

Classifying rotationally-closed languages having greedy universal cycles

Joseph DiMuro

Department of Mathematics and Computer Science
Biola University
La Mirada, CA, U.S.A.

`joseph.dimuro@biola.edu`

Submitted: Jun 14, 2018; Accepted: Feb 4, 2019; Published: Mar 8, 2019

© The author. Released under the CC BY-ND license (International 4.0).

Abstract

Let $\mathbf{T}(n, k)$ be the set of strings of length n over the alphabet $\Sigma = \{1, 2, \dots, k\}$. A universal cycle for $\mathbf{T}(n, k)$ can be constructed using a greedy algorithm: start with the string k^n , and continually append the least symbol possible without repeating a substring of length n . This construction also creates universal cycles for some subsets $\mathbf{S} \subseteq \mathbf{T}(n, k)$; we will classify all such subsets that are closed under rotations.

Mathematics Subject Classifications: 05A05

1 Introduction

Let $\mathbf{T}(n, k)$ be the set of strings of length n over the alphabet $\Sigma = \{1, 2, \dots, k\}$. Given a subset $\mathbf{S} \subseteq \mathbf{T}(n, k)$, we will be interested in finding a “universal cycle” for \mathbf{S} : that is, a string of length $|\mathbf{S}|$ where each element of \mathbf{S} occurs exactly once in that string when it is viewed as a cycle.

For example, we could consider the subset $\mathbf{S}_1 \subseteq \mathbf{T}(3, 5)$ consisting of ascending strings and their rotations. That is,

$$\mathbf{S}_1 = \{123, 124, 125, 134, 135, 145, 231, 234, 235, 241, 245, 251, 312, 341, 342, \\ 345, 351, 352, 412, 413, 423, 451, 452, 453, 512, 513, 514, 523, 524, 534\}$$

Below is a universal cycle for \mathbf{S}_1 . To make it easier to locate certain strings in the cycle, the string 534 is repeated at both the start and the end of the cycle. (The 534 is put in parentheses at the start as a reminder that the 534 is repeated.)

$$(534)123124134234512513514523524534$$

This universal cycle was constructed using a greedy algorithm: after choosing 534 as the starting string, each subsequent digit was chosen by looking for the smallest digit that could be chosen without duplicating any length-3 substrings (except for 534 itself, to conclude the cycle).

As another example, consider the subset $\mathbf{S}_2 \subseteq \mathbf{T}(2, 9)$ consisting of all two-digit strings where both the string and its reverse yield composite numbers in base 10. For example, $43 \notin \mathbf{S}_2$ because 43 is prime; $34 \notin \mathbf{S}_2$ because 34 is the reverse of a prime number.

$$\begin{aligned} \mathbf{S}_2 = \{ & 12, 15, 18, 21, 22, 24, 25, 26, 27, 28, 33, 36, 39, 42, 44, 45, \\ & 46, 48, 49, 51, 52, 54, 55, 56, 57, 58, 62, 63, 64, 65, 66, 68, \\ & 69, 72, 75, 77, 78, 81, 82, 84, 85, 86, 87, 88, 93, 94, 96, 99 \} \end{aligned}$$

If we try to construct a universal cycle for \mathbf{S}_2 using a greedy algorithm starting from 99, here's what we get:

$$(99)33621224251526394454648182728496556685758699$$

Unfortunately, this cycle isn't quite universal: 77, 78, 87, and 88 are missing. This raises the question: can we find necessary and sufficient conditions on $\mathbf{S} \subseteq \mathbf{T}(n, k)$ so that a universal cycle for \mathbf{S} can be generated from a greedy algorithm? In this paper, we will find such conditions, under the assumption that \mathbf{S} is closed under rotations.

2 Results

Some notational conventions: when we are working with a particular set $\mathbf{T}(n, k)$, we will use α and β (possibly with subscripts) to represent strings in $\mathbf{T}(n, k)$. Other Greek letters (like γ) will represent strings in $\mathbf{T}(m, k)$ for some $m \leq n$. (This includes the possibility of γ being an empty string.) Latin letters (a, b , etc.) will represent individual elements of $\{1, \dots, k\}$. We will sometimes use exponential notation to write strings with repetitions in a shorter form. A couple of examples: 2^53 represents the string 222223, and $4(21)^3$ represents the string 4212121.

Given a set $\mathbf{S} \subseteq \mathbf{T}(n, k)$ (where $\mathbf{S} \neq \emptyset$), let $\alpha \in \mathbf{S}$. Define the string $\text{Greedy}_\alpha(\mathbf{S})$ as follows: let $\beta_0 = \alpha$, and having defined β_0 through $\beta_j = a\gamma$, let $\beta_{j+1} = \gamma a_{j+1}$, where a_{j+1} is the least element of $\{1, \dots, k\}$ such that $\gamma a_{j+1} \in \mathbf{S}$ and $\gamma a_{j+1} \neq \beta_i$ for all $1 \leq i \leq j$. The process halts when we reach an β_m where either $\beta_m = \alpha$, or it is impossible to define β_{m+1} . (The latter occurs when, given $\beta_m = a\gamma$, we have $\gamma b \in \{\beta_i\}_{i=1}^m$ for all $b \in \{1, \dots, k\}$ such that $\gamma b \in \mathbf{S}$.) Let m be the largest positive integer for which β_m exists. We then define $\text{Greedy}_\alpha(\mathbf{S})$ to be:

$$\text{Greedy}_\alpha(\mathbf{S}) = a_1 a_2 \cdots a_m.$$

In the case where the process ends because $\beta_m = \alpha$, the string $\text{Greedy}_\alpha(\mathbf{S})$ may be viewed as a cycle; each of the strings from β_1 to $\beta_m = \alpha$ appears exactly once in that cycle.

