

Post's System of Tag — A Simple Discrete Nonlinear System

Peter R.J. Asveld

*Department of Computer Science, Twente University of Technology
P.O. Box 217, 7500 AE Enschede, The Netherlands*

Abstract – We consider an instance T_0 of Post's system of tag for which we show that it is nonlinear. Then we discuss some computer simulations and study the periodic behavior of T_0 .

1. Introduction

One of the oldest and simplest rewriting systems is Post's system of tag. Such a system T with $T = (\Sigma, d, P, \omega_0)$ consists of an *alphabet* (a finite set of symbols) Σ , a natural number d (the *deletion number* of T), a function $P: \Sigma \rightarrow \Sigma^*$ from symbols from Σ to strings over Σ , and an *input* or *initial string* ω_0 ($\omega_0 \in \Sigma^+$; Σ^+ denotes the set of all nonempty strings over Σ and $\Sigma^* = \Sigma^+ \cup \{\lambda\}$, where λ is the empty word, i.e., the string of length 0). Each Post's system of tag is deterministic and possesses a state space which consists of all strings over Σ . The way in which state changes take place — or, equivalently, how the string ω_{t+1} is obtained from the string ω_t ($t \geq 0$), denoted by $\omega_t \Rightarrow \omega_{t+1}$ — is as follows:

- If the left-most symbol of ω_t equals σ , then we append the string $P(\sigma)$ to the right end of ω_t . We denote the string obtained in this fashion by ω'_t .
- We delete the first d symbols from ω'_t ; this yields ω_{t+1} .

Post's systems of tag have been introduced in [7,8]; cf. [6] for an introduction. They are quite powerful, since they are able to simulate Turing-machine computations [5] and thus to perform every effective computation. However, in this note we restrict our attention to a single simple instance T_0 of such a system (§2). In §3 we show that this instance is nonlinear. Computer simulations and the periodic behavior of T_0 are discussed in §4 and §5, respectively. §6 consists of conclusions and an open problem.

2. Examples

We consider two instances of Post's system of tag and discuss their behavior. The first one happens to be linear whereas the second one turns out to be nonlinear; cf. §3.

Example 2.1. Let $T_1 = (\Sigma, d, P, \omega_0)$ with $\Sigma = \{a\}$, $d = 3$, $P(a) = a^2$ and $\omega_0 = a^4$. Then $aaaa \Rightarrow aaa \Rightarrow aa \Rightarrow a \Rightarrow \lambda$. Thus ω_0 vanishes in four steps: there is no rule to rewrite λ .

Slightly more general is the system $T_2 = (\Sigma, d, P, \omega_0)$ with $\Sigma = \{a\}$, $P(a) = a^k$ and $\omega_0 \in \Sigma^+$. If $|\omega|$ denotes the length of ω , then T_2 can be described by $|\omega_{t+1}| = |\omega_t| + k - d$, i.e., by a first-order linear difference equation, of which the solution results in $\omega_t = a^{(k-d)t} \omega_0$. Now for $d > k$, ω_t vanishes, i.e., $\omega_0 \Rightarrow^\tau \lambda$ with $\tau = \lceil |\omega_0| / (d-k) \rceil$; for $d = k$, ω_0 is a fixed point: $\omega_0 \Rightarrow^t \omega_0$ ($t \geq 0$); and for $d < k$, ω_0 explodes: $|\omega_t| \rightarrow \infty$ as $t \rightarrow \infty$. \square

Of the simple instances of Post's system of tag the following one — already studied by Post in 1921-1922; cf. [8] — is one of the simplest nontrivial cases; cf. [1,2,3,6,7,8,9,10]. The remaining part of this note is devoted to this instance.

Example 2.2. Consider $T_0 = (\Sigma, d, P, \omega_0)$ with $\Sigma = \{0, 1\}$, $d = 3$, $P(0) = 00$ and $P(1) = 1101$. Examples of its behavior are in Figure 1 for the input string ω_0 equal to 1000 and to 1001, respectively. \square

For an arbitrary Post's system of tag T we can distinguish the following types of behavior:

- (1) After τ steps we enter a periodic cycle with (minimal) period π : $\omega_{\tau+\pi} = \omega_\tau$. For T_0 with $\omega_0 = 1001$, we have $\tau = 3$ and $\pi = 2$; cf. Figure 1.
- (2) After τ steps we reach a fixed point ($\pi = 1$). T_0 has no fixed points other than λ .
- (3) After τ steps the string vanishes ($\pi = 0$). For T_0 strings 0^n vanish with $\tau = n$. See also Figure 1.
- (4) ω_0 explodes, i.e., $|\omega_t| \rightarrow \infty$ as $t \rightarrow \infty$. This is the only form of chaotic behavior for T ($\tau + \pi = \infty$). For T_0 no examples of this behavior are known up to now.

t	ω_t with $\omega_0 = 1000$	ω_t with $\omega_0 = 1001$
0	1000	1001
1	01101	11101
2	0100	011101
3	000	10100
4	00	001101
5	0	10100
6		001101
7		10100
8		...

