(z_{n-1}) = f_{\leftarrow}(z_{n-1}) = f(z_n)$. □

Theorem 4.75 *Let A be (weakly) red-k-generic. Then A is (weakly) k-C-reg-generic ($k \in \mathbb{N} \cup \{\omega\}$).*

PROOF. By Lemma 4.74 and definition. □

The preceding two theorems allow us to apply results about the standard bounded finite-state geniricity concepts and the results about the Cantor style bounded finite-state geniricity concepts. For instance we obtain the following results on the strength of the new genericity notions.

Theorem 4.76 *(a) For any weakly red-1-reg-generic set A, $A \notin$ REG.*

(b) For any weakly red-ω-reg-generic set A, A is saturated.

(c) For any red-1-reg-generic set A, A is saturated and REG*-bi-immune.*

PROOF. Parts (a), (b) and the second claim in (c) follow from Theorems 4.75 and 4.62. The first claim in (c) follows from Theorems 4.72 and 4.7. □

Moreover we can adapt some of the previous ideas to prove the following hierarchy theorem.

Theorem 4.77 *For $k \geq 2$ the following and only the following implications hold (up to transitive closure).*

$$\begin{array}{ccc}
\textit{red-}\omega\textit{-reg-generic} & \Rightarrow & \textit{weakly red-}\omega\textit{-reg-generic} \\
\Updownarrow & & \Downarrow \\
\textit{red-}(k+1)\textit{-reg-generic} & & \textit{weakly red-}(k+1)\textit{-reg-generic} \\
\Updownarrow & & \Downarrow \\
\textit{red-}k\textit{-reg-generic} & & \textit{weakly red-}k\textit{-reg-generic} \\
\Updownarrow & & \Downarrow \\
\textit{red-}1\textit{-reg-generic} & & \textit{weakly red-}1\textit{-reg-generic}
\end{array} \tag{4.45}$$

PROOF. We only sketch the proof. Note that the unique implication from left to right and the downward implications are immediate by definition. In order to justify the upward implications in the first column we adapt the proof of Theorem 4.4 to show that red-1-reg-genericity and red-ω-reg-genericity coincide. To show that, in general, weak red-k-reg-genericity does not imply weak red-$(k+1)$-reg-genericity, it suffices to observe that any weakly red-k-reg-generic set is k-ω-saturated (by Theorem 4.75 and Lemma 4.59) but, in general, not $(k+1)$-1-saturated. (To show the latter, the proof of Lemma 4.8 can be easily adapted to build a weakly red-k-reg-generic set which is not $(k+1)$-1-saturated.) This only leaves to show that weakly red-ω-reg-genericity in general does not imply red-1-reg-genericity. Since, by Theorem 4.76 the latter implies REG-bi-immunity it suffices to show that there is a weakly red-ω-reg-generic set which is not REG-bi-immune. But this can be easily done, e.g., by a straightforward finite extension argument, we can construct a weakly red-ω-reg-generic set A such that $\{0\}^*$ is contained in A. □

Theorem 4.76 gives us some lower bounds on the strength of the finite-state genericity notions based on the redundant representation of initial segments. Moreover, Theorems 4.72 and 4.75 show that these new genericity notions imply the corresponding genericity notions based on the standard representation of initial segments and the corresponding Cantor style genericity concepts. Of course it is interesting to also obtain some upper bounds on the strength of the new concepts and to analyze for what instances the implications among the different types of genericity notions are strict. The previously obtained results give some but not all answers.

For instance, for $k \geq 2$ the implications

$$\text{weakly red-}k\text{-reg-generic} \Rightarrow \text{weakly } k\text{-reg-generic}$$

and

$$\text{weakly red-}k\text{-reg-generic} \Rightarrow \text{weakly } k\text{-C-reg-generic}$$

are strict. This follows directly from our previous result that (for $k \geq 2$) neither weak k-reg-genericity implies weak k-C-reg-genericity nor weak k-C-reg-genericity implies weak k-reg-genericity (see Lemmas 4.66 and 4.64).

Questions left open by our previous results are the strictness of the following relations

$$\text{weakly red-1-reg-generic} \Rightarrow \text{weakly 1-C-reg-generic} \tag{4.46}$$

$$\text{weakly red-}\omega\text{-reg-generic} \Rightarrow \text{weakly } \omega\text{-C-reg-generic} \tag{4.47}$$

$$\text{red-}\omega\text{-reg-generic} \Rightarrow \omega\text{-C-reg-generic} \tag{4.48}$$

Note that by Theorem 4.62, a negative answer to the first two questions is equivalent to affirmatively answering the following interesting questions about the power of weak red-1-reg-genericity and of weak red-ω-reg-genericity:

$$A \text{ weakly red-1-reg-generic} \Leftrightarrow A \notin \text{REG} \tag{4.49}$$

$$A \text{ weakly red-}\omega\text{-reg-generic} \Leftrightarrow A \text{ saturated} \tag{4.50}$$

Though we conjecture that these equivalences can be established by extending some of our previous related arguments, we leave these questions open. In the following we will show, however, that the implication in (4.48) is strict.

Theorem 4.78 *There is an* ω-*C-reg-generic set A which is not red*-1-*reg-generic.*

Note that this implies that red-1-reg-genericity is strictly stronger than all of the finite-state genericity concepts introduced in the previous sections. For a proof of Theorem 4.78 it suffices to establish the following two lemmas.

Lemma 4.79 *For any* $k \geq 1$ *there is an* ω-*C-reg-generic set A such that*

$$\forall x, y \in A\, (|x| < |y| \Rightarrow |y| \leq |x| + 1 \ \vee \ |y| \geq |x| + k) \tag{4.51}$$

PROOF. Fix $k \geq 1$ and let $\{f_n : n \geq 0\}$ be an enumeration of all partial regular bounded diagonalization functions where w.l.o.g. we may assume that the function f_n is k-bounded for some $k \leq n$. By a finite extension argument we construct an ω-C-reg-generic set A satisfying (4.51): Given the finite initial segment $A_{s-1} = A \upharpoonright l(s-1)$ of A defined in the first $s-1$ stages of the construction, at stage s we define an extension $A_s = A \upharpoonright l(s)$ of A_{s-1} in such a way that A meets f_s – provided that f_s is dense – and at the same time (4.51) is satisfied. If f_s is not dense, obviously this is achieved by letting $l(s) = l(s-1)+1$ and by not adding any strings of length $l(s-1)$ to A, i.e., by letting $A_s = A_{s-1}$ (viewing A_{s-1} and A_s as sets). If f_s is dense then we choose the least number m such that $m > l(s-1)+k$ and $s < 2^m$ and the least corresponding number n such that $|z_n| \geq m$ and $f_s(z_n)\downarrow$. (Note that, by density of f_s, such a string z_n must exist.) Then we let $l(s) = |z_n| + 2$ and set

$$A \upharpoonright l(s) = (A \upharpoonright l(s-1))\, 0^{n-2^{l(s-1)}}\ f_s(z_n)\, 0^{2^{|z_n|+2}-(n+s)}.$$

(Less formally, we pick the least string $z = z_n$ of length $\geq l(s-1)+k$ such that $f_s(z)$ is defined and such that meeting f_s at z will only require to put strings of length $|z|$ and $|z|+1$ into A. Then we use this string to meet f_s and choose $l(s)$ big enough.) Obviously this implies that f_s is met. Moreover, for any strings $x, y \in A_s$ such that $|x| < |y|$, (4.51) holds: If $x, y \in A_{s-1}$ this is true by inductive hypothesis; if $x \in A_{s-1}$ and $y \in A_s \setminus A_{s-1}$ then $z_n \leq y$ whence, by choice of m and n, $|x| < l(s-1) \leq l(s-1)+k \leq m \leq |z_n| \leq |y|$, hence $|x|+k \leq |y|$; finally, if $x, y \in A_s \setminus A_{s-1}$, then x and y enter A for meeting f_s at z_n whence there are numbers $i < j < s$ such that $x = z_{n+i}$ and $y = z_{n+j}$, hence, by $s < 2^m \leq 2^{|z_n|}$, $|y| \leq |x|+1$. □

Lemma 4.80 *Let* $k \geq 1$ *and let A be red*-1-*reg-generic. There are infinitely many numbers n such that* $\{0^n, 0^{n+1}, \ldots, 0^{n+k-1}\} \subset A$.

PROOF. The proof is by induction on k. For $k = 1$ it suffices to show that for given $n_0 \geq 0$ there is a number $n \geq n_0$ such that $0^n \in A$. Consider the 1-bounded partial red-extension function f_1 induced by the partial function $f'_1 : \Sigma^{\#} \to \Sigma$ where

$$f'_1(x_0\#i_0\#\ldots\#x_{n-1}\#i_{n-1}) = \begin{cases} 1 & \exists m \geq n_0\ (x_{n-1} = 1^m) \\ \uparrow & \text{otherwise.} \end{cases}$$

Then f'_1 – hence f_1 – is regular and $f_1(A \upharpoonright_r 0^n)$ is defined for all numbers $n > n_0$. So, by red-1-reg-genericity, A meets f_1. But, by definition of f_1, this implies that there is a number $n > n_0$ such that $(A \upharpoonright 0^n)1 = (A \upharpoonright 0^n) f_1(A \upharpoonright_r 0^n) \sqsubset \chi(A)$, i.e., $0^n \in A$.

