Growth Functions, Rates and Classes of String-Based Multiway Systems

Yorick Zeschke
Junior Research Affiliate
Wolfram Research, Inc.
yzeschke@gmail.com

Inspired by the recently emerging Wolfram Physics Project where "Multiway Systems," graph representations of abstract rewriting systems equipped with a causal structure, have played an important role in discrete models of spacetime and quantum mechanics, this paper establishes several more fundamental properties about the growth (number of states over steps in a system's evolution) of string rewriting systems in general. While proving the undecidability of exactly determining a system's growth function, we show several asymptotic properties all growth functions of arbitrary string rewriting systems share. Through an explicit construction, it is proven that string rewriting systems, while never exceeding exponential functions in their growth, are capable of growing arbitrarily slowly, that is, slower than the asymptotic inverse of every Turing-computable function. Additionally, an elementary classification scheme partitioning the set of string rewriting systems into finitely many nontrivial subsets is provided. By introducing arithmetic-like operations under which Multiway Systems form a weakened semiring structure, it is furthermore demonstrated that their growth functions, while always being primitively recursive, can approximate many well-known elementary functions classically used in calculus, which underlines the complexity and computational diversity of Multiway Systems. In the context of the Wolfram Physics Project, some implications of these findings are discussed as well.

Keywords: string rewriting systems; growth functions; Wolfram Physics Project

1. Introduction and Overview

String rewriting systems have been thoroughly researched in mathematical logic, proof theory and theoretical computer science for the past few decades [1, 2]. Recently however, these systems and the related hypergraph rewriting systems have been shown to be of significant interest in a fundamental physics context, as they possess various connections to theories of relativity, gravity and quantum mechanics [3–5], providing a new motivation for their investigation from a different perspective. A key trait of this perspective will be regarding

https://doi.org/10.25088/ComplexSystems.31.1.123

abstract rewriting systems as complex systems, as it is necessary to computationally simulate a given rewriting system to predict its states after a number of applications of its rewrite rules; that is, the behavior of an abstract rewriting system, or string rewriting system in our case, is not directly obtainable from its initial state and rewriting rules in general. Still, some attributes of these systems can be approximately predicted. In this paper, we prove several asymptotic statements about the growth (defined below) of graph representations of string rewriting systems that we call "string-based Multiway Systems" (see [6, Section 2.1] for a definition of string rewriting systems and [3, Definitions 10 and 11] for formal definitions of Multiway Systems).

While string rewriting systems were already investigated from a complex systems point of view by Wolfram in 2002 [7, pp. 204, 939], it was the Wolfram Physics Project (www.wolframphysics.org), launched in 2019, that thoroughly explored and established various links between Multiway Systems and group theory (see [7], notes for Section 5.6), homotopy type theory [8], category theory [5] and numerics of partial differential equations [9], setting aside the already mentioned connections to what the project calls "Wolfram Models," certain discrete formalisms for spacetime and quantum mechanics (see [10] for a general overview and [11] for a technical introduction, as well as the appendix of [5] for a glossary of terminology). In the physical framework of the actual Wolfram Model (see Section 2 in [4] for a formal definition), "hypergraph-based" Multiway Systems, that is, systems for which the "elements" or "objects" of the underlying abstract rewriting system are hypergraphs, are used. We consider string-based Multiway Systems instead because of their simpler structure, which makes them more easily amenable for mathematical analysis. Most likely, our results can be generalized to hypergraph-based Multiway Systems without much effort.

For related discrete complex systems such as cellular automata, various growth-related investigations have been conducted. For example, Brummit and Rowland give a systematic empirical analysis of the growth rates of boundaries of one-dimensional two-color cellular automata, classify the automata by their (approximate) growth exponent and construct a cellular automaton for which no such exponent can be determined [12], a methodology similar to the one used in this paper, although concerned with a different object of study. However, apart from utilizing the growth functions of specific string rewriting systems in proofs with other focus, as it was done, for example, in [13] or [14, Section 9], there have not been many investigations of growth functions of string rewriting systems in general, to our knowledge. Still, our results demonstrate how studying them yields some significant new insights into the general principles underlying abstract rewriting and strengthens the theoretical foundations of the Physics Project as well as the connections between discrete mathematics, theoretical physics and theoretical computer science (see Section 4).

Even despite our investigations being rather theoretical and aiming at understanding the mathematical background and structure of Multiway Systems in themselves, we comment on several potential applications in the Wolfram Physics Project in Section 4 and demonstrate our results by computational simulations of specific examples. Our visualizations have been made using Mathematica and all code for simulating Multiway Systems is available in the Wolfram Functions Repository. Readers interested in running simulations and visualizations themselves may find the documentation of the MultiwaySystem resource function to be useful references.

The subsequent subsections start by formally defining what we mean by Multiway System growth functions, rates and classes. Next, we investigate the boundaries of possible growth rates and find that Multiway Systems are, simply put, bounded in the speed but unbounded in the slowness of their growth rate (Theorem 1). After that, we show that the growth classes of the Multiway Systems we defined cover the entire set of Multiway Systems and apart from one trivially empty class, all of them contain infinitely many Multiway Systems (Theorem 2). To do this, we define arithmetic-like operations equipping the set of Multiway Systems with a semiring structure. Combining the two theorems, we conclude various interesting properties of the "computational diversity" and "-complexity" of Multiway Systems, showing that their growth functions constitute an interesting domain of further research.

1.1 Multiway Growth Functions

Consider a string-based Multiway System M [3, Definition 10], represented as a triplet $(R, s_{\text{init}}, \Sigma)$ where Σ is a finite alphabet, $R = \{r_1 \to t_1, \ldots, r_n \to t_n\}$ is a set of string replacement rules over Σ and $s_{\text{init}} \in \Sigma^*$ is the initial string where Σ^* denotes the set of all words over the alphabet Σ. We define the "state-set of generation n" as the set of all new (previously nonexistent) states added to the Multiway System in its n^{th} generation. These states are precisely the nodes of the states graph (cf. [11, Section 5.3]) to which the shortest path from the initial state has length n. In [11] they are called "merged states." Now, the "growth function" $g_M(n)$ is simply the cardinality of the state-set of generation n (see Figure 1). Note that we use the terms "sequence" and "function" interchangeably for functions $f : \mathbb{N}_+ \to \mathbb{N}_+$.

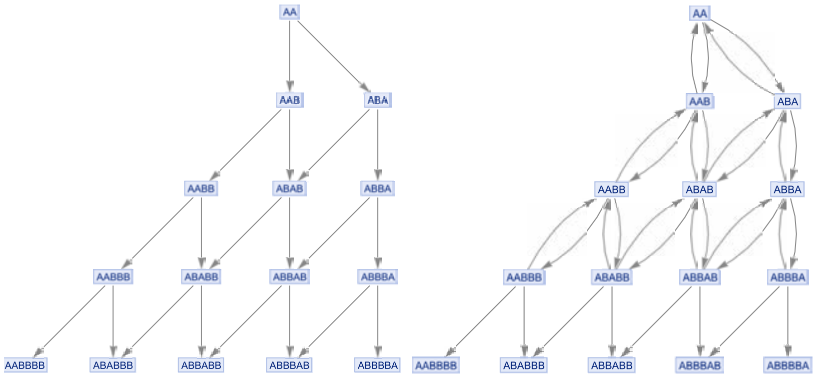


Figure 1. Both (a) $M_1 = (\{"A" \to "AB"\}, "AA", \{A, B\})$ and (b) $M_2 = \{"A" \to "AB", "AB" \to "A"\}, "AA", \{A, B\})$ have the same growth function $g(n) = n$ because cycles in the states graph do not lead to new states.

In general, it is very hard or even undecidable (see Section 3) to prove that some Multiway System has a certain growth function. It is also not obvious that the growth functions of Multiway Systems should be elementary functions or "simple" by any other definition. Examples of systems where the growth function is hard to describe were already given in *A New Kind of Science* [7, pp. 204 ff.]. Therefore, we will approximate the growth functions of Multiway Systems by continuous, strictly monotonically increasing, unbounded (and hence bijective on $\mathbb{R}_{\geq 0}$) functions, which can be analyzed more easily. This way, many similar growth functions will be considered members of the same equivalence class. We will say that the corresponding Multiway Systems have the same "growth rate."

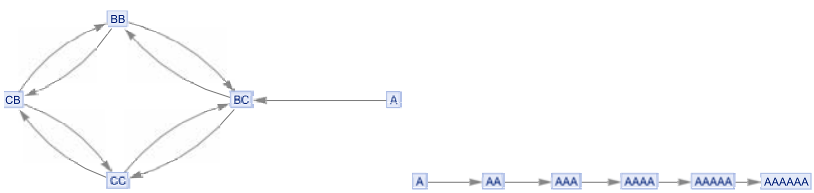
1.2 Multiway Growth Rates

To formalize the notion of approximating functions, we use the asymptotic growth classes from complexity theory, defined in the following way: For a function $f : A \to B$ where $A = B = \mathbb{R}_{\geq 0}$ or $A = B = \mathbb{R}$, $O(f)$ is defined as the set of functions $g : A \to B$ for which $\limsup_{x\to\infty}\left|\frac{g(x)}{f(x)}\right|$ exists and is a real number. An equivalent definition is $g \in O(f) \Longleftrightarrow \exists\, C \in \mathbb{R}_+ : \exists\, x_0 \in A : \forall\, x > x_0 : C|f(x)| \geq |g(x)|$ but we use the previous one for convenience. The subset of $O(f)$ for which the limit superior is zero is denoted $o(f)$. Similarly, $\Omega(f) := \{g : A \to B \,|\, f \in O(g)\}$ and $\omega(f) := \{g : A \to B \,|\, f \in o(g)\}$. Finally, $\Theta(f) := \Omega(f) \cap O(f)$. It is straightforward to show that $f \sim_\Theta g \Longleftrightarrow f \in \Theta(g)$ is an equivalence relation. Thus, we may speak of functions that are "asymptotically equal."

As already mentioned, we want to approximate growth functions by bijective functions for the subsequent mathematical analysis. For some Multiway growth function a, we will define the sequences $\overline{a}$ and $\underline{a}$ as its tightest upper and lower bounds that are monotonically increasing, even if a itself is not monotonic at all. From these sequences, we will then construct two equivalence classes of continuous functions that are all asymptotically equal to and hence "close approximations" of $\overline{a}$ or $\underline{a}$ respectively. Two representatives of these classes will be called "tight bounds" and, since we are generally concerned with unbounded growth functions (bounded growth functions will be discussed shortly), both are bijective on $\mathbb{R}_{\geq 0}$.

Notice that we have only defined asymptotic growth classes for functions on $\mathbb{R}$ or $\mathbb{R}_{\geq 0}$. However, since the Multiway growth function is always a function on $\mathbb{N}_+$, we consider its linear interpolation, a continuous function from $\mathbb{R}_{\geq 0}$ to $\mathbb{R}_{\geq 0}$ that is equal to the sequence for natural arguments and always bounded by consecutive values of the sequence (see Definition 4), instead.

Definition 1. Let M be a Multiway System and g_M its growth function. We call M "finite" if $\exists\, n \in \mathbb{N}_+ : g_M(n) = 0$ (as this implies that at a certain point, no further states will be added). We call M "bounded" if $\exists\, b \in \mathbb{N}_+ : \forall\, n \in \mathbb{N}_+ : g_M(n) \leq b$ and M is not finite. Systems that are neither finite nor bounded are called "unbounded" (see Figure 2).



(a) (b)

Figure 2. (a) States graphs of the finite system $M_1 = (\{"A" \to "BC", "B" \to "C", "C" \to "B"\}, "A", \{A, B\})$ and (b) first six steps of the bounded system $M_2 = (\{"A" \to "AA"\}, "A", \{A\})$. Notice that $\forall\, n \in \mathbb{N}_+ : g_{M_2}(n) = 1$, despite the fact that the rule can be applied in many different positions, because we are only considering merged states.

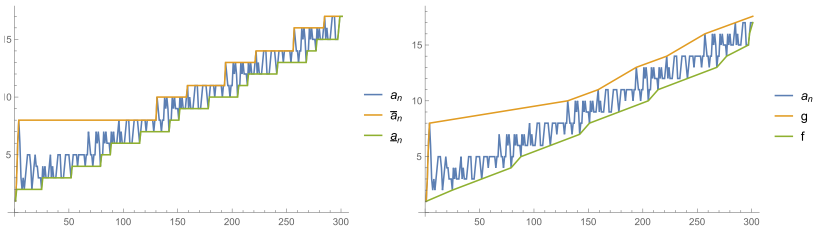
Definition 2. Let $a : \mathbb{N}_+ \to \mathbb{N}_+$ be the growth function of an unbounded Multiway System and let

$$\overline{a}_n := \max(\{a_k \mid k \leq n\})$$

and

$$\underline{a}_n := \max(\{a_k \mid k \le n \wedge (\forall\, l \ge k : a_l \ge a_k)\} \cup \{1\}).$$

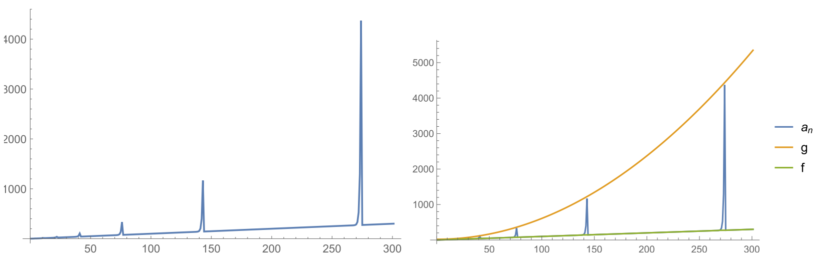
We call two continuous functions $f, g : \mathbb{R}_{\ge 0} \to \mathbb{R}_{\ge 0}$ “tight bounds of a” if $f \in \Theta(L_{\mathbb{N}_+}(\overline{a})) \wedge g \in \Theta(L_{\mathbb{N}_+}(\underline{a}))$ where $L_{\mathbb{N}_+}$ denotes the linear interpolation over $\mathbb{N}_+$ (according to Definition 4). See Figure 3 for a visual illustration.



