Study of Limits of Solvability in Tag Systems

Liesbeth De Mol

Center for Logic and Philosophy of Science
University of Ghent, Blandijnberg 2, 9000 Gent, Belgium
elizabeth.demol@ugent.be

Abstract. In this paper we will give an outline of the proof of the solvability of the *halting* and *reachability problem* for 2-symbolic tag systems with a deletion number $v = 2$. This result will be situated in a more general context of research on limits of solvability in tag systems.

Key words: Tag Systems, Limits of solvability, Reachability problem, Halting Problem

1 Introduction

Tag systems were invented by Emil Post in 1921 [15],[16] and played an important role in Post's earlier work on normal systems. After 9 months of research on tag systems, he came to the conclusion that proving the Entscheidungsproblem for first-order predicate calculus solvable might be impossible. He never proved the unsolvability of this Entscheidungsproblem, but was able to show that there are certain unsolvable decision problems for normal systems [4],[5], [10], [18].
Although Post mentions some results on tag systems [15], [16] he never published any of the proofs. He considered two different decision problems for tag systems: the *halting problem* and the *reachability problem.*

Definition 1 *The halting problem for tag systems is the problem to determine for a given tag system and any initial string A_0 whether the tag system will halt.*

Definition 2 *The reachability problem for tag systems is the problem to determine for a given tag system T, a fixed initial string A_0 and any arbitrary string A over the alphabet Σ, whether T will ever produce A when started with A_0.*

Note that the halting problem for tag systems is in fact just a special case of the reachability problem. Post mentioned that he was able to prove the solvability of both decision problems for a specific class of tag systems, a result he considered as the major success of his research on tag systems. The main purpose of this paper is to give an outline of a proof of this result.

1.1 Definition of Tag Systems and Notational Conventions

A tag system T, consists of a finite alphabet $\Sigma = \{a_0, a_1, ..., a_{\mu-1}\}$ of μ symbols, a deletion number $v \in \mathbb{N}$ and a finite set of μ words, $w_0, w_1, ..., w_{\mu-1}$ defined over the alphabet, including the empty word ϵ. Each of these words corresponds with one of the letters from the alphabet as follows:

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

where each $a_{i,j} \in \Sigma, 0 \leq i < \mu$. Given an initial string A_0, the tag system *first* tags the word associated with the leftmost letter of A_0 at the end of A_0, and then deletes the first v symbols of A_0.[1] This process is iterated until the tag system halts, i.e. produces the empty string ϵ. If this does not happen the tag system can become periodic or show divergent behaviour. Post mentions one example of a tag system with $v = 3$, $0 \rightarrow 00$, $1 \rightarrow 1101$ [15]. It is still not known whether this particular tag system is recursively solvable, despite its formal simplicity.
Let T be a tag system with a deletion number v with μ symbols and words $w_0, w_1, ..., w_{\mu-1}$. We shall write l_{w_i} to indicate the length of a word w_i, $l_{\mathbf{max}}$ and $l_{\mathbf{min}}$ denote the length of the lengthiest word w_i rsp. the length of the shortest word w_j of T, $0 \leq i, j < \mu$. The total sum of the number of a_i's in the words $w_0, ...w_{\mu-1}$ will be denoted as $\#a_i$. We will use $\dot{x}$ rsp. x to indicate an odd rsp. an even number. Given a string $A = a_1a_2...a_{l_A}$, we will say that A is entered with shift x, when the tag system erases its first x symbols, the first symbol scanned in A being a_{x+1}.

1.2 Results on the Limits of Solvability in Tag Systems

Post never proved that tag systems are recursively unsolvable. It was Minsky who proved the result in 1961 [8], after the problem was suggested to him by Martin Davis. He showed that any Turing machine can be reduced to a tag system with $v = 6$. The result was improved by Cocke and Minsky [2], [9]. They proved that any Turing machine can be reduced to a tag system with $v = 2$. The same result was proven by Wang [19]. 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 [7].
The result from [2], [9] can be used to determine the size of the smallest universal tag system known. If we define the size of a tag system T as the product of μ and v, it is possible to reduce any 2-symbolic Turing machine with m states to a tag system with $v = 2$, $\mu = 16m$. Using the universal Turing machine constructed by Neary in the class TM(18, 2) [14] or the machine constructed by Baiocchi which is in the class TM(19,2) [1], where TM(m, n) denotes the class of Turing