For the inductive step, fix $k \geq 1$ and assume that there are infinitely many numbers n such that $\{0^n, 0^{n+1}, \ldots, 0^{n+k-1}\} \subset A$ holds. Then, given $n_0 \geq 0$, we have to show that there is a number $n \geq n_0$ such that $\{0^n, 0^{n+1}, \ldots, 0^{n+k}\} \subset A$. Consider the 1-bounded partial red-extension function f_{k+1} induced by the partial function $f'_{k+1} : \Sigma^{\#} \to \Sigma$ where $f'_{k+1}(x_0\#i_0\#\ldots\#x_{n-1}\#i_{n-1}) = 1$ if $x_{n-1} = 1^m$ for some number $m \geq n_0 + k$, there are at least k numbers j, $0 \leq j < n-1$ such that $x_j \in \{0\}^*$, and, for the last k such strings $x_{j_1} < \cdots < x_{j_k}$, $i_{j_1} = \cdots = i_{j_k} = 1$; and where $f'_{k+1}(x_0\#i_0\#\ldots\#x_{n-1}\#i_{n-1})$ is undefined otherwise. Then f'_{k+1} – hence f_{k+1} – is regular. Moreover, $f_{k+1}(A \upharpoonright_r x)$ is defined if and only if $x = 0^{n+k}$ for some number $n > n_0$ and $A(0^n) = \ldots A(0^{n+k-1}) = 1$. So, by inductive hypothesis, f_{k+1} is dense along A whence, by red-1-reg-genericity, A meets f_{k+1}. But, by definition of f_{k+1}, this implies that there is a number $n > n_0$ such that $\{0^n, 0^{n+1}, \ldots, 0^{n+k-1}\} \subset A$.

□

CHAPTER 5

Unbounded Finite-State Genericity

After our detailed analysis of bounded finite-state genericity we now turn to more general finite-state genericity notions based on extensions of nonconstant length. We will obtain stronger and stronger concepts by considering more and more general notions of regular functions of type $\Sigma^* \to \Sigma^*$ for modelling the finite-state extension strategies. In Section 5.1 we start with (weak) Moore genericity based on (total) Moore functions. By showing that weakly Moore generic sets are saturated we show that (weak) Moore genericity refines bounded reg-genericity. Moreover, by analyzing the gaps in (weakly) Moore generic sets we show that – in contrast to all of the bounded finite-state genericity concepts considered in Chapter 4 – the class of the (weakly) Moore generic sets has measure 0. As we also observe, however, Moore genericity does not forces REG-(bi-)immunity. In fact, this is true if we strengthen this concept by considering nondeterministic Moore functions (see Section 5.2). By considering generalized Moore functions, however, we obtain a corresponding finite-state genericity concept which does not only force REG-bi-immunity but also CF-bi-immunity (Section 5.3).

5.1 Moore Genericity

We start our investigation of unbounded finite-state genericity by introducing genericity based on Moore functions, the most restrictive concept of an unbounded finite-state function (see Definition 2.33). We call the corresponding genericity concept Moore genericity.

Definition 5.1 A set A is *Moore generic* if A meets all partial extension functions $f : \Sigma^* \to \Sigma^*$ which are Moore functions and which are dense along A; and A is *weakly Moore generic* if A meets all total extension functions $f : \Sigma^* \to \Sigma^*$ which are Moore functions.

A sequence α is *(weakly) Moore generic* if the set $S(\alpha)$ corresponding to α is (weakly) Moore generic.

Recall that an automaton M computing a Moore function f produces on input x of length n the value $f(x) = y$ of length $n+1$ bit by bit. To be more precise, before reading the first bit of x, M determines the first bit of y and then for every bit read a bit is appended to the part of y produced before. For partial f, after having read the entire input x, M decides whether y should be taken as the value of $f(x)$ or whether $f(x)$ is undefined. This computation procedure immediately implies the following length and extension properties of (partial) Moore functions f where $v, w \in \Sigma^*$ (see Lemma 2.34).

$$f(w)\downarrow \Rightarrow\ |f(w)| = |w| + 1 \tag{5.1}$$

and

$$(v \sqsubseteq w \;\&\; f(v)\downarrow \;\&\; f(w)\downarrow) \;\Rightarrow\; f(v) \sqsubseteq f(w). \tag{5.2}$$

By being able to specify extensions of growing length one might expect that Moore extension functions are more powerful than regular bounded extension functions. On the other hand, however, an extension strategy based on a regular extension function f has to make a decision on the value a set A has to have at z_n in order to meet f at n only after the extension strategy has seen all of $A \upharpoonright n$ whereas a strategy based on a Moore function has to determine this value already before it has seen any part of $A \upharpoonright n$. So it is not obvious that (weak) Moore genericity implies bounded reg-genericity. Before we will turn to this question we first list some basic facts on Moore genericity including some technical lemmas which will be very useful for the following investigations.

5.1.1 Some Basic Properties

We first consider some closure and invariance properties of Moore genericity and weak Moore genericity. We first observe that both concepts are closed under complement. We then show that Moore genericity and the corresponding infinitely-often genericity concept coincide whereas this is not the case for weak Moore genericity. In particular, this shows that weak Moore genericity and Moore genericity do not coincide.

Lemma 5.2 *The class of the (weakly) Moore generic sets is closed under complement.*

PROOF. As one can easily show, for any (partial) Moore function f, the dual function $\hat{f}$ defined by $\hat{f}(\overline{x}) = \overline{f(x)}$ is a (partial) Moore function again. By Lemma 3.40 this implies the claim. □

Next we show that a Moore generic set A meets any partial Moore extension function which is dense along A not just once but infinitely often.

Lemma 5.3 *Let A be Moore generic. Then A infinitely often meets any partial extension function $f : \Sigma^* \to \Sigma^*$ which is a Moore function and which is dense along A.*

PROOF. It suffices to show that, for any partial Moore function f and any number n, the finite variant f' of f defined by

$$f'(x) = \begin{cases} f(x) & \text{if } |x| > n \\ \uparrow & \text{otherwise} \end{cases}$$

is a partial Moore function again. Then we can argue as in the proof of Lemma 3.37.

Now, given a Moore automaton $M = (\Sigma, S, \delta, s_0, F, \lambda)$ which computes f, we can convert M into an automaton $M' = (\Sigma, S', \delta', s'_0, F', \lambda')$ computing f' by letting $S' = S \cup \{s^k : s \in S \ \&\ k \leq n\}$, $s'_0 = s^0_0$, $F' = F$, $\lambda'(s^k) = \lambda'(s) = \lambda(s)$ for $s \in S$ and $k \leq n$, and by setting $\delta'(s^k, a) = \delta(s, a)^{k+1}$, $\delta'(s^n, a) = \delta(s, a)$ and $\delta'(s, a) = \delta(s, a)$ for $s \in S$, $k < n$, and $a \in \Sigma$. Intuitively, M' works as M but in the first n steps of a computation the current states $s_0, s_1, ...s_n$ are replaced by corresponding non-final states $s^0_0, s^1_1, \ldots, s^n_n$, respectively. □

As one can easily show, the class of (partial) Moore functions is closed under finite replacement (in the sense of Definition 3.38). By Lemmas 3.39 and 5.3 this implies that the class of Moore generic sets is closed under finite variants.

Lemma 5.4 *The class of the Moore generic sets is closed under finite variants.*

In contrast to the two preceding lemmas, however, there is a weakly Moore generic set which meets some total Moore extension function just once, and the class of weakly Moore generic sets is not closed under finite variants.

Lemma 5.5 *There is a weakly Moore generic set A which meets the total Moore extension function $f : \Sigma^* \to \Sigma^*$ defined by $f(x) = 1^{|x|+1}$ only at $n = 0$.*

PROOF. The required set A is constructed by a finite extension argument, i.e., at stage $s \geq 0$ of the construction of A we specify a finite initial segment $\alpha \upharpoonright l(s)$ of the characteristic sequence α of A.