Figure 1.

3. Nonlinearity of T_0

If Σ consists of a single symbol, then T is equivalent to a linear difference equation (Example 2.1). But when Σ contains more than one symbol, T becomes nonlinear. For T_0 we have the following result.

Proposition 3.1. T_0 is equivalent to the following nonlinear difference equation

$$u_{t+1} = \begin{cases} \lfloor (26 \cdot 2^{\text{lng}(u_t)} + u_t - 7)/8 \rfloor & \text{if } u_t \text{ is even,} \\ \lfloor (3 \cdot 2^{\text{lng}(u_t)} + u_t - 7)/8 \rfloor & \text{if } u_t \text{ is odd,} \end{cases}$$

where $\text{lng}(u_t)$ is the length of the 2-adic representation of u_t , i.e., if

$$u_t = \sum_{i=0}^n d_i \cdot 2^i$$

with $d_i \in \{1, 2\}$ ($0 \leq i \leq n$), then $\text{lng}(u_t) = n$.

Proof (sketch): Starting from T_0 we first change each 1 into 2 and each 0 into 1. In this way we obtain T_1 . Next we interchange ‘‘left’’ and ‘‘right’’, which results in T_2 . The effect of these manipulations on P and on a string ω_t are shown in the following table.

	T_0	T_1	T_2
P	$P(0) = 00, P(1) = 1101$	$P(1) = 11, P(2) = 2212$	$P(1) = 11, P(2) = 2122$
ω_t	001101	112212	212211

Finally, we interpret all strings over $\{1, 2\}$ as 2-adic representations, keeping in mind that 3 and 26 in decimal notation correspond to 11 and 2122, respectively. \square

In essence T_0 has the same structure as a younger but more famous simple nonlinear system, viz.

The $3x + 1$ problem. (Also: Ulam’s problem, Collatz’s problem, the Syracuse problem, etc. [4]):

$$v_{t+1} = \begin{cases} v_t/2 & \text{if } v_t \text{ is even,} \\ 3 \cdot v_t + 1 & \text{if } v_t \text{ is odd.} \end{cases} \quad \square$$

The $3x + 1$ conjecture. For each initial value v_0 ($v_0 \in \mathbb{N}$) the ultimate behavior of v_t is the 3-cycle $\dots \Rightarrow 4 \Rightarrow 2 \Rightarrow 1 \Rightarrow 4 \dots$ (*). \square

But comparing T_0 and the $3x + 1$ problem yields the following two differences. (i) The periodic behavior of T_0 is much richer than the single 3-cycle (*); cf. §4 and §5. (ii) Each state of the $3x + 1$ problem has a predecessor: for each n in \mathbb{N} there exists a number m in \mathbb{N} (viz. $m = 2 \cdot n$) such that $m \Rightarrow n$. On the other hand T_0 possesses so-called *Garden-of-Eden states*, i.e., states that can only exist at time $t = 0$ (viz. all strings ending in 11, 10, 001 or 0101).

4. Computer Simulations of T_0

We performed two series of computer simulations of T_0 's behavior. Both series were obtained by varying the initial string ω_0 . First we consider small inputs, i.e., $|\omega_0| \leq 15$. There is no need to consider all these small strings, since many of them show the same ultimate behavior. We introduce the following obvious equivalence relation.

Definition 4.1. Let d be a natural number. A position i in a string ω is called *critical* if $i \bmod d = 1$. Two strings ω and ω' are *equivalent* — notation: $\omega \equiv \omega'$ — if $|\omega| = |\omega'|$, and they are equal at their critical positions. □

Observation 4.2. Each equivalence class $[\omega]$ with respect to \equiv contains a minimal element ρ with respect to lexicographical order. This representative ρ is obtained from ω by changing the 1's at the non-critical positions in ω into 0's. □

Example 4.3. Let $d = 3$ and $\omega = 1010111$. Then $\omega \equiv 1110111$ and $\omega \equiv 1100011$, but $\omega \not\equiv 1000000$. The representative ρ for $[\omega]$ is 1000001. □

In the first series of simulations (small inputs) we take all nonempty strings over $\{0, 1\}$ of length at most 15. Next we determine the 186 equivalence classes with respect to \equiv and the corresponding representatives $\rho(n)$ ($1 \leq n \leq 186$). E.g. $\rho(1) = 0$, $\rho(2) = 1$, $\rho(3) = 00$, $\rho(4) = 10$, $\rho(5) = 000$, and $\rho(186) = 100100100100100$. The corresponding instances $T_0(n) = (\Sigma, d, P, \omega_0(n))$ with $\omega_0(n) = \rho(n)$ of T_0 are indexed by n with $1 \leq n \leq 186$.

