Dipartimento di Informatica
Università degli Studi di Verona

Rapporto di ricerca
Research report
October 2004

25/2004

ω-rewriting the Collatz problem

Giuseppe Scollo

Questo rapporto è disponibile su Web all'indirizzo:
This report is available on the web at the address:
http://www.di.univr.it/report

Abstract

The Collatz problem is comprised of two distinct, separate questions: existence of eventually periodic Collatz reductions with a nontrivial period, and existence of period-free Collatz reductions. This paper introduces a few distinct, related formalizations of the Collatz dynamics as term rewriting systems, using elementary concepts from the theory of state transition dynamics. Some of the subject systems act on finite terms, whereas others rewrite terms that are endowed with a countable, recursively defined structure. The latter presents a convenient framework for the investigation of extensions of the Collatz dynamics to dense systems.

Keywords: State Transition Dynamics, Attractor, Periodicity, Quasiperiodicity, Infinitary Rewriting

Models of Natural Computation Research Group

1 Introduction

The $3x+1$ problem, also known as the Collatz problem, and under several other names, concerns the behavior of the iterates of the function which takes odd integers n to $3n+1$ and even integers n to $\frac{n}{2}$. The $3x+1$ Conjecture asserts that, starting from any positive integer n, repeated iteration of this function eventually produces the value 1. One immediately sees that, in one step, the function in question takes any odd integer to an even one, so it can be replaced by the function that takes odd integers n to $\frac{3n+1}{2}$. This is the first variant of the Collatz problem that is considered in this paper. A second variant of interest considers the restriction of Collatz iterates to odd positive integers, whereby the subject function is replaced by that which takes every such integer n to the greatest odd divisor of $\frac{3n+1}{2}$ (which may or may not be $\frac{3n+1}{2}$ itself). We refer the interested reader to [4] for a detailed, historical account of the Collatz problem, together with a comprehensive annotated bibliography.[1]

The Collatz problem is comprised of two distinct, separate questions:

1. existence of eventually periodic Collatz reductions with a nontrivial period (if only transitions between odd positive integers are considered, then the trivial period is of length 1, otherwise it is of length 2 and is unique up to circular permutation),

2. existence of Collatz reductions which would not be eventually periodic.

The Conjecture says that both questions have a negative answer. The proof of both conjectures has turned out to be awfully hard to find, in that it is affected by all difficulties which one may expect from the purported chaotic behaviour of Collatz reductions.

The two variants described above will be our starting framework to explore further variants of Collatz-like dynamics, according to the following collection of variations on the theme.

In each case we start from a formalization of the state space (positive integers, odd positive integers, etc.) as a quotient term algebra with a constant, two unary constructors and a relation; the latter turns out to have an interesting connection with a constant-free specification of the Collatz dynamics. This is readily seen by our further formalization of Collatz (or "restricted" Collatz) dynamics by a term rewriting system on a suitable expansion of the previous algebra. In turn, this provides one with a slightly more abstract understanding of Collatz dynamics, that is, by forgetting the

[1] Both are available and kept up-to-date on the Web, at address http://www.cecm.sfu.ca/organics/papers/lagarias

arithmetic interpretation of terms, while taking the dynamic behaviour of the labelled state transition system, and its connection with the computational behaviour of the rewriting system, as the main subjects of investigation.

With labelled transitions, states in the labelled transition system are congruence classes of the (possibly expanded) word algebra, and canonical syntactic representatives for them are readily found. On the other hand, it is possible to get rid of transition labels by encoding the labelling information (the behaviour "history") in states, which thereby take a richer structure—basically that of a pair, where a record of the sequence of Collatz rule applications that have led to the state, adjoins the previously mentioned representative of that state. This record is closely related to what is called the "parity vector" in [4], following [8, 9], and it is actually possible to get such records be *exactly* those parity vectors, even when our system acts on odd positive numbers only, under the arithmetic interpretation.

The previous outline justifies the "rewriting" in the title of this paper; we now briefly introduce and motivate its Greek prefix—standing for countable infinity, as it were. This plays a twofold role in the extension of the rewriting systems of our concern, as follows.

1. All operators of the word algebras which represent the state space are unary ones, with the obvious exception of the constant symbol, hence term trees are just plain operator sequences, ending either in the constant symbol if they are ground terms (i.e. words), or in a variable otherwise. We first extend the previously introduced rewriting framework by replacing the constant symbol with an infinite sequence of unary operators. Words thus become infinite, and variable substitution by such words turns a finite term structure into an infinite one.

2. Since rewrite rules act on infinite terms, the concept of limit of an infinite sequence of rewriting steps comes into play [3, 1, 2]. Such limits are needed in order to obtain a faithful extension of the rewrite relations, previously defined on objects represented by finite terms, to the same objects represented by infinite terms.

The reader may wonder what's the point with introducing the complication of infinity in a representation where an only finite prefix of the term structure conveys all the information about it, while the neverending tail is just the iteration of a given finite period. The point is twofold.

1. The fear of complication is actually defeated by a greater simplicity of the resulting system of rewrite rules, in that all the rules which, in any of the finite-term rewriting systems of our interest, are needed to

formalize the basis clause of an inductive specification, those rules may just be dispensed of in the ω-rewriting extension of that system. This happens not just because “there’s no basis anymore”, but because the rewriting outcomes specified by the aforementioned rules turn out to be (at the limit, possibly) the outcomes of (possibly infinite) sequences of applications of the other rules, under the new representation of the constant.

