On the complex behavior of simple tag systems: An experimental approach

Liesbeth de Mol

On the complex behavior of simple tag systems. An experimental approach.

Liesbeth De Mol ⋆

Centre for Logic and Philosophy of Science, Universiteit Gent, Belgium

Abstract

It is a well-known fact that apparently simple systems can give rise to complex behavior. But why exactly does a given system behave in a complex manner? There are two main approaches to tackle this and other related questions. One can take on a theoretical approach or else start from a more experimental study of the behavior of such systems with the help of a computer. In this paper, the experimental approach is applied to tag systems with a very small program size. After a discussion of some of the main theoretical results on tag systems, several results from a computer-assisted and experimental study on tag systems are analyzed. Special attention is given to the well-known example studied by Post with only 2 symbols and a deletion number $v = 3$. These results are combined with some theoretical results on tag systems in order to gain more insight into the computational power of simple tag systems.

1 Introduction

The idea that very small computational devices can give rise to complex behavior is not new. Emil Post was probably one of the first to understand this in the early twenties when he was exploring the behavior of tag systems and when he was able to prove that large parts of *Principia Mathematica* could be reduced to a class of very simple computational devices. This idea has now very clearly arrived. It is a well-known fact that very small computational devices are capable of universal computation. In the meantime, the search for still smaller universal systems is still going on.

⋆ Part of this research was done during a research stay at the Boole centre for research in informatics, Cork university College. It was supported by the Fund for Scientific Research – Flanders (FWO).

Email address: `elizabeth.demol@ugent.be` (Liesbeth De Mol).

One approach to study small computational devices with complex behavior is to use a computer-assisted experimental approach. In this paper this approach is applied to a specific class of small tag systems that includes one famous example provided by Post. The purpose of this study is not only to gain a better understanding of the behavior of this class but also to think about the limits and possibilities of the experimental method within this research context. Besides these experimental results, several theoretical results on tag systems will be discussed. It is argued that a combination of theoretical with experimental results is the most promising approach to gain a better insight into the computational power of small tag systems.

This paper is an extended version of [13]. Besides a more detailed discussion of the results, some new results are added.

2 Computer-assisted research on simple systems with complex behavior

It is only fair to say that with the rise of the computer new areas in the universe of mathematics have been disclosed to the mathematician. Because of its speed and memory, the computer has opened up new possibilities for exploration and experimentation. Also in research on small computational devices with complex behavior the use of the computer and with it, experimentation and exploration, have proven their merit.

For example, in the context of cellular automata, computer-assisted and experimental research has directly or indirectly led to a wide range of interesting results, new concepts, methods and problems (See for example [6], [42], [48], [49]). Another example of experimental computer-assisted research on simple systems with complex behavior comes from the context of Turing machines, where the computer is an indispensable instrument to study the (generalized) Busy Beaver problem, i.e., the problem to determine for a given class of Turing machines with m states and n symbols those Turing machine(s) that halt and output the maximum number of 1s when started from a blank tape. Although this research is experimental in nature it can lead (and has led) to the solution of the problem for specific classes of Turing machines (see for example [2,20]) and made possible the connection between Busy Beaver winners and Collatz-like problems [24].

These examples illustrate that experimental and computer-assisted research can lead to different kinds of advancements in the domain going from supported conjectures and giving important new observations (which are heuristic results) to the development of new methods and even rigorous results. One major goal of this approach is to trace down explanations of why a *given* (class of) system(s) does or does not behave in a certain way at the level of system dynamics, rather than on the level of the actual formal description, or program(s), underlying these dynamics.

Of course, the experimental computer-assisted approach is not guaranteed to lead to a rigorous result. On the contrary, in many cases one will only find clues of how to proceed or insights and heuristic answers why a given class of systems has complex behavior. Still finding such clues or gaining such insights can clear the ground for rigorous results that one would not be able to establish without such clues or insights.

3 Tag systems

3.1 Post's frustrating problem of "tag"

Tag systems were invented and studied by Emil Leon Post [34,35] during his Procter fellowship at Princeton in the academic year 1920-21. They inspired the formulation of his normal systems, which he also developed during that time, and led to the reversal of his program to prove the recursive *solvability* of the Entscheidungsproblem for first-order predicate calculus. Indeed, after nine months of intensive research on tag systems, Post first came to the conclusion that proving the decidability of this Entscheidungsproblem might be impossible. He never proved that this decision problem is undecidable. This was done by Church and Turing in their seminal 1936 papers [3,43]. However, he did formulate a thesis in 1921, called Post's thesis [8,9], which is now known to be logically equivalent to Church's and Turing's and proved on the assumption of this thesis that there are other decision problems, related to the Entscheidungsproblem, that cannot be decided by finite means. [1]

Definition 1 (v-tag system) *A v-tag system T consists of a finite alphabet Σ of μ letters, a deletion number $v \in \mathbb{N}$ and a finite set of rules $a_i \rightarrow w_i, a_i \in \Sigma, w_i \in \Sigma^*$. The words w_i are called the appendants.*

Note that it is not necessary that there is a rule for every letter $a_i \in \Sigma$ and that there is at most one rule for every such letter a_i. In a computation step of a tag system T on a word $A \in \Sigma^*$, if there is a rule for the leftmost letter in

[1] Post did not submit these results to a journal in the twenties. Later, in the forties, he provided a detailed description of his results from this period in his *Absolutely unsolvable problems and relatively undecidable propositions - Account of an anticipation* [35], a posthumously published manuscript edited by Martin Davis. A seriously shortened version was published in 1943 as [34]. More detailed information on these historical matters can be found in [8,9,10,41].

A, then T first appends the appendant associated with this letter at the end of A and then deletes the first v symbols of A. This computational process is iterated until the tag system produces the empty word ϵ or until T produces a word for which there is no rule for the leftmost letter. In those cases, T is said to halt.

Note that in this definition (Definition **I**) a tag system first appends a word and then deletes the first v letters in one computation step. This is also the order used by Minsky [26], Post [34,35] and Watanabe [47]. There is also another definition of v-tag systems T (Definition **II**) where in one computation step, T first deletes the first v letters and then appends the appendant. This definition is, amongst others, used by Minsky [4,5,25,27], Rogozhin [37,38] and Wang [45]. This makes a difference in the way a tag system halts. I.e., in Definition **I** a tag system halts when it produces the empty word, whereas in Definition **II** a tag system halts when it produces a word having a length smaller than v. Note that in both definitions a tag system halts when it produces a word for which there is no rule corresponding with the leftmost letter of that word.

Following the notation of [50], $A_i \vdash A_{i+1}$ means that A_{i+1} is produced from A_i after one computation step, $A_i \vdash^n A_{i+n}$ that A_{i+n} is produced after n computation steps from A_i. The length of a word A will be written as l_A and a^n means that a is repeated n times.

To give an example, let us consider a specific 3-tag system mentioned by Post with $v = 3$ and having rules $\{0 \rightarrow 00, 1 \rightarrow 1101\}$ [34,35]. If $A_0 = 110111010000$ we get the following productions:

$$\vdash \cancel{110}111010000\mathbf{1101}$$
$$\vdash \cancel{111}0100001101\mathbf{1101}$$
$$\vdash \cancel{010}00011011101\mathbf{00}$$
$$\vdash \cancel{000}1101110100\mathbf{00}$$

The word A_0 is reproduced after 4 computation steps and is thus an example of a periodic word. Note that A_0 is reproduced from A_0 after the tag system has deleted all the letters of A_0. In the remainder of this paper, if a given v-tag system T has deleted all the letters of a given word A we will call this a *round* of T on the word A. Note that a round on A takes exactly $\lceil l_A/v \rceil$ computation steps.

Post called the behavior of this one tag system "intractable". Despite its formal simplicity, it is still not known if there exists a method to decide in a finite number of steps for any word $A \in \{0,1\}^*$ and some initial word $A_0 \in \{0,1\}^*$ whether or not this tag system will produce A from A_0. I.e., it is not known if this tag system has a decidable reachability problem (See Definition 6). In Sec. 4.1 this tag system will be discussed in more details.

One can identify three classes of ultimate behavior in tag systems.

Definition 2 (halt) *A tag system T is said to* halt *on an initial word A_0 when there is an $n \in \mathbb{N}$ such that T produces the empty word ϵ after n computation steps on A_0, i.e., $A_0 \vdash^n \epsilon$ or T produces a word A_n after n computation steps on A_0 for which there is no rule for the leftmost letter of A_n.*

Definition 3 (periodicity) *A tag system T is said to be* periodic *on an initial word A_0 if there are $n, p \in \mathbb{N}$ such that $A_0 \vdash^n A_n$ and $A_n \vdash^p A_{n+p} = A_n$ in T. A_n is said to be a periodic word in T with period p.*

Definition 4 (unbounded growth) *A tag system T is said to have* unbounded growth *on an initial word A_0, if for each $n \in \mathbb{N}$ there is an $i \in \mathbb{N}$ such that for each $j > i$, any word A_j, $A_0 \vdash^j A_j$, $l_{A_j} > n$.*

Post considered two decision problems for tag systems, which we will call the *halting problem* and the *reachability problem* for tag systems.

Definition 5 (halting problem) *The halting problem for tag systems is the problem to determine for a given tag system T and any initial word A_0 whether or not T will halt on A_0.*

Definition 6 (reachability problem) *The reachability problem for tag systems is the problem to determine for a given tag system T, a fixed initial word A_0 and any arbitrary word $A \in \Sigma^*$, whether or not there is an n such that $A_0 \vdash^n A$ in T.*

3.2 Preliminaries

Let T be a v-tag system with μ symbols and a finite set of rules $a_i \rightarrow w_i$. Then, given a periodic word P_1 with period p such that $P_1 \vdash P_2 \vdash \ldots P_p \vdash P_1$ then $[P_1] = \{P_1, P_2, ..., P_p\}$ is called *the set of p periodic words* generated by P_1. The *periodic structure* $S(P)$ of a periodic word $P = a_1 a_2 \ldots a_{l_P}$ is defined as:

$$S(P) = a_1 a_{v+1} a_{2v+1} \ldots a_{v(\lceil l_P/v \rceil - 1)+1}$$

i.e., the word formed by concatenating all the letters in P that are read during one round of T on P. For example, if we take the word $P_1 = \underline{1}10\underline{1}11\underline{0}10\underline{0}00$ from the example of Sec. 3.1, then the periodic structure $S(P_1) = 1100$. The set of periodic words $[P_1]$ generated by P_1 is the set of the productions of the example of Sec. 3.1.