[1] It should be noted that we follow Post's original definition of tag systems, instead of the one that is now commonly used. I.e. in the definition used here, the tagging and deletion operation are not performed in one and the same step. The proof of the main theorem only needs some minor changes in order to be applicable to a definition of tag systems where tagging and deletion operation are performed synchronously.

machines with m states and n symbols, it is possible to construct universal tag systems in the classes TS(288, 2) rsp. TS(304,2), TS(m, v) denoting the class of tag systems with m symbols and a deletion number v.
Despite the relatively large size of the smallest universal tag systems known, there are some clear indications that proving very small classes of tag systems solvable will be very hard, if not impossible. The fact that the tag system mentioned above from the class TS(2, 3) is still not known to be solvable serves as an indication of this problem. A further indication is given by the result from [12], where it is shown that the $3n+1$-problem can be reduced to a tag system from the class TS(3, 2), i.e., $w_0 \to w_1w_2, w_1 \to w_0, w_2 \to w_0w_0w_0$. The reduction of the $3n+1$-problem, which is known as a hard problem of number theory, to a very small tag system, illustrates how hard it might be to prove this class of tag systems solvable.
Post mentions that the halting and reachability problem for the class of tag systems for which $v = 1$ or $\mu = 1$ is trivially solvable and remarks that he completely solved the case $\mu = v = 2$ [16]. But, as was said, the proofs were never published. The case $\mu = 1$ is indeed trivially solvable. Wang [19] provided the proof for the case $v = 1$. In this paper we will outline the proof for the class $\mu = v = 2$.
Both μ and v can be regarded as *decidability criteria* [6] for tag systems, since their solvability depends on the size of these parameters. Another such criterion is the length of the words. Wang proved that any tag system for which $l_{\mathbf{min}} \geq v$ or $l_{\mathbf{max}} \leq v$ is solvable [19].

2 Solvability of the Halting and Reachability Problem of the Class TS(2,2)

In [16] Post remarks that his proof of the solvability of the halting and reachability problem of the class TS(2, 2) involved "*considerable labor*". This is indeed true, as will become clear from the outline of the proof we will give here. The proof involves a study of a rather large number of subcases. We will merely outline the structure of the proof and restrict ourselves to detailed proofs for only some of the subcases, because of the length of the actual proof, and the fact that several subcases can be solved by using similar methods. A detailed proof is made available on-line [11].
Post differentiates between three different classes of behaviour a tag system can converge to, i.e., a tag system can halt, it can become periodic, or it can show unbounded growth. The reachability and halting problem can be proven solvable, if one can determine for any initial condition, for a given tag system, that it will lead to one of these three classes of behaviour after a finite number of steps. In case of unbounded growth, one should be able to prove that for any given number n, the tag system will always produce a string A_i of length $l_{A_i} > n$ after a finite number of iterations i, such that no string $A_j, j > i$, will ever be produced again for which $l_{A_j} \leq n$.
In our proof, we have indeed been able to show that one can determine for any

tag system T from the class TS(2, 2) and any initial condition over the alphabet $\Sigma = \{0, 1\}$, that T will always become periodic, halt or show unbounded growth after a finite number of steps. We have thus been able to prove the following theorem:

Theorem 1 *For any given tag system T, if $\mu = v = 2$ then the halting problem and the reachability problem for T are solvable.*