(a) (b)

Figure 3. The growth function a_n of $M = (\{"AB" \to "", "ABA" \to "ABBAB", "ABABBB" \to "AAAAABA"\}, "ABABAB", \{A, B\})$ together with $\overline{a}_n$, $\underline{a}_n$ and a pair (f, g) of tight bounds. Note: Only f and g are continuous; the other lines are drawn for visual appearance.

One might ask why we introduce an upper and a lower bound instead of approximating the growth function with a single function. Doing so would, however, be a poor approximation, as there are Multiway Systems for which even the tightest upper and lower bounds are never in the same asymptotic growth class (compare Figure 4). We will call these Multiway Systems “strongly oscillating” and all others



(a) (b)

Figure 4. For this special system, we can prove (see Section 3 for an explanation) the tight bounds f, g to be in different asymptotic growth classes: $f \in \Theta(x)$ and $g \in \Theta(x^2)$. Note again that only f and g are continuous, while a_n is drawn as a line for visual appearance.

(i.e., systems where all tight bounds are asymptotically equal) "regular." Notice that for every regular Multiway System, a pair of bijective tight bounds exists because its tight bounds will be in the asymptotic equivalence class of two unbounded strictly monotonically increasing functions and tight bounds are continuous on $\mathbb{R}_+$. Strongly oscillating systems, on the other hand, are much more difficult to analyze since we cannot easily come up with criteria for measuring the rate of oscillation and it is not clear at all whether there has to be any periodicity or regularity in the way in which they oscillate. Thus, for our basic investigations about the fundamental structure of Multiway Systems, we will focus on regular systems.

Definition 2 suggests a natural way to define classes of Multiway Systems with "similar" growth functions by considering growth functions with approximately equal tight bounds as equivalent. Let $f, g : \mathbb{N}_+ \to \mathbb{N}_+$ be functions and $(a_1, b_1), (a_2, b_2)$ be tight bounds of f and g, respectively. We define $\sim_R$ by $f \sim_R g \Longleftrightarrow (a_1 \sim_\Theta a_2 \wedge b_1 \sim_\Theta b_2)$. Since tight bounds always exist, $\sim_R$ is an equivalence relation because $\sim_\Theta$ is one. For some Multiway System M with growth function g_M, we call the equivalence class of $\sim_R$ that g_M falls into the "growth rate" of g_M (or sometimes the growth rate of just M).

It is obvious that every Multiway System has exactly one growth function and exactly one growth rate. The converse, that is, that every function $f : \mathbb{N}_+ \to \mathbb{N}_+$ is the growth function of some Multiway System or, respectively, that every pair of bijective functions on $\mathbb{R}_{\geq 0}$ is a pair of tight bounds of some Multiway growth function, is clearly not true, as emphasized in Lemma 1. However, if we define much more general classes of growth functions, which we will call "Multiway growth classes," we will see (in Theorem 2) that they indeed partition the set of all Multiway Systems into a finite set of infinite subsets.

1.3 Multiway Growth Classes

To further distinguish between types of Multiway Systems on a more abstract level and demonstrate which kinds of growth functions can be achieved, we want to define very broad classes of Multiway Systems whose growth functions show similar behavior on a large scale. We have already distinguished among finite, bounded and unbounded systems, as well as dividing the latter into regular and strongly oscillating systems. As outlined earlier, we will focus on regular systems. To group these into sets of systems of similar behavior, we use commonly known classes of functions such as polynomial or exponential functions or, more precisely, functions bounded by polynomials or exponentials, as well as intermediately (faster than polynomial and slower than exponential) growing functions and some others.

More precisely: Let G_{pol} be defined as the set of all continuous bijections $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ that satisfy $f \in \Omega(x^n) \cap O(x^{n+1})$ for some $n \in \mathbb{N}_+$ and define $G_{\exp}$ as

$$\{f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \,|\, f \in \Omega(a^x) \cap O((a+1)^x)\}$$

for some $a \in \mathbb{N}_{>1}$. Similarly, let $G_{\sup\exp}$ be the set

$$\left\{f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \,\middle|\, \forall\, g \in G_{\exp} : f \in \omega(g)\right\}$$

where f must be continuous and bijective. Additionally, denote by G_{int} the set of all continuous bijections $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ fulfilling $\forall\, g \in G_{\text{pol}}, h \in G_{\exp} : f \in \omega(g) \cap o(h)$. Now, it is easy to analogously define

$$G_{\text{inv pol}} := \left\{f \,\middle|\, f^{-1} \in G_{\text{pol}}\right\},$$

$$G_{\text{inv exp}} := \left\{f \,\middle|\, f^{-1} \in G_{\exp}\right\},$$

$$G_{\text{inv sup exp}} := \left\{f \,\middle|\, f^{-1} \in G_{\sup\exp}\right\}$$

and

$$G_{\text{inv int}} := \left\{f \,\middle|\, f^{-1} \in G_{\text{int}}\right\}.$$

These eight sets give a partition of the set of all continuous bijections on $\mathbb{R}_{\geq 0}$ because a function grows either slower than $f : x \mapsto x$, in which case its inverse grows faster than f, or it grows faster (or equal to) f, in which case it is contained in one of the first four classes. We call a Multiway System a member of the growth class C_i if its growth function has tight bounds f, g belonging to G_i. Note that this definition applies only to regular Multiway Systems. Also, as Definition 2 is not applicable for finite or bounded Multiway Systems, we handle them separately.

Let C_{fin} and C_{bnd} be the sets of all finite and bounded Multiway Systems, respectively. For all Multiway Systems in $C_{\text{fin}} \cup C_{\text{bnd}}$, the growth rate is defined to be (1, 1). While finite and bounded systems can have quite an intricate structure, their growth functions are not very interesting for our purposes. As a side note, it should be remarked that what seems to be "complex behavior" in a *New Kind of Science*-fashion (compare [7]), occurred much more frequently in our empirical investigations of finite systems, but these observations are far from conclusive. They might be useful for applications not directly related to the Wolfram Physics Project, but in this paper, they will not be discussed in great detail.

As every continuous bijection on $\mathbb{R}_+$ belongs to exactly one of the G_i, every Multiway System that can be imagined is either strongly oscillating or in one of those classes (including C_{fin} and C_{bnd}). We will furthermore show in Theorem 2 that every one of these classes (except $C_{\text{sup exp}}$, which is empty by Lemma 1) contains infinitely many Multiway Systems.

Summarizing the previous section, we introduced the three main concepts of Multiway growth functions, Multiway growth rates and Multiway growth classes. We will now present the first important result of this paper, a theorem about the boundaries of possible growth rates, and spend the next section proving and illustrating it.

2. The Spectrum of Possible Growth Rates

Having defined Multiway growth rates, we might ask ourselves which growth rates are possible, that is, how the equivalence classes of $\sim_R$ are distributed in the set of all possible pairs of bijective functions on $\mathbb{R}_{\geq 0}$. First of all, it is quite easy to give an upper bound for growth rates that can be achieved. In fact, no Multiway System can grow faster than exponentially.

Lemma 1. Let (f, g) be the growth rate of some Multiway System. There exists some constant $c \in \mathbb{R}$ for which $f, g \in o(e^{cx})$.

Proof. Denote by $s_{\max}(n)$ the maximum string length that states of generation n can have. For every Multiway System $M = (R, s_{\text{init}}, \Sigma)$, the set of rules remains constant during the whole evolution, so $s_{\max}$ can at most increase constantly, that is, $s_{\max} \in O(n)$. Since the number of words with length l is given by $|\Sigma|^l$, the growth function $g_M(n)$ will never exceed $|\Sigma|^{s_{\max}(n)} = e^{\ln(|\Sigma|) s_{\max}(n)} \in \Theta(e^{cn})$ and the claim follows. □

So what about a lower bound for Multiway growth rates? Formally, a trivial Multiway System with no rules and thus only one state has the lowest possible growth function by pointwise value comparison. In general, "terminating" or "constant" asymptotic growth functions of finite or bounded Multiway Systems (which have the growth rate (1, 1)) are the slowest by means of asymptotic comparison, that is, with respect to the total ordering $\leq_O$ defined by $f \leq_O g \Longleftrightarrow f \in O(g)$, but examples like this are not very illuminating. Therefore, we might ask what the slowest growth rate faster than constant is, that is, what the smallest (by asymptotic comparison) functions $f, g \in \omega(1)$ are for which (f, g) is the growth rate of some Multiway System. It turns out

https://doi.org/10.25088/ComplexSystems.31.1.123

however, that no such smallest growth rate exists, which means that Multiway Systems can, in a certain sense, grow arbitrarily slowly.

To understand why this is the case and make the even stronger statement that Multiway Systems can grow slower than all computable functions (see Corollary 2), we need to introduce a couple of constructions. First of all, we will show how Multiway Systems can emulate Turing machines, meaning there are systems such that the successive states of their evolution correspond to steps in the machine's evaluation. Phrased differently, it is possible to construct a Multiway System that has exactly one new state for $T(n)$ steps, where $T(n)$ is the number of operations that a certain Turing machine $\mathcal{T}$ carries out before halting when provided with the input n. Such a machine, with some additional constraints, will be called a "T-halter" and T its "halting function."

By adding some specific rules to the Multiway System that emulates $\mathcal{T}$, it will be possible to evaluate $\mathcal{T}$ indefinitely for increasing inputs $n = 1, 2, \ldots$. Additionally, the Multiway System will be constructed in a way such that every time the underlying Turing machine is "started again" on the next input, the number of new states per time step is increased by one. This way, we will obtain a growth function informally described by the sequence "n occurs $T(n)$ times" (see Definition 3), for example, the sequence "n occurs n times," which would be given by 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, ... We will then show that this growth function is approximated by the inverse of the linear interpolation (see Definition 4) over the summatory function of T (see Lemma 2). From this we conclude the following theorem:

Theorem 1. Let $T \in \Omega(n)$ be the halting function of some Turing machine (see Figure 5). There is a Multiway System with growth rate (a, b) such that $a, b \in O(T_*^{-1})$ where $T_*(x) = L_{\mathbb{N}_+}\left(\sum_{k=1}^{n} T(k)\right)$ and $L_{\mathbb{N}_+}$ denotes the linear interpolation over $\mathbb{N}_+$ (see Definition 4).

2.1 Proof of Theorem 1

Now let us formalize the proof outlined in the preceding section. For some function $T : \mathbb{N} \to \mathbb{N}$, we define a "$T$-halter" to be a Turing machine $\mathcal{T}$ such that $\mathcal{T}$ executes precisely $T(n)$ operations when given the input n, taking into account the input and output constraints depicted in Figure 6(a). These constraints will later allow us to "enchain" the Multiway Systems corresponding to T-halters. Of course, neither is there a T-halter for arbitrary T nor must there be a unique T-halter for a given T. However, and this is the part we care about, there is a T-halter such that $T \in \Omega(f)$ for any computable function f because we can just take the Turing machine that computes f and add some logic to write $n + 1$ after the computation is finished.

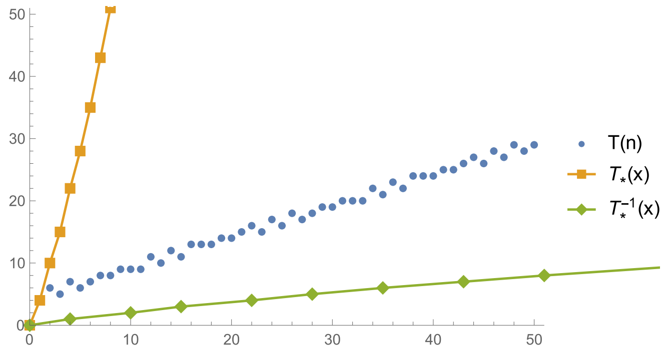
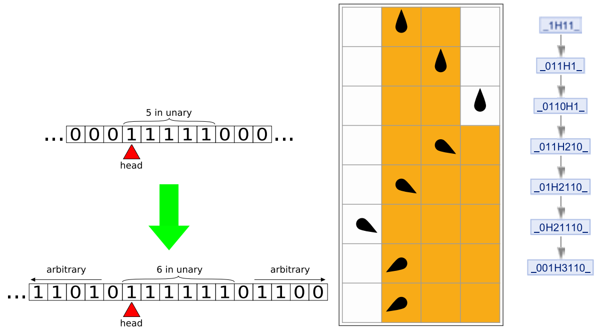


Figure 5. Graphical illustration of Theorem 1. The theorem asserts that there is a Multiway System for which the growth rate (f, g) is asymptotically less than $T_*^{-1}(x)$.



