On a Post's System of Tag*

Peter R.J. Asveld

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

Abstract – We investigate instances of Post's system of tag with alphabet $\{0,1\}$, deletion number $n=3$, set of productions $\{0 \rightarrow 00, 1 \rightarrow 1101\}$, and initial strings of the form $(100)^m$ where m ranges from 1 to 32. Some other initial strings from the set $\{000,100\}^+$ are considered as well.

1. Introduction

One of the oldest rewriting systems is Post's system of tag [4, 5]. Informally, an instance of this rewriting system may be described as follows; cf. [3] p. 267.

Given an initial string ω_0 consisting of 0's and 1's, examine the first letter of ω_0. If it is equal to 0, append 00 to the right of ω_0, and delete the first n symbols of this intermediate string, yielding the string ω_1. If the first letter is equal to 1, now append 1101 to the right and delete the first n letters too. The resulting string is also denoted by ω_1. Perform the same procedure to ω_1, yielding the string ω_2, which in turns yields ω_3, and so on.

For instance, taking $n = 3$ and $\omega_0 = 000100000$ results in the following sequence, which vanishes after 13 steps; cf. Figure 1.

t	ω_t
0	000100000
1	10000000
2	000001101
3	00110100
4	1010000
5	00001101
6	0110100
7	010000
8	00000
9	0000
10	000
11	00
12	0
13	

Figure 1.

But in case $\omega_0 = 100000100$ with $n = 3$ the sequence becomes infinite, since $\omega_{12} = \omega_{18} = \omega_{24} = \omega_{30} = \cdots$; cf. Figure 2.

Formally, Post's system of tag T consists of an alphabet Σ, a natural number n – called the deletion number –, a finite set P of productions, and an initial string ω_0 over Σ. The set P of production satisfies the following two conditions:

(1) The left-hand sides of all productions in P have the same length.

---

* This paper appeared in *Bull. Europ. Assoc. for Theoret. Comput. Sci.* No. 36 (1988) 96-102.

t	ω_t
0	100000100
1	0001001101
2	100110100
3	1101001101
4	10011011101
5	110111011101
6	1110111011101
7	01110111011101
8	1011101110100
9	11011101001101
10	111010011011101
11	0100110111011101
12	011011101110100
13	01110111010000
14	1011101000000
15	11010000001101
16	100000011011101
17	0000110111011101
18	011011101110100
19	01110111010000

Figure 2.

(2) The right-hand side of a production depends only on the first letter of the associated left-hand side.

These requirements guarantee that each string has at most one successor string or, equivalently, the rewriting system is deterministic or monogenic as it is called in [3]. For a precise formulation of this problem in terms of a Post's normal canonical system we refer to [3] again.

As examples, consider the systems discussed above; they are equal to $T = (\Sigma, n, P, \omega_0)$ where $\Sigma = \{0,1\}$, $n = 3$, $P = \{0 \rightarrow 00,\ 1 \rightarrow 1101\}$, whereas $\omega_0 = 000100000$ and $\omega_0 = 100000100$, respectively. Other examples can be found in [3], where Minsky remarks

> ''The reader might try, for example, [$\omega_0 =$] $(100)^7$, that is, 100100100100100100100, but he will almost certainly give up without answering the question: ''Does this string, too, become repetitive?'' In fact the answer to the more general question ''Is there an effective way to decide, for any string *S*, whether this process will ever repeat when started with *S*?'' is still unknown. 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. . . . While the solvability of the (00,1101) problem is still unsettled (some partial results are discussed by Watanabe [6]), it is now known that *some* problems of the same general character are unsolvable.'' [3] pp. 267-268; cf. also [2].

The aim of the present note is to show that the instance of this Post's System of tag with initial string $(100)^7$ is not that ''bad''. It compares, for instance, with the cases $(100)^m$ where m equals 10, 12, 18 or 28; see Table 1. In the range $1 \le m \le 32$, the ''worst'' case is $m = 24$: the sequence becomes repetitive after 4346269 steps, the length of the first string that occurs twice in this sequence is 37. But before entering a cycle with length 6 of relatively short strings, really long strings do occur in this sequence, viz. strings with length up to 4432. Note also the case $m = 14$, in which ω_t vanishes after 37912 steps.

m	$H(m)$	$\tau(m)$	$\pi(m)$	$\lvert\omega_{\tau(m)}\rvert$	$M(m)$	$T(m)$
1	∞	4	2	5	6	3
2	∞	15	6	15	16	14
3	∞	10	6	15	16	9
4	∞	25	6	19	22	16
5	411	411	0	0	56	97
6	∞	47	10	31	34	34
7	∞	2128	28	85	176	1293
8	∞	853	6	37	76	400
9	∞	372	10	31	62	91
10	∞	2805	6	37	208	1734
11	∞	366	6	55	62	49
12	∞	2603	6	37	208	1532
13	703	703	0	0	68	51
14	37912	37912	0	0	768	18168
15	∞	612	6	91	104	271
16	∞	127	28	85	88	78
17	∞	998	10	31	106	395
18	∞	2401	6	127	224	674
19	∞	1200	10	31	146	265
20	∞	623	6	33	134	260
21	∞	5280	6	37	226	2701
22	1778	1778	0	0	172	1068
23	∞	1462	6	37	132	143
24	∞	4346269	6	37	4432	935110
25	4129	4129	0	0	206	2949
26	∞	3241	6	73	232	1664
27	∞	7018	6	73	378	2781
28	∞	3885	6	163	206	2874
29	∞	14632	6	55	432	9883
30	∞	7019	6	19	380	3186
31	∞	4564	6	73	208	1313
32	∞	4277	52	157	290	996

Table 1.