2. A more profound point of motivation for the subject extensions is the mathematical interest in extensions of the Collatz dynamics to richer domains, such as the integers, the rational field, dyadic number systems, etc. In this paper we show that, by removing the constraint that the ω-words of interest must end in an iterated one-symbol tail, the ground is settled for a straightforward extension of the Collatz problem to a dense domain. This leads us to the final kind of variations on the theme that are addressed in this paper.

Even when states in Collatz dynamics are represented by finite words, behaviour is inherently infinite; it can be either finitely presentable, that is to say, eventually periodic, or period-free. The latter is at least a possibility, in the standard Collatz problem, as long as its second half is not settled in the negative; but it is more than that, plain reality as a matter of fact, in certain well-known cases of generalized formulations of the Collatz problem. Generalization might be pushed to the extent of giving rise to further options of finitely presentable behaviour, yet with no eventual periodicity; for instance, almost-periodicity as well as quasiperiodicity would be of theoretical as well as practical interest according to [7, 6]. In all cases, behaviour in state transition dynamics is an infinite sequence of observations—an infinite parity vector indeed.

When that behaviour becomes the central subject of investigation, one is naturally led to investigate conceptual and formal tools to operate on behaviours, or representations thereof, to link their infinite evolution to that of the states they come across, and maybe to lift questions and techniques to the upper floor of abstraction. An interesting point in case is the equivalent formulation of the Collatz conjecture proposed by Lagarias [4], in connection with the extension of the Collatz transformation to the dyadic numbers $\mathbf{Z}_2$. This is briefly summarized as follows. One defines a transformation $\mathbf{Q}_\infty$ on this field by the infinite parity vector, whereby the image of any dyadic number state is the dyadic number whose digits are the components of the infinite parity vector under the Collatz transform starting from that state. The Collatz conjecture is then equivalent to the inclusion of the $\mathbf{Q}_\infty$-image of the positive integer subset of $\mathbf{Z}_2$ in the $\frac{1}{3}\mathbf{Z}$ subfield of $\mathbf{Z}_2$.

To get a first essay of what an ω-rewriting framework may have to offer in respect of behaviour analysis in Collatz-like dynamics, just consider the following question. What does a variable stand for in an ω-rewriting system? The basic mechanism of substitution is not unlike that of ordinary rewriting, yet the infinite size of ground terms (which may convey an amount of information that may need infinite time to be appraised) prompts us to contemplate a sort of high-school algebra view of variables as "unknowns", but actually in an informational sense. That is to say, by the substitution of a variable with a term that may contain variables, but definitely contains a finite, or finitely described, operators' structure, a gain of information takes place about the evolution of the original term, whereby its set of ground instances (its "substitution extent") shrinks to a subset of the original set that was available before applying the substitution. This may sound a bit too familiar, but the net effect is that of opening the way to a conceptual view of substitution as a process that takes time, in contrast to its instantaneous, timeless view that seems traditional.

Here is an outline of the contents of the forthcoming sections. The basic ingredients and notation adopted to establish the formal and conceptual framework presented in this paper, are introduced in the next section. We then move to the core of our subject, which consists in building a sequence of term rewriting systems, first the standard, finite-term rewriting ones, then their extensions to ω-rewriting, according to the series of variations on the theme outlined above. We finally draw brief conclusions and have a glimpse at forthcoming research directions.

2 Preliminaries

Basic concepts and notation are established as follows.

The *one-step, binary-labelled Collatz reduction relation* is another form of the first variant of the Collatz function that is considered here, actually a ternary relation in $\mathbf{N}_+\times\mathbf{2}\times\mathbf{N}_+$, with $\mathbf{N}_+=\mathbf{N}\setminus\{0\}$, that is defined as the smallest such relation which satisfies the following clauses:

$$x\xrightarrow{1}\tfrac{3x+1}{2} \text{ if } x \text{ is odd}, \quad x\xrightarrow{0}\tfrac{x}{2} \text{ if } x \text{ is even}.$$

By taking $\mathbf{N}_+$ as the set of states and any element n of $\mathbf{N}_+$ as initial state, one gets a (deterministic) *labelled transition system* (LTS) starting at n and thereafter evolving according to the binary-labelled transitions between states as specified by the reduction relation displayed above.

The binary label b in the $x\xrightarrow{b}y$ reduction step, is thus the parity of x. By lifting the label set to the set of nonempty finite bitstrings $\mathbf{2}^+$, one-step reductions are embedded into many-step reductions in a straightforward manner. The *bitstring-labelled Collatz reduction relation* is defined by extending the previous basic clauses with the following inductive clause:

$$x \xrightarrow{\sigma b} y \text{ if } x \xrightarrow{\sigma} z \xrightarrow{b} y \text{ for some } z \in \mathbf{N}_+$$

where b is the one-bit suffix of the σb bitstring. As a matter of notation, in this paper σ is always a variable ranging over bitstrings, and the one-bit suffix operation on bitstrings is denoted by juxtaposition.

It is easy (and will prove inspiring) to reformulate the basic clauses which specify one-step reductions in such a form that neither fractional expressions nor explicit parity conditions occur; given that x ranges over $\mathbf{N}_+$, one may equivalently write them as follows:

$$2x \xrightarrow{0} x, \qquad 2x-1 \xrightarrow{1} 3x-1 \ .$$

This reformulation does not affect the resulting transition system, but it affects the computational means employed to let the transition relation determine the progress of the system through the state space.

