Tag systems and Collatz-like functions

Liesbeth De Mol
elizabeth.demol@ugent.be
Gent University
Centre for Logic and Philosophy of Science
Blandijnberg 2, 9000 Gent, Belgium.

Abstract

After a short introduction to tag systems and their significance in the context of research on the limits of solvability, a method will be described to reduce the $3n+1$-problem to a tag system which is surprisingly small. This method will be generalized to arbitrary Collatz-like functions, proving how any such functions can be easily simulated by a tag system, thus providing an alternative proof of the general unsolvability of tag systems. Finally, these results will be related to some findings concerning the reducibility of Collatz-like functions to Turing machines studied in the context of the boundaries of solvability in Turing machines.

1 Introduction

Already in 1921 Emil Leon Post proved the unsolvability of certain decision problems, rooted in what Martin Davis has called Post's thesis [7][8]. In 1943 a paper appeared by Post that summarizes the main results from this earlier research [37], but it was only in 1965 that Martin Davis posthumously published a manuscript by Post describing these earlier results in more detail [38]. As was argued in [29], basic to these results was Post's construction of tag systems. After nine months of research, trying to prove the solvability of tag systems, he concluded for a reversal of his entire program. He was now convinced that there might exist unsolvable decision problems in mathematics and logic. Since he never wanted to work on tag systems again, he never proved them unsolvable. In the end it was Marvin Minsky who proved that tag systems are indeed unsolvable, by proving that any Turing machine can be simulated by a tag system [25].

1.1 A short introduction to tag systems

A tag system consists of a finite alphabet $\Sigma = \{a_0, a_1, ..., a_{\mu-1}\}$ of μ symbols, a shift number $v \in \mathbb{N}$ and a finite set of μ words defined over the alphabet,

including the empty word ϵ. Each of these words corresponds with one of the letters from the alphabet as follows:

$$\begin{array}{lcl} a_0 & \rightarrow & a_{0,1}a_{0,2}...a_{0,n_0} \\ a_1 & \rightarrow & a_{1,1}a_{1,2}...a_{1,n_1} \\ ... & ... & ... \\ a_{\mu-1} & \rightarrow & a_{\mu-1,1}a_{\mu-1,2}...a_{\mu-1,n_{\mu-1}} \end{array}$$

where each $a_{i,j} \in \Sigma, 0 \leq i < \mu$. Now, given an initial word w, the tag system tags the word associated with the leftmost letter of w at the end of w, and deletes its first v symbols. This process is iterated until the tag system halts, i.e. produces a word w_i, after i iterations, having a length smaller than v. If this does not happen the tag system can become periodic or show divergent behaviour.
To give an example, consider the case where $v = 3$, $0 \rightarrow 00$, $1 \rightarrow 1101$, with $w = 10111011101000000$. We then get:

10111011101000000
110111010000001101
1110100000011011101
01000000110111011101
0000011011101110100
001101110111010000
10111011101000000

The word w is reproduced after 6 steps, w thus giving rise to a period of length 6.
As simple as the definition of a tag system might be, they are very hard to study, and until now not very much is really known about these systems. Even the case where $\mu = 2, v > 2$ is still an open problem. Indeed, the seemingly simple example mentioned above, first described by Post [38],[37] is still not known to be solvable. Watanabe [44] studied this one specific case, trying to get a more formal grip on this tag system, without much success. Although Watanabe's paper is very interesting, it contains rather fundamental errors, leading to wrong conclusions about the possible periodic structures found in this tag system. Besides Watanabe, Minsky and Brian Hayes did some research on this one tag system (See for example [11], [12], [13], [28]) again without any definite results concerning the solvability of this tag system.
Despite the fact that tag systems have not been that well-studied, there are some results concerning their limits of solvability. In his posthumously published paper [38], Post mentions that the class of tag systems for which $v = 1$ or $\mu = 1$ is trivially solvable. He furthermore notes that he completely solved the case with $\mu = v = 2$, and considered this proof as the major result from his Procter fellowship in Princeton (1920–1921). The proofs however were never published. Wang [43] provided the proof of the solvability for the case where $v = 1$. We were able to find such a proof for the class $\mu = v = 2$, involving the application of a combinatorial kind of method applied to a rather large number of different

