An Automated Approach to the Collatz Conjecture

(Explorations with Rewriting Systems, Matrix Interpretations, and SAT)

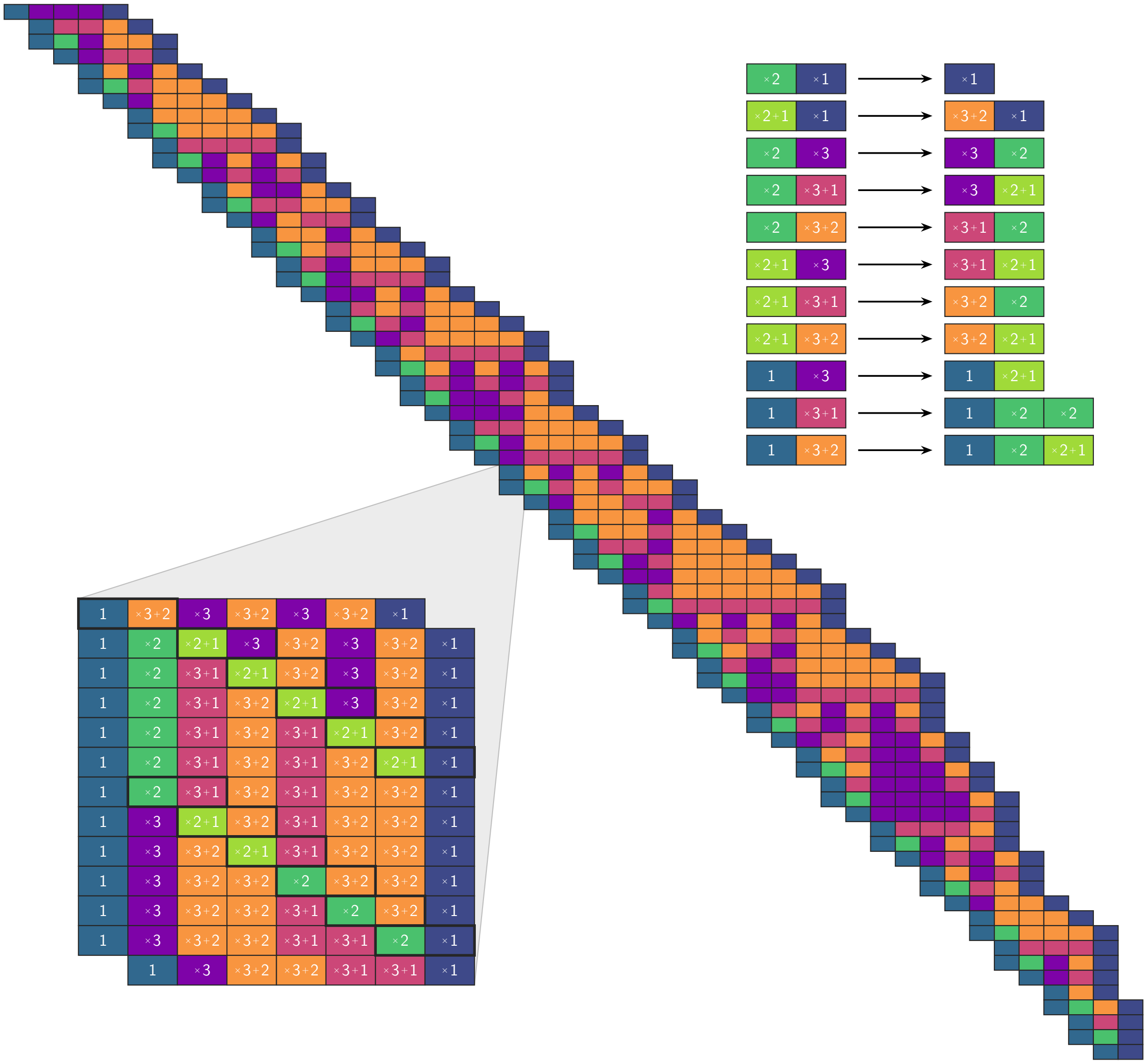
Emre Yolcu* Scott Aaronson† Marijn J. H. Heule*

Abstract

We explore the Collatz conjecture and its variants through the lens of termination of string rewriting. We construct a rewriting system that simulates the iterated application of the Collatz function on strings corresponding to mixed binary–ternary representations of positive integers. We prove that the termination of this rewriting system is equivalent to the Collatz conjecture. We also prove that a previously studied rewriting system that simulates the Collatz function using unary representations does not admit termination proofs via natural matrix interpretations, even when used in conjunction with dependency pairs. To show the feasibility of our approach in proving mathematically interesting statements, we implement a minimal termination prover that uses natural/arctic matrix interpretations and we find automated proofs of nontrivial weakenings of the Collatz conjecture. Although we do not succeed in proving the Collatz conjecture, we believe that the ideas here represent an interesting new approach.

*`eyolcu@cs.cmu.edu`, `marijn@cmu.edu`. Computer Science Department, Carnegie Mellon University.

†`scott@scottaaronson.com`. Department of Computer Science, University of Texas at Austin.



×2	×1	→	×1		
×2+1	×1	→	×3+2	×1	
×2	×3	→	×3	×2	
×2	×3+1	→	×3	×2+1	
×2	×3+2	→	×3+1	×2	
×2+1	×3	→	×3+1	×2+1	
×2+1	×3+1	→	×3+2	×2	
×2+1	×3+2	→	×3+2	×2+1	
1	×3	→	1	×2+1	
1	×3+1	→	1	×2	×2
1	×3+2	→	1	×2	×2+1

1	×3+2	×3	×3+2	×3	×3+2	×1		
1	×2	×2+1	×3	×3+2	×3	×3+2	×1	
1	×2	×3+1	×2+1	×3+2	×3	×3+2	×1	
1	×2	×3+1	×3+2	×2+1	×3	×3+2	×1	
1	×2	×3+1	×3+2	×3+1	×2+1	×3+2	×1	
1	×2	×3+1	×3+2	×3+1	×3+2	×2+1	×1	
1	×2	×3+1	×3+2	×3+1	×3+2	×3+2	×1	
1	×3	×2+1	×3+2	×3+1	×3+2	×3+2	×1	
1	×3	×3+2	×2+1	×3+1	×3+2	×3+2	×1	
1	×3	×3+2	×3+2	×2	×3+2	×3+2	×1	
1	×3	×3+2	×3+2	×3+1	×2	×3+2	×1	
1	×3	×3+2	×3+2	×3+1	×3+1	×2	×1	
	1	×3	×3+2	×3+2	×3+1	×3+1	×1	

Diagrammatic view of the Collatz trajectory of 27. Each row in the cascade is a mixed binary–ternary representation of an iterate, with each box corresponding to a digit. Rewrite rules are shown at the top right corner. The zoomed-in slice at the bottom left corner highlights the intermediate rewrite steps performed to change the least significant digit into binary while preserving the numeric value of the representation.

Contents

1 Introduction

Let $\mathbb{N} = \{0, 1, 2, \dots\}$ denote the natural numbers and $\mathbb{N}^+ = \{1, 2, 3, \dots\}$ denote the positive integers. We define the *Collatz function* $C\colon \mathbb{N}^+ \to \mathbb{N}^+$ as

$$C(n) = \begin{cases} n/2 & \text{if } n \equiv 0 \pmod 2 \\ 3n+1 & \text{if } n \equiv 1 \pmod 2). \end{cases}$$

Given a function f and a number $k \in \mathbb{N}$, the function f^k denotes the *kth iterate of f*, defined as $f^k := \underbrace{f \circ f \circ \cdots \circ f}_{k \text{ times}}$. The well-known *Collatz conjecture* is the following:

Conjecture 1.1. *For all $n \in \mathbb{N}^+$, there exists some $k \in \mathbb{N}$ such that $C^k(n) = 1$.*

This is a longstanding open problem and there is a vast literature dedicated to its study. For its history and comprehensive surveys on the problem, we refer the reader to Lagarias [Lag10, Lag11, Lag12]. See also the work by Tao [Tao20] for dramatic recent progress on the problem.

Definition 1.2 (Convergent function)**.** Consider a function $f\colon X \to X$ and a designated element $z \in X$. Given $x \in X$, the sequence of iterates $f^{\mathbb{N}}(x) := (x, f(x), f^2(x), \dots)$ is called the *f-trajectory of x*. If the trajectory $f^{\mathbb{N}}(x)$ reaches a cycle containing z, the trajectory is *convergent to z for x*. If for all $x \in X$ the trajectory $f^{\mathbb{N}}(x)$ is convergent to z, the function f is *convergent to z*.[1]

As of 2020, it has been checked by computation that for all $n \leq 2^{68}$ the Collatz trajectory of n is convergent [Bar21].[2] This fact, along with the heuristic argument that on average the iterates in a trajectory tend to decrease [Lag85], is often used as evidence to support the belief that the conjecture is true. On a different note, Conway proved that a generalization of the problem is undecidable [Con72], which Kurtz and Simon later extended to a variant even closer to the Collatz conjecture [KS07]. While the prevailing opinion is that the Collatz conjecture holds, and the heuristic arguments suggest that an average trajectory is convergent, the undecidability results help explain why this seems to be so hard to show.

In this paper, we describe an approach based on termination of string rewriting to automatically search for a proof of the Collatz conjecture. Although trying to prove the Collatz conjecture via automated deduction is clearly a moonshot goal, there are two recent technological advances that provide reasons for optimism that at least some interesting variants of the problem might be solvable. First, the invention of the method of matrix interpretations [HW06, EWZ08] and its variants such as arctic interpretations [KW09, ST14] turns the quest of finding a ranking function to witness termination into a problem that is suitable for systematic search. Second, the progress in satisfiability (SAT) solving makes it possible to solve many seemingly difficult combinatorial problems efficiently in practice. Their combination, i.e., using SAT solvers to find interpretations, has so far been an effective strategy in solving challenging termination problems.

We make the following contributions:

[1] For the functions we define throughout this paper, the designated element z is clear from the context, so we refer to convergence without explicitly mentioning z. It is also apparent in all cases that z is in a cycle, and we simply call a trajectory convergent if it contains z.

[2] See https://pcbarina.fit.vutbr.cz for David Barina's ongoing verification effort.

- We show how a generalized Collatz function can be expressed as a rewriting system that is terminating if and only if the function is convergent.
- We prove that no termination proof via natural matrix interpretations exists for a certain system by Zantema [Zan05] that simulates the Collatz function using unary representations of numbers. This result holds even when natural matrix interpretations are used in conjunction with the dependency pair transformation [AG00].
- We show that translations into rewriting systems that use non-unary representations of numbers are more amenable to automated methods, compared with the previously and commonly studied unary representations.
- We automatically prove various weakenings of the Collatz conjecture. We observe that, for some of these weakenings, the only matrix interpretations that our termination tool was able to find involved relatively large matrices (of dimension 5). Existing termination tools often limit their default strategies to search for small interpretations as they are tailored for the setting where the task is to quickly solve a large quantity of relatively easy problems. We make the point that, given more resources, the method of matrix interpretations has the potential to scale.
- We observe that the phase-saving heuristic used by default in modern SAT solvers can degrade the performance of CDCL (conflict-driven clause learning) solvers [MS99] on formulas encoding the existence of matrix interpretations, whereas using negative branching can improve solver performance.

In Appendix C, we present adaptations of our rewriting system that allow reformulating several more open problems in mathematics as termination problems of small size.

2 Preliminaries

2.1 String Rewriting Systems

We briefly review the notions related to string rewriting that are used in the rest of the paper. We adopt the notation of Zantema [Zan05] and Endrullis et al. [EWZ08]. For a comprehensive introduction to string or term rewriting, see Book and Otto [BO93] or Baader and Nipkow [BN98], respectively.

Definition 2.1 (String rewriting system)**.** Let Σ be an alphabet, i.e., a set of symbols. A *string rewriting system* (SRS) over Σ is a relation $R \subseteq \Sigma^* \times \Sigma^*$. Elements $(\ell, r) \in R$ are called *rewrite rules* and are usually written as $\ell \to r$. The system R induces a *rewrite relation* $\to_R \coloneqq \{(s\ell t, srt) \mid s, t \in \Sigma^*,\ \ell \to r \in R\}$ on the set Σ^* of strings.

Definition 2.2 (Termination)**.** A relation $\to$ on A is *terminating* if there is no infinite sequence $s_0, s_1, \ldots \in A$ such that $s_i \to s_{i+1}$ for all $i \geq 0$. A terminating relation is also commonly called Noetherian, well-founded, or strongly normalizing. We write $\mathrm{SN}(\to)$ to denote that $\to$ is terminating.

We often conflate an SRS R with the rewrite relation it induces when discussing termination. In particular, we write "R is terminating" instead of "$\to_R$ is terminating". We also often do not specify the alphabet and take it to be the set of all the symbols that appear in a given rewriting system.

Example 2.3.

1. $R_1 = \{aa \to a\}$ is terminating, which is proved easily by observing that each rewrite decreases the length of a string by 1.
2. $R_2 = \{a \to aa\}$ is not terminating, since even a single occurrence of a allows indefinitely producing more of the symbol a.
3. $R_3 = \{ab \to ba\}$ is terminating, since all rewrite sequences eventually convert a given string into the form b^*a^* and stop. An example sequence is shown below, with the underlines indicating the substrings that the rules are applied to.

$$\underline{ab}bab \to_{R_3} b\underline{ab}ab \to_{R_3} bba\underline{ab} \to_{R_3} bb\underline{ab}a \to_{R_3} bbbaa$$

A useful generalization of termination is relative termination:

Definition 2.4 (Relative termination)**.** Let $\to_1, \to_2$ be relations on A. Then $\to_1$ is said to be *terminating relative to* $\to_2$ if there is no infinite sequence $s_0, s_1, \ldots \in A$ such that

- $s_i \to_1 s_{i+1}$ for infinitely many values of i,
- $s_i \to_2 s_{i+1}$ for all other values of i.

We write $\mathrm{SN}(\to_1 / \to_2)$ to denote that $\to_1$ is terminating relative to $\to_2$.

Equivalently, R is said to be terminating relative to S if every sequence of rewrites for the system $R \cup S$ applies the rules from R at most finitely many times. Note that we have $\mathrm{SN}(R / \varnothing) = \mathrm{SN}(R)$ by the definition, so relative termination is indeed a generalization.

Example 2.5. Let $R = \{aa \to aba\}$ and $S = \{b \to bb\}$. The system $R \cup S$ is not terminating since S gives way to infinite rewrite sequences; however, R is terminating relative to S because R by itself is terminating and applying the rule $b \to bb$ does not facilitate further applications of the rule $aa \to aba$.

Relative termination allows proofs of termination to be broken into steps as codified by the following theorem.

Theorem 2.6 (Rule removal [Zan05, Theorem 1])**.** *Let R be an SRS. If there exists a subset $T \subseteq R$ such that $\mathrm{SN}(T / R)$ and $\mathrm{SN}(R \setminus T)$, then $\mathrm{SN}(R)$.*

This theorem allows us to "remove rules" in the following way. When proving $\mathrm{SN}(R)$, if we succeed at finding a subset T satisfying $\mathrm{SN}(T / R)$, the proof obligation becomes weakened to $\mathrm{SN}(R \setminus T)$, where the rules of T are no longer present. This removal of rules can be repeated until no rules remain, thus producing a stepwise proof of termination.

Another useful technique is reversal:

Definition 2.7 (Reversal)**.** Let $s = s_1 \ldots s_n \in \Sigma^*$ be a string and R be an SRS. Letting $s^{\mathrm{rev}} := s_n \ldots s_1$, the *reversal of* R is defined as $R^{\mathrm{rev}} := \{\ell^{\mathrm{rev}} \to r^{\mathrm{rev}} \mid \ell \to r \in R\}$.

Lemma 2.8 (Rule reversal [Zan05, Lemma 2])**.** *Let R, S be SRSs. Then,* $\mathrm{SN}(R\,/\,S)$ *if and only if* $\mathrm{SN}(R^{\mathrm{rev}}\,/\,S^{\mathrm{rev}})$*.*

Reversal is of interest because methods for proving termination are not necessarily invariant under reversal, that is, a given technique may fail to show the termination of a system R while succeeding for its reversal R^{rev}.

Yet another important notion is top termination:

Definition 2.9 (Top termination)**.** Let R be an SRS over Σ. The *top rewrite relation* induced by R is defined as $\to_{R_{\mathrm{top}}} \coloneqq \{(\ell s, rs) \mid s \in \Sigma^*,\ \ell \to r \in R\}$. If $\to_{R_{\mathrm{top}}}$ is terminating, R is said to be *top terminating*.

In plain language, top termination allows rewrites to be performed only at the leftmost end of a string, so it requires a condition less strict than that of termination. As we will see in the next section (Theorem 2.15), top termination problems can admit proofs of a more relaxed form compared with termination.

Relative top termination, i.e., proving $\mathrm{SN}(R_{\mathrm{top}}\,/\,S)$ for SRSs R and S, is a crucial component in the dependency pair approach [AG00], which reduces a termination problem to a relative top termination problem that often ends up being easier to solve via automated methods compared with the original problem. In order to avoid requiring familiarity with the dependency pair approach, we omit its discussion and instead prove a self-contained result (Lemma 3.18) that encapsulates dependency pairs in a more elementary manner for the specific rewriting systems that we consider in this paper.

2.2 Interpretation Method

Let R be an SRS over the alphabet Σ. The main idea when proving termination is to show that there exists a well-founded (strict) order $>$ on Σ^* such that, for all strings $s, t \in \Sigma^*$, if $s \to_R t$ then $s > t$. When this is the case, R is terminating since any rewrite causes a "strict decrease" with respect to $>$ and by its well-foundedness there cannot exist an infinitely decreasing sequence. The condition over infinitely many strings can be replaced by another one that involves only the rules of R by employing a *reduction order* [BN98, Section 5.2; BO93, Section 2.2].

Definition 2.10 (Reduction order)**.** Let $>$ be a well-founded order on Σ^*. If for all $s, t, p, q \in \Sigma^*$, $s > t$ implies $psq > ptq$, then $>$ is called a *reduction order*.

Theorem 2.11 ([BN98, Theorem 5.2.3; BO93, Theorem 2.2.4])**.** *Let R be an SRS over the alphabet Σ. The system R is terminating if and only if there exists a reduction order $>$ on Σ^* such that $\ell > r$ holds for all $\ell \to r \in R$.*

The interpretation method is a particular way of defining a reduction order. Instead of considering the strings directly, we interpret each symbol $\sigma \in \Sigma$ as a function $[\sigma]\colon A \to A$, where A is some carrier set, for example of natural numbers or vectors, and we extend this interpretation from symbols to strings $s = s_1 \dots s_n \in \Sigma^*$ as $[s] \coloneqq [s_1] \circ \dots \circ [s_n]$. When the domain A is already equipped with a well-founded order $>$, we can define a well-founded order $>_{\mathfrak{A}}$ on Σ^* by

$$s >_{\mathfrak{A}} t \quad \text{if and only if} \quad [s](x) > [t](x) \text{ for all } x \in A, \tag{1}$$

where $\mathfrak{A} = (A, [\cdot]_\Sigma, >)$ is a structure with $[\cdot]_\Sigma \coloneqq \{[\sigma] \mid \sigma \in \Sigma\}$ denoting the collection of interpretations. For appropriately chosen interpretations, $>_{\mathfrak{A}}$ ends up being a reduction order:

Theorem 2.12 ([BN98, Theorem 5.3.3])**.** *Let Σ be an alphabet and let $>$ be a well-founded order on A. If for all symbols $\sigma \in \Sigma$ the interpretation $[\sigma]\colon A \to A$ is monotone with respect to $>$, then $>_{\mathfrak{A}}$ as defined in (1) is a reduction order on Σ^*.*

Thus, after fixing $(A, >)$, proving termination reduces to finding suitable (monotone) interpretations of the symbols as functions [for details see BN98, Section 5.3].[3] As a concrete example, here is a simple proof of termination via the interpretation method.

Example 2.13. Recall $R_3 = \{ab \to ba\}$ from Example 2.3 with $\Sigma = \{a, b\}$. Fix $(\mathbb{N}^+, >)$ with the usual order on positive integers, which is well-founded by the well-ordering principle.

Let $[a](x) = x^2$ and $[b](x) = x + 1$. Denote $\mathfrak{N} = (\mathbb{N}^+, \{[a], [b]\}, >)$. Both functions are monotone with respect to $>$, so by Theorem 2.12 the relation $>_{\mathfrak{N}} = \{(s, t) \in \Sigma^* \times \Sigma^* \mid [s](x) > [t](x) \text{ for all } x \in \mathbb{N}^+\}$ is a reduction order. For the rule $ab \to ba$, since

$$([a] \circ [b])(x) = (x+1)^2 > x^2 + 1 = ([b] \circ [a])(x)$$

holds for all $x \in \mathbb{N}^+$, we have $ab >_{\mathfrak{N}} ba$. Thus, by Theorem 2.11 the SRS R_3 is terminating.

In order to use the interpretation method for relative termination (resp. top termination), the above ideas can be generalized in the form of extended (resp. weakly) monotone algebras [EWZ08].

Definition 2.14 (Extended/weakly monotone algebra)**.** Let

- Σ be an alphabet,
- A a set,
- $[\sigma]\colon A \to A$ an interpretation for every $\sigma \in \Sigma$,
- $>$ and $\gtrsim$ order relations over A such that $>$ is well-founded and $\gtrsim$ satisfies $> \cdot \gtrsim \subseteq >$.

Letting $[\cdot]_\Sigma = \{[\sigma] \mid \sigma \in \Sigma\}$, the structure $(A, [\cdot]_\Sigma, >, \gtrsim)$ is a *weakly monotone Σ-algebra* if for every $\sigma \in \Sigma$ the interpretation $[\sigma]$ is monotone with respect to $\gtrsim$. It is an *extended monotone Σ-algebra* if, additionally, for every $\sigma \in \Sigma$ the interpretation $[\sigma]$ is monotone with respect to $>$.

Relative termination (resp. top termination) is characterized as the existence of extended (resp. weakly) monotone algebras.

Theorem 2.15 ([EWZ08, Theorem 2])**.** *Let R and S be SRSs over the alphabet Σ. We have* $\mathrm{SN}(R \,/\, S)$ *(resp.* $\mathrm{SN}(R_{\mathrm{top}} \,/\, S)$*) if and only if there exists an extended (resp. weakly) monotone Σ-algebra $(A, [\cdot]_\Sigma, >, \gtrsim)$ such that*

- *for each rule $\ell \to r \in R$ we have $[\ell](x) > [r](x)$ for all $x \in A$,*
- *for each rule $\ell \to r \in S$ we have $[\ell](x) \gtrsim [r](x)$ for all $x \in A$.*

[3] Termination is in fact equivalent to the existence of such interpretations [Zan94, Proposition 1].