Consider now the second variant of the Collatz problem, where the state space is restricted to the *odd* positive integers. There are no numbers to halve in this space anymore, but the one-step labelled reduction relation now has a different label set, whereby each transition step is labelled by a finite bitstring 10^k, where k is the (possibly zero) number of divisions by 2 that are required to get an odd number out of $\frac{3n+1}{2}$, for any given source state (that is, odd positive integer) n; the target state then is $\frac{3n+1}{2^{k+1}}$.

3 Collatz dynamics by term rewriting

When symbolic computation means are of interest, then the design of a suitable *term rewriting system* (TRS) is a straightforward choice. This proceeds by devising a syntactic representation of the state space, and then by transferring the state transition relation over this representation. In the case of a *labelled* transition system, the TRS rewrite rules will conveniently act upon an extended representation of the LTS state space, that will be referred to as *TRS state space*, where terms representing LTS states are paired with label sequences which summarize the history of transitions leading from the initial state (paired with the empty sequence) to the given state. For the problem under consideration, a solution is obtained as follows.

LTS states are represented by words over the signature Σ_C which consists of a constant symbol $\perp$ and the two unary prefix operators $\mathtt{0}$, $\mathtt{1}$, modulo the

word congruence specified by the equation $\mathtt{1}\bot = \bot$. The correspondence of this representation to the given state space, that is $\mathbf{N}_+$, is established by interpreting $\bot$ as the number 1, that is the bottom of the natural order on $\mathbf{N}_+$, and the two unary operators respectively as follows:
$\mathtt{0} \mapsto \lambda x.2x, \quad \mathtt{1} \mapsto \lambda x.2x-1$ ("$\mapsto$" denotes application of the interpretation map).

To start with, let the TRS state space consist of the terms built by the binary infix operator #, pairing a (possibly empty) bitstring on the left (a transition label history) with a Σ_C-word on the right (the reached state, modulo the aforementioned congruence). Please note the typeface difference to distinguish the bits 0, 1, on the left of #, from the unary Σ_C-operators $\mathtt{0}$, $\mathtt{1}$, on the right of #. It would be nice if one could formalize the state transition relation just by rewrite rules acting on terms over the aforementioned TRS state representation signature, possibly with variables, but life is not always easy. The difficulty here arises from the severely limited computational power of the available operations, whereby multiplication by 3, among others, is hard to express. What seems purposeful to overcome this problem, is the expansion of the Σ_C-term algebra to a richer operators' signature, so as to include the needed operations to the computational means of the TRS under design. We do so by introducing three unary *auxiliary operators*, that is we extend Σ_C to the signature $\Sigma_A = \Sigma_C + \{\mathtt{t}, \mathtt{r}, \mathtt{s}\}$, with intended interpretation of the new symbols as the following functions on $\mathbf{N}_+$:
$\mathtt{t} \mapsto \lambda x.3x, \quad \mathtt{r} \mapsto \lambda x.3x-1, \quad \mathtt{s} \mapsto \lambda x.3x-2 \quad .$

The rewrite rules are going to act on terms to express either dynamical properties of the original LTS or computational properties of its representation. This is to be expected, in that the pairing operator # will now have a Σ_A-word on the right, where occurrence of any of the newly introduced operators is the symptom of an as yet unfinished computation of the target state of a transition. The rewrite rules are thus conveniently grouped under two categories: the set $\mathcal{D}$ of *dynamical* rules, where the pattern only has operators from $\{\#\} \cup \Sigma_C$ (and possibly variables), and the set $\mathcal{C}$ of *auxiliary computational* rules, where some of the auxiliary operators occurs in the pattern. We further orthogonally classify the rewrite rules by distinguishing the *boundary* rules, which are those where the constant symbol $\bot$ occurs in the pattern, from the *nonboundary* ones. The set $\mathcal{B}$ of boundary rules is thus comprised of those which are dynamical, viz. its subset $\mathcal{B}_{\mathcal{D}}$, and those which formalize auxiliary computation at the constant, viz. its subset $\mathcal{B}_{\mathcal{C}}$. Similarly, the set $\mathcal{A}$ of nonboundary rules is comprised of its subsets $\mathcal{A}_{\mathcal{D}}$ and $\mathcal{A}_{\mathcal{C}}$, respectively gathering the dynamical and auxiliary computational cases thereof.

$\mathcal{B}_{\mathcal{D}}: \quad \sigma\#\bot \to \sigma 1\#0\bot$	$\mathcal{B}_{\mathcal{C}}: \quad \mathtt{r}\bot \to \mathtt{0}\bot$	$\mathcal{A}_{\mathcal{C}}: \quad \mathtt{r0}x \to \mathtt{1t}x, \quad \mathtt{r1}x \to \mathtt{0s}x$
$\mathcal{A}_{\mathcal{D}}: \quad \sigma\#0x \to \sigma 0\#x$	$\mathtt{s}\bot \to \bot$	$\mathtt{s0}x \to \mathtt{0r}x, \quad \mathtt{s1}x \to \mathtt{1s}x$
$\sigma\#1x \to \sigma 1\#\mathtt{r}x$	$\mathtt{t}\bot \to \mathtt{10}\bot$	$\mathtt{t0}x \to \mathtt{0t}x, \quad \mathtt{t1}x \to \mathtt{1r}x$

Consider now the second variant of the Collatz problem, where the state space is restricted to the *odd* positive integers. We may essentially keep the same rewrite rules as with the previous system, but under a different classification. This arises from the fact that certain states from the previous state space, viz. the even positive numbers, are no longer found in the restricted state space. This entails that not all Σ_C-words represent states of the restricted LTS; only those which do not have '0' as outermost operator do so. One may consider the Σ_C-representations of the removed states, that occur at the right of the pairing operator, as auxiliary computational states of the new TRS. This boils down to move the first $\mathcal{A}_{\mathcal{D}}$-rule from the set of the dynamical rules to that of the auxiliary computational ones.