Let $\{f_n : n \geq 1\}$ be an enumeration of the total Moore extension functions g such that $g(\varepsilon) = 0$, let $f_0 = f$, and let $l(-1) = 0$. Then, for $s = 0$, let $l(0) = 1$ and

$$\alpha \upharpoonright l(0) = 1 = f(\alpha \upharpoonright 0).$$

For the inductive step, given $\alpha \upharpoonright l(s)$, let $l(s+1) = 2l(s) + 1$ and

$$\alpha \upharpoonright l(s+1) = (\alpha \upharpoonright l(s))\, f_{s+1}(\alpha \upharpoonright l(s)).$$

(Note that, by f_{s+1} being a Moore function, $|f_{s+1}(\alpha \upharpoonright l(s))| = |\alpha \upharpoonright l(s)| + 1$ whence $\alpha \upharpoonright l(s+1)$ is well defined.) Now, by a straightforward induction on $s \geq 0$, A meets f_s at $l(s-1)$ and $\alpha(l(s)) = 0$. (The latter follows from the fact, that for a Moore function f, $(f(x))(0) = f(\varepsilon)$ by the extension property (5.2) whence, by construction and by choice of f_{s+1}, $\alpha(l(s)) = (f_{s+1}(\alpha \upharpoonright l(s))(0) = f_{s+1}(\varepsilon) = 0$.) Now the former implies that A is weakly Moore generic. Namely, given a total Moore function h, either $h(\varepsilon) = 1 = \alpha(0)$ whence A meets h at 0 or there is a number $s \geq 1$ such that $h = f_s$ whence A meets h at $l(s-1)$. It remains to show that A does not meet f at any number $n \geq 1$, i.e., that $\alpha(n)...\alpha(2n) \neq 1^{n+1}$ for all $n \geq 1$. But this easily follows from the fact that $l(0) = 1$, $l(s+1) = 2l(s) + 1$ and $\alpha(l(s)) = 0$ for $s \geq 0$. □

Lemma 5.6 *The class of the weakly Moore generic sets is not closed under finite variants.*

PROOF. Choose A and f as in Lemma 5.5 and let $A' = A \setminus \{z_0\}$. Then A is weakly Moore generic but the finite variant A' of A is not weakly Moore generic since, as one can easily check, A' does not meet the total (length invariant) Moore function f. □

Theorem 5.7 *The class of the Moore generic sets is strictly contained in the class of the weakly Moore generic sets.*

PROOF. Obviously any Moore generic set is weakly Moore generic. So it suffices to show that the class of the Moore generic sets and the class of the weakly Moore generic sets do not coincide. But this is immediate by Lemmas 5.4 and 5.6 (or by Lemmas 5.3 and 5.5). □

In the remainder of this subsection we prove some more technical facts on (weakly) Moore generic sets. We first show of a class of simple length invariant functions that they are of Moore type.

Definition 5.8 Let $\hat{f}:\mathbb{N}\rightarrow\Sigma$ be a total function. The function $f:\Sigma^*\rightarrow\Sigma^*$ *induced by* $\hat{f}$ is defined by $f(x)=\hat{f}(0)\ldots\hat{f}(|x|)$.

Lemma 5.9 *Let $\hat{f}:\mathbb{N}\rightarrow\Sigma$ be a total function such that, for some $i\in\Sigma$, $\hat{f}(n)=i$ for almost all numbers n. Then the function $f:\Sigma^*\rightarrow\Sigma^*$ induced by $\hat{f}$ is a total Moore function.*

PROOF. Straightforward. □

For the next technical lemma we need the following definition.

Definition 5.10 A total Moore function f is *compatible with* a string y if, for any string x, $f(x)$ is compatible with y, i.e., $f(x)\sqsubseteq y$ if $|x|<|y|$ and $y\sqsubseteq f(x)$ if $|x|\geq|y|$. f is *compatible with y along* a set A if, for any number n, $f(A\upharpoonright n)$ is compatible with y.

Lemma 5.11 *Let A be any set, let $A(z_0)=i$ ($i=0,1$), and let $n,p\geq 1$ be numbers such that*

$$|\{m:0<m\leq n\ \&\ A(z_m)=i\}|=p. \tag{5.3}$$

Then there is a string y such that $|y|=1+n-p$ and such that, for any total Moore function f which is compatible with y along A, A does not meet f at any number $m\leq n$.

PROOF. Define y by letting $y=y_0\ldots y_n$ where $y_0=1-A(z_0)=1-i$ and, for $0<m\leq n$, y_m is defined as follows. If $A(z_m)=i$ then $y_m=\varepsilon$. Otherwise, $y_m\in\Sigma$ is given by $y_m=1-A(z_{m+|y_0\ldots y_{m-1}|})$. Then, by definition of p, there are p numbers m with $0\leq m\leq n$ such that $y_m=\varepsilon$. So $|y|=1+n-p$. It remains to show that, given a total Moore function f which is compatible with y along A and given a number $m\leq n$, A does not meet f at m. Distinguish the following two cases.

If $A(z_m)=i$ then, by $y_0=1-i$ and by compatibility of f with y along A, $f(A\upharpoonright m)(0)=1-i$. So $(A\upharpoonright m)i\sqsubset\chi(A)$ while $(A\upharpoonright m)(1-i)\sqsubseteq(A\upharpoonright m)f(A\upharpoonright m)$ whence A does not meet f at m.

If $A(z_m) \neq i$ then $y_m = 1 - A(z_{m+|y_0 \dots y_{m-1}|})$. So, for $k = |y_0 \dots y_{m-1}|$, the strings $A(z_m) \dots A(z_{m+k})$ and $y_0 \dots y_m$ are incompatible: Namely, both strings have length $k+1$ and, for their last bits $A(z_{m+k})$ and y_m, respectively, $A(z_{m+k}) \neq y_m$. On the other hand, by compatibility of f with y along A, $y_0 \dots y_m \sqsubseteq f(A \upharpoonright m)$. So $A(z_m) \dots A(z_{m+k})$ and $f(A \upharpoonright m)$ are incompatible, hence A does not meet f at m. □

Note that Lemma 5.11 in particular implies that for any set A such that $A(z) = A(z_0)$ for infinitely many strings z, for any number p there is a number n and a string y of length $n - p$ such that A does not meet any Moore extension function f compatible with y at any number $m \leq n$. In order to show that this applies to any weakly Moore generic A set, we next observe that the characteristic sequence of any weakly Moore generic set contains infinitely many zeroes and ones.

Lemma 5.12 *Let A be weakly Moore generic and let α be the characteristic sequence of A. Then $\{n : \alpha(n) = 0\}$ and $\{n : \alpha(n) = 1\}$ are infinite.*

PROOF. By symmetry it suffices to show that $\{n : \alpha(n) = 0\}$ is infinite. For a contradiction assume that there are only finitely many occurrences of 0 in α and fix n_0 such that $\alpha = (\alpha \upharpoonright n_0)1^\omega$. Define the function $\hat{f} : \mathbb{N} \to \Sigma$ by letting $\hat{f}(n) = 1 - \alpha(2n)$. Note that $\hat{f}(n) = 0$ for $n > n_0$. Hence, by Lemma 5.9, the function f induced by $\hat{f}$ via $f(x) = \hat{f}(0) \dots \hat{f}(|x|)$ is a total Moore function. Moreover, A does not meet f at any number n. Namely, for any $n \geq 0$, $\alpha \upharpoonright 2n+1 \neq (\alpha \upharpoonright n) f(\alpha \upharpoonright n)$ since, by definition of f and $\hat{f}$, $\alpha \upharpoonright 2n+1$ and $(\alpha \upharpoonright n) f(\alpha \upharpoonright n)$ differ in the last bit:

$$[(\alpha \upharpoonright n) f(\alpha \upharpoonright n)](2n) = f(\alpha \upharpoonright n)(n) = \hat{f}(n) = 1 - \alpha(2n).$$

It follows that A is not weakly Moore generic contrary to assumption. □

Our final technical lemma bounds the gaps in the domains of partial finite-state functions. It does not only apply to partial Moore functions but to bounded partial regular functions as well.

Lemma 5.13 *Let f be a partial bounded regular extension function or a partial Moore extension function computed by a finite automaton with k states. Then*

$$\forall x \in \Sigma^* \ ([\exists y \in \Sigma^* \ (f(xy) \downarrow)] \Rightarrow [\exists y' \in \Sigma^{<k} \ (f(xy') \downarrow)] \qquad (5.4)$$

holds.

PROOF. Assume that f is computed by the automaton $M = (\Sigma, S, \delta, s_0, F, \lambda)$ where $|S| = k$ and that, for given x, there is a string y such that $f(xy)$ is defined. Since $f(z)$ is defined if and only if M is a final state after reading z, it follows that there are states s and s' such that $\delta^*(s_0, x) = s$, $\delta^*(s, y) = s'$ and $s' \in F$. So the state s' is reachable from s. But then, for the shortest string y' with $\delta^*(s, y') = s'$, $|y'| < |S| = k$, since the corresponding run $s = s_1, s_2, \ldots s_{|y'|+1} = s'$ of M does not contain any loops (i.e. repetitions). So y' is the requested string of length $< k$ with $f(xy') \downarrow$. □

5.1.2 Moore Genericity and Saturation

In order to show that weak Moore genericity (hence Moore genericity) is a refinement of bounded reg-genericity we will show that weakly Moore generic sets are saturated. For a Moore generic set A we can show that A is saturated by applying Lemma 5.3. Namely, in order to show that a given string x occurs in the characteristic sequence α of A we define a total Moore function f by letting $f(y) = x \upharpoonright (|y|+1)$ if $|y| < |x|$ and by letting $f(y) = x0^{|y|-|x|+1}$ otherwise. Since, by Lemma 5.3, A meets f infinitely often, there is a number $n > |x|$ such that A meets f at n. So $f(\alpha \upharpoonright n) = x0^{n-|x|+1}$ occurs in α, hence, in particular, x occurs in α. To show that weakly Moore generic sets are saturated too we need a somewhat more sophisticated argument.