First of all, it should be noted that we only have to consider those cases with $l_{\mathbf{min}} < 2, l_{\mathbf{max}} > 2$, given the theorem proven by Wang mentioned in Sec. 1.2. In the remainder, we assume that $l_{\mathbf{max}} = l_{w_1}, l_{\mathbf{min}} = l_{w_0}$, the symmetrical case of course being equivalent to this case.
There are three global cases to be taken into account, i.e., $w_0 = \epsilon, w_0 = 1, w_0 = 0$. Each of these cases is subdivided into several subcases, determined by the following parameters: l_{w_1}, the parity of l_{w_1},[2] #1, and the parity of the number of 0's separating consecutive 1's in w_1. It should be noted that, contrary to classes of Turing machines TM(m, n), the three global cases to be taken into account contain an infinite number of tag systems. In this sense it has been basic for the proof to determine certain threshold values for two of these parameters, i.e., l_{w_1} and #1. If the values of these parameters exceed a given value, the infinite class of tag systems determined by the parameters will always show unbounded growth (except for a specific class of initial strings), else they will halt or become periodic (except for a specific class of initial strings).
There is one method that has been basic to solve the majority of cases, called the *table method*. What one basically does with this method is to look at a certain number of substrings that can be produced theoretically in a given tag system, by starting from the possible productions from the respective words $w_0, ..., w_{\mu-1}$. Given a tag system T with a deletion number v, it is clear that given a word $w_i = a_{i,1}a_{i,2}...a_{i,l_{w_i}}$, some letters in w_i will be 'scanned', others not. The sequence of letters that is scanned is determined by the number n, $0 \leq n \leq v - 1$, of leading letters of w_i that is erased but not scanned by the tag system and which leads to the concatenation or tagging of the words corresponding to the letters from the sequence at the tail of a given string. For example, if $v = 3$, there are three different sequences of letters in w_i that might be scanned by the tag system: $a_{i,1}a_{i,4}...a_{i,t_0}$, $a_{i,2}a_{i,5}...a_{i,t_1}$, $a_{i,3}a_{i,6}...a_{i,t_2}$, with:

$$t_j = l_{w_i} - [(l_{w_i} - j) \bmod 3]$$

Now, given a tag system T, with deletion number v and μ letters. The table method is applied to the tag system by first looking at all the possible strings v that can be produced from each of the words w_i, $0 \leq i < \mu$, by concatenating the words corresponding to the letters of each of the different sequences in each of the w_i, determined as above. If one of these new strings is equal to one of the words w_i it is marked. If all the strings produced in this way are marked or equal to ϵ it follows that the tag system will always halt or become periodic,

[2] The parity of a number x is the property of it being even or odd.

since the length of the strings that can be produced from the respective words is bounded. If this is not the case, the same procedure is applied to all the strings left unmarked and not equal to ϵ,...If we e.g. apply this method to the two words 00 and 1101 of the tag system mentioned above (Sec. 1.1), only one (11011101) of the 6 possible strings produced will be left unmarked, and differs from ϵ. If we apply the method to this one string it becomes clear very soon that the method will never come to a halt, i.e., there will always remain strings left unmarked.
As will become clear in the proof, the table method is not only useful if, for a given tag system, all the strings become marked or are equal to ϵ at a given time, but can also be used to e.g. prove that a tag system will either halt or show unbounded growth. In general, it should be noted here that, although this method is very simple, it is an important instrument to study tag systems.
The method is called the table method, because the results from the method can best be represented through tables. We will explain how such a table should be read, in the first application of the method in the proof.
Note that from now on, l_{w_0} and l_{w_1} will be abbreviated as l_0 rsp. l_1.

Proof.
Case 1. $w_0 = \epsilon$

Case 1.1. $\#1 = 0$. It is trivial to prove that tag systems from this class will always halt, since only 0's can be scanned.

Case 1.2. $\#1 = 1$, $l_1 \equiv 0 \bmod 2$. Let $w_1 = 0^{x_1}10^{y_1}$. To prove this case, we need the table method mentioned above. The following table proves the case:

	w_0	w_1
S_0	ϵ	ϵ
S_1	ϵ	$w_1 \checkmark$

The row headed with S_0 gives the string produced from a given string S (in this case w_0 or w_1) when the first letter of the string S is scanned by the tag system. Similarly, the row headed S_1 gives the possible productions from a given string S when its first letter is erased without being scanned. Clearly, since in this case $w_0 = \epsilon$ actually no letter in w_0 can be scanned or erased. The only possible non-empty string that can be produced for this case, is the string resulting from w_1 when entered with a shift 1, i.e., its first letter is erased but not scanned.
As is clear from the table, a tag system from this class will either halt or become periodic. It will become periodic when at least one 1 is scanned in the initial condition, such that the first letter from the word w_1 thus produced, will not be scanned. This is determined by the parity of the length of the initial condition. In all other cases, tag systems from this class halt. A similar proof can be given for the case $w_1 = 0^{x_1}10^{y_1}$.