An effective way to automatically prove relative termination (or top termination) is to try to satisfy the conditions of the above theorem by fixing $(A, >, \gtrsim)$ and algorithmically searching for appropriate interpretations of the symbols. Matrix interpretations is a specific instance of this generic method. In the following section we describe (at a high level) the method of matrix interpretations. For more details and background, we refer the reader to existing work [HW06, EWZ08, KW09, ST14].

It may seem an odd choice to focus on matrix interpretations and implement our own prover given that there is a wealth of other techniques for proving termination that are already implemented in the existing tools (e.g., AProVE [GAB+17], Matchbox [Wal04], $\mathsf{T_TT_2}$ [KSZM09]). We did experiment with other methods; however, at least for the rewriting systems of the specific form in this paper it appears that the method of matrix interpretations is the most effective. For instance, one of the most difficult instances in this paper (in the sense of the smallest proof of termination that we could find) appears to require arctic matrix interpretations in all the proofs that we (and the authors of AProVE and Matchbox) could find. (There remain some versions of matrix interpretations that we have not yet experimented with [CGP10, NM11].)

Additionally, we chose to implement our own minimal prover to have finer control over the SAT solving aspect of the search for an interpretation. We give a summary of our findings regarding this topic in Section 4.6.

2.3 Natural/Arctic Matrix Interpretations

2.3.1 Natural Numbers

Let R be an SRS over the alphabet Σ. Fix a dimension $d \in \mathbb{N}^+$. In the method of natural matrix interpretations [EWZ08] we set $A = \mathbb{N}^d$ and define, for vectors $\mathbf{x}, \mathbf{y} \in \mathbb{N}^d$,

$$\begin{aligned} \mathbf{x} > \mathbf{y} &\iff x_1 > y_1 \ \wedge \ x_i \geq y_i \text{ for all } i \in \{2, \ldots, d\}, \\ \mathbf{x} \gtrsim \mathbf{y} &\iff x_i \geq y_i \text{ for all } i \in \{1, \ldots, d\}. \end{aligned}$$

For interpreting each symbol $\sigma \in \Sigma$, we consider an affine function $[\sigma] \colon \mathbb{N}^d \to \mathbb{N}^d$:

$$[\sigma](\mathbf{x}) = \mathbf{M}_\sigma \mathbf{x} + \mathbf{v}_\sigma$$

In this way, the structure $(\mathbb{N}^d, [\cdot]_\Sigma, >, \gtrsim)$ satisfies the requirements of Definition 2.14 for a weakly monotone algebra. Additionally setting $(M_\sigma)_{1,1} = 1$ satisfies the requirements for an extended monotone algebra.

Furthermore, since the composition of affine functions is affine, for all $s \in \Sigma^*$ there exist $\mathbf{M}_s \in \mathbb{N}^{d \times d}$, $\mathbf{v}_s \in \mathbb{N}^d$ such that $[s](\mathbf{x}) = \mathbf{M}_s \mathbf{x} + \mathbf{v}_s$. As a result, given a collection $[\cdot]_\Sigma$ of natural matrix interpretations, the conditions of Theorem 2.15 can be checked by computing for each rule $\ell \to r \in R$ the corresponding matrices $\mathbf{M}_\ell$, $\mathbf{M}_r$ and vectors $\mathbf{v}_\ell$, $\mathbf{v}_r$ and comparing them elementwise [EWZ08, Lemma 4]. Extend $\gtrsim$ to matrices $\mathbf{M}, \mathbf{N} \in \mathbb{N}^{d \times d}$ as

$$\mathbf{M} \gtrsim \mathbf{N} \iff M_{i,j} \geq N_{i,j} \text{ for all } i, j \in \{1, \ldots, d\}.$$

Then, we have

$$\begin{aligned} \mathbf{M}_\ell \gtrsim \mathbf{M}_r \ \wedge \ \mathbf{v}_\ell > \mathbf{v}_r &\iff [\ell](\mathbf{x}) > [r](\mathbf{x}) \text{ for all } \mathbf{x} \in \mathbb{N}^d, \\ \mathbf{M}_\ell \gtrsim \mathbf{M}_r \ \wedge \ \mathbf{v}_\ell \gtrsim \mathbf{v}_r &\iff [\ell](\mathbf{x}) \gtrsim [r](\mathbf{x}) \text{ for all } \mathbf{x} \in \mathbb{N}^d. \end{aligned} \tag{2}$$

This shows that it is decidable to check whether a given collection of interpretations of the form considered here constitutes a proof of termination.

Example 2.16. Recall $R = \{aa \to aba\}$ and $S = \{b \to bb\}$ from Example 2.5 with $\Sigma = \{a, b\}$. The following functions constitute a matrix interpretations proof that shows $\mathrm{SN}(R \,/\, S)$.

$$[a](\mathbf{x}) = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad [b](\mathbf{x}) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

It is easy to check that the above interpretations give an extended monotone algebra and that they satisfy the below relations for all $\mathbf{x} \in \mathbb{N}^2$, which implies $\mathrm{SN}(R \,/\, S)$ via Theorem 2.15.

$$[aa](\mathbf{x}) = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} > \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = [aba](\mathbf{x})$$

$$[b](\mathbf{x}) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \gtrsim \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = [bb](\mathbf{x})$$

In order to automate the search for matrix interpretations given a rewriting system R, an approach that turns out to be effective is to encode all of the aforementioned constraints as a propositional formula in conjunctive normal form and use a SAT solver to look for a satisfying assignment. This additionally involves fixing a finite domain for the coefficients that can occur in the interpretations and encoding arithmetic over the chosen finite domain using propositional variables. (We discuss our choice of encoding in Section 4.6.)

2.3.2 Arctic Integers

Natural matrix interpretations can also be adapted to the max–plus algebra of arctic[4] natural numbers $\mathbb{A}_{\mathbb{N}} = \mathbb{N} \cup \{-\infty\}$ or arctic integers $\mathbb{A}_{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty\}$ as coefficients. This adaptation enables termination proofs in some cases where either there is no other automated method that leads to a proof or only relatively complicated proofs are known [KW09].

Let R be an SRS over the alphabet Σ, and let $\mathbb{A}$ be either one of $\mathbb{A}_{\mathbb{N}}$ or $\mathbb{A}_{\mathbb{Z}}$. In the method of arctic matrix interpretations [KW09, ST14] we use the arctic semiring $(\mathbb{A}, \oplus, \otimes)$, with the following operations:

$$\begin{aligned} x \oplus y &= \max\{x, y\} \\ x \otimes y &= x + y \end{aligned}$$

Let $>_{\mathbb{Z}}$ and $\geq_{\mathbb{Z}}$ denote the usual order relations over integers. Adopting the conventions that $x >_{\mathbb{Z}} -\infty$ for all $x \in \mathbb{Z}$ and that $-\infty \geq_{\mathbb{Z}} -\infty$, we define, for $x, y \in \mathbb{A}$,

$$\begin{aligned} x \gg y &\iff x >_{\mathbb{Z}} y \;\vee\; x = y = -\infty, \\ x \geq y &\iff x \geq_{\mathbb{Z}} y. \end{aligned}$$

We fix a dimension $d \in \mathbb{N}^+$ and we extend both $\gg$ and $\geq$ to arctic vectors and arctic matrices elementwise. As these definitions allow $-\infty \gg -\infty$, the order $\gg$ is not well-founded. To ensure well-foundedness of $\gg$ we restrict the domain to $\mathbb{N} \times \mathbb{A}^{d-1}$. Then, for interpreting each symbol $\sigma \in \Sigma$, we consider an affine function $[\sigma] \colon \mathbb{A}^d \to \mathbb{A}^d$, written as

$$[\sigma](\mathbf{x}) = \mathbf{M}_\sigma \otimes \mathbf{x} \oplus \mathbf{v}_\sigma,$$

[4] The name "arctic" was chosen to contrast with the min–plus algebra of the "tropical" semiring.

where we require $(M_\sigma)_{1,1} \geq 0$ or $(v_\sigma)_1 \geq 0$ to ensure that $[\sigma](\mathbf{z}) \in \mathbb{N} \times \mathbb{A}^{d-1}$ for $\mathbf{z} \in \mathbb{N} \times \mathbb{A}^{d-1}$. In this way, the structure $(\mathbb{N} \times \mathbb{A}^{d-1}, [\cdot]_\Sigma, \gg, \geq)$ satisfies the requirements of Definition 2.14 for a weakly monotone algebra. Additionally choosing $\mathbb{A} = \mathbb{A}_\mathbb{N}$ and setting each $\mathbf{v}_\sigma$ to be the $-\infty$ vector (consequently enforcing $(M_\sigma)_{1,1} \geq 0$) satisfies the requirements for an extended monotone algebra. This means that arctic integers (as opposed to arctic natural numbers) and vectors that contain finite elements (i.e., not all $-\infty$) are used only when proving top termination.

As the composition of affine functions over arctic vectors is affine, given a collection $[\cdot]_\Sigma$ of arctic matrix interpretations we can compute for each rule $\ell \to r \in R$ the corresponding arctic matrices $\mathbf{M}_\ell$, $\mathbf{M}_r$ and arctic vectors $\mathbf{v}_\ell$, $\mathbf{v}_r$. Elementwise comparisons of these matrices and vectors give sufficient conditions for the inequalities in Theorem 2.15. In particular, due to Koprowski and Waldmann [KW09, Lemma 6.5], we have

$$\begin{aligned} \mathbf{M}_\ell \gg \mathbf{M}_r \ \wedge\ \mathbf{v}_\ell \gg \mathbf{v}_r &\implies [\ell](\mathbf{x}) \gg [r](\mathbf{x}) \text{ for all } \mathbf{x} \in \mathbb{A}^d,\\ \mathbf{M}_\ell \geq \mathbf{M}_r \ \wedge\ \mathbf{v}_\ell \geq \mathbf{v}_r &\implies [\ell](\mathbf{x}) \geq [r](\mathbf{x}) \text{ for all } \mathbf{x} \in \mathbb{A}^d. \end{aligned}$$

Thus, the search for arctic matrix interpretations can be automated in a manner similar to natural matrix interpretations by encoding the aforementioned constraints as a SAT instance.

Example 2.17. Let $\mathcal{P}$ denote the following SRS[5].

$$\begin{aligned} \triangleleft\mathtt{b} &\to \triangleleft\mathtt{ax}\\ \mathtt{aax} &\to \mathtt{bxa}\\ \mathtt{ab} &\to \mathtt{ba}\\ \mathtt{xb} &\to \mathtt{ax}\\ \mathtt{a}\triangleright &\to \mathtt{ax}\triangleright \end{aligned}$$

As an example of the different capabilities of arctic versus natural matrix interpretations, consider trying to prove the relative termination statement $\mathrm{SN}(\{\triangleleft\mathtt{b} \to \triangleleft\mathtt{ax}\} \mathbin{/} \mathcal{P})$. To our knowledge, there is no direct proof of this statement via natural matrix interpretations although there is one via the 4-dimensional arctic matrix interpretations below. In the interpretations, $-$ indicates $-\infty$, and the multiplication operator $\otimes$ as well as the vectors of all $-\infty$s are omitted.

$$[\mathtt{a}](\mathbf{x}) = \begin{bmatrix} 0 & - & - & - \\ 1 & 1 & - & 0 \\ - & 0 & - & - \\ - & - & 0 & - \end{bmatrix}\mathbf{x} \qquad [\mathtt{b}](\mathbf{x}) = \begin{bmatrix} 0 & - & - & - \\ 1 & 1 & - & 0 \\ 1 & 0 & 1 & - \\ - & - & 0 & - \end{bmatrix}\mathbf{x} \qquad [\mathtt{x}](\mathbf{x}) = \begin{bmatrix} 0 & - & - & - \\ - & - & 0 & - \\ - & - & - & 0 \\ - & 0 & - & - \end{bmatrix}\mathbf{x}$$

$$[\triangleleft](\mathbf{x}) = \begin{bmatrix} 0 & - & 0 & - \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix}\mathbf{x} \qquad [\triangleright](\mathbf{x}) = \begin{bmatrix} 0 & - & - & - \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix}\mathbf{x}$$

With the above interpretations, the rules of $\mathcal{P}$ satisfy the following relations, which, by Theorem 2.15, proves that $\{\triangleleft\mathtt{b} \to \triangleleft\mathtt{ax}\}$ is terminating relative to $\mathcal{P}$.

[5] This SRS simulates the Abelian sandpile model [BTW87] over an infinite path. The symbols $\triangleleft$, $\mathtt{x}$, $\triangleright$ represent the vertices on the path. At each instant, the weight of a vertex is identified by the number of $\mathtt{a}$s and $\mathtt{b}$s to its left until the next occurrence of a vertex symbol.

$$[\triangleleft\texttt{b}](\mathbf{x}) = \begin{bmatrix} 1 & 0 & 1 & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} \end{bmatrix}\mathbf{x} \quad \gg \quad \begin{bmatrix} 0 & \text{—} & 0 & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} \end{bmatrix}\mathbf{x} = [\triangleleft\texttt{ax}](\mathbf{x})$$

$$[\texttt{aax}](\mathbf{x}) = \begin{bmatrix} 0 & \text{—} & \text{—} & \text{—} \\ 2 & 1 & 2 & 0 \\ 1 & 0 & 1 & \text{—} \\ \text{—} & \text{—} & 0 & \text{—} \end{bmatrix}\mathbf{x} \quad \geq \quad \begin{bmatrix} 0 & \text{—} & \text{—} & \text{—} \\ 1 & 1 & \text{—} & 0 \\ 1 & 0 & 1 & \text{—} \\ \text{—} & \text{—} & 0 & \text{—} \end{bmatrix}\mathbf{x} = [\texttt{bxa}](\mathbf{x})$$

$$[\texttt{ab}](\mathbf{x}) = \begin{bmatrix} 0 & \text{—} & \text{—} & \text{—} \\ 2 & 2 & 0 & 1 \\ 1 & 1 & \text{—} & 0 \\ 1 & 0 & 1 & \text{—} \end{bmatrix}\mathbf{x} \quad \geq \quad \begin{bmatrix} 0 & \text{—} & \text{—} & \text{—} \\ 2 & 2 & 0 & 1 \\ 1 & 1 & \text{—} & 0 \\ \text{—} & 0 & \text{—} & \text{—} \end{bmatrix}\mathbf{x} = [\texttt{ba}](\mathbf{x})$$

$$[\texttt{xb}](\mathbf{x}) = \begin{bmatrix} 0 & \text{—} & \text{—} & \text{—} \\ 1 & 0 & 1 & \text{—} \\ \text{—} & \text{—} & 0 & \text{—} \\ 1 & 1 & \text{—} & 0 \end{bmatrix}\mathbf{x} \quad \geq \quad \begin{bmatrix} 0 & \text{—} & \text{—} & \text{—} \\ 1 & 0 & 1 & \text{—} \\ \text{—} & \text{—} & 0 & \text{—} \\ \text{—} & \text{—} & \text{—} & 0 \end{bmatrix}\mathbf{x} = [\texttt{ax}](\mathbf{x})$$

$$[\texttt{a}\triangleright](\mathbf{x}) = \begin{bmatrix} 0 & \text{—} & \text{—} & \text{—} \\ 1 & \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} \end{bmatrix}\mathbf{x} \quad \geq \quad \begin{bmatrix} 0 & \text{—} & \text{—} & \text{—} \\ 1 & \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} \end{bmatrix}\mathbf{x} = [\texttt{ax}\triangleright](\mathbf{x})$$

2.4 Generalized Collatz Functions

We consider instances of the following generalization of the Collatz function. Its variants have commonly appeared in the literature [Lag85, Raw85, Wag85, BM90, Kaš92, Koh07, Mic15].

Definition 2.18 (Generalized Collatz function)**.** Let X be one of $\mathbb{N}$, $\mathbb{N}^+$, or $\mathbb{Z}$ and define $X_\perp \coloneqq X \cup \{\perp\}$. A function $f\colon X_\perp \to X_\perp$ is a *generalized Collatz function* if $f(\perp) = \perp$ and there exist an integer $d \geq 2$ and rational numbers $q_0, \ldots, q_{d-1}, r_0, \ldots, r_{d-1}$ such that for all $0 \leq i \leq d-1$ and all $n \in X$, we have a definition

$$\begin{aligned} &\text{either of the form} \quad f(n) = q_i n + r_i \quad \text{if } n \equiv i \pmod{d}, \\ &\text{or of the form} \quad f(n) = \perp \qquad\quad\; \text{if } n \equiv i \pmod{d}. \end{aligned}$$

In the above definition, we allow the representation of a partially defined function by mapping to $\perp$ in the undefined cases. We call a partial f convergent if all f-trajectories contain $\perp$.

Note that the Collatz function corresponds to the case $d = 2$, $q_0 = 1/2$, $r_0 = 0$, $q_1 = 3$, $r_1 = 1$. Although the Collatz function is by far the most widely studied case, there are several other concrete examples of generalized Collatz functions the convergence of which is worth studying due to their connections to open problems in number theory and computability theory. We discuss these cases in Appendix C.

3 Rewriting the Collatz Function

We start our discussion with systems that use unary representations and then demonstrate via examples that mixed-base representations can be more suitable for use with automated methods.

3.1 Rewriting in Unary

The following system of Zantema [Zan05] simulates the iterated application of the Collatz function to a number represented in unary, and it terminates upon reaching 1.

Example 3.1. $\mathcal{Z}$ denotes the following SRS, consisting of 5 symbols and 7 rules.

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl} \mathtt{h11} & \to & \mathtt{1h} & \mathtt{11h}\diamond & \to & \mathtt{11s}\diamond & \mathtt{h1}\diamond & \to & \mathtt{t11}\diamond \\ & & & \mathtt{1s} & \to & \mathtt{s1} & \mathtt{1t} & \to & \mathtt{t111} \\ & & & \diamond\mathtt{s} & \to & \diamond\mathtt{h} & \diamond\mathtt{t} & \to & \diamond\mathtt{h} \end{array}$$

This system can be seen as encoding the execution of a Turing machine with cells that can be contracted/expanded. The symbols $\mathtt{1}$ and $\diamond$ (blank) form the tape alphabet, while the symbols $\mathtt{h}$ (half), $\mathtt{s}$ (shift), $\mathtt{t}$ (triple) indicate the head along with the state of the machine. Through the following result, the Collatz conjecture can be reformulated as termination of string rewriting.

Theorem 3.2 ([Zan05, Theorem 16])**.** *$\mathcal{Z}$ is terminating if and only if the Collatz conjecture holds.*

While the forward direction of the above theorem is easy to see (since $\diamond\mathtt{h1}^{2n}\diamond \to^*_{\mathcal{Z}} \diamond\mathtt{h1}^{n}\diamond$ for $n > 1$ and $\diamond\mathtt{h1}^{2n+1}\diamond \to^*_{\mathcal{Z}} \diamond\mathtt{h1}^{3n+2}\diamond$ for $n \geq 0$), the backward direction is far from obvious because not every string corresponds to a valid configuration of the underlying machine.

As another example, consider the system below, which follows a theme similar to Example 3.1.

Example 3.3. $\mathcal{W}$ denotes the following SRS, consisting of 4 symbols and 4 rules.

$$\begin{array}{rcl} \mathtt{h11} & \to & \mathtt{1h} \\ \mathtt{1h}\diamond & \to & \mathtt{1t}\diamond \\ \mathtt{1t} & \to & \mathtt{t111} \\ \diamond\mathtt{t} & \to & \diamond\mathtt{h} \end{array}$$

Johannes Waldmann shared the above system (originally due to Zantema[6]) with us, mentioning that its termination has yet to be proved via automated methods. Nevertheless, there is a simple reason for its termination: It simulates the iterated application of a partial generalized Collatz function $W\colon \mathbb{N}^+_\perp \to \mathbb{N}^+_\perp$ defined as follows, which is easily seen to be convergent.

$$W(n) = \begin{cases} 3n/2 & \text{if } n \equiv 0 \pmod 2 \\ \perp & \text{if } n \equiv 1 \pmod 2 \end{cases}$$

[6] https://www.lri.fr/~marche/tpdb/tpdb-2.0/SRS/Zantema/z079.srs

3.1.1 Nonexistence of Proofs via Natural Matrix Interpretations

Natural matrix interpretations cannot be used to directly remove any of the rules from the above kind of unary rewriting systems that simulate certain maps, in particular the Collatz function.

Definition 3.4 ($\mathbb{N}$-rational sequence)**.** A sequence $x_1, x_2, \ldots \in \mathbb{N}$ is called $\mathbb{N}$-*rational* if there exists a matrix $\mathbf{M} \in \mathbb{N}^{d\times d}$ and vectors $\mathbf{v}, \mathbf{w} \in \mathbb{N}^d$ of some dimension d such that, for all n, we have $x_n = \mathbf{v}^\intercal \mathbf{M}^n \mathbf{w}$.

The following is a consequence of Berstel's theorem [RB10, Theorem 8.1.1]. (See also the remark by Soittola [Soi76, page 318].)

Theorem 3.5. *Let $x_1, x_2, \ldots$ be an $\mathbb{N}$-rational sequence. Then there exist $m, p \in \mathbb{N}$ such that, for all $j \in \{0, 1, \ldots, p-1\}$, each of the subsequences $\{x_{m+kp+j}\}_{k=0}^{\infty}$ has its nth element equal to $(1 + o(1))P(n)\alpha^n$ for some nonzero polynomial P and a constant $\alpha > 0$. In particular, each of these subsequences is monotonically nondecreasing after some finite point.*

Corollary 3.6. *There exists no $\mathbb{N}$-rational sequence $x_1, x_2, \ldots$ satisfying $x_{8n+1} > x_{9n+2}$ for all n.*

Proof.[7] Let p be as in Theorem 3.5, and consider n of the form $qp - 1$ for some large enough q. Then $(9n + 2) - 8n + 1 = n + 1 = qp$, so x_{8n+1} and x_{9n+2} belong to the same subsequence. This means that for all sufficiently large n of this form, we must have $x_{8n+1} \le x_{9n+2}$. □

To make use of the above facts in ruling out the existence of natural matrix interpretations of the form described in Section 2.3.1, we also need the following result proving that we can work with interpretations that are purely linear (instead of affine) by moving up to a higher dimension.