subcases (unpublished). Cocke and Minsky proved that any Turing machine can be simulated by a tag system for which $v = 2$ (See [1], [26]). Maslov generalized this result and proved that for any $v > 1$ there exists at least one tag system with an unsolvable decision problem and, independent of Wang, furthermore proved that any tag system for which $v = 1$ is solvable [21].
Both μ and v can thus be regarded as *decidability criteria* [20] for tag systems, since their solvability depends on the size of these parameters. Another such criterion for tag systems, is the length of the words. Let l_{min} denote the length of the smallest word of a tag system and l_{max} the length of the lengthiest word (introduced by [34]). Wang proved that any tag system for which $l_{min} \geq v$ or $l_{max} \leq v$ is solvable [43]. It should be added here that Maslov proved that the tag systems with an unsolvable decision problem that can be constructed using his method, for each $v > 1$ all satisfy the following condition: $l_{min} = v-1$, $l_{max} = v+1$ [21]. Taking into account Wang's result, he describes this condition as a kind of minimal condition for unsolvability in tag systems. This result was independently proven by Wang for a tag system with $v = 2$ [43].
As is clear from the previous, while μ, the number of symbols of a tag system determining the number of words and thus production rules, is not taken into account in studying the limits of solvability in tag systems, except by Post, it is clear that its role should not be underestimated in this context. Since the common measure used to determine the size of a Turing machine also includes the number of states and symbols, while μ is a decidability criterion for tag systems, it is natural to include μ in the definition of a measure for the size of tag systems. In this respect we would like to propose the following measure for tag systems:

Definition 1.1 *The size of a tag system is defined as the product of μ and v, where $TS(\mu, v)$ denotes the class of tag systems with μ symbols and a shiftnumber v.*

The length of the words is not taken into account, since the decidability criterion with respect to l_{min} and l_{max} is defined relative to v.

1.2 Tag systems and small universal systems

Although tag systems have not been studied as intensively as e.g. Turing machines, they are basic to the research on small universal systems and thus the limits of solvability. In [27], Minsky constructed a small 4-symbols, 7-state universal Turing machine, a machine able to simulate any tag system with a shift number $v = 2$. For years Minsky's machine was the smallest universal Turing machine known. For about the last twenty years a number of researchers have tried to find still smaller machines, their research being situated in the context of the limits of solvability and unsolvability. The smallest known classes containing a universal Turing machines are: TM(18,2) (Neary 2006, mentioned in [33]), TM(9,3) (Neary 2006, mentioned in [33]), TM(7,4) (Minsky 1961, [25]), TM(5,5) (Rogozhin 1982, [42]), TM(4,6) (Rogozhin 1982, [42]), TM(3,9) (Kudlek and Rogozhin 2002, [16]) and TM(2, 18) (Rogozhin 1996, [41]), where TM(m, n)

denotes the class of Turing machines with m states and n symbols. As far as solvability is concerned, it should be noted that Minsky mentions that he and Bobrow had been able to prove that the class of machines TM(2,2) is decidable, through a reduction to thirty-odd cases (See [28], p. 281), a shorter proof was published by Pavlotskaya [35]. She also proved that the class of machines TM(3,2) is solvable [36].

Significant here is the fact that all the known small Turing machines are proven to be universal because they are able to simulate any tag system with a shift number $v = 2$. One of the long-standing problems with respect to these small universal machines was that they are inefficient simulators of Turing machines. The reason for this is the fact that the universality of tag systems, with $v = 2$, is proven through their ability to simulate any Turing machine. This simulation however is exponentially slow. Recently this problem was resolved by Neary and Woods. They proved that tag systems, with $v = 2$, efficiently simulate cyclic tag systems [31], and furthermore proved that cyclic tag systems efficiently simulate Turing machines [32].

Turing machines are not the only class of computational systems in which small universal systems are proven to exist on the basis of the universality of tag systems. Matthew Cook's proof of the universality of the very small cellular automaton rule 110 is indirectly based on the simulation of tag systems, through its simulation of cyclic tag systems [4]. This proof however does not follow the definition of universality as given by Davis [5], since it involves the infinite repetition of the encoding of the production rules of the cyclic tag system in the initial condition. Another such class of examples is the universality of small circular Post machines [15].

Since tag systems lie at the basis of the known small universal systems, it is important to further study the limits of unsolvability in tag systems. Such research might provide a better insight in the question as to why one needs these systems to find small universal machines. A possible explanation for tag systems lying at the basis of the small universal systems is that their syntax is in a certain way much more simple. As a consequence, one expects that their boundaries of unsolvability will be considerably lower as compared to those for other computational systems such as cellular automata and Turing machines. The fact that there is no clear method at all to prove the solvability of the very simple class of tag systems $\mu = 2$, $v > 2$, serves as an indication of this idea. In this paper we will show that the $3n+1$-problem can be reduced to a tag system for which $\mu = 3, v = 2$.[1] As will be argued in Sec. 3, this fact is yet a further reason to suppose that the limits of unsolvability in tag systems might indeed be very low.

[1] It should be noted that Brian Hayes also mentioned the possible connection between the $3n + 1$-problem and tag systems [11].

