

Oskar's Coin Game

Tom Verhoeff

August 31, 1990

1 Description and First Problem

On a table lie N coins ($N \geq 1$). The coins are numbered 1 through N . Suppose the total number of heads facing upwards is k . Now the coin with number k is turned upside down. What happens if the procedure is repeated, that is, counting heads and turning the corresponding coin?

This is the first question Oskar van Deventer asked me about his coin game back in 1986. In this article, I will present a solution and also deal with some other questions. There is not enough space to include complete proofs, but I suspect that the interested reader is capable of filling in the details.

Before presenting a solution to the first problem, let us take a closer look at the problem statement. The above description is not complete. It is unclear what to do when all coins show their tail side ($k = 0$), because there is no coin numbered 0. Let us agree that for $k = 0$ the configuration is left unchanged. Also, it may be unclear what should be considered a solution to the first problem. Each configuration has exactly one successor configuration and, hence, the game simply continues forever. Obviously, that is not a very satisfactory answer. Since the number of configurations is finite (in fact, 2^N), the game must eventually enter some configuration a second time and thereafter repeat the same cycle of configurations forever. The first problem now is to find out which cycles are possible.

If you want to try and find a solution yourself, then this is the right moment to stop reading and to take some numbered coins from your pocket.

2 Formalization

Before spoiling (part of) the fun, we introduce some notation to formalize Oskar's coin game.

A *configuration* in the game may be modeled by a mapping from the set of numbers 1 through N to $\{0, 1\}$, where the image 0 stands for tail and 1 for head. We denote function application by an infix dot. Thus, for configuration c , we have $c.i = 0$ precisely when coin i shows tail and $c.i = 1$ when it shows head.

The *weight* of a configuration is the number of heads in it. More formally, we define mapping w from configurations to natural numbers by

$$w.c = (\mathbf{N}i : 1 \leq i \leq N : c.i = 1) \quad \text{for all } c,$$

where $(\mathbf{N}i : D : P)$ is the number of i 's in domain D with property P . In the sequel, variable i will always range over (a subset of) $\{1 \dots N\}$ and we omit the condition $1 \leq i \leq N$ from the domain expressions. Notice that we have $0 \leq w.c \leq N$.

The *successor* of configuration c is obtained by turning coin $w.c$ upside down if $w.c \neq 0$, and doing nothing otherwise. Formally, mapping s from configurations to configurations is defined by

$$s.c.i = \begin{cases} c.i & \text{if } i \neq w.c \\ 1 - c.i & \text{if } i = w.c \end{cases} \quad \text{for all } c \text{ and } i,$$

where function application is left associative, that is, $s.c.i$ stands for $(s.c).i$. Application of s constitutes a *move* in the game. Observe that $s.c = c$ if and only if $w.c = 0$.

Let us now consider a small example with $N = 7$. In Table 1, are shown a configuration and some successors obtained by repeated application of s (s^m stands for m applications of s). The weight of each configuration is shown in the rightmost column. From this example it is not immediately clear what eventually happens. The changes in configuration seem quite erratic. Playing around with some more examples, however, will quickly convince you that the game always converges to the no-heads (zero-weight) configuration.

x	$x.1$	$x.2$	$x.3$	$x.4$	$x.5$	$x.6$	$x.7$	$w.x$
c	1	0	0	1	1	0	1	4
$s.c$	1	0	0	0	1	0	1	3
$s^2.c$	1	0	1	0	1	0	1	4
$s^3.c$	1	0	1	1	1	0	1	5
$s^4.c$	1	0	1	1	0	0	1	4
$s^5.c$	1	0	1	0	0	0	1	3

Table 1: A sequence of configurations

3 Solution to the First Problem

We will prove that the no-heads configuration is the only cycle in the game. In fact, we can even give an explicit expression for the number of s -moves required for configuration c to reach the zero-weight configuration. We will do so by defining a mapping n from configurations to integers and proving the following three

properties.

$$\begin{aligned}
 P_1: & \quad n.c \geq 0 \\
 P_2: & \quad n.c = 0 \quad \text{if } w.c = 0 \\
 P_3: & \quad n.(s.c) = n.c - 1 \quad \text{if } w.c \neq 0
 \end{aligned}$$

First, we analyze the game a little further and we construct an expression for $n.c$ that is easy to justify informally, but not so easy to manipulate formally. Nevertheless, the two properties P_1 and P_2 are easy to prove. Then we rewrite it twice to improve manipulation and we show property P_3 .