Theorem 3.7 ([EWZ08, Theorem 6])**.** *Let R be an SRS over the alphabet Σ, and let $[\sigma](\mathbf{x}) = \mathbf{M}_\sigma \mathbf{x} + \mathbf{v}_\sigma$ denote a d-dimensional affine interpretation for each $\sigma \in \Sigma$. Define a $(d+1)$-dimensional linear interpretation for each $\sigma \in \Sigma$ as*

$$[\sigma]^+(\mathbf{x}) := \mathbf{D}_\sigma \mathbf{x} = \begin{bmatrix} \mathbf{M}_\sigma & \mathbf{v}_\sigma \\ \mathbf{0}^\intercal & 1 \end{bmatrix} \mathbf{x},$$

where $\mathbf{0}$ denotes the d-dimensional zero vector. For each rule $\ell \to r \in R$, let $\mathbf{M}_\ell$, $\mathbf{M}_r$, $\mathbf{v}_\ell$, $\mathbf{v}_r$, $\mathbf{D}_\ell$, $\mathbf{D}_r$ denote the matrices and the vectors occurring in the corresponding interpretations (as obtained by composing the interpretations of the symbols). If we have $\mathbf{M}_\ell \gtrsim \mathbf{M}_r$ and $\mathbf{v}_\ell \gtrsim \mathbf{v}_r$, then $\mathbf{D}_\ell \gtrsim \mathbf{D}_r$. Additionally, if $\mathbf{v}_\ell > \mathbf{v}_r$, then $(D_\ell)_{1,d+1} > (D_r)_{1,d+1}$.

In informal terms, the above theorem states that if we have a collection of affine interpretations to remove a rule from a system (due to the strict decrease across the first elements of the corresponding two vectors), then there is also a collection of purely linear interpretations such that the top-right elements decrease strictly across the two matrices corresponding to this rule. With this fact at hand, we can prove the following.

[7] We thank Gjergji Zaimi for this argument (see https://mathoverflow.net/a/270372).

Theorem 3.8. *Let $\Sigma = \{\mathtt{1}, \diamond, \mathtt{h}, \mathtt{s}, \mathtt{t}\}$. There exists no collection $[\cdot]_\Sigma$ of natural matrix interpretations of any dimension d satisfying the requirements for an extended monotone Σ-algebra such that*

- *for at least one rule $\ell \to r \in \mathcal{Z}$ we have $[\ell](\mathbf{x}) > [r](\mathbf{x})$ for all $\mathbf{x} \in \mathbb{N}^d$, and*
- *for the remaining $\ell' \to r' \in \mathcal{Z}$ we have $[\ell'](\mathbf{x}) \gtrsim [r'](\mathbf{x})$ for all $\mathbf{x} \in \mathbb{N}^d$.*

Proof. Assume for a contradiction that such a collection of natural matrix interpretations exists for some dimension $d-1$, with each symbol $\sigma \in \Sigma$ interpreted as $[\sigma](\mathbf{x}) = \mathbf{M}_\sigma \mathbf{x} + \mathbf{v}_\sigma$ and each matrix $\mathbf{D}_\sigma$ defined as in Theorem 3.7. Let $\ell \to r \in \mathcal{Z}$ be a rule for which $[\ell](\mathbf{x}) > [r](\mathbf{x})$ for all $\mathbf{x} \in \mathbb{N}^{d-1}$, and denote $\rho = \{\ell \to r\}$. As the interpretations satisfy the requirements for extended monotonicity, for all $s, t \in \Sigma^*$, if $s \to_\rho t$, then $[s](\mathbf{x}) > [t](\mathbf{x})$ for all $\mathbf{x} \in \mathbb{N}^{d-1}$. This conclusion implies by (2) that $\mathbf{M}_s \gtrsim \mathbf{M}_t$ and $\mathbf{v}_s > \mathbf{v}_t$, which, by Theorem 3.7, further imply $\mathbf{D}_s \gtrsim \mathbf{D}_t$ and $(D_s)_{1,d} > (D_t)_{1,d}$. Thus, for all $s, t \in \Sigma^*$,

$$s \to_\rho t \quad \Longrightarrow \quad \mathbf{D}_s \gtrsim \mathbf{D}_t \text{ and } (D_s)_{1,d} > (D_t)_{1,d}. \tag{3}$$

In Zantema's system $\mathcal{Z}$, we can represent an arbitrary integer k by the string $\diamond\mathtt{h}\mathtt{1}^k\diamond$, interpreted as

$$\left[\diamond\mathtt{h}\mathtt{1}^k\diamond\right]^+(\mathbf{x}) = \mathbf{D}_{\diamond\mathtt{h}}\mathbf{D}_{\mathtt{1}}^k\mathbf{D}_\diamond\mathbf{x}.$$

Let $\mathbf{e}_i \in \mathbb{N}^d$ denote the ith standard basis vector, i.e., with 1 at the ith position and 0s elsewhere. Define

$$\begin{aligned} f(k) &\coloneqq \mathbf{e}_1^\intercal\left[\diamond\mathtt{h}\mathtt{1}^k\diamond\right]^+(\mathbf{e}_d) \\ &= \mathbf{e}_1^\intercal(\mathbf{D}_{\diamond\mathtt{h}}\mathbf{D}_{\mathtt{1}}^k\mathbf{D}_\diamond)\mathbf{e}_d \\ &= (\mathbf{e}_1^\intercal\mathbf{D}_{\diamond\mathtt{h}})\mathbf{D}_{\mathtt{1}}^k(\mathbf{D}_\diamond\mathbf{e}_d) \\ &= \mathbf{v}^\intercal\mathbf{D}_{\mathtt{1}}^k\mathbf{w}, \end{aligned} \tag{4}$$

where $\mathbf{v} = \mathbf{D}_{\diamond\mathtt{h}}^\intercal\mathbf{e}_1$ and $\mathbf{w} = \mathbf{D}_\diamond\mathbf{e}_d$ are fixed, so $f(1), f(2), \dots$ is an $\mathbb{N}$-rational sequence.

Now, since the system $\mathcal{Z}$ simulates the Collatz function (with the odd case incorporating an additional division by 2), for all numbers of the form $8n+1$ it can simulate the sequence of mappings $8n+1 \mapsto 12n+2 \mapsto 6n+1 \mapsto 9n+2$ that results from applying the Collatz map three times. Thus, for all n, we have

$$\diamond\mathtt{h}\mathtt{1}^{8n+1}\diamond \ \to_{\mathcal{Z}}^* \ \diamond\mathtt{h}\mathtt{1}^{9n+2}\diamond,$$

which crucially requires the use of every single rule in $\mathcal{Z}$ (as it involves applying both $m \mapsto m/2$ and $m \mapsto (3m+1)/2$). This implies in particular that the rule $\ell \to r$ is used in the above derivation. Recall also that the interpretations of all rules in $\mathcal{Z}$ are at least nonstrictly decreasing by assumption. This means along with (3) that, letting $s = \diamond\mathtt{h}\mathtt{1}^{8n+1}\diamond$ and $t = \diamond\mathtt{h}\mathtt{1}^{9n+2}\diamond$, we have $(D_s)_{1,d} > (D_t)_{1,d}$. From the definition of f, for all n,

$$f(8n+1) = \mathbf{e}_1^\intercal\mathbf{D}_s\mathbf{e}_d = (D_s)_{1,d} > (D_t)_{1,d} = \mathbf{e}_1^\intercal\mathbf{D}_t\mathbf{e}_d = f(9n+2),$$

but we already argued in Corollary 3.6 that no $\mathbb{N}$-rational sequence can satisfy this inequality for all n, contradiction. □

Natural matrix interpretations fail to be useful for the unary rewriting systems even when used in conjunction with dependency pairs. For the reader familiar with dependency pairs, we show how to adapt the above proof to go through in the setting where the dependency pair transformation is applied before searching for natural matrix interpretations. Recall that, for an SRS R over Σ, letting $\mathrm{DP}(R)$ denote the SRS consisting of all dependency pairs of R as defined by Arts and Giesl [AG00, Definition 3], we have $\mathrm{SN}(R)$ if and only if $\mathrm{SN}(\mathrm{DP}(R)_{\mathrm{top}} \,/\, R)$ [AG00, Theorem 6]. For a defined symbol $s \in \Sigma$, we write $s_\#$ to denote the corresponding tuple symbol.

Let $\mathcal{I} = \{\diamond_\#\mathtt{s} \to \diamond_\#\mathtt{h},\ \diamond_\#\mathtt{t} \to \diamond_\#\mathtt{h}\}$, and note that $\mathcal{I} \subseteq \mathrm{DP}(\mathcal{Z})$. With an automated termination prover, the following is straightforward to establish for Zantema's system $\mathcal{Z}$ after applying the dependency pair transformation and performing several steps of rule removal (e.g., using natural matrix interpretations).

Lemma 3.9. $\mathrm{SN}(\mathrm{DP}(\mathcal{Z})_{\mathrm{top}} \,/\, \mathcal{Z})$ *if and only if* $\mathrm{SN}(\mathcal{I}_{\mathrm{top}} \,/\, \mathcal{Z})$.

We prove below that neither rule in $\mathcal{I}$ can be removed using natural matrix interpretations. The proof is similar to that of Theorem 3.8, so we keep it relatively brief.

Theorem 3.10. *Let* $\Sigma = \{\mathtt{1}, \diamond, \diamond_\#, \mathtt{h}, \mathtt{s}, \mathtt{t}\}$. *There exists no collection* $[\cdot]_\Sigma$ *of natural matrix interpretations of any dimension* d *satisfying the requirements for a weakly monotone* Σ*-algebra such that*

- *for at least one rule* $\ell \to r \in \mathcal{I}$ *we have* $[\ell](\mathbf{x}) > [r](\mathbf{x})$ *for all* $\mathbf{x} \in \mathbb{N}^d$, *and*
- *for every rule* $\ell' \to r' \in \mathcal{Z}$ *we have* $[\ell'](\mathbf{x}) \gtrsim [r'](\mathbf{x})$ *for all* $\mathbf{x} \in \mathbb{N}^d$.

Proof. Assume for a contradiction that such a collection of natural matrix interpretations exists for some dimension $d-1$, with each symbol $\sigma \in \Sigma$ interpreted as $[\sigma](\mathbf{x}) = \mathbf{M}_\sigma\mathbf{x} + \mathbf{v}_\sigma$ and each matrix $\mathbf{D}_\sigma$ defined as in Theorem 3.7. Let $\ell \to r \in \mathcal{I}$ be a rule for which $[\ell](\mathbf{x}) > [r](\mathbf{x})$ for all $\mathbf{x} \in \mathbb{N}^{d-1}$, and denote $\rho = \{\ell \to r\}$. As the interpretations satisfy the requirements for only weak monotonicity, for all $s, t \in \Sigma^*$, if $s \to_{\rho_{\mathrm{top}}} t$, then $[s](\mathbf{x}) > [t](\mathbf{x})$ for all $\mathbf{x} \in \mathbb{N}^{d-1}$. Thus, for all $s, t \in \Sigma^*$,

$$s \to_{\rho_{\mathrm{top}}} t \qquad \Longrightarrow \qquad \mathbf{D}_s \gtrsim \mathbf{D}_t \text{ and } (D_s)_{1,d} > (D_t)_{1,d}. \tag{5}$$

We represent an arbitrary integer k by the string $\diamond_\#\mathtt{h}\mathtt{1}^k\diamond$, interpreted as

$$\left[\diamond_\#\mathtt{h}\mathtt{1}^k\diamond\right]^+(\mathbf{x}) = \mathbf{D}_{\diamond_\#\mathtt{h}}\mathbf{D}_{\mathtt{1}}^k\mathbf{D}_\diamond\mathbf{x}.$$

Similar to (4), we let $\mathbf{e}_i \in \mathbb{N}^d$ denote the ith standard basis vector and define $g(k) \coloneqq \mathbf{u}^\intercal\mathbf{D}_{\mathtt{1}}^k\mathbf{w}$, where $\mathbf{u} = \mathbf{D}_{\diamond_\#\mathtt{h}}^\intercal\mathbf{e}_1$ and $\mathbf{w} = \mathbf{D}_\diamond\mathbf{e}_d$ are fixed, so $g(1), g(2), \ldots$ is an $\mathbb{N}$-rational sequence.

For all n, we have

$$\diamond_\#\mathtt{h}\mathtt{1}^{8n+1}\diamond \ \to^*_{\mathcal{I}\cup\mathcal{Z}} \ \diamond_\#\mathtt{h}\mathtt{1}^{9n+2}\diamond,$$

which crucially requires the use of both rules in $\mathcal{I}$, and, moreover, these rules are applied only at the leftmost end of a string. In particular, the rule $\ell \to r$ is used in the above derivation to perform a top rewrite. Thus, by (5), for all n, we have $g(8n+1) > g(9n+2)$, which contradicts Corollary 3.6. □

Analogues of Theorems 3.8 and 3.10 also hold for the system $\mathcal{Z}^{\mathrm{rev}}$, with its rules reversed.[8] Moreover, by arguments similar to above, we can show that natural matrix interpretations fail

[8] Specifically, Theorem 3.10 holds for $\mathcal{Z}^{\mathrm{rev}}$ with $\mathcal{J} = \{\diamond_\#\mathtt{h11} \to \diamond_\#\mathtt{s11},\ \diamond_\#\mathtt{1h} \to \diamond_\#\mathtt{11t}\}$ in place of $\mathcal{I}$. The corresponding version of Lemma 3.9 is again straightforward to establish using an automated termination prover.

to prove the termination of $\mathcal{W}$. It is natural to expect that if a proof of the Collatz conjecture is to be produced by some automated method that relies on rewriting, then that method better be able to prove a statement as simple as the convergence of W. With this in mind, we describe an alternative rewriting system that simulates the Collatz function and terminates upon reaching 1. We then provide examples where the alternative system is more suitable for use with termination tools (for instance allowing a matrix interpretations proof of the convergence of W). The arguments in this section that prove Theorems 3.8 and 3.10 also appear not to apply to the alternative system, because the connection between rewrite sequences and $\mathbb{N}$-rational sequences gets lost when not using unary representations. More specifically, the proofs crucially use the fact that, since the system $\mathcal{Z}$ represents numbers in unary, it is possible to form a sequence such that the nth element is represented by some product $\mathbf{v}^{\intercal}\mathbf{M}^{n}\mathbf{w}$, where $\mathbf{v}$ and $\mathbf{w}$ are vectors and $\mathbf{M}$ is a matrix. There is no single such matrix that one can extract when using mixed binary–ternary, or even just binary, representations. Thus, the same kind of analysis, which uses the results known about $\mathbb{N}$-rational sequences, does not go through for the alternative system.

3.2 Rewriting in Mixed Base

In the mixed-base scheme, the overall idea is as follows. Given a number $n \in \mathbb{N}^{+}$, we write a mixed binary–ternary representation for it (noting that this representation is not unique). With this representation, as long as the least significant digit is binary, the parity of the number can be recognized by checking only this digit, as opposed to scanning the entire string when working in unary. This allows us to easily determine the correct case when applying the Collatz function. If the least significant digit is ternary, then the representation is rewritten (while preserving its value) to make this digit binary. Afterwards, since computing $n/2$ corresponds to erasing a trailing binary 0 and computing $3n+1$ corresponds to inserting a trailing ternary 1, applying the Collatz function takes a single rewrite step.

Intuitively, the rewriting system we present can be seen as applying the Collatz map to natural numbers written in binary, with some secondary computation performed using auxiliary symbols to facilitate the application of the map $n \mapsto 3n+1$. However, since termination of rewriting is concerned with sequences that arise from *any initial string* and *any possible application of the rules*, we need to consider the behavior of the system over any string that may be encountered, which includes even the strings that could not result from computing in binary with a prescribed strategy for rule application. The mixed-base scheme is simply a way to give an arithmetical meaning to all strings and all possible rewrite steps.

More formally, a mixed-base numeral system [Can69] is a numeral system where the base changes across positions, which we define as follows.

Definition 3.11 (Mixed-base representation)**.** Fix $B \subseteq \mathbb{N}$ to be a set of bases. Let $n_1, \ldots, n_k \in \mathbb{N}$ and let $b_1, \ldots, b_k \in B$. If we have for each $2 \leq i \leq k$ that $n_i < b_i$, then the matrix

$$N = \begin{bmatrix} n_1 & n_2 & \ldots & n_k \\ b_1 & b_2 & \ldots & b_k \end{bmatrix}$$

is called a *mixed B-ary representation*. To reduce clutter, we identify the matrix N with a string $(n_1)_{(b_1)}(n_2)_{(b_2)} \ldots (n_k)_{(b_k)}$, where each $(n_i)_{(b_i)}$ is viewed as a single symbol denoting the *b_i-ary digit n_i*.

The string N in the above definition represents the number

$$\mathrm{Val}(N) \coloneqq \sum_{i=1}^{k} n_i \prod_{j=i+1}^{k} b_j. \tag{6}$$

Observing that the addition of leading zeros (symbols 0_b for any b) to a string does not change its value, we may assume without loss of generality that $n_1 > 0$. Furthermore, b_1 does not affect the value of the string, so we replace it by zero (noting that by Definition 3.11 no other base is zero).

Example 3.12.

1. With $B = \{2\}$, we have the binary numeral system. In the notation of Definition 3.11 we write $1_2 0_2 1_2 1_2$ to represent the number 11.

2. With $B = \{2, 3\}$, we obtain the mixed binary–ternary system that we will use in the rest of this paper. In this system we may write, for instance, $1_0 2_3 1_2$ or $2_0 1_2 1_2$ or $1_0 1_2 2_3$ to represent the number 11.

Now, define $\beta_b^n(x) \coloneqq bx + n$. After rearranging (6), we see that for a mixed B-ary string $N = (n_1)_{(b_1)}(n_2)_{(b_2)} \dots (n_k)_{(b_k)}$, its value $\mathrm{Val}(N)$ is also given by evaluating the composite function

$$\Gamma_N(x) \coloneqq (\beta_{b_k}^{n_k} \circ \beta_{b_{k-1}}^{n_{k-1}} \circ \cdots \circ \beta_{b_1}^{n_1})(x) \tag{7}$$

at any value (because $b_1 = 0$ implies that the innermost function $\beta_0^{n_1}(x) = n_1$ is constant). This gives us a string and a function view of the same representation, and we will switch between them as appropriate. In doing so, we also conflate the symbols and the corresponding functions, referring to β_b^n as n_b.

As the last ingredient before describing the rewriting system, we observe that we can write $(\beta_b^n \circ \beta_c^m)(x) = bcx + bm + n$ equivalently as another composition $(\beta_c^{m'} \circ \beta_b^{n'})(x) = cbx + cn' + m'$ for some suitable $0 \leq n' < b$ and $0 \leq m' < c$. This allows us to swap the bases of adjacent positions while preserving the value of the string.

From this point on, we constrain ourselves to the mixed $\{2, 3\}$-ary (binary–ternary) representations as we shift our focus to simulating the Collatz function (noting that it is possible to adapt the resulting rewriting system for other instances of the general case). More precisely, we simulate the following redefinition of the Collatz function, where the odd case incorporates an additional division by 2.

$$T(n) = \begin{cases} \frac{n}{2} & \text{if } n \equiv 0 \pmod 2 \\ \frac{3n+1}{2} & \text{if } n \equiv 1 \pmod 2 \end{cases}$$

We will describe an SRS $\mathcal{T}$ over the symbols $\{\mathtt{f}, \mathtt{t}, \mathtt{0}, \mathtt{1}, \mathtt{2}, \triangleleft, \triangleright\}$ that simulates the iterated application of the Collatz function and terminates upon reaching 1. The symbols $\mathtt{f}, \mathtt{t}$ correspond to binary digits $0_2, 1_2$; and $\mathtt{0}, \mathtt{1}, \mathtt{2}$ to ternary digits $0_3, 1_3, 2_3$. The symbol $\triangleleft$ marks the beginning of a string while also standing for the most significant digit (without loss of generality assumed to be 1_0) and $\triangleright$ marks the end of a string while also standing for the redundant trailing digit 0_1. Consider the functional view of these symbols:

$$\begin{array}{rcl} \mathtt{f}(x) &=& 2x \\ \mathtt{t}(x) &=& 2x+1 \end{array} \qquad \begin{array}{rcl} \mathtt{0}(x) &=& 3x \\ \mathtt{1}(x) &=& 3x+1 \\ \mathtt{2}(x) &=& 3x+2 \end{array} \qquad \begin{array}{rcl} \triangleleft(x) &=& 1 \\ \triangleright(x) &=& x \end{array} \tag{8}$$

Each positive integer can be expressed as some composition of these functions, which corresponds to a string as per our previous discussion.

Example 3.13. Viewing the expression $\lhd(x)$ as the constant 1, we can write

$$19 = \mathrm{Val}(\lhd\texttt{0f1}\rhd) = \rhd(\texttt{1}(\texttt{f}(\texttt{0}(\lhd(x))))).$$

The string representation ends with a ternary symbol, so we will rewrite it.

With the function view, we have

$$\texttt{1}(\texttt{f}(x)) = 3(2x) + 1 = 6x + 1 = 2(3x) + 1 = \texttt{t}(\texttt{0}(x)).$$

This shows that we could also write $19 = \mathrm{Val}(\lhd\texttt{00t}\rhd)$, which now ends with the binary digit 1_2. This gives us the rewrite rule $\texttt{f1} \to \texttt{0t}$. We can now apply the Collatz function to this representation by rewriting only the rightmost two symbols of the string since

$$T(\rhd(\texttt{t}(x))) = \frac{3(2x+1)+1}{2} = \frac{6x+4}{2} = 3x + 2 = (\rhd(\texttt{2}(x))).$$

This gives us the rewrite rule $\texttt{t}\rhd \to \texttt{2}\rhd$. After applying this rule to the string $\lhd\texttt{00t}\rhd$, we indeed obtain $T(19) = 29 = \mathrm{Val}(\lhd\texttt{002}\rhd)$.

In the manner of the above example, we compute all the necessary transformations and obtain the following 11-rule SRS $\mathcal{T}$.

$$\mathcal{D}_T = \left\{ \begin{array}{l} \texttt{f}\rhd \to \rhd \\ \texttt{t}\rhd \to \texttt{2}\rhd \end{array} \right\} \qquad \mathcal{A} = \left\{ \begin{array}{ll} \texttt{f0} \to \texttt{0f} & \texttt{t0} \to \texttt{1t} \\ \texttt{f1} \to \texttt{0t} & \texttt{t1} \to \texttt{2f} \\ \texttt{f2} \to \texttt{1f} & \texttt{t2} \to \texttt{2t} \end{array} \right\} \qquad \mathcal{B} = \left\{ \begin{array}{l} \lhd\texttt{0} \to \lhd\texttt{t} \\ \lhd\texttt{1} \to \lhd\texttt{ff} \\ \lhd\texttt{2} \to \lhd\texttt{ft} \end{array} \right\}$$

This SRS is split into subsystems $\mathcal{D}_T$ (dynamic rules for T) and $\mathcal{X} = \mathcal{A} \cup \mathcal{B}$ (auxiliary rules). The two rules in $\mathcal{D}_T$ encode the application of the Collatz function T, while the rules in $\mathcal{X}$ serve to push binary symbols towards the rightmost end of the string by swapping the bases of adjacent positions without changing the represented value.