Let $\mathcal{G}(n, k)$ be the collection of subsets $\mathbf{S} \subseteq \mathbf{T}(n, k)$ such that, for some $\alpha \in \mathbf{S}$, $\text{Greedy}_\alpha(\mathbf{S})$ is a universal cycle for \mathbf{S} . That is, the length- n suffix of $\text{Greedy}_\alpha(\mathbf{S})$ is α , and

for all $\beta \in \mathbf{S}$, β is a substring of $\text{Greedy}_\alpha(\mathbf{S})$ (treated as a cycle). The goal is to find a characterization of the sets in $\mathcal{G}(n, k)$ that are closed under rotations.

Given $\mathbf{S} \subseteq \mathbf{T}(n, k)$, and given $\alpha, \beta \in \mathbf{S}$, we will say that β is “increasable in \mathbf{S} to α ” if we can transform β into α by continually increasing individual symbols, and if the resulting string after each such increase is in \mathbf{S} . (By convention, we will say that $\alpha \in \mathbf{S}$ is increasable in \mathbf{S} to α .) For example, for our “no primes” set $\mathbf{S}_2 \subseteq \mathbf{T}(2, 9)$, 57 is increasable in \mathbf{S}_2 to 99:

$$57 \rightarrow 58 \rightarrow 68 \rightarrow 69 \rightarrow 99$$

Given this definition, our ultimate result is the following:

Theorem 1. *Let $\mathbf{S} \subseteq \mathbf{T}(n, k)$ be closed under rotations, and let $\alpha, \beta \in \mathbf{S}$. Then β is a substring of $\text{Greedy}_\alpha(\mathbf{S})$ (treated as a cycle) if and only if β is increasable in \mathbf{S} to a rotation of α .*

Thus, if $\mathbf{S} \subseteq \mathbf{T}(n, k)$ is closed under rotations, then $\mathbf{S} \in \mathcal{G}(n, k)$ if and only if there exists an $\alpha \in \mathbf{S}$ such that every $\beta \in \mathbf{S}$ is increasable in \mathbf{S} to α .

This theorem explains the absence of 77, 78, 87, and 88 in the string $\text{Greedy}_{99}(\mathbf{S}_2)$: none of those four strings are increasable in \mathbf{S}_2 to 99, since none of 79, 89, 97, or 98 is in \mathbf{S} . Note that this theorem is a generalization of Theorem 3, from [2].

The proof of this result will rely on an analysis of a combinatorial game, which we’ll call the Warden’s Game. The rules for this game will be given in Section 2, and the proof of Theorem 1 will follow in Section 3. In Section 4, we will look at several interesting families of sets $\mathbf{S} \subseteq \mathbf{T}(n, k)$ where a universal cycle can be generated with the greedy algorithm. Lastly, a possible avenue for future work will be detailed in Section 5.

3 The Warden’s Game

Consider the following fanciful scenario: there’s a certain prison warden who loves playing games. He sometimes makes an offer to let his prisoners out of prison, if they can beat him at a particular game. The game works as follows:

The warden shows the prisoner a row of n k -sided dice on a table. On each die, the faces are numbered from 1 to k . (It’s possible to have $k = 2$ here; the “dice” would then be coins with a 1 on one side and a 2 on the other.) A certain string $\alpha \in \mathbf{T}(n, k)$ is chosen: the prisoner will earn his freedom if, after any move, the dice on the table are showing the string α . (If the dice are showing α at the start of the game, the prisoner does not immediately win; the prisoner only wins when the dice show α after a move.) This game will be played at a rate of one move per day. So, the prisoner wants to reach α as quickly as possible; the warden wants to delay this as long as possible (indefinitely, if he can).

Each day, the rightmost die in the row will be moved to the far left, and possibly rotated to show a different number. The warden always has priority; he may transfer the rightmost die to the far left, and lower the number on that die. If he doesn’t want to do that (or he can’t, because he can’t lower the number any further), then the warden passes; then the prisoner must transfer the rightmost die to the far left, and optionally increase the number on that die.

As an example: let $n = 3$ and $k = 6$, so that the game is being played with three 6-sided dice. Let's say the current position is 513; the leftmost die shows 5, the middle die shows 1, and the rightmost die shows 3. The warden may transfer the rightmost die to the far left, lowering its value to 1 or 2 (thus producing the position 151 or 251). Or the warden may pass, in which case the prisoner must transfer the rightmost die and optionally increase its value (producing one of the positions 351, 451, 551, or 651). Let's say the warden chooses to move to the position 251. Then on the next move, the warden can't lower the value showing on the rightmost die. So the warden must pass, and the prisoner can move to 125, 225, 325, 425, 525, or 625. And so on.