Case 1.3. $\#1 = 1$, $l_1 \equiv 1$ mod 2. The table that can be constructed for this class of tag systems, is identical to that of the previous case, with $w_1 = 0^{\dot{x}_1}10^{\dot{y}_1}$. Despite the table being identical, tag systems from this class can be proven to always halt. The reason for this is that the number of surviving 1's is at most half of what it was when the tag system has scanned (and erased) all the letters of the current word at a given stage of the computation. The reader is referred to the on-line details of the proof.

Case 1.4. $\#1 = 2$, $l_1 \equiv 0$ mod 2. To prove the case we have to differentiate between two subcases, i.e. $w_1 = 0^{x_1}10^{y_1}10^{z_1}$ and $w_1 = 0^{\dot{x}_1}10^{\dot{y}_1}10^{z_1}$ (the proof for $w_1 = 0^{\dot{x}_1}10^{y_1}10^{\dot{z}_1}$ is similar to the first case, the proof of $w_1 = 0^{x_1}10^{\dot{y}_1}10^{\dot{z}_1}$ is similar to the second case).

Case 1.4.1. $w_1 = 0^{x_1}10^{y_1}10^{z_1}$. The first case is proven through the following table:

Table 2: $w_1 = 0^{x_1}10^{y_1}10^{z_1}$

	w_1
S_0	w_1 ✓
S_1	w_1 ✓

From this table it follows that any tag system from this class will always become periodic, except when no 1 is scanned in the initial condition, then it always halts.

Case 1.4.2. $w_1 = 0^{\dot{x}_1}10^{\dot{y}_1}10^{z_1}$. The case is proven through the following table:

Table 3: $w_1 = 0^{\dot{x}_1}10^{\dot{y}_1}10^{z_1}$

	w_1	w_1w_1	...	$(w_1w_1)^n$
S_0	ϵ	ϵ	...	ϵ
S_1	w_1w_1	$w_1w_1w_1w_1$	....	$(w_1w_1)^{2n}$

As is clear from the table, tag systems from this class will either halt or show unbounded growth depending on the parity of the length of the initial condition.

Case 1.5. $\#1 = 2$, $l_1 \equiv 1$ mod 2. The proof is almost identical to that of case 1.4., except that now we have to consider the cases $w_1 = 0^{x_1}10^{\dot{y}_1}10^{z_1}$ (or similarly $w_1 = 0^{\dot{x}_1}10^{\dot{y}_1}10^{\dot{z}_1}$) and $w_1 = 0^{\dot{x}_1}10^{y_1}10^{z_1}$ (or similarly $w_1 = 0^{x_1}10^{y_1}10^{\dot{z}_1}$).

Case 1.6. $\#1 = 3$, $l_1 \equiv 0 \bmod 2$. Again we have to consider several cases, depending on the parity of the number of 0's separating pairs of symbols 1 in w_1. The proofs of the several cases are similar to those for Case 1.4. Tag systems from this class will always show unbounded growth, halt or become periodic, depending on the parity of the initial condition and the spacings between the several 1's.

Case 1.7. $\#1 = 3$, $l_1 \equiv 1 \bmod 2$. We have to differentiate between two cases: $w_1 = 0^{s_1}10^{x_1}10^{y_1}10^{t_1}$ (and all variants) or $w_1 = 0^{s_1}10^{\dot{x}_1}10^{\dot{y}_1}10^{t_1}$ (and all variants). It can be proven that tag systems from both classes will either halt (if no 1 is scanned in the string produced from the initial condition) or show unbounded growth after a finite number of steps, by applying the table method. See [11] for the detailed table.

Case 1.8. $\#1 > 3$, $l_1 \equiv 0 \bmod 2$ and Case 1.9. $\#1 > 3$, $l_1 \equiv 1 \bmod 2$. For any tag system from these classes and any initial condition it can be determined that it will either halt, become periodic, or show unbounded growth after a finite number of iterations. The result follows from the proofs of cases 1.4.–1.7.

Case 2. $w_0 = 1$

Case 2.1. $\#1 = 1$ In this case the length of w_1 is a determining factor to predict the behaviour of a tag system from this class, since w_1 only consists of 0's. We have to differentiate between the following two cases: $2 < l_1 < 5$ or $5 \leq l_1$.