Simply stated, the dynamical rules for the restricted system are those which only exhibit operators from $\{\#\}\cup\Sigma_C\setminus\{\mathtt{0}\}$ (and possibly variables) in their pattern. Another formal difference is that transition label histories, at the left of the pairing operator, now have bitstrings of form 10^k as one-step transition (or atomic) labels of the restricted LTS. This fact corroborates the auxiliary computational nature of the first $\mathcal{A}_{\mathcal{D}}$-rule, since this contributes to the computation of an as yet unfinished atomic label.

The syntactic representation outlined above for the restricted LTS has an apparent drawback in that, under the given interpretation of Σ_C, not all Σ_C-words represent states of the LTS, but on the other hand no proper subsignature of Σ_C generates the LTS state space. For example, the number 3 is represented by the Σ_C-word $\mathtt{10}\bot$ (modulo $\mathtt{1}\bot = \bot$), hence the $\mathtt{0}$ occurrence cannot be dispensed of. However, an alternative interpretation of Σ_C exists that does not feature such a drawback. This coincides with the previous interpretation for the constant symbol $\bot$ and the $\mathtt{1}$ operator, but differs for $\mathtt{0}$: $\mapsto \lambda x.2x+1$. Furthermore, the two functions of our concern here are restricted to the odd positive numbers. Then one may easily see that the restricted LTS state space is represented, under this interpretation, by the Σ_C-word algebra modulo the same congruence recalled above.

One may expect the just introduced change of interpretation to have a somewhat dramatic impact on the syntactic representation of the LTS dynamics by a TRS. On the whole, this is indeed the case. However, not everything of the previous TRS structure is lost: 1) the signature extension to Σ_A is kept unchanged, whereas only a consistent change of interpretation for two of the three auxiliary operators is needed; 2) a further auxiliary unary operator is also needed, to deal with the unbounded length of 0-bit substrings in one-step transition labels; 3) the dynamical rules are substantially different from those of the previous TRS, whilst the present computational rules can be kept unchanged, but for extension of the $\mathcal{A}_\mathcal{C}$ and $\mathcal{B}_\mathcal{C}$ sets with rules relating to the new auxiliary operator. The technically detailed meaning of these statements and validity of related claims are deferred to Appendix A. For the purposes of the present analysis, the greater formal simplicity of the first variant of the Collatz problem turns out to offer the best basis for extensions and further investigation. The next section is aimed at pushing that formal simplicity even further.

4 Collatz dynamics by ω-rewriting

Following [3], infinitary rewriting extends term rewriting by allowing infinite terms and infinite rewriting sequences. These naturally arise in programming, for example in the correspondence between graph rewriting and term rewriting that is the foundation of useful implementation techniques for functional languages. Cyclic graphs frequently occur as an important ingredient of optimization techniques, and they naturally correspond to infinite terms. The rewriting of cyclic graphs then corresponds to infinite computations on those terms. Cyclic graphs also provide one with a most natural representation in state transition dynamics, whenever periodicity is of concern. The previously recalled statement of the Collatz problem is just one out of many examples of this obvious fact.

There seems to be sufficient motivation to try a variant of the TRS formalization of the Collatz dynamics presented in the previous section, whereby the presence of suitable cycles is rendered by the infinitary means of ω-rewriting. To this purpose, as well as to exemplify which cycles fit the purpose, consider the following facts.

On the one hand, the dynamics of the subject system, as formalized by the $\mathcal{D}$-rules, immediately reveal the presence of a 2-state cycle, through the states represented by the Σ_C-words $\perp$ and $\mathtt{0}\perp$. On the other hand, a (different kind of) cycle is also found among the static features of the representation, whereby Σ_C-words represent states modulo the congruence $\mathtt{1}\perp = \perp$. This corresponds to the fact that the bottom state, represented by $\perp$, is the fixed point of the function $\lambda x.2x{-}1$, interpreting the operator $\mathtt{1}$. This fact prompts us to consider the representation of the bottom state by the infinite Σ_C-word $\mathtt{1}^\omega$ as a natural alternative to its representation by a constant symbol. This is indeed a viable alternative in the framework of infinitary rewriting.