Note: in the case where $k = 2$, the rules can be stated even more simply. If the rightmost coin is showing a 2, the warden transfers that coin, and optionally flips it to 1. If the rightmost coin is showing a 1, the prisoner transfers that coin, and optionally flips it to 2.

We can generalize this game still further, by limiting the legal positions in the game. We can choose any subset $\mathbf{S} \subseteq \mathbf{T}(n, k)$, closed under rotations, to be the set of legal positions. (We'll assume that the goal state α is in \mathbf{S} .) Then each move of the game, whether made by the prisoner or the warden, must be to a position in \mathbf{S} . We require that \mathbf{S} be closed under rotations so that there is a legal move from every legal position; if the warden ever passes, the prisoner always has the option to transfer the rightmost die without changing its value.

In [3], Weiss analyzed the Warden's Game (though not under that name) in the case where $k = 2$, $\mathbf{S} = \mathbf{T}(n, 2)$, and α is the string 2^n . Weiss proved that the game tree for the game is summarized by the lexicographically minimal de Bruijn sequence for $\mathbf{T}(n, 2)$; if both players play optimally, the game will proceed backwards through the de Bruijn sequence, one move at a time. For example, if $n = 4$, the lexicographically minimal de Bruijn sequence is the following:

(2222)1111211221212222

For this game, consider the position 2212. If we move one step backwards in the de Bruijn sequence from 2212, we get 1221; thus, the optimal move from 2212 must be for the warden to flip the rightmost coin before moving it, producing the position 1221. Similarly, the next optimal move is for the prisoner to move from 1221 to 1122, and so on, until the goal position 2222 is finally reached.

As we will prove in Section 3, the same holds true for any values of n and k , any subset $\mathbf{S} \subseteq \mathbf{T}(n, k)$ of legal positions (closed under rotations), and any goal state $\alpha \in \mathbf{S}$. The greedy algorithm always generates the full game tree for the Warden's Game. As an example, let's once again consider the subset $\mathbf{S}_2 \subseteq \mathbf{T}(2, 9)$ consisting of those strings where both the string itself and its reverse are 2-digit composite numbers. Here, once again, is the (not quite universal) cycle generated by the greedy algorithm, starting from $\alpha = 99$.

(99)33621224251526394454648182728496556685758699

For example, consider the position 82. The preceding substring of length 2 is 18; thus, the optimal move must be for the warden to move the rightmost die and reduce its value

from 2 to 1. The next optimal move must be to 81; both the prisoner and the warden refuse to change the value on the rightmost die. The next optimal move is to 48, which means the warden passes, and the prisoner increases the value on the rightmost die from 1 to 4. And so on.

Remember, four positions from \mathbf{S}_2 do not appear in this cycle: 77, 78, 87, and 88. Why don't they appear? Because they are losing positions for the prisoner! From any of those positions, the warden has a simple way to keep the game going indefinitely: he refuses to ever decrease the value on a die, and passes every time. The prisoner will never be able to increase a die to a 9, because 79, 89, 97, and 98 are all illegal positions. So the prisoner will never be able to reach the goal state, 99.

Our goal for the next section is to prove that this sort of thing happens regardless of the choices of \mathbf{S} and α . We will show that the prisoner can win from a given position $\beta \in \mathbf{S}$ if and only if he can win from β with the warden always passing: this happens when β is increasable in \mathbf{S} to a rotation of α . We will also show that the greedy algorithm generates the game tree for this game; thus, the greedy algorithm generates a universal cycle if and only if every $\beta \in \mathbf{S}$ is increasable in \mathbf{S} to a rotation of α .

4 Proof of Theorem 1

Assume we are given values of n and k , a set $\mathbf{S} \subseteq \mathbf{T}(n, k)$ of legal positions (closed under rotations), and a goal state $\alpha \in \mathbf{S}$. Define the "remoteness function" r on \mathbf{S} as follows: given $\beta \in \mathbf{S}$, the remoteness of β , $r(\beta)$, is the number of moves the game will last starting from β if both players play optimally. If the warden can keep the game going forever, then $r(\beta) = \infty$. This definition of remoteness is similar to the concept of remoteness used in [1].

Note: we can consider α to either be an end position (of remoteness 0) or a start position (of nonzero remoteness). We will always use the notation $r(\alpha)$ for the number of moves the game will last **starting** from α ; thus, $r(\alpha) > 0$.

Lemma 2. *Let $\beta_1, \beta_2 \in \mathbf{S}$ be such that $\beta_1 = \gamma a$ and $\beta_2 = a' \gamma$ for $a' \geq a$. (Thus, if the current position is β_1 and the warden passes, then the prisoner may move to β_2 .) Then, starting from β_1 , the prisoner has a strategy which can force the position to eventually reach β_2 .*

Proof. This can be proven by induction on the sum of the symbols in β_1 . Starting from $\beta_1 = b_1 b_2 \cdots b_{n-1} a$, if the warden passes, then the prisoner may move immediately to β_2 . Otherwise, the warden must move to $\beta_3 = a'' b_1 b_2 \cdots b_{n-1}$ for some $a'' < a$. But the sum of the symbols of β_3 is less than the sum of the symbols of β_1 . So by the inductive hypothesis, the prisoner has a strategy to eventually force the position to $b_{n-1} a'' b_1 \cdots b_{n-2}$, then to $b_{n-2} b_{n-1} a'' b_1 \cdots b_{n-3}$, and so on to $b_1 b_2 \cdots b_{n-1} a''$. We still have a smaller sum than the sum of the symbols in β_1 , so the prisoner can eventually force the position to $\beta_2 = a' b_1 b_2 \cdots b_{n-1}$, since $a' \geq a''$. \square