2 A simple, efficient encoding of Collatz-like functions in tag systems

Let $C : \mathbb{N} \to \mathbb{N}$ be defined by:

$$C(n) = \begin{cases} \frac{n}{2} & \text{if n is even} \\ 3n+1 & \text{if n is odd} \end{cases}$$

The $3n+1$-problem is the problem to determine for any $n \in \mathbb{N}$, whether $C(n)$ will end in a loop $C(4) = 2, C(2) = 1, C(1) = 4$, after a finite number of iterates. The well-known $3n+1$-problem is one of those problems from number theory for which the statement of the problem is as simple as the problem is intractable. A survey on the $3n+1$ problem can be found in [17, 18], where [18] is a more recent and seriously extended version of [17]. An annotated bibliography can be found via Arxiv [19]. Although the general consensus is that $C(n)$ will always end in the same loop after a finite number of iterates for arbitrary n, no proof has been found until now. Some even claim that the $3n+1$-problem is unprovable (see e.g. [9]). A nice illustration of the difficulties involved with the $3n+1$-problem is given by the following quote by Kakutani:[2]

> For about a month everybody at Yale worked on it, with no result. A similar phenomenon happened when I mentioned it at the University of Chicago. A joke was made that this problem was part of a conspiracy to slow down mathematical research in the U.S.

In [24] Pascal Michel considers generalized functions of C, called Collatz-like functions. These are based on the following equivalent form of the $3n+1$-function:

$$C(2m) = m,$$
$$C(2m+1) = 3m+2.$$

Given integers $d \geq 2; a_0, a_1, ..., a_{d-1}; r_0, r_1, ..., r_{d-1}; x \in \mathbb{N}$ a Collatz-like function is defined as follows:[3]

$$G(n) = \begin{cases} m_0 & \text{If } n \equiv 0 \bmod d \\ m_1 & \text{If } n \equiv 1 \bmod d \\ \vdots & \\ m_{d-1} & \text{If } n \equiv (d-1) \bmod d \end{cases}$$

where m_i is either undefined or denotes an operation of the following form:

$$\frac{a_i(n-i)}{d} + r_i$$

[2] Quoted in [18] from a private conversation dated 1981, Kakutani describing what happened after he circulated the problem around 1960

[3] It should be noted that Michel further extends these functions to functions of pairs of integers.

Similar generalizations were already considered by Conway [2] in 1972. He proved that these generalizations lead to Collatz-like problems which are generally unsolvable. I.e. he proved that there exists no method to decide whether a Collatz-like function G, when applied to a number n, will produce 1 after a finite number of iterates by proving that any register machine can be simulated by such a function. About 15 years later, Conway developed a developed a simple universal programming language called Fractran [3], for doing arithmetic, its syntax being based on the methods he used in 1972. He furthermore constructed a universal fraction game, called the Polygame, on the basis of which one can rather easily construct a universal Collatz-like function. In [14], Kaščák gives an explicit construction of a universal *one-state linear operator algorithm*, involving a generalization of the Collatz-problem similar to Michel's, with a small modulus, equal to 396.

2.1 Reduction of the $3n+1$-function in tag systems

In this section we will prove the following theorem:

Theorem 2.1 *The function $C(n)$ is reducible to a tag system T_C with $\mu = 3$, $v = 2$.*

Let A^i denote a string A repeated i times, $A \xrightarrow{\circ} B$ is the string B produced from A, after all the letters from A have been erased. Let $\Sigma = \{\alpha, c, y\}$ and $n \in \mathbb{N}$. Then, each iteration of $C(n)$ corresponds to the production of a string $\alpha^{C(n)}$ from a string α^n in T_C. The production rules are:

$$\begin{array}{lcl} \alpha & \rightarrow & cy \\ c & \rightarrow & \alpha \\ y & \rightarrow & \alpha\alpha\alpha \end{array}$$

Now, if n is of the form $2m$, T_C produces $\alpha^{\frac{n}{2}}$ from α^n:

$$\begin{array}{lcl} \alpha^n & \xrightarrow{\circ} & (cy)^{\frac{n}{2}} \\ (cy)^{\frac{n}{2}} & \xrightarrow{\circ} & \alpha^{\frac{n}{2}} \end{array}$$

If n is of the form $2m+1$, T_C produces $\alpha^{3(\frac{n-1}{2})+2}$ $(= \alpha^{3m+2})$ from α^n:

$$\begin{array}{lcl} \alpha^n & \xrightarrow{\circ} & y(cy)^{\frac{n-1}{2}} \\ y(cy)^{\frac{n-1}{2}} & \xrightarrow{\circ} & \alpha^{3(\frac{n-1}{2})+2} \end{array}$$