The additive complement $\overline{(x \bmod y)}$ of a given number x relative to a modulus

y is defined as follows:

$$\overline{(x \bmod y)} = \begin{cases} y - (x \bmod y) & \text{if } \; x \neq 0 \bmod y \\ 0 & \text{if } \; x \equiv 0 \bmod y \end{cases}$$

Given some word $A_i = a_1 a_2 \ldots a_{l_A}$ over the alphabet Σ then the word $\overrightarrow{A}_{i+\lceil l_{A_i}/v \rceil}$ denotes the word $w_{a_1} w_{a_{v+1}} \ldots w_{a_{v(\lceil l_A/v \rceil -1)+1}}$, with w_{a_i} the appendant corresponding with the symbol $a_i \in \Sigma$. In other words, this is the word resulting after one round on A_i, without its first $\overline{(l_{A_i} \bmod v)}$ letters being erased. For example, if T is Post's example and $A_i = 1001110$ then $\overrightarrow{A}_{i+\lceil l_{A_i}/v \rceil} = 1101110100$. Note that if $l_{A_i} \equiv 0 \bmod v$, $\overrightarrow{A}_{i+\lceil l_{A_i}/v \rceil} = A_{i+\lceil l_{A_i}/v \rceil}$. The additive complement $\overline{(l_{A_i} \bmod v)}$ thus computes the effect of l_{A_i} on the length of $\overrightarrow{A}_{i+\lceil l_{A_i}/v \rceil}$.

3.3 Decidability and Universality in Tag Systems

After his frustrating experiences with tag system, Post never wanted to work on these systems again. He was convinced that they would turn out undecidable, but never proved this. [2] It was Minksy who proved this in 1961 [25] after the problem of tag was suggested to him by Martin Davis. He showed that any Turing machine can be reduced to a 6-tag system. This reduction is rather involved. It was improved by Cocke and Minsky [4,5,26]. They showed that any Turing machine can be reduced to a 2-tag system. Maslov generalized this result. He proved that for any $v > 1$ there is at least one tag system with an undecidable decision problem [23]. Wang [45] proved that any tag system with $v = 1$ has a decidable reachability problem. It thus follows that the deletion number v is one decidability criterion [21] for tag systems with $v = 2$ as the frontier value, i.e., the minimum value n such that for any class of v-tag systems with $v \geq n$ there is at least one tag system with an undecidable reachability problem.

Another such criterion is the length of the appendants. Wang proved that any tag system with the length of the smallest appendant $l_{\mathbf{min}} \geq v$ or the length of the longest appendant $l_{\mathbf{max}} \leq v$ has a decidable reachability problem [45]. [3]

---

[2] From a private communication with Martin Davis.

[3] Note that Wang uses Definition **II** for tag systems. It is easily checked that this result remains valid if Definition **I** is used. To see this quickly note that if $l_{\mathbf{max}} \leq v$ then T can never produce a word of length longer than the initial word and thus T will either halt or become periodic. If $l_{\mathbf{min}} \geq v$ we have that for every word A_n resp. A_{n+1} produced after n resp. $n+1$ computation steps on some initial word A_0 we have that either $l_{A_{n+1}} = l_{A_n}$ or $l_{A_{n+1}} > l_{A_n}$ and thus the decidability of the reachability problem also easily follows in that case.

He furthermore proved that there is a universal tag system with $v = 2, l_{\mathbf{max}} = 3, l_{\mathbf{min}} = 1$. This result was proven independently by Maslov [23]. Minsky and Cocke also constructed a universal tag system with the same parameters [4]. This criterion was also studied by Pager [32]. It follows from these results that $l_{\mathbf{max}} - v$ resp. $v - l_{\mathbf{min}}$ are decidability criteria for tag systems with 1 as the frontier value.

A third decidability criterion is the number of symbols μ. Let $\mathrm{TS}(\mu, v)$ denote the class of tag systems with μ symbols and a deletion number v. It was proven by Post that the classes $\mathrm{TS}(1, v)$, $\mathrm{TS}(\mu, 1)$ and TS(2, 2) have a decidable reachability problem. Regretfully, Post never published these results. He does mention that the proof for the classes TS(1, v) and $\mathrm{TS}(\mu, 1)$ is trivial, while the proof for the class TS(2,2) involved "*considerable labor*". A proof for the class TS(2,2) has recently been reestablished (See [11,15] for more details). The proof is quite involved due to the large number of studied cases. The main method of the proof is called the table method. This method is a very useful tool to study the behavior of tag systems.

Until recently the number of symbols μ was never really studied, with Post as an exception. As a consequence, although one has constructed the smallest possible universal tag systems with respect to $v, l_{\mathbf{max}}$ and $l_{\mathbf{min}}$, the value of μ for these universal tag systems is still relatively large. In fact, the universal tag systems that can be constructed with the current methods all have a very large number of symbols. It immediately follows from the results of [4,26] that it is possible to reduce any Turing machine with m states and 2 symbols to a tag system with $v = 2, \mu = 32m$. Let $\mathrm{TM}(m, n)$ denote the class of Turing machines with m states and n symbols. Using the universal Turing machine constructed by Neary and Woods in TM(15,2), which simulates a variant of tag systems called bi-tag systems [30], or Baiocchi's machine in TM(19,2) [1] which simulates 2-tag systems, it is possible to construct universal tag systems in the classes TS(480, 2) resp. TS(608,2). The encoding by Cocke and Minsky can be easily generalized resulting in the possibility of reducing any Turing machine with m states and n symbols to a tag system with $v = n$, $\mu = nm(4n + 8)$. Note that this encoding does not allow to directly reduce the weak and semi-weak machines by Neary and Woods [29,51] and Cook [6] to a tag system since these Turing machines make use of an infinitely repeated periodic word to the left and right of the input (in case of weak universality) and left or right of the input (in case of semi-weak universality). This cannot be directly translated into tag systems since they cannot work on infinite words. In order to simulate these weak and semi-weak machines one needs to add some extra machinery that generates these periodic words every time they are needed.

3.4 Significance of Tag Systems

Research on tag systems for their own sake has remained relatively limited as compared to research on Turing machines and cellular automata. This is quite surprising. Given, on the one hand, the simplicity of the form of tag, and, on the other hand, its computational power, tag systems might be very good candidates for finding the "simplest" possible universal system. However, this is not the only motivation for studying tag systems.

3.4.1 Tag systems and small universal devices

Tag systems have played and still play a fundamental role in research on small universal devices. A lot of universal devices have been proven universal through the simulation of 2-tag systems or a variant of tag systems. Minsky was the first to construct a very small universal Turing machine in TM(7,4) that simulates 2-tag systems [26]. Rogozhin [37,38] constructed several small universal Turing machines by 2-tag simulation and improved Minsky's machine. Also Baiocchi's machines [1] are 2-tag simulators. For a more detailed overview see [21]. Neary and Woods [28,30] recently found universal Turing machines in TM(9, 3), TM(5,5), TM(6,4) and TM(15,2) simulating what they have called bi-tag systems, a variant of tag systems. Matthew Cook proved that cellular automaton rule 110 is weakly universal through the simulation of cyclic tag systems, yet another variant of tag systems. Also the semi-weakly universal machines by Woods and Neary [51,52] simulate cyclic tag systems. Tag systems have also been used in the context of small universal circular Post machines [19].

Previously it was the case that the universal 2-tag systems that can be constructed using the Cocke-Minsky method all suffer from an exponential slowdown. As a result, all the universal devices simulating 2-tag systems suffered from this same defect. This problem was resolved by Neary and Woods: they showed that 2-tag systems can simulate Turing machines in polynomial time by proving (1) that cyclic tag systems simulate Turing machines in polynomial time and (2) that 2-tag systems are efficient simulators of cyclic tag systems [31,50]. It should also be pointed out that their bi-tag simulators are polynomial.

3.4.2 Tag systems and number theory

In his *Account of an anticipation* Post mentions that he was confronted with problems of ordinary number theory during his research on tag systems. He even writes about an "*intrusion of number theory*" into his research. This is not surprising. In a certain sense tag systems can be understood as a kind of modulo systems due to the regularity induced by always removing the same number of symbols at the beginning of a word. This is the reason why it is so easy to determine remainders with tag systems.

Lemma 1 *There is a v-tag system T with $2v+2$ symbols that computes n* mod *v for any $n \in \mathbb{N}$.*

Proof Let T be a v-tag system, $\mu = 2v+2$ symbols, $\Sigma = \{a, e, b_0, \ldots, b_{v-1}, 0, ..., v-1\}$ and the following set of $v+2$ production rules:

$$\begin{aligned}
&a \to b_0 b_{v-1} b_{v-2} ... b_1 e^v \\
&e \to \epsilon \\
&b_i \to e^{v-i} i && \text{if } i > 0 \\
&b_i \to i && \text{if } i = 0
\end{aligned}$$

Note that every number $0 \leq i < v$ should be regarded as a letter of the alphabet. Given an initial word $A_0 = a^v e^l$ of length $l+v$, then T will output $l \bmod v$ after $\lceil l/v \rceil + 3$ computation steps. Indeed, after one round on A_0, T will read the letter with the index $l \bmod v$ which, in its turn produces the desired output i. Note that this technique works for both Definitions **I** and **II** of tag systems discussed in Sec. 3.1. $\square$

This simple technique is one of the main techniques of the Cocke-Minsky scheme where it is used to determine whether a given word is odd or even. It is also one of the main techniques used in the reduction of the Collatz problem to a very small tag system. 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 Collatz 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 iterations. This number-theoretical problem can be simulated by a very small 2-tag system with the following production rules: $a_0 \to a_1 a_2, a_1 \to a_0, a_2 \to a_0 a_0 a_0$. If $n = 2m$, then this tag system will produce a_0^m after two rounds on a_0^{2m} and if $n = 2m+1$, then it produces a_0^{3m+2} after two rounds on a_0^{2m+1} (Note that if $n = 2m+1$ then $C(C(2m+1)) = C(6m+4) = 3m+2$). If we use Definition **I** then this tag system is periodic on a_0^4. If we use Definition **II** this tag system halts when the Collatz function becomes periodic (See [14] for more details).