Case 2.1.1. $2 < l_1 < 5$ Tag systems from this class will always become periodic, except when the initial condition is equal to 0, then it will halt. The result can be proven through the table method, the details of the proof can be found on-line.

Case 2.1.2. $5 \leq l_1$ It can be easily checked that tag systems from this class will always show unbounded growth, except for a finite class of initial conditions, for which the tag systems will halt or become periodic after a finite number of steps. The proof follows from the fact that once a tag system from this class produces a string that consists of at least two times w_1 it will show unbounded growth. The proof follows from the table method (see the on-line proof).

Case 2.2. $\#1 = 2, l_1 = 3$. It can be determined for any tag system from this class that it will either halt or become periodic. There are three different tag systems to be taken into account: $0 \to 1; 1 \to 100$, $0 \to 1; 1 \to 010$, and $0 \to 1; 1 \to 001$. The result can be proven for each of the cases by applying the table method. (see [11] for detailed tables.)

Case 2.3. $\#1 = 2, l_1 > 3$. It can be determined for any tag system from this class that it will either halt, become periodic or show unbounded growth. To prove the result, we have to differentiate between $l_1 = 4$ and $l_1 > 4$

Case 2.3.1. $l_1 = 4$ The result of the case follows from a rather complicated application of the table method. The table is about half a page long and needs some further deductions. The details of the proof can be found on-line.

Case 2.3.2. $l_1 > 4$ For the second case it can be shown rather easily that once $w_1 w_1$ is produced as a substring, tag systems from this class will always lead to unbounded growth. For more details the reader is referred to the on-line proof.

Case 2.4. $\#1 > 2$. Each tag system from this class will either halt, become periodic, or show unbounded growth. The proof differentiates between two subcases $l_1 = 3$ and $l_1 > 4$. The first case involves a more complicated application of the table method. The proof of the second case is rather straightforward and follows from case 2.3.2. See the on-line proof for more details.

Case 3. $w_0 = 0$

Case 3.1. $\#1 = 0$, $l_1 > 2$. It is trivial to prove that any tag system from this class will halt, since any sequence of 0's always leads to ϵ.

Case 3.2. $\#1 = 1$, $l_1 > 2$. It can be determined that any tag system from this class will either halt or become periodic by applying the table method (see the on-line proof).

Case 3.3. $\#1 = 2$, $l_1 > 2$, $l_1 \equiv 0 \bmod 2$. It can be determined for any tag system from this class that it will either halt, become periodic or show unbounded growth after a finite number of iterations. From now on, we will write x instead of 0^x for the ease of notation. We have to take into account two cases. The 1's can be separated by an even or an odd number of 0's, i.e., $w_1 = t_1 1 x_1 1 s_1$ (or similarly $w_1 = \dot{t_1} 1 x_1 1 \dot{s_1}$), or $w_1 = t_1 1 \dot{x_1} 1 \dot{s_1}$ (or, similarly $w_1 = t_1 1 \dot{x_1} 1 \dot{s_1}$).

Case 3.3.1. $w_1 = t_1 1 x_1 1 s_1$ The proof of the first case results from the application of the table method. It proves that any tag system from this class will always become periodic after a finite number of steps for any initial condition except for those conditions in which no 1 is scanned by the tag system, then a halt occurs.

Case 3.3.2. $w_1 = t_1 1 \dot{x_1} 1 \dot{s_1}$ It can be shown that any tag system from this class will either halt, become periodic or show unbounded growth after a finite number of iterations. The proof of this case is more complicated, and we have to subdivide the case into two subcases: t_1, $\dot{x_1}$ or $\dot{s_1} > 1$ and $t_1 = 0$, $\dot{x_1} = 1$,