Needless to emphasize that this note does not contain ‘‘a theory for this problem’’ either. But we used a computer ‘‘for clerical aid in studying examples’’ with initial strings of the form $(100)^m$ where $1 \le m \le 32$, and of the form $\omega_0 \in \{000, 100\}^+$ with $3 \le |\omega_0| \le 12$.

2. Results

First, we consider the family of systems $T_m = (\Sigma, n, P, \omega_0(m))$, where $\omega_0(m) = (100)^m$ with $m \geq 1$, whereas Σ, n and P are as in the examples of the previous section. Note that (some of) the 0's in $\omega_0(m)$ may be replaced by 1's without affecting the ultimate behavior of the sequence. The elements of the sequence defined by T_m are denoted by $\omega_0(m)$, $\omega_1(m)$, ..., $\omega_t(m)$, ... Initial strings of the form $(100)^m$ constitute the ''worst case'' in the sense that the first m (or actually, even the first $m+2$) steps in the rewriting process are length-increasing steps.

In order to describe the behavior of such sequences the following concepts turn out to be useful. First, we consider the time $H(m)$ at which the string vanishes,

$$H(m) = \min\{t \mid \omega_t(m) = \lambda\}.$$

(We use λ to denote the empty string). In case the string never vanishes $H(m)$ is taken equal to ∞. If the string does not ultimately vanish, then it might become repetitive with *period* $\pi(m)$, and *threshold* $\tau(m)$,

$$\pi(m) = \min\{p \mid \exists t \in \mathbb{N}: \omega_t(m) = \omega_{t+p}(m)\},$$

$$\tau(m) = \min\{t \mid \omega_t(m) = \omega_{t+\pi(m)}(m)\}.$$

Finally, $M(m)$ denotes the maximum length of the string in the sequence

$$M(m) = \max\{|\omega_t(m)| \mid t \geq 0\},$$

and $T(m)$ is the first time that a string of maximum length occurs

$$T(m) = \min\{t \mid |\omega_t(m)| = M(m)\}.$$

Note that if a string does not vanish and it does not become repetitive, we have $M(m) = \infty$. On the other hand if it does vanish we have $\tau(m) = H(m)$, $\pi(m) = 0$ and $|\omega_t(m)| = 0$ for each $t \geq \tau(m)$.

In Table 1 the values of $H(m)$, $\tau(m)$, $\pi(m)$, $|\omega_{\tau(m)}|$, $M(m)$, and $T(m)$, are displayed for $m = 1, 2, \cdots, 32$. Table 2 contains the corresponding values of $\omega_{\tau(m)}$. Finally, we consider initial strings ω_0 over $\{A, B\}$ with $A = 000$ and $B = 100$, whereas $3 \leq |\omega_0| \leq 12$. The results for these initial strings are mentioned in Table 3 and Table 4.

3. Concluding Remarks

Except for the case $m = 24$ (Table 1) all the results have been obtained in a straightforward way. Viz. it is easy to write a small Pascal program to generate the first few thousand strings of a sequence. A file containing these strings can be sorted suppressing all but one in each number of equal strings by means of the UNIX* `sort -u` command. (In case the file is very long we first ought to split it, sort the subfiles, and finally merge the sorted subfiles). Then the word count command `wc -l` yields the position where the first repetition in the sequence occurs. By inspection in the neighborhood of this point all relevant information can be obtained apart from the values of M and T. But it is easy to write a separate program to compute these values.

However, apart from determining of the values of M and T, this obvious approach does not work in case m is equal to 24 unless you have hundreds (or thousands?) of megabytes as well as enormous amounts of computing time (days or weeks?) available. Instead we simply counted the lengths of all ω_t in a variable of type `array[1..4432] of integer` (remember that $M(24) = 4432$), while we let t range from 0 to 10^7 and to 2.10^7, respectively (Of course any other

---

* UNIX™ is a trademark of AT&T/Bell Laboratories.