Example 3.14 (Rewrite sequence of $\mathcal{T}$)**.** Consider the string $s = \lhd\texttt{ff0}\rhd$, which represents the number 12. Below is a possible rewrite sequence of $\mathcal{T}$ that starts from s, with the corresponding values (under the interpretations from (8)) displayed above the strings. Underlines indicate the parts of the strings where the rules are applied.

$$\overset{12}{\lhd\texttt{f}\underline{\texttt{f0}}\rhd} \to_{\mathcal{A}} \overset{12}{\lhd\texttt{f0}\underline{\texttt{f}\rhd}} \to_{\mathcal{D}_T} \overset{6}{\lhd\underline{\texttt{f0}}\rhd} \to_{\mathcal{A}} \overset{6}{\lhd\texttt{0}\underline{\texttt{f}\rhd}} \to_{\mathcal{D}_T} \overset{3}{\underline{\lhd\texttt{0}}\rhd} \to_{\mathcal{B}} \overset{3}{\lhd\underline{\texttt{t}\rhd}} \to_{\mathcal{D}_T} \overset{5}{\underline{\lhd\texttt{2}}\rhd} \to_{\mathcal{B}} \overset{5}{\lhd\texttt{f}\underline{\texttt{t}\rhd}}$$

$$\to_{\mathcal{D}_T} \overset{8}{\lhd\underline{\texttt{f2}}\rhd} \to_{\mathcal{A}} \overset{8}{\underline{\lhd\texttt{1}}\texttt{f}\rhd} \to_{\mathcal{B}} \overset{8}{\lhd\texttt{ff}\underline{\texttt{f}\rhd}} \to_{\mathcal{D}_T} \overset{4}{\lhd\texttt{f}\underline{\texttt{f}\rhd}} \to_{\mathcal{D}_T} \overset{2}{\lhd\underline{\texttt{f}\rhd}} \to_{\mathcal{D}_T} \overset{1}{\lhd\rhd}$$

The trajectory of T would continue upon reaching 1; however, in order to be able to formulate the Collatz conjecture as a termination problem, $\mathcal{T}$ is made in such a way that its rewrite sequences stop upon reaching the string representation $\lhd\rhd$ of 1 since no rule is applicable.

Termination of the subsystems of $\mathcal{T}$ with $\mathcal{B}$ or $\mathcal{D}_T$ removed is easily seen. However, since we have matrix interpretations at our disposal, let us give a compact and formal proof.

Lemma 3.15. $\mathrm{SN}(\mathcal{T} \setminus \mathcal{B})$ *and* $\mathrm{SN}(\mathcal{T} \setminus \mathcal{D}_T)$.

Proof. It is easily checked that the interpretations below show $\mathrm{SN}((\mathcal{T} \setminus \mathcal{B})^{\mathrm{rev}})$, which implies $\mathrm{SN}(\mathcal{T} \setminus \mathcal{B})$ by Lemma 2.8.

$$[\mathtt{f}](x) = [\mathtt{t}](x) = 2x + 1 \qquad [\triangleright](x) = x \qquad [\mathtt{0}](x) = [\mathtt{1}](x) = [\mathtt{2}](x) = 2x$$

Similarly, the below interpretations show $\mathrm{SN}((\mathcal{T} \setminus \mathcal{D}_T)^{\mathrm{rev}})$, which implies $\mathrm{SN}(\mathcal{T} \setminus \mathcal{D}_T)$ by Lemma 2.8.

$$[\mathtt{f}](x) = [\mathtt{t}](x) = [\triangleleft](x) = x + 1 \qquad [\mathtt{0}](x) = [\mathtt{1}](x) = [\mathtt{2}](x) = 4x$$ □

When considering the termination of $\mathcal{T}$ (on all initial strings), it suffices to limit the discussion to initial strings of a specific form that we have been working with so far, e.g., in Examples 3.13 and 3.14.

Lemma 3.16. *If* $\mathcal{T}$ *is terminating on all initial strings of the* canonical form $\triangleleft(\mathtt{f}|\mathtt{t}|\mathtt{0}|\mathtt{1}|\mathtt{2})^*\triangleright$, *then* $\mathcal{T}$ *is terminating (on all initial strings).*

For the proof of Lemma 3.16, we will use *type introduction*, which is a technique for proving termination by switching to typed rewriting. We refer the reader to Sabel and Zantema [SZ17, Section 3.1] for background. We also offer a more elementary proof in Appendix A.

Proof of Lemma 3.16. Let $\mathbb{T} = \{\rho, \sigma, \tau\}$ be a set of types. It is straightforward to see that, with respect to the typing

$$\triangleleft : \sigma \to \tau, \qquad \mathtt{f}, \mathtt{t}, \mathtt{0}, \mathtt{1}, \mathtt{2} : \sigma \to \sigma, \qquad \triangleright : \rho \to \sigma,$$

the SRS $\mathcal{T}$ is well-typed, so it is terminating (in the untyped setting) if and only if it is terminating in the typed setting [SZ17, Corollary 3.4]. Due to Lemma 3.15, a well-typed string with σ as its source or target type cannot admit an infinite rewrite sequence for $\mathcal{T}$. Thus, a well-typed string admits an infinite rewrite sequence only if it is of type $\rho \to \tau$, which implies that it is of the canonical form. □

Note that the converse of the above lemma is obvious, since if $\mathcal{T}$ is terminating then it is terminating on all initial strings of any form, in particular $\triangleleft(\mathtt{f}|\mathtt{t}|\mathtt{0}|\mathtt{1}|\mathtt{2})^*\triangleright$.

As a whole, the rewriting system $\mathcal{T}$ simulates the iterated application of T (except at 1).

Theorem 3.17. $\mathcal{T}$ *is terminating if and only if* T *is convergent (i.e., the Collatz conjecture holds).*

Proof. Without loss of generality (due to Lemma 3.16), consider only the strings of the canonical form $\triangleleft(\mathtt{f}|\mathtt{t}|\mathtt{0}|\mathtt{1}|\mathtt{2})^*\triangleright$ throughout this proof. We will show that, given such a string N, all rewrite sequences for $\mathcal{T}$ that start from N simulate the T-trajectory of $\mathrm{Val}(N)$, except when $\mathrm{Val}(N) = 1$. More specifically, we prove the following and use them to deduce the theorem statement.

Claim 1. *For any pair* N, N' *of strings, if* $N \to_{\mathcal{X}} N'$ *then* $\mathrm{Val}(N') = \mathrm{Val}(N)$, *and if* $N \to_{\mathcal{D}_T} N'$ *then* $\mathrm{Val}(N') = T(\mathrm{Val}(N))$.

Claim 2. *For any string* N *with* $\mathrm{Val}(N) = 1$, *there exists no string* N' *satisfying* $N \to_{\mathcal{T}} N'$.

Claim 3. *For any integer $\nu > 1$ and any string N with $\mathrm{Val}(N) = \nu$, there exists a string N' such that $N \to^*_{\mathcal{T}} N'$ and $\mathrm{Val}(N') = T(\nu)$.*

Proof of Claim 1. Assume $N \to_{\mathcal{X}} N'$. Then there exist strings X, Y and some rule $\ell \to r \in \mathcal{X}$ such that $N = X\ell Y$ and $N' = XrY$. Recalling the definition of Γ from (7), we have $\mathrm{Val}(N') = \mathrm{Val}(N)$ if and only if $\Gamma_Y \circ \Gamma_r \circ \Gamma_X = \Gamma_Y \circ \Gamma_\ell \circ \Gamma_X$ for all $\ell \to r \in \mathcal{X}$. This holds if and only if $\Gamma_r = \Gamma_\ell$ for all $\ell \to r \in \mathcal{X}$. With the functional views of the symbols as in (8), we indeed have:

$$\begin{array}{lcllllll}
\Gamma_{\mathtt{0f}}(x) &=& \mathtt{f}(\mathtt{0}(x)) &=& 6x &=& \mathtt{0}(\mathtt{f}(x)) = \Gamma_{\mathtt{f0}}(x) \\
\Gamma_{\mathtt{0t}}(x) &=& \mathtt{t}(\mathtt{0}(x)) &=& 6x+1 &=& \mathtt{1}(\mathtt{f}(x)) = \Gamma_{\mathtt{f1}}(x) \\
\Gamma_{\mathtt{1f}}(x) &=& \mathtt{f}(\mathtt{1}(x)) &=& 6x+2 &=& \mathtt{2}(\mathtt{f}(x)) = \Gamma_{\mathtt{f2}}(x) \\
\Gamma_{\mathtt{1t}}(x) &=& \mathtt{t}(\mathtt{1}(x)) &=& 6x+3 &=& \mathtt{0}(\mathtt{t}(x)) = \Gamma_{\mathtt{t0}}(x) \\
\Gamma_{\mathtt{2f}}(x) &=& \mathtt{f}(\mathtt{2}(x)) &=& 6x+4 &=& \mathtt{1}(\mathtt{t}(x)) = \Gamma_{\mathtt{t1}}(x) \\
\Gamma_{\mathtt{2t}}(x) &=& \mathtt{t}(\mathtt{2}(x)) &=& 6x+5 &=& \mathtt{2}(\mathtt{t}(x)) = \Gamma_{\mathtt{t2}}(x) \\
\Gamma_{\triangleleft\mathtt{t}}(x) &=& \mathtt{t}(\triangleleft(x)) &=& 3 &=& \mathtt{0}(\triangleleft(x)) = \Gamma_{\triangleleft\mathtt{0}}(x) \\
\Gamma_{\triangleleft\mathtt{ff}}(x) &=& \mathtt{f}(\mathtt{f}(\triangleleft(x))) &=& 4 &=& \mathtt{1}(\triangleleft(x)) = \Gamma_{\triangleleft\mathtt{1}}(x) \\
\Gamma_{\triangleleft\mathtt{ft}}(x) &=& \mathtt{t}(\mathtt{f}(\triangleleft(x))) &=& 5 &=& \mathtt{2}(\triangleleft(x)) = \Gamma_{\triangleleft\mathtt{2}}(x)
\end{array}$$

This proves the first part of the claim.

Now assume $N \to_{\mathcal{D}_T} N'$. Again, there exist a string X and some rule $\ell \to r \in \mathcal{D}_T$ such that $N = X\ell$ and $N' = Xr$ (since the rules in $\mathcal{D}_T$ can be applied only at the rightmost end of the string). We have $\mathrm{Val}(N') = T(\mathrm{Val}(N))$ if and only if $\Gamma_r \circ \Gamma_X = T \circ \Gamma_\ell \circ \Gamma_X$ for all $\ell \to r \in \mathcal{D}_T$. This holds if and only if $\Gamma_r = T \circ \Gamma_\ell$ for all $\ell \to r \in \mathcal{D}_T$. There are only two rules to check, and indeed we have:

$$\begin{array}{lcllllllll}
\Gamma_{\triangleright}(x) &=& \triangleright(x) &=& x &=& T(2x) &=& T(\triangleright(\mathtt{f}(x))) &= T(\Gamma_{\mathtt{f}\triangleright}(x)) \\
\Gamma_{\mathtt{2}\triangleright}(x) &=& \triangleright(\mathtt{2}(x)) &=& 3x+2 &=& T(2x+1) &=& T(\triangleright(\mathtt{t}(x))) &= T(\Gamma_{\mathtt{t}\triangleright}(x))
\end{array}$$

This proves the second part of the claim. □

Proof of Claim 2. The string $\triangleleft\triangleright$ has $\mathrm{Val}(\triangleleft\triangleright) = 1$, and inserting any symbol from $\{\mathtt{f}, \mathtt{t}, \mathtt{0}, \mathtt{1}, \mathtt{2}\}$ in between the delimiters increases the value of the string, so $\triangleleft\triangleright$ is the unique string representing 1. None of the rules in $\mathcal{T}$ apply to $\triangleleft\triangleright$, which proves the claim. □

Proof of Claim 3. Let $\nu > 1$ be an integer, and let N be a string with $\mathrm{Val}(N) = \nu$. We will describe a rewrite sequence $N \to^*_{\mathcal{X}} N' \to_{\mathcal{D}_T} N''$, which implies by Claim 1 that $\mathrm{Val}(N'') = T(\nu)$. For the rest of the proof let b_i and t_i stand for binary and ternary symbols, respectively. Assume that N contains some binary symbol ($\mathtt{f}$ or $\mathtt{t}$), since otherwise it is of the form $\triangleleft(\mathtt{0}|\mathtt{1}|\mathtt{2})^*\triangleright$ and we can apply some rule from $\mathcal{B}$ to produce a binary symbol. Then the string contains some substring $b_1 t_2 t_3 \ldots t_k \triangleright$. We rewrite this substring into $t'_1 b_2 t_3 \ldots t_k \triangleright$ by applying some rule from $\mathcal{X}$. For each $1 < i < k$, assuming the current string contains some substring $b_i t_{i+1} \ldots t_k \triangleright$, we rewrite it into $t'_i b_{i+1} t_{i+2} \ldots t_k \triangleright$. In the end we obtain a string N' with $\mathrm{Val}(N') = \nu$ that contains some $b_k \triangleright$, so we apply some rule from $\mathcal{D}_T$ to obtain N'' such that $\mathrm{Val}(N'') = T(\nu)$. □

Assume that the SRS $\mathcal{T}$ is not terminating. Let $(N_i)_{i=0}^\infty$ be an infinite rewrite sequence for $\mathcal{T}$, and consider the sequence $(\nu_i)_{i=0}^\infty$, where $\nu_i = \mathrm{Val}(N_i)$. By Claim 1, for all $i \in \mathbb{N}$, we have either $\nu_{i+1} = \nu_i$ or $\nu_{i+1} = T(\nu_i)$ depending on the rewrite rule used. Let $I = \{i \in \mathbb{N} \mid \nu_{i+1} = T(\nu_i)\}$, which is the same as the set of indices at which some rule from $\mathcal{D}_T$ is applied in the rewrite sequence. By Lemma 3.15, the subsystem $\mathcal{X} = \mathcal{T} \setminus \mathcal{D}_T$ is terminating, so the rewrite $\rightarrow_{\mathcal{D}_T}$ is performed infinitely many times in the sequence. This implies that I is infinite, proving that the sequence $(\nu_i)_{i\in I}$ is a Collatz trajectory. Moreover, this trajectory is nonconvergent, i.e., it does not contain 1, since otherwise the rewrite sequence could not be infinite (because of Claim 2). This proves the backward implication in the theorem statement.

Assume that the function T is not convergent. Let $(\nu_i)_{i=0}^{\infty}$ be a nonconvergent trajectory, and note that $\nu_i > 1$ for all $i \in \mathbb{N}$. We will inductively define a sequence $(N_i)_{i=0}^{\infty}$ such that $\mathrm{Val}(N_i) = \nu_i$ for all $i \in \mathbb{N}$. Let N_0 be any string with $\mathrm{Val}(N_0) = \nu_0$. For each $i \in \mathbb{N}^+$, assuming $\mathrm{Val}(N_{i-1}) = \nu_{i-1}$, let N_i be a string such that $N_{i-1} \rightarrow_{\mathcal{T}}^{*} N_i$ and $\mathrm{Val}(N_i) = T(\nu_{i-1}) = \nu_i$, which exists by Claim 3. Defined in this manner, the sequence $(N_i)_{i=0}^{\infty}$ satisfies $N_i \rightarrow_{\mathcal{T}}^{*} N_{i+1}$ for all $i \in \mathbb{N}$, meaning that $\mathcal{T}$ is not terminating. This proves the forward implication in the theorem statement, which concludes the proof. □

When trying to remove a rule in $\mathcal{D}_T$ or $\mathcal{B}$ it suffices to show relative top termination, allowing us to use weakly (instead of extended) monotone algebras when applying Theorem 2.15 and take advantage of the more relaxed constraints when searching for matrix interpretations. As we mentioned back in Section 2.1, the lemma below encapsulates dependency pairs, and it can in fact be automatically proved by the existing termination tools via the dependency pair transformation followed by an application of the dependency pair framework [GTSF06].[9]

Lemma 3.18.

1. *For each subset* $\mathcal{R} \subseteq \mathcal{B}$*, if* $\mathrm{SN}(\mathcal{R}_{\mathrm{top}} \,/\, \mathcal{T})$ *then* $\mathrm{SN}(\mathcal{R} \,/\, \mathcal{T})$*.*
2. *For each subset* $\mathcal{Q} \subseteq \mathcal{D}_T$*, if* $\mathrm{SN}(\mathcal{Q}^{\mathrm{rev}}_{\mathrm{top}} \,/\, \mathcal{T}^{\mathrm{rev}})$ *then* $\mathrm{SN}(\mathcal{Q}^{\mathrm{rev}} \,/\, \mathcal{T}^{\mathrm{rev}})$*.*

Proof. Without loss of generality (due to Lemma 3.16), consider only the rewrite sequences that start with some string of the canonical form $\triangleleft(\mathtt{f}|\mathtt{t}|\mathtt{0}|\mathtt{1}|\mathtt{2})^{*}\triangleright$ (resp. its reversal).

Let $\mathcal{R} \subseteq \mathcal{B}$ (resp. $\mathcal{Q} \subseteq \mathcal{D}_T$). The rules in $\mathcal{R}$ (resp. $\mathcal{Q}^{\mathrm{rev}}$) can be applied only at the leftmost end as they are all of the form $\triangleleft s \rightarrow \triangleleft t$ for some s, t not containing $\triangleleft$ (resp. $\triangleright s' \rightarrow \triangleright t'$ for some s', t' not containing $\triangleright$).

Assume $\neg\,\mathrm{SN}(\mathcal{R} / \mathcal{T})$ (resp. $\neg\,\mathrm{SN}(\mathcal{Q}^{\mathrm{rev}} / \mathcal{T}^{\mathrm{rev}})$). Then there exists an infinite rewrite sequence for $\mathcal{T}$ (resp. $\mathcal{T}^{\mathrm{rev}}$) where the rules from $\mathcal{R}$ (resp. $\mathcal{Q}^{\mathrm{rev}}$) are applied infinitely many times. As these rules can be applied only at the leftmost end, this implies relative top nontermination, i.e., $\neg\,\mathrm{SN}(\mathcal{R}_{\mathrm{top}} \,/\, \mathcal{T})$ (resp. $\neg\,\mathrm{SN}(\mathcal{Q}^{\mathrm{rev}}_{\mathrm{top}} \,/\, \mathcal{T}^{\mathrm{rev}})$). □

3.3 Rewriting with a Hybrid System

As another alternative to the rewriting system $\mathcal{Z}$, instead of fully discarding unary in favor of a positional numeral system we may adopt a hybrid between the two approaches and design a system to increment/decrement by 1 or multiply/divide by 2 or 3 in a single rewrite. We will describe an SRS over the symbols $\{\mathtt{i}, \mathtt{f}, \mathtt{0}, \triangleleft, \triangleright\}$ viewed as follows.

$$\begin{aligned} \mathtt{i}(x) &= x + 1 \\ \mathtt{f}(x) &= 2x \\ \mathtt{0}(x) &= 3x \\ \triangleleft(x) &= 1 \\ \triangleright(x) &= x \end{aligned}$$

[9] We thank an anonymous reviewer of CADE for pointing out this fact.

In essentially the same manner as we did for the system $\mathcal{T}$, we have a set of dynamic rules that simulate the application of the Collatz function and a set of auxiliary rules that update the string representation so that the parity of the corresponding number can be recognized in a single step. We have the following 7-rule SRS $\mathcal{L}$ (designed by Luke Schaeffer).

$$\mathcal{D}'_T = \left\{ \begin{array}{rcl} \mathtt{f}\triangleright & \to & \triangleright \\ \mathtt{fi}\triangleright & \to & \mathtt{0ii}\triangleright \end{array} \right\} \quad \mathcal{A}' = \left\{ \begin{array}{rcl} \mathtt{fii} & \to & \mathtt{if} \\ \mathtt{i0} & \to & \mathtt{0iii} \\ \mathtt{f0} & \to & \mathtt{0f} \end{array} \right\} \quad \mathcal{B}' = \left\{ \begin{array}{rcl} \triangleleft\mathtt{i} & \to & \triangleleft\mathtt{f} \\ \triangleleft\mathtt{0} & \to & \triangleleft\mathtt{fi} \end{array} \right\}$$

As long as the string ends with $\mathtt{f}\triangleright$ or $\mathtt{fi}\triangleright$, the parity of the represented number is apparent without needing to scan the whole string, which allows the Collatz function to be applied in a single rewrite. After the application of a dynamic rule, the string may no longer end with $\mathtt{f}\triangleright$ or $\mathtt{fi}\triangleright$, so the auxiliary rules in $\mathcal{X}' = \mathcal{A}' \cup \mathcal{B}'$ push the symbol $\mathtt{f}$ towards the rightmost end of the string while preserving its value. Thus, $\mathcal{L}$ is yet another rewriting system that simulates the iterated application of the Collatz function.

Theorem 3.19. *$\mathcal{L}$ is terminating if and only if T is convergent.*

The proof of the above theorem is essentially the same as that of Theorem 3.17 once it is checked that the dynamic rules do indeed apply T and that the auxiliary rules do indeed preserve the value of a string under the functional view of the symbols.

4 Automated Proofs

We adapt the rewriting system $\mathcal{T}$ for different generalized Collatz functions to explore the effectiveness of the mixed-base scheme on weakened variants of the Collatz conjecture.

All of the rewriting systems, scripts to reproduce the proofs, and our minimal implementation of a termination prover are available at https://github.com/emreyolcu/rewriting-collatz. In particular, any omitted relative termination proof that we refer to in this section is available at https://github.com/emreyolcu/rewriting-collatz/tree/main/proofs. We ran the large-scale experiments in this section on AWS EC2 m5.metal instances.

4.1 A Simple Example

Earlier we mentioned a generalized Collatz function W as a simple example that could serve as a sanity check for an automated method aiming to solve Collatz-like problems. With the mixed binary–ternary scheme, this function can be seen to be simulated by the system $\mathcal{W}' = \{\mathtt{f}\triangleright \to \mathtt{0}\triangleright\} \cup \mathcal{X}$. A small matrix interpretations proof is found for this system in less than a second, in contrast to its variant $\mathcal{W}$ that uses unary representations for which no automated proof is known. Additionally, the function W can be simulated by the rewriting system $\mathcal{W}'' = \{\mathtt{f}\triangleright \to \mathtt{0}\triangleright\} \cup \mathcal{X}'$ adapted from the hybrid system $\mathcal{L}$, although we were unable to find an automated proof of termination for $\mathcal{W}''$.

Theorem 4.1. $\mathrm{SN}(\mathcal{W}')$.