This encoding allows for efficient simulation of $C(n)$ for any n. If n is even, C_T needs n iterations, with n uneven, $n+1$, to simulate one iteration of $C(n)$. The reason for the simplicity of this encoding is that $C(n)$ relies on modulo operations, while tag systems themselves can be regarded as some kind of modulo systems. Indeed, the encoding is based on this one feature of tag systems. Consider a string A of length $|A|$, and let $A \xrightarrow{\circ} B$. Clearly, the length of B depends on $|A| \bmod v$, in that the "original" length of B (the addition of the lengths of

the words produced from A) will be decreased with the additive complement of $|A| \bmod v$ (the additive complement of $b \bmod v$ is defined as $-b \bmod v$ evaluated to its least positive remainder, 0 included) In this respect, $|A| \bmod v$ determines what sequence of letters in B will and will not be scanned by the tag system. This feature is not only basic to our encoding, but is also the main ingredient in Minsky's and Cocke's proof of the universality of tag systems with $v = 2$ (See Sec. 1.1). To return to our encoding of C in T_C, if $|\alpha^n|$ is even, $|\alpha^n| \overset{\circ}{\rightarrow} (cy)^{\frac{n}{2}}$, with $|(cy)^{\frac{n}{2}}| \bmod v = 0$, guaranteeing that only the letter c will be scanned in B. Similarly, since $|(cy)^{\frac{n}{2}}|$ is even, no letter from $\alpha^{\frac{n}{2}}$ will have been erased after all the letters of $|(cy)^{\frac{n}{2}}|$ have been erased. In case $|\alpha^n|$ is uneven, $|\alpha^n| \overset{\circ}{\rightarrow} B$, with $|B| \bmod v = 1$, the first leading c being erased when the last α in α^n has been scanned. As a result, the tag system will scan the sequence of letters y. Although, taking together all the y's results in $\alpha^{3(\frac{n-1}{2})+3}$, the oddness of $y(cy)^{\frac{n-1}{2}}$ guarantees that the leading α will be erased after the last y has been scanned, thus leading to the desired result.
It should be noted here that T_C satisfies the minimal condition discussed by Maslov (Sec. 1.1). Indeed, $l_{min} = v - 1$ and $l_{max} = v + 1$.
Furthermore, the problem to decide for any n, whether $C(n)$ will ever lead to 1 after a finite number of iterations, reduces to the question of whether T_C will ever produce α. In other words, the $3n + 1$-problem can be reduced to a reachability problem for T_C.

2.2 Generalization of the method to arbitrary Collatz-like functions

By generalizing and slightly changing the encoding from the previous section, we were able to prove the following theorem:

Theorem 2.2 *Given an arbitrary Collatz-like function $G(n)$, with modulus d. Then, there is always a tag system T_G with $v = d, \mu \leq 2d+3, \Sigma = \{h, \alpha, \alpha_0, \beta_0, \beta_1, ..., \beta_{d-1}, b_0, b_1, ..., b_{d-1}\}$ that simulates $G(n)$ for any n.*

Note that μ and v are completely determined by the modulus. The symbol h functions as a kind of halting symbol, used for those cases when $G(n)$, $n = dm + i$, $0 \leq i < d$, is undefined for i. It is also important to note that the encoding of the present section needs the extra symbols $\alpha_0, \beta_0, \beta_1, ..., \beta_{d-1}$.
Each iteration of G over a number n corresponds to the production of a string $\alpha_0\alpha^{G(n)}$ from a string $\alpha_0\alpha^n$. The production rules for α_0, α are:

$$\begin{array}{lcl} \alpha_0 & \rightarrow & \beta_{d-1}\beta_{d-2}...\beta_0 \\ \alpha & \rightarrow & b_{d-1}b_{d-2}...b_0 \end{array}$$

If $G(n)$ is defined, with $n = dm + i$, $0 \leq i < d$, the production rules for β_i and b_i are :

$$\begin{array}{lcl} \beta_i & \rightarrow & (\alpha)^j\alpha_0(\alpha)^{r_i} \\ b_i & \rightarrow & (\alpha)^{a_i} \end{array}$$

where j is the additive complement of $(i+1)$ relative to d [i.e. : $-(i+1) \bmod d$ evaluated to its least positive remainder], with $i = n \bmod d$.
If $G(n)$ is undefined, $n = dm + i$, $0 \leq i < d$, the production rules for β_i and b_i are:

$$\begin{aligned} \beta_i &\rightarrow h \\ b_i &\rightarrow h \end{aligned}$$

The production rule for h is:

$$h \rightarrow \epsilon$$

Now, applying the production rules of T_G to a given string $\alpha_0\alpha^n$, in case $G(n)$ is defined, we get:

$$\alpha_0\alpha^n \overset{\circ}{\rightarrow} \beta_i\beta_{i-1}...\beta_0(b_{d-1}b_{d-2}...b_0)^{\frac{n-i}{d}} \tag{1}$$