Lemma 3. *Given $\beta \in \mathbf{S}$, the prisoner can win from β if and only if β is increasable in \mathbf{S} to a rotation of α .*

Proof. If β is not increasable in \mathbf{S} to a rotation of α , then the warden can keep the game going indefinitely, simply by passing on every turn. Since the prisoner can only increase values, if the warden always passes, the prisoner will only be able to reach positions γ where β is increasable in \mathbf{S} to a rotation of γ . Since α is not such a position, the prisoner can never win.

Now assume that β is increasable in \mathbf{S} to a rotation of α . Then, if the warden chooses to pass on every move, then there is a sequence of moves $\beta = \beta_0, \beta_1, \beta_2, \dots, \beta_m = \alpha$ that the prisoner may make to win. By Lemma 2, if the game starts from $\beta = \beta_0$, then prisoner can eventually force the position to be β_1 , then β_2 , and so on until finally reaching α and winning. \square

Note: while this shows that the prisoner can win eventually from any position β that is increasable in S to α , the recursive strategy described above will probably **not** be the prisoner's optimal strategy.

Lemma 4. *Given any positive integer m , if there are no positions of remoteness m , then there are no positions of remoteness $m + 1$. (Thus, by induction, there are no positions of remoteness m' for any integer $m' \geq m$.)*

Proof. If there were a position of remoteness $m + 1$, then with optimal play, the first move from such a position would be to a position of remoteness m . . . and no such position exists. So there are no positions of remoteness $m + 1$. \square

Lemma 5. *Given positions $\gamma a_1, \gamma a_2 \in \mathbf{S}$, if $a_1 < a_2$, then $r(\gamma a_1) \leq r(\gamma a_2)$.*

The point here is that, the greater the rightmost symbol in the string, the better off the warden is. From γa_1 , the warden may move to any $a\gamma \in \mathbf{S}$ such that $a < a_1$, or the warden may give the prisoner the choice to move to any $a\gamma \in \mathbf{S}$ where $a \geq a_1$. From γa_2 , the warden still may move to any $a\gamma \in \mathbf{S}$ such that $a < a_1$, or the warden can ensure that the next move is to $a\gamma \in \mathbf{S}$ for some $a \geq a_1$. . . but in the latter case, the warden may choose a specific $a\gamma \in \mathbf{S}$ such that $a_1 \leq a < a_2$, if he so desires. This extra option can only help the warden, never hurt him. So we must have $r(\gamma a_1) \leq r(\gamma a_2)$.

Note: we will later see that if $r(\gamma a_1)$ and $r(\gamma a_2)$ are both finite, then $r(\gamma a_1) < r(\gamma a_2)$.

Lemma 6. *For any nonnegative integer m , there is at most one position of remoteness m .*

This is a significant result; combined with Lemma 4, the conclusion is that the “game tree” is really a chain, not a tree. There is one position of remoteness 0 (namely, α), one position of remoteness 1, one position of remoteness 2, and so on until all the winning positions for the prisoner have been exhausted. And given any position β that is winning for the prisoner, if a game starting from β is played optimally, the game will pass through all positions of remoteness less than $r(\beta)$ until finally reaching α .

Proof. We will prove this by contradiction. Let m be the smallest integer where there are multiple positions of remoteness m . There is only one position of remoteness 0 (namely, α), so $m \geq 1$. Let $a\gamma$ be the one position of remoteness $m - 1$; this position must be

reachable in one move from all positions of remoteness m , so all such positions must have the form γb , for some $b \in \{1, 2, \dots, k\}$.

Let $b_1 < b_2 < \dots < b_l$ be the elements of $\{1, 2, \dots, k\}$ such that $\gamma b_i \in \mathbf{S}$ for each i . By Lemma 5, $r(\gamma b_1) \leq r(\gamma b_2) \leq \dots \leq r(\gamma b_l)$. If j is the smallest natural number such that $r(\gamma b_j) = m$, then because there is just one position of each remoteness less than m , we must have

$$r(\gamma b_1) < r(\gamma b_2) < \dots < r(\gamma b_{j-1}) < r(\gamma b_j) = r(\gamma b_{j+1}).$$

For each natural number $i \leq j$, let $a_i \in \{1, 2, \dots, k\}$ be such that $a_i \gamma$ is the next position reached from γb_i if both sides play optimally. (The a_i 's for $1 \leq i \leq j$ are all distinct.) We can now show that it is impossible to have $r(\gamma b_j) = r(\gamma b_{j+1}) = m$:

Consider the two sets $\{a_i\}_{i=1}^j$ and $\{b_i\}_{i=1}^j$. If these two sets are identical, then consider what happens if the warden passes from the position γb_{j+1} . The prisoner is then forced to move to $c\gamma$ for some $c \geq b_{j+1}$. This c will not be an element of $\{a_i\}_{i=1}^j$, and hence $c\gamma$ will not have remoteness at most $m - 1$. So by passing, the warden can force the next move to be to a position of remoteness at least m ; γb_{j+1} can't have remoteness m .