The second series of simulations (worst case inputs) is easier to describe. The instances are $T_0(m) = (\Sigma, d, P, \omega_0(m))$ with $\omega_0(m) = (100)^m$ and will be indexed by m with $1 \leq m \leq 200$. The fact that (at least) the first $m + 1$ steps are in the direction of an explosion accounts for the “worst case” character. Note that the intersection of these two series consists of $m = 1, 2, 3, 4, 5$ or, equivalently, $n = 6, 18, 43, 90, 186$ respectively.

Due to space limitations we only give some examples of the results of these simulations; a more complete overview will appear in [3]. Generally spoken, little can be said with respect to the initial behavior ($0 \leq t \leq \tau$); regular patterns only turn up in the periodic behavior ($t \geq \tau$). See also §5.

n	$\rho(n)$	$\tau(n)$	$M(n)$	$T(n)$	n	$\rho(n)$	$\tau(n)$	$M(n)$	$T(n)$
130	00000010010010	422	56	108	159	0000001000000000	17	15	0
131	00010000000000	16	14	0	160	000000100000100	19	15	0
133	00010000010000	18	14	0	164	000100000000100	21	15	0
136	00010010000010	24	15	3	165	000100000100000	23	15	0
141	10000000010000	20	15	1	166	000100000100100	419	56	105
143	10000010000000	22	15	1	168	000100100000100	417	56	103
144	10000010000010	420	56	106	173	100000000100000	421	56	107
145	10000010010000	418	56	104	176	100000100000100	415	56	101
149	10010000010000	416	56	102	178	100000100100100	413	56	99
150	10010000010010	414	56	100	181	100100000100000	27	18	7
154	10010010010010	412	56	98	186	100100100100100	411	56	97

Table 1. $\omega_0(n) = \rho(n)$ ($130 \leq n \leq 186$)

In §2 we already mentioned that all inputs of the form 0^n vanish in n steps. But there are many more inputs that vanish ultimately; see Tables 1 and 2 where M is the maximum string length encountered on the interval $[0, \tau]$ and T is the first moment at which this maximum length occurs. Some of these inputs ($m = 106, 160$ and 195) need a considerable time to disappear.

Next we turn to periodic cycles; see also §5. In the sequel we use $A = P(0) = 00$, $B = P(1) = 1101$, $x = AB$ and $y = BBAA$ as abbreviations. Examples of periodic cycles are in Table 3 of which the last column contains a string over $\{A, B, x, y\}$ preceded by its position in the cycle. Clearly, this string completely determines the cycle. So for $m = 24, 33, 109$ and 167 we have the same cycle. The same cycle may also be entered at different points; see $m = 7$ and 16 .

π_i and $|p_i| \equiv 0 \pmod{3}$, then $p_1 p_2$ is periodic with period π and $\pi \leq \pi_1 + \pi_2$. But sometimes we have $\pi = \pi_1 + \pi_2$, e.g., for xy or yx we have $\pi = 6$.

Proposition 5.2. *For each k ($k \geq 1$), there exists a string that generates a cycle of period $2 \cdot k$.*

Proof: For $k = 1, 2, 3$ we take x, y and xy , respectively. Otherwise in case $2 \cdot k = 6 \cdot i + 2$, $2 \cdot k = 6 \cdot i + 4$ or $2 \cdot k = 6 \cdot i + 6$ we take $x(xy)^i$, $(xy)^i y$ and $x(xy)^i y$, respectively. E.g., for $\pi = 8$ we have

$$xxy \Rightarrow^2 xyx \Rightarrow^2 yxx \Rightarrow^4 xxy \Rightarrow \dots \quad \square$$

Observation 5.3. *If a string p of which the length is a multiple of three is periodic with period π , then each string p^n ($n \geq 1$) is also periodic with period π .* \square

Case 2: The periodic cycle does not contain a string of which the length is a multiple of three. Now we only have a single period.