$\dot{s_1} = 1$.
Case 3.3.2.1. t_1, $\dot{x_1}$ *or* $\dot{s_1} > 1$ For any tag system from this class it can be determined that it will either halt, become periodic or show unbounded growth. Set $w_1 = t_1 1 \dot{x_1} 1 \dot{s_1}$. In shift 1, the tag system will produce a sequence of 0's from w_1, ultimately leading to a halt. In shift 0, we get:

$$A_1 = t_2 w_1 \lfloor \dot{x_1}/2 \rfloor w_1 s_2 \tag{1}$$

Depending on the shift, if $\dot{s_1} + \lfloor \dot{x_1}/2 \rfloor + t_1$ is even, we get:

$$t_3 A_1 0^{n_1} \tag{2}$$

or:

$$t_3 0^{n_1} A_1 \tag{3}$$

from (1). It thus follows that if $\dot{s_1} + \lfloor \dot{x_1}/2 \rfloor + t_1$ is even, and at least one w_1 is produced such that its first 1 will be scanned, the tag system will ultimately become periodic, since the lengths of the possible strings produced from w_1 in this case are bounded, but never produce the empty string. If $\dot{x_1} + \lfloor \dot{x_1}/2 \rfloor + t_1$ is odd, the tag system produces:

$$A_2 = t_4 A_1 \lfloor \dot{x_1}/4 \rfloor A_1 s_3 \tag{4}$$

from (1), or a string merely consisting of a certain number of 0's (ultimately converging to ϵ), depending on the shift. If $\dot{s_1} + s_2 + \lfloor \dot{x_1}/4 \rfloor + t_2 + t_1$ is even, we get:

$$t_5 A_2 0^{n_2} \tag{5}$$

or:

$$t_5 0^{n_2} A_2 \tag{6}$$

from A_2, again depending on the shift. Thus if $\dot{s_1} + s_2 + \lfloor \dot{x_1}/2 \rfloor + t_2 + t_1$ is even, the tag system will always halt or become periodic. A halt occurs, if no A_2 is produced. If $\dot{s_1} + s_2 + (\dot{x}_1 - 1)/4$ is odd, the tag system produces:

$$A_3 = t_6 A_2 \lfloor (\dot{x}_1)/8 \rfloor A_2 s_4 \tag{7}$$

from (4), or a sequence of 0's depending on the shift.
Generally, tag systems from this class will become periodic or halt once a sequence $\dot{s_1} + s_2 + s_3 + ... + s_n + \lfloor (\dot{x}_1)/2^n \rfloor + t_n + ... + t_2 + t_1$, separating two consecutive A_{n-1} in A_n ($n \in \mathbb{N}, A_0 = w_1$) becomes even. Indeed, given a string $A_n = t_i A_{n-1} \lfloor \dot{x_1}/2^n \rfloor A_{n-1} \dot{s_i}$, with $\dot{s_1} + s_2 + s_3 + ... + s_n + \lfloor \dot{x_1}/2^n \rfloor + t_n + ... + t_2 + t_1$ even, the tag system will produce either $t_i A_n 0^{n_j}$ or $t_i 0^{n_j} A_n$, with the number of 0's surrounding each A_n being bounded. If for a given tag system, there is no n such that the sequence $\dot{s_1} + s_2 + s_3 + ... + s_n + \lfloor (x_1/2^n \rfloor + t_n + ... + t_2 + t_1$ between a pair of A_{n-1} in A_n is even, the tag system will either halt or show unbounded growth.
Now, it can be easily determined (in a finite number of steps) for any tag system from this class whether there exists an n such that $\dot{s_1} + s_2 + s_3 + ... + s_n + \lfloor (x_1 -$

$1)/2^n\rfloor + t_n + ... + t_2 + t_1$ between a pair of A_{n-1} in A_n is even. This follows from the following lemma:[3]

Lemma 1 *For any tag system from the class 3.3.2.1. it can be proven that there is always an n, $n \in \mathbb{N}$ such that for any $i \geq n$ the sequence of 0's* $\dot{s_1} + s_2 + s_3 + ... + s_i + \lfloor (x_1 - 1)/2^i \rfloor + t_i + ... + t_2 + t_1$ *between a pair of A_{i-1} in A_i is of the same length as* $\dot{s_1} + s_2 + s_3 + ... + s_n + \lfloor (x_1 - 1)/2^n \rfloor + t_n + ... + t_2 + t_1$.

The proof of the lemma can be found in the on-line version. It follows from this lemma that one can determine for any tag system from this class whether a sequence of 0's separating two consecutive A_{i-1} in A_i will ever become even or not, since it only takes a finite number of steps before a sequence A_n is produced for which the number of 0's separating a pair of A_{n-1} becomes constant. We have thus proven the case: tag systems from this class will either halt, become periodic or show unbounded growth.
Case 3.3.2.2. $t_1 = 0$, $x_1 = 1$, $s_1 = 1$ It can be proven that the only tag system in this class, with $w_1 = 1010$, will either halt or show unbounded growth. The result can easily be obtained through the table method or by pure reasoning. See the on-line proof for more details.