(a) (b)

Figure 6. Visual explanation of the T-halter constraints and Turing machine plot with an associated Multiway System plot. (a) $\mathcal{T}$ is always started on an empty tape containing only n in unary representation. After halting, $\mathcal{T}$ is required to have written $n+1$ in unary on the tape and placed its head onto or left of the first digit. The number $n+1$ must be preceded and followed by at least one empty symbol. (b) Evolution of $\mathcal{T}_1$ (compare Figure 7(a)) next to the evolution of M_1. The position and orientation of the "droplet" symbol indicate the machine's head position and state, respectively.

Having defined T-halters, the next step is to show how Multiway Systems can emulate these (and all other Turing machines). Since we are talking about deterministic Turing machines, no branching shall occur in the corresponding Multiway System; that is, the system should have exactly one state in generation n that corresponds to the state of the Turing machine after $n-1$ operations. Because at every state in the machine's evolution only a finite part of the tape contains non-blank symbols, we include only the symbols already "touched" by the machine (meaning the head was on that symbol at least once) in the states of the Multiway System and abbreviate the infinite strings of zeros on both sides of the tape with an underscore. The position and state of the head are indicated by an H right of the symbol the head is currently on, followed by the current state number. Hence, there are four additional symbols (two underscores, an H and a number) used in the Multiway System but not written on the machine's tape (see Figure 6(b)).

Using this representation, plain read/write operations and state changes would be straightforward to implement as replacement rules, as we could just introduce a rule "xHn" → "yHm" for every combination of currently read symbol x and head state n (writing symbols next to each other in this context simply denotes their concatenation to a string). However, since the head must move left or right after each such operation, those rules are not suitable. What is needed instead to encode the operation "when x_1 is read in state n, write y_1, change state to m and move the head right," is a rule of the form "x_1Hnx_2" → "y_1x_2Hm" for every possible value of x_2. Similarly, a left move of the head is encoded as "x_2x_1Hn" → "x_2Hmy_1". If n is not a halting state (in which case we would not need any rules), exactly one of these two rule patterns will be applicable for every possible value of x_1 or, respectively, every state transition arrow starting at H in the state transition diagram.

For a Turing machine with N states working on an alphabet of S symbols, that already gives a worst-case (no halting states) of $N \cdot S^2$ rules in the corresponding Multiway System. However, $N \cdot S$ (worst case) more rules have to be added to handle the literal "edge cases" in which the head is next to one of the underscores bounding the tape. The two rule patterns, of which, as before, exactly one will match for every state transition arrow, are "$\_xHn$" → "$\_0Hmy$" for a left move and "$xHn\_$" → "$y0Hm\_$" for a right move (in both cases x is read and y written). Now, the resulting rule set captures all of the Turing machine's properties and is able to extend the tape to any required length by itself. As an initial state of the Multiway System to emulate the machine, any string of characters from the machine's alphabet together with an "Hs", where s is the starting state, and the two bounding underscores can be used.

To illustrate this construction, consider the Turing machine $\mathcal{T}_1$ shown in Figure 7(a). It is, in some sense, the easiest possible T-halter for it does nothing more than increase the number on the tape by one and place its head back at the beginning. Figure 6(b) shows the successive states of the machine (and tape) next to a Multiway System M_1 emulating $\mathcal{T}_1$. The rule set for this specific instance is

```
{"00H1" -> "0H21", "10H1" -> "1H21", "1H10" -> "10H1",
"1H11" -> "11H1", "01H2" -> "0H21", "11H2" -> "1H21",
"0H20" -> "00H3", "0H21" -> "01H3", "_0H1" -> "_0H21",
"1H1_" -> "10H1_", "_1H2" -> "_0H21", "0H2_" -> "00H3_"}
```

Now it is clear that a Multiway System emulating some T-halter $\mathcal{T}$ has exactly one state for $T(n)$ generations when started with the initial condition "$\_1H11^{n-1}\_$" (1^{n-1} denotes $n-1$ times the symbol 1, the first 1 is left of the head and the head starts in state 1). Now, we need a way to "enchain" this Multiway System with itself. To achieve this, first add the rules "Hf" → "X" for every halting state f where X is one fixed symbol not contained in $\mathcal{T}$'s alphabet. These additional rules will cause the multiway state in generation $T(n)+1$ to look like "$\_w_1 01X1^n 0w_2\_$" (if the head is left of the first digit, "$\_w_1 0X1^{n+1} 0w_2\_$" works analogously) where w_1 and w_2 are arbitrary words that might be created as byproducts in the working of $\mathcal{T}$. This is due to the T-halter constraints depicted in Figure 6(a), or rather, the T-halter constraints were chosen precisely to cause such a configuration of the tape.

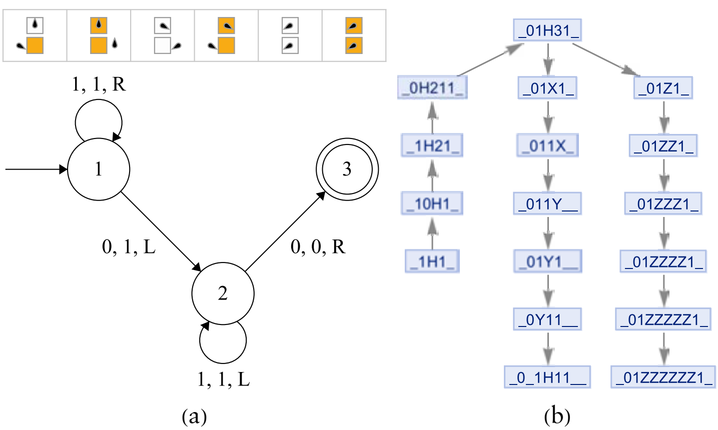


Figure 7. Two different representations of the same computational system: a Turing machine and a Multiway System. (a) Rule plot and state transition diagram of $\mathcal{T}_1$. The arrow labels indicate "read, write, move". (b) States graph of the Multiway System constructed from $\mathcal{T}_1$ (see below).

Adding the rules "$X1$" → "$1X$", "$X0$" → $\_0$", "$1Y$" → "$Y1$" and "$0Y1$" → "0_1H1" will cause exactly one state where the X has "moved" one position to the right for n generations (first rule), then add an underscore behind the n ones (second rule), "move back to the left" using the Y for $n+1$ generations (third rule) and finally add an underscore at the left side, replacing Y by the starting state symbol

"$H1$" of $\mathcal{T}$ (see Figure 7(b)). Now, the whole process can start again because the new underscores ensure a "fresh" new tape for $\mathcal{T}$, which now contains, by the T-halter constraints, $n+1$ as the next input for $\mathcal{T}$ to continue with while everything outside the bounding underscores will be ignored.

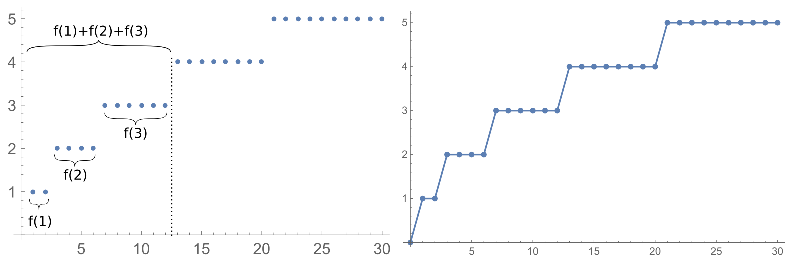
The resulting Multiway System of this continued reevaluation of $\mathcal{T}$ will run indefinitely, subsequently running instances of $\mathcal{T}$ with larger and larger values of n. Despite that, the Multiway System still has only one state in all generations. In order to make the number of states increase exactly when n increases; that is, when some instance of $\mathcal{T}$ has finished working, we add the rules "$0Y1$" $\to$ "Z" and "Z" $\to$ "ZZ". This way, the Multiway states graph branches every time the system starts a new instance of $\mathcal{T}$ into a main branch where the evaluation of $\mathcal{T}$ continues and a diverging branch where the second rule just creates longer and longer strings of Z forever, constantly adding one new state to every generation. Thus, the number of divergent branches is always equal to $n-1$ and these branches grow constantly forever, causing the desired behavior as shown in Figure 7(b).

First, it takes a "preparation time" of $p(n) = 2(n+1)+1$ (or $p(n) = 2(n+2)$ if the head starts at the left of the first digit instead) steps because the head moves over $n+1$ symbols including the new 1 before iteration $n+1$ of $\mathcal{T}$ can start after the n^{th} iteration is done. Thus, there will be n states for $T(n)+p(n)$ steps in this Multiway System construction, before the number of states increases by one. Let us generally investigate the sequences obtained this way:

Definition 3. Let $f: \mathbb{N}_+ \to \mathbb{N}_+$ be a function. The sequence "n occurs $f(n)$ times" is defined by $A_f\left(\sum_{k=1}^{n} f(k)\right) = A_f\left(m + \sum_{k=1}^{n} f(k)\right) = n$ for all $n, m \in \mathbb{N}$ with $n \geq 1 \wedge m < f(n+1)$.

Definition 4. Let $f: \mathbb{N}_+ \to \mathbb{N}_+$ be a function and $S \subseteq \mathbb{N}_+$ an infinite set. The "linear interpolation of f over S," denoted $L_S(f)$, is defined as the polygonal chain starting at $(0, 0)$ and passing through all points $(n, f(n))$, $n \in S$ ordered by n.

Since Definition 3 requires f to always be greater than zero, every natural number can be represented as some sum over consecutive values of f plus a remainder and, as Figure 8(a) shows, this definition indeed matches the informal description of "n occurs $f(n)$ times." Notice as well that the linear interpolation, despite being defined as a curve in $\mathbb{R}^2$, can be regarded as a continuous function from $\mathbb{R}_{\geq 0}$ to $\mathbb{R}_{\geq 0}$ because for all $n \in S$, the function to be interpolated assigns precisely one y value, and since S is an infinite subset of $\mathbb{N}$, the linear interpolation function is defined everywhere on $\mathbb{R}_{\geq 0}$.



(a) (b)

Figure 8. Plots for illustrating Definitions 3 and 4. (a) Example for Definition 3: the sequence "n occurs $f(n) = 2n$ times" (A_{2n}). (b) Example for Definition 4: the linear interpolation of A_{2n} over $\mathbb{N}_+$, now a continuous function from $\mathbb{R}_{\geq 0}$ to $\mathbb{R}_{\geq 0}$.

Now, to express some sequence A_f explicitly, define the set of increase indices of A_f as $I(A_f) := \{n \in \mathbb{N}_+ \mid A_f(n-1) < A_f(n)\}$. It follows that $L_{I(A_f)}(A_f)$ will always be strictly monotonically increasing and unbounded. Therefore, its inverse function $L_{I(A_f)}(A_f)^{-1}$ exists and we can formulate the following lemma:

Lemma 2. For all functions $f : \mathbb{N}_+ \to \mathbb{N}_+$, the following identity holds: $L_{I(A_f)}(A_f)(x) = L_{\mathbb{N}_+}(\sum_{k=1}^{n} f(k))^{-1}(x)$.

Proof. For readability, let T_σ be the function $\sum_{K=1}^{n} T(k)$, $\varphi(x) := L_{I(A_f)}(A_f)(x)$ and $\psi(x) := L_{\mathbb{N}_+}(\sum_{k=1}^{n} f(k))(x)$. By Definition 4, the linear interpolation of a function equals that function on the interpolation set, so

$$\forall\, n \in \mathbb{N}_+ : (\varphi \circ \psi)(n) = \varphi\left(\sum_{k=1}^{n} f(k)\right) = n.$$

For values $x \in (n, n+1)$, $n \in \mathbb{N}$, the linear interpolation gives

$$\begin{aligned}\psi(x) &= \frac{\Delta y}{\Delta x}(x-n) + \psi(n) = \frac{\psi(n+1) - \psi(n)}{n+1-n}(x-n) + \psi(n) \\ &= (\psi(n+1) - \psi(n))(x-n) + \psi(n).\end{aligned} \tag{1}$$

Letting $y = \psi(x)$, we know that

$$(\varphi \circ \psi)(x) = \varphi(y) = \frac{\varphi(y_2) - \varphi(y_1)}{y_2 - y_1}(y - y_1) + \varphi(y_1)$$

for some $y_1, y_2 \in I(A_f)$ where $y_1 < y < y_2$ and y_1, y_2 are the values in $I(A_f)$ closest to y. Since ψ is strictly monotonically increasing, y_1 and y_2 must be given by $\psi(n)$ and $\psi(n+1)$, respectively. Thus,

$$\begin{aligned}
(\varphi\circ\psi)(x) &= \frac{\varphi(\psi(n+1)) - \varphi(\psi(n))}{\psi(n+1) - \psi(n)}(\psi(x) - \psi(n)) + \varphi(\psi(n)) \\
&= \frac{n+1-n}{\psi(n+1) - \psi(n)}(\psi(x) - \psi(n)) + n \\
&= \frac{(\psi(n+1) - \psi(n))(x-n) + \psi(n) - \psi(n)}{\psi(n+1) - \psi(n)} + n \qquad \text{by equation (1)} \\
&= x - n + n = x.
\end{aligned}$$

So φ is a left-inverse of ψ on $\mathbb{R}_{\geq 0}$. Analogously, it can be shown that φ is also a right-inverse of ψ, so indeed, $L_{I(A_f)}(A_f)$ and $L_{\mathbb{N}_+}(\sum_{k=1}^{n} f(k))$ are inverse functions. □

Putting it all together, we conclude from the previous Turing machine investigation that for every halting function $T \in \Omega(n)$, there is a Multiway System that has the growth function $g_M(n) = A_{T+p}(n)$ for some $p \in \Theta(n)$. Additionally, $p \in \Theta(n)$, so g_M will even become strictly less than $L_{\mathbb{N}_+}(T_\sigma)^{-1}$ very soon. In most practical cases, g_M is much lower. Hence, p "delays" the growth function even more; that is, $\forall\, n \in \mathbb{N}_+ : A_{T+p}(n) \leq A_T(n)$, and it follows that

$$\begin{aligned}
&L_{I(g_M)}(g_M)(n) \leq L_{I(A_T)}(A_T)(n) = L_{\mathbb{N}_+}(T_\sigma)^{-1}(x) \\
&\qquad \Rightarrow g_M(n) \in O\left(L_{\mathbb{N}_+}\left(\sum_{k=1}^{n} T(k)\right)^{-1}\right)
\end{aligned}$$

by Lemma 2, which concludes the proof of Theorem 1.