Now, consider what happens with the previous TRS when its rules are turned into infinitary ones by replacing every occurrence of $\perp$ with $\mathtt{1}^\omega$. It just turns out that all rules where this occurs, that is to say, all boundary rules, can be deduced from the other rules by transfinite induction. For example, when considering the rewriting of the ω-term $\sigma\#1^\omega$, one finds:

$$\begin{array}{llll} \sigma\#\mathtt{1}^\omega & \to & \sigma 1\# r\mathtt{1}^\omega & \text{by the second } \mathcal{A}_\mathcal{D} - \text{rule} \\ & \to & \sigma 1\# 0 s\mathtt{1}^\omega & \text{by the second } \mathcal{A}_\mathcal{C} - \text{rule} \\ & \to^\omega & \sigma 1\# \mathtt{01}^\omega & \text{by the fourth } \mathcal{A}_\mathcal{C} - \text{rule } (\omega \text{ times}). \end{array}$$

By a similar reasoning, all other boundary rules can be dispensed of. The aesthetic appeal of the resulting ω-TRS accrues from the fact that it is reduced to a “strong core”, where two dynamical rules formalize the corresponding basic clauses of the original LTS dynamics, while the only auxiliary computational rules are the nonboundary ones. The form of these appears to be “robust” even with respect to changes of interpretation, such as those which help one reusing the rewrite rules to formalize variants of the dynamical system like that mentioned at the end of the previous section.

Now, in our view, the main reason for interest in formal robustness, in the sense just outlined above, is to be found in that it highlights a greater role for *syntactic abstraction* in the characterization, analysis and understanding of essential features and inherent properties of state transition dynamics. The Collatz problem, and variants thereof, all arise from, and live within, an arithmetic world; however, the hard questions they address, such as characterizations of periodicity and of attractors, truly are of a dynamical nature, and similar questions can be found in much diverse worlds, such as systems biology or wheather forecasting—just to mention a couple of them. While the building of syntactic representations of those kinds of problems surely is an interesting and potentially useful exercise in most cases, we believe it brings a greater added value when representations enjoy neatness and robustness, because of their potential to yield deeper insight and wider understanding.

Indeed, coming back to the subject problem, a similar situation occurs with the infinitary TRS for the aforementioned “restricted” Collatz dynamics that is worked out in Appendix A, that is, boundary rules can be elegantly dispensed of there as well.

Even more interesting tests of the usefulness of ω-rewriting to the aforementioned purpose are to be expected from exploring the converse direction of variations on the theme, that is to say, extensions of the problem under study. An effort in this direction is what comes next.

5 Extended Collatz dynamics

As shown in the previous section, the representation of the constant by an infinite word in the ω-TRS formalization of the Collatz dynamics has the pleasant effect of letting the boundary rewrite rules become redundant. On the other hand, the infinite word which represents the constant may also be seen as the unfolding of a fixed point equation which corresponds to the congruence relation on the word algebra that is induced by its arithmetic interpretation. This yields a straightforward method to design and investigate ω-TRS formalizations of a certain class of extensions of the Collatz problem to larger domains. In a nutshell, the method consists of the following steps:

1. specify the extension of the Collatz problem domain by introducing new constants, so that the new elements in the larger domain are generated by the new constants and available operations;

2. determine the congruence relation that is induced on the extended word algebra by its interpretation in the extended domain;

3. determine the ω-words which represent the new constants by unfolding the equations that specify the interpretation congruence and involve the new constant symbols, which thus become redundant.

A simple illustration of this method is found in the extension of the Collatz problem to the integers, thus including 0 as well as the negative integers, as follows.

1. The set of available operations, consisting of the constant 1 and the unary $\lambda x.2x$ and $\lambda x.2x{-}1$, is extended with a new constant: the number 0. Every negative integer may then be obtained as the result of a suitable composition of the available operations, applied to the new constant.

2. The word algebra is extended with the constant symbol $\top$ to designate the new constant; its interpretation congruence is specified by adjoining the previous equation with a new one: $\mathtt{0}\top = \top$.

3. The ω-word which represents the new constant is thus $\mathtt{0}^\omega$. This corresponds to the fact that the new constant is the fixed point of the function $\lambda x.2x$, interpreting the operator $\mathtt{0}$. The $\top$ symbol may be dispensed of.

Now, the extension of the ω-TRS with the word $\mathtt{0}^\omega$ is all that is needed to represent the extension of the Collatz problem to the integers. This happens because:

- the two nonboundary dynamical rules do not change, since they formalize the dynamics of the transition system, which is extended to the new state space precisely in the sense that the form of the dynamical rules is kept unchanged;

- the $\mathcal{A}_\mathcal{C}$ rules do not change, essentially for a similar reason—they formalize the nonboundary dynamics of the auxiliary computation, and this is extended to the new computational state space much in the same sense;

- were we to deal with a finitary TRS representation, new boundary rules would be badly needed, to cater for dynamics and auxiliary computation at the new constant; we leave it to the reader as an exercise to write down such rules, and then to check that their translations under the ω-TRS representation of the new constant are subsumed by the previous two classes of rules.

In the variants as well as in the extension of the Collatz problem hitherto considered, the state space is generated from the operations starting from constants that are represented by given ω-words. One may then look for further extensions by considering the interpretation of additional ω-words in a given state space: if that is not defined, then an extension of the state space can be contemplated.