Case 3.4. $\#1 = 2$, $l_1 > 2$, $l_1 \equiv 1 \bmod 2$. It can be determined for any tag system from this class that it will always halt or become periodic. Again we have to consider two cases, depending on the parity of the spacing between the two 1's, i.e. $w_1 = \dot{t_1}1x_11s_1s$ and $w_1 = t_11\dot{x_1}1s_1$. The proof of the first case is similar to the proof of case 3.3.1. For the second case, we have to differentiate between two subcases, i.e., t_1, x_1 or $s_1 > 1$ and $t_1 = 0$, $x_1 = 1$, $s_1 = 0$. The proof of the first subcase is almost identical to that of case 3.3.2.1., the second subcase easily follows by applying the table method. See the on-line proof for more details.

Case 3.5. $\#1 > 2$, $l_1 > 2$, $l_1 \equiv 0 \bmod 2$. It can be determined for each tag system from this class that it will either show unbounded growth, become periodic or halt after a finite number of iterations, depending on the initial condition. To prove this, we merely have to consider the case $\#1 = 3$, since the generalization to $\#1 > 3$ trivially follows from the proof of the case $\#1 = 3$. There are two possible subcases to be proven: either all 1's are separated by an odd number of 0's, or only one pair of 1's is separated by an odd number of 0's. The proofs of both cases use methods similar to those used for Case 3.3.2. Detailed proofs are available on-line.

Case 3.6. $\#1 > 2$, $l_1 > 2$, $l_1 \equiv 1 \bmod 2$. For any tag system from this class it can be determined that it will either halt, become periodic or show unbounded growth after a finite number of iterations, depending on the initial condition. The

[3] We are indebted to an anonymous referee for pointing out a serious error in a previous proof of this case concerning the number of 0's separating a pair of A_{n-1} and having provided us with the necessary lemma and an outline of its proof to solve the case.

proofs for the several subcases are similar to those for case 3.5.
The proof of theorem 1 follows from the proofs of cases 1–3. □

3 Discussion

As is clear from the outline of the main theorem of this paper, proving the solvability of the halting and reachability problem for the class TS(2, 2) indeed involves considerable labor. Most probably the proofs of some cases might be simplified. For example,the solvability of cases 1.2, 1.4, 1.6., 1.8. follows from the following theorem:

Theorem 2 *Given a tag system T with deletion number v, $\Sigma = \{a_0, a_1, ..., a_{\mu-1}\}$ and words $w_{a_0}, w_{a_1}, ..., w_{a_{\mu-1}}$. Then, if the lengths l of the words and v are not relative prime the solvability of a given decision problem for T can be reduced to the solvability of the decision problem for n different tag systems, λ being the greatest common divisor of v, $l_{w_{a_0}}, ..., l_{w_{a_{\mu-1}}}$, with deletion number $v' = v/\lambda$.*

which is proven in [13]. It follows from this theorem that the halting and reachability problem for all the tag systems with $w_0 = \epsilon, l_{w_1} \equiv 0 \bmod 2$ from the class TS(2, 2) can be reduced to the halting and reachability problem of tag systems with $v = 1$. Since Wang proved that these problems are solvable for any tag system with $v = 1$ (See Sec. 1.2), the result easily follows.
As was explained in Sec. 1.2, it might be very hard, if not impossible, to prove the solvability of those classes of tag systems that are closest to TS(2, 2), i.e., TS(2,3) and TS(3,2). In fact, as far as our experience goes with these classes of tag systems, they seem to be completely intractable. The methods used in the present proof do not work for these classes. The only reasonable explanation we have been capable to find for this basic difference is related to the balance between the total number $\#a_0, \#a_1, ..., \#a_{\mu-1}$ of each of the symbols $a_0, a_1, ..., a_{\mu-1}$ in the respective words for a given tag system. For each symbol a_i, we can measure the effect of scanning a_i on the length of a string produced in a tag system, i.e., it can lead to a decrease or an increase. This effect of scanning a symbol a_i on the length of a string produced, can be computed by taking the absolute value of $l_{w_{a_i}} - v$. If we then sum up the products $\#a_i \cdot (l_{w_{a_i}} - v)$ for each of the symbols, and the result is a negative rsp. a positive number, one might expect that the tag system will always halt rsp. show unbounded growth.
Although we have been able to show that this method cannot be used in general, it is clear that this method might be applied to certain infinite classes of tag systems to prove them solvable. Not taking into account the case with $w_0 = \epsilon$ it can be proven for the class TS(2, 2) that there is but a finite subclass of tag systems for which this sum is equal to 0.[4] This is in sharp contrast with the classes TS(2,3) and TS(3,2) for which it can be proven that they each contain an infinite class of such tag systems, even if no word is equal to ϵ. We consider this as a fundamental difference between the class TS(2,2) and the classes TS(3,2),