Note, that we again use the property, mentioned in Sec. 2.1, that the length of a string B produced from a string A, through $\overset{\circ}{\rightarrow}$, is completely determined through $|A| \bmod v$, i.e. if the additive complement c of $|A| \bmod v > 0$, then the first c letters of the first word(s) produced from A will be erased, when the last letter of A has been scanned. Note that it is because the number of letters erased is equal to c, that the order of the indices of the letters in the words produced from $\alpha_0, \alpha, \beta_i, b_i, 0 \leq i < d$ is reversed, thus being able to keep track of the remainder. Furthermore, by adding the extra symbol α_0, the rules assure that $b_{d-1}b_{d-2}...b_0$ will be repeated $m = \frac{n-i}{d}$ times.
After the application of one iteration on the string produced in (1), T_G produces:

$$b_ib_{i-1}...b_0(b_{d-1}b_{d-2}...b_0)^{\frac{x-i}{d}-1}(\alpha)^j\alpha_0(\alpha)^{r_i} \tag{2}$$

From (2), T_G produces

$$\underbrace{b_ib_{i-1}...b_0(\alpha)^j}_{d}\,\alpha_0(\alpha)^{a_i\left(\frac{n-i}{d}-1\right)+r_i} \tag{3}$$

after (n-i)/d - 1 iterations. As is clear, the symbol β_i produced through α_0 is used to assure the tag system will start scanning α_0 after one iteration of G has been completed, through the addition of j times α, since

$$i + 1 + j = d.$$

Furthermore, β_i is used to add r_i if $G(n)$ is defined and $r_i > 0$. The letter b_i is used to perform the multiplication of m with a_i, since b_i is repeated $m = (n-i)/d$ times.
From (3) T_G finally produces:

$$\alpha_0(\alpha)^{a_i\frac{n-i}{d}+r_i} \tag{4}$$

after one more iteration.
If we apply the production rules to a string $\alpha_0\alpha^n$, in the case $G(n)$ is undefined, the production given in (1) remains unchanged. Then

$$\beta_i\beta_{i-1}...\beta_0(b_{d-1}b_{d-2}...b_0)^{\frac{n-i}{d}} \overset{\circ}{\rightarrow} h^{\frac{n-1}{d}+1} \tag{5}$$

From (5) we finally get:

$$h^{\frac{n-1}{d}+1} \stackrel{\circ}{\rightarrow} \epsilon \tag{6}$$

leading the tag system to a halt.
As is clear, the encoding of Collatz-like functions into tag systems is very straightforward, the input n for G being directly encoded as a string of length $n+1$. As was the case for the reduction of the $3n+1$-problem, the simulation of Collatz-like functions is efficient, where one iteration of $G(n)$ maximally takes $2\big(\lfloor n/d \rfloor + 1\big)$ iterations in T_G.
Given the fact that any Turing machine can be reduced to a Collatz-like function, the reduction of the present section serves as another proof of the existence of a universal tag system. Furthermore, the unsolvable problem to determine whether a Collatz-like function G, given an integer n, will ever produce the number 1 after a finite number of steps, reduces to the reachability problem to determine for any tag system T_G whether it will ever produce the string $\alpha_0\alpha$.
In comparing the encoding of the present section with that from Sec. 2.1, it is clear that the encoding of the present section leads to the simulation of the $3n+1$-problem in a larger tag system, with $\mu = 6$. This is due to the use of the symbol α_0. One might thus wonder whether there is a condition under which a tag system T_G, encoding a function $G(n)$ using α_0, can be reduced to a smaller tag system T'_G, without α_0.[4] The following theorem gives such a condition as well as the production rules of T'_G, which are based on the encoding of the $3n+1$-problem from Sec. 2.1 in T_C.

Theorem 2.3 *Given a Collatz-like function $G(n)$ with modulus d, where for each n, $G(n)$ either undefined or equal to $\frac{a_i(n-i)}{d} + r_i$, $i = 0, 1, ..., d-1$. Then $G(n)$ can always be reduced to a tag system T'_G with $v = d, \mu \leq 2+d, \Sigma = \{h, \alpha, b_0, b_1, ..., b_{d-1}\}$ iff. for every i defined, $\bar{i} < a_i$, if $i > 0$, $r_i = a_i - \bar{i}$, if $i = 0, r_i = 0$, where $\bar{i}$ is the additive complement of i. For each i defined, the production rules of T'_G are: $\alpha \rightarrow b_0 b_{d-1}..b_2 b_1; b_i \rightarrow \alpha^{a_i}$. For i undefined, the production rules are $b_i \rightarrow h$; $h \rightarrow \epsilon$*

The details of the proof are left to the reader.