To start with, consider one-symbol ω-words. In the case under study, both such words for the two unary operators in Σ_C have a defined interpretation in the integers, hence one is led to consider the auxiliary operators. It turns out that two of them yield a similar situation, no extension thus, in that the numbers 0 and 1 respectively solve the fixed point equations $\mathtt{t}x = x$ and $\mathtt{s}x = x$, hence yield interpretation of the ω-words $\mathtt{t}^\omega$ and $\mathtt{s}^\omega$. This is not the case for the $\mathtt{r}$ operator; its interpretation $\lambda x.3x-1$ has no integer fixed point, but just the rational $\frac{1}{2}$. The rational extension of $\mathbf{Z}$ that is generated by composition of the available operations applied to this new constant, represented by $\mathtt{r}^\omega$, is not a terribly interesting one, however. This extension is just a discrete subset of $\mathbf{Q}$, viz. $\frac{1}{2}\mathbf{Z}$; its noninteger elements, *i.e.* those in $\frac{1}{2}\mathbf{Z}\backslash\mathbf{Z}$, are the interpretations of infinite Σ_A-words where no Σ_C operator occurs.

In order to generate a dense extension of $\mathbf{Z}$ by application of the available operations to new, noninteger constants interpreting infinite words, these must exhibit infinite occurrences of more than one operator. Let's thus admit all periodic ω-words over the unary operators of Σ_C as constant symbols. Unlike the previous extension by $\mathtt{r}^\omega$, this one keeps the auxiliary, derived nature of the operations which interpret the operators in $\Sigma_A \backslash \Sigma_C$, using the $\mathcal{A}_\mathcal{C}$ rules as computation rules. On the other hand, this extension does generate a dense, albeit proper subset of $\mathbf{Q}$. This is the set

$$\left\{ \frac{m}{2^n - 1} \mid m \in \mathbf{Z}, n \in \mathbf{N}_+ \right\}$$

and a proof that it is dense is worked out in Appendix B. To realize that this is indeed the set generated by the subject extension, consider the interpretation of the ω-words in $\{\pi^\omega | \pi \in \{\mathtt{0}, \mathtt{1}\}^*, |\pi| > 0\}$. Clearly, the interpretation of π is a linear function $\lambda x.2^n x – m$, with $n = |\pi|$, for some integer $m \geq 0$ that only depends on π, hence the interpretation of π^ω is the unique fixed point of that function, viz. the solution of $2^n x - m = x$. One may write this by using the fixed point operator: $\mu x.2^n x - m$; thus, for example, $(\mathtt{10})^\omega \mapsto \mu x.4x - 1 = \frac{1}{3}$, $(\mathtt{01})^\omega \mapsto \mu x.4x - 2 = \frac{2}{3}$, $(\mathtt{110})^\omega \mapsto \mu x.8x - 3 = \frac{3}{7}$. The negative elements in the aforementioned set are then generated by application of compositions of the available operations to the constants in question.

A useful outcome of this way of extending the Collatz problem domain is that the rewrite rules in $\mathcal{A}_\mathcal{D} \cup \mathcal{A}_\mathcal{C}$ are unaffected. That is to say, one gets a straightforward way of extending both the dynamical and the computational mechanisms of the Collatz problem to the wider domain. An alternative, known way of doing so goes by embedding the wider domain into the dyadic number field $\mathbf{Z}_2$, whereby the first dyadic digit tells the parity [4]. This alternative exhibits a distinct advantage over the fixed point semantics employed here, in that the latter breaks down when one tries to widen the ω-language of constants symbols beyond the limits of periodicity, e.g. by admitting quasiperiodic ω-words [6], whilst no such drawback affects the interpretation of any infinite binary word in $\mathbf{Z}_2$.

Another use of quasiperiodic binary words in the investigation of the Collatz problem, and of extensions thereof, contemplates their occurrence as parity vectors, or as labels of possibly infinite transition sequences. There is no problem with fixed point semantics in this case, since the structure of labels rather than states is of concern now. Existence of quasiperiodic parity vectors that are not eventually periodic is only possible in those extensions of the Collatz problem where the second half of the Conjecture fails.

6 Conclusions

The use of infinitary rewriting has been investigated in this note as a formalization tool to specify both dynamical and computational aspects of the Collatz problem and extensions thereof. Merits of this approach, in comparison with standard, finite term rewriting, have been highlighted. Limitations of the proposed approach have been pointed out, when dealing with extensions of the Collatz problem where space structure is not only dense (which is no problem *per se*), but also beyond (eventual) periodicity.

Here are a few directions of forthcoming research on the problems considered in this note. Known open questions relating to the Collatz problem, and extensions thereof, can be formulated in the theory of state transition dynamics [5], where, for example, the two halves of the original Collatz problem can be respectively stated as the uniqueness of the trivial period and the (non)existence of "flights" (*i.e.* divergent, or period-free reductions) in the attractor of the Collatz dynamics. The difficulty of these questions seems to arise from what has been often called "chaotic" behaviour of Collatz reductions. We should like to try to get a more precise characterization of this purported property of the reduction systems under study, by analysing which of those features of chaos in state transition dynamics that are identified in [5], may be recognized in the transition systems of our concern. Furthermore, the existence of extensions of the Collatz dynamics that would exhibit quasiperiodic flights is an open question.

Acknowledgements

The author wishes to thank Vincenzo Manca for having made him aware of the problem, in the fruitful and inspiring context of our joint work, together with Giuditta Franco, on state transition dynamics.

References

[1] I. Bethke, J. W. Klop, and R. de Vrijer. Descendants and origins in term rewriting. *Information and Computation*, 159:59–124, 2000.

[2] R. Kennaway. Infinitary rewriting and cyclic graphs. *Electronic Notes in Theoretical Computer Science*, 2, 2000.