m	$\omega_{\tau(m)}$
1	10100
2	011011101110100
3	011011101110100
4	0000011011101110100
5	λ
6	0001101110100000011011101110100
7	00000110111011101000011011101000000110111011101000000110111011101000000011 011101110100
8	0000011011101110100000011011101110100
9	0001101110100000011011101110100
10	0000011011101110100000011011101110100
11	00000110111011101000000110111011101000000110111011101100
12	0000011011101110100000011011101110100
13	λ
14	λ
15	0000011011101110100000011011101110100000011011101110100000011011101110100 000011011101110100
16	00011011101000000110111011101000000110111011101000000110111011101000000011 011101110100
17	0001101110100000011011101110100
18	0000011011101110100000011011101110100000011011101110100000011011101110100 00001101110111010000001101110111010000001101110110100
19	0001101110100000011011101110100
20	01101110111010000001101110110100
21	0000011011101110100000011011101110100
22	λ
23	0000011011101110100000011011101110100
24	0000011011101110100000011011101110100
25	λ
26	0000011011101110100000011011101110100000011011101110100000011011101110100
27	0000011011101110100000011011101110100000011011101110100000011011101110100
28	0000011011101110100000011011101110100000011011101110100000011011101110100 0000110111011101000000110111011101000000110111011101000000110111011101000 00011011101110100
29	00000110111011101000000110111011101000000110111011101100
30	0000011011101110100
31	0000011011101110100000011011101110100000011011101110100000011011101110100
32	00011011101000000110111011101000000110111011101000000110111011101000000011 0111011101000000110111011101000000110111011101000000110111011101000000110 11101110100

Table 2.

two initial intervals that are long enough will also do). These two array values are equal except for the entries with indices 35, 36, 37 and 38; see Table 5. Since the cycle can only be entered by

ω_0	$H(\omega_0)$	$\tau(\omega_0)$	$\pi(\omega_0)$	$\lvert\omega_{\tau(\omega_0)}\rvert$	$M(\omega_0)$	$T(\omega_0)$
A	3	3	0	0	3	0
B	∞	4	2	5	6	3
AA	6	6	0	0	6	0
AB	∞	2	2	6	6	0
BA	∞	17	6	15	16	16
BB	∞	15	6	15	16	14
AAA	9	9	0	0	9	0
AAB	11	11	0	0	9	0
ABA	13	13	0	0	9	0
ABB	∞	14	6	15	16	13
BAA	∞	16	6	15	16	15
BAB	∞	12	6	15	16	11
BBA	∞	24	6	15	16	23
BBB	∞	10	6	15	16	9
AAAA	12	12	0	0	12	0
AAAB	∞	19	6	15	16	18
AABA	14	14	0	0	12	0
AABB	∞	23	6	15	16	22
ABAA	∞	21	6	15	16	20
ABAB	∞	4	2	12	12	0
ABBA	∞	13	6	15	16	12
ABBB	∞	9	6	15	16	8
BAAA	∞	25	6	15	16	24
BAAB	∞	11	6	15	16	10
BABA	∞	29	6	19	22	20
BABB	∞	27	6	19	22	18
BBAA	∞	3	4	13	14	2
BBAB	∞	7	6	15	16	6
BBBA	420	420	0	0	56	106
BBBB	∞	25	6	19	22	16

Table 3.

rewriting a string of length 34 or 39, we wrote a separate program to determine the last time a string of length 34 or 39 occurred in these intervals. These facts happen at 4346238 and at 4346255, respectively. Closer inspection of the neighborhood of this latter point finally yields the values of $\tau(24)$, $\pi(24)$, $\omega_{\tau(24)}$ and $\lvert\omega_{\tau(24)}\rvert$.

References

1. M. Davis: *The Undecidable – Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions* (1965), Raven, New York.

ω_0	$\omega_{\tau(\omega_0)}$	ω_0	$\omega_{\tau(\omega_0)}$
A	λ	*AAAA*	λ
B	10100	*AAAB*	011011101110100
AA	λ	*AABA*	λ
AB	001101	*AABB*	011011101110100
BA	011011101110100	*ABAA*	011011101110100
BB	011011101110100	*ABAB*	001101001101
AAA	λ	*ABBA*	011011101110100
AAB	λ	*ABBB*	011011101110100
ABA	λ	*BAAA*	011011101110100
ABB	011011101110100	*BAAB*	011011101110100
BAA	011011101110100	*BABA*	0000011011101110100
BAB	011011101110100	*BABB*	0000011011101110100
BBA	011011101110100	*BBAA*	0001101110100
BBB	011011101110100	*BBAB*	011011101110100
		BBBA	λ
		BBBB	0000011011101110100

Table 4.

interval	35	36	37	38
$0 - 10^7$	942303	1884594	1884592	942301
$0 - 2.10^7$	2608969	5217927	5217926	2608968

Table 5.

2. 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.
3. M.L. Minsky: *Computation – Finite and Infinite Machines* (1972), Prentice-Hall, Englewood Cliffs, N.J.
4. E.L. Post: Formal reductions of the general combinatorial decision problem, *Amer. J. Math.* **65** (1943) 197-215.
5. E.L. Post: Absolutely unsolvable problems and relatively undecidable propositions – account of an anticipation, pp. 340-433 in [1].
6. S. Watanabe: Periodicity of Post's normal process of tag, Proceedings of the Symposium on Mathematical Theory of Automata 1962, Polytechnic Press, Brooklyn, N.Y. (1963) pp. 83-99.