Theorem 5.14 *Let A be weakly Moore generic. Then A is saturated.*

PROOF. For a contradiction assume that A is not saturated and fix a nonempty string x such that x does not occur in the characteristic sequence α of A, and let $i = A(\varepsilon)$. Since, by Lemma 5.12, the bit i occurs in α infinitely often, we may choose a number $n \geq 1$ such that, for $p = |x| + 1$, (5.3) holds. So, by Lemma 5.11, there is a string y of length $1 + n - p$ such that, for any total Moore function f which is compatible with y along A, A does not meet f at any number $m \leq n$. Fix such a string y and define $\hat{f} : \mathbb{N} \to \Sigma$ by letting $\hat{f}(0) \ldots \hat{f}(n-1) = yx$ (note that $|yx| = |y| + |x| = (1+n-p) + (p-1) = n$) and by letting $\hat{f}(m) = 0$ for $m \geq n$. Then, by Lemma 5.9, the extension function f induced by $\hat{f}$ via $f(x) = \hat{f}(0) \ldots \hat{f}(|x|)$ is a Moore function. By definition, f is compatible with yx, hence compatible with y. By choice of y, the latter implies that A does not meet f at any number $m \leq n$. On the other hand, for $m > n$, by compatibility of f with yx (and by $|xy| = n < m+1 = |f(A \upharpoonright m)|$), $yx \sqsubseteq f(A \upharpoonright m)$. So if A meets f at $m > n$ then x will occur in α. Since, by assumption, x does not occur in α we may conclude that A does not meet f. It follows that A is not weakly Moore generic which gives the desired contradiction. □

By coincidence of the bounded reg-generic sets with the saturated sets, the preceding theorem shows that the class of weakly Moore generic sets is contained in the class of bounded reg-generic sets. Moreover, since saturated sets are nonregular, we may conclude that no regular set is weakly Moore generic.

Corollary 5.15 *Any weakly Moore generic set is bounded reg-generic.*

Corollary 5.16 *No regular set is weakly Moore generic.*

Note that the implication in Corollary 5.15 is strict since, in contrast to weak Moore genericity, bounded reg-genericity is closed under finite variants. In the next subsection we will show this in a different way: In contrast to any bounded genericity concept, the class of weakly Moore generic sets has measure 0. We will show the latter by analyzing the gaps occurring in (weakly) Moore generic sets.

5.1.3 Moore Genericity, Gaps and Measure

We will next compare the length of gaps occurring in the characteristic sequences of bounded-reg generic sets, weakly Moore generic sets, and Moore generic sets.

Definition 5.17 Let $f : \mathbb{N} \rightarrow \mathbb{N}$ be total. We say that a sequence α has an *f-gap at n* if $(\alpha \upharpoonright n)0^{f(n)} \sqsubset \alpha$; α has *f-gaps* if α has f-gaps at infinitely many numbers n, i.e., if there are infinitely many numbers n such that $(\alpha \upharpoonright n)0^{f(n)} \sqsubset \alpha$. α has *k-gaps* ($k \geq 0$) if α has f-gaps for the constant function $f(n) = k$.

We extend Definition 5.17 to sets by saying that a set A has f-gaps if its characteristic sequence has such gaps.

A characterization of the gaps occurring in all bounded reg-generic sets follows from the next two lemmas.

Lemma 5.18 *Any bounded reg-generic set A has k-gaps for all numbers $k \geq 1$.*

PROOF. By the coincidence of bounded reg-genericity with ω-ω-saturation, for any number k, the word 0^k occurs in the characteristic sequence of any bounded reg-generic set infinitely often. □

Lemma 5.19 *Let $f : \mathbb{N} \rightarrow \mathbb{N}$ be a nondecreasing total function which is unbounded (i.e., for any number $k \geq 1$, $f(n) > k$ for some n). Then there is a bounded reg-generic set A which does not have f-gaps.*

PROOF. Given an unbounded, nondecreasing function $f : \mathbb{N} \rightarrow \mathbb{N}$, by a finite extension argument we define a bounded reg-generic set A without f-gaps. Let $\{f_e : e \geq 0\}$ be an enumeration of the total regular bounded extension functions where w.l.o.g. f_e is e-bounded. Then, given a finite initial segment $\alpha_{s-1} = \alpha \upharpoonright l(s-1)$ of the characteristic sequence α of the set A under construction, we define an extension $\alpha_s = \alpha \upharpoonright l(s)$ of α_{s-1} which guarantees that A meets f_e. Moreover, this extension is chosen so that A does not have f-gaps.

For the definition of $l(s)$ and α_s, choose $n > l(s-1)$ minimal such that $f(n) > s$. (Note that, by f being unbounded and nondecreasing, for any number s, $f(n) > s$ for almost all numbers n, hence such an n exists.) Then let $l(s) = n+s+1$ and set $\hat{\alpha}_s = \alpha_{s-1}1^{n-l(s-1)}$ and $\alpha_s = \hat{\alpha}_s f_s(\hat{\alpha}_s)1$. Obviously this ensures that A meets f_s at n. So A is weakly ω-reg-generic, hence bounded reg-generic by Theorem 4.7. Moreover, the construction ensures that, A does not have f-gaps. Namely, whenever the word 0^k, $k > 0$ occurs in α, say $(\alpha \upharpoonright m)0^k \sqsubset \alpha$ then there are numbers n and s such that $(\alpha \upharpoonright n-1)1f_s(\alpha \upharpoonright n)1 \sqsubset \alpha$, $s < f(n)$ and 0^k is a subword of $f_s(\alpha \upharpoonright n)$. It follows that $k \leq s < f(n) \leq f(m)$ (where the latter follows from $n \leq m$ since f is nondecreasing). So this occurrence of 0^k in α does not establish an f-gap. □

Theorem 5.20 *Let $f : \mathbb{N} \to \mathbb{N}$ be nondecreasing and total. The following are equivalent.*

1. *f is bounded, i.e., there is a number k such that $f(n) \leq k$ for all numbers $n \geq 0$.*
2. *Every bounded reg-generic set has f-gaps.*

PROOF. This is immediate by Lemmas 5.18 and 5.19 □

The following two lemmas determine the size of gaps occurring in all Moore generic sets.

Lemma 5.21 *Any Moore generic set A has $(n+k)$-gaps for all numbers $k \geq 1$.*

PROOF. Let α be the characteristic sequence of a Moore generic set A and fix $k \geq 1$. Define the partial function $f : \Sigma^* \to \Sigma^*$ by letting $f(w)$ be defined if and only if $w = v0^k$ for some $v \in \Sigma^+$ and by letting $f(w) = 0^{|w|+1}$ if $f(w)$ is defined. Then, as one can easily check, f is a partial Moore function and f is dense along any saturated set. Since, by Theorem 5.14, Moore generic sets are saturated it follows, by Lemma 5.3, that A meets f infinitely often. But, for any number n such that A meets f at n,

$$(\alpha \upharpoonright n)f(\alpha \upharpoonright n) = (\alpha \upharpoonright n-k)0^k0^{n+1} \sqsubset \alpha$$

hence α has an $(n+k)$-gap at $n-k$. This completes the proof. □

Lemma 5.22 *Let $f : \mathbb{N} \to \mathbb{N}$ be a total, nondecreasing function such that, for any number $k \geq 1$, $f(n) > n+k$ for some n. Then there is a Moore generic set A which does not have f-gaps.*

PROOF. The proof is similar to that of Lemma 5.19. But since, in contrast to ω-reg-genericity and weak ω-reg-genericity, Moore genericity and weak Moore genericity do not coincide, here we have to work with partial extension functions. We will use Lemma 5.13 in order to adapt the argument for total functions to partial functions.

Let f be as in the premise of the lemma. Then, for any number k, $f(n) > n+k$ for almost all numbers n. By a finite extension argument we define a Moore generic set A without f-gaps. Let $\{f_e : e \geq 0\}$ be an enumeration of the partial Moore extension functions where w.l.o.g. we may assume that f_e is computed by an automaton with at most e states. Then, given a finite initial segment $\alpha_{s-1} = \alpha \upharpoonright l(s-1)$ of the characteristic sequence α of the set A under construction, we define an extension $\alpha_s = \alpha \upharpoonright l(s)$ of α_{s-1} which guarantees that A meets f_e. Moreover, this extension is chosen so that A does not have f-gaps.