Another result illustrating the connection between tag systems and number theory is a theorem that proves that any decision problem for a v-tag system T with appendants $w_0, \ldots, w_{\mu-1}$ for which $v, l_{w_1}, \ldots, l_{w_{\mu-1}}$ are not relatively prime reduces to the same decision problem for n smaller tag systems, with n the greatest common divisor of $v, l_{w_1}, \ldots, l_{w_{\mu-1}}$ [12]. The reverse of this theorem allows to make a composite tag system out of any given set of tag systems.

4 Playing with tag systems

> "Post found this (00, 1101) problem "intractable", and so did I even with the help of a computer. Of course, unless one has a theory, one cannot expect much help from a computer (unless *it* has a theory) except for clerical aid in studying examples; but if the reader tries to study the behavior of 100100100100100100100 without such aid, he will be sorry."

Marvin Minsky, 1967.

4.1 Post's example

The most famous tag system is Post's example from Sec. 3.1. Several researchers have studied this specific tag system and came to the conclusion that it is an example of a very small tag system that has very complex behavior. As explained in Sec. 3.1 it is still not known whether this particular example has a decidable reachability problem, despite its apparent simplicity. More research on this and related tag systems is thus very important in the context of research on small devices that have complex, possibly universal, behavior. In fact, if this tag system would turn out to be universal, it would be one of the simplest universal devices known.

It is clear from Post's *Account of an anticipation* [35] that he spend a lot of time investigating the example from Sec 3.1. Among other things, he remarks that "*[n]umerous initial sequences actually tried led in each case to termination or periodicity, usually the latter.*" Several other researchers including Hayes and Minsky [17,18,27] did the same kind of "experiment" with the help of the computer and came to the same conclusion. Minsky [27] remarks about this tag system that, if one looks at its description, one might expect that it will always halt or become periodic. In Post's example we have that the total number of letters 0 in the appendants (written as #0) is the same as the total number of letters 1 in the appendants (written as #1), i.e., #0 = #1 = 3. We also have that the effect of reading a 0 on the length of a given word A_n cancels out the effect of reading a 1 on the length of a word A_n (if the leftmost letter of A_n is a 0 then $l_{A_{n+1}} = l_{A_n} - 1$, if it is a 1 then $l_{A_{n+1}} = l_{A_n} + 1$). These two features of the production rules can be summarized as $\#1 \times (l_{w_1} - v) + \#0 \times (l_{w_0} - v) = 0$. Because $\#0 = \#1$ one could then expect that for any n and some initial word A_0, the probability that the leftmost letter of the word $A_0 \vdash^n A_n$ is 0 or 1 is the same. If this would be true, then, statistically speaking, periodicity or a halt can indeed be expected.

Post's tag system was also studied by Watanabe, who is known for his work on constructing small universal Turing machines in the 60s (see e.g. [46]). He made a detailed theoretical analysis of the periodic behavior of the tag system "*as a preliminary of obtaining a simple universal process*" [47]. Let $a = 00, b = 1101$. Watanabe deduced wrongly that there are only four kinds of periodic words in Post's tag system, i.e., $a^2b^3(a^3b^3)^n$ with period 6, ba with period 2, b^2a^2 with period 4, or any concatenation of the last two. In some preliminary runs on Post's tag system we found three other kinds of periodic words, a period 10, 40 and 66. The period 10 ($b^2a^3b^3a^2$) is similar to the periodic words found by Watanabe, the period 40 and 66 are very different from these periodic words. This will be explained in Sec. 4.2.3.

Brain Hayes [18] also did some experimental research on the periodic behavior of Post's tag system. He observed that all the periods are even numbers. Shearer [40] proved that for any number $2n$ there is a periodic word in Post's tag system with period $2n$, i.e., any word $110111010000(001101)^m$ is a periodic word with period $4 + 2m$. See 4.2.3 for more details.

Post also mentions that he had "*an easily derived "probability" prognostication*" to determine for a given initial word whether it would halt or become periodic. This is probably related to the number of 1s relative to the number of 0s in an initial word. This has been checked by an experiment that studies the effect of increasing the number of 1s in an initial word on the probability that an initial word either results in a halt, periodicity or none of these two after a given number n of computation steps. The preliminary results of this experiment show that an increase in the number of 1s in the initial word does have an important effect in this context. Fig. 1 shows that an increase of the number of 1s in the initial word indeed decreases the probability of a halt and increases the probability of periodicity or a word for which the tag system has not become periodic after 10^5 computation steps.

4.2 Six computers experiments on the class TS(2,v)

Given the difficulties involved with Post's tag system, 6 different computer experiments were performed on 52 related tag systems. 50 were generated through a randomized algorithm, one was developed by hand and one is Post's

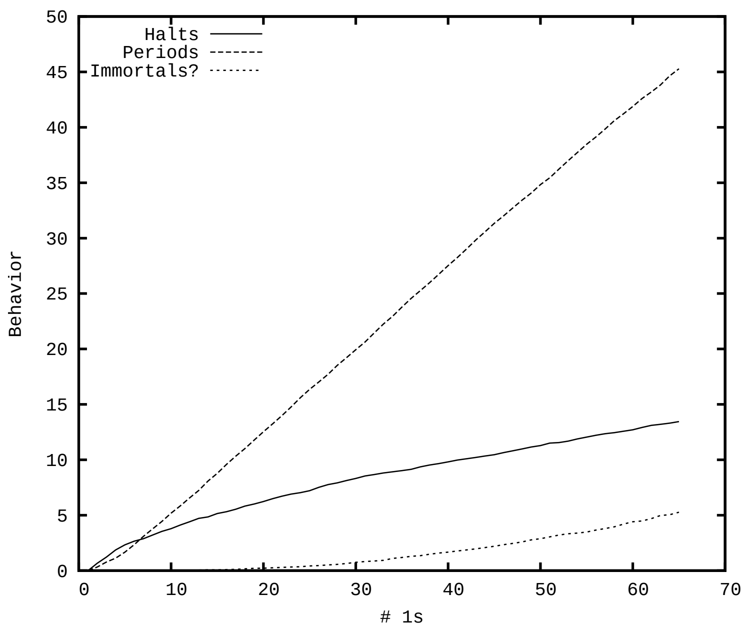

Figure 1. Evolution of the number of initial words of length 200 that become periodic, halt or have not halted or become periodic after 10^5 computation steps for an increasing number of 1s in the initial word.

tag system. In what follows the main focus will be on the results from Experiments 1 and 2.

The experiments serve different purposes. First of all, by studying Post's tag system in relation to other tag systems it is possible not only to situate Post's tag system in a broader class and thus possibly to determine some more general properties, but also to explore the behavior of a whole class of related tag systems that lies very close to the decidable class TS(2,2). Some of the experiments were also used to verify some of the experimentally established properties of Post's tag system or to find a better explanation for some of these properties. In general, these experiments make it possible to draw certain heuristic and theoretical conclusions about the behavior of very small tag systems, similar to Post's tag system, i.e., small tag systems for which it is unclear for now whether they have a decidable reachability problem or not.

4.2.1 Generating intractable tag systems

As explained in Sec. 3.1 a tag system can have three kinds of ultimate behavior: it can halt, it can become periodic or it can have unbounded growth on an initial word. There are several tag systems for which it can be easily determined what kind of behavior they will have given the production rules and the initial word. For example, a tag system for which $l_{\mathbf{min}} > v$ will always have unbounded growth since for every word A_n produced after n computation steps on A_0 we have that $l_{A_n} > l_{A_{n-1}}$. The difficult cases are those tag systems which, when run on the computer, show behavior that, although it might ultimately result in a halt or periodicity, is very erratic. To illustrate this, Fig. 2 gives a visualization of a number of productions of Post's tag system. Since

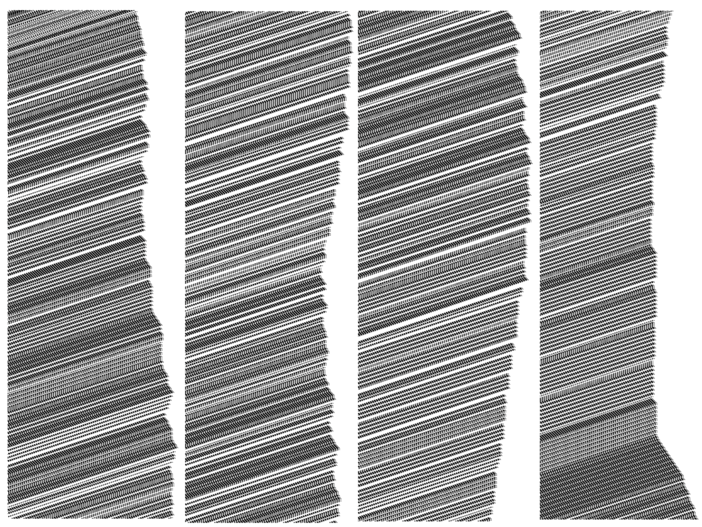

Figure 2. Sample of the behavior of Post's tag system with an arbitrary initial word

Post's example was the only tag system known that has this kind of behavior, 50 other tag systems were computer-generated that can be considered similar to Post's tag system. These tag systems were selected from a randomly generated set of tag systems. The algorithm posed a limit on the value of the deletion number v: it is a random number $3 \leq v \leq 15$. Also the length of $l_{\mathbf{max}}$ was a random number bounded by a constant. As the main interest is in tag systems with a very small alphabet Σ, Σ was set to $\{0, 1\}$. Several selection criteria were then used to generate and select the tag systems. Besides Wang's decidability criterion with $l_{\mathbf{max}} - v \geq 1, v - l_{\mathbf{min}} \geq 1$ the two most important selection criteria used are heuristic in nature.

The first criterion is related to the observation that, in Post's tag system, $\#1 \times (l_{w_1} - v) + \#0 \times (l_{w_0} - v) = 0$ (See Sec. 4.1). The algorithm that generated the other 50 tag systems incorporates this property and thus for any of the tag systems used in the experiments we have that $\#1 \times (l_{w_1} - v) + \#0 \times (l_{w_0} - v) = 0$. Note that this does not imply that for each of the tag systems $\#0/(\#1 + \#0) =$

$\#1/(\#1 + \#0)$ (as is the case for Post's tag system). One could then expect for each of the tag systems thus generated that for any n and some initial word A_0, the probability that the leftmost letter of the word $A_0 \vdash^n A_n$ is 0 or 1 is $\#0/(\#1 + \#0)$ or $\#1/(\#1 + \#0)$ respectively. If this would be true then statistically speaking, periodicity or a halt can always be expected because $\#1 \times (l_{w_1} - v) + \#0 \times (l_{w_0} - v) = 0$.