On the other hand, assume $\{a_i\}_{i=1}^j$ and $\{b_i\}_{i=1}^j$ are not identical. That means there is some $b_i < b_{j+1}$ that is not in $\{a_i\}_{i=1}^j$. If the warden moves from γb_{j+1} to $b_i \gamma$, then the warden has not moved to a position of remoteness at most $m - 1$. So again, the warden was able to force the next move to be to a position of remoteness at least m ; γb_{j+1} can't have remoteness m .

This completes the contradiction; it is impossible to have two positions of the same finite remoteness. \square

Lemma 7. *If $\alpha \in \mathbf{S}$ is the goal state, then the position in \mathbf{S} of highest finite remoteness is α .*

The reason: since α is (trivially) increasable in \mathbf{S} to a rotation of α , $r(\alpha)$ is finite. Say $r(\alpha) = m > 0$ (we are treating α as a start position, not an end position). If there were any position β such that $r(\beta) = m + 1$, then with optimal play, the next move from β would be to a position of remoteness m : namely, α . But that means, with optimal play, β is just one move from the goal state; so $r(\beta) = 1$, not $m + 1$. Thus, α has the maximal finite remoteness of any string in \mathbf{S} .

This also means that the positions in \mathbf{S} that are winning for the prisoner form a cycle. The only question remaining is why this is the same cycle that we would get from the greedy algorithm.

Theorem 8. *The greedy algorithm generates the game "tree" for the warden's game.*

Proof. Let $\{\beta_m\} \subseteq \mathbf{S}$ be the sequence of strings generated by the greedy algorithm, starting from α . We have $\beta_0 = \alpha$, and for each $m \geq 0$, if $\beta_m = a\gamma$, then $\beta_{m+1} = \gamma b$, where b is the least element of $\{1, \dots, k\}$ such that $\gamma b \in \mathbf{S}$ and γb does not appear in the set $\{\beta_i\}_{i=1}^m$. (If $\gamma b \in \{\beta_i\}_{i=1}^m$ for all $b \in \{1, \dots, k\}$, then there is no β_{m+1} ; β_m is the last string in the sequence.) Obviously, $\beta_0 = \alpha$ is the one position of remoteness 0. We must show that $r(\beta_m) = m$ for all $m > 0$; we will prove this by induction.

Assume we have $r(\beta_i) = i$ whenever $0 < i \leq m$. Let $\beta_m = a\gamma$. Assume there is a position of remoteness $m + 1$; it must be of the form $\gamma b \in \mathbf{S}$ (so that there is a move available to $a\gamma$), where γb is not in $\{\beta_i\}_{i=1}^m$ (since all strings in that set have remoteness m or less). Let $b_1 < b_2 < \dots < b_j$ be the elements of $\{1, \dots, k\}$ such that $\gamma b_i \in \mathbf{S}$, but γb_i is not in $\{\beta_i\}_{i=1}^m$. (Thus, $\beta_{m+1} = \gamma b_1$.) By Lemma 5, $r(\gamma b_1) \leq r(\gamma b_2) \leq \dots \leq r(\gamma b_j)$; by Lemma 6, all of those inequalities are strict except for where we have multiple positions of infinite remoteness. We have $r(\gamma b_1) > m$, so the only i where we can have $r(\gamma b_i) = m + 1$ is $i = 1$. Thus, we must have $r(\beta_{m+1}) = r(\gamma b_1) = m + 1$.

Now assume that m is the largest finite remoteness of any position in \mathbf{S} ; that is, $r(\alpha) = m$. The above inductive argument shows that $\beta_m = \alpha$. And the way we defined $\text{Greedy}_\alpha(\mathbf{S})$, the process halts if we ever have $\beta_m = \alpha$. So we do have $r(\beta_m) = m$ for all $m > 0$; the length- n substrings of $\text{Greedy}_\alpha(\mathbf{S})$ are exactly the winning positions for the prisoner, in order of remoteness. \square

We thus have proven Theorem 1; the length- n substrings of $\text{Greedy}_\alpha(\mathbf{S})$ are exactly the winning positions for the prisoner, which are exactly the strings in \mathbf{S} which are increasable in \mathbf{S} to a rotation of α .

As a final note for this section, here's a comment on the optimal strategy for the warden:

Corollary 9. *Given any position $\gamma b \in \mathbf{S}$, if a is the greatest number less than b such that $a\gamma \in \mathbf{S}$, then the optimal move for the warden from γb is either to move to $a\gamma$, or to pass. (So, when the warden does decrease a number, he should always do so by the smallest amount possible.)*

Proof. Assume not. Assume there is a position $\gamma b \in \mathbf{S}$, where a is the greatest number less than b where $a\gamma \in \mathbf{S}$, but the warden's optimal move is to $c\gamma$, where $c < a$. Let $r(\gamma b) = m$; then the remoteness of $c\gamma$ is $m - 1$ (where, if $c\gamma = \alpha$, we are treating α as an end position).