For the definition of $l(s)$ and α_s, choose $n > l(s-1)$ minimal such that $f(n) > n+s+1$, set $\hat{\alpha}_s = \alpha_{s-1}1^{n-l(s-1)}$ and distinguish the following two cases. If there is a string y of length $\leq s$ such that $f_s(\hat{\alpha}_s y)$ is defined than fix the least such y and let $\alpha_s = \hat{\alpha}_s y f_s(\hat{\alpha}_s y)1$. Otherwise, let $\alpha_s = \hat{\alpha}_s$. Note that in the former case the choice of α_s ensures that A meets f_s. In the latter case, by Lemma 5.13, there is no extension of $\alpha_s = \hat{\alpha}_s$ on which f_s is defined, whence f_s is not dense along A. It follows that A is Moore generic.

Finally, by construction, a substring 0^k of α has to be contained in the $f_s(\hat{\alpha}_s y)$ part of an initial segment $\alpha_s = \hat{\alpha}_s y f_s(\hat{\alpha}_s y)1$. By construction, however, $|y| < s$ and, for $n = |\hat{\alpha}_s|$, $f(n) > n+2s$. So

$$k < |f_s(\hat{\alpha}_s y)| \leq (n+s)+1 < f(n).$$

Since f is nondecreasing this implies that the occurrence of 0^k in α does not induce an f-gap in α. □

Theorem 5.23 *Let $f : \mathbb{N} \to \mathbb{N}$ be nondecreasing and total. The following are equivalent.*

1. *There is a number k such that $f(n) \leq n+k$ for all numbers $n \geq 0$.*

2. *Every Moore generic set has f-gaps.*

PROOF. This is immediate by Lemmas 5.21 and 5.22 □

Theorem 5.23 does not carry over to weak Moore genericity. For instance, by considering the complement of the weakly Moore generic set of Lemma 5.5 we obtain a weakly Moore generic set which does not have $(n+1)$-gaps.

Lemma 5.24 *There is a weakly Moore generic set A which does not have $(n+1)$-gaps.*

On the other hand, in contrast to bounded-reg genericity, there is an unbounded nondecreasing function f such that there are f-gaps in all weakly Moore generic sets.

Lemma 5.25 *Every weakly Moore generic set A has $\frac{n}{4}$-gaps.*

PROOF. Let A be weakly Moore generic and fix $n_0 \geq 0$. We have to show that there is a number $n \geq n_0$ such that $(\alpha \upharpoonright n)0^{\frac{n}{4}} \sqsubset \alpha$ where α is the characteristic sequence of A.

Fix $i \leq 1$ such that

$$\exists^\infty n\ (|\{z_m : m < n\ \&\ A(z_m) = i\}| \geq \frac{n}{2}).$$

and fix $n_1 \geq n_0$ such that $|\{z_m : m < n_1\ \&\ A(z_m) = i\}| \geq \frac{n_1}{2}$ holds. Distinguish the following cases depending on the relation between i and $A(z_0)$.

First assume that $A(z_0) = i$. Then, by Lemma 5.11, there is a string x of length $\leq \frac{n_1}{2} + 1$ such that, for any Moore function f compatible with x, A does not meet f at any number $\leq n_1$. So, in particular, this is true for the Moore function f induced by the function $\hat{f} : \mathbb{N} \to \mathbb{N}$ defined by $\hat{f}(0) \dots \hat{f}(|x|-1) = x$ and $\hat{f}(n) = 0$ for $n \geq |x|$. On the other hand, by weak Moore genericity of A, A meets f. So A meets f at some number $n_2 > n_1$. It follows that

$$(\alpha \upharpoonright n_2) f(\alpha \upharpoonright n_2) \sqsubset \alpha.$$

Since $f(\alpha \upharpoonright n_2) = x0^{n_2+1-|x|}$ and $|x| \leq \frac{n_1}{2} + 1 \leq \frac{n_2}{2} + 1$ it follows that, for $n = n_2 + |x|$,

$$(\alpha \upharpoonright n)0^{\frac{n}{4}} \sqsubseteq (\alpha \upharpoonright n)0^{n_2+1-|x|} = (\alpha \upharpoonright n_2) f(\alpha \upharpoonright n_2) \sqsubset \alpha.$$

This completes the proof for the first case.

Now if $A(z_0) \neq i$ then we can apply Lemma 5.11 to the complement of A. Hence we can argue as in the first case with $\overline{A}$ in place of A. So, by replacing the function $\hat{f}$ defined there by $\hat{f}(0) \dots \hat{f}(|x|-1) = x$ and $\hat{f}(n) = 1$ for $n \geq |x|$ we can argue that we can find $n \geq n_1$ such that

$$(\overline{\alpha} \upharpoonright n)1^{\frac{n}{4}} \sqsubset \overline{\alpha}$$

where $\overline{\alpha}$ denotes the characteristic sequence of $\overline{A}$. Obviously this implies $(\alpha \upharpoonright n)0^{\frac{n}{4}} \sqsubset \alpha$.

This completes the proof. □

As announced above, we can use the gaps occurring in all weakly Moore generic sets to show that the class of these sets has measure 0.

Theorem 5.26 *Let A be a set such that the characteristic sequence* α *of A satisfies the law of large numbers, i.e.,*

$$\lim_{n\to\infty} \frac{|\{m : m < n \ \& \ \alpha(m) = 0\}|}{n} = \frac{1}{2}.$$

Then A is not weakly Moore generic.

PROOF. For a contradiction assume that A is weakly Moore generic and satisfies the law of large numbers. By the latter, we may choose $n_0 > 0$ such that

$$\forall n \geq n_0 \left(\frac{1}{2} - \frac{1}{50} < \frac{|\{m : m < n \ \& \ \alpha(m) = 0\}|}{n} < \frac{1}{2} + \frac{1}{50}\right). \tag{5.5}$$

On the other hand, by weak Moore genericity of A and by Lemma 5.25, we may choose $n > n_0$ such that $(\alpha \upharpoonright n)0^{\frac{n}{4}} \sqsubset \alpha$. It follows, for $n' = n + \frac{n}{4}$,

$$\begin{array}{rcll} \frac{|\{m:m<n' \ \& \ \alpha(m)=0\}|}{n'} & = & \frac{|\{m:m<n \ \& \ \alpha(m)=0\}|+\frac{n}{4}}{n+\frac{n}{4}} & \\ & = & \frac{4}{5}\frac{|\{m:m<n \ \& \ \alpha(m)=0\}|}{n} + \frac{1}{5} & \\ & > & \frac{4}{5}(\frac{1}{2} - \frac{1}{50}) + \frac{1}{5} & \text{(by (5.5))} \\ & > & \frac{1}{2} + \frac{1}{50}. & \end{array}$$

But this contradicts (5.5). □

Corollary 5.27 *The class of the weakly Moore generic sets has measure 0.*

PROOF. This is immediate by Theorem 5.26 since the class of the sets satisfying the law of large numbers has measure 1. □

Corollary 5.28 *The class of weakly Moore generic sets is strictly contained in the class of bounded reg-generic sets.*

PROOF. This is immediate by Corollary 5.27 since the class of the bounded reg-generic sets has measure 1. □

Note that, for any bounded genericity concept, the class of the corresponding generic sets has measure 1 (Theorem 3.35). So none of the bounded finite-state genericity concepts introduced in Chapter 4 implies weak Moore genericity. Another interesting consequence of Theorem 5.26 is that the characteristic sequences of (weakly) Moore generic sets are saturated (by Theorem 5.14) but not normal. The latter follows from Theorem 5.26 since, by definition, normal sequences satisfy the law of large numbers.

5.1.4 Moore Genericity and Immunity

By Corollary 5.16, (weakly) Moore generic sets are not regular. In order to get a better impression on the diagonalization power of the Moore genericity concept it is natural to ask whether (weakly) Moore generic sets are REG-bi-immune. Here we give a negative answer.

Theorem 5.29 *There is a Moore generic set A which is not* REG-*immune.*

PROOF. It suffices to construct a Moore generic set A such that

$$\{0^{2n} : n \geq 0\} \subseteq A. \tag{5.6}$$

By the latter, A will not be REG-immune. The construction of A closely follows the proof of Lemma 5.22. As there we fix an enumeration $\{f_e : e \geq 0\}$ of the partial Moore-functions such that the eth function f_e is computed by an automaton with at most e states. So, by Lemma 5.13, for $e \geq 0$,

$$\forall x \in \Sigma^* \, ([\exists y \in \Sigma^* \, (f_e(xy)\downarrow)] \Rightarrow [\exists y' \in \Sigma^{<e} \, (f_e(xy')\downarrow)]) \tag{5.7}$$

holds.