After the determination of $v, l_{w_0}, l_{w_1}, \#1$ and $\#0$ the appendants were generated through a biased random generator (using $\#1/(\#1+\#0)$ and $\#0/(\#1+\#0)$). A second heuristic criterion was then applied to the tag systems thus generated. It selects tag systems that are able to keep going for a huge number of computations steps without resulting in periodicity, a halt or "predictable" unbounded growth. In order to check this, each generated tag system was run with 20 different and randomly selected initial words of length 300. If the tag system did not lead to a halt, periodicity or was not recognized as a possible case of "predictable" unbounded growth it was selected.

Since it is very hard to trace down "predictable" unbounded growth, we simply placed a bound on the lengths of the words produced. If the tag system produced a word W with $L_W > 15000$ it was excluded. Note that this does not necessarily mean that the tag system is really a case of unbounded growth. The reason for choosing such a limit is that for those tag systems $T \in TS(2, 2)$ that were proven to have unbounded growth in [11,15], the length of the produced words grows very fast. It thus seemed reasonable to assume that if one has a tag system that can be easily proven to have unbounded growth, then the length of the words produced by this tag system will grow very fast. If this is not the case one expects that as long as the tag system does not halt or become periodic the average length of the words will increase very slowly.

This algorithm resulted in 50 different tag systems. The smallest resp. the largest deletion number v was 3 resp. 13. The smallest resp. largest value for $l_{\mathbf{max}} - v$ and $v - l_{\mathbf{min}}$ was 1 and 4. Table 1 gives an overview of these 50 tag systems (**T3**–**T52**). **T1** is Post's tag system, **T2** is a tag system that was constructed by hand.

Table 1: Tag systems generated by Algorithm 2

Tag System	w_0	w_1	v
T1	00	1101	3
T2	00101	1011010	6
T3	111	01000	4

Continued on next page

Table 1 – continued from previous page			
Tag System	w_0	w_1	v
T4	11101	1100000	6
T5	010110	11100100	7
T6	0	01011	3
T7	101011	00011010	7
T8	011	111100	5
T9	101	0000111	5
T10	001	10110	4
T11	001	01110	4
T12	0	01011	3
T13	0110001	10000101111	9
T14	1010	110100	5
T15	111	0110000	5
T16	111000	11010110011000	10
T17	1001111	10100000011	9
T18	000110	101001010000	8
T19	110	001111	5
T20	1011000	111011000	8
T21	11011011	1110000000	9
T22	101001001	0101100110011	11
T23	001	010100	4
T24	11	00111000	5
T25	10000111	1000100111	9
T26	00111	0111000	6
T27	11011	0011000	6
T28	111000	11000110011100	10
T29	110	01001	4
Continued on next page			

Table 1 – continued from previous page			
Tag System	w_0	w_1	v
T30	000111	11000011	7
T31	1	10100	3
T32	111010101110	00110101010000	13
T33	10001	1110010	6
T34	010	001001	4
T35	0010101	01010100100011	10
T36	1011	010100	5
T37	1111	010000	5
T38	000101	000000111	7
T39	00101	1001000110	7
T40	001	110000	4
T41	101	00001110011	7
T42	10111	0000011	6
T43	100	11001	4
T44	1111	00110000	6
T45	101	0011010	5
T46	1011	110000	5
T47	0	1001101	4
T48	11010011110	1111000010000	12
T49	001	100100	4
T50	110	11000	4
T51	1110010	00111110000	9
T52	01101	0111000	6

4.2.2 Experiment 1: Distribution of the three classes of behavior

What are the chances that a random initial word will result in a halt or periodicity? How probable is it that, given some initial word, a tag system will keep going for millions of computations steps without resulting in one of the three classes of behavior (periodicity, halt and unbounded growth)? These kind of questions were explored in the first experiment. It checked the distribution of the three classes of behavior in the 52 tag systems for a set of random initial words.

Each of the tag systems was run for 1998 different randomly generated initial words. The experiment/program kept track of the number of initial words that resulted in a halt, periodicity or unbounded growth and those that did not lead to either one of these three classes of behavior after 10.000.000 computation steps. These last words were tentatively classified as "Immortals?". The results from the experiment show that there is a clear variation between the different tag systems concerning the chances that a the tag system will result in one of the three classes of behavior or not. Table 2 shows the results for some of the tag systems. [4]

Table 2: Number of initial words that halt, become periodic, result in the production of a word W, $l_W > 15000$ (Growth), or cannot be classified in neither of these classes after 10^7 computation steps ("Immortals?").

Tag System	**Halts**	**Periodics**	**Immortal?**	$l_W > 15.000$
T1	358	1598	37	5
T12	1917	18	57	6
T22	0	1303	617	78
T28	0	1966	24	8
T37	0	1636	362	0
T47	1067	885	27	19

The most significant difference between the tag systems is the fact that only 5 out of the 52 tag systems have initial words that resulted in a halt (including Post's tag system). Upon further inspection of the details of the tag systems that did not result in one halt it was proven that their halting problem is

[4] The complete results can be found in the on-line document available at http://logica.ugent.be/liesbeth/results.pdf

decidable (see [12] for more details). For Post's tag system, it is clear that the chances for periodicity are very high: about 80% of the initial words tested resulted in periodicity, while only about 18% resulted in a halt and about 2% were classified as "Immortals?" The number of initial words that result in the production of a word W with $l_W > 15.000$ is negligible (this is the case for almost every one of the tag systems). These results show that the chances that an initial word will result in a halt or periodicity for this tag system are very high and thus confirm the observations on this tag system made by other researchers. Furthermore, it might well be that the remaining 2% will also ultimately result in a halt or periodicity if they were to be run for more than 10^7 computation steps.

Knowing that the chances are high that a given tag system will halt or become periodic gives some more information about that tag system. However, the experiment did more than just that: if an initial word did result in a halt, periodicity or the production of a word $W, l_W > 15000$, the experiment also stored the number of computation steps it took the tag system before either one of these three cases occurred. On the basis of this count plots were made for each of the 52 tag systems mapping the number of initial words that has not yet resulted in a halt, periodicity or unbounded growth against the number of computation steps. Fig. 3 shows two of these plots. [5] . The plots show

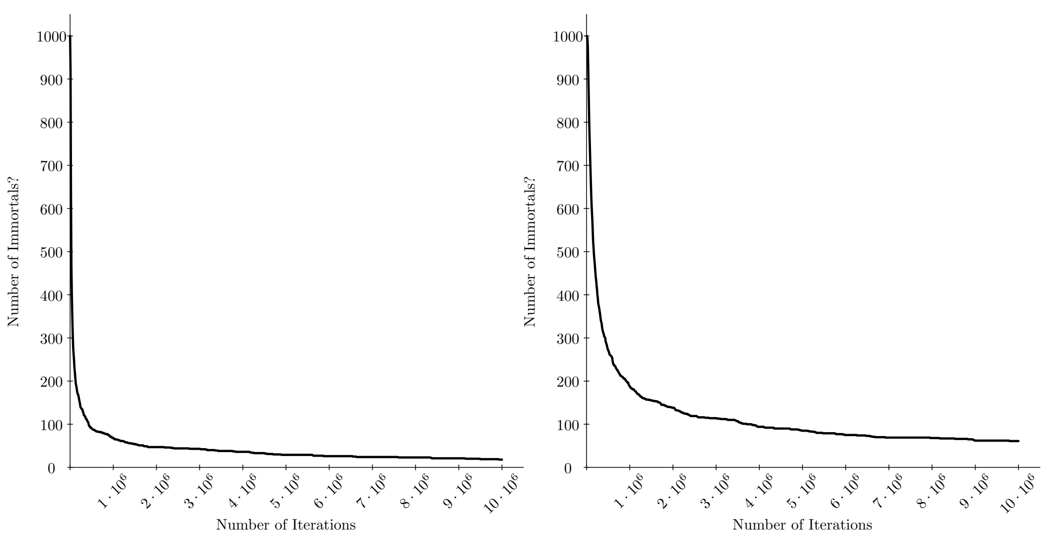

Figure 3. Plots of the evolution of the number of "Immortals?" of Post's tag system (the left plot) and the plot of **T36** with $v = 5, 0 \rightarrow 1011, 1 \rightarrow 010100$.

that the number of "Immortals?" decreases with the number of computation steps. Observe that there is a kind of "phase transition" in this behavior. In a first phase, the number of "Immortals?" decreases exponentially fast, in a second phase, the number of "Immortals?" decreases exponentially slow. This

[5] All 52 plots can be found in the on-line document available at http://logica.ugent.be/liesbeth/results.pdf

means that it does not take a huge number of computation steps before a halt or periodicity occurs for most initial words. Indeed, as the plots show, it only takes about 1.000.000 computation steps before most initial words have resulted in a halt or periodicity. However, once past this point, the number of initial words that results in a halt or periodicity at a given time $n < 10.000.000$ increases very slowly. This is not only the case for the tag systems shown in Fig. 5 but for all of the tag systems tested. [6] This suggests that it might be relatively easy to prove for most initial words that they will result in a halt or periodicity but that there is a small percentage of initial words for which this is not the case.

The plots also suggest that in the second phase of slow decrease the number of "Immortals?" converges to a limit. One important question to be asked is whether this limit is positive. I.e., is there a finite point at which the plots intersect with the x-axis or not? If we would be able to prove that for every class of initial words of arbitrary length l, this intersection point is finite for a given tag system T, we would have proven its reachability problem. If however this is not the case, then there are "Immortals" for T. The presence of "Immortals" adds to the unpredictability of these tag systems. Indeed, this would mean that given any n, it is always possible to find an initial word that will not have halted or become periodic after n computation steps. Of course, this is always the case for what one could call trivial initial words, i.e., initial words of a length that is not significantly smaller than the number of computation steps n. This is also the case for tag systems that can be proven to have unbounded growth, i.e., tag systems with a decidable reachability problem. However, the results of this experiment and Experiments 3–5 show that the behavior of the "Immortals?" is far from being trivial.