2.2 Applications of Theorem 1

Computing linear interpolations and their inverse functions seems hard to do analytically because in most cases, there are no elementary closed-form expressions describing them. Therefore, it might seem difficult to actually apply Theorem 1. However, since we are only interested in growth rates, we can use approximations to make calculations much easier.

Lemma 3. If $f : \mathbb{N}_+ \to \mathbb{N}_+$ is strictly increasing, $g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is continuous and bijective and $\forall\, n \in \mathbb{N}_+ : g(n) = f(n)$, then $L_{\mathbb{N}_+}(f)^{-1} \in \Theta(g^{-1})$.

Proof. Because g is continuous and takes the same values as f for natural arguments, we know that

$$\forall\, n \in \mathbb{N}_+ : \forall\, x \in (n, n+1) : f(n) \leq g(x) \leq f(n+1).$$

Using the fact that the linear interpolation equals the function for natural arguments, this equation becomes $L_{\mathbb{N}_+}(f)(n) \leq g(x) \leq L_{\mathbb{N}_+}(f)(n+1)$. This implies $L_{\mathbb{N}_+}(f)^{-1}(y_1) \leq g^{-1}(y) \leq L_{\mathbb{N}_+}(f)^{-1}(y_2)$ for values $y \in (y_1, y_2)$ where $y_1 = f(n)$ and $y_2 = f(n+1)$. Expanding out gives

$$\begin{aligned} &L_{\mathbb{N}_+}(f)^{-1}(f(n)) \leq g^{-1}(y) \leq L_{\mathbb{N}_+}(f)^{-1}(f(n+1)) \\ &\qquad \Rightarrow n \leq g^{-1}(y) \leq n+1, \end{aligned}$$

which means that the difference of $g^{-1}(y)$ and $L_{\mathbb{N}_+}(f)^{-1}(y)$ is always bounded by 1. Therefore, $g^{-1} \in \Theta\left(L_{\mathbb{N}_+}(f)^{-1}\right)$. □

Whenever it is possible to express some halting function in a closed form (e.g., $T : \mathbb{N}_+ \to \mathbb{N}_+, n \mapsto 2n^2 + 3n$) that could also describe a bijective function on $\mathbb{R}_{\geq 0}$ (like $f(x) = 2x^2 + 3x$), we can use Lemma 3 to simplify calculations: Since f is monotonically increasing and the linear interpolation equals the summatory function for natural arguments, we have

$$\begin{aligned} &\sum_{k=0}^{\lfloor x \rfloor} f(k) \leq \int_0^x f(t)\,\mathrm{d}t \leq \sum_{k=1}^{\lceil x \rceil} f(k) \\ &\Rightarrow L_{\mathbb{N}_+}(T_\sigma)(\lfloor x \rfloor) \leq F(x) \leq L_{\mathbb{N}_+}(T_\sigma)(\lceil x \rceil). \end{aligned} \tag{2}$$

Therefore, Lemma 3 tells us that we can approximate the inverse of the summatory function used in Theorem 1 just by computing the inverse of the integral of f. Especially in the case of logarithms or exponential functions, solving integrals is much easier than computing sums, so this lemma can be very useful.

To demonstrate this and assist with the proof of Theorem 2, let us imagine we wanted to construct a Multiway System of logarithmic growth rate. We can approach this problem by designing a T-halter for some $T_{\exp} \in \Theta(2^n)$ and implementing the construction described in Section 2 to get a Multiway System with the inverse growth rate. As an example, take the Turing machine $\mathcal{T}_{\exp}$ shown in Figure 9. It is started in state 1, which simply moves the head to the right end of the word on the tape and changes to state 2. In this state, the head moves left again, replacing 2 by 1 until it encounters a 1, which it changes to a 2 and returns to state 1, repeating the process. It is easy to see that

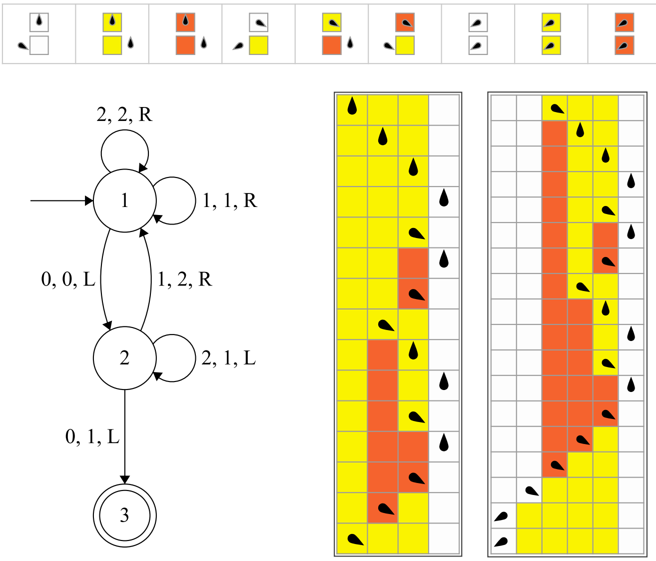


Figure 9. The rule plot, state transition diagram and one example evolution (starting from three ones) for the Turing machine $\mathcal{T}_{\text{exp}}$.

this is precisely the process of incrementing a binary number where 1 corresponds to a zero and 2 to a one. The process is repeated until the head moves to the left of the word, which, by then, consists only of 1 since the previous string was the symbol 2 repeated n times. When the head encounters the first blank symbol on the left, it writes one more 1 to satisfy the T-halter constraint of incrementing the unary number, and then halts. The process is shown in Figure 9 for $n = 3$.

Consider the action of the machine when started at the right of some binary word of 1s and 2s in state 1: The head moves b symbols to the left until it encounters the rightmost 1, and then b symbols back after changing it. Since there are 2^{n-b} binary words of length n where the rightmost 1 is at position b, the Turing machine takes

$$\begin{aligned}\sum_{k=1}^{n} 2k \cdot 2^{n-k} &= 2\sum_{k=1}^{n}(n-k)2^k = 2\left(n\sum_{k=1}^{n}2^k - \sum_{k=1}^{n}k2^k\right) \\ &= 2\left(n\left(2^{n+1}-1\right) - \frac{2-(n+1)2^{n+1}+n2^{n+2}}{(2-1)^2}\right) [15, \text{p. } 36] \\ &= 2\left(n2^{n+1} - n - 2 + n2^{n+1} + 2^{n+1} - n2^{n+2}\right) \\ &= 2\left(-n-2+2^{n+1}\right) = 2^{n+2} - 2n - 4\end{aligned}$$

steps before the head is at the right of 1^n and in state 1. Since the machine takes $n+2$ steps to move to the left again, write the new 1 and halt, as well as taking n steps to move the head to the right in the first place, the total number of states, including starting and halting state, simplifies to

$$T_{\text{exp}}(n) = 2n + 3 + 2^{n+2} - 2n - 4 = 2^{n+2} - 1.$$

In combination with Lemma 3, another strategy for simplifying calculations is to give easily computable bounds for T. In this case, we use the fact that $2^{n+1} < 2^{n+2} - 1 < 2^{n+2}$ for all $n \in \mathbb{N}$ to obtain $2^{x+1} < L_{\mathbb{N}_+}\left(T_{\text{exp}}\right)(x) < 2^{x+2}$ for all $x \in \mathbb{R}_{\geq 0}$. Letting $l(x) := L_{\mathbb{N}_+}\left(T_{\text{exp}}\right)(x)$ for readability, this becomes

$$\begin{aligned} 2^{x+1} < l(x) < 2^{x+2} &\iff \int_0^x 2^{t+1}\,\mathrm{d}t < \int_0^x l(t)\mathrm{d}t < \int_0^x 2^{t+2}\,\mathrm{d}t \\ &\iff \frac{2}{\ln(2)}\left(2^x - 1\right) < \int_0^x l(t)\mathrm{d}t < \frac{4}{\ln(2)}\left(2^x - 1\right). \end{aligned}$$

Since the inverse of

$$x \mapsto \frac{a}{\ln(2)}\left(2^x - 1\right)$$

is

$$y \mapsto \log_2\left(y\frac{\ln(2)}{a} + 1\right),$$

and $f(x) < g(x) \iff f^{-1}(x) > g^{-1}(x)$, the equation is equivalent to

$$\log_2\left(y\frac{\ln(2)}{2} + 1\right) > \left(\int_0^x l(t)\mathrm{d}t\right)^{-1} > \log_2\left(y\frac{\ln(2)}{4} + 1\right).$$

Now, notice that

$$\log_2(y+c) = \log_2(y) + \log_2\left(1 + \frac{c}{y}\right) \overset{x\to\infty}{\longrightarrow} \log_2(y) \quad \left(\text{since } 1 + \frac{c}{y} \overset{x\to\infty}{\longrightarrow} 1\right)$$

and

$$\log_2(yc) = \log_2(y) + \log_2(c) \wedge \log_2(y) + \log_2(c) \overset{x\to\infty}{\longrightarrow} \log_2(y).$$

From this, we know $\left(\int_0^x l(t)\mathrm{d}t\right)^{-1} \in \Theta(\log_2(x))$ because the upper and lower bound asymptotically equal $\log_2(x)$. By Lemma 3 and equation (2), $\left(\int_0^x l(t)\mathrm{d}t\right)^{-1}$ is also in $\Theta\left(L_{I\left(A_{T_{\text{exp}}}\right)}\left(A_{T_{\text{exp}}}\right)\right)$ and we can conclude that $A_{\mathcal{T}_{\exp}} \sim_{\Theta} \log_2(x)$. Simulating the Multiway System and measuring the growth function empirically supports this, as Figure 10 shows. In future examples, most steps of the argumentation presented here can be shortened. However, this method of estimation does not work in all cases because the inverse bounds might not be accurate enough to be asymptotically equal. This can happen because in general, $f \in \Theta(g)$ does not necessarily imply $f^{-1} \in \Theta(g^{-1})$ (consider e.g., $f(x) = \ln(x)$ and $g(x) = 2\ln(x)$).

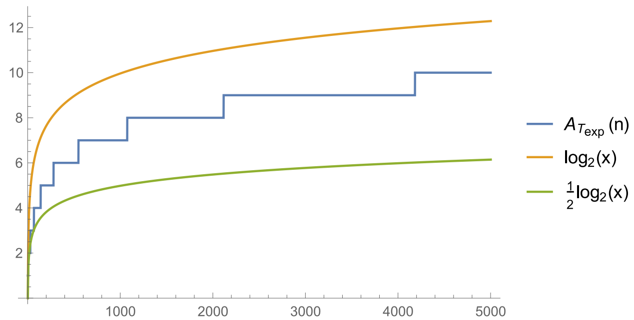


Figure 10. The growth function of the Multiway System emulating $\mathcal{T}_{\exp}$ is bounded by $\log_2(x)$ and $1/2\log_2(x)$, demonstrating that it is in $\Theta(\log_2(x))$.

2.3 Implications of Theorem 1

What we have seen in the previous example is just a simple demonstration of the power of Theorem 1. Besides helping us later to prove Theorem 2, it tells us a lot about the abstract structure of Multiway growth functions, their "growth spectrum." By providing the following two corollaries, Theorem 1 gives us knowledge about what this spectrum of possible growth rates looks like, that is, which kinds of growth rates are possible and which kinds are not. In addition to that, it establishes connections between Multiway growth functions and other classes of functions, namely computable functions and primitive recursive functions.

Corollary 1. For every computable function $f : \mathbb{N} \to \mathbb{N}$, $f \in \omega(1)$, there is a Multiway System with growth rate (a, b) such that $a, b \in O(f^{-1})$ where f^{-1} is an asymptotic inverse of f, that is, some function satisfying $(f^{-1})^{-1} \in \Theta(f)$ (this becomes important when f is not properly invertible).