Now, in stage s of the construction, we will determine A for all strings x with $2s \leq |x| < 2s+2$, i.e., the values $\alpha(2^{2s}-1)\ldots\alpha(2^{2s+2}-2)$ of the characteristic sequence α of A. In order to satisfy (5.6) we let $A(0^{2s}) = \alpha(2^{2s}-1) = 1$. Then we ask whether there is a string y of length $< s$ such that $f_s((\alpha \restriction 2^{2s})y)$ is defined. If so, then, for the least such string y we let

$$\alpha \restriction 2^{2s+2} - 1 = (\alpha \restriction 2^{2s} - 1)1y f_s((\alpha \restriction 2^{2n})y)0^{2^{2s+2}-(1+2(2^{2s}+|y|)+1)}.$$

Obviously, this extension ensures that A meets f_s. If there is not string y as above then, by (5.7), there is no extension of $\alpha \restriction 2^{2s}$ on which f_s is defined. So, in this case, f_s is not dense along A and we can define

$$\alpha \restriction 2^{2s+2} - 1 = (\alpha \restriction 2^{2s} - 1)10^{2^{2s+2}-(2^{2s}-2)}.$$

□

In the next two sections we will discuss two strengthenings of Moore genericity which are based on nondeterministic Moore functions and generalized Moore functions, respectively.

5.2 Nondeterministic Moore Genericity

Some of the limitations of Moore functions, namely the extension property (5.2) (but not the length property (5.1)), can be eliminated by considering nondeterministic Moore functions (see Example 2.40). We call the corresponding genericity notion (weak) nondeterministic Moore genericity or (weak) NM-genericity for short.

Definition 5.30 A set A is *nondeterministic Moore generic* (or NM- *generic* for short) if A meets all partial extension functions $f : \Sigma^* \to \Sigma^*$ which are nondeterministic Moore functions and which are dense along A; and A is *weakly nondeterministic Moore generic* (or *weakly* NM- *generic* for short) if A meets all total extension functions $f : \Sigma^* \to \Sigma^*$ which are nondeterministic Moore functions.

As in the deterministic case we can show that the class of the (weakly) nondeterministic Moore generic sets is closed under complement. In contrast to the deterministic case, however, not only nondeterministic Moore genericity but also weak nondeterministic Moore genericity is an infinitely-often genericity concept.

Lemma 5.31 *If A is NM-generic then A infinitely often meets any partial extension function $f : \Sigma^* \to \Sigma^*$ which is a nondeterministic Moore function and which is dense along A. Similarly, if A is weakly NM-generic then A infinitely often meets any total extension function $f : \Sigma^* \to \Sigma^*$ which is a nondeterministic Moore function.*

PROOF. We give a proof of the second part of the lemma (which can be easily modified to prove the first part). Given a weakly NM-generic set A, a total extension function f which is nondeterministically Moore computable and a number n_0, it suffices to show that A meets f at some number $n \geq n_0$.

Consider the finite variant f' of f defined by

$$f'(x) = \begin{cases} (1 - A(z_{|x|}))^{|x|+1} & \text{if } |x| \leq n_0 \\ f(x) & \text{otherwise.} \end{cases}$$

Then f' is a nondeterministic Moore function again: A nondeterministic automaton M' computing $f'(x)$ first guesses whether or not the input x has length $> n_0$. If M' guesses that $|x| > n_0$ then M' simulates a nondeterministic automaton computing f on input x. In addition, M' counts the first n_0 steps of the computation and does not accept if the computation is completed before step $n_0 + 1$. If M' guesses that $|x| \leq n_0$ then M' makes an additional guess about the length of $|x|$. If M' guesses that $|x| = m$ then M' runs through $m+1$ states $s_0^m, \ldots, s_m^m$ all labelled with $1 - A(z_m)$.

Only the last of these states, s^m_m, is accepting; moreover, if the computation is not completed when entering s^m_m, m' will enter a rejecting state which it will never leave later.

It follows, by weak NM-genericity of A that A meets f' at some number n. The function f' has been defined in such a way, however, that A does not meet f' at any number $\leq n_0$. So A meets f' at some number $n > n_0$. Since f and f' coincide on inputs of length greater than n_0, this implies that A meets f above n_0. □

Lemmas 5.5 and 5.31 immediately imply that weak NM-genericity is strictly stronger than weak Moore genericity. This difference is also reflected by the gaps occurring in the weakly Moore generic sets and the weakly NM-generic sets as we will show next. The following results on gaps in weakly NM-generic sets will also show that in general weak NM-genericity does not imply Moore genericity, i.e., that nondeterministic total Moore extension functions in general cannot simulate deterministic partial Moore functions.

Theorem 5.32 *Any weakly NM-generic set has $(n+1)$-gaps but there is a weakly NM-generic set A which does not have $(n+2)$-gaps.*

PROOF. For a proof of the first part, assume that A is weakly NM-generic. By Lemma 5.31, A infinitely often meets the (deterministic length invariant) Moore extension function f defined by $f(x) = 0^{|x|+1}$. Obviously, this implies that A has $(n+1)$-gaps. For a proof of the second part, by a standard finite extension argument we construct a weakly NM-generic set without $(n+2)$-gaps. Given an enumeration $\{f_e : e \geq 0\}$ of the nondeterministic total Moore extension functions, in stage s of the construction we define a finite extension α_s of the previously specified initial segment α_{s-1} of the characteristic sequence α of A by letting $\alpha_s = \alpha_{s-1} f_s(\alpha_{s-1})1$. Obviously this ensures that A meets f_s whence A is weakly NM-generic. Moreover, since the nondeterministic Moore functions f have the length property (5.1), $|f_s(\alpha_{s-1})| = |\alpha_{s-1}| + 1$. Hence inserting a 1 at the end of each extension step ensures that there is no number n with $(\alpha \upharpoonright n)0^{n+2} \sqsubset \alpha$. So A does not have $(n+2)$-gaps. □

Corollary 5.33 *There is a weakly NM-generic set which is not Moore generic. Hence, in particular, NM-genericity is strictly stronger than weak NM-genericity.*

PROOF. This is immediate by the second part of Theorem 5.32 and by Lemma 5.21. □

The results on gaps in Moore generic sets easily carry over to NM-generic sets. Similarly, the proof of Theorem 5.29 easily extends to NM-genericity.

Theorem 5.34 *There is a nondeterministic Moore generic set A which is not* REG-*immune.*

So replacing Moore extension functions by nondeterministic Moore extension functions does not lead to extension strategies forcing REG-bi-immunity. As we will show in the next section, however, generalized Moore extension strategies have this power.

5.3 Generalized Moore Genericity

Our second refinement of Moore genericity is based on (partial) generalized Moore functions.

Definition 5.35 A set A is *generalized Moore generic* (or GM- *generic* for short) if A meets all partial extension functions $f : \Sigma^* \to \Sigma^*$ which are generalized Moore functions and which are dense along A; and A is *weakly generalized Moore generic* (or *weakly* GM- *generic* for short) if A meets all total extension functions $f : \Sigma^* \to \Sigma^*$ which are generalized Moore functions.

We can further refine this genericity concept by considering Moore extension functions which are both, generalized and nondeterministic.

Definition 5.36 A set A is *generalized nondeterministic Moore generic* (or GNM- *generic* for short) if A meets all partial extension functions $f : \Sigma^* \to \Sigma^*$ which are generalized nondeterministic Moore functions and which are dense along A; and A is *weakly generalized nondeterministic Moore generic* (or *weakly* GNM- *generic* for short) if A meets all total extension functions $f : \Sigma^* \to \Sigma^*$ which are generalized nondeterministic Moore functions.

As we have observed in Section 2.3, generalized (deterministic or nondeterministic) partial Moore functions f in general do not have the length property (5.1) but satisfy the more relaxed length condition

$$\exists c \geq 1 \, \forall w \in \Sigma^* \, (f(w) \downarrow \Rightarrow \, |f(w)| \leq c(|w|+1)). \tag{5.8}$$

In addition, the generalized deterministic Moore functions – but in general not the generalized nondeterministic Moore functions – have the extension property (5.2).

The relaxation of the length property leads to larger gaps in the characteristic sequences of (weakly) generalized Moore generic sets.

Lemma 5.37 *Any weakly GM-generic set has cn-gaps (for any $c \geq 1$).*

PROOF. Let A be weakly GM-generic, let α be the characteristic sequence of A, let $c \geq 1$, and let $n_0 \geq 0$. It suffices to show that there is a number $n \geq n_0$ such that $(\alpha \upharpoonright n)0^{cn} \sqsubset \alpha$.

First define a string y of length n_0 by letting $y = y_0 \dots y_{n_0-1}$ where $y_n = 1 - A(z_{2n})$ for $0 \leq n < n_0$. By definition, of y,

$$\forall n < n_0 \, ((\alpha \upharpoonright n) y \not\sqsubset \alpha).$$

So, for any total extension function f such that $y \sqsubseteq f(\varepsilon)$ and f has the extension property (5.2), A does not meet f at any number $n < n_0$.