4.2.3 Experiment 2: Periodicity in tag systems

In the second experiment the periodic behavior of each of the tag systems was studied. Research on the periodic behavior of a certain class of computational systems can be very fruitful. For example, Cook used periodic words to prove that cellular automaton rule 110 is weakly universal [6]. A detailed analysis was performed on the periodic words found during Experiment 1. The main purpose of the experiment was to explore what kind of different periods and periodic words one can expect for these tag systems. The experiment first of all checked the different periods p found for each of the tag systems. These results show that there is a great variety in the periodic behavior of the 52 tag systems, some tag systems having a very low number of different periods, others having

[6] A few of the plots are not as smooth as those from Fig. 5, and have more discrete transitions. A few have more than one (discrete or smooth) transition between a fast decrease resp. slow decrease in the number of left-overs.

a great variety of different periods. Most tag systems only produce periods of even length although there are some exceptions. There are also some tag systems with very long periods, the longest being of length no less than 462321 (**T34**). Table 3 gives some of the typical results for some of the tag systems.[7]

Table 3: Results from Experiment 2. The first column identifies the tag system, the second gives the total number of periodic words and the last the different periods p (in bold) found and the percentage (between brackets) of the number of times a given period p was found

T.S.	Tot.	Periods and # of each period rel. to Tot. periods
T1	790	**6** (84.2), **10** (9.37), **28** (1.39), **36** (1.27), **34** (0.89), **22** (0.76), **46** (0.38), **16** (0.38), **40** (0.38), **20** (0.38), **32** (0.25), **54** (0.13), **14** (0.13), **70** (0.13)
T34	912	**3** (52.7), **462321** (26.5), **22302** (17.3), **522** (3.18), **636** (0.11), **465** (0.11)
T35	954	**7** (53.6), **42** (17.5), **28** (10.7), **56** (6.18), **63** (3.46), **126** (2.73), **70** (1.99), **84** (1.47), **2002** (0.73), **784** (0.73), **2709** (0.42), **11760** (0.31), **112** (0.21)
T46	955	**74** (6.18), **70** (5.24), **66** (4.83), **62** (4.82), **34** (4.61), **50** (4.5), **38** (4.29), **58** (3.87), **78** (3.66), **82** (3.25), **94** (3.14), **54** (2.94), **86** (2.83), **98** (2.72), **4** (2.51), **72** (2.2), **42** (2.2), **60** (1.88), **88** (1.68), **64** (1.68), **90** (1.68), **118** (1.57), **52** (1.57), **102** (1.47), **110** (1.36), **5382** (1.36), **236** (1.36), **106** (1.36), **68** (1.36), **76** (1.36), **46** (1.26), **122** (1.15), **114** (1.15), **160** (0.94), **96** (0.94), **84** (0.94), **80** (0.94), **40** (0.84), **56** (0.84), **48** (0.73), **112** (0.63), **36** (0.52), **104** (0.52), **130** (0.52), **134** (0.52), **128** (0.42), **138** (0.42), **180** (0.42), **126** (0.42), **1194** (0.42), **152** (0.42), **100** (0.31), **30** (0.31), **108** (0.31), **166** (0.21), **124** (0.21), **146** (0.21), **32** (0.21), **170** (0.21), **116** (0.21), **178** (0.21), **142** (0.21), **120** (0.21), **136** (0.21), **92** (0.21), **144** (0.1), **154** (0.1), **186** (0.1), **770** (0.1), **132** (0.1), **174** (0.1), **218** (0.1), **148** (0.1), **156** (0.1)

The more important results from this experiment concern the detection of a fundamental difference between the different kinds of periodic words that

[7] The complete table can be found in the on-line document available at: http://logica.ugent.be/liesbeth/results.pdf

can be produced by the 52 tag systems. This resulted from a more detailed analysis of the periodic structure and the lengths of the periodic words found. The analysis was inspired by previous explorations of the periodic behavior in Post's tag system which resulted in the detection of what seemed to be two fundamentally different types of periodic words. The current analysis initially resulted in no less than four different types [13].

The first, type I, contains periodic words P for which the period p is always less or equal to the length of the periodic structure, i.e., $p \leq l_{S(P)}$. Roughly speaking, this means that a periodic word will have reproduced itself at least once after all its letters have been erased. This type can be split into two subtypes Ia and Ib. A periodic word P of type Ia, is a word for which $l_{S(P)} \equiv 0 \bmod p$, a periodic word P of type Ib, is a word for which $l_{S(P)} \bmod p \neq 0$. The second type II contains periodic words P for which the length of the periodic structure $l_{S(P)}$ is always strictly smaller than the period, i.e., $p > l_{S(P)}$. This means that a word will not have reproduced itself after all its letters have been erased. Originally this type was also split into two subtypes IIa and IIb. A word P is of type IIa, when $p \equiv 0 \bmod l_{S(P)}$, a word P is of type IIb when $p \bmod l_{S(P)} \neq 0$.

An important generalization has made it possible to prove for all types that, given a word of one these types, it is possible to generate an infinite number of periodic words with different periods. This is the reason why the differentiation into type IIa and IIb has now become superfluous, and we will thus not discuss the two types separately. The split-up into type Ia and Ib remains. The reason for this is that it is possible, on the basis of words of type Ia, to generate an infinite set of different periodic words with the same period. The method for generating this set cannot be applied to words of type Ib and II.

Type Ia An example of a periodic word of type Ia was already provided in Sec. 3.1. Here is another example of type Ia in Post's tag system. The periodic structure $S(P_i)$ is underlined for each of the periodic words P_i:

$P_1 = \underline{0}01\underline{1}01$
$\vdash P_2 = \underline{1}01\underline{0}0$
$\vdash P_1 = \underline{0}01\underline{1}01$

Clearly P_1 has period $p = 2$: it is reproduced after 2 computation steps. For each of the periodic words produced from P_1, the period p is always less or equal to $l_{S(P_i)}$. Also note that $l_{S(P_1)} \equiv 0 \bmod p$. This last property implies that P_1 will reproduce itself every $p(l_{S(P_i)}/p)$-th computation step, and thus also after every round on P_1. An immediate consequence of this property is that given a periodic word P_i of type Ia with $l_{S(P_i)} \equiv 0 \bmod p$, one can construct an infinite number of different periodic words with the same period p, namely any word $P_i(\overrightarrow{P}_i)^n, n \in \mathbb{N}$. This is not possible for words of type Ib and II.

Another consequence for periodic words P_i with $l_{S(P_i)} \equiv 0 \bmod v$ is that it is also fairly easy to construct an infinite number of periodic words *with different periods* for any v-tag system that has periodic words $P_{a_1}, P_{a_2}, \ldots, P_{a_n}$ of type Ia with periods $p_1, p_2, ..., p_n$, $l_{S(P_{a_i})} \equiv 0 \bmod p_i$, $l_{\overrightarrow{P}_{a_i}} - \overline{l_{P_{a_{i-1}}} \bmod v} = l_{P_{a_i}}$ (1). Indeed, given such words P_{a_i} for any number $p = l_{S(P_{a_1})} + m_1 l_{S(P_{a_1})} + m_2 l_{S(P_{a_2})} + \ldots + m_n l_{S(P_{a_n})}, m_i \in \mathbb{N}$ the word $P = P_{a_1}\overrightarrow{P}_{a_1}^{m_1}\overrightarrow{P}_{a_2}^{m_2}\ldots\overrightarrow{P}_{a_n}^{m_n}$ is a periodic word of type Ia with period p. Note that the extra condition (1) is necessary to ensure that the right letters will be read in each of the P_{a_i} for every round on P.

Type Ib The following productions give an example of a periodic word P_1 of type Ib in the tag system **T3** with $v = 4, 0 \rightarrow 111, 1 \rightarrow 01000$:

$P_1 = \underline{1}111\underline{0}100\underline{0}010\underline{0}001\underline{0}000\underline{1}000\underline{1}111\underline{1}111\underline{1}111\underline{0}100\underline{0}010\underline{0}001\underline{0}00$
$\vdash P_2 = \underline{0}100\underline{0}010\underline{0}001\underline{0}000\underline{1}000\underline{1}111\underline{1}111\underline{1}111\underline{0}100\underline{0}010\underline{0}001\underline{0}000\underline{1}000$
$\vdash P_3 = \underline{0}010\underline{0}001\underline{0}000\underline{1}000\underline{1}111\underline{1}111\underline{1}111\underline{0}100\underline{0}010\underline{0}001\underline{0}000\underline{1}000\underline{1}11$
$\vdash P_4 = \underline{0}001\underline{0}000\underline{1}000\underline{1}111\underline{1}111\underline{1}111\underline{0}100\underline{0}010\underline{0}001\underline{0}000\underline{1}000\underline{1}111\underline{1}1$
$\vdash P_5 = \underline{0}000\underline{1}000\underline{1}111\underline{1}111\underline{1}111\underline{0}100\underline{0}010\underline{0}001\underline{0}000\underline{1}000\underline{1}111\underline{1}111\underline{1}$
$\vdash P_6 = \underline{1}000\underline{1}111\underline{1}111\underline{1}111\underline{0}100\underline{0}010\underline{0}001\underline{0}000\underline{1}000\underline{1}111\underline{1}111\underline{1}111$
$\vdash P_7 = \underline{1}111\underline{1}111\underline{1}111\underline{0}100\underline{0}010\underline{0}001\underline{0}000\underline{1}000\underline{1}111\underline{1}111\underline{1}111\underline{0}1000$
$\vdash P_8 = \underline{1}111\underline{1}111\underline{0}100\underline{0}010\underline{0}001\underline{0}000\underline{1}000\underline{1}111\underline{1}111\underline{1}111\underline{0}100\underline{0}010\underline{0}0$
$\vdash P_1 = \underline{1}111\underline{0}100\underline{0}010\underline{0}001\underline{0}000\underline{1}000\underline{1}111\underline{1}111\underline{1}111\underline{0}100\underline{0}010\underline{0}001\underline{0}00$

The period of this set of periodic words is 8 since P_1 repeats itself exactly after 8 computation steps. As in the previous example, the periodic structure $S(P_i)$ of every word P_i is underlined. For each $P_i, p < l_{S(P_i)}$ and $l_{S(P_i)} \bmod p \neq 0$. Now, since the length of the periodic structure is, for none of these words, divisible by the period, it is not the case that one of these words will repeat itself every $p(l_{S(P_i)}/p)$ computation steps and thus also not after one round. As a consequence it becomes impossible to generate an infinite number of different periodic words with the same period p. However, it does remain possible to generate an infinite number of periodic words with different periods. We need the following lemma to prove this:

Lemma 2 *Given a v-tag system T, a periodic word P_1 with period p of type I and the set of periodic words $[P_1]$ generated by P_1, then there is at least one $P_i \in [P_1]$ for which it takes at most p rounds of T on P_i to reproduce P_i.*

Proof Given a v-tag system T, a word P_1 with period p of type I and the set of periodic words $[P_1] = \{P_1, P_2, \ldots, P_p\}$ generated by P_1. Now starting from any of the words $P_{i_1} \in [P_1]$, after one round of T on P_{i_1} the word $P_{i_2} \in [P_1]$ with $i_2 = (i_1 + l_{S(P_{i_1})}) \bmod p$ will be produced. To see this note that for words of type I, a round always consists of $np + k, 0 \leq k < p, n \in \mathbb{N}$ computation steps. Now, clearly, if $l_{S(P_{i_1})} \equiv 0 \bmod p$ $(k = 0)$, $P_{i_2} = P_{i_1}$. If this is not the case, then after one more round of T on P_{i_2}, T produces the word $P_{i_3} \in [P_1], i_3 = (i_2 + l_{S(P_{i_2})}) \bmod p$. Again, if $l_{S(P_{i_1})} + l_{S(P_{i_2})} \equiv 0 \bmod p$, then $P_{i_3} = P_{i_1}$. To see this note that $i_3 = (i_1 + l_{S(P_{i_1})} + l_{S(P_{i_2})}) \bmod p$. If $l_{S(P_{i_2})} \equiv 0 \bmod p$, then $P_{i_3} = P_{i_2}$. If none of these two cases occur, then after one more round of T on P_{i_3}, T produces $P_{i_4}, i_4 = (i_3 + l_{S(P_{i_3})}) \bmod p$, etc.

Generally speaking, after n rounds of T on a word $P_{i_1} \in [P_1]$, T produces the word $P_{i_n} \in [P_1], i_n = (i_1 + l_{S(P_{i_1})} + \ldots + l_{S(P_{i_{n-2}})} + l_{S(P_{i_{n-1}})}) \bmod p$. If there is an m such that $l_{S(P_{i_m})} + l_{S(P_{i_{m+1}})} + \ldots + l_{S(P_{i_{n-1}})} \equiv 0 \bmod p, 0 < m < n$, then it must be the case that $P_{i_n} = P_{i_m}$. If this is not the case, then after one more round of T on P_{i_n} T produces the word $P_{i_{n+1}} \in [P_1]$.

It now easily follows that there is at least one periodic word $P_{i_j} \in [P_1]$ such that T reproduces P_{i_j} after n rounds of T on P_{i_j} and $n \leq p$. The reason for this is that every word P_{i_j} produced after j rounds on some word $P_{i_1} \in [P_1]$ is also in $[P_1]$. $\square$

An immediate consequence of Lemma 2 is that if a word $P_{i_1} \in [P_1]$ repeats itself after n rounds of T on P_{i_1} then there are at least n periodic words $P_{i_j} \in [P_1]$ that repeat themselves after n rounds, i.e., P_{i_1} plus the $n-1$ different words P_{i_j} produced from P_{i_1} after $1 < j \leq n$ rounds. Since no word of type Ib reproduces itself after one round, we thus also have that for words of type Ib, there are at least two words that reproduce themselves after at most p rounds.

Using Lemma 2, we can now prove:

Theorem 1 *Given a v-tag system T, a periodic word P_1 with period p of type Ib and the set of periodic words $[P_1]$ generated by P_1 and one of the words $P_{i_j} \in [P_1]$ that reproduces itself after n rounds, $2 \leq n \leq p$, then one can construct an infinite number of periodic words with different periods in T.*

Proof Given such a set $[P_1]$ and one of the words $P_{i_1} \in [P_1]$ that reproduce themselves after n rounds, $2 \leq n \leq p$, then the word $\mathbf{P_1} = P_{i_1} \overrightarrow{P}_{i_2} \ldots \overrightarrow{P}_{i_n} \overrightarrow{P}_{i_1}$, with each $P_{i_j} \vdash^{l_{\mathbf{S(P_{i_j})}}} P_{i_{j+1}}$, $P_{i_n} \vdash^{l_{\mathbf{S(P_{i_n})}}} P_{i_1}$ must also be a periodic word.

Indeed, we then get the following set of productions:

$$
\begin{array}{lll}
 & P_{i_1}\overrightarrow{P}_{i_2}\ldots\overrightarrow{P}_{i_{n-1}}\overrightarrow{P}_{i_n}\overrightarrow{P}_{i_1} & = \mathbf{P_1} \\
\vdash^{l_{\mathbf{S}(\mathbf{P_1})}} & P_{i_2}\overrightarrow{P}_{i_3}\ldots\overrightarrow{P}_{i_n}\overrightarrow{P}_{i_1}\overrightarrow{P}_{i_2} & = \mathbf{P_2} \\
\vdash^{l_{\mathbf{S}(\mathbf{P_2})}} & P_{i_3}\overrightarrow{P}_{i_4}\ldots\overrightarrow{P}_{i_1}\overrightarrow{P}_{i_2}\overrightarrow{P}_{i_3} & = \mathbf{P_3} \\
\vdots & \vdots & \vdots \\
\vdash^{l_{\mathbf{S}(\mathbf{P_{n-1}})}} & P_{i_n}\overrightarrow{P}_{i_1}\ldots\overrightarrow{P}_{i_{n-2}}\overrightarrow{P}_{i_{n-1}}\overrightarrow{P}_{i_n} & = \mathbf{P_n} \\
\vdash^{l_{\mathbf{S}(\mathbf{P_n})}} & P_{i_1}\overrightarrow{P}_{i_2}\ldots\overrightarrow{P}_{i_{n-1}}\overrightarrow{P}_{i_n}\overrightarrow{P}_{i_1} & = \mathbf{P_1}
\end{array}
$$

Note that every new production $\mathbf{P_i}$ is the result of a round on the previous production $\mathbf{P_{i-1}}$. It easily follows from these productions that the period $p_{\mathbf{P_1}}$ of $\mathbf{P_1}$ is:

$$(n+1)\sum_{j=1}^{n} l_{S(P_{i_j})} \tag{1}$$

Given the productions from $\mathbf{P_1}$, it is easily seen that given a word P_{i_1} with period p of type Ib that reproduces itself after $2 \leq n \leq p$ rounds, one can construct an infinite number of periodic words with different periods. Indeed, any word of the form $\mathbf{P}_1\overrightarrow{\mathbf{P}}_\mathbf{2}\ldots\overrightarrow{\mathbf{P}}_\mathbf{n}(\overrightarrow{\mathbf{P}}_\mathbf{1}\overrightarrow{\mathbf{P}}_\mathbf{2}\ldots\overrightarrow{\mathbf{P}}_\mathbf{n})^m\overrightarrow{\mathbf{P}}_\mathbf{1}$, with

$$\overrightarrow{\mathbf{P}}_\mathbf{j} = \overrightarrow{P}_{i_j}\overrightarrow{P}_{i_{j+1}}\ldots\overrightarrow{P}_{i_n}\overrightarrow{P}_{i_1}\ldots\overrightarrow{P}_{i_j}$$

is a periodic word of type Ib with period $p_{\mathbf{P_1}} \times (m+2)$. $\square$

Type II Here is an example of a periodic word of type II in Post's example. Note that Watanabe did not consider periodic words of this type. This is why he did not detect the following periodic word in Post's tag system (see Sec. 4.1):

$P_1 = \underline{0}10\underline{0}00\underline{0}00\underline{0}00\underline{1}10\underline{1}11\underline{0}11\underline{1}01\underline{0}01\underline{1}01\underline{1}10\underline{1}11\underline{0}10\underline{0}00$

$\vdash^{l_{S(P_1)}}$ $P_{15} = \underline{0}00\underline{0}00\underline{0}01\underline{1}01\underline{1}10\underline{1}00\underline{1}10\underline{1}00\underline{1}10\underline{1}11\underline{0}11\underline{1}01\underline{0}00\underline{0}$

$\vdash^{l_{S(P_{15})}}$ $P_{28} = \underline{0}00\underline{0}11\underline{0}11\underline{1}01\underline{1}10\underline{1}11\underline{0}11\underline{1}01\underline{1}10\underline{1}11\underline{0}10\underline{0}11\underline{0}10\underline{0}00$

$\vdash$ $P_{29} = \underline{0}11\underline{0}11\underline{1}01\underline{1}10\underline{1}11\underline{0}11\underline{1}01\underline{1}10\underline{1}11\underline{0}10\underline{0}11\underline{0}10\underline{0}00\underline{0}0$

$\vdash$ $P_{30} = \underline{0}11\underline{1}01\underline{1}10\underline{1}11\underline{0}11\underline{1}01\underline{1}10\underline{1}11\underline{0}10\underline{0}11\underline{0}10\underline{0}00\underline{0}00\underline{0}$

$\vdash$ $P_{31} = \underline{1}01\underline{1}10\underline{1}11\underline{0}11\underline{1}01\underline{1}10\underline{1}11\underline{0}10\underline{0}11\underline{0}10\underline{0}00\underline{0}00\underline{0}00$

$\vdash$ $P_{32} = \underline{1}10\underline{1}11\underline{0}11\underline{1}01\underline{1}10\underline{1}11\underline{0}10\underline{0}11\underline{0}10\underline{0}00\underline{0}00\underline{0}00\underline{1}10\underline{1}$

$\vdash$ $P_{33} = \underline{1}11\underline{0}11\underline{1}01\underline{1}10\underline{1}11\underline{0}10\underline{0}11\underline{0}10\underline{0}00\underline{0}00\underline{0}00\underline{1}10\underline{1}11\underline{0}1$

$\vdash$ $P_{34} = \underline{0}11\underline{1}01\underline{1}10\underline{1}11\underline{0}10\underline{0}11\underline{0}10\underline{0}00\underline{0}00\underline{0}00\underline{1}10\underline{1}11\underline{0}11\underline{1}01$

$\vdash$ $P_{35} = \underline{1}01\underline{1}10\underline{1}11\underline{0}10\underline{0}11\underline{0}10\underline{0}00\underline{0}00\underline{0}00\underline{1}10\underline{1}11\underline{0}11\underline{1}01\underline{0}0$

$\vdash$ $P_{36} = \underline{1}10\underline{1}11\underline{0}10\underline{0}11\underline{0}10\underline{0}00\underline{0}00\underline{0}00\underline{1}10\underline{1}11\underline{0}11\underline{1}01\underline{0}01\underline{1}01$