3 Collatz-like functions and limits of solvability.

It is generally known that in order to prove the halting problem for a *particular* Turing machine is solvable or unsolvable, one must either find a (finite) method that proves it solvable, or else prove that it is universal. The same goes for any other class of systems equivalent to Turing machines, such as tag systems.
It is equally known that there still exists a rather huge gap between the known limits of solvability and unsolvability for Turing machines, and there are thus some classes of Turing machines for which it is not known whether they are solvable or not. Furthermore, for many of these machines it is by no means clear whether they are solvable – they show intractable behaviour – nor is there

[4] I am indebted to Pascal Michel for pointing out this problem to me.

any clear method to prove them universal. Indeed, as is noted by Neary and Woods ([30], p. 29):

> At present it seems technically challenging to further reduce the size of our machines so we suspect that a radically different approach is required.

An example that might show the way to such methods, is given by the proof of the universality of rule 110, and the universality of some rather small Turing machines by Matthew Cook (based on the proof of the universality of rule 110). Although there are clear problems involved with the notion of universality used by Cook (See 1.2), it is clear that the proofs heavily rely on a detailed analysis of the behaviour of rule 110. Closely connected to the intricate question of closing the gap between the known limits of solvability and unsolvability, as compared to the known ones, is the problem to prove a *specific* known Turing machine unsolvable independent of its universality. However, here one is confronted with the similar problem of finding a method to prove this.
In [23], Pascal Michel proved that the $3n + 1$-problem is reducible to the class TM(6, 3) of Turing machines. In [20], Margenstern proved that it can be reduced to the classes TM(11,2), TM(5,3), TM(4,4), TM(3,6) and TM(2,10) and calls the set of these machines the present $3n + 1$-line, to be situated between the present solvability and universality line. Margenstern furthermore mentions that Baiocchi has further improved this result, through reduction to the classes TM(10,2), TM(3, 5), and TM(2, 8). Michel also proved that the halting problem for some other Turing machine classes depend on the decision problem of some other Collatz-like functions. He proved this for the classes TM(5,2) [23], TM(2,4), TM(3,3) and TM(5,2) [24] and calls this set of machines the present Collatz-like line, situated between the present $3n+1$-line and solvability line. In Fig. 1 a summary is given of the known limits of solvability and unsolvability in Turing machines, including the $3n + 1$-line. As was said, the $3n + 1$-problem is known as an intractable problem. Proving the reducibility of the $3n+1$-problem to classes of machines considerably smaller than the known universal ones, can then serve as an indication of the difficulties that might be involved in proving machines from this class solvable. Based on these results, Margenstern [20] conjectures that all points on the $3n+1$-line contain a machine with an undecidable halting problem or an undecidable reachability problem or an undecidable modified reachability problem (a conjecture that assumes of course nothing about the status of the $3n + 1$-problem). One of the important questions to be asked here is what methods would be needed here to prove these classes of Turing machines unsolvable.
In drawing from this research, the reduction of the $3n + 1$-problem to a tag systems with $\mu = 3$, $v = 2$, implies that proving the solvability of this class of tag systems will be very hard. This is further strengthened by the fact that even the tag system mentioned by Post, with $\mu = 2$, $v = 3$, is still not known to be solvable. It should be noted here that we have studied the general class of tag systems with $\mu = 2, v > 3$, including Post's tag system. Many of the tag systems investigated give rise to the same kind of intractability one is confronted

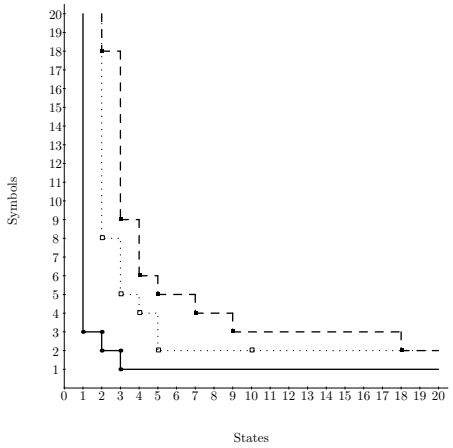

Figure 1: Limits of solvability and unsolvability in Turing machines. A full line denotes the solvability line, the dotted line the current $3n + 1$-line, and the dashed line is the current unsolvability line.

with in studying Post's tag system, there being no clear reason to consider them solvable nor a method to prove them universal.
Given the intractability of the class of tag systems for which $\mu = 2, v > 2$, and the reducibility of the $3n + 1$ problem to a tag system with $\mu = 3$, $v = 2$, we would like to propose the following conjecture:

Conjecture 1 *There exists at least one unsolvable tag system in every set of tag systems for which* $\mu = 2, v > 2$ *or* $\mu > 2, v = 2$