Now consider the length invariant extension function f defined by $f(\varepsilon) = y0^{c \cdot n_0}$ and $f(x) = f(\varepsilon)0^{c \cdot |x|} = y0^{c \cdot (n_0 + |x|)}$ for nonempty x. Obviously, f is a generalized Moore function. So, by weak GM-genericity of A, A meets f at some n. Moreover, since generalized Moore functions have the extension property, by the above observation, A meets f at some number $n \geq n_0$. Fix such a number n. Then, by definition of f and by A meeting f at n,

$$(\alpha \upharpoonright n)y0^{c \cdot (n_0+n)} = (\alpha \upharpoonright n) f(\alpha \upharpoonright n) \sqsubset \alpha.$$

Since $|y| = n_0$ this implies

$$(\alpha \upharpoonright (n+n_0))0^{c \cdot (n_0+n)} \sqsubset \alpha.$$

Hence α has an cn-gap at $n+n_0$. □

As the following lemma shows, the preceding result is optimal and the length of gaps produced by generalized Moore extension functions cannot be increased if we allow the functions to be partial and nondeterministic.

Lemma 5.38 *Let $f : \mathbb{N} \to \mathbb{N}$ be a nondecreasing and total function such that $f(n) \notin O(n)$. Then there is a GNM-generic set A which does not have f-gaps.*

PROOF. Since the proof resembles the proofs of previous results of the same type (as for example Lemma 5.22), we only sketch the proof. The desired set A is constructed by a finite extension argument. We fix an enumeration $\{f_e : e \geq 1\}$ of the partial nondeterministic generalized Moore functions. W.l.o.g. we may assume that, for any number e and any nonempty string x,

$$(f_e(x)\downarrow \Rightarrow |f_e(x)| \leq e|x|) \ \& \ ([\exists y \in \Sigma^* \ (f(xy)\downarrow)] \Rightarrow [\exists y' \in \Sigma^{<e} \ (f(xy')\downarrow)] \quad (5.9)$$

holds. (Note that the former can be achieved by (5.8) while the latter can be achieved by extending Lemma 5.13 to nd. generalized partial Moore functions.)

Then, given the previously defined initial segment $\alpha_{s-1} = \alpha \upharpoonright l(s-1)$ of the characteristic sequence α of the set A under construction, at stage $s \geq 1$ of the construction the extension α_s of α_{s-1} is chosen so that A will meet f_s if f_s is dense along A and such that the extension does not lead to any f-gap. To achieve this, we first pick $n_0 > l(s-1), s$ such that, for all numbers $n \geq n_0$, $f(n) > 2(s+1)n$, and we let $\hat{\alpha}_s$ be the extension of α_{s-1} of length n_0 obtained by appending $1^{n_0 - l(s-1)}$. Moreover, if there is a string y with $|y| < s$ such that $f_s(\hat{\alpha}_s y)$ is defined then we let

$$\alpha_s = \hat{\alpha}_s \, y \, f_s(\hat{\alpha}_s y) \, 1$$

for the least such string y thereby meeting f_s at $n_0 + |y|$. To show that this extension does not introduce any f-gap into α observe that

$$\alpha_s = \alpha \restriction l(s-1)\; 1^{n_0 - l(s-1)}\; y\; f_s(\hat{\alpha}_s y)\; 1$$

i.e., the only zeroes occurring in the new part of α_s occur in $y\; f_s(\hat{\alpha}_s y)$. So it suffices to observe that $|y\; f_s(\hat{\alpha}_s y)| < f(n_0)$:

$$\begin{array}{rcll} |y\; f_s(\hat{\alpha}_s y)| & \leq & |y| + s\cdot|\hat{\alpha}_s y| & \text{(by (5.9))} \\ & < & s + s\cdot(n_0+s) & \text{(by } |y| < s \text{ and } |\hat{\alpha}_s| = n_0\text{)} \\ & < & 2(s+1)n_0 & \text{(by } n_0 > s \geq 1\text{)} \\ & < & f(n_0) & \text{(by choice of } n_0\text{).} \end{array}$$

Finally, if no string y as above exists, then we let $\alpha_s = \hat{\alpha}_s$. Note that in this case, by (5.9), f_s is not dense along A. Moreover, since the extension does not add any zeroes to α, this choice of α_s will not introduce any f-gaps. □

By combining Lemmas 5.37 and 5.38 we obtain the following characterization of the gaps occurring in all generalized Moore-generic sets.

Theorem 5.39 *Let $f : \mathbb{N} \to \mathbb{N}$ be nondecreasing and total. The following are equivalent.*

1. *$f \in O(n)$.*
2. *Every weakly GM-generic set has f-gaps.*
3. *Every GM-generic set has f-gaps.*
4. *Every weakly GNM-generic set has f-gaps.*
5. *Every GNM-generic set has f-gaps.*

The above established gaps imply that any weakly Moore generic set is REG-bi-immune. In fact we obtain CF-bi-immunity.

Corollary 5.40 *Let A be weakly GM-generic. Then A is bi-immune to the class of context free languages.*

PROOF. Since, as one can easily show, weak GM-genericity is closed under complement, it suffices to show that A is CF-immune. For a contradiction assume that B is an infinite context free subset of A. By the pumping lemma for context-free languages we can find a number p and words $x_n \in B \subseteq A$ such that $|x_{n+1}| = |x_n| + p$ ($n \geq 0$). Hence there is a number n_0 such that, for any $n \geq n_0$, $\Sigma^{[n,n+p]} \cap A \neq \emptyset$, where $\Sigma^{[n,n+p]} = \{x \in \Sigma^* : n \leq |x| \leq n+p\}$. Since $\Sigma^{[n,n+p]}$ has cardinality less than

$(p+1)2^{n+p} = (p+1) \cdot 2^p \cdot 2^n$ (and since $|\Sigma^{<n}| = 2^n - 1$), it follows that the characteristic sequence α of A does not have 2^{p+1} gaps. But this contradicts Lemma 5.37. □

Recall that the red-1-reg-generic sets introduced in Section 4.4 which were based on bounded red-extension functions were REG-bi-immune too (see Theorem 4.76). We conjecture, however, that a red-1-reg generic set in general is not CF-immune. This would show that the diagonalization power of (total) Moore extension functions is greater than that of the bounded partial regular red-extension functions.

We can modify the (generalized) Moore genericity concepts introduced above by considering red-extension functions of the corresponding type, i.e., (generalized) Moore functions for which the input is interpreted as a redundant representation of an initial segment in the sense of (4.44). Then, obviously, any generalized (nondeterministic) Moore function f can be simulated by a generalized (nondeterministic) red-Moore function f'. (Note that a finite automaton M' computing f' skips the parts $z_0\#,..., z_n\#$ on input $z_0\#i_0 \dots z_n\#i_n$ and simulates M on the remaining input $i_0 \dots i_n$. Also note, that in the non-generalized case such a simulation is not possible since M' has to produce an output bit for any letter it reads.) So (weak) red-GM-genericity and (weak) red-GNM-genericity imply (weak) GM-genericity and (weak) GNM-genericity, respectively. In fact, these variants of Moore genericity based on red-extension functions are strictly stronger than the corresponding genericity notions based on standard extension functions. This can be shown by analyzing the gaps occurring in the weakly red-GM-generic sets: Since a red-extension function gets in place of the initial segment $\alpha \upharpoonright n$ of length n the string $\alpha \upharpoonright_r n$ as an input, where $|\alpha \upharpoonright_r n|$ is of order $n \cdot \log n$, one can easily show that – in contrast to Theorem 5.39 – every weakly red-GM-generic set has gaps of order $n \cdot \log n$.

CHAPTER 6

Conclusion

In our thesis we have started the investigation of finite-state genericity. Our work focussed on the bounded case. Here we introduced a variety of genericity notions by distinguishing between total and partial extension functions and between extensions of fixed and of constant but arbitrary norm. By comparing the strength of the various concepts, by analyzing lower bounds on the complexity of the corresponding generic sets, and by relating these concepts to saturation we have illustrated the diagonalization power of finite automata in the setting of bounded finite extension arguments. By considering extension strategies which are either given only partial information on the initial segment which has to be extended or which are given this initial segment in a more redundant form we could further illustrate the diagonalization strength of finite automata. In particular, we have shown that the question whether we can force REG-bi-immunity by bounded finite-state extension strategies depends on the representation of initial segments.

Our treatment of unbounded finite-state genericity is less detailed. Here we introduced genericity concepts based on the common types of regular functions treated in the literature and established some of their basic properties. In part we also demonstrated that the more general function classes also lead to stronger genericity concepts but in this setting we left some of the basic questions open.

Possible future work on finite-state genericity might address the following questions. For a further understanding of the notions introduced in this thesis one might analyse further structural properties of the corresponding generic sets. In particular we have not addressed the question, which of the common structural properties based on regular reducibilities – like incompressibility, autoreducibility, hardness – are forced or avoided by the different finite-state genericity notions. For the genericity notions in computability and complexity theory the investigation of the corresponding questions proved to be very useful. In case of the regular languages, however, it seems that the corresponding reducibilities and their structural properties have not yet been more closely analyzed so that it seems that there is wide range of questions to be addressed here.