Since \mathbf{S} is closed under rotations and $a\gamma \in \mathbf{S}$, we have $\gamma a \in \mathbf{S}$. From Lemmas 5 and 6, since $a < b$, we have $r(\gamma a) < r(\gamma b)$. So $r(\gamma a) < m$. But from the position γa , the warden can move to $c\gamma$, a position of remoteness $m - 1$. So $r(\gamma a) \geq m$, contradiction. \square

There seems to be no similar statements we can make about the optimal strategy for the prisoner; depending on the situation, the prisoner may want to increase the value on a die by the least amount possible, the greatest amount possible, or some amount in between. For example, all such possibilities occur in our "no primes" example, $\mathbf{S}_2 \subseteq \mathbf{T}(2, 9)$. There seems to be no way for the prisoner to determine anything about his optimal next move from β , other than to generate the entire string $\text{Greedy}_\alpha(\mathbf{S})$ until β is reached.

5 Interesting examples

In [2], there are a number of examples of interesting sets $\mathbf{S} \subseteq \mathbf{T}(n, k)$ where the greedy algorithm produces a universal cycle for \mathbf{S} . Here are some new such sets derived from

Theorem 1.

5.1 Strings increasable to a rotation of α

Choose any $\alpha \in \mathbf{T}(n, k)$, and let $\mathbf{S} \subseteq \mathbf{T}(n, k)$ be the strings that are increasable in $\mathbf{T}(n, k)$ to a rotation of α . Obviously, \mathbf{S} is then closed under rotations.

Given any $\beta \in \mathbf{S}$, β is increasable in $\mathbf{T}(n, k)$ to a rotation of α . So there are strings $\beta_0, \beta_1, \dots, \beta_m \in \mathbf{T}(n, k)$ such that $\beta_0 = \beta$, β_m is a rotation of α , and each β_i can be changed to β_{i+1} by increasing one symbol. Then each β_i is increasable in $\mathbf{T}(n, k)$ to a rotation of α , so each $\beta_i \in \mathbf{S}$. But that means β is actually increasable in \mathbf{S} to a rotation of α . Since this is true for all $\beta \in \mathbf{S}$, the greedy algorithm starting from α generates a universal cycle of \mathbf{S} .

For example, let $n = 3$, $k = 4$, and $\alpha = 143$. Here's a universal cycle for the strings in $\mathbf{T}(3, 4)$ increasable to a rotation of 143:

$$(143)1112113122123132133141142143$$

Note: we get the same collection of strings if α is either 314 or 431. But the resulting universal cycle would be different in either such case. Here's the universal cycle for $\alpha = 314$:

$$(314)1112113114212213214312313314$$

And here's the universal cycle for $\alpha = 431$:

$$(431)1121131221231321331411421431$$

5.2 Unions

Let $\mathbf{S}_1, \mathbf{S}_2 \in \mathcal{G}(n, k)$, where \mathbf{S}_1 and \mathbf{S}_2 are both closed under rotations. Assume all strings in \mathbf{S}_1 and \mathbf{S}_2 are increasable (in their respective sets) to rotations of a single string α . Then all strings in $\mathbf{S}_1 \cup \mathbf{S}_2$ are increasable in $\mathbf{S}_1 \cup \mathbf{S}_2$ to a rotation of α , so the greedy algorithm starting from α generates a universal cycle of $\mathbf{S}_1 \cup \mathbf{S}_2$.

This raises the question of whether the same can be said of intersections. However, this turns out to be false: even if the greedy algorithm generates universal cycles for \mathbf{S}_1 and \mathbf{S}_2 , the same may not be true of $\mathbf{S}_1 \cap \mathbf{S}_2$. One simple example will demonstrate why. Let $\mathbf{S}_1, \mathbf{S}_2 \subseteq \mathbf{T}(2, 3)$ be as follows:

$$\mathbf{S}_1 = \{11, 13, 31, 33\}$$

$$\mathbf{S}_2 = \{11, 12, 21, 23, 32, 33\}$$

The greedy algorithm (starting from 33) generates universal cycles for both \mathbf{S}_1 and \mathbf{S}_2 , but does not do so for $\mathbf{S}_1 \cap \mathbf{S}_2 = \{11, 33\}$. The problem is that there may be an element $\beta \in \mathbf{S}_1 \cap \mathbf{S}_2$ which is increasable to a rotation of α in both \mathbf{S}_1 and in \mathbf{S}_2 , but the paths from β to α may be different in each set. (Here, $\beta = 11$; we have $11 \rightarrow 13 \rightarrow 33$ in \mathbf{S}_1 , and $11 \rightarrow 12 \rightarrow 32 \rightarrow 33$ in \mathbf{S}_2 .)

5.3 Rotations of increasing strings

Assume that $n \leq k$. Let \mathbf{S} be the set containing all strictly increasing strings in $\mathbf{T}(n, k)$ and their rotations. For example, in $\mathbf{T}(3, 5)$, \mathbf{S} contains the following strings and their rotations:

$$\{123, 124, 125, 134, 135, 145, 234, 235, 245, 345\}$$

By definition, \mathbf{S} is closed under rotations. Let α be the lexicographically maximal, strictly increasing string in $\mathbf{T}(n, k)$:

$$\alpha = (k - n + 1)(k - n + 2) \cdots (k - 1)k.$$

Any string $\beta \in \mathbf{S}$ is increasable in \mathbf{S} to a rotation of α ; the greatest symbol in β can be increased to k , then the next-greatest symbol can be increased to $k - 1$, and so on. So a universal cycle for \mathbf{S} can be generated with the greedy algorithm.