$\vdash$ $P_{37} = \underline{1}11\underline{0}10\underline{0}11\underline{0}10\underline{0}00\underline{0}00\underline{0}00\underline{1}10\underline{1}11\underline{0}11\underline{1}01\underline{0}01\underline{1}01\underline{1}10\underline{1}$

$\vdash$ $P_{38} = \underline{0}10\underline{0}11\underline{0}10\underline{0}00\underline{0}00\underline{0}00\underline{1}10\underline{1}11\underline{0}11\underline{1}01\underline{0}01\underline{1}01\underline{1}10\underline{1}11\underline{0}1$

$\vdash$ $P_{40} = \underline{0}11\underline{0}10\underline{0}00\underline{0}00\underline{0}00\underline{1}10\underline{1}11\underline{0}11\underline{1}01\underline{0}01\underline{1}01\underline{1}10\underline{1}11\underline{0}10\underline{0}$

$\vdash \qquad P_1 = \underline{0}10\underline{0}00\underline{0}00\underline{0}00\underline{1}10\underline{1}11\underline{0}11\underline{1}01\underline{0}01\underline{1}01\underline{1}10\underline{1}11\underline{0}10\underline{0}00$

P_1 is reproduced after exactly 40 computation steps. The example shows that for every one of the periodic words P_i, $l_{S(P_i)} < p$. This is also the case for the periodic words $P_i, 1 < i < 15, 15 < i < 28$ not shown here for the sake of brevity. There are two important observations to be made with respect to words of type II. First of all, even though several examples were found during Experiment 2 of words of type I with relatively long periods (for Post's tag system up to length 70), the very long periods found are typically of type II. There is a logical explanation for this. First note that in order to have very long periods of type I, one needs equally long periodic words. Since a bound was put on the size of the words produced in Experiment 1 (i.e. 15.000), the possibility of finding very long periods for words of type I was made impossible because of the specific set-up of this experiment. Secondly, it should be noted that the lengths of the periodic structures of periodic words of type II do not increase significantly for increasing periods. For example, periodic words of period 22302 in **T34** have periodic structures of lengths varying between about 35 and 100.

Since it must take at least two rounds for a periodic word of type II to reproduce itself (since $l_{S(P_i)} < p$) it is not possible to apply the method of type Ia to generate an infinite number of periodic words with the same period. However, it is possible to generate an infinite class of different periodic words with different periods given a periodic word of type II. This is proven by Lemma 3 (similar to Lemma 2) and Theorem 2 (similar to Theorem 1):

Lemma 3 *Given a tag system T, a periodic word P_1 of type II and the set of periodic words $[P_1]$ generated by P_1, then there is at least one $P_{i_1} \in [P_1]$ for which it takes n rounds of T on P_{i_1}, $2 \leq n \leq p$, to reproduce P_{i_1}.*

Proof The proof is almost identical to that of Lemma 2 and is left to the reader. $\square$

Theorem 2 *Given a v-tag system T, a periodic word P_1 with period p of type II, the set of periodic words $[P_1]$ generated by P_1 and one of the words $P_{i_1} \in [P_1]$ that is reproduced after n rounds of T on P_{i_1}, $2 \leq n \leq p$, then one can construct an infinite number of periodic words with different periods.*

Proof The proof is identical to that of Theorem 1. $\square$

The following theorem explains the observation that some tag systems seem only capable to produce even periods, as in the case of Post's tag system, or the fact that other tag systems are capable to, for example, produce products of 3 (for example **T34**) or 7 (for example **T35**).

Theorem 3 *For any v-tag system T with $l_{w_0}, l_{w_1}, \ldots, l_{w_{\mu-1}}$, the lengths of the appendants, and any word P that is periodic in T of type I or II with period p, then $p = n_0 + n_1 + ... + n_{\mu-1}$ where $\{n_0, n_1, ..., n_{\mu-1}\}$ is a solution to the equation:*

$$n_0 l_{w_0} + n_1 l_{w_1} + \ldots + n_{\mu-1} l_{w_{\mu-1}} = vp \tag{2}$$

Proof Given a v-tag system T with alphabet Σ of μ letters, with $l_{w_1}, l_{w_2}, .., l_{w_i}$, $i \leq \mu$ the lengths of the appendants and some word P_1 that is periodic in T with period p of type I or II. We evidently have that P_1 will be reproduced by T after p letters have been read and vp letters have been erased by T. Let S_1 be the word formed by all the vp letters erased, i.e.:

$$S_1 = a_1 a_2 \ldots . a_v a_{v+1} a_{v+2} \ldots a_{vp}$$

Now clearly, since P_1 is periodic, it must be the case that S_1 either reproduces itself after one round on S_1 (if P_1 is of type I) or that S_1 is generated piecewise (if P_1 is of type II) from the letters of S_1 in every periodic loop. It now easily follows that the number of times n_i each of the different letters $a_{1+jv}, 0 \leq j \leq p-1$ is read in S_1 must satisfy Equation (2). $\square$

Given Theorem 3 we can now explain why, for example, Post's tag system only produces words that are divisible by 2. Remember that for this tag system $l_{w_0} = 2, l_{w_1} = 4, v = 3$. Using eq. (2) we get:

$$2n_0 + 4n_1 = 3p$$

Since the left-handside of this equation must be an even number, it immediately follows that $p = 2n$ for some $n \in \mathbb{N}$.

Once the two types were detected, every one of the periodic words produced during the experiment were classified (with the help of the computer) as type I or II. The results are quite interesting in the sense that every one of the tag systems produced at least one periodic word of type II, while not all produced words of type I. This is an important difference between the different tag systems.

4.2.4 Experiment 3 – 6: Measuring "chaotic" behavior.

The three remaining experiments were used to study how unpredictable each of the 52 tag systems actually is by making use of certain statistical tools.

Experiment 3. Flipping coins As explained in Sec. 4.2.1, the 52 tag systems studied here have the property that $\#1 \times (l_{w_1} - v) + \#0 \times (l_{w_0} - v) = 0$. It then follows for each of these 52 tag systems that if the probability that the first letter of any word produced during an actual computation is a 0 resp. 1 is $\#1/(\#1 + \#0)$ resp. $\#0/(\#1 + \#0)$, then one expects the tag system to halt or become periodic. The purpose of Experiment 3 was to check what the actual probabilities are for the computations resulting from the initial words classified as "Immortals?" during Experiment 1. I.e., each of the 52 tag systems was rerun for 10^7 computation steps with two different sets of ten "Immortals?" found during Experiment 1. In each computation step a counter kept track of the number of times a 0 or a 1 is read by the tag system. These results allowed to measure the mean $\mu_{i,N}$, $i \in \{0, 1\}$, for the number of times a 0 resp. a 1 was read, where N is the size of the sample space.[8] The means were computed after 5.000.000 and after 10.000.000 computation steps in order to check whether they converge to some value or not. The results show for each of the tag systems that $\mu_{0,N}$ is always a bit smaller than the expected value $\#0/(\#1 + \#0)$ and thus $\mu_{1,N}$ is always a bit greater than $\#1/(\#1 + \#0)$. These results can be considered as a statistical explanation why the initial words classified as "Immortals?" had (not yet) resulted in a halt or periodicity after 10.000.000 computation steps. However, the results also indicate that $\mu_{0,N}$ and $\mu_{1,N}$ converge to their expected values $\#0/(\#1 + \#0)$ and $\#1/(\#1 + \#0)$ respectively. For example, in the case of Post's tag system, the means for the two different sets of 10 initial words computed after 5.000.000 computation steps are $\mu_{0,N} = 0,49938814$ and $\mu_{1,N} = 0,50061186$ for the first set and $\mu_{0,N} = 0,49925946$ and $\mu_{1,N} = 0,50074054$ for the second set, while the means computed for the two sets after 10.000.000 computation steps are $\mu_{0,N} = 0,49955107$ and $\mu_{1,N} = 0,50044893$ for the first set and $\mu_{0,N} = 0,49961642$ and $\mu_{1,N} = 0,50038358$ for the second. This means that the difference between 0.5 and $\mu_{0,N}$ resp. $\mu_{1,N}$ decreases with an increased number of computations steps. This indicates that the chances for a halt or periodicity increase.

Experiment 4. Sensitive dependence on initial words Experiment 4 was used to measure sensitive dependence on initial words. Sensitive dependence here means that one very small change in the initial word results in a non-linear change in the long-term behavior. In the experiment this change was, in a first run, a change of one letter in the initial word, and, in a second run, a change of the length of the initial word with one letter. The results of both runs show for each of the tag systems a high sensitive dependence on the initial words. This is considered as a sign of chaotic behavior [33], and thus indicates that these tag systems are indeed "unpredictable".

Experiment 5: Measuring randomness Experiment 5 checked whether

[8] The mean $\mu_{i,N} = \Sigma_{j=1}^{N} \frac{x_{i,j}}{N}$.

the distribution of the 0s and 1s read in the words produced from the initial conditions tentatively classified as “Immortals?” is random or not. In order to check this, DIEHARD, a battery of tests for randomness developed by Marsaglia was used [22]. This battery contains 12 different tests and is one of the standard batteries currently used. None of the tag systems passed every one of the tests. Except for two of the tag systems, all the tag systems passed at least some of the tests (about 3 on the average). There was only one tag system **T41** that passed 9 of the 12 tests. The two tag systems that failed every one of the tests are **T34** and **T1**, Post’s tag system. Another quick visual test verified this difference between, on the one hand, **T34** and **T1**, and, on the other hand, the remaining 50 tag systems.[9] The fact that Post’s tag system, despite its apparent unpredictable behavior cannot be considered random in the sense described here points at an important feature of this tag system.

Experiment 6: Measuring the entropy In a last experiment (Experiment 6) yet another classical tool for measuring unpredictability was used, i.e., Shannon’s information-theoretical entropy [39]. The entropy was computed by measuring for each combination C of length n ($2 \leq n \leq 10$) the probability that C occurs. By summing up these probabilities for a given n and normalizing the sum to 1.0 one gets the information-theoretical entropy. The results showed for each of the tag systems a high entropy, some were even very close to the maximum value 1.0, although there was a slight decrease for increasing n.