The strong dependence of the strength of finite-state extension strategies on the representation of the input initial segments leads to another type of questions, namely the question of the impact of changes of the representation of input and output. In case of the input we have addressed this problem in detail (see Section 4.4). For the case of output consider the following example. As discussed before, a Moore extension strategy has to produce the first bit of the extension already after reading the first bit of the given initial segment which imposes severe limitations on the possible strategies. We may avoid this by taking the mirror image of the value of the Moore function for defining the extension. Another approach which might lead to stronger or more robust finite-state genericity notions is to replace extension functions by extension relations (i.e., condition sets). In computability and complexity theory in general this approach is equivalent to the functional approach but in the low complexity setting of finite automata it might lead to stronger notions. In particular, in case of nondeterministic automata this approach might be useful.

A further area of research is the introduction of genericity notions for other low Chomsky classes. By the coincidence of the Chomsky-0 languages with the recursively enumerable sets, Chomsky-0 genericity coincides with the well understood and extensively studied 1-genericity concept of computability theory. Similarly, by the coincidence of the class of the context-sensitive languages with the nondeterministic space class NSPACE(n), genericity notions for this class may be obtained along the lines of the work on resource-bounded genericity in computational complexity (see Ambos-Spies (1996)) though most of the work there only deals with complexity classes extending DTIME(2^n). It seems, however, that nothing is known about adequate genericity notions for the class of the context-free languages (and the standard subclasses of CF like the deterministic context-free or linear languages). Here the development of genericity notions based on push down automata seems to be an interesting research direction which in part may build on our analysis of finite-state genericity. Also the results in the first part of our thesis on bi-immunity and on the Chomsky hierarchy of sequences might become useful here.

Bibliography

E. Allender, R. Beigel, U. Hertrampf, and S. Homer. Almost-everywhere complexity hierarchies for nondeterministic time. *Theoret. Comput. Sci.*, 115(2): 225–241, 1993.

K. Ambos-Spies. Resource-bounded genericity. In *Computability, Enumerability, Unsolvability*, volume 224 of *London Math. Soc. Lecture Note Ser.*, pages 1–59. Cambridge Univ. Press, 1996.

K. Ambos-Spies and E. Busse. Computational aspects of disjunctive sequences. In *Mathematical Foundations of Computer Science (Prague, 2004)*, volume 3153 of *Lecture Notes in Comput. Sci.*, pages 711–723. Springer, 2004.

K. Ambos-Spies, H. Fleischhack, and H. Huwig. Diagonalizations over polynomial time computable sets. *Theoret. Comput. Sci.*, 51(1-2):177–204, 1987.

K. Ambos-Spies, H. Fleischhack, and H. Huwig. Diagonalizations over deterministic polynomial time. In *Computer Science Logic (Karlsruhe, 1987)*, volume 329 of *Lecture Notes in Comput. Sci.*, pages 1–16. Springer, 1988.

K. Ambos-Spies, K. Weihrauch, and X. Zheng. Weakly computable real numbers. *J. Complexity*, 16(4):676–690, 2000.

J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural complexity. II*. EATCS Monographs on Theoretical Computer Science. Springer, 1990.

J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural complexity. I*. EATCS Monographs on Theoretical Computer Science. Springer, 1995.

J. L. Balcázar and U. Schöning. Bi-immune sets for complexity classes. *Math. Systems Theory*, 18(1):1–10, 1985.

C. Calude, L. Priese, and L. Staiger. Disjunctive sequences: an overview. CDMTCS Research Report 63, 1997.

C. Calude and S. Yu. Language-theoretic complexity of disjunctive sequences. *Discrete Appl. Math.*, 80(2-3):203–209, 1997.

S. Feferman. Some applications of the notions of forcing and generic sets: Summary. In *Proc. International Symposium on Theory of Models (Berkeley, 1963)*, pages 89–95. North-Holland, 1965.

S. A. Fenner. Notions of resource-bounded category and genericity. In *Proc. 6th Structure in Complexity Theory Conference*, pages 196–212. IEEE Comput. Soc. Press, 1991.

S. A. Fenner. Resource-bounded baire category: a stronger approach. In *Proc. 10th Structure in Complexity Theory Conference*, pages 182–192. IEEE Comput. Soc. Press, 1995.

P. Flajolet and J. M. Steyaert. On sets having only hard subsets. In *Automata, Languages and Programming (Saarbrücken, 1974)*, volume 14 of *Lecture Notes in Comput. Sci.*, pages 446–457. Springer, 1974.

H. Fleischhack. *On Diagonalizations over Complexity Classes*. Dissertation, Universität Dortmund, Dep. Comput. Sci. Tech. Rep. 210, 1985.

H. Fleischhack. p-genericity and strong p-genericity. In *Mathematical foundations of computer science (Bratislava, 1986)*, volume 233 of *Lecture Notes in Comput. Sci.*, pages 341–349. Springer, 1986.

J. G. Geske, D. T. Huỳnh, and A. L. Selman. A hierarchy theorem for almost everywhere complex sets with application to polynomial complexity degrees. In *Symposium on Theoretical Aspects of Computer Science (Passau, 1987)*, volume 247 of *Lecture Notes in Comput. Sci.*, pages 125–135. Springer, 1987.

G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers*. The Clarendon Press Oxford University Press, 1979.

P. G. Hinman. Some applications of forcing to hierarchy problems in arithmetic. *Z. Math. Logik Grundlagen Math.*, 15:341–352, 1969.

J. E. Hopcroft and J. D. Ullman. *Introduction to automata theory, languages, and computation*. Addison-Wesley Publishing Co., 1979.

C. G. Jockusch, Jr. Degrees of generic sets. In *Logic Colloq. of Recursion Theory: its Generalisation and Applications (Leeds, 1979)*, volume 45 of *London Math. Soc. Lecture Note Ser.*, pages 110–139. Cambridge Univ. Press, 1980.

C. G. Jockusch, Jr. Genericity for recursively enumerable sets. In *Recursion theory week (Oberwolfach, 1984)*, volume 1141 of *Lecture Notes in Math.*, pages 203–232. Springer, 1985.

H. Jürgensen and G. Thierrin. Some structural properties of ω-languages. *"Applications of Mathematics in Technology"*, pages 56–63, 1988.

S. C. Kleene and E. L. Post. The upper semi-lattice of degrees of recursive unsolvability. *Ann. of Math. (2)*, 59:379–407, 1954.

J. H. Lutz. Category and measure in complexity classes. *SIAM J. Comput.*, 19(6): 1100–1131, 1990.

W. Maass. Recursively enumerable generic sets. *J. Symbolic Logic*, 47(4):809–823 (1983), 1982.

E. Mayordomo. Almost every set in exponential time is P-bi-immune. *Theoret. Comput. Sci.*, 136(2):487–506, 1994.

K. Mehlhorn. On the size of sets of computable functions. In *14th Annual IEEE Symposium on Switching and Automata Theory (Iowa City, 1973)*, pages 190–196. IEEE Comput. Soc. Press, 1973.

W. Merkle and J. Reimann. On selection functions that do not preserve normality. In *Mathematical Foundations of Computer Science (Bratislava, 2003)*, volume 2747 of *Lecture Notes in Comput. Sci.*, pages 602–611. Springer, 2003.

J. Myhill. Category methods in recursion theory. *Pacific J. Math.*, 11:1479–1486, 1961.

P. Odifreddi. *Classical recursion theory*, volume 125 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., 1989.

J. C. Oxtoby. *Measure and category*, volume 2 of *Graduate Texts in Mathematics*. Springer, 1980.

E. L. Post. Recursively enumerable sets of positive integers and their decision problems. *Bull. Amer. Math. Soc.*, 50:284–316, 1944.

G. Rozenberg and A. Salomaa. *Handbook of Formal Languages*. Springer, 1997.

A. Salomaa. *Formal languages*. Academic Press, 1973.

C.-P. Schnorr and H. Stimm. Endliche Automaten und Zufallsfolgen. *Acta Informat.*, 1(4):345–359, 1971/72.

H. J. Shyr. Disjunctive languages on a free monoid. *Information and Control*, 34 (2):123–129, 1977.

R. I. Soare. *Recursively enumerable sets and degrees*. Perspectives in Mathematical Logic. Springer, 1987.

L. Staiger. Reguläre Nullmengen. *Elektron. Informationsverarbeit. Kybernetik*, 12 (6):307–311, 1976.

L. Staiger. Rich ω-words and monadic second-order arithmetic. In *Computer Science Logic (Aarhus, 1997)*, volume 1414 of *Lecture Notes in Comput. Sci.*, pages 478–490. Springer, 1998.

L. Staiger. How large is the set of disjunctive sequences? *J.UCS*, 8(2):348–362 (electronic), 2002.

S. Yu. Regular languages. In *Handbook of Formal Languages*, volume 1. Word, Language, Grammar, pages 41–110. Springer, 1997.