5.4 Maximum cyclic increment or cyclic decrement

Choose integers $I > 0$ and $D > 0$. Let $\mathbf{S} \subseteq \mathbf{T}(n, k)$ be the set of strings with no cyclic increment of size greater than I and no cyclic decrement of size greater than D . For example, if we take $\mathbf{T}(3, 4)$, $I = 2$, and $D = 1$, then \mathbf{S} consists of the following strings and their rotations:

$$\{111, 112, 122, 132, 222, 223, 233, 243, 333, 334, 344, 444\}$$

By definition, \mathbf{S} is closed under rotations. Let $\alpha = k^n$; α contains no cyclic increments or decrements, so $\alpha \in \mathbf{S}$.

To show that the greedy algorithm works here: choose any $\beta \in \mathbf{S}$ such that $\beta \neq k^n$. Assume $\beta = \gamma_1 a \gamma_2$, where a is the least symbol in β . Let $\beta' = \gamma_1 (a + 1) \gamma_2$. This change from β to β' will either decrease the size of cyclic increments/decrements, or will produce a new cyclic increment or decrement of size 1 (which is legal). So $\beta' \in \mathbf{S}$. Thus, for any $\beta \in \mathbf{S}$, it's possible to increase a symbol of β by 1 to produce another string in \mathbf{S} . This process can be continued until α is reached. So any $\beta \in \mathbf{S}$ is increasable in \mathbf{S} to α , and the greedy algorithm (starting from α) produces a universal cycle.

5.5 Minimum span, maximum span

Choose integers m and M such that $0 \leq m < M < k$. Let $\mathbf{S} \subseteq \mathbf{T}(n, k)$ be the set of strings β whose span is at least m and at most M . (The “span” of a string β is the difference between the least and greatest symbols in β .) For example, if we take $\mathbf{T}(3, 4)$, $m = 1$, and $M = 2$, then \mathbf{S} consists of the following strings and their permutations:

$$\{112, 113, 122, 123, 133, 223, 224, 233, 234, 244, 334, 344\}$$

Clearly, \mathbf{S} is closed under rotations. Any $\beta \in \mathbf{S}$ can be increased in \mathbf{S} to a rotation of $\alpha = (k - m)k^{n-1}$, as follows: if the span of β is greater than m , increase the least

symbol of β by 1. If the span of β equals m , increase the greatest symbol of β by 1, unless the greatest symbol is k . Repeat this process until a string β containing the symbol k is reached. At that point, the least symbol in β will be $k - m$; leave that one symbol alone, and increase all the other symbols of β to k .

Thus, the greedy algorithm generates a universal cycle for \mathbf{S} .

5.6 Avoiding a substring

Choose a string $\gamma \in \mathbf{T}(m, k)$ for some $m \geq 1$, and let $\mathbf{S} \subseteq \mathbf{T}(n, k)$ be the set of strings that do not contain γ as a cyclic substring. It was proven in [2] that if γ does not contain k , then \mathbf{S} can be generated by the greedy algorithm. If γ does contain k , then we can still make a weaker statement: let $i, j \in \{1, \dots, k\}$ be two symbols such that $i < j$. If γ contains i but not j , then \mathbf{S} can be generated by the greedy algorithm.

As an example, let $\mathbf{S} \subseteq \mathbf{T}(3, 3)$ be the set of strings not containing 13 as a cyclic substring. (In this case, $i = 1$ and $j = 2$.) Then \mathbf{S} consists of the rotations of the following strings:

$$\{111, 112, 122, 123, 222, 223, 233, 333\}$$

The reason why the greedy algorithm works: $k^n \in \mathbf{S}$, since the forbidden substring γ includes a symbol $a < k$. Given any $\beta \in \mathbf{S}$, we can increase β in \mathbf{S} to k^n , as follows: replace any occurrences of a in β with b , then increase all symbols in β to k . So all strings in \mathbf{S} are increasable in \mathbf{S} to k^n .

It would seem to be a difficult question to completely categorize the forbidden substrings γ for which \mathbf{S} can be generated by the greedy algorithm. I have not found any examples of a forbidden substring γ containing a symbol $a \leq k - 2$ where the greedy algorithm fails. And I would conjecture that there are none:

Conjecture 10. Let $\gamma \in \mathbf{T}(m, k)$ be a string containing a symbol $a \leq k - 2$. Let $\mathbf{S} \subseteq \mathbf{T}(n, k)$ (for some $n \geq m$) be the set of strings not containing γ as a cyclic substring. Then the greedy algorithm starting from k^n generates a universal cycle for \mathbf{S} .

Now, if all symbols in γ are either k or $k - 1$, then the greedy algorithm may fail. Let $\mathbf{S} \subseteq \mathbf{T}(n, 9)$ (for some $n \geq 4$) be the set of strings not containing a particular γ as a cyclic substring. I leave it as an exercise to the interested reader to show that the greedy algorithm (starting from $\alpha = 9^n$) succeeds if γ is in the following set...

$$\{8899, 8989\}$$

... but the greedy algorithm fails if γ is in the following set.

$$\{89, 889, 899, 8889, 8999\}$$

Note: for some strings γ , the outcome depends on the value of n . One example is $\gamma = 8998$; the greedy algorithm will fail if and only if n is a multiple of 3. (All strings in \mathbf{S} will be increasable in \mathbf{S} to 9^n , except for $\beta = (889)^{n/3}$ and its rotations.)