If Table 1 is extended and marked appropriately, viz. by highlighting $c.(w.c)$, then a pattern appears. Table 2 shows this for the same initial configuration as above (it is turned ninety degrees counterclockwise to save space). In this table some features are very prominent. For instance, application of s changes the weight by one (if it was non-zero to begin with):

$$w.(s.c) = \begin{cases} w.c & \text{if } w.c = 0 \\ w.c + 1 & \text{if } w.c \neq 0 \text{ and } c.(w.c) = 0 \\ w.c - 1 & \text{if } w.c \neq 0 \text{ and } c.(w.c) = 1 \end{cases}$$

Furthermore, the box resembles an elevator: it travels up and down between the zeroes below it and the ones above it (these are underlined in Table 2), “picking” them up when they are reached (imagine a zero at “level” 0).

m																	1	1	1	1	1	1	1	1	1	1
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8							
$s^m.c.7$	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	0	0	0	0	0	0							
$s^m.c.6$	0	0	0	0	0	0	0	0	0	0	<u>0</u>	1	<u>1</u>	0	0	0	0	0	0							
$s^m.c.5$	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	0	0	0	0	0	<u>0</u>	1	1	1	<u>1</u>	0	0	0	0	0							
$s^m.c.4$	<u>1</u>	0	<u>0</u>	1	<u>1</u>	0	0	0	<u>0</u>	1	1	1	1	1	<u>1</u>	0	0	0	0							
$s^m.c.3$	<u>0</u>	<u>0</u>	1	1	1	<u>1</u>	0	<u>0</u>	1	1	1	1	1	1	1	<u>1</u>	0	0	0							
$s^m.c.2$	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	1	1	1	1	1	1	1	1	<u>1</u>	0	0							
$s^m.c.1$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	<u>1</u>	0							
$w.(s^m.c)$	4	3	4	5	4	3	2	3	4	5	6	7	6	5	4	3	2	1	<u>0</u>							

Table 2: A sequence of marked configurations

We can break up the trajectory of the elevator by considering the moments when the elevator returns to the level at which it started. In the above example, the elevator starts at level 4 and returns there four times: twice after “picking up” a zero and twice after a one. It takes 2 moves to pick up the first zero (at level 3) and return to level 4, then 2 more moves for picking up the first one (at level 5) and returning, 4 more for the next zero (at level 2), and another 6 for the last one (at level 7). Notice that the number of moves to pick up something at level i and return to level k is $2 * |k - i|$. After the last return to level 4, it takes 4 more moves to go

down all the way. The number of moves in the final rush down is simply the weight of the initial configuration, because that is the level where the elevator starts.

Thus we arrive at the following expression for $n.c$:

$$\begin{aligned} n.c &= 2 * u.c + w.c \quad \text{with} \\ u.c &= (\mathbf{S}i : (i < w.c \wedge c.i = 0) \vee (i > w.c \wedge c.i = 1) : |w.c - i|), \end{aligned}$$

where $(\mathbf{S}i : D : E)$ equals the sum of expressions E taken over all i in the domain D . For the above example, we have $u.c = 1 + 2 + 1 + 3 = 7$ and, hence, $n.c = 2 * 7 + 4 = 18$.

Of course, the way we constructed the expression for $n.c$ does not prove its correctness as expressed by P_1 through P_3 (for instance, why does it not matter whether the elevator starts on a zero or a one?). But it looks promising. Note that from the definitions of u and n it is immediately clear that $u.c \geq 0$ and $n.c \geq 0$, proving P_1 . On account of $w.c \geq 0$, we now also have $n.c = 0$ if and only if $w.c = 0$, proving P_2 . All that remains to be proven is P_3 : $n.(s.c) = n.c - 1$ if $w.c \neq 0$. In order to do so we first simplify the expression for $n.c$.

The weight of a configuration enjoys the following fundamental property:

$$(\mathbf{N}i : i \leq w.c : c.i = 0) = (\mathbf{N}i : w.c < i : c.i = 1).$$

Phrased in elevator terminology, the number of zeroes at or below the level of the elevator equals the number of ones above it. Using this property, one can simplify the expression for $u.c$ to

$$u.c = (\mathbf{S}i : w.c < i \wedge c.i = 1 : i) - (\mathbf{S}i : i \leq w.c \wedge c.i = 0 : i).$$

For the above example, we would now compute $u.c = 5 + 7 - 2 - 3 = 7$. Note that from this new expression for $u.c$ it is no longer immediately obvious that $u.c \geq 0$.