Proof. Since f is computable, there exists some Turing machine computing $f(n)$ when given n. If we require the machine to read and write input and output in unary coding, computing $f(n)$ must take $T(n) \geq f(n)$ steps simply because writing the result takes that long. Now, let $g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a bijective tight lower bound of T. As $T \in \omega(1)$, T is unbounded, so g always exists. From $g(x) \leq T(x)$, it follows that $g^{-1}(x) \geq T^{-1}(x)$ and by Theorem 1, there is a Multiway System for which the growth function has tight bounds $a, b \in O(g^{-1}) \Rightarrow a, b \in O(f^{-1})$ for some asymptotic inverse of f. □

Corollary 2. For every computable function $f : \mathbb{N} \to \mathbb{N}$, $f \in \omega(1)$, there is a Multiway System with growth rate (a, b) such that $a, b \in o\left(L_{\mathbb{N}_+}(f)\right)$.

Proof. The function $L_{\mathbb{N}_+}(\overline{f})$ (for the upper bounding sequence $\overline{f}$ from Definition 2) is always greater than or equal to f, asymptotically equal to f and computable (because equal to f) on the set of increase-indices $I(f)$. Therefore, $\overline{f}^{-1}$ is a computable function on $\mathbb{N}_+$ and so is $g : x \mapsto \overline{f}^{-1}(x^2)$. Using Corollary 1, this gives us a way to construct a Multiway System with a growth rate in $O(g^{-1}) = O\left(\sqrt{L_{\mathbb{N}_+}(f)}\right)$ that is definitely in $o\left(L_{\mathbb{N}_+}(f)\right)$. □

This quite remarkable fact also shows that for every Multiway System growing faster than a bounded function, a more slowly growing Multiway System exists because the growth function of every Multiway System is obviously computable. We might therefore say that Multiway Systems can grow arbitrarily slowly; that is, the set of regular Multiway Systems excluding constant and finite systems is "open" in some sense. Remember, however, that they cannot grow arbitrarily quickly, as shown in Lemma 1.

3. Computational Capabilities of Growth Functions

After marking out the boundaries of the space of possible growth rates, we will investigate its underlying structure. First of all, we will see that it contains no "holes"; that is, all of the Multiway growth classes defined in Section 1 (except $C_{\sup \exp}$, which we have already shown to be empty and just defined for completeness) are nonempty and, furthermore, contain infinitely many systems. In addition, we will have some insights into which functions are "Multiway growth-computable" and "Multiway growth-approximable." We say, a function $f : \mathbb{N}_+ \to \mathbb{N}_+$ is Multiway growth-computable if there is a

Multiway System M such that $\forall\, n \in \mathbb{N}_+ : g_M(n) = f(n)$ and we call a function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ Multiway growth-approximable if there is a Multiway System M such that $f \sim_\Theta L_{\mathbb{N}}(g)$.

First, we will define two operations, "Multiway addition" and "Multiway multiplication," which will enable us to combine systems into more complex ones of which the growth function is computable immediately from the growth functions of the parts. These two fairly simple operations will be sufficient for demonstrating that Multiway growth functions are interesting from an algebraic point of view as well as regarding questions of their computational capabilities (see Section 3.3). Still, some basic Multiway Systems have to be constructed without using these operations as the building blocks of further systems. Combining the Multiway operations and specifically constructed systems will then yield the following theorem and several other interesting results:

Theorem 2. The classes C_{fin}, C_{bnd}, C_{pol}, C_{int}, C_{exp}, $C_{\text{inv pol}}$, $C_{\text{inv int}}$, $C_{\text{inv exp}}$, $C_{\text{inv sup exp}}$ partition the set of regular Multiway Systems into infinite subsets.

3.1 Arithmetic-Like Operations on Multiway Systems

Let $M_1 = (R_1, s_1, \Sigma_1)$, $M_2 = (R_2, s_2, \Sigma_2)$ and $M_3 = (R_3, s_3, \Sigma_3)$ be Multiway Systems. Additionally, let X be a unique (equal for all Multiway Systems) symbol not included in any Multiway Systems alphabet. Now, we define the "sum system" by

$$M_1 \oplus M_2 = \left(R_1 \cup R_2 \cup \{X \to s_i \mid s_i \in S_2(M_1) \cup S_2(M_2)\}, \text{"X"}, \Sigma_1 \cup \Sigma_2\right)$$

where $S_2(M)$ is the state-set of M in generation 2, that is, all nodes with distance 1 to the initial state in the respective states graphs (see Figure 11). The "product system" of M_1 and M_2 is now defined as

$$M_1 \odot M_2 = \left(R_1 \cup R_2, s_1 s_2, \Sigma_1 \cup \Sigma_2\right)$$

where $s_1 s_2$ denotes the concatenation of s_1 and s_2 (see Figure 12).

To calculate the growth functions of systems obtained by these operations, we will require the parts M_1 and M_2 to be "rule independent," meaning that their rules do not interfere with each other. Formally, rule independence can be defined as the property that the states graph of M_1 is isomorphic to the states graph of $(R_1 \cup R_2, s_1, \Sigma_1 \cup \Sigma_2)$, that is M_1 with all rules of M_2 added, and vice versa. This works because if adding all the rules of M_1 to M_2 does not change its behavior, then these rules will not influence M_2's states even if the states of M_1 get appended to them. Rule independence can always be achieved by requiring the underlying alphabets to be disjoint, as in this case it is impossible for a given string to match rules from R_1 and R_2 at the same time.

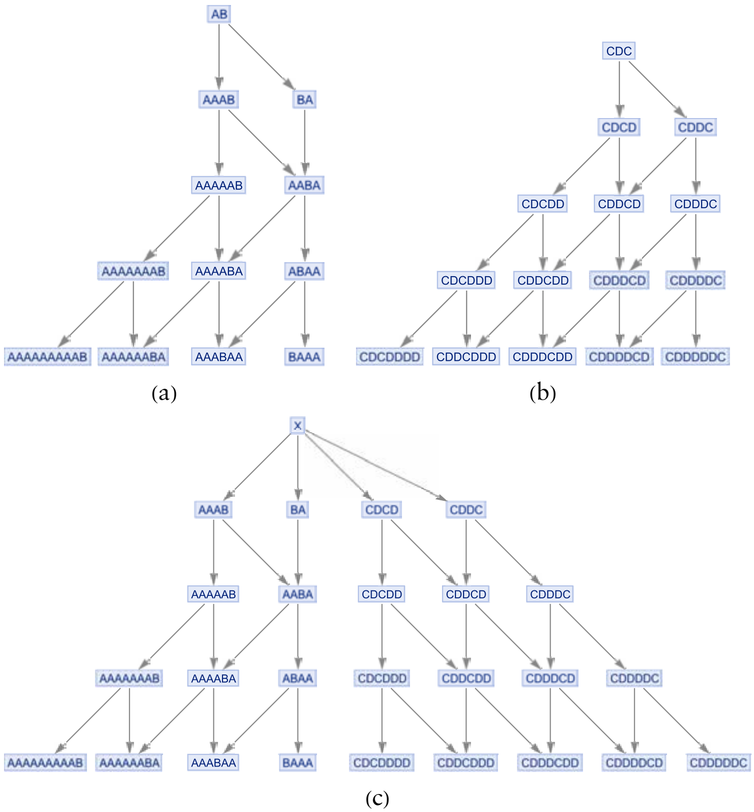
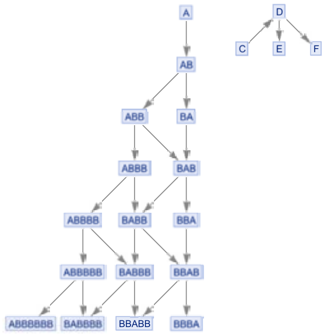
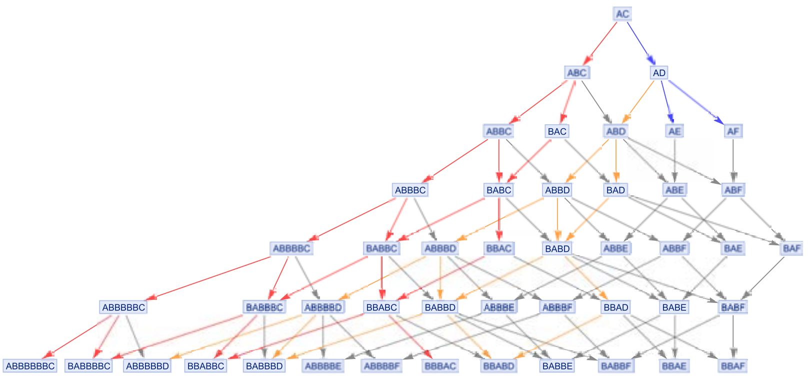


Figure 11. States graphs of rule-independent Multiway Systems and their sum system. The systems used are $(\{"AB" \to "BA", "B" \to "AAB"\}, "AB", \{A, B\})$ and $(\{"CD" \to "CDD", "C" \to "CD"\}, "CDC", \{C, D\})$.

If we recall the definition of growth function $g_M(n)$ as the number of nodes to which the shortest path from the initial state has length n, it is easy to see that the growth function of $M_1 \oplus M_2$ is one in the first iteration, as "X" is the only state. In further iterations, we can imagine a path of length n simply as a path of length 1 entering the states graph of either M_1 or M_2, followed by a path of length $n-1$ originating at some state in the second layer of the chosen subgraph, as if the other graph were not there. This works because in the entire evolution, the initial state is the only one containing an X, so for all other



(a)



(b)

Figure 12. States graphs of $(\{"A" \to "AB", "AB" \to "BA"\}, "A", \{A, B\})$, $(\{"C" \to "D", "D" \to "E", "D" \to "F"\}, "C", \{C, D, E, F\})$ and their product system. Instances of the first system are highlighted in red and orange in the product system's graph.

states, the usual rules will apply and the rules replacing X will have no effect. This concludes that the growth function of the sum system is precisely

$$g_{M_1 \oplus M_2}(n) = g_{M_1}(n) + g_{M_2}(n) \quad \text{with} \quad g_{M_1 \oplus M_2}(1) := 1. \tag{3}$$

For the product system, remember that in the states graph of some system M_1, two nodes u, v are connected by an edge if string u gets transformed into v by a rule from R_1. The analogous holds for M_2. Since the initial node of $M_1 \odot M_2$ consists of a concatenation of s_1 and s_2, any node in the states graph of $M_1 \odot M_2$ corresponds to some combination of a node of M_1 and one of M_2. Hence, the states graph of $M_1 \odot M_2$ is the Cartesian product graph (see [16, p. 30] for a definition) of the states graphs of M_1 and M_2. Note how this is only possible because M_1 and M_2 are rule independent since otherwise more edges could be added due to rule matches overlapping between the M_1- and M_2-parts of the string or rules of one system getting applied to states of the other one.

To obtain the number of nodes reachable in this Cartesian product graph, we might without loss of generality first traverse a path of length k in the "pure M_1-part" (i.e., the second half of the string is still s_2) and then take $n-k$ steps through the "pure M_2-part." For the first subpath, we have $g_{M_1}(k)$ options by definition of the growth function. This gets multiplied by the $g_{M_2}(n-k)$ choices for the second subpath. Since we can choose k freely, the resulting total count of nodes in the product graph is given by

$$\begin{aligned} g_{M_1 \odot M_2}(n) &= \sum_{k=0}^{n} g_{M_1}(k) \cdot g_{M_2}(n-k) \\ &= f(n) + g(n) + \sum_{k=1}^{n-1} g_{M_1}(k) \cdot g_{M_2}(n-k), \end{aligned} \tag{4}$$

where we set $g_{M_1}(0) = g_{M_2}(0) = 1$ for convenience.

For these two formulas, we have assumed the systems to be rule independent. For systems where this is not the case, they still give lower bounds of the combined systems' growth rate since, generally speaking, in any system, the number of edges and nodes of the states graph can only remain constant or be increased when new rules are added. This might sound surprising, as we could imagine "deletion rules," but rules that cause fewer rules to be applied in the future do this only in newly added branches of the states graph (or not at all), not affecting the already existent graph. To make systems rule independent, we required the alphabets to be disjoint; however, this is not necessary as any Multiway System can be emulated by a system over some binary alphabet, so we can always make the alphabets of M_1 and M_2 equal.

Lemma 4. For any Multiway System $M = (R, s, \Sigma)$, there is a Multiway System $M' = (R', s', \{a, b\})$ where a and b are two distinct symbols, such that the states graphs of M and M' are isomorphic.

Proof. To show this, we will perform a "translation" from M to M', that is, replace every symbol in Σ by a word over $\{a, b\}$ using some bijection $f : \Sigma \to T \subset \{a, b\}^*$. By altering not only s but also all rules, any word $w \in \Sigma^*$ matched by some rule in R will correspond to the translated word in $w' \in T$ being matched by a rule in R'. Additionally, one must ensure that no two words in T can overlap since otherwise, the rules could match in more places than before. Since there exist non-overlapping codes of arbitrary length, we can use these as elements in T so there always exists some f with the required properties. Thus, the actions of the rules on the states will be equal and isomorphic states graphs will be created. □

Now, let us consider the algebraic properties of our Multiway operations. We declare two Multiway Systems to be isomorphic (written $M_1 \cong M_2$) if and only if their states graphs are isomorphic. Isomorphic Multiway Systems always have equal growth functions. In the following analysis, we consider only the set of different equivalence classes of $\cong$, that is, the set of all Multiway Systems up to isomorphism, and denote it by $\mathbb{M}$.