If this conjecture could be proven to be true, the gap between the known limits of solvability and unsolvability in tag systems would be closed. Fig. 2 gives an overview of the present situation of the limits of solvability and unsolvability in tag systems. Note that that shortest universal tag system known, depends on the smallest known two-symbolic universal Turing machine TM(18,2), since it is constructed by using the encoding by Cocke and Minsky from Turing machines into tag systems [1], [26]. In general, given this encoding, any two-symbolic Turing machine with m states, can be reduced to a tag system with $v = 2, \mu = 16m$ that simulates it.
In comparing the $3n + 1$-line for Turing machines and tag systems it is clear that T_C is considerably smaller than the size of the known Turing machines to which the $3n+1$-problem can be reduced. Furthermore, whereas the class of tag systems TS(3, 2), contains T_C, the class of Turing machines TM(3, 2) is known to be solvable. Given this result, together with the known intractability of the class of tag systems TS(2,3) (see Post's tag system), one is led to the conclusion that the limits of unsolvability in tag systems are indeed considerably lower as compared to those in Turing machines. This is indeed what we suspected given the fact that tag systems lie at the basis of the known small universal machines.

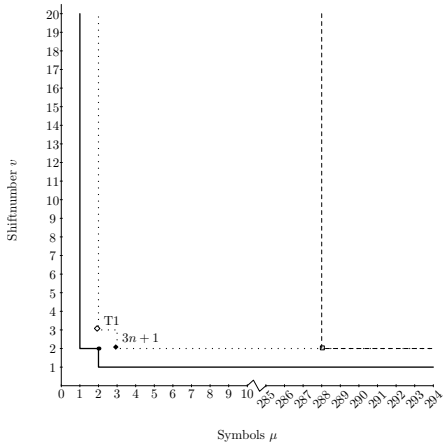

Figure 2: Limits of solvability and unsolvability in Tag systems. The full line indicates the solvability line, the dotted line is the conjectured unsolvability line, the dashed line is the known unsolvability line.

Concluding this section, it is suggested here that to further explore the limits of solvability and unsolvability – possibly independent of the universality line – tag systems might offer a valuable complementary framework. Especially in connecting them to known intractable problems from domains further removed from mathematical logic, further research on these systems seems interesting.

References

[1] John Cocke and Marvin Minsky, *Universality of tag systems with $p = 2$*, 1963, Artificial Intelligence Project – RLE and MIT Computation Center, memo 52.

[2] John H. Conway, *Unpredictable iterations*, Proceedings of the 1972 Number Theory conference.

[3] ———, *Fractran- a simple universal computing language for arithmetic*, Open Problems in Communication and Computation (New York) (T.M. Coper and B. Gopinath, eds.), Springer Verlag, 1987, pp. 3–27.

[4] Matthew Cook, *Universality in elementary cellular automata*, Complex Systems **15** (2004), no. 1, 1–40.

[5] Martin Davis, *A note on universal Turing machines*, [22], 1956, pp. 167–177.

[6] ———, *The undecidable. Basic papers on undecidable propositions, unsolvable problems and computable functions*, Raven Press, New York, 1965, Corrected republication (2004), Dover publications, New York.

[7] _____, *Why Gödel didn't have Church's thesis*, Information and Control **54** (1982), 3–24.

[8] _____, *Emil L. Post. His life and work.*, Solvability, Provability, Definability: The collected works of Emil L. Post [39], 1994, pp. xi–xviii.

[9] Craig A. Feinstein, *The Collatz 3n+1 conjecture is unprovable*, e-print, available at: http://arxiv.org/PS-cache/math/pdf/0312/0312309.pdf.

[10] Jeremy Fox (ed.), *Mathematical theory of automata*, Microwave Research Institute Symposia Series, vol. XII, Brooklyn, NY, Polytechnic Press, 1963.

[11] Brian Hayes, *Theory and practice: Tag-you're it*, Computer Language **XX** (1986), 21–28.

[12] _____, *A question of numbers*, American Scientist **84** (1996a), 10–14, Available at: http://oldweb.cecm.sfu.ca/news/clippings/96-01-0a/index.html.

[13] _____, *Latest results in the search for tag-system periods*, 1996b, Available at http://oldweb.cecm.sfu.ca/news/clippings/96-01-0a/update.html.

[14] F. Kaščák, *Small universal one-state linear operator algorithm*, Mathematical foundations of Comuter Science (L.M. Havel and V. Koubek, eds.), Lecture notes in Computer Science, vol. 629, 1992, pp. 327–335.

[15] M. Kudlek and Yurii Rogozhin, *New small universal circular post machines*, Fundamentals of Computation Theory : 13th International Symposium, FCT 2001, Riga, Latvia, August 22-24, 2001. (XX, ed.), Lecture notes in computer science, vol. 2138, 2001, pp. 217–226.

[16] _____, *A universal turing machine with 3 states and 9 symbols*, Proc. 5th International Conference on Developments in Language Theory (G. Rozenberg W. Kuich and A. Salomaa, eds.), Lectore Notes in Computer Science, vol. 2295, 2002, pp. 311–318.