[3] R. Kennaway, J. W. Klop, R. Sleep, and F.-J. de Vries. Transfinite reductions in orthogonal term rewriting systems. *Information and Computation*, 119(1):18–38, 1995.

[4] J. Lagarias. The 3x+1 problem and its generalizations. *American Mathematical Monthly*, 92:3–23, 1985.

[5] V. Manca, G. Franco, and G. Scollo. State transition dynamics: basic concepts and molecular computing perspectives. In M. Gheorghe and M. Holcombe, editors, *Molecular Computational Models: Unconventional Approaches*. Idea Group, Hershey, PA, USA, 2004.

[6] S. Marcus. Quasiperiodic infinite words. *Bull. of the EATCS*, 82:170–174, 2004.

[7] S. Marcus and G. Paun. Infinite (almost periodic) words, formal languages and dynamical systems. *Bull. of the EATCS*, 54:224–231, 1994.

[8] R. Terra. A stopping time problem on the positive integers. *Acta Arith.*, 30:241–252, 1976.

[9] R. Terra. On the existence of a density. *Acta Arith.*, 35:101–102, 1979.

A TRS and ω-TRS for the restricted variant of the Collatz problem

As mentioned in section 3, the LTS state space of the Collatz dynamics restricted to the odd positive integers, has a syntactic representation as the Σ_C-word algebra (modulo $\mathtt{1}\perp = \perp$), corresponding to the interpretation of Σ_C over the odd positive integers specified by:
$\perp \mapsto 1, \;\; \mathtt{0} \mapsto \lambda x.2x{+}1, \;\; \mathtt{1} \mapsto \lambda x.2x{-}1.$

A (finitary) TRS that formalizes this restricted Collatz dynamics by syntactic computations is built by extending Σ_C to $\Sigma_A + \{\mathtt{u}\}$, where again $\Sigma_A = \Sigma_C + \{\mathtt{t}, \mathtt{r}, \mathtt{s}\}$, but $\mathtt{u}$ is a new, unary auxiliary operator. The interpretation of the auxiliary operators in Σ_A differs from that seen in Section 3, consistently with the changed interpretation of Σ_C, in that we now take the following functions over the odd positive integers:
$\mathtt{t} \mapsto \lambda x.3x{+}2, \;\; \mathtt{r} \mapsto \lambda x.3x, \;\; \mathtt{s} \mapsto \lambda x.3x{-}2 \;\; .$

We need not specify an interpretation for the new auxiliary operator $\mathtt{u}$, since it only plays a merely syntactic role in the computation dynamics, that is easier to grasp in the context of the presentation of the TRS rewrite rules. These are designed according to the following facts.

Let $\mathtt{p}$, $\mathtt{q}$ be metavariables, ranging over the unary operators $\{\mathtt{0},\mathtt{1}\}$ in Σ_C, and x a variable, ranging over the odd positive integers. Evaluation of $\mathtt{pq}x$ always yields an odd positive integer, hence its residual $\pmod 8$ may take one out of only four values. This is actually independent of the value assigned to x, since it is determined by the outermost operators' pair pq, under the aforementioned interpretation of Σ_C, according to the following evaluation map $\pmod 8$: $\texttt{11}x \mapsto 1,\ \ \texttt{01}x \mapsto 3,\ \ \texttt{10}x \mapsto 5,\ \ \texttt{00}x \mapsto 7\ \ .$

Now, if we evaluate $\texttt{pq}x$ in the restricted Collatz LTS state space, the case $\texttt{10}x$ is the only one which somewhat complicates our TRS design a little, as follows. If $8n+m$ is an evaluation of $\texttt{pq}x$ for some $n \in \mathbf{N}$, $m \in \{1,3,5,7\}$, taken as the source state of a transition in our LTS, then the target state of that transition will be $\frac{3(8n+m)+1}{2^{k+1}}$, where $k+1$ is the exponent of 2 in the prime factorization of $3(8n+m)+1$. The reader may check that $k > 1$ iff $m = 5$ iff pq=10. In all other cases, the atomic label 10^k for the transition to the target state will be computed by just one rewrite rule of our TRS. On the contrary, states represented by the $\texttt{10}x$ pattern are starting states of a computation aimed at establishing the value of k. This is accomplished by means of the u operator. As the reader will see, each rule where this operator occurs in the pattern, mimics a corresponding rule without it in the pattern, in that its right-hand-side term coincides with that of the rule it mimics, except for the absence of a bit 1 occurrence in the extension of the label history (clearly, each occurrence of bit 1 in the label history marks the start of an atomic label).

Given this premiss, here are the rewrite rules of the subject TRS (modulo $\texttt{1}\bot = \bot$):