It is clear from experiments 3–6 that the 52 tag systems studied have certain heuristic properties that are often used in the literature as indicators of complexity. Still, given the results from Experiment 3 and to a certain extent 5, most of these tag systems cannot, by any means, be regarded as completely chaotic systems. Except for one tag system, most tag systems only pass about three of the tests for randomness. Furthermore, it is the fact that the chances of reading a 1 or a 0 deviate just that little bit from what one might expect, that makes it possible for words to keep going for millions of computation steps, at least, statistically speaking.

[9] This visual test is described in [33]. It concerns a quick visualization method of fractals called the chaos game, that needs a pseudo-random number generator in order to work. If the generator is not random, the resulting fractal image will be incomplete and biased.

5 Discussion

As becomes clear from the six experiments that were done on the 52 different tag systems, including Post's example, the experimental approach offers a lot of possibilities but it also has several limitations. The time it takes to set-up an experiment and to study the results is often in disproportion with the results one ultimately gets. In the end, most results from the experiments are heuristic in nature. They do not immediately lead to rigorous results like "tag system x is universal". Furthermore, any computer experiment is finite and one thus needs to implement certain limits. One consequence of this is that for tag systems like Post's example, there is the problem that one can always only show the beginning of a computation as long as a tag system has not halted or become periodic. As a consequence, one cannot know if the observations made on these first n computations steps are representative for what happens later on.

This does not mean that one should throw out the baby with the bath water. First of all, one should not forget that the experimental approach seems the best one available for now to study very small tag systems like Post's example. Von Neumann once said that for some problems, computer experiments are the only way out to build up an intuition of these problems, where intuition is a necessary prerequisite to make progress on the problem [44]. This is the first motivation to start with experimentation. Indeed, how can one build up an intuition of a certain problem, like the one offered by Post, if one does not have any idea of how this system behaves?

The fact that the experiments show that these tag systems behave quite unpredictably is indeed but a heuristic fact about a finite sample of the behavior of these tag systems. However, it does give an idea of how difficult proving these tag systems decidable might be, a fact that is also supported by some theoretical results on small tag systems. First of all, the class of tag systems proven decidable in TS(2,2) does not behave in the same way as the tag systems studied here, so the methods used in that proof cannot be applied here. Secondly, the reduction of the $3n + 1$-problem to a very small tag system is a further indication of the difficulties involved with very small tag systems. One typical kind of result from computer experiments is the formulation of conjectures on the basis of the observations made. Although it is very tempting to conjecture on the basis of the results discussed here that tag systems like Post's example have an undecidable reachability problem, it will not be conjectured here. The mere presence of complex behavior is in my opinion not enough to make the conjecture. [10] However, it does provide enough reason to do more research in this direction and to find more arguments in order to

[10] In the past I did make the conjecture [12]. However, my point of view has changed.

make such a conjecture and perhaps, to prove it.

The results from Experiment 3, a statistical experiment, are significant on another level: they illustrate that one should be careful if one draws conclusions related to the dynamics of a tag system on the basis of the rules underlying these dynamics.

The results from Experiment 1 indicate that for every one of the tag systems, most initial words result in a halt or periodicity very quickly. However, there are always some that seem to be able to keep going for millions of computation steps. This is another sign of the difficulty of these tag systems. The results suggest that a more detailed research (theoretical and experimental) on specific classes of initial words can lead to new interesting results. This is yet another feature of this approach: it can help to select possibly interesting approaches to tackle a given problem. Experiments can in a certain sense provide clues of how to tackle a given problem (or, how not to).

The more theoretically appealing results come from Experiment 2. Without this experiment, the two types of periodic words would most probably not have been detected, witness Watanabe's theoretical analysis (Sec. 4.1). The theoretical analysis on the results from the experiment has made it possible to explain certain observations and to prove some facts about these periodic types. The connection between tag systems and number theory is strengthened. Furthermore, these results suggest that more research on the periodic behavior of small tag systems might help to study their computational power. The fact that, on the one hand, one can do certain things with the periods (they can e.g. represent numbers), and, on the other hand, periodic words have a certain stability (they reproduce themselves), could be a way to find small tag systems that compute certain arithmetical functions or even to find small universal tag systems.

To summarize, even though one should always be extremely careful when drawing conclusions on the basis of computer experiments, one cannot neglect that they do result in progress in the domain of small tag systems. It is important not to lose sight of one of the main goals behind such experiments, i.e., to establish rigorous results. In this sense it is paramount to find a good balance between theory and experiment. The approaches are not opposite to but complement each other.

References

[1] Claudio Baiocchi, *Three small universal Turing machines*, Proc. 3rd International Conference on Machines, Computations, Universality (Berlin) (Yu. Rogozhin M. Margenstern, ed.), LNCS, vol. 2055, 2001, pp. 1–10.

[2] Allen H. Brady, *The determination of the value of Rado's noncomputable function σ for four-state Turing machines*, Mathematics of Computation **40** (1983), no. 162, 647–665.

[3] Alonzo Church, *An unsolvable problem of elementary number theory*, American Journal of Mathematics (1936), no. 58, 345–363, Also published in [7], 88–107.

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

[5] ———, *Universality of tag systems with $p = 2$*, Journal of the ACM **11** (1964), no. 1, 15–20.

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

[7] 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.

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

[9] ———, *Emil L. Post. His life and work*, 1994, in: [36], xi–xviii.

[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] ———, *Study of limits of solvability in tag systems*, Machines, Computations, and Universality. Fifth International Conference, MCU 2007 Orléans (Berlin) (J. Durand-Lose and M. Margenstern, eds.), LNCS, vol. 4664, Springer, 2007, pp. 170–181.

[12] ———, *Tracing unsolvability: A historical, mathematical and philosophical analysis with a special focus on tag systems*, Ph.D. thesis, University of Ghent, 2007.

[13] ———, *On the boundaries of solvability and unsolvability in tag systems. Theoretical and experimental results.*, The Complexity of Simple Programs. Proceedings 6th-7th December, 2008, Cork, Ireland (A. K. Seda N. Murphy T. Neary, D. Woods, ed.), 2008, pp. 63–76.

[14] ———, *Tag systems and Collatz-like functions*, Theoretical Computer Science **390** (2008), no. 1, 92–101.

[15] ———, *Solvability of the halting and reachability problem for binary 2-tag systems*, Fundamenta Informaticae **99** (2010), no. 4, 435–471.

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

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

[18] ———, *A question of numbers*, American Scientist **84** (1996), 10–14.

[19] Manfred 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., LNCS, vol. 2138, 2001, pp. 217–226.

[20] Shen Lin and Tibor Rádo, *Computer studies of Turing machine problems*, Journal of the ACM **12** (1965), no. 2, 196–212.

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

[22] George Marsaglia, *The Marsaglia random number CD-rom, with the Diehard battery of tests of randomness*, Department of statistics and supercomputer computations research institute, 1996.

[23] Sergei. J. Maslov, *On E. L. Post's 'Tag' problem. (in 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.

[24] Pascal Michel, *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] ———, *Size and structure of universal Turing machines using tag systems*, Proceedings Symposia Pure Mathematics, American Mathematical Society **5** (1962), 229–238.

[27] ———, *Computation. finite and infinite machines*, Series in Automatic Computation, Prentice Hall, Englewood Cliffs, New Jersey, 1967.

[28] Turlough Neary and Damien Woods, *Four small universal Turing machines*, Machines, Computations, and Universality. Fifth International Conference, MCU 2007 Orléans (J. Durand-Lose and M. Margenstern, eds.), vol. 4664, 2007, pp. 242–254.

[29] ———, *Small weakly universal Turing machines*, FCT 2009 - 17th International Symposium on Fundamentals of Computation Theory, Wroclaw, Poland (M. Kutylowski, M. Gebala, W. Charatonik (eds.), LNCS, vol. 5699, Springer, 2009, pp. 262–273.

[30] ———, *Four small universal Turing machines*, Fundamenta Informaticae **91** (2009), no. 1, 123–144.

[31] ———, *P-completeness of cellular automaton rule 110*, International Colloquium on Automata Languages and Programming (ICALP), LNCS, vol. 4051, 2006, pp. 132–143.

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

[33] Heinz-Otto Peitgen, Hartmut Jürgens, and Dietmar Saupe, *Chaos and fractals. New frontiers of science*, Springer Verlag, New York, 1992.

[34] Emil Leon Post, *Formal reductions of the general combinatorial decision problem*, American Journal of Mathematics **65** (1943), no. 2, 197–215.

[35] ______, *Absolutely unsolvable problems and relatively undecidable propositions - account of an anticipation*, 1965, in: [7], 340–433. Also published in [36].

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

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

[38] ______, *Small universal Turing machines*, Theoretical Computer Science **168** (1996), 215–240.

[39] Claude E. Shannon, *A mathematical theory of communication*, Bell System Technical Journal **27** (1948), 379–423 en 623–656.

[40] James B. Shearer, *Periods of strings (letter to the editor)*, American Scientist **86** (1996), 207.

[41] John Stillwell, *Emil Post and his anticipation of Gödel and Turing*, Mathematics Magazine **77** (2004), no. 1, 3–14.

[42] Klaus Sutner, *Cellular automata and intermediate degrees*, Theoretical Computer Science **296** (2003), no. 2, 365–375.

[43] Alan M. Turing, *On computable numbers with an application to the Entscheidungsproblem*, Proceedings of the London Mathematical Society (1936–37), no. 42, 230–265, A correction to the paper was published in the same journal, vol. 43, 1937, 544–546. Both were published in [7], 116–151.

[44] John von Neumann, *The general and logical theory of automata*, University of Illinois Press, Urbana, London, 1966.

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

[46] Shigeru Watanabe, *5-symbol 8-state and 5-symbol 6-state universal Turing machines*, Journal of the ACM **8** (1961), no. 4, 476–483.

[47] ______, *Periodicity of Post's normal process of tag*, [16], 1963, pp. 83–99.

[48] Stephen Wolfram, *Cellular automata and complexity. Collected papers*, Addison-Wesley, New York, 1994.

[49] ______, *A new kind of science*, Wolfram Media, Champaign, 2002.

[50] Damien Woods and Turlough Neary, *On the time complexity of 2-tag systems and small universal Turing machines*, Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, 2006, pp. 439–448.

[51] ______, *Small semi-weakly universal Turing machines*, Machines, Computations, and Universality. Fifth International Conference, MCU 2007 Orléans (J. Durand-Lose and M. Margenstern, eds.), LNCS, vol. 4664, Springer, 2007, pp. 303–315.

[52] ______, *Small semi-weakly universal Turing machines*, Fundamenta Informaticae **91** (2009), 161–177.