[17] Jeffrey C. Lagarias, *The 3x + 1 problem and its generalizations*, American Mathematical Monthly **92** (1985), no. XX, 3–23, Available at: http://www.cecm.sfu.ca/organics/papers/lagarias/paper/html/paper.html.

[18] _____, *The 3x+1 problem and its generalisations*, Organic Mathematics. Proceedings Workshop Simon Fraser University, Burnaby (Providence) (J. Borwein et al., ed.), AMS, 1995, Available at http://www.cecm.sfu.ca/organics/papers/lagarias.

[19] _____, *The 3x + 1 problem: An annotated bibliography (1963–2000)*, 2006, Available at: http://arxiv.org/PS-cache/math/pdf/0608/0608208.pdf.

[20] Maurice Margenstern, *Frontier between decidability and undecidability: A survey*, Theoretical Computer Science **231** (2000), no. 2, 217–251.

[21] Sergei. J. Maslov, *On E. L. Post's 'Tag' problem. (russian)*, Trudy Matematicheskogo Instituta imeni V.A. Steklova (1964b), no. 72, 5–56, XX:trans.

[22] John McCarthy and Claude E. Shannon (editors) (eds.), *Automata studies*, Annals of Mathematics Studies, no. 34, Princeton University Press, Princeton, 1956, Second Printing 1958.

[23] Pascal Michel, *Busy Beaver competition and Collatz-like problems*, Archive for Matheamtical Logic **32** (1993), no. 5, 351–367.

[24] _____, *Small Turing machines and generalized Busy Beaver competition*, Theoretical Computer Science **326** (2004), no. 1–3, 45–56.

[25] Marvin Minsky, *Recursive unsolvability of Post's problem of tag and other topics in the theory of Turing machines*, Annals of Mathematics **74** (1961), 437–455.

[26] _____, *Universality of (p = 2) tag systems and a 4 symbol 7 state universal Turing machine*, 1961/62?, Artificial Intelligence Project – RLE and MIT Computation Center, memo 33.

[27] _____, *Size and structure of universal Turing machines using tag systems: a 4-symbol 7-state machine*, Proceedings Symposia in Pure Mathematics, American Mathematical Society **5** (1962), 229–238.

[28] _____, *Computation. Finite and infinite machines*, Series in Automatic Computation, Prentice Hall, Englewood Cliffs, New Jersey, 1967.

[29] Liesbeth De Mol, *Closing the circle: An analysis of Emil Post's early work.*, The Bulletin of Symbolic Logic **12**, no. 2, 267–289.

[30] Turlough Neary and Damien Woods, *Small fast universal Turing machines*, Technical report NUIM-CS-2005-TR-11, Department of Computer Science, NUI Maynooth, 2005a, Accepted for publication in Theoretical Computer Science.

[31] _____, *On the time complexity of 2-tag systems and small universal Turing machines*, 47th Annual IEEE Symposium on Foundations of Computer Science, 2006, accepted.

[32] _____, *P-completeness of cellular automaton rule 110*, International Colloquium on Automata Languages and Programming (ICALP), Lecture Notes in Computer Science, vol. 4051, 2006, pp. 132–143.

[33] _____, *Remarks on the computational complexity of small universal Turing machines*, Proceedings of MFCSIT 2006 (Cork, Ireland), 2006, pp. 334–338.

[34] D. Pager, *The categorization of tag systems in terms of decidability*, Journal of the London Mathematical Society **2** (1970), no. 2, 473–480.

[35] L. Pavlotskaya, *Solvability of the halting problem for certain classes of Turing machines.*, Mathematical Notes Academy of Science USSR, **13** (1973), no. 6, 537541,.

[36] _____, *Sufficient conditions for the halting problem decidability of Turing machines*, Avtomaty i Mashiny **XX** (1978), no. XX, 91–118.

[37] Emil Leon Post, *Formal reductions of the general combinatorial decision problem*, American Journal of Mathematics (1943), no. 65, 197–215.

[38] _____, *Absolutely unsolvable problems and relatively undecidable propositions - Account of an anticipation*, [6], 1965, pp. 340–433.

[39] _____, *Solvability, provability, definability: The collected works of Emil L. Post*, Birkhauser, Boston, 1994, edited by Martin Davis.

[40] Tibor Rádo, *On non-computable functions*, The Bell System Technical Journal **41** (1962), no. 3, 877–884.

[41] Yurii Rogozhin, *Small universal Turing machines*, Theoretical Computer Science **168** (1996), 215–240.

[42] Yurii Rogozhin, *Seven universal Turing machines (in Russian)*, Mat. Issledovaniya **69** (1982), 76–90.

[43] Hao Wang, *Tag systems and Lag systems*, Mathematische Annalen **152** (1963a), 65–74.

[44] Shigeru Watanabe, *Periodicity of Post's normal process of tag*, [10], 1963, pp. 83–99.