Proof. The interpretations below prove $\mathrm{SN}(\{\triangleright\mathtt{f} \to \triangleright\mathtt{0}\} \;/\; \mathcal{X}^{\mathrm{rev}})$:

$$[\mathtt{f}](\mathbf{x}) = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\mathbf{x} + \begin{bmatrix}1\\1\end{bmatrix} \qquad [\mathtt{t}](\mathbf{x}) = \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}1\\0\end{bmatrix}$$

$$[\triangleleft](\mathbf{x}) = \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\mathbf{x} \qquad [\triangleright](\mathbf{x}) = \begin{bmatrix}1 & 2\\0 & 0\end{bmatrix}\mathbf{x}$$

$$[\mathtt{0}](\mathbf{x}) = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\mathbf{x} + \begin{bmatrix}2\\0\end{bmatrix} \qquad [\mathtt{1}](\mathbf{x}) = \begin{bmatrix}1 & 0\\1 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}2\\2\end{bmatrix} \qquad [\mathtt{2}](\mathbf{x}) = \begin{bmatrix}1 & 0\\1 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}2\\2\end{bmatrix}$$

With the above interpretations of the symbols, the rules of $\mathcal{W}'^{\mathrm{rev}}$ satisfy the following relations, which, by Theorem 2.15, proves that $\{\triangleright\mathtt{f} \to \triangleright\mathtt{0}\}$ is terminating relative to $\mathcal{X}^{\mathrm{rev}}$.

$$[\triangleright\mathtt{f}](\mathbf{x}) = \begin{bmatrix}1 & 2\\0 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}3\\0\end{bmatrix} \quad > \quad \begin{bmatrix}1 & 2\\0 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}2\\0\end{bmatrix} = [\triangleright\mathtt{0}](\mathbf{x})$$

$$[\mathtt{0f}](\mathbf{x}) = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\mathbf{x} + \begin{bmatrix}3\\1\end{bmatrix} \quad \gtrsim \quad \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\mathbf{x} + \begin{bmatrix}3\\1\end{bmatrix} = [\mathtt{f0}](\mathbf{x})$$

$$[\mathtt{1f}](\mathbf{x}) = \begin{bmatrix}1 & 0\\1 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}3\\3\end{bmatrix} \quad \gtrsim \quad \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}3\\0\end{bmatrix} = [\mathtt{t0}](\mathbf{x})$$

$$[\mathtt{2f}](\mathbf{x}) = \begin{bmatrix}1 & 0\\1 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}3\\3\end{bmatrix} \quad \gtrsim \quad \begin{bmatrix}1 & 0\\1 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}3\\3\end{bmatrix} = [\mathtt{f1}](\mathbf{x})$$

$$[\mathtt{0t}](\mathbf{x}) = \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}3\\0\end{bmatrix} \quad \gtrsim \quad \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}3\\0\end{bmatrix} = [\mathtt{t1}](\mathbf{x})$$

$$[\mathtt{1t}](\mathbf{x}) = \begin{bmatrix}1 & 0\\1 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}3\\3\end{bmatrix} \quad \gtrsim \quad \begin{bmatrix}1 & 0\\1 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}3\\3\end{bmatrix} = [\mathtt{f2}](\mathbf{x})$$

$$[\mathtt{2t}](\mathbf{x}) = \begin{bmatrix}1 & 0\\1 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}3\\3\end{bmatrix} \quad \gtrsim \quad \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}3\\0\end{bmatrix} = [\mathtt{t2}](\mathbf{x})$$

$$[\mathtt{0}\triangleleft](\mathbf{x}) = \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}2\\0\end{bmatrix} \quad > \quad \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}1\\0\end{bmatrix} = [\mathtt{t}\triangleleft](\mathbf{x})$$

$$[\mathtt{1}\triangleleft](\mathbf{x}) = \begin{bmatrix}1 & 0\\1 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}2\\2\end{bmatrix} \quad \gtrsim \quad \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}2\\2\end{bmatrix} = [\mathtt{ff}\triangleleft](\mathbf{x})$$

$$[\mathtt{2}\triangleleft](\mathbf{x}) = \begin{bmatrix}1 & 0\\1 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}2\\2\end{bmatrix} \quad \gtrsim \quad \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\mathbf{x} + \begin{bmatrix}2\\0\end{bmatrix} = [\mathtt{tf}\triangleleft](\mathbf{x})$$

By Lemmas 2.8 and 3.15, $\mathcal{X}^{\mathrm{rev}}$ is terminating. As a result, $\mathcal{W}'^{\mathrm{rev}}$ is terminating, which by Lemma 2.8 implies that $\mathcal{W}'$ is terminating. □

4.2 Farkas' Variant

Let $2\mathbb{N}+1 = \{1, 3, 5, \dots\}$ denote the odd natural numbers. Farkas [Far05] studied a slight modification $F'\colon 2\mathbb{N}+1 \to 2\mathbb{N}+1$ of the Collatz function for which it becomes possible to prove convergence via induction. We consider automatically proving the convergence of this function as another test case for the mixed-base scheme that is easier than the Collatz conjecture without being entirely trivial. Farkas defined this function as

$$F'(n) = \begin{cases} \frac{n}{3} & \text{if } n \equiv 0 \pmod 3 \\ \frac{n+1}{2} & \text{if } n \not\equiv 0 \pmod 3 \text{ and } n \equiv 1 \pmod 4 \\ \frac{3n+1}{2} & \text{if } n \not\equiv 0 \pmod 3 \text{ and } n \equiv 3 \pmod 4. \end{cases}$$

In this paper, we define another function $F\colon \mathbb{N} \to \mathbb{N}$ that resembles the Collatz function more closely than Farkas' F' (with respect to the definitions of the cases) while being equivalent to F' in terms of convergence. It is obtained by introducing an additional case in the Collatz function for $n \equiv 1 \pmod 3$ and applying T otherwise:

$$F(n) = \begin{cases} \frac{n-1}{3} & \text{if } n \equiv 1 \pmod 3 \\ \frac{n}{2} & \text{if } n \equiv 0 \text{ or } n \equiv 2 \pmod 6 \\ \frac{3n+1}{2} & \text{if } n \equiv 3 \text{ or } n \equiv 5 \pmod 6 \end{cases}$$

Given an F'-trajectory, the bijection $n \mapsto (n-1)/2$ applied to each of its iterates maps the trajectory to the corresponding F-trajectory. Thus, an F'-trajectory reaches 1 if and only if the corresponding F-trajectory reaches 0. Furthermore, an F-trajectory reaches 0 if and only if it reaches 1, which is a convenient fact since in our string representations we do not represent 0, so in order to prove that F is convergent it suffices to show that for all $n \in \mathbb{N}^+$ the trajectory $F^{\mathbb{N}}(n)$ contains 1.

Note that F can also be viewed as a generalized Collatz function with 5 cases, which makes it straightforward to translate the cases of F into a set $\mathcal{D}_F$ of dynamic rules (shown below) to obtain a rewriting system that simulates the iterated application of F on positive integers.

$$\begin{aligned} \texttt{1}\triangleright &\to \triangleright \\ \texttt{0f}\triangleright &\to \texttt{0}\triangleright \\ \texttt{1f}\triangleright &\to \texttt{1}\triangleright \\ \texttt{1t}\triangleright &\to \texttt{12}\triangleright \\ \texttt{2t}\triangleright &\to \texttt{22}\triangleright \end{aligned}$$

Termination of the rewriting system $\mathcal{F} = \mathcal{D}_F \cup \mathcal{X}$ is equivalent to the convergence of F. The proof of the equivalence is essentially the same as that of Theorem 3.17 except for the step where we construct a nonterminating rewrite sequence from a nonconvergent trajectory. To construct such a rewrite sequence for $\mathcal{F}$, we take any nonconvergent F-trajectory and write the first number in the trajectory in *ternary* (instead of binary). If the leading digit turns out to be 2_3 we replace it by $1_0 0_2$ since the leading digit is always assumed to be 1_0 in our mixed-base representations. Then repeatedly performing the rightmost possible rewrite results in a rewrite sequence that simulates the F-trajectory.

Farkas gave an inductive proof of convergence for F' via case analysis. We found an automated proof that $\mathcal{F}$ is terminating via arctic matrix interpretations. Below, we present both a manual proof of convergence for F and the automated proof of $\mathrm{SN}(\mathcal{F})$. It is worth mentioning that the default configurations of the existing termination tools (e.g., AProVE, Matchbox) are too conservative to prove the termination of this system, but after their authors tweaked the strategies they were also able to find automated proofs via arctic matrix interpretations. It also remains an interesting question whether the automated proof somehow captures the argument in the inductive proof.

Theorem 4.2. *For all $n \in \mathbb{N}^+$, the trajectory $F^{\mathbb{N}}(n)$ contains* 1.

Proof (in the manner of Farkas [Far05, Theorem 1]). We proceed by induction. For $n = 1$ the result holds since $F^{\mathbb{N}}(1) \ = (1, 0, 0, \dots)$. Assuming it holds for all positive integers less than n, we will show that it holds for n. In particular, we will show that the trajectory of n reaches a number strictly smaller than n, which implies by the induction hypothesis that the trajectory contains 1. We split into three cases.

(i) Assume $n \equiv 1 \pmod 3$. Then $F(n) = (n-1)/3$ is smaller than n.

(ii) Assume $n \equiv 0$ or $n \equiv 2 \pmod 6$. Then $F(n) = n/2$ is smaller than n.

(iii) Assume $n \equiv 3$ or $n \equiv 5 \pmod 6$. Denote n by N_1, and consider the trajectory $F^{\mathbb{N}}(N_1) = (N_1, N_2, N_3, \dots)$. Let

$$k = \sup \left\{ i \in \mathbb{N}^+ \;\middle|\; N_j = \frac{3N_{j-1}+1}{2} \text{ for all } 1 < j \leq i \right\}.$$

We cannot have $k = \infty$, i.e., F cannot apply $N \mapsto (3N+1)/2$ indefinitely, since for such a sequence we have

$$N_j + 1 = \left(\frac{3}{2}\right)(N_{j-1}+1) = \left(\frac{3}{2}\right)^{j-1}(N_1+1).$$

As all the elements in the trajectory have to be integers, $k-1$ cannot exceed the number of times that $N_1 + 1$ can be divided by 2, so k is finite. Consider

$$N_{k+1} = \left(\frac{3}{2}\right)^k (N_1+1) - 1.$$

It is of the form $3^k \cdot L - 1$ for some positive integer L, so we know that either $N_{k+1} \equiv 2 \pmod 6$ or $N_{k+1} \equiv 5 \pmod 6$. Due to the way k is defined, N_{k+1} cannot be congruent to $5 \pmod 6$, so we have $N_{k+1} \equiv 2 \pmod 6$. In particular, N_{k+1} is even, so $\left(\frac{3}{2}\right)^k (N_1+1)$ is odd, which in turn implies that $\frac{N_1+1}{2^k}$ is odd, i.e., we have $N_1 = 2^k \cdot (2M+1) - 1$ for some natural number M. Then, N_{k+1} satisfies

$$N_{k+1} = 3^k \cdot (2M+1) - 1 = 3^k \cdot 2M + 3^k - 1.$$

Since $N_{k+1} \equiv 2 \pmod 6$, we can deduce that

$$F(N_{k+1}) = \frac{N_{k+1}}{2} = \frac{3^k \cdot 2M + 3^k - 1}{2} \equiv 1 \pmod 3,$$

and furthermore,

$$F^2(N_{k+1}) = \frac{\frac{3^k \cdot 2M + 3^k - 1}{2} - 1}{3} = 3^{k-1} \cdot 2M + 3^{k-1} - 1 \equiv 2 \pmod 6.$$

Repeatedly applying F in the above manner gives $F^{2k}(N_{k+1}) = 2M$. Since $k \geq 1$, we have $2M < 2^k \cdot (2M+1) - 1 = N_1 = n$, and so the trajectory $F^{\mathbb{N}}(n)$ reaches a number strictly smaller than n. □

Proof (via arctic matrix interpretations). We will show $\mathrm{SN}(\mathcal{F})$. By Lemmas 2.8 and 3.15, we have $\mathrm{SN}(\mathcal{X}^{\text{rev}})$. Consider the arctic matrix interpretations below. Recall that $-$ indicates $-\infty$ in the matrices. Also recall that arctic addition is max and arctic multiplication is $+$.

$$[\mathtt{f}](\mathbf{x}) = \begin{bmatrix} - & - & - & 2 & - \\ - & 2 & 0 & - & - \\ 2 & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \end{bmatrix} \qquad [\mathtt{t}](\mathbf{x}) = \begin{bmatrix} - & - & - & - & 2 \\ 0 & 2 & 0 & - & 0 \\ 2 & - & 2 & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \end{bmatrix}$$

$$[\triangleleft](\mathbf{x}) = \begin{bmatrix} 0 \\ 2 \\ - \\ - \\ 4 \end{bmatrix} \qquad [\triangleright](\mathbf{x}) = \begin{bmatrix} 0 & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \tag{9}$$

$$[\mathtt{0}](\mathbf{x}) = \begin{bmatrix} 0 & 4 & 0 & - & - \\ - & 4 & - & - & - \\ - & 4 & 0 & - & - \\ 0 & 3 & 0 & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \qquad [\mathtt{1}](\mathbf{x}) = \begin{bmatrix} 1 & - & - & - & - \\ - & 4 & 0 & - & - \\ - & 4 & 0 & - & - \\ 0 & - & - & - & - \\ 0 & 3 & 0 & - & - \end{bmatrix} \mathbf{x} \qquad [\mathtt{2}](\mathbf{x}) = \begin{bmatrix} 0 & - & 0 & - & - \\ - & 4 & - & - & - \\ 0 & - & 1 & - & 0 \\ - & - & - & - & - \\ 0 & - & 0 & - & 0 \end{bmatrix} \mathbf{x}$$

Above interpretations prove $\mathrm{SN}(\mathcal{D}_{F_{\text{top}}^{\text{rev}}} \,/\, \mathcal{X}^{\text{rev}})$ by Theorem 2.15. (See Appendix B for the corresponding interpretations of the rules of $\mathcal{F}^{\text{rev}}$.) We then get $\mathrm{SN}(\mathcal{D}_{F^{\text{rev}}} / \mathcal{X}^{\text{rev}})$ by Lemma 3.18. As we know that $\mathcal{X}^{\text{rev}}$ is terminating, by Theorem 2.6 we can conclude $\mathrm{SN}(\mathcal{D}_{F^{\text{rev}}} \cup \mathcal{X}^{\text{rev}})$, which implies $\mathrm{SN}(\mathcal{F})$ via Lemma 2.8. □

4.3 Subsets of the Collatz System

It is also interesting to consider whether we can automatically prove termination of proper subsets of $\mathcal{T}$. Specifically, we considered the 11 subsystems obtained by leaving out a single rewriting rule from $\mathcal{T}$, and we found termination proofs via matrix interpretations for all of the 11 subproblems.

The reason for our interest in these problems is threefold:

1. Termination of $\mathcal{T}$ implies termination of all of its subsystems, so proving its termination is at least as difficult a task as proving termination of the 11 subsystems. Therefore, the subproblems serve as additional sanity checks that an automated approach aspiring to succeed for the Collatz conjecture ought to be able to pass. Furthermore, they form an additional source of weakened variants that give us a heuristic idea of the suitable choice of parameters to use when searching for interpretations for the full problem.

2. When proving termination in a stepwise manner, we solve a sequence of relative termination problems. Having proved termination of all 11 subsystems is a partial solution to the full problem, since it implies that for any single rule $\ell \to r \in \mathcal{T}$, proving $\mathrm{SN}(\{\ell \to r\} \,/\, \mathcal{T})$ settles the Collatz conjecture.

3. After the removal of a rule, the termination of the remaining system still encodes a valid mathematical question about the Collatz trajectories. The question of the termination of a proper subset is equivalent to asking if every corresponding Collatz trajectory that does not require the use of the left-out rule is convergent. For instance, removing $\{\mathtt{t}\triangleright \to \mathtt{2}\triangleright\}$ deletes the odd case of the Collatz function, which gives a clearly convergent function. Removing the auxiliary rules in $\mathcal{X}$ lead to potentially more interesting questions: For instance, the termination of $\mathcal{T} \setminus \{\mathtt{t1} \to \mathtt{2f}\}$ implies that the trajectories where the pattern $\mathtt{t1}$ is never encountered during the simulation are all convergent.

   Note that this type of inquiry is less meaningful for the unary system $\mathcal{Z}$ as its simulation of the Collatz function depends crucially on each rule being present. Leaving out some subset of the rules from $\mathcal{Z}$ causes the system to become terminating for a trivial reason, i.e., the computation it expresses no longer corresponds to the iterated application of some function.

Example 4.3. As an instance of leaving out a rule, consider the subsystem $\mathcal{T} \setminus \{\mathtt{f1} \to \mathtt{0t}\}$. There is a single-step natural matrix interpretations proof that this system is terminating:

$$[\mathtt{f}](\mathbf{x}) = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \mathbf{x} \qquad [\mathtt{t}](\mathbf{x}) = \begin{bmatrix} 1 & 3 \\ 3 & 4 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$[\triangleleft](\mathbf{x}) = \begin{bmatrix} 1 & 5 \\ 0 & 0 \end{bmatrix} \mathbf{x} \qquad [\triangleright](\mathbf{x}) = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$[\mathtt{0}](\mathbf{x}) = \begin{bmatrix} 7 & 2 \\ 2 & 5 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} \qquad [\mathtt{1}](\mathbf{x}) = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad [\mathtt{2}](\mathbf{x}) = \begin{bmatrix} 2 & 2 \\ 2 & 4 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

It can be checked that under these interpretations the rules of $\mathcal{T} \setminus \{\mathtt{f1} \to \mathtt{0t}\}$ satisfy the following relations:

$$[\mathtt{f}\triangleright](\mathbf{x}) = \begin{bmatrix} 2 & 0 \\ 1 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad > \quad \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = [\triangleright](\mathbf{x})$$

$$[\mathtt{t}\triangleright](\mathbf{x}) = \begin{bmatrix} 4 & 0 \\ 7 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 5 \\ 8 \end{bmatrix} \quad > \quad \begin{bmatrix} 4 & 0 \\ 6 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 4 \\ 8 \end{bmatrix} = [\mathtt{2}\triangleright](\mathbf{x})$$

$$[\mathtt{f0}](\mathbf{x}) = \begin{bmatrix} 9 & 7 \\ 7 & 2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 3 \\ 2 \end{bmatrix} \quad > \quad \begin{bmatrix} 9 & 7 \\ 7 & 2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} = [\mathtt{0f}](\mathbf{x})$$

$$[\mathtt{f2}](\mathbf{x}) = \begin{bmatrix} 4 & 6 \\ 2 & 2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad > \quad \begin{bmatrix} 3 & 2 \\ 2 & 1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} = [\mathtt{1f}](\mathbf{x})$$

$$[\texttt{t0}](\mathbf{x}) = \begin{bmatrix} 13 & 17 \\ 29 & 26 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 6 \\ 11 \end{bmatrix} \quad > \quad \begin{bmatrix} 5 & 10 \\ 4 & 7 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 4 \\ 2 \end{bmatrix} = [\texttt{1t}](\mathbf{x})$$

$$[\texttt{t1}](\mathbf{x}) = \begin{bmatrix} 5 & 4 \\ 10 & 7 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 2 \\ 4 \end{bmatrix} \quad > \quad \begin{bmatrix} 4 & 2 \\ 6 & 2 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} = [\texttt{2f}](\mathbf{x})$$

$$[\texttt{t2}](\mathbf{x}) = \begin{bmatrix} 8 & 14 \\ 14 & 22 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 7 \\ 9 \end{bmatrix} \quad > \quad \begin{bmatrix} 8 & 14 \\ 14 & 22 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 4 \\ 8 \end{bmatrix} = [\texttt{2t}](\mathbf{x})$$

$$[\triangleleft\texttt{0}](\mathbf{x}) = \begin{bmatrix} 17 & 27 \\ 0 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 7 \\ 0 \end{bmatrix} \quad > \quad \begin{bmatrix} 16 & 23 \\ 0 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 6 \\ 0 \end{bmatrix} = [\triangleleft\texttt{t}](\mathbf{x})$$

$$[\triangleleft\texttt{1}](\mathbf{x}) = \begin{bmatrix} 7 & 6 \\ 0 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad > \quad \begin{bmatrix} 7 & 6 \\ 0 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = [\triangleleft\texttt{ff}](\mathbf{x})$$

$$[\triangleleft\texttt{2}](\mathbf{x}) = \begin{bmatrix} 12 & 22 \\ 0 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 10 \\ 0 \end{bmatrix} \quad > \quad \begin{bmatrix} 9 & 22 \\ 0 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 7 \\ 0 \end{bmatrix} = [\triangleleft\texttt{ft}](\mathbf{x})$$

Note that the left-out rule $\texttt{f1} \to \texttt{0t}$ ends up receiving a nondecreasing interpretation as there is no constraint to enforce otherwise:

$$[\texttt{f1}](\mathbf{x}) = \begin{bmatrix} 3 & 2 \\ 2 & 1 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \not> \quad \begin{bmatrix} 13 & 29 \\ 17 & 26 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 11 \\ 8 \end{bmatrix} = [\texttt{0t}](\mathbf{x})$$

The above interpretations witness for instance that the Collatz trajectory starting at 3 (represented as $\triangleleft\texttt{t}\triangleright$) is convergent, because the missing rule is not used in any derivation of 1 ($\triangleleft\triangleright$) from 3. Below is an example derivation along with the values each string represents and a vector value of each string under the interpretations above (setting $\mathbf{x} = (0, 0)$ for the purpose of demonstration). We omit the subscripts from the rewrite relations and simply write →.

$$\begin{array}{ccccccccccccccccc}
3 & & 5 & & 5 & & 8 & & 8 & & 8 & & 4 & & 2 & & 1 \\
\triangleleft\underline{\texttt{t}\triangleright} & \to & \underline{\triangleleft\texttt{2}}\triangleright & \to & \triangleleft\texttt{f}\underline{\texttt{t}\triangleright} & \to & \triangleleft\underline{\texttt{f2}}\triangleright & \to & \underline{\triangleleft\texttt{1}}\texttt{f}\triangleright & \to & \triangleleft\texttt{ff}\underline{\texttt{f}\triangleright} & \to & \triangleleft\texttt{f}\underline{\texttt{f}\triangleright} & \to & \triangleleft\underline{\texttt{f}\triangleright} & \to & \triangleleft\triangleright \\
\begin{bmatrix} 79 \\ 0 \end{bmatrix} & > & \begin{bmatrix} 78 \\ 0 \end{bmatrix} & > & \begin{bmatrix} 68 \\ 0 \end{bmatrix} & > & \begin{bmatrix} 62 \\ 0 \end{bmatrix} & > & \begin{bmatrix} 41 \\ 0 \end{bmatrix} & > & \begin{bmatrix} 40 \\ 0 \end{bmatrix} & > & \begin{bmatrix} 26 \\ 0 \end{bmatrix} & > & \begin{bmatrix} 14 \\ 0 \end{bmatrix} & > & \begin{bmatrix} 12 \\ 0 \end{bmatrix}
\end{array}$$

Table 1 shows the parameters of the matrix interpretations proofs that we found for the termination of each subsystem. For each rule $\ell \to r$ that is left out, we searched for a stepwise proof to show that $\mathcal{B} \setminus \{\ell \to r\}$ is terminating relative to $\mathcal{T} \setminus \{\ell \to r\}$ (freely utilizing weakly monotone algebras due to Lemma 3.18). Such a proof requires at most three steps since there are at most three rules in $\mathcal{B} \setminus \{\ell \to r\}$. In the table, we report the smallest parameters (in terms of matrix dimension) that work for all of these steps. As we already know that $\mathrm{SN}(\mathcal{T} \setminus \mathcal{B})$ holds (by Lemma 3.15), the interpretations found allow us to conclude the termination of each subsystem. This is not the only way to prove termination of the subsystems; however, we chose this uniform strategy for the sake of comparison. In the experiments we searched for matrices of up to 7 dimensions, with the coefficients taking at most 8 different values.