[4] These are tag systems with $l_0 = 1, l_1 = 3, \#1 = \#0 = 2$

TS(2,3). In fact, we suspect that further research on this method might help to considerably simplify the proof of Theorem 1. For more details on this issue the reader is referred to [13].

References

1. Claudio Baiocchi, *Three small universal Turing machines*, Machines, Computations and Universality, MCU 2001 (M. Margenstern and Yu. Rogozhin, eds.), Lecture Notes in Computer Science, vol. 2055, 2001, pp. 1–10.
2. John Cocke and Marvin Minsky, *Universality of tag systems with $p = 2$*, 1963, Artificial Intelligence Project – RLE and MIT Computation Center, memo 52.
3. Martin Davis, *The undecidable. Basic papers on undecidable propositions, unsolvable problems and computable functions*, Raven Press, New York, 1965, Corrected republication (2004), Dover publications, New York.
4. ———, *Why Gödel didn't have Church's thesis*, Information and Control **54** (1982), 3–24.
5. ———, *Emil L. Post. His life and work*, 1994, in: [17], xi–xviii.
6. Maurice Margenstern, *Frontier between decidability and undecidability: A survey*, Theoretical Computer Science **231** (2000), no. 2, 217–251.
7. Sergei. J. Maslov, *On E. L. Post's 'Tag' problem. (russian)*, Trudy Matematicheskogo Instituta imeni V.A. Steklova (1964b), no. 72, 5–56, English translation in: American Mathematical Society Translations Series 2, 97, 1–14, 1971.
8. 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.
9. ———, *Size and structure of universal turing machines using tag systems: a 4-symbol 7-state machine*, Proceedings Symposia Pure Mathematics, American Mathematical Society **5** (1962), 229–238.
10. Liesbeth De Mol, *Closing the circle: An analysis of Emil Post's early work.*, The Bulletin of Symbolic Logic **12** (2006), no. 2, 267–289.
11. ———, *Solvbility of the halting and reachability problem for tag systems with $\mu = v = 2$*, Preprint 343, Center for Logic and Philosophy of Science, 2007, http://logica.ugent.be/centrum/writings/pubs.php.
12. ———, *Tag systems and Collatz-like functions*, (2007), Submitted to Theoretical Computer Science (under revision).
13. ———, *Tracing unsolvability: A historical, mathematical and philosophical analysis with a special focus on tag systems*, Ph.D. thesis, 2007, submitted to obtain the degree of Doctor in Philosophy, University of Ghent, 2007. Available at: http://logica.ugent.be/liesbeth/dissertation.pdf.
14. Turlough Neary, *Small polynomial time universal Turing machines*, Proceedings of MFCSIT 2006 (Cork, Ireland), 2006.
15. Emil Leon Post, *Formal reductions of the general combinatorial decision problem*, American Journal of Mathematics **65** (1943), no. 2, 197–215.
16. ———, *Absolutely unsolvable problems and relatively undecidable propositions - account of an anticipation*, 1965, in: [3], 340–433. Also published in [17].
17. ———, *Solvability, provability, definability: The collected works of Emil L. Post*, Birkhauser, Boston, 1994, edited by Martin Davis.
18. John Stillwell, *Emil Post and his anticipation of Gödel and Turing*, Mathematics Magazine **77** (2004), no. 1, 3–14.
19. Hao Wang, *Tag systems and lag systems*, Mathematische Annalen **152** (1963a), 65–74.