Let $M_1, M_2, M_3 \in \mathbb{M}$ be Multiway Systems and, without loss of generality, rule independent. It is easy to see that $\oplus$ is commutative and associative since set unions are. More interestingly, the system $0_M := (\{\}, X, \{\})$ is a neutral element of $\oplus$ since

$$\begin{aligned} M_1 \oplus 0_M &= \left(R_1 \cup \{\} \cup \{X \to s \mid s \in S_2(M_1) \cup \{\}\}, X, \Sigma_1 \cup \{\}\right) \\ &= \left(R_1 \cup \{X \to s \mid s \in S_2(M_1)\}, X, \Sigma_1\right), \end{aligned}$$

which is isomorphic to (R_1, s_1, Σ_1) because the X symbol is used only once, acting precisely as s_1 would have and therefore keeping the states graph structure unchanged.

The commutativity of $\odot$ is granted because we required M_1 and M_2 to be rule independent, so the order in which their states are concatenated does not matter because no overlaps where rules could apply on the intersection of M_1- and M_2-states can be created. Similarly, $\odot$ is associative, simply because string concatenation is. We can also prove both properties with the commutativity and associativity of the Cartesian graph product. There also is a neutral element of $\odot$, namely $1_M := (\{\}, "\ ", \{\})$ or actually any system with no rules since its initial state will just be appended onto every state of the system one is multiplying with and the states graph will not change.

Also notice that $\odot$ distributes from the left over $\oplus$:

$$\begin{aligned} & M_1 \odot (M_2 \oplus M_3) \\ &\quad = (R_1, s_1, \Sigma_1) \odot \left(R_2 \cup R_3 \cup \{X \to S_2(M_2) \cup S_2(M_3)\}, X, \Sigma_2 \cup \Sigma_3\right) \\ &\quad = \left(R_1 \cup R_2 \cup R_3 \cup \{X \to S_2(M_2) \cup S_2(M_3)\}, s_1 X, \Sigma_1 \cup \Sigma_2 \cup \Sigma_3\right) \\ &\quad = \left((R_1 \cup R_2) \cup (R_1 \cup R_3) \cup \right. \\ &\qquad\qquad \left(\{X \to S_2(M_1)\} \cup \{X \to S_2(M_2)\} \cup \{X \to S_2(M_3)\}\right), \\ &\qquad\qquad\qquad \left. X, (\Sigma_1 \cup \Sigma_2) \cup (\Sigma_1 \cup \Sigma_3)\right) \end{aligned}$$

$$= (R_1 \cup R_2, s_1 s_2, \Sigma_1 \cup \Sigma_2) \oplus (R_1 \cup R_3, s_1 s_3, \Sigma_1 \cup \Sigma_3)$$
$$= ((R_1, s_1, \Sigma_1) \odot (R_2, s_2, \Sigma_2)) \oplus ((R_1, s_1, \Sigma_1) \odot (R_3, s_3, \Sigma_3))$$
$$= (M_1 \odot M_2) \oplus (M_1 \odot M_3).$$

Thus, we get right distributivity from commutativity and conclude that $\odot$ distributes over $\oplus$. Therefore, we can conclude that $(\mathbb{M}, \oplus, \odot)$ is a semiring, however with a weakened annihilation property, which does not hold in general. As a consequence, their growth functions also form a semiring with weakened annihilation under the operations $(g_1 + g_2)(n) := g_1(n) + g_2(n)$ (with $(g_1 + g_2)(1)$ defined to be 1) and $(g_1 * g_2)(n) = \sum_{k=0}^{n} g_1(k) \cdot g_2(n-k)$ (with $g_1(0) = g_2(0) := 1$). This demonstrates the potential of Multiway Systems to generate quite diverse and intricate growth functions as the two operations can be used to combine systems in various very interesting ways. The next section elaborates on this.

3.2 Proof of Theorem 2

Let us now construct Multiway Systems in the various growth classes to prove Theorem 2. First, take the product system of a finite "chain" system M_1^N having one new state for N generations until terminating and a constant system $M_2 = (\{"A" \to "AA"\}, "A", \{"A"\})$. After the first $N-1$ steps, $M_1^N \odot M_2$ will have a constant growth function of value N as the first N terms in $\sum_{k=0}^{n} g_{M_1}(k) g_{M_2}(n-k)$ are one and all others zero because M_1 is finite. While this system produces (asymptotically) constant growth functions, it is not suited for multiplying arbitrary growth functions by constants. To achieve the latter, adding some system to itself several times resolves the issue.

The next system to consider, M_3^N, is again given by the rule set $\{"A" \to "AB"\}$ and started on a string of N copies of $"A"$ denoted $"A^N"$. For calculating its growth function, we can represent it differently as the product system of N instances of itself started on a single $"A"$ and thus having the growth rate $g_{M_3}^1(n) = 1$. From this point of view, we can write the growth function of M_3 started on $"A^N"$ recursively as

$$g_{M_3}^N(n) = \sum_{k=0}^{n} g_{M_3}^{N-1}(k) \cdot g_{M_3}^1(n-k) = \sum_{k=0}^{n} g_{M_3}^{N-1}(k).$$

https://doi.org/10.25088/ComplexSystems.31.1.123

This might be recognized as the sequence of $(N-1)$-polytopic numbers, also known as "figurate numbers," of which the n^{th} element is given by ([17, p. 7])

$$\binom{N-1+n-1}{N-1}.$$

Hence, $g_{M_3}^N$ is clearly a polynomial of degree $N-1$ and thus asymptotically equal to x^{N-1}.

Now consider the system $M_4^N = (\{"Q" \to "Qx_i" \mid i = 1, \ldots, N\}, "Q", \{Q, x_1, \ldots, x_N\})$ for distinct symbols x_i. In the n^{th} step of evolution, it has basically generated all words of length n over the alphabet of all x_i. Every node $"Qw"$ (where $w \in \{x_1, \ldots, x_N\}^*$) in the states graph has N outgoing edges to the nodes $"Qx_iw"$ for $1 \le i \le N$. Thus, its growth function is precisely $g_{M_4}^N(n) = N^n$, allowing the possibility of Multiway Systems growing like all exponential functions.

A more sophisticated example is the system

$$M_5^N = \Big(\bigcup_{i=1,\ldots,N} \{"\mathrm{TL}" \to "Tx_iR", "RT" \to "Lx_iT", "Rx_i" \to "x_iR", \\ "x_iL" \to "Lx_i"\}, "TLT", \{"L", "R", "T", x_1, \ldots, x_N\}\Big).$$

Similarly to the previous system, L and R work as generators for words over the alphabet $\{x_1, \ldots, x_N\}$, however, only on the left and right ends of the word, respectively. In every word ever produced by the system, there are exactly two $"T"$ symbols, one at the beginning and one at the end. After generating some new symbol between itself and the $"T"$, the generator $"L"$ or $"R"$ moves one step left or right, respectively, thereby not generating any new symbols, as it is not next to a $"T"$. Since a new symbol is created every time a $"L"$ or $"R"$ reaches the respective $"T"$, the length of the word is increased every n steps where n is the previous word length. Thus, their word length is the sequence "n occurs n times" denoted A_n and asymptotically equal to

$$\left(\sum_{k=0}^{n} k\right)^{-1} = \left(\frac{n(n+1)}{2}\right)^{-1} = \frac{\sqrt{1+8n}-1}{2} \in \Theta\left(\sqrt{n}\right)$$

by Lemma 2. But the system's growth function is given by the number of possible words, that is, N^n and hence asymptotically equal to $N^{\sqrt{n}}$.

The growth function of the previous system is noteworthy because it grows "intermediately," that is, faster than every polynomial function and slower than all exponential functions. Formally, we check this by noticing that

$$\lim_{x\to\infty} \frac{\ln\left(N^{\sqrt{x}}\right)}{x} = 0$$

(if the logarithm grows slower than x, the function is subexponential) and

$$\lim_{x\to\infty} \frac{\ln\left(N^{\sqrt{x}}\right)}{\ln(x)} = \infty$$

(if the logarithm grows faster than $\ln(x)$ times a constant, the function grows faster than x^n for all n). In the study of groups and semigroups, which are related to Multiway Systems (see [7, notes for Section 5.6]), it has long been an open problem, finally solved by Grigorchuk [18], to find groups of intermediate growth. For Multiway Systems, this turns out to be remarkably easy, supporting the claim that Multiway growth functions are computationally diverse and powerful (see Figure 13).

Now that we have shown the existence of infinitely many systems in C_{fin}, C_{bnd}, C_{pol}, $C_{\exp}$ and C_{int}, only the classes of inverse functions remain. As mentioned earlier, Theorem 1 appears to be very useful for this. In Section 2.2, we have already shown the existence of a system with a growth function asymptotically equal to $\log_2(x)$. The Turing machine from that example can be generalized to perform counting in any number system base a, yielding a halting function asymptotically equal to $n \mapsto a^n$, so that the same construction can be used to obtain Multiway Systems growing like $\log_a x$. It is not essential to go through the details here because all $\log_a x$ are asymptotically equal (since they only differ by constants by the base-change law). It is also clear that infinitely many systems in the class $C_{\text{inv}\exp}$ can be created because we could, for example, use Multiway addition to add systems with constant growth functions to logarithmic systems. Notice how this shows generally that every class containing at least one system contains infinitely many systems.

By the same style of argument, we conclude that there are infinitely many systems in $C_{\text{inv pol}}$. Consider a weaker version M_6 of the system of intermediate growth rate M_5^1 for $N = 1$. This system has one state forever but the length of its strings is still the sequence "n occurs n times." Now simply apply the Turing machine construction from Section 2.1 to this system by adding the rules $"LT" \to "Z"$, $"TR" \to "Z"$ and $"Z" \to "ZZ"$, respectively, to generate different branches of the states graph. As described in Section 2.1, this yields the desired growth function of $g_{M_6}(n) = A_n(n) \in \Theta\left(\sqrt{n}\right) \Rightarrow M_6 \in C_{\text{inv pol}}$.

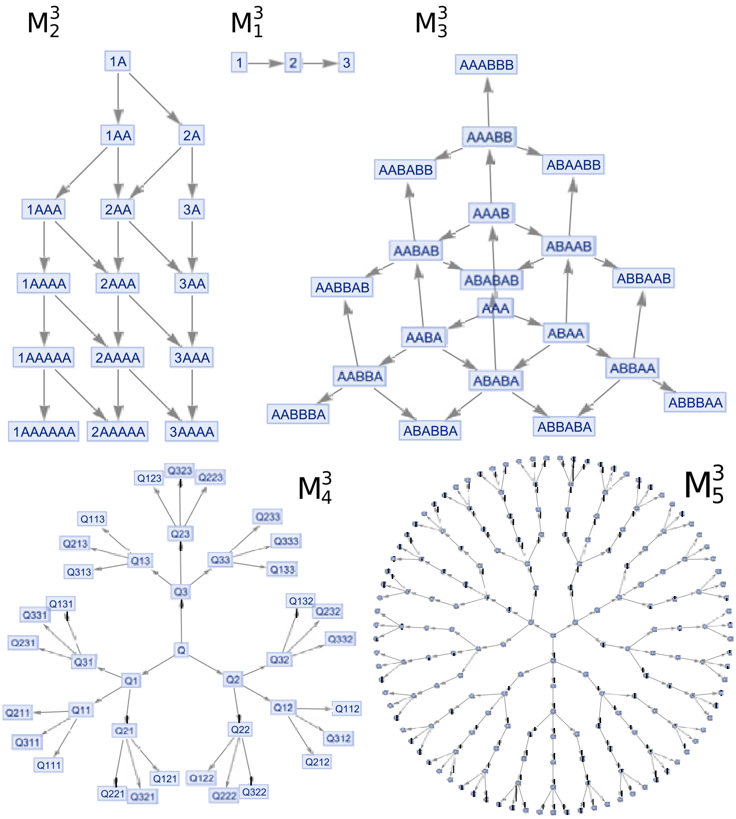


Figure 13. States graphs of exemplary systems M_1^3, M_2^3, M_3^3, M_4^3 and M_5^3 with growth functions asymptotically equal to $n \mapsto 0$, $n \mapsto 3$, $n \mapsto n^2$, $n \mapsto 3^n$ and $n \mapsto 3^{\sqrt{n}}$, respectively.

Using Corollary 2, it is also obvious that there are infinitely many Multiway Systems in $C_{\text{inv sup exp}}$ as we might, for example, just construct a Multiway System growing slower than the inverse Ackermann function. It remains to show that the class of inverses of intermediately growing functions $C_{\text{inv int}}$ is nonempty. To show this, first note that there is a Turing machine $\mathcal{T}$ that computes $\lfloor \sqrt{n} \rfloor$ when given n in unary in polynomial time. If we feed the result of this computation in the $\mathcal{T}_{\exp}$ machine from Section 2, we can construct a machine that halts after $T(n) = p(n) + 2^{\lfloor \sqrt{n} \rfloor}$ steps for some polynomial function $p(n)$. Similar to the argumentation from the proof of Lemma 3, we see that T_σ is asymptotically equal to the integral of

$p(x) + 2^{\sqrt{x}}$ given by $F(x) = q(x) + c\sqrt{x}\, 2^{\sqrt{x}}$ where q is some polynomial (because $\int 2^{\sqrt{x}}\, dx = \frac{2^{\sqrt{x}+1}(\sqrt{x}\ln(2)-1)}{\ln^2(2)}$). Since q is a polynomial, this function grows still intermediately, which we can easily verify by calculating $\lim_{x\to\infty} \frac{\ln(F(x))}{\ln(x)}$ and $\lim_{x\to\infty} \frac{\ln(F(x))}{x}$. By Theorem 1 and Lemma 3, there is a Multiway System M_7 that has a growth function asymptotically equal to $F^{-1}(x)$. Since F grows intermediately, this system is in $C_{\text{inv int}}$. Together with the previous paragraphs, this provides the proof for Theorem 2.