Table 1: Smallest proofs found for termination of subsystems of $\mathcal{T}$ in under 30 seconds. The columns show the matrix dimension D and the maximum number V of distinct coefficients that appear in the matrices, along with the median time to find an entire termination proof across 25 repetitions for the fixed D and V.

	Natural			Arctic		
Rule removed	D	V	Time	D	V	Time
$\mathtt{f}\triangleright \to \triangleright$	3	4	1.42s	3	5	15.95s
$\mathtt{t}\triangleright \to \mathtt{2}\triangleright$	1	2	0.27s	1	3	0.28s
$\mathtt{f0} \to \mathtt{0f}$	4	2	0.92s	3	4	2.46s
$\mathtt{f1} \to \mathtt{0t}$	1	3	0.50s	1	4	0.51s
$\mathtt{f2} \to \mathtt{1f}$	1	2	0.38s	1	3	0.39s
$\mathtt{t0} \to \mathtt{1t}$	4	3	1.20s	3	4	0.87s
$\mathtt{t1} \to \mathtt{2f}$	5	2	0.89s	4	3	0.84s
$\mathtt{t2} \to \mathtt{2t}$	4	4	10.00s	2	5	0.62s
$\triangleleft\mathtt{0} \to \triangleleft\mathtt{t}$	2	2	0.40s	2	3	0.42s
$\triangleleft\mathtt{1} \to \triangleleft\mathtt{ff}$	3	3	0.53s	3	4	0.57s
$\triangleleft\mathtt{2} \to \triangleleft\mathtt{ft}$	4	4	7.51s	4	3	4.04s

4.4 Odd Trajectories

In the originally defined Collatz function C, applying $2n+1 \mapsto 6n+4$ produces an even number, so we incorporate a single division by 2 into the definition of the odd case and obtain the function T with the same overall dynamics as C. Taking this idea further by performing as many divisions by 2 as possible leads to the so-called Syracuse function $\mathrm{Syr}\colon 2\mathbb{N}+1 \to 2\mathbb{N}+1$, defined as

$$\mathrm{Syr}(n) = \frac{3n+1}{2^k}, \text{ where } k = \max\{k \in \mathbb{N}^+ \mid 2^k \text{ divides } 3n+1\}.$$

The Syracuse function maps each odd number to the next odd number in its Collatz trajectory. In this sense, it has the same overall dynamics as the Collatz functions C or T we have previously defined, i.e., Syr is convergent if and only if C or T is convergent. The function T takes fewer steps than C to reach 1 and can be thought of as an accelerated Collatz function, making the Syracuse function a further accelerated version. Reducing the number of steps towards convergence is of interest mainly because the existence of various kinds of termination proofs is dependent on the derivational complexity[10] of the rewriting system staying below certain thresholds [HL89]. For instance, given a rewriting system, if its termination can be proved in a single-step via natural matrix interpretations then its derivational complexity is at most exponential [EWZ08]. Similarly, a single-step termination proof via arctic matrix interpretations implies a linear derivational complexity [KW09]. Thus, it can be preferable to work with an alternative rewriting system with smaller derivational complexity that still simulates the dynamics of the Collatz function.

Expressing the Syracuse function as a generalized Collatz function would require infinitely

[10] Derivational complexity is defined as the maximal length of rewrite sequences in the system as a function of the length of the initial string in the sequence.

many cases to account for all of the possible appearances of 2^k as the denominator with different values of k. As a result, we are unable to simulate it with a finite rewriting system. Nevertheless, we may compromise and accelerate the Collatz function by a smaller amount. We first observe that if $n \equiv 1 \pmod 8$ then $\mathrm{Syr}(n) = \frac{3n+1}{4}$ and if $n \equiv 3 \pmod 4$ then $\mathrm{Syr}(n) = \frac{3n+1}{2}$. Furthermore, for any $n \in \mathbb{N}$ we have $\mathrm{Syr}(8n+5) = \mathrm{Syr}(2n+1)$ since $3(8n+5)+1 = 24n+16 = 4(6n+4) = 4(3(2n+1)+1)$. Putting these observations together, we can define a generalized Collatz function $S\colon 2\mathbb{N}+1 \to 2\mathbb{N}+1$ as follows.

$$S(n) = \begin{cases} \frac{3n+1}{4} & \text{if } n \equiv 1 \pmod 8 \\ \frac{n-1}{4} & \text{if } n \equiv 5 \pmod 8 \\ \frac{3n+1}{2} & \text{if } n \equiv 3 \pmod 4 \end{cases}$$

Example 4.4. The three trajectories below illustrate how the dynamics of C, Syr, and S compare to one another.

- $C^{\mathbb{N}}(19) = (19, 58, 29, 88, 44, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, \dots)$
- $\mathrm{Syr}^{\mathbb{N}}(19) = (19, 29, 11, 17, 13, 5, 1, 1, \dots)$
- $S^{\mathbb{N}}(19) = (19, 29, \mathbf{7}, 11, 17, 13, \mathbf{3}, 5, 1, 1, \dots)$

Note that a Syr-trajectory simply consists of the odd numbers in the corresponding C-trajectory; however, an S-trajectory is not necessarily contained in the corresponding C-trajectory. This is due to the $n \equiv 5 \pmod 8$ case of S that maps n to a number not necessarily contained in the C-trajectory (marked in bold above). Once one of the other two cases is encountered, the S-trajectory arrives back at the number $\mathrm{Syr}(n)$.

S is convergent if and only if C (or T) is convergent, and the number of steps that S takes to converge is between that of T and Syr. In a manner similar to before, we can translate the cases of S into a set $\mathcal{D}_S$ of dynamic rules (shown below). Since we are working with odd numbers we use a new symbol $\bullet$ to mark the end of a string, viewed functionally as $\bullet(x) = 2x + 1$.

$$\begin{aligned} \mathtt{ff}\bullet &\to \mathtt{0}\bullet \\ \mathtt{tf}\bullet &\to \bullet \\ \mathtt{t}\bullet &\to \mathtt{2}\bullet \end{aligned}$$

Termination of the rewriting system $\mathcal{S} = \mathcal{D}_S \cup \mathcal{X}$ is equivalent to the convergence of S. Similar to $\mathcal{T}$, proving the termination of $\mathcal{S}$ is currently beyond our reach, although it may potentially be an easier path to the Collatz conjecture (compared to proving $\mathrm{SN}(\mathcal{T})$). Failing to prove the termination of $\mathcal{S}$ itself, we considered the subsystems of $\mathcal{S}$ as we did for $\mathcal{T}$ in Section 4.3. With matrix interpretations, termination of all but two of the 11-rule subsystems of $\mathcal{S}$ was automatically proved. Despite devoting thousands of CPU hours, we were not able to find interpretations to prove that $\mathcal{S}_1 = \mathcal{S} \setminus \{\mathtt{ff}\bullet \to \mathtt{0}\bullet\}$ or $\mathcal{S}_2 = \mathcal{S} \setminus \{\mathtt{tf}\bullet \to \bullet\}$ is terminating. By the discussion in Section 4.3, termination of subsystems of $\mathcal{S}$ corresponds to questions about the trajectories of S. In particular, for $\mathcal{S}_1$ and $\mathcal{S}_2$ we have the following.

Proposition 4.5.

1. $\mathrm{SN}(\mathcal{S}_1)$ *if and only if all nonconvergent S-trajectories contain some* $n \equiv 1 \pmod 8$.

2. SN($\mathcal{S}_2$) *if and only if all nonconvergent S-trajectories contain some* $n \equiv 5 \pmod 8$.

We can prove automatically via matrix interpretations that $\mathcal{S}_3 = \mathcal{S} \setminus \{\texttt{t}\bullet \to \texttt{2}\bullet\}$ is terminating, which implies the following through an equivalence of the above form for $\mathcal{S}_3$.

Proposition 4.6. *All nonconvergent S-trajectories contain some* $n \equiv 3 \pmod 4$.

Leaving out some dynamic rules from a mixed-base rewriting system that simulates a generalized Collatz function allows us to phrase the above kinds of questions as termination problems. Recall that, although S is equivalent to C in terms of convergence, its trajectories are not necessarily contained in the Collatz trajectories, so we cannot substitute C for S in the above statements (except for the second item of Proposition 4.5, since the left-out rule is the one that maps to a number not contained in the Collatz trajectory). In the next section, we perform the same kind of inquiry into the original Collatz trajectories.

4.5 Collatz Trajectories Modulo 8

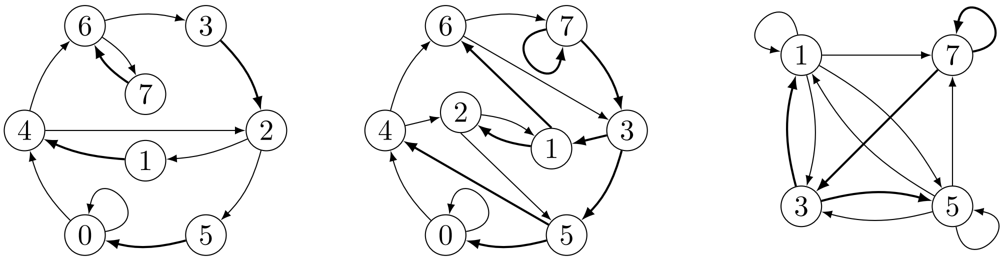


Figure 1: Transition graphs of the iterates in the Collatz trajectories across residue classes modulo 8 for the functions C (left), T (middle), S (right). For each function f, the edge $u \to v$ is part of its transition graph if and only if there exists some $n \equiv u \pmod 8$ such that $f(n) \equiv v \pmod 8$. Bold edges indicate transitions where $f(n) > n$.

Our initial motivation for the previous section was to find rewriting systems with reduced derivational complexity. This inquiry led us to surprisingly difficult Collatz-like problems, which we believe may be on the border of the reach of automated methods. In this section we explore more problems of a similar form and present the unsolved ones as potentially reachable targets for automated termination proving.

Let m be a power of 2. Given $k \in \{0, 1, \ldots, m-1\}$, is it the case that all nonconvergent Collatz trajectories contain some $n \equiv k \pmod m$? For several values of k this can be proved by inspecting the transitions of the iterates in the Collatz trajectories across residue classes modulo m (shown on Figure 1 for $m = 8$). As an example, consider the case $k = 6$.

Proposition 4.7. *All nonconvergent C-trajectories contain some* $n \equiv 6 \pmod 8$.

Proof. Let t be a C-trajectory. The path that t takes over the transition graph of C in Figure 1 is a concatenation of the simple cycles of the graph. If t does not contain any $n \equiv 6 \pmod 8$, then the cycles containing 6 cannot occur in the path, which leaves eight cycles (listed below). By composing the functions applied at each transition in the graph, we can compute the mapping (call it c) applied to an iterate after traversing a cycle:

Cycle	$c(n)$
$(0,4,2,5,0)$	$(3n+8)/8$
$(0,0)$	$n/2$
$(1,4,2,1)$	$(3n+1)/4$
$(2,5,0,4,2)$	$(3n+2)/8$
$(2,1,4,2)$	$(3n+2)/4$
$(4,2,5,0,4)$	$(3n+4)/8$
$(4,2,1,4)$	$(3n+4)/4$
$(5,0,4,2,5)$	$(3n+1)/8$

For t to be nonconvergent, it needs to contain infinitely many occurrences of cycles that increase the values of the iterates. All of the above cycles decrease the values of the iterates, so each iterate of t eventually reaches a smaller number, implying t is convergent. □

Propositions of the above kind can also be formulated as termination of some rewriting systems. With this approach we found automated proofs for several cases.

We start by studying the smallest interesting case $m = 4$. Consider the following two sets of dynamic rules to simulate the Collatz function.

$$\mathcal{D}_C^{\text{Even}/4} = \left\{ \begin{array}{r} \mathtt{ff}\triangleright \to \mathtt{f}\triangleright \\ \mathtt{tf}\triangleright \to \mathtt{t}\triangleright \\ \mathtt{t}\triangleright \to \mathtt{2f}\triangleright \end{array} \right\} \qquad \mathcal{D}_T^{\text{Odd}/4} = \left\{ \begin{array}{r} \mathtt{f}\triangleright \to \triangleright \\ \mathtt{ft}\triangleright \to \mathtt{1f}\triangleright \\ \mathtt{tt}\triangleright \to \mathtt{2t}\triangleright \end{array} \right\}$$

It is straightforward to see that the SRS $\mathcal{C}^{\text{Even}/4} = \mathcal{D}_C^{\text{Even}/4} \cup \mathcal{X}$ simulates the function C and the SRS $\mathcal{T}^{\text{Odd}/4} = \mathcal{D}_T^{\text{Odd}/4} \cup \mathcal{X}$ simulates the function T. The first two rules of $\mathcal{D}_C^{\text{Even}/4}$ express the application of C to numbers of the form $4n$ and $4n+2$. Similarly, the last two rules of $\mathcal{D}_T^{\text{Odd}/4}$ express the application of T to numbers of the form $4n+1$ and $4n+3$. Observing that a T-trajectory contains the same odd numbers as the corresponding C-trajectory, we have the following equivalences.

Proposition 4.8.

1. $\text{SN}(\mathcal{C}^{\text{Even}/4} \setminus \{\mathtt{ff}\triangleright \to \mathtt{f}\triangleright\})$ *if and only if all nonconvergent C-trajectories contain some* $n \equiv 0 \pmod 4$.

2. $\text{SN}(\mathcal{C}^{\text{Even}/4} \setminus \{\mathtt{tf}\triangleright \to \mathtt{t}\triangleright\})$ *if and only if all nonconvergent C-trajectories contain some* $n \equiv 2 \pmod 4$.

3. $\text{SN}(\mathcal{T}^{\text{Odd}/4} \setminus \{\mathtt{ft}\triangleright \to \mathtt{1f}\triangleright\})$ *if and only if all nonconvergent C-trajectories contain some* $n \equiv 1 \pmod 4$.

4. $\text{SN}(\mathcal{T}^{\text{Odd}/4} \setminus \{\mathtt{tt}\triangleright \to \mathtt{2t}\triangleright\})$ *if and only if all nonconvergent C-trajectories contain some* $n \equiv 3 \pmod 4$.

We were able to prove termination of the above four systems via matrix interpretations, thus establishing the following.

Theorem 4.9. *If there exists a nonconvergent Collatz trajectory, it cannot avoid any of the residue classes modulo* 4*.*

Next, we look at the case $m = 8$. It can be studied via the analogously defined rewriting systems:

$$\mathcal{D}_C^{\mathrm{Even}/8} = \left\{\begin{array}{r} \mathtt{fff}\triangleright \to \mathtt{ff}\triangleright \\ \mathtt{ftf}\triangleright \to \mathtt{ft}\triangleright \\ \mathtt{tff}\triangleright \to \mathtt{tf}\triangleright \\ \mathtt{ttf}\triangleright \to \mathtt{tt}\triangleright \\ \mathtt{t}\triangleright \to \mathtt{2f}\triangleright \end{array}\right\} \qquad \mathcal{D}_T^{\mathrm{Odd}/8} = \left\{\begin{array}{r} \mathtt{f}\triangleright \to \triangleright \\ \mathtt{fft}\triangleright \to \mathtt{0tf}\triangleright \\ \mathtt{ftt}\triangleright \to \mathtt{1ft}\triangleright \\ \mathtt{tft}\triangleright \to \mathtt{2ff}\triangleright \\ \mathtt{ttt}\triangleright \to \mathtt{2tt}\triangleright \end{array}\right\}$$

The SRS $\mathcal{C}^{\mathrm{Even}/8} = \mathcal{D}_C^{\mathrm{Even}/8} \cup \mathcal{X}$ simulates the function C and the SRS $\mathcal{T}^{\mathrm{Odd}/8} = \mathcal{D}_T^{\mathrm{Odd}/8} \cup \mathcal{X}$ simulates the function T. We do not formally state the analogue of Proposition 4.8 — terminations of the eight subsystems obtained from $\mathcal{C}^{\mathrm{Even}/8}$ or $\mathcal{T}^{\mathrm{Odd}/8}$ are each equivalent to the question of whether all nonconvergent Collatz trajectories have to encounter the corresponding residue class modulo 8. Through automated proofs, we were able to establish the following.

Theorem 4.10. *If there exists a nonconvergent Collatz trajectory, it cannot avoid the residue classes of* 2, 3, 4, 6 *modulo* 8*.*

Theorems 4.9 and 4.10 are also not difficult to prove by hand; however, it remains open whether an analogue of Theorem 4.10 holds for the residue classes of 0, 1, 5, 7 modulo 8, that is, we have neither manual nor automated proofs for these cases. They may serve well as easier[11] Collatz-like open problems to test our abilities against, so we offer a $500 reward for a resolution of any of the following conjectures.[12]

Conjecture 4.11. *All nonconvergent C-trajectories contain some $n \equiv 1 \pmod 8$.*

Conjecture 4.12. *All nonconvergent C-trajectories contain some $n \equiv 5 \pmod 8$.*

Conjecture 4.13. *All nonconvergent C-trajectories contain some $n \equiv 7 \pmod 8$.*

Even the following easier statement remains unproved, although we were able to prove (both manually and automatically) its versions with $\{1, 5\}$ and $\{1, 7\}$ as the residue classes.

Conjecture 4.14. *All nonconvergent C-trajectories contain either some $n \equiv 5 \pmod 8$ or some $n \equiv 7 \pmod 8$.*

We can alternatively formulate each of the above conjectures as the convergence problem of some partial generalized Collatz function. For instance, Conjecture 4.14 is equivalent to the convergence of the function $H \colon \mathbb{N}_\perp^+ \to \mathbb{N}_\perp^+$ defined as follows.

$$H(n) = \begin{cases} \frac{3n}{4} & \text{if } n \equiv 0 \pmod 4 \\ \frac{9n+1}{8} & \text{if } n \equiv 7 \pmod 8 \\ \perp & \text{otherwise} \end{cases} \tag{10}$$

[11] See Denend [Den18, Section 4] for a heuristic prediction of the relative easiness of those cases. Denend uses an empirical measure based on the number of odd elements occurring in a trajectory until either a certain residue class or 1 is encountered, which resembles the *ones-ratio* and the *stopping time ratio* defined by Applegate and Lagarias [AL03].

[12] The case 0 is omitted as its answer follows from the case 5 by Figure 1.

Proposition 4.15. *H is convergent if and only if all nonconvergent C-trajectories contain either some $n \equiv 5 \pmod 8$ or some $n \equiv 7 \pmod 8$.*

Proof. Recall from Figure 1 the transition graph of the iterates in C-trajectories across residue classes modulo 8. Observe that if a nonconvergent trajectory avoids the residue classes of 5 and $7 \pmod 8$, then it has to visit the residue class of $1 \pmod 8$, so we may assume without loss of generality that each nonconvergent trajectory starts with some $n \equiv 1 \pmod 8$.

Then a nonconvergent trajectory that avoids the residue classes of 5 and $7 \pmod 8$ can traverse only the cycle of classes $(1, 4, 2, 1)$ or $(1, 4, 6, 3, 2, 1)$. The former, which is traversed only if $n \equiv 1 \pmod{32}$, effects the map $n \mapsto (3n + 1)/4$. The latter, which is traversed only if $n \equiv 57 \pmod{64}$, effects the map $n \mapsto (9n + 7)/8$. If n belongs to any other residue class, either the trajectory is convergent, or it contains some $k \equiv 5 \pmod 8$ or some $k \equiv 7 \pmod 8$. As a result, the function $G\colon \mathbb{N}_\perp^+ \setminus \{1\} \to \mathbb{N}_\perp^+ \setminus \{1\}$ defined as

$$G(n) = \begin{cases} \frac{3n+1}{4} & \text{if } n \equiv 1 \pmod{32} \\ \frac{9n+7}{8} & \text{if } n \equiv 57 \pmod{64} \\ \perp & \text{otherwise} \end{cases}$$

is nonconvergent if and only if there exists a nonconvergent C-trajectory that avoids the residue classes of 5 and $7 \pmod 8$. Moreover, H as defined in (10) is convergent if and only if G is convergent, because the bijection $n \mapsto (n - 1)/8$ maps a G-trajectory to an H-trajectory. □

4.6 SAT Solving Considerations

Phase-saving heuristic. Most top-tier termination tools, such as AProVE, Matchbox, and TTT2, use the SAT solver MiniSat [ES04] to search for matrix interpretations. This choice is somewhat surprising as MiniSat has not been updated since 2008 and the performance of SAT solvers has improved significantly in the last decade. The use of MiniSat in these provers is motivated by its observed effectiveness in finding interpretations. We investigated the reason for this, which turned out to be a heuristic that MiniSat disables in its default configuration.

Most SAT solvers sort variables by the frequency of their occurrences in recent conflict clauses, selecting the highest ranked one as the variable to branch on [MMZ+01]. This procedure is similar in MiniSat and modern SAT solvers. However, once a variable is picked, the solvers differ in the truth value assignment that they explore first. MiniSat uses negative branching [ES04], which explores the “false” branch first for all decision variables. The empirical effectiveness of this heuristic likely stems from its interaction with the common choices of propositional encodings used when translating problems into SAT instances. Modern SAT solvers use phase-saving [PD07], which first explores the branch corresponding to the truth value to which the variable was forced most recently during unit propagation. We offer the following explanation for the effectiveness of negative branching.