Proposition 5.4. *For each n ($n \geq 0$), the string $BAAx(yx)^n$ is periodic with $\pi = 6$.*

Proof: For $n \geq 1$, $BAAx(yx)^n = BAAx(yx)(xy)^{n-1} \Rightarrow^2 Axyx(yx)^{n-1}BB = AABBBAAx(yx)^{n-1}BB \Rightarrow^4 BAAx(yx)^{n-1}BBAAAB = BAAx(yx)^n$ and for $n = 0$, $BAAx \Rightarrow^2 AxBB = AABBB \Rightarrow^4 BAAAB = BAAx$. \square

Comparing this proof and Theorem 5.1 yields $p = BAAxB$, $q = BAAx(yx)^{n-1}$, $r = yx$, $u = BAAx$, $v = B$ for $n \geq 1$, and $p = BAAxB$, $q = B^{-1}$, $r = yx$, $u = BAAx$ and $v = B$ for $n = 0$. The “strange” value for q in case $n = 0$ corresponds to a substring B that is both generated and consumed during a single period. For Case 2 we have no analogue of 5.3.

In [10] it is claimed that there are no other periodic cycles than those mentioned above.

6. Conclusions and Open Problem

Simple discrete nonlinear systems like T_0 and the $3x + 1$ problem are still as intractable and intriguing as ever. We showed that the periodic behavior of T_0 is more complicated than the simple 3-cycle in the $3x + 1$ conjecture; cf. §4 and §5. For T_0 we have, apart from vanishing inputs, that for each k ($k \geq 1$), there exist infinitely many periodic cycles with period $2 \cdot k$. And there is still the possibility of “exploding” input strings.

Figure 2.

Figure 3.

A suitable candidate for this last possibility is the case $m = 110$ for which the computer simulation did not yet produce a final answer. After $6864.41 \cdot 10^9 = 6.86441 \cdot 10^{12}$ simulation steps we only know that $M(110) \geq 17\,144\,686$ and $T(110) \geq 5\,395\,443\,171\,968$. See Figures 2-4 where we plotted the string length as function of the number of steps (in 10^9 's). This is very long computation indeed, viz. 12.4 MIPSyear. (A MIPSyear is the number of instructions performed on a 1 MIPS machine during a

Figure 4.**Figure 5.**

year, i.e., $1 \text{ MIPSyear} = 3.1536 * 10^{13}$ instructions). So on a DEC μ VAX-2000 (1 MIPS machine), a SUN 3/60 (2.5 MIPS), a SUN SPARC (12 MIPS) and a SUN SPARC+ (16 MIPS) it takes 12.4, 5.0, 1.0 and 0.8 year, respectively. Per hour we simulate on these machines $0.09 * 10^9$, $0.22 * 10^9$, $1.10 * 10^9$ and $1.35 * 10^9$ steps, respectively. In case T_0 will explode for $m = 110$, then it is a very slow (or “soft”) explosion since it takes 2000 to 3000 steps in the average to increase the string length by 1. Cf. Figure 5 where we show $M(t)/t$ as function of t ; $M(t)$ being the maximum string length reached on the interval $[0, t]$.

References

1. P.R.J. Asveld: On a Post's system of tag, *Bull. Europ. Assoc. Theor. Comp. Sci.* No. 36 (1988) 96-102.
2. P.R.J. Asveld: A simple system with chaotic behavior, *Memoranda Informatica* 88-51 (Dept. of Comp. Sci., Twente University of Technology). Also in: D. Kleima et al (Eds.): *Liber Amicorum for E.W. Gröneveld* (1988), Dept. of Elec. Eng., Twente University of Technology, Enschede, The Netherlands.
3. P.R.J. Asveld: On Post's system of tag with $0 \rightarrow 00$ and $1 \rightarrow 1101$ as productions, *Memoranda Informatica* (in preparation), (Dept. of Comp. Sci., Twente University of Technology).
4. J.C. Lagarias: The $3x + 1$ problem and its generalizations, *Amer. Math. Monthly* **92** (1985) 3-23.
5. M.L. Minsky: Recursive unsolvability of Post's problem of “tag” and other topics in the theory of Turing machines, *Annals of Math.* **74** (1961) 437-455.
6. M.L. Minsky: *Computation — Finite and Infinite Machines* (1972), Prentice-Hall, Englewood Cliffs, N.J.
7. E.L. Post: Formal reductions of the general combinatorial decision problem, *Amer. J. Math.* **65** (1943) 197-215.
8. E.L. Post: Absolutely unsolvable problems and relatively undecidable propositions — account of an anticipation, pp. 340-433 in: M. Davis: *The Undecidable — Basic Papers on Undecidable Propositions, Unsolv-able Problems and Computable Functions* (1965), Raven, New York.
9. H. Wang: Tag systems and lag systems, *Math. Ann.* **152** (1963) 65-74.
10. S. Watanabe: Periodicity of Post's normal process of tag, Proc. of the Symp. on Math. Theory of Automata 1962, Polytechnic Press, Brooklyn, N.Y. (1963) pp. 83-99.