On the basis of the equality $(\mathbf{S}i : 1 \leq i \leq k : i) = k * (k + 1) / 2$, a further simplification yields

$$\begin{aligned} u.c &= v.c - w.c * (w.c + 1) / 2 \quad \text{with} \\ v.c &= (\mathbf{S}i : c.i = 1 : i). \end{aligned}$$

Substituting this expression for $u.c$ in the definition of $n.c$ results in

$$n.c = 2 * v.c - (w.c)^2.$$

For the above example, we get $v.c = 2 + 3 + 5 + 7 = 17$ and $n.c = 2 * 17 - 4^2 = 18$. From this expression for $n.c$ it certainly is not trivial to deduce P_1 and P_2 . However, it is straightforward to deduce

$$v.(s.c) = \begin{cases} v.c & \text{if } w.c = 0 \\ v.c + w.c & \text{if } w.c \neq 0 \text{ and } c.(w.c) = 0 \\ v.c - w.c & \text{if } w.c \neq 0 \text{ and } c.(w.c) = 1 \end{cases}$$

Assuming $w.c \neq 0$ we can now derive

$$\begin{aligned}
n.(s.c) &= 2 * v.(s.c) - w.(s.c) \\
&= 2 * (v.c \pm w.c) - (w.c \pm 1)^2 \\
&= 2 * v.c \pm 2 * w.c - (w.c)^2 \mp 2 * w.c - 1 \\
&= n.c - 1,
\end{aligned}$$

which proves P_3 . This completes the proof that $n.c$ is the number of s -moves needed for c to reach the zero-weight configuration. Notice that $n.c$ can also be expressed concisely as

$$n.c = 2 * (\mathbf{S}i :: i * c.i) - (\mathbf{S}i :: c.i)^2.$$

4 Second Problem

We now know that every configuration eventually decays to the zero-weight configuration. Naturally, the second question Oskar asked me about his coin game was:

Which configurations take longest to decay, that is, are furthest from the zero-weight cycle?

More formally, this can be phrased as: For which c is $n.c$ maximal?

First we show that any configuration c with $c.i = 1$ and $c.j = 0$ for some $i < j$ (that is, with a one “left of” a zero) does not have maximal $n.c$. The argument is as follows. Consider configuration b obtained from c by swapping the images for i and j . Then we have $w.b = w.c$ and $v.b = v.c - i + j$. Thus we can derive

$$\begin{aligned}
n.b &= 2 * v.b - (w.b)^2 \\
&= 2 * v.c - 2 * i + 2 * j - (w.c)^2 \\
&= n.c + 2 * (j - i) \\
&> n.c,
\end{aligned}$$

which shows that $n.c$ is not maximal. Therefore, the only candidates for maximal $n.c$ are configurations of the form c_k defined by

$$c_k.i = \begin{cases} 0 & \text{if } i \leq N - k \\ 1 & \text{if } N - k < i \end{cases} \quad \text{for all } i,$$

that is, with $N - k$ zeroes “left of” (or “under”) k ones. We have

$$\begin{aligned}
w.c_k &= k, \\
v.c_k &= k * (N - k + 1 + N)/2, \\
n.c_k &= k * (2 * N + 1 - 2 * k).
\end{aligned}$$

Hence, $n.c_k$ is maximal for $k = (N + 1) \mathbf{div} 2$ and the value of this maximum is $N * (N + 1)/2$. Apparently the configuration attaining maximal $n.c$ is unique.

5 Third Problem

Oskar had even a third question:

What can you say about the distribution of $n.c$ for fixed N ?

In Table 3, some statistics are collected. The entry at row N , column m shows the number of configurations in the N -coin game that reach the zero-weight configuration in exactly m moves, that is, the number of c with $n.c = m$. The remarkable thing about this distribution is the left-right symmetry of each row. Can you prove this symmetry? The proof that I know is quite involved. Can you come up with a closed expression for the distribution?

N, m	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	total
1	1	1															2
2	1	1	1	1													4
3	1	1	1	2	1	1	1										8
4	1	1	1	2	2	2	2	2	1	1	1						16
5	1	1	1	2	2	3	3	3	3	3	3	2	2	1	1	1	32

Table 3: Distribution statistics (open entries are 0)

6 Conclusion

Although Oskar's coin game is quite simple, there are some subtle sides to it. In this article, I have analyzed a few of these subtleties. Anyone able to shed more light on them is welcome to write me. I am particularly interested in the third problem.

As you probably have found out, Oskar's coin game is not a very interesting game to play, because each configuration allows only one move. However, playing the game backwards makes it more interesting, that is, by allowing a move from c to b if and only if $s.b = c$. In the backwards game, configurations allow either zero, one, or two moves. This could, for instance, be played as a two-person game: players take alternate moves and the player that cannot move loses. I have not analyzed the backwards game in detail, but it made its appearance when I looked into the third question about symmetry.

I wish to acknowledge Gerard Wiltink for his help in simplifying the solution to the first problem.