We use the *order encoding* [TTKB09, PJ11] when translating the integer constraints from Section 2.3 into CNF formulas. In this encoding, each variable X that takes values in a finite domain $A = \{0, 1, \ldots, n + 1\}$ is represented by a corresponding list of $n + 1$ Boolean variables $(x_0, x_1, \ldots, x_n)$. Each Boolean variable x_i indicates $X > i$. As an example, $X = 2$ has the corresponding assignment $(1, 1, 0, 0, \ldots, 0)$. This encoding gives a natural way of representing inequality constraints. For instance, the constraint $a < X \leq b$ can be enforced simply by setting $x_a = 1$ and $x_b = 0$. As another example, for variables X, Y the inequality $X \leq Y$ can be expressed by the conjunction $\bigwedge_{i=0}^{n}(\neg x_i \vee y_i)$. Consequently, when using the order encoding, representations that correspond to smaller numbers require setting a larger fraction of the Boolean variables to false. For instance, the representation of zero or minus infinity assigns all variables to false. Moreover, the matrices and the vectors that occur in the interpretations are often sparse and contain relatively few large numbers. As a result, negative branching possibly directs the search towards a satisfying assignment more often.

Additionally, in contrast to phase-saving, negative branching tends to guide the solver to a potentially different part of the search space after a restart (which is performed frequently in SAT solvers). The increased diversity of the search space (across restarts) that is explored due to negative branching is also possibly helpful for the kinds of the problems discussed in this paper.

One can enable negative branching in most SAT solvers, which in our case improves solver performance for some of the formulas that encode the existence of interpretations.

Heavy-tailed behavior of SAT solvers. Combinatorial search algorithms can have a tendency to show a large variance in runtime across different, randomly selected initial conditions of the search. For instance, Gomes et al. [GSCK00] observed that DPLL solvers [DP60, DLL62] had a substantial fraction of very short runs when dealing with relatively hard instances. This suggests that trying to solve such instances using several parallel runs is a reasonable strategy, since some of the runs will have a nonnegligible chance of finishing early.

Modern SAT solvers incorporate heuristics that alleviate the heavy-tailed behavior to some extent; however, the variability in runtime (especially on satisfiable instances) is not completely eliminated. In our termination prover, we run multiple instances of the SAT solver (specifically, CaDiCaL [Bie19]) in parallel on the formula that encodes the existence of an interpretation. We introduce additional randomness to the procedure by running each parallel instance with a different shuffling of the clauses in the formula. Ideally, solver performance would be invariant to the ordering of the clauses; however, the clause order indirectly affects the variable selection heuristic, which ends up influencing runtime.

Experiments with solver configurations. Table 2 shows the difference in runtime for both MiniSat and CaDiCaL due to negative branching and parallelism with a few small-scale experiments. From all the automated proofs in this paper, we selected the steps where the SAT solving phase took longer than 2 seconds, and reran the prover for these specific steps with different configurations of those SAT solvers. When running only a single instance of a solver, we used the nonshuffled formula. In the parallel case, we ran 8 solver instances with one of the instances receiving the nonshuffled formula and the other 7 receiving random shuffles. Running multiple instances of a solver in parallel is a considerable improvement in all cases where the allotted time is enough to find an interpretation. Negative branching has an adverse effect in some cases, although in two of the most difficult instances ($\mathcal{F}$ and $\mathcal{C}^{\mathrm{Even}/8} \setminus \{\mathtt{ttf}\triangleright \to \mathtt{tt}\triangleright\}$) it makes the difference between finding an interpretation relatively quickly and timing out. It also appears that, at least for the more difficult cases, replacing MiniSat by CaDiCaL can be beneficial. These experiments are not extensive enough to derive broad conclusions; however, they suggest that running multiple instances of a modern SAT solver with negative branching enabled in some and disabled in others can be a reasonably diverse strategy when searching for matrix interpretations.

Table 2: Difference in runtime due to negative branching and parallelism. Each row corresponds to a single step of a relative termination proof occurring in the previous sections of this paper. The leftmost columns show the original system being proved as terminating, along with the parameters of the interpretations being searched for (D for dimension, V for the number of different values each coefficient may take) that remove at least a single rule from the system. For each solver, each of the remaining columns shows the median time the experiment takes over 25 repetitions. Each run times out after 120 seconds, and runs that time out are counted as having taken double the allowed time.

					Phase-saving		Negative branching	
Problem	Interp.	D	V	Solver	Single	Parallel	Single	Parallel
$\mathcal{F}$	Arctic	5	8	MiniSat	240.00s	240.00s	85.60s	**5.47s**
$\mathcal{T} \setminus \{\texttt{f}\triangleright \to \triangleright\}$	Arctic	3	5		0.62s	**0.09s**	67.69s	15.74s
$\mathcal{T} \setminus \{\texttt{f0} \to \texttt{0f}\}$	Arctic	3	4		3.18s	0.91s	2.19s	0.91s
$\mathcal{T} \setminus \{\texttt{t2} \to \texttt{2t}\}$	Natural	4	4		240.00s	63.71s	240.00s	87.85s
$\mathcal{T} \setminus \{\triangleleft\texttt{2} \to \triangleleft\texttt{ft}\}$	Natural	4	4		240.00s	240.00s	69.73s	5.25s
$\mathcal{T} \setminus \{\triangleleft\texttt{2} \to \triangleleft\texttt{ft}\}$	Arctic	4	3		1.39s	**0.18s**	7.65s	1.71s
$\mathcal{C}^{\text{Even}/8} \setminus \{\texttt{ttf}\triangleright \to \texttt{tt}\triangleright\}$	Natural	3	11		240.00s	240.00s	240.00s	240.00s
$\mathcal{T}^{\text{Odd}/8} \setminus \{\texttt{ftt}\triangleright \to \texttt{1ft}\triangleright\}$	Arctic	3	12		2.35s	0.36s	117.37s	0.88s
$\mathcal{F}$	Arctic	5	8	CaDiCaL	240.00s	240.00s	44.45s	9.22s
$\mathcal{T} \setminus \{\texttt{f}\triangleright \to \triangleright\}$	Arctic	3	5		1.52s	0.13s	29.95s	13.12s
$\mathcal{T} \setminus \{\texttt{f0} \to \texttt{0f}\}$	Arctic	3	4		3.75s	**0.83s**	3.27s	1.71s
$\mathcal{T} \setminus \{\texttt{t2} \to \texttt{2t}\}$	Natural	4	4		75.78s	19.12s	29.62s	**8.13s**
$\mathcal{T} \setminus \{\triangleleft\texttt{2} \to \triangleleft\texttt{ft}\}$	Natural	4	4		75.05s	**5.22s**	24.31s	6.43s
$\mathcal{T} \setminus \{\triangleleft\texttt{2} \to \triangleleft\texttt{ft}\}$	Arctic	4	3		3.33s	0.52s	11.55s	3.84s
$\mathcal{C}^{\text{Even}/8} \setminus \{\texttt{ttf}\triangleright \to \texttt{tt}\triangleright\}$	Natural	3	11		240.00s	240.00s	240.00s	**79.05s**
$\mathcal{T}^{\text{Odd}/8} \setminus \{\texttt{ftt}\triangleright \to \texttt{1ft}\triangleright\}$	Arctic	3	12		1.94s	**0.33s**	3.28s	0.38s

5 Related Work

There are several previous studies of the Collatz conjecture through alternative models of computation.

String rewriting systems. To our knowledge, Zantema [Zan05], with his system $\mathcal{Z}$, which we saw in Section 3.1, was the first to attempt using an automated method and string rewriting to search for a proof of the Collatz conjecture. In addition, although we independently discovered the mixed binary–ternary system described in Section 3.2, Scollo [Sco05] had essentially the same idea, the distinction being that he adopted a functional view of the digits that is different than in (8). Scollo was not concerned with proving termination, though, and proposed rewriting primarily as a formalism that forgoes the arithmetic interpretation of the iterates and instead emphasizes the dynamic/computational behavior.

Tag systems. An m-tag system [Pos43] is a computational model described by a set of production rules that map symbols to strings (or "tags"). Given an initial string $X = x_0x_1 \dots x_n$, the tag system finds the rule $x_0 \rightharpoonup Z$ whose LHS matches the leftmost symbol of X, appends the tag Z to X, and deletes the first m symbols of XZ; resulting in a new string $X' = x_mx_{m+1} \dots x_nZ$. The tag system repeats this transformation until reaching a string of length less than m, at which point it halts.

De Mol [DM08] showed the existence of a small 2-tag system with the following rules that simulates the iterated application of the Collatz function given a unary representation:

$$\begin{aligned} \mathtt{1} &\rightharpoonup \triangleleft\triangleright \\ \triangleleft &\rightharpoonup \mathtt{1} \\ \triangleright &\rightharpoonup \mathtt{111} \end{aligned}$$

This tag system halts if and only if the Collatz conjecture holds, giving yet another formulation of the problem. De Mol further extended this scheme to allow the simulation of an arbitrary generalized Collatz function with modulus d by a d-tag system.

Cellular automata. Kari [Kar12] designed 1D cellular automata that perform multiplication by 3 and 3/2 in base 6, and reformulated both the Collatz conjecture and Mahler's 3/2 problem as sets of constraints to be satisfied by the space-time diagrams of these cellular automata.

String arithmetic. Kauffman [Kau95] developed a formalism to perform arithmetic, which he called *string arithmetic*, and expressed the Collatz conjecture within it. This formalism works with unary representations of numbers, and uses the three symbols $\mathtt{1}$, $\triangleleft$, $\triangleright$. Letting ϵ denote the empty string and $\mathtt{N}$ be any string representing a number, string arithmetic consists of the following bidirectional rewrite rules (or "identities") to convert between different strings representing the same number:

$$\begin{aligned} \triangleright\triangleleft &\longleftrightarrow \epsilon \\ \mathtt{11} &\longleftrightarrow \triangleleft\mathtt{1}\triangleright \\ \mathtt{1N} &\longleftrightarrow \mathtt{N1} \end{aligned}$$

Along with the above identities, Kauffman's encoding of the Collatz function uses the following two rules:

$$\begin{aligned} \triangleleft\mathtt{N}\triangleright &\longrightarrow \mathtt{N} \\ \triangleleft\mathtt{N}\triangleright\mathtt{1} &\longrightarrow \triangleleft\mathtt{N1}\triangleright\mathtt{N} \end{aligned}$$

The Collatz conjecture becomes the question of whether for strings of $\mathtt{1}$s of all lengths there exists a rewrite sequence using the five rules above to reach the string $\mathtt{1}$.

6 Conclusions

We presented an approach to prove convergence of instances of generalized Collatz functions by translating them into string rewriting systems and applying SAT solving to find matrix interpretations that show termination. Important components of our approach include the quality of the translation into rewriting, the use of weakly monotone Σ-algebras, and dedicated heuristics in the SAT solving phase. We considered some interesting, simpler variants of the Collatz conjecture to gauge the feasibility of this approach in proving mathematically interesting statements. We observed that some variants could be solved only via natural matrix interpretations, while others required arctic matrix interpretations.

Several extensions to this work can further our understanding of the potential of rewriting techniques for answering mathematical questions. For instance, it is of interest to study the efficacy of different termination proving techniques on the problems that we considered. In particular, we found matrix interpretations to be the most successful for our purposes despite the existence of newer techniques developed for automatically proving termination of a few select challenging instances. Although matrix interpretations lead to automated proofs of several weakened variants discussed in this paper, it might still be the case that there exist no such interpretations to establish the termination of the Collatz system $\mathcal{T}$ (as we have shown in the case of natural matrix interpretations for Zantema's system $\mathcal{Z}$). This would be an interesting result in itself. Another issue is the matter of representation; specifically, it is worth exploring whether there exists a suitable translation of the Collatz conjecture into the termination of a term, instead of string, rewriting system since many automated termination proving techniques are generalized to term rewriting. Finally, injecting problem-specific knowledge into the rewriting systems or the termination techniques would be helpful as there exists a wealth of information about the Collatz conjecture that could simplify the search for a termination proof.

Acknowledgments

We thank Johannes Waldmann for insightful discussions regarding arctic matrix interpretations, for pointing us to [Kar12] and the rewriting system in Example 3.3, for responding to our challenge to solve Farkas' variant with Matchbox, and for feedback on an early draft. We thank Carsten Fuhs and Jürgen Giesl for responding to our challenge to solve Farkas' variant with AProVE. We additionally thank Carsten Fuhs for his thorough explanations of the dependency pair framework and AProVE's strategies. We thank Florian Frohn for responding to the challenge to solve the subsystems from Section 4.3 with AProVE. We thank Jeffrey Lagarias for discussions regarding the problems in Section 4.5. We thank Luke Schaeffer and Chris Lynch for discussions on alternative rewriting systems that simulate the Collatz map. We thank Amazon Web Services for computing resources. We thank Jeremy Avigad and Jasmin Blanchette for their detailed comments on an early draft. Finally, we thank the reviewers of CADE for their comments on the preliminary version of this paper and the reviewers of the Journal of Automated Reasoning for their comments on the journal version.

This material is based upon work supported by the National Science Foundation under grant CCF-2006363.

References

[AG00] Thomas Arts and Jürgen Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236(1):133–178, 2000.

[AL03] David Applegate and Jeffrey C. Lagarias. Lower bounds for the total stopping time of $3x + 1$ iterates. *Mathematics of Computation*, 72(242):1035–1049, 2003.

[Bar21] David Barina. Convergence verification of the Collatz problem. *The Journal of Supercomputing*, 77(3):2681–2688, 2021.

[Bie19] Armin Biere. CaDiCaL at the SAT Race 2019. In *Proceedings of the SAT Race 2019: Solver and Benchmark Descriptions*, number B-2019-1 in Department of Computer Science Series of Publications B, pages 8–9. University of Helsinki, 2019.

[BM90] Robert N. Buttsworth and Keith R. Matthews. On some Markov matrices arising from the generalized Collatz mapping. *Acta Arithmetica*, 55(1):43–57, 1990.

[BN98] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[BO93] Ronald V. Book and Friedrich Otto. *String-Rewriting Systems*. Springer, 1993.

[BTW87] Per Bak, Chao Tang, and Kurt Wiesenfeld. Self-organized criticality: An explanation of the $1/f$ noise. *Physical Review Letters*, 59(4):381–384, 1987.

[Can69] Georg Cantor. Über einfache zahlensysteme. *Zeitschrift für Mathematik und Physik*, 14:121–128, 1869.

[CGP10] Pierre Courtieu, Gladys Gbedo, and Olivier Pons. Improved matrix interpretation. In *Proceedings of the 36th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, number 5901 in Lecture Notes in Computer Science, pages 283–295. Springer, 2010.

[Con72] John H. Conway. Unpredictable iterations. In *Proceedings of the Number Theory Conference (University of Colorado, Boulder, Colorado)*, pages 49–52, 1972.

[Den18] Matthew Denend. Challenging variants of the Collatz conjecture. Master's thesis, University of Texas at Austin, 2018.

[DLL62] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.

[DM08] Liesbeth De Mol. Tag systems and Collatz-like functions. *Theoretical Computer Science*, 390(1):92–101, 2008.

[DP60] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201–215, 1960.

[Erd79] Paul Erdős. Some unconventional problems in number theory. *Mathematics Magazine*, 52(2):67–70, 1979.

[ES04] Niklas Eén and Niklas Sörensson. An extensible SAT-solver. In *Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, number 2919 in Lecture Notes in Computer Science, pages 502–518. Springer, 2004.

[EWZ08] Jörg Endrullis, Johannes Waldmann, and Hans Zantema. Matrix interpretations for proving termination of term rewriting. *Journal of Automated Reasoning*, 40(2):195–220, 2008.

[Far05] Hershel M. Farkas. Variants of the $3N+1$ conjecture and multiplicative semigroups. In *Geometry, Spectral Theory, Groups, and Dynamics*, number 387 in Contemporary Mathematics, pages 121–127. American Mathematical Society, 2005.

[GAB$^+$17] Jürgen Giesl, Cornelius Aschermann, Marc Brockschmidt, Fabian Emmes, Florian Frohn, Carsten Fuhs, Jera Hensel, Carsten Otto, Martin Plücker, Peter Schneider-Kamp, Thomas Ströder, Stephanie Swiderski, and René Thiemann. Analyzing program termination and complexity automatically with AProVE. *Journal of Automated Reasoning*, 58(1):3–31, 2017.

[GSCK00] Carla P. Gomes, Bart Selman, Nuno Crato, and Henry Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of Automated Reasoning*, 24(1):67–100, 2000.

[GTSF06] Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, and Stephan Falke. Mechanizing and improving dependency pairs. *Journal of Automated Reasoning*, 37(3):155–203, 2006.

[HL89] Dieter Hofbauer and Clemens Lautemann. Termination proofs and the length of derivations. In *Proceedings of the 3rd International Conference on Rewriting Techniques and Applications (RTA)*, number 355 in Lecture Notes in Computer Science, pages 167–177. Springer, 1989.

[HW06] Dieter Hofbauer and Johannes Waldmann. Termination of string rewriting with matrix interpretations. In *Proceedings of the 17th International Conference on Rewriting Techniques and Applications (RTA)*, number 4098 in Lecture Notes in Computer Science, pages 328–342. Springer, 2006.

[Kar12] Jarkko Kari. Cellular automata, the Collatz conjecture and powers of 3/2. In *Proceedings of the 16th International Conference on Developments in Language Theory (DLT)*, number 7410 in Lecture Notes in Computer Science, pages 40–49. Springer, 2012.

[Kaš92] František Kaščák. Small universal one-state linear operator algorithm. In *Proceedings of the 17th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, number 629 in Lecture Notes in Computer Science, pages 327–335. Springer, 1992.

[Kau95] Louis H. Kauffman. Arithmetic in the form. *Cybernetics and Systems*, 26(1):1–57, 1995.

[Koh07] Stefan Kohl. Wildness of iteration of certain residue-class-wise affine mappings. *Advances in Applied Mathematics*, 39(3):322–328, 2007.

[KS07] Stuart A. Kurtz and Janos Simon. The undecidability of the generalized Collatz problem. In *Proceedings of the 4th International Conference on Theory and Applications of Models of Computation (TAMC)*, number 4484 in Lecture Notes in Computer Science, pages 542–553. Springer, 2007.

[KSZM09] Martin Korp, Christian Sternagel, Harald Zankl, and Aart Middeldorp. Tyrolean Termination Tool 2. In *Proceedings of the 20th International Conference on Rewriting Techniques and Applications (RTA)*, number 5595 in Lecture Notes in Computer Science, pages 295–304. Springer, 2009.

[KW09] Adam Koprowski and Johannes Waldmann. Max/plus tree automata for termination of term rewriting. *Acta Cybernetica*, 19(2):357–392, 2009.

[Lag85] Jeffrey C. Lagarias. The $3x + 1$ problem and its generalizations. *The American Mathematical Monthly*, 92(1):3–23, 1985.

[Lag10] Jeffrey C. Lagarias. *The Ultimate Challenge: The* $3x + 1$ *Problem.* American Mathematical Society, 2010.

[Lag11] Jeffrey C. Lagarias. The $3x + 1$ problem: An annotated bibliography (1963–1999), 2011. arXiv:math/0309224.

[Lag12] Jeffrey C. Lagarias. The $3x+1$ problem: An annotated bibliography, II (2000–2009), 2012. arXiv:math/0608208.

[Mah68] Kurt Mahler. An unsolved problem on the powers of 3/2. *Journal of the Australian Mathematical Society*, 8(2):313–321, 1968.

[Mic15] Pascal Michel. Problems in number theory from busy beaver competition. *Logical Methods in Computer Science*, 11(4:10), 2015.

[MMZ$^+$01] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In *Proceedings of the 38th Design Automation Conference (DAC)*, pages 530–535. Association for Computing Machinery, 2001.

[MS99] João P. Marques-Silva and Karem A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Transactions on Computers*, 48(5):506–521, 1999.

[NM11] Friedrich Neurauter and Aart Middeldorp. Revisiting matrix interpretations for proving termination of term rewriting. In *Proceedings of the 22nd International Conference on Rewriting Techniques and Applications (RTA)*, number 10 in Leibniz International Proceedings in Informatics, pages 251–266. Schloss Dagstuhl, 2011.

[PD07] Knot Pipatsrisawat and Adnan Darwiche. A lightweight component caching scheme for satisfiability solvers. In *Proceedings of the 10th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, number 4501 in Lecture Notes in Computer Science, pages 294–299. Springer, 2007.

[PJ11] Justyna Petke and Peter Jeavons. The order encoding: From tractable CSP to tractable SAT. In *Proceedings of the 14th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, number 6695 in Lecture Notes in Computer Science, pages 371–372. Springer, 2011.

[Pos43] Emil L. Post. Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics*, 65(2):197–215, 1943.

[Rad62] Tibor Radó. On non-computable functions. *The Bell System Technical Journal*, 41(3):877–884, 1962.

[Raw85] Daniel A. Rawsthorne. Imitation of an iteration. *Mathematics Magazine*, 58(3):172–176, 1985.

[RB10] Christophe Reutenauer and Jean Berstel. *Noncommutative Rational Series with Applications*. Cambridge University Press, 2010.

[Sco05] Giuseppe Scollo. ω-rewriting the Collatz problem. *Fundamenta Informaticae*, 64(1–4):405–416, 2005.

[Soi76] Matti Soittola. Positive rational sequences. *Theoretical Computer Science*, 2(3):317–322, 1976.

[ST14] Christian Sternagel and René Thiemann. Formalizing monotone algebras for certification of termination and complexity proofs. In *Proceedings of the 25th International Conference on Rewriting Techniques and Applications (RTA)*, number 8560 in Lecture Notes in Computer Science, pages 441–455. Springer, 2014.

[SZ17] David Sabel and Hans Zantema. Termination of cycle rewriting by transformation and matrix interpretation. *Logical Methods in Computer Science*, 13(1:11), 2017.

[Tao20] Terence Tao. Almost all orbits of the Collatz map attain almost bounded values, 2020. arXiv:1909.03562.

[TTKB09] Naoyuki Tamura, Akiko Taga, Satoshi Kitagawa, and Mutsunori Banbara. Compiling finite linear CSP into SAT. *Constraints*, 14(2):254–272, 2009.

[Wag85] Stan Wagon. The Collatz problem. *The Mathematical Intelligencer*, 7(1):72–76, 1985.

[Wal04] Johannes Waldmann. Matchbox: A tool for match-bounded string rewriting. In *Proceedings of the 15th International Conference on Rewriting Techniques and Applications (RTA)*, number 3091 in Lecture Notes in Computer Science, pages 85–94. Springer, 2004.

[WdE10] Johannes Waldmann, Roel de Vrijer, and Jörg Endrullis. Local termination: Theory and practice. *Logical Methods in Computer Science*, 6(3:20), 2010.

[Zan94] Hans Zantema. Termination of term rewriting: Interpretation and type elimination. *Journal of Symbolic Computation*, 17(1):23–50, 1994.

[Zan05] Hans Zantema. Termination of string rewriting proved automatically. *Journal of Automated Reasoning*, 34(2):105–139, 2005.

A Alternative Proof of Lemma 3.16

Let us recall the statement of the result before proceeding with the proof.