$\mathcal{B}_\mathcal{D}$:	$\mathcal{B}_\mathcal{C}$:	$\mathcal{A}_\mathcal{C}$:
$\sigma\#\bot \to \sigma 10\#\bot$	$\sigma\#\texttt{u}\bot \to \sigma 0\#\bot$	$\texttt{r0}x\to \texttt{1t}x$, $\texttt{r1}x\to \texttt{0s}x$
$\sigma\#\texttt{0}\bot \to \sigma 1\#\texttt{10}\bot$	$\sigma\#\texttt{u0}\bot \to \sigma\#\texttt{10}\bot$	$\texttt{s0}x\to \texttt{0r}x$, $\texttt{s1}x\to \texttt{1s}x$
$\mathcal{A}_\mathcal{D}$:		$\texttt{t0}x\to \texttt{0t}x$, $\texttt{t1}x\to \texttt{1r}x$
$\sigma\#\texttt{11}x \to \sigma 10\#\texttt{s}x$	$\texttt{r}\bot \to \texttt{0}\bot$	$\sigma\#\texttt{u11}x \to \sigma 0\#\texttt{s}x$
$\sigma\#\texttt{01}x \to \sigma 1\#\texttt{1r}x$	$\texttt{s}\bot \to \bot$	$\sigma\#\texttt{u01}x \to \sigma\#\texttt{1r}x$
$\sigma\#\texttt{10}x \to \sigma 100\#\texttt{u}x$	$\texttt{t}\bot \to \texttt{10}\bot$	$\sigma\#\texttt{u10}x \to \sigma 00\#\texttt{u}x$
$\sigma\#\texttt{00}x \to \sigma 1\#\texttt{0t}x$		$\sigma\#\texttt{u00}x \to \sigma\#\texttt{0t}x$

By transfinite induction, it is easy to see that, also in this representation of the restricted variant of the Collatz problem, the representation of the constant by the infinite word $\texttt{1}^\omega$ in the ω-TRS formalization of the Collatz dynamics, makes the boundary rewrite rules to become redundant.

B Proof of claim in Section 4

The claim is stated in section 4, that the subset $\left\{\frac{m}{2^n-1} \mid m \in \mathbf{Z},\ n \in \mathbf{N}_+\right\}$ of $\mathbf{Q}$ is dense. Here is a proof of this statement. Let $\frac{m}{2^n-1} < \frac{p}{2^q-1}$ by assumption;

the goal is to show that $\exists x \in \mathbf{Z},\ y \in \mathbf{N}_+ \,.\, \frac{m}{2^n-1} < \frac{x}{2^y-1} < \frac{p}{2^q-1}$.

One may restrict the assumption to $0 \leq \frac{m}{2^n-1} < \frac{p}{2^q-1} \leq 1$, for otherwise either of the following two cases occurs:

- $\exists i \in \mathbf{Z}.\ \frac{m}{2^n-1} < i < \frac{p}{2^q-1}$, in which case the goal is achieved by $x = i,\ y = 1$;

- $\exists i \in \mathbf{Z}.\ i \leq \frac{m}{2^n-1} < \frac{p}{2^q-1} \leq i+1$, which case is amened to the stated assumption by considering $\frac{m}{2^n-1} - i < \frac{p}{2^q-1} - i$, that is $0 \leq \frac{m-(2^n-1)i}{2^n-1} < \frac{p-(2^q-1)i}{2^q-1} \leq 1$, and then from a solution $\frac{x'}{2^{y'}-1}$ to this, a solution $\frac{x}{2^y-1}$ to the case in question is easily obtained by letting $x = x' + (2^{y'}-1)i,\ y = y'$.

Assume $n \geq q$. Let then $y = 2n, x = m(2^n+1)+1$. This entails

$$\frac{x}{2^y-1} = \frac{m(2^n+1)+1}{2^{2n}-1} = \frac{m}{2^n-1} + \frac{1}{2^{2n}-1} \text{ therefore } \frac{m}{2^n-1} < \frac{x}{2^y-1}$$

and moreover $\frac{x}{2^y-1} < \frac{m}{2^n-1} + \frac{1}{(2^n-1)(2^q-1)}$ because $2^n+1 > 2^q-1$. Now, since

$$\frac{m}{2^n-1} + \frac{1}{(2^n-1)(2^q-1)} = \frac{m(2^q-1)+1}{(2^n-1)(2^q-1)}$$

and

$$\frac{m}{2^n-1} < \frac{p}{2^q-1} \Longrightarrow m(2^q-1)+1 \leq p(2^n-1)$$

it follows that

$$\frac{x}{2^y-1} < \frac{m}{2^n-1} + \frac{1}{(2^n-1)(2^q-1)} \leq \frac{p(2^n-1)}{(2^n-1)(2^q-1)} = \frac{p}{2^q-1}\ .$$

Conversely, assume $q \geq n$. Let then $y = 2q, x = p(2^q+1)-1$. This entails

$$\frac{x}{2^y-1} = \frac{p(2^q+1)-1}{2^{2q}-1} = \frac{p}{2^q-1} - \frac{1}{2^{2q}-1} \text{ therefore } \frac{x}{2^y-1} < \frac{p}{2^q-1}$$

and moreover $\frac{x}{2^y-1} > \frac{p}{2^q-1} - \frac{1}{(2^n-1)(2^q-1)}$ because $2^q+1 > 2^n-1$. Now, "symmetrically" to the previous case, since

$$\frac{p}{2^q-1} - \frac{1}{(2^n-1)(2^q-1)} = \frac{p(2^n-1)-1}{(2^n-1)(2^q-1)}$$

and

$$\frac{m}{2^n-1} < \frac{p}{2^q-1} \Longrightarrow p(2^n-1)-1 \geq m(2^q-1)$$

it follows that

$$\frac{x}{2^y-1} > \frac{p(2^n-1)-1}{(2^n-1)(2^q-1)} \geq \frac{m(2^q-1)}{(2^n-1)(2^q-1)} = \frac{m}{2^n-1}\ .$$

University of Verona
Department of Computer Science
Strada Le Grazie, 15
I-37134 Verona
Italy

http://www.di.univr.it