3.3 Multiway Growth Approximability and Undecidability

As we have shown the existence of systems with growth function g_M in $\Theta(1)$, $\Theta(x^n)$, $\Theta(a^x)$, $\Theta\left(a^{\sqrt{x}}\right)$, $\Theta\left(\sqrt{x}\right)$ and $\Theta(\ln(x))$, it is possible to find a Multiway growth function asymptotically equal to any combination of these functions using pointwise addition and discrete convolution. This works because asymptotic equivalence is preserved under addition, summation and multiplication ([19, Section 2.2]), so we can take the appropriate Multiway Systems for the atomic parts of the function's expression and combine them using the Multiway sum and product. The functions $\Theta(1)$, $\Theta(x^n)$, $\Theta(a^x)$, $\Theta\left(a^{\sqrt{x}}\right)$, $\Theta\left(\sqrt{x}\right)$ and $\Theta(\ln(x))$ are all monotonously increasing, and addition as well as discrete convolution preserves this property (because all functions have values in $\mathbb{R}_{\geq 0}$). This proves the following corollary to Theorem 2:

Corollary 3. If for some function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ there is a function $g \in \Theta(f)$ expressible as a finite combination of the functions $x \mapsto c$, $x \mapsto x^n$, $x \mapsto a^x$, $x \mapsto a^{\sqrt{x}}$, $x \mapsto \sqrt{x}$, $x \mapsto \ln(x)$ $(x, a, c \in \mathbb{N}_+)$ using the operations pointwise addition and discrete convolution, then f is Multiway growth-approximable.

The corollary provides further insights into the Multiway growth-approximability and Multiway growth-computability of functions. Let $\mathcal{M}_C$ and $\mathcal{M}_A$ be the sets of Multiway growth-computable and Multiway growth-approximable functions, respectively. Note that $\mathcal{M}_C \subset \mathcal{M}_A$ and $\mathcal{M}_C$ is countably infinite, since the set of Multiway Systems is countably infinite. This follows from the fact that every Multiway System can be reduced to use only the alphabet $\{A, B\}$ (Lemma 4) and then be expressed using the symbols "A", "B", "$\to$", "{", "}", "(", ")" and "," by writing down its signature. The set of words over this finite eight-symbol alphabet is countably infinite. However, $\mathcal{M}_A$ is uncountably infinite because it contains, for example, all constant functions from $\mathbb{R}_{\geq 0}$ to $\mathbb{R}_{\geq 0}$. Using Corollary 3, we notice that a variety of classes of functions are Multiway growth-approximable:

1. For any function $f \in \mathcal{M}_C$, the function $\lambda \cdot f$ where $\lambda \in \mathbb{N}$ is a constant is also in $\mathcal{M}_C$.
2. $\mathcal{M}_A$ contains all polynomials with natural coefficients because they can be built by adding powers x^n multiplied by constants.
3. For all polynomials $p(x)$ in $\mathcal{M}_A$, $\mathcal{M}_A$ contains functions asymptotically equal to $x \mapsto \ln(x) \cdot p(x)$.

Consequence 3 takes a little longer to prove but will be worth explaining in detail because similar methods may be used to generalize it, for example, showing that $\mathcal{M}_A$ contains polylogarithmic functions.

Proof. Consider the product system of a polynomial and a logarithmic system. By equation (4) and the fact that asymptotic equivalence is preserved under addition and multiplication, the system's growth function is asymptotically equal to

$$n^a + \ln(n) + \sum_{k=1}^{n} k^a \ln(n-k) = O(n^a) + \sum_{k=1}^{n-1} h(k).$$

Consider some fixed input n for g. Let B_k be the k^{th} Bernoulli number, $B_m(x)$ the periodic continuation of the m^{th} Bernoulli polynomial and choose $m = n+1$. Using the Euler–Maclaurin formula [20, pp. 501 ff.], we obtain

$$\sum_{k=1}^{n-1} h(k) = \int_1^{n-1} h(x)\mathrm{d}x + \frac{h(n)+h(1)}{2} + S_m + R_m.$$

First of all, $\frac{h(n)+h(1)}{2}$ simply evaluates to $\frac{\ln(n)}{2} \in O(1)$. Next, it is easy to show inductively that the k^{th} derivative of h is of the form $h^{(k)}(x) = x^{a-k}(c\ln(n-x) + p(x))$ for a real constant c and a rational function $p(x) \in O(1)$ as long as $k \geq a$. The derivative $a+1$ is some rational function in $O\left(\frac{1}{x}\right)$. Thus, the remainder sum and integral satisfy

$$S_m = \sum_{k=2}^{n+1} \frac{(-1)^k B_k}{k!}\left(h^{(k)}(n) - h^{(k)}(1)\right)$$

$$\sim_\Theta \sum_{k=2}^{n+1} x^{a-k} \ln(n-x) \in O\left(x^{a-2} \ln(n-x)\right)$$

$$R_m = \frac{(-1)^{n+2}}{(n+1)!} \int_1^{n-1} f^{(n+1)}(x) B_{m+1}(x) \mathrm{d}x$$

$$\sim_\Theta \int_1^{n-1} O\left(\frac{1}{x}\right) \mathrm{d}x \in O(\ln(x)).$$

Hence,

$$\sum_{k=1}^{n-1} h(k) \sim_\Theta \int_1^{n-1} h(x) \mathrm{d}x.$$

Using the fact that

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(x + (n-x) \ln(n-x)\right) = -\ln(n-x)$$

and applying integration by parts, we obtain

$$I_a = \int x^a \ln(n-x) \mathrm{d}x = -x^a\left(x + (n-x)\ \ln(n-x)\right) + \int\left(x + (n-x)\ \ln(n-x)\right) \mathrm{a}\mathrm{x}^{a-1} \mathrm{d}x = T + J$$

$$J = a \int x^a \mathrm{d}x + na \int x^{a-1} \ln(n-x) \mathrm{d}x - a \int x^a \ln(n-x) \mathrm{d}x = \frac{a x^{a+1}}{a+1} + na I_{a-1} - a I_a$$

$$\Rightarrow (a+1) I_a = -x^a\left(x + (n-x)\ \ln(n-x)\right) + \frac{a}{a+1} x^{a+1} + na I_{a-1} + \mathrm{C}$$

$$\Rightarrow \int_1^{n-1} x^a \ln(n-x) \mathrm{d}x = I_a(n-1) - I_a(1)$$

$$= \frac{1}{a+1}\left(-(n-1)^a(n-1+1\cdot 0) + \frac{a(n-1)^{a+1}}{a+1} + na I_{a-1}(n-1) + 1^a\left(1 + (n-1)\ \ln(n-1)\right) - \frac{a}{a+1} 1^{a+1} - na I_{a-1}(1)\right)$$

$$= \frac{1}{a+1}\left(-\frac{(n-1)^{a+1}}{a+1} + \frac{1}{a+1} + (n-1)\ \ln(n-1) + na\left(I_{a-1}(n-1) - I_{a-1}(1)\right)\right)$$

$$= \frac{1}{a+1}\left(\frac{1-(n-1)^{a+1}}{a+1} + (n-1)\ln(n-1) + na\int_1^{n-1} x^{a-1}\ln(n-x)\mathrm{d}x\right).$$

For $a = 1$ we know

$$\begin{aligned}\int_1^{n-1} x^{a-1}\ln(n-x)\mathrm{d}x &= \left[-\left(x + (n-x)\ln(n-x)\right)\right]_1^{n-1} \\ &= -(n-1) + \left(1 + (n-1)\ln(n-1)\right) \\ &= -n + 2 + n\ln(n-1) - \ln(n-1) \\ &\in \Theta\left(n\ln(n-1)\right)\end{aligned}$$

$$\Rightarrow \int_1^{n-1} x^a \ln(n-x)\mathrm{d}x = \frac{1}{2}\left(\frac{1-(n-1)^2}{2} + (n-1)\ln(n-1) + n\Theta\left(n\ln(n-1)\right)\right)$$

$$\Rightarrow \int_1^{n-1} x^1 \ln(n-x)\mathrm{d}x \in \Theta\left(n^2\ln(n-1)\right).$$

Now, by assuming $\int_1^{n-1} x^a \ln(n-x)\mathrm{d}x \in \Theta\left(n^{a+1}\ln(n-1)\right)$ we have

$$\begin{aligned}&\int_1^{n-1} x^a \ln(n-x)\mathrm{d}x \\ &\quad = \frac{1}{a+1}\left(\frac{1-(n-1)^{a+1}}{a+1} + (n-1)\ln(n-1) + na\int_1^{n-1} x^{a-1}\ln(n-x)\mathrm{d}x\right) \\ &\quad = \Theta\left(-(n-1)^{a+1}\right) + \Theta\left((n-1)\ln(n-1)\right) + a\cdot n\cdot\Theta\left(n^a\ln(n-1)\right)\end{aligned}$$

$$\Rightarrow \int_1^{n-1} x^a \ln(n-x)\mathrm{d}x \in \Theta\left(n^{a+1}\ln(n-1)\right),$$

proving $\int_1^{n-1} x^a \ln(n-x)dx \in \Theta\left(n^{a+1}\ln(n-1)\right)$ inductively. Finally, this means the growth function of our Multiway System is asymptotically equal to

$$O(n^a) + \sum_{k=1}^{n-1} h(k) \sim_\Theta n^{a+1} \ln(n-1) \sim_\Theta n^{a+1} \ln(n),$$

proving, in fact, that for every x^a-system there exists a $x^a \ln(x)$-system. □

From these three properties, we might already conclude that a significant number of functions usually investigated in mathematical analysis can be approximated by Multiway growth functions. This demonstrates the computational diversity of Multiway growth functions, which is neither a trivial nor an expected property.

We know that Multiway Systems themselves are capable of universal computation, as they can emulate Turing machines, but it is unknown so far which computations their growth functions are able to perform. Thus, it may be considered remarkable that many common mathematical functions are expressible (i.e., approximable) as Multiway growth functions. Conversely, this could later allow us to make statements about a Multiway System's complexity or structure by considering only its growth function. Maybe Multiway Systems can even be used to make general statements about the mathematical functions themselves since they give a new way of looking at them.

Notice, however, that Multiway growth functions are strictly less powerful than computable functions in general due to the following lemma:

Lemma 5. Every Multiway growth function is primitively recursive.

Proof. Given some Multiway System, we can compute the growth function $g(n)$ in the following way: use two lists S_n and T_n ($S_0 = \{\}$, $T_0 = \{s\}$) to store all states the system has had until generation n and all states of generation n. In every iteration, the length of T_n is the value of $g(n)$. In step $n+1$, the algorithm iterates through all rules and searches through the characters of all strings in T_n to check if any rule applies. If this happens, the string with some part replaced will be added to T_{n+1} if it is not in S_n or T_{n+1} already (only new states get added). After all possible such operations are done, T_{n+1} contains all new states of the system and we set S_{n+1} to be $S_n \cup T_{n+1}$. When repeated, this process simulates the system's evolution and thus yields the correct $g(n)$. All loops required can be implemented using DO-loops. If strings are treated as lists of characters (numbers), the string replacement and substring matching operations can be implemented using only list insertions, deletions and searches through the list. No further data structures and no comparisons are needed. Hence, the entire program is primitively recursive by [21]. □

As primitive recursive functions still contain some superexponential functions, Multiway growth functions are also strictly weaker than those. Still, Multiway growth functions can approximate a lot of elementary functions so they might even be stronger than elementary arithmetic (EA) while probably weaker than EA+ (EA and the axiom that the superexponential function is total). It remains an open question to find out how $\mathcal{M}_A$ or $\mathcal{M}_C$ can be characterized elegantly.

Besides their use for investigating Multiway growth-computability, the tools obtained in this paper allow us to elegantly prove some statements about the undecidability of Multiway growth-related questions. For example, deciding if a given Multiway System is finite is undecidable, as a system could simulate some arbitrary Turing machine and have zero new states when the machine halts, reducing the question to the halting problem. Additionally, even for an infinite Multiway System, deciding whether its growth function is equal to some conjectured function is undecidable in general since the system could be the sum of some usual system and a Turing machine emulator that becomes, for example, exponentially growing after the machine halts but has only one state before that. This observation makes it especially important in the context of the Wolfram Physics Project to not only use empirical (computed) observations about a system's growth function, rate or class but also take into account the system's rule when conjecturing about it.

This "trick" of integrating a system that grows very differently once some Turing machine halts into some larger system was also the strategy used to construct the strongly oscillating system in Figure 4. Specifically, we first construct a system similar to M_5^N from Section 3.2, defined as

$$M_8^{N,M} = \bigcup\nolimits_{i=1,\ldots,N} \{\text{"}RA\text{"} \to \text{"}x_i R\text{"}\}, \text{"}A^M\text{"}, \{\text{"}R\text{"}, A, x_1, \ldots, x_N\}.$$

When started on a string of M As, R moves to the right while replacing the A it just moved over by any of the N x_i. Hence, the states of the system after N steps are precisely the words of length N over $\{x_1, \ldots, x_N\}$ since R moved N steps to the right. When R reaches the right end, the system terminates. Thus, this system has M^N new states after M steps and 0 after that.