Lemma 3.16. *If $\mathcal{T}$ is terminating on all initial strings of the* canonical form $\triangleleft(\texttt{f}|\texttt{t}|\texttt{0}|\texttt{1}|\texttt{2})^*\triangleright$*, then $\mathcal{T}$ is terminating (on all initial strings).*

Proof. Assume there exists some string $X \in \{\texttt{f}, \texttt{t}, \texttt{0}, \texttt{1}, \texttt{2}, \triangleleft, \triangleright\}^*$ that admits an infinite rewrite sequence for $\mathcal{T}$. View X as split into blocks delimited by $\triangleleft$ or $\triangleright$, i.e., for some $k \in \mathbb{N}^+$, write

$$X = N_1 d_1 N_2 d_2 \dots d_{k-1} N_k,$$

where $d_i \in \{\triangleleft, \triangleright\}$ for each $1 \le i < k$, and $N_i \in \{\texttt{f}, \texttt{t}, \texttt{0}, \texttt{1}, \texttt{2}\}^*$ for each $1 \le i \le k$. Any string containing X also admits an infinite rewrite sequence, so instead consider $Z = d_0 X d_k$, written as

$$Z = d_0 N_1 d_1 N_2 d_2 \dots d_{k-1} N_k d_k,$$

where $d_0, d_k \in \{\triangleleft, \triangleright\}$.

As $\mathcal{T}$ is not terminating on Z, there exists some rewrite rule applicable to it. Consider an arbitrary block of Z along with its neighbors (to which some rewrite rule of $\mathcal{T}$ applies), written as

$$V = PcQdR,$$

where $c, d \in \{\triangleleft, \triangleright\}$ and $P, Q, R \in \{\texttt{f}, \texttt{t}, \texttt{0}, \texttt{1}, \texttt{2}\}^*$. Due to the shapes of the rules of $\mathcal{T}$, the string V can be rewritten only into one of the following:

$$\begin{aligned} V_1 &= \widetilde{P}cQdR && \text{such that} \quad Pc \to_{\mathcal{T}} \widetilde{P}c \\ V_2 &= Pc\widetilde{Q}dR && \text{such that} \quad cQd \to_{\mathcal{T}} c\widetilde{Q}d \\ V_3 &= PcQd\widetilde{R} && \text{such that} \quad dR \to_{\mathcal{T}} d\widetilde{R} \end{aligned}$$

In any case, the delimiters are unchanged, and only a single block is affected by the rewrite. This means that any rewrite sequence that starts from Z consists of several sequences that each operate entirely on some block of Z. Thus, since Z has finitely many blocks, there exists some block that admits an infinite rewrite sequence. In particular, there exists a string cNd, where $c, d \in \{\triangleleft, \triangleright\}$ and $N \in \{\texttt{f}, \texttt{t}, \texttt{0}, \texttt{1}, \texttt{2}\}^*$, that can be rewritten indefinitely. We claim that this requires $c = \triangleleft$ and $d = \triangleright$, so the string cNd is of the canonical form $\triangleleft(\texttt{f}|\texttt{t}|\texttt{0}|\texttt{1}|\texttt{2})^*\triangleright$. As shown below, the other cases are all terminating.

(i) $\triangleright N \triangleright$: This string does not contain $\triangleleft$, so it can be rewritten using only the rules in $\mathcal{T} \setminus \mathcal{B}$, but we know from Lemma 3.15 that $\mathrm{SN}(\mathcal{T} \setminus \mathcal{B})$, so there can be no infinite rewrite sequence.

(ii) $\triangleleft N \triangleleft$: This string does not contain $\triangleright$, so it can be rewritten using only the rules in $\mathcal{T} \setminus \mathcal{D}_T$, but we know from Lemma 3.15 that $\mathrm{SN}(\mathcal{T} \setminus \mathcal{D}_T)$, so there can be no infinite rewrite sequence.

(iii) $\triangleright N \triangleleft$: The SRS $\mathcal{T}$ contains no rules of the form $s\triangleleft \to t\triangleleft$ or $\triangleright s \to \triangleright t$ for any s, t, so this string can be rewritten using only the rules in $\mathcal{T} \setminus (\mathcal{B} \cup \mathcal{D}_T)$. Since each subset of a terminating SRS is terminating, by Lemma 3.15 we conclude $\mathrm{SN}(\mathcal{T} \setminus (\mathcal{B} \cup \mathcal{D}_T))$, so there can be no infinite rewrite sequence.

This proves the contrapositive of the lemma statement. □

B Remaining Part of the Automated Proof for Farkas' Variant

With the interpretations from (9), the rules of $\mathcal{F}^{\mathrm{rev}}$ satisfy the following relations:

$$[\triangleright\mathtt{1}](\mathbf{x}) = \begin{bmatrix} 1 & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} - \\ - \\ - \\ - \\ - \end{bmatrix} \gg \begin{bmatrix} 0 & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} - \\ - \\ - \\ - \\ - \end{bmatrix} = [\triangleright](\mathbf{x})$$

$$[\triangleright\mathtt{f0}](\mathbf{x}) = \begin{bmatrix} 2 & 5 & 2 & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \end{bmatrix} \gg \begin{bmatrix} 0 & 4 & 0 & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} - \\ - \\ - \\ - \\ - \end{bmatrix} = [\triangleright\mathtt{0}](\mathbf{x})$$

$$[\triangleright\mathtt{f1}](\mathbf{x}) = \begin{bmatrix} 2 & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \end{bmatrix} \gg \begin{bmatrix} 1 & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} - \\ - \\ - \\ - \\ - \end{bmatrix} = [\triangleright\mathtt{1}](\mathbf{x})$$

$$[\triangleright\mathtt{t1}](\mathbf{x}) = \begin{bmatrix} 2 & 5 & 2 & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \end{bmatrix} \gg \begin{bmatrix} 1 & 4 & 0 & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} - \\ - \\ - \\ - \\ - \end{bmatrix} = [\triangleright\mathtt{21}](\mathbf{x})$$

$$[\triangleright\mathtt{t2}](\mathbf{x}) = \begin{bmatrix} 2 & - & 2 & - & 2 \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \end{bmatrix} \gg \begin{bmatrix} 0 & - & 1 & - & 0 \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} - \\ - \\ - \\ - \\ - \end{bmatrix} = [\triangleright\mathtt{22}](\mathbf{x})$$

$$[\mathtt{0f}](\mathbf{x}) = \begin{bmatrix} 2 & 6 & 4 & 2 & - \\ - & 6 & 4 & - & - \\ 2 & 6 & 4 & - & - \\ 2 & 5 & 3 & 2 & - \\ - & - & - & - & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ 0 \\ - \end{bmatrix} \geq \begin{bmatrix} 2 & 5 & 2 & - & - \\ - & 6 & 0 & - & - \\ 2 & 6 & 2 & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \end{bmatrix} = [\mathtt{f0}](\mathbf{x})$$

$$[\mathtt{1f}](\mathbf{x}) = \begin{bmatrix} - & - & - & 3 & - \\ 2 & 6 & 4 & - & - \\ 2 & 6 & 4 & - & - \\ - & - & - & 2 & - \\ 2 & 5 & 3 & 2 & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} 1 \\ - \\ - \\ 0 \\ 0 \end{bmatrix} \geq \begin{bmatrix} - & - & - & - & - \\ 0 & 6 & 0 & - & - \\ 2 & 6 & 2 & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix}\mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \end{bmatrix} = [\mathtt{t0}](\mathbf{x})$$

$$[\texttt{2f}](\mathbf{x}) = \begin{bmatrix} 2 & - & - & 2 & - \\ - & 6 & 4 & - & - \\ 3 & - & - & 2 & - \\ - & - & - & - & - \\ 2 & - & - & 2 & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ 0 \\ - \\ 0 \end{bmatrix} \quad \geq \quad \begin{bmatrix} 2 & - & - & - & - \\ - & 6 & 2 & - & - \\ 3 & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \end{bmatrix} = [\texttt{f1}](\mathbf{x})$$

$$[\texttt{0t}](\mathbf{x}) = \begin{bmatrix} 4 & 6 & 4 & - & 4 \\ 4 & 6 & 4 & - & 4 \\ 4 & 6 & 4 & - & 4 \\ 3 & 5 & 3 & - & 3 \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ 0 \\ - \end{bmatrix} \quad \geq \quad \begin{bmatrix} 2 & 5 & 2 & - & - \\ 1 & 6 & 2 & - & - \\ 3 & 6 & 2 & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \end{bmatrix} = [\texttt{t1}](\mathbf{x})$$

$$[\texttt{1t}](\mathbf{x}) = \begin{bmatrix} - & - & - & - & 3 \\ 4 & 6 & 4 & - & 4 \\ 4 & 6 & 4 & - & 4 \\ - & - & - & - & 2 \\ 3 & 5 & 3 & - & 3 \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 1 \\ - \\ - \\ 0 \\ 0 \end{bmatrix} \quad \geq \quad \begin{bmatrix} - & - & - & - & - \\ 0 & 6 & 1 & - & 0 \\ 2 & - & 2 & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \end{bmatrix} = [\texttt{f2}](\mathbf{x})$$

$$[\texttt{2t}](\mathbf{x}) = \begin{bmatrix} 2 & - & 2 & - & 2 \\ 4 & 6 & 4 & - & 4 \\ 3 & - & 3 & - & 2 \\ - & - & - & - & - \\ 2 & - & 2 & - & 2 \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ 0 \\ - \\ 0 \end{bmatrix} \quad \geq \quad \begin{bmatrix} 2 & - & 2 & - & 2 \\ 0 & 6 & 1 & - & 0 \\ 2 & - & 3 & - & 2 \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 0 \\ - \\ - \\ - \\ - \end{bmatrix} = [\texttt{t2}](\mathbf{x})$$

$$[\texttt{0}\triangleleft](\mathbf{x}) = \begin{bmatrix} - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 6 \\ 6 \\ 6 \\ 5 \\ - \end{bmatrix} \quad \geq \quad \begin{bmatrix} - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 6 \\ 4 \\ 2 \\ - \\ - \end{bmatrix} = [\texttt{t}\triangleleft](\mathbf{x})$$

$$[\texttt{1}\triangleleft](\mathbf{x}) = \begin{bmatrix} - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 1 \\ 6 \\ 6 \\ 0 \\ 5 \end{bmatrix} \quad \geq \quad \begin{bmatrix} - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 0 \\ 6 \\ 2 \\ - \\ - \end{bmatrix} = [\texttt{ff}\triangleleft](\mathbf{x})$$

$$[\texttt{2}\triangleleft](\mathbf{x}) = \begin{bmatrix} - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 0 \\ 6 \\ 4 \\ - \\ 4 \end{bmatrix} \quad \geq \quad \begin{bmatrix} - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 0 \\ 6 \\ 4 \\ - \\ - \end{bmatrix} = [\texttt{tf}\triangleleft](\mathbf{x})$$

C More Problems to Approach via Rewriting

We adapt the mixed base scheme for several other open problems and reformulate them as termination of relatively small rewriting systems.

C.1 Mahler's 3/2 Problem

Definition C.1. Let $\xi \in \mathbb{R}_{>0}$ be a positive real number. It is called a *Z-number* if for all $k \in \mathbb{N}$ we have $\operatorname{frac}\left(\xi \left(\frac{3}{2}\right)^k\right) < \frac{1}{2}$, where $\operatorname{frac}(\cdot)$ denotes the fractional part of the number.

Mahler [Mah68] conjectured that there are no Z-numbers. Moreover, he considered a generalized Collatz function $M\colon \mathbb{N}^+ \to \mathbb{N}^+$, defined as follows.

$$M(n) = \begin{cases} \frac{3n}{2} & \text{if } n \equiv 0 \pmod 2 \\ \frac{3n+1}{2} & \text{if } n \equiv 1 \pmod 2 \end{cases}$$

He related the behaviors of M-trajectories to the existence of Z-numbers:

Theorem C.2. *For $n \in \mathbb{N}^+$, if a Z-number exists in the interval $[n, n+1)$, then there is no $k \in \mathbb{N}$ for which $M^k(n) \equiv 3 \pmod 4$.*

In order to formulate this as a termination problem, we split the odd case of M into two and leave one undefined:

$$M'(n) = \begin{cases} \frac{3n}{2} & \text{if } n \equiv 0 \pmod 2 \\ \frac{3n+1}{2} & \text{if } n \equiv 1 \pmod 4 \\ \bot & \text{if } n \equiv 3 \pmod 4 \end{cases}$$

Given $n \in \mathbb{N}^+$, if the trajectory ${M'}^{\mathbb{N}}(n)$ contains $\bot$ then by the contrapositive of Theorem C.2 there is no Z-number in the interval $[n, n+1)$. When proving nonexistence, we may assume without loss of generality that a Z-number is at least 1, since if $\xi \in (0,1)$ is a Z-number then $\xi \left(\frac{3}{2}\right)^m \in [1, \infty)$ is another one for sufficiently large $m \in \mathbb{N}$. Thus, the nonexistence of Z-numbers can be established by proving that M' is convergent, which is equivalent to the termination of the following rewriting system $\mathcal{M}$. In order to ensure termination at the case $n \equiv 3 \pmod 4$, there is no rule with the LHS $\mathtt{tt}\triangleright$.

$$\begin{array}{lllll}
\mathtt{f}\triangleright \to \mathtt{0}\triangleright & \quad \mathtt{f0} \to \mathtt{0f} & \quad \mathtt{t0} \to \mathtt{1t} & \quad \triangleleft\mathtt{0} \to \triangleleft\mathtt{t} \\
\mathtt{ft}\triangleright \to \mathtt{10}\triangleright & \quad \mathtt{f1} \to \mathtt{0t} & \quad \mathtt{t1} \to \mathtt{2f} & \quad \triangleleft\mathtt{1} \to \triangleleft\mathtt{ff} \\
 & \quad \mathtt{f2} \to \mathtt{1f} & \quad \mathtt{t2} \to \mathtt{2t} & \quad \triangleleft\mathtt{2} \to \triangleleft\mathtt{ft}
\end{array}$$

C.2 Halting Problem for Busy Beavers

The busy beaver problem [Rad62] concerns finding binary-alphabet Turing machines with n states that, when given an input tape of all 0s, write the largest number of 1s on the tape upon halting. For each n, a machine that achieves this is called an *n-state Busy Beaver* (or BB-n for short). Note that this definition only requires the machines to halt on all-0 inputs, leaving the behavior on other inputs unspecified and allowing them not to halt in general.

Michel [Mic15] observed that for $n \in \{2, 3, 4\}$, the busy beaver machines are all *total Turing machines*, i.e., they halt on all inputs, and moreover proved that they all simulate some generalized Collatz function. It is an open problem whether all busy beavers are total Turing machines. In particular, it is unknown whether the current BB-5 candidate is total. Michel showed that the BB-5 candidate simulates the following generalized Collatz function.

$$B(n) = \begin{cases} \frac{5n+18}{3} & \text{if } n \equiv 0 \pmod 3 \\ \frac{5n+22}{3} & \text{if } n \equiv 1 \pmod 3 \\ \perp & \text{if } n \equiv 2 \pmod 3 \end{cases}$$

Convergence of the above function can be studied via the termination of a corresponding rewriting system, although the translation is not as straightforward as the other generalized Collatz functions we have in this paper. Specifically, the mixed base scheme results in a compact rewriting system only when all the mappings $an + b \mapsto cn + d$ applied by the generalized Collatz function at hand satisfy $d < c$. This is not the case for the function B above since it requires us to express $3n \mapsto 5n + 6$ and $3n + 1 \mapsto 5n + 9$. These mappings are expressed more easily by the help of unary representations, so we can have the following hybrid rewriting system $\mathcal{BB}'$ over the alphabet $\{\texttt{i}, \texttt{v}, \texttt{0}, \triangleleft, \triangleright\}$ to simulate the function B, where the new symbol $\texttt{v}$ has the functional view $\texttt{v}(x) = 5x$.

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl}
\texttt{0}\triangleright & \to & \texttt{ivi}\triangleright & \texttt{0iii} & \to & \texttt{i0} & \triangleleft\texttt{ii} & \to & \triangleleft\texttt{0} \\
\texttt{0i}\triangleright & \to & \texttt{iviiii}\triangleright & \texttt{iv} & \to & \texttt{viiiii} & \triangleleft\texttt{v} & \to & \triangleleft\texttt{0ii} \\
 & & & \texttt{0v} & \to & \texttt{v0} & & &
\end{array}$$

We did not succeed in proving the termination of this rewriting system, which is not too surprising since the hybrid scheme does not even pass the test of proving the convergence of W. Alternatively, we can split the cases of B further for the congruence classes modulo 3^k with $k > 1$ until the iterated application of B can either be expressed as $an + b \mapsto cn + d$ with $d < c$, or it reaches $\perp$. After we perform this procedure while ensuring that the number of newly introduced cases are kept to a minimum, we end up with the following mappings to simulate B in an accelerated manner (similar to the idea in Section 4.4).

$$\begin{array}{rcl}
9n & \mapsto & 25n + 16 \\
9n + 1 & \mapsto & 25n + 21 \\
27n + 6 & \mapsto & 125n + 64 \\
27n + 7 & \mapsto & 125n + 71 \\
27n + 16 & \mapsto & 125n + 114 \\
81n + 51 & \mapsto & 625n + 459 \\
243n + 78 & \mapsto & 3125n + 1116 \\
243n + 159 & \mapsto & 3125n + 2159
\end{array} \tag{11}$$

Consider the symbols $\{\texttt{0}, \texttt{1}, \texttt{2}, \texttt{v}, \texttt{w}, \texttt{x}, \texttt{y}, \texttt{z}, \triangleleft, \triangleright\}$ with the following functional views.

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl}
 & & & \texttt{v}(x) & = & 5x & & & \\
\texttt{0}(x) & = & 3x & \texttt{w}(x) & = & 5x + 1 & & & \\
\texttt{1}(x) & = & 3x + 1 & \texttt{x}(x) & = & 5x + 2 & \triangleleft(x) & = & 0 \\
\texttt{2}(x) & = & 3x + 2 & \texttt{y}(x) & = & 5x + 3 & \triangleright(x) & = & x \\
 & & & \texttt{z}(x) & = & 5x + 4 & & &
\end{array}$$

With these symbols, we have the following mixed $\{3, 5\}$-ary (ternary–quinary) system $\mathcal{BB}$ the termination of which is equivalent to the convergence of B. The dynamic rules on the left implement the mappings from (11) and the rest are auxiliary rules that push ternary symbols towards the rightmost end of the string while preserving its value. We were unable to prove the termination of this system, so we include it as yet another challenge for automated termination proving.

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl@{\qquad}rcl@{\qquad}rcl}
\texttt{00}\triangleright & \to & \texttt{yw}\triangleright & \texttt{0v} & \to & \texttt{v0} & \texttt{1v} & \to & \texttt{w2} & \texttt{2v} & \to & \texttt{y1} & \triangleleft\texttt{v} & \to & \triangleleft\texttt{0} \\
\texttt{01}\triangleright & \to & \texttt{zw}\triangleright & \texttt{0w} & \to & \texttt{v1} & \texttt{1w} & \to & \texttt{x0} & \texttt{2w} & \to & \texttt{y2} & \triangleleft\texttt{w} & \to & \triangleleft\texttt{1} \\
\texttt{020}\triangleright & \to & \texttt{xxz}\triangleright & \texttt{0x} & \to & \texttt{v2} & \texttt{1x} & \to & \texttt{x1} & \texttt{2x} & \to & \texttt{z0} & \triangleleft\texttt{x} & \to & \triangleleft\texttt{2} \\
\texttt{021}\triangleright & \to & \texttt{xzw}\triangleright & \texttt{0y} & \to & \texttt{w0} & \texttt{1y} & \to & \texttt{x2} & \texttt{2y} & \to & \texttt{z1} & \triangleleft\texttt{y} & \to & \triangleleft\texttt{10} \\
\texttt{121}\triangleright & \to & \texttt{zxz}\triangleright & \texttt{0z} & \to & \texttt{w1} & \texttt{1z} & \to & \texttt{y0} & \texttt{2z} & \to & \texttt{z2} & \triangleleft\texttt{z} & \to & \triangleleft\texttt{11} \\
\texttt{1220}\triangleright & \to & \texttt{yywz}\triangleright & & & & & & & & & & & & \\
\texttt{02220}\triangleright & \to & \texttt{wyzyw}\triangleright & & & & & & & & & & & & \\
\texttt{12220}\triangleright & \to & \texttt{yxwwz}\triangleright & & & & & & & & & & & &
\end{array}$$

C.3 Ternary Expansions of 2^n

Erdős [Erd79] asked: When does the ternary expansion of 2^n omit the digit 2? This is the case for $2^0 = (1)_3$, $2^2 = (11)_3$, and $2^8 = (100111)_3$. He conjectured that it does not happen for $n > 8$. This conjecture can be reformulated as the statement that the following rewriting system $\mathcal{E}$, where $\overleftarrow{\mathcal{X}} = \{r \to \ell \mid \ell \to r \in \mathcal{X}\}$, is terminating on all initial strings of the form $\triangleleft\texttt{f}^8\texttt{f}^+\triangleright$.

$$\begin{array}{rcl}
\texttt{0}\triangleright & \to & \triangleright \\
\texttt{1}\triangleright & \to & \triangleright \\
\triangleleft\triangleright & \to & \triangleleft\triangleright
\end{array}
\qquad\qquad
\overleftarrow{\mathcal{X}} = \left\{\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl}
\texttt{0f} & \to & \texttt{f0} & \texttt{1t} & \to & \texttt{t0} & \triangleleft\texttt{t} & \to & \triangleleft\texttt{0} \\
\texttt{0t} & \to & \texttt{f1} & \texttt{2f} & \to & \texttt{t1} & \triangleleft\texttt{ff} & \to & \triangleleft\texttt{1} \\
\texttt{1f} & \to & \texttt{f2} & \texttt{2t} & \to & \texttt{t2} & \triangleleft\texttt{ft} & \to & \triangleleft\texttt{2}
\end{array}\right\}$$

Given a string that corresponds to the binary representation of a power of 2, the inverted system $\overleftarrow{\mathcal{X}}$ essentially rewrites the string into ternary by pushing ternary symbols to the right without altering the value that the string represents. The two rules $\{\texttt{0}\triangleright \to \triangleright,\ \texttt{1}\triangleright \to \triangleright\}$ remove the occurrences of the ternary digits `0` and `1` (but not `2`). If the ternary expansion does not contain the digit `2` then all digits will be removed, resulting in the string $\triangleleft\triangleright$ that can then be rewritten to itself indefinitely.

This problem, as described, is an instance of "local termination" [WdE10] since it is concerned with termination on not all possible strings but a subset of them. We have not performed experiments with this system or local termination yet and we leave this for future work.