6 Conclusion and future work: necklaces

At this point, we have classified the subsets $\mathbf{S} \subseteq \mathbf{T}(n, k)$, closed under rotations, where the greedy algorithm produces a universal cycle for \mathbf{S} . But there's one major problem with generating universal cycles with the greedy algorithm: we must store the entire cycle (which could be exponential in length) in order to generate the cycle. Fortunately, when $\mathbf{S} = \mathbf{T}(n, k)$, there is a faster method:

Given $\alpha \in \mathbf{T}(n, k)$, α is a “necklace” if, out of all rotations of α , α itself is the lexicographically earliest such rotation. If we take all such necklaces in lexicographic order, and append their aperiodic prefixes, we obtain a de Bruijn cycle for $\mathbf{T}(n, k)$ (the same cycle produced by the greedy algorithm). For example, here's the resulting universal cycle for $\mathbf{T}(3, 3)$ (with spaces added between the prefixes):

(333)1 112 113 122 123 132 133 2 223 233 3

In [2], this is called the FKM algorithm. It is proved in [2] that this algorithm produces a universal cycle for $\mathbf{S} \subseteq \mathbf{T}(n, k)$ (the same universal cycle produced by the greedy algorithm) if

1. \mathbf{S} is closed under rotations, and
2. every necklace in \mathbf{S} remains a necklace in \mathbf{S} if the suffix of length i (whenever $1 \leq i \leq n$) is replaced with k^i . (When this second condition holds true, \mathbf{S} is referred to as a “ k -suffix language”.)

However, this is not an “if and only if” situation. The following example of a set $\mathbf{S} \subseteq \mathbf{T}(4, 3)$ is given in [2]:

$\mathbf{S} = \{1112, 1121, 1122, 1211, 1212, 1221, 1222, 1322, 2111,$
 $2112, 2121, 2122, 2132, 2211, 2212, 2213, 2221, 3221\}$

Not all necklaces in \mathbf{S} remain in \mathbf{S} when a suffix is replaced with all 3's. However, the FKM algorithm works here. The necklaces in \mathbf{S} , in lexicographic order, are 1112, 1122, 1212, 1222, and 1322. Reduce 1212 to its aperiodic prefix 12, then concatenate all the strings, and you obtain a universal cycle:

(1322)1112 1122 12 1222 1322

So it is natural to ask whether there is a necessary and sufficient condition on $\mathbf{S} \subseteq \mathbf{T}(n, k)$ so that the FKM algorithm generates a universal cycle for \mathbf{S} . I have a possible candidate for just such a condition:

Let's generalize the concept of a k -suffix language as follows. Given a string $\alpha \in \mathbf{T}(n, k)$, call a set $\mathbf{S} \subseteq \mathbf{T}(n, k)$ an “ α -suffix language” if, for any $\beta \in \mathbf{S}$, each symbol in β is at most the corresponding symbol in α , and we obtain another element of \mathbf{S} if we replace any suffix of β with an equal-length suffix of α . That is, if $\beta = b_1 \cdots b_n$ and $\alpha = a_1 \cdots a_n$, then for all m such that $1 \leq m \leq n$, we have $b_m \leq a_m$ and $b_1 \cdots b_{m-1} a_m \cdots a_n \in \mathbf{S}$. With this definition in place, I would conjecture the following:

Conjecture 11. Let $\mathbf{S} \subseteq \mathbf{T}(n, k)$ be a set that is closed under rotations. Then the FKM algorithm generates a universal cycle for \mathbf{S} if and only if the set of necklaces in \mathbf{S} is an α -suffix language, where α is the lexicographically maximal necklace in \mathbf{S} .

The reason for this conjecture: assume that $\mathbf{S} \subseteq \mathbf{T}(n, k)$ is closed under rotations, and assume the set of necklaces in \mathbf{S} is an α -suffix language, where α is the lexicographically maximal necklace in \mathbf{S} . Under these circumstances, it appears that the prisoner has a strategy such that, if the current position is a necklace $\beta \in \mathbf{S}$, the prisoner can ensure that the next necklace position reached is lexicographically earlier than β . Thus, in the universal cycle generated by the greedy algorithm, the necklaces appear in lexicographic order. Perhaps then, the FKM algorithm generates the same cycle as the greedy algorithm.

References

- [1] E. Berlekamp, J. Conway, and R. Guy, *Winning Ways for Your Mathematical Plays, Vol. 2*. A K Peters, Ltd., Natick, MA, 2003.
- [2] J. Sawada, A. Williams, and D. Wong, Generalizing the Classic Greedy and Necklace Constructions of de Bruijn Sequences and Universal Cycles, *The Electronic Journal of Combinatorics* **23(1)** #P1.24(2016).
- [3] G. Weiss, A Combinatorial Game Approach to State Nullification by Hybrid Feedback, in *46th IEEE Conference on Decision and Control*, (2007).