The oscillating system from Figure 4 is now the sum of a linearly growing system and a version of the logarithm system from Section 2.2. However, the customized logarithm system does not increase its number of states after $\mathcal{T}_{\text{exp}}$ halts but just triggers an instance of

$M_8^{4,N}$ on the string of ones $\mathcal{T}_{\exp}$ has written. Since after roughly 2^n steps the n^{th} version of $\mathcal{T}_{\exp}$ halts, n ones are written on the tape so about n steps later, the system has 4^n states for one generation and then "collapses" into one state again. The smallest monotonically increasing upper bound for this growth function is one that stays $4^n = (2^n)^2$ for roughly 2^n steps and then increases to 4^{n+1}. Denote this sequence $\overline{a}(n)$. Now $\log_2 \sqrt{\overline{a}(n)}$ is approximately the sequence "n occurs 2^n times," which is in $\Theta(\log_2(n))$. Thus, $\log_2 \sqrt{\overline{a}(n)} \in \Theta(\log_2(n)) \Longleftrightarrow \sqrt{\overline{a}(n)} \in \Theta(n) \Longleftrightarrow \overline{a}(n) \in \Theta(n^2)$. Since the whole system con sists of this and an added linear system, the total growth rate has lower and upper tight bounds of $\Theta(x)$ and $\Theta(x^2)$, respectively. If we gen eralize the methods used in this paper, they might be used to perform a kind of "Multiway System engineering"; that is, they could help to construct systems for specific purposes.

4. Concluding Remarks

While being motivated by applications in fundamental physics, this paper rather focuses on mathematical statements about string rewriting systems in general, thereby strengthening the foundations of the related physical theories. Especially, we make several statements about regularities, structures and restrictions Multiway Systems exhibit despite their status as complex systems. It is important to keep in mind that all our statements apply to all string rewriting systems, although we use the term "Multiway System" to highlight the connection to fundamental physics. We can summarize our main results as follows:

1. We have introduced the notions of Multiway growth functions, rates and classes, which have large potential for mathematically investigating abstract rewriting systems.
2. We have given the exponential upper bound on string-based Multiway System growth and shown that they can grow slower than any computable function, thereby describing boundaries of the space of possible growth rates. This asymmetry implies a more fundamental connection between abstract rewriting systems and computable functions, which will be subject to further research.
3. Furthermore, we have provided a basic but nontrivial classification scheme that partitions the set of all string rewriting systems into a finite number of subsets. From an algebraic point of view, our arithmetic-like operations that equip the set of all Multiway Systems (or string-rewriting systems, for that matter) with a weakened semiring structure seem very promising to build algebraic structures upon string rewriting systems.

4. In the same fashion, we have shown that the growth functions of string-based Multiway Systems can approximate quite an extent of known functions while being contained in the set of primitive recursive functions. It remains unclear and intriguing to find out which status the sets $\mathcal{M}_A$ and $\mathcal{M}_C$ have among other well-known sets of functions, but our Theorem 2 starts a characterization by subdividing them into nontrivial classes.
5. Additionally, we have exemplarily constructed a couple of elementary systems that can be combined purposefully to yield systems giving a desired growth function. This "Multiway engineering" could be generalized and turn out to be useful for getting intuition about string rewriting systems as well as potentially constructing (counter-)examples to empirically grounded conjectures existing in the Wolfram Physics Project.

There are many potential applications of these results in theoretical computer science and the Physics Project. Some ideas are outlined below:

1. While some "types" or "classes" of string rewriting systems have already been studied (see e.g., [22, 23]), we can build on our growth-based classification approach, for example, by allowing combined classes like "polynomial times inverse intermediate"-functions, and try to connect growth classes of string rewriting systems with their properties related to monoids (similar to [13], which uses the growth of a monoid to show that it is not finitely representable through a certain string rewriting system). Also, trying to quantify and find regularity in the oscillations of a system's growth function (consider the system in Figure 3 as an example) and analyzing strongly oscillating systems like the one depicted in Figure 4 are relevant projects to pursue.
2. Related to the regularity of Multiway Systems, we can ask whether determining which growth rate (or growth class) a given Multiway System has from its rules is possible in general (it probably is not). Maybe methods from automated theorem proving could be used for this (as abstract rewriting systems themselves can be used for theorem proving and are amendable to theorem proving techniques [24]). Another question is whether there are Multiway Systems that show no regularity in their growth functions. The latter would be important for complex systems–related research as it has been performed in [7].
3. Speaking of regularity, an important direction of research is the connection between a Multiway System's complexity and its growth function, rate or class. Section 3 and the construction in 3 have already hinted at the hypothesis that a system's rules need to be more complex to achieve a growth rate in one of the slow-growing "inverse"-classes. To build on this, we could try to "restrict" a system's growth by enforcing "boundary conditions" and consider the complexity it can produce in this limited form, similar to the approach employed by LuValle [25] for cellular automata. Additionally, relations between slow growth and complexity of cellular automaton boundaries appear in [12] as well,

especially in the construction of a cellular automaton whose boundary growth "oscillates" similarly to the Multiway System shown in Figure 4. This suggests that exploring the relation between cellular automaton boundary growth and Multiway System growth can be a fruitful enterprise.

4. Regarding the Physics Project, the most obvious next step is generalizing our theorems to hypergraph rewriting systems to make them more meaningful in a physical context. While it might be difficult to generalize them to arbitrary abstract rewriting systems, the hypergraph case should not be difficult to do because of the similarities of set replace systems (which can emulate hypergraph rewriting systems) and string rewriting systems. Developing the algebraic operations we defined for Multiway Systems, we are likely to gain new insights about abstract rewriting systems using the connection to category theory and quantum mechanics established in [5]. Additionally, more general forms of our results could be obtained in the context of the relation between Multiway Systems and the foundations of homotopy type theory [8].
5. More specifically, the changes in structure of branchial space (see glossary of [5]) over time and thus the growth functions of Multiway Systems are related to the states of quantum systems and measurements of these [3]. Our upper bound on Multiway System growth rate may be used to give upper bounds on entanglement speed and maximum possible information entropy in Wolfram Models. Similarly, the fact that slowness of Multiway growth rates is "unbounded" could be used to investigate very slowly developing quantum systems that might be especially stable and hence useful for quantum computation, but that is mere speculation. More straightforward is the application of Multiway growth classes to estimating the complexity of quantum computational algorithms or make predictions about quantum supremacy using the Wolfram model [26].
6. Another potential physical application is early-universe cosmology. Since there seems to be an empirical connection [27], formalizing that would also be an important project, between the growth rates of physical and branchial space; that is, in our context string-length (or hypergraph size) and Multiway growth functions, our boundaries on growth functions and especially the "unboundedness" of its slowness may be related to the physical expansion of the early universe in the view of the Wolfram Model's formalism.

The preceding points are just a few possibilities showing how much potential the investigation of the growth functions, rates and classes of abstract rewriting systems has. This paper marks only the beginning of many further research projects. However, while we lay a very basic foundation, we succeed in doing so, as our results are formally proven and computationally applicable.

Acknowledgments

First and foremost, the author would like to thank Stephen Wolfram for suggesting this project and giving important advice concerning the general directions of research and methodology. Speaking of methodology, I have to mention my mentor Xerxes D. Arsiwalla, for whose guidance and feedback I am very grateful. Many thanks also to Jonathan Gorard for repeatedly proofreading this paper and assisting with the proof that Multiway Systems form a semiring with weakened annihilation property (Section 3.1). Additionally, I highly appreciated the encouragement and support of Peter Barendse, who helped to kick-start this research project at the Wolfram Summer Camp 2020, and Paul Siewert, whose critique and explanations were very useful for formalizing the proofs presented in this paper.

References

[1] F. Baader and T. Nipkow, *Term Rewriting and All That*, New York: Cambridge University Press, 1998.

[2] M. Bezem, J. W. Klop and R. de Vrijer, eds., *Term Rewriting Systems/ Terese*, New York: Cambridge University Press, 2003.

[3] J. Gorard, "Some Quantum Mechanical Properties of the Wolfram Model," *Complex Systems*, **29**(2), 2020 pp. 537–598. doi:10.25088/ComplexSystems.29.2.537.

[4] J. Gorard, "Some Relativistic and Gravitational Properties of the Wolfram Model," *Complex Systems*, **29**(2), 2020 pp. 599–654. doi:10.25088/ComplexSystems.29.2.599.

[5] J. Gorard, M. Namuduri and X. D. Arsiwalla, "ZX-Calculus and Extended Hypergraph Rewriting Systems I: A Multiway Approach to Categorical Quantum Information Theory." arxiv.org/abs/2010.02752.

[6] R. V. Book and F. Otto, *String-Rewriting Systems*, New York: Springer-Verlag, 1993.

[7] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, Inc., 2002. www.wolframscience.com/nks.

[8] X. D. Arsiwalla, "Homotopic Foundations of Wolfram Models" from Wolfram Community Posts, July 2020. community.wolfram.com/groups/-/m/t/2032113.

[9] Y. Marji, "Wolfram Models as Discretization Methods for Numerical PDE Solver" from Wolfram Community Posts, July 2020. community.wolfram.com/groups/-/m/t/2030068.

[10] S. Wolfram, *A Project to Find the Fundamental Theory of Physics*, Champaign, IL: Wolfram Media, 2020.

[11] S. Wolfram, "A Class of Models with the Potential to Represent Fundamental Physics," *Complex Systems*, **29**(2), 2020 pp. 107–536. doi:10.25088/ComplexSystems.29.2.107.

[12] C. D. Brummit and E. Rowland, "Boundary Growth in One-Dimensional Cellular Automata," *Complex Systems*, **21**(2), 2012 pp. 85–116. doi:10.25088/ComplexSystems.21.2.85.

[13] Y. Kobayashi, "A Finitely Presented Monoid Which Has Solvable Word Problem but Has No Regular Complete Presentation," *Theoretical Computer Science*, **146**(1), 1995 pp. 321–329. doi:10.1016/0304-3975(94)00264-J.

[14] F. Otto and Y. Kobayashi, "Properties of Monoids That Are Presented by Finite Convergent String-Rewriting Systems: A Survey," 1996. static.aminer.org/pdf/PDF/000/066/293/properties_of_monoids_that_are_presented_by_finite_convergent_string.pdf.

[15] A. Jeffrey and H.-H. Dai, *Handbook of Mathematical Formulas and Integrals*, 4th ed., Burlington, MA: Academic Press/Elsevier, 2008.

[16] A. Bondy and U. S. R. Murty, *Graph Theory*, New York: Springer, 2008.

[17] L. E. Dickson, *History of the Theory of Numbers, Volume II: Diophantine Analysis*, Mineola, NY: Dover Publications, 2005.

[18] R. I. Grigorchuk, "On the Milnor Problem of Group Growth," *Doklady Akademii Nauk SSSR*, **271**(1), 1983 pp. 30–33. mi.mathnet.ru/dan10037.

[19] A. J. Hildebrand. "Asymptotic Analysis Chapter 2 - Lecture Notes." (Feb 14, 2021) faculty.math.illinois.edu/~hildebr/595ama/ama-ch2.pdf.

[20] U. Storch and H. Wiebe, *Lehrbuch der Mathematik - Band 1: Analysis einer Veränderlichen*, Wiesbaden, Germany: Spektrum Akademischer Verlag, 2010.

[21] A. R. Meyer and M. D. Ritchie, "The Complexity of Loop Programs," in *ACM: Proceedings of the 22nd Annual Conference*, 1967, pp. 465–469. doi:10.1145/800196.806014.

[22] V. Manca, "Logical String Rewriting," *Theoretical Computer Science*, **264**(1), 2001, pp. 25–51. doi:10.1016/S0304-3975(00)00212-7.

[23] V. Diekert, A. J. Duncan and A. G. Myasnikov, "Geodesic Rewriting Systems and Pregroups," *Combinatorial and Geometric Group Theory* (O. Bogopolski, I. Bumagin, O. Kharlampovich and E. Ventura, eds.), Basel, Switzerland: Birkhauser Basel, 2010 pp. 55–91. doi:10.1007/978-3-7643-9911-5_3.

[24] J. Hsiang, H. Kirchner, P. Lescanne and M. Rusinowitch, "The Term Rewriting Approach to Automated Theorem Proving," *The Journal of Logic Programming*, **14**(1), 1992 pp. 71–99. doi:10.1016/0743-1066(92)90047-7.

[25] B. J. LuValle, "The Effects of Boundary Conditions on Cellular Automata," *Complex Systems*, **28**(1), 2019 pp. 97–124. doi:10.25088/ComplexSystems.28.1.97.

[26] J. Gorard, M. Namuduri and X. D. Arsiwalla, "ZX-Calculus and Extended Wolfram Model Systems II: Fast Diagrammatic Reasoning with an Application to Quantum Circuit Simplification." arxiv.org/abs/2103.15820.

[27] M. Namuduri, "Comparing Expansion in Physical and Branchial Space," from Wolfram Community Posts, July 2020, community.wolfram.com/groups/-/m/t/2029454.