A puzzle on coin flipping

"There are 2^n coins arranged on a roundtable. You have been blindfolded, so you do not know the orientations of the coins (heads or tails). Each turn, you are allowed to flip any number of coins, but after the turn the table will be rotated by a random *angle. You win if the coins are either all heads or all tails. (You win automatically if the initial orientations are all heads or all tails.)*
Is there an algorithm to guarantee victory?"

(NUS CS1010S 2022 Sem 1, Side Quest 7.1; question phrasing is mine)

The puzzle can be viewed from a different perspective: rather than rotating the table, we will keep it fixed. Rather, the choice of coins to be flipped is first rotated by a random amount, then the flip is applied. In other words, the player can only choose which coins to flip, up to a rotation of the table. This observation makes the puzzle more amenable to analysis.

Notation

The coin orientations on the ith turn can be represented as a length 2^n bitstring x_i. The player's choice of coins is represented by another bitstring t_{i+1}, and based on our new perspective on the puzzle we have $x_{i+1} = x_i + r(t_{i+1})$, where r is a random bitstring rotation.* The player wins if $x_i = \mathbf{0}$ or $\mathbf{1}$ for some i (where $\mathbf{0}$ and $\mathbf{1}$ are the bitstrings with all 0s and all 1s respectively). I define an *algorithm* to be a sequence of bitstrings $(t_1, t_2, \ldots, t_n)$; for it to considered a valid algorithm, victory must be achieved by turn n.

For $n = 1$ it is obvious that (10) is an algorithm. Already the $n = 2$ case is not so straightforward, but it turns out that one algorithm is (A, B, A, C, A, B, A) or more compactly $ABACABA$, where

$$A = 1010, \quad B = 1100, \quad C = 1000.$$

Here's a sample run where the initial orientation is $x_0 = 0001$:

$$\begin{aligned} x_0 &= 0001, & x_1 &= 0001 + 0101 = 0100, & x_2 &= 0100 + 0110 = 0010 \\ x_3 &= 0010 + 1010 = 1000, & x_4 &= 1000 + 0100 = 1100, & x_5 &= 1100 + 1010 = 0110 \\ x_6 &= 0110 + 0011 = 0101, & x_7 &= 0101 + 0101 = 0000. \end{aligned}$$

Just nice, victory is achieved on the very last turn.

Cumulative effects and transversals

The 'cumulative effect' of the player's first i moves is equal to the bitstring

$$x_i - x_0 = r_1(t_1) + r_2(t_2) + \ldots + r_i(t_i), \quad r_1, \ldots, r_i \text{ are random rotations.}$$

Definition. *Let P_n be the set of length 2^n bitstrings. A transversal of P_n is a subset $T \subseteq P_n \setminus \{\mathbf{0}, \mathbf{1}\}$ such that for all $x \in P_n$, T contains exactly one of $\{x, \neg x\}$.*

Lemma. *If the set of cumulative effects is a transversal of P_n, then victory is guaranteed.*

Proof. All we need is for one of the cumulative effects $x_i - x_0$ to be either x_0 or $\neg x_0$, so that $x_i = x_0 + (x_i - x_0) = \mathbf{0}$ or $\mathbf{1}$. Furthermore, we know that x_0 is neither $\mathbf{0}$ nor $\mathbf{1}$, otherwise the player would have already won. ∎

Indeed, in the sample run above the cumulative effects are $(0101, 0011, 1001, 1101, 0111, 0100, 0001)$, which is a transversal of P_4:

$$\begin{array}{c|ccccccc} 0000 & \mathbf{0001} & 0010 & \mathbf{0100} & 1000 & \mathbf{0011} & 0110 & \mathbf{0101} \\ 1111 & 1110 & \mathbf{1101} & 1011 & \mathbf{0111} & 1100 & \mathbf{1001} & 1010 \end{array}$$

The rest of the solution is devoted to inductively constructing an algorithm whose cumulative effects are always a transversal.

* Here addition is not the usual binary addition with carrying, but rather the digitwise, mod 2 addition.

The solution

I will introduce a crucial device, which is a tower of subsets associated with each n. Here is what the tower looks like for small n:

$n = 1$: $\{00, 11\} \subseteq \{00, 11, 01, 10\}$

$n = 2$: $\{0000, 1111\} \subseteq \{0000, 1111, 0101, 1010\} \subseteq \{0000, 1111, 0101, 1010, 0011, 1100, 0110, 1001\} \subseteq P_2$

Already the $n = 3$ case is too much to show, but we can define the tower recursively using some new notation. Given two bitstrings x, y, let xy denote their concatenation, and let x^2 denote xx. Given a subset $S \subseteq P_n$, define S^2 to be the set

$$\{xx \mid x \in S\} \subseteq P_{n+1},$$

and S' to be the set

$$\{xy \mid x, y \in S \text{ and } x - y \in S\} \subseteq P_{n+1}.$$

For example, $\{\mathbf{0}\}' = P_n^2$ and $\{\mathbf{0}, \mathbf{1}\}' = \{x(\neg x) \mid x \in P_n\}$. If $S_1 \subseteq S_2 \subseteq \ldots \subseteq S_k$ is the tower of subsets for n, then the tower of subsets for $n + 1$ is given by

$$S_1^2 \subseteq S_2^2 \subseteq \ldots \subseteq S_k^2 \subseteq S_1' \subseteq S_2' \subseteq \ldots \subseteq S_k'.$$

In particular the length of the tower is always doubled, and therefore $k = 2^n$. For $n = 2$, we can rewrite the tower in this form:

$$\{00, 11\}^2 \subseteq \{00, 11, 01, 10\}^2 \subseteq \{00, 11\}' \subseteq \{00, 11, 01, 10\}' = P_2.$$

I can now give the outline of the solution, which builds up a transversal of P_{n+1} via transversals of the subsets in the tower.

1. By induction, assume that there is an algorithm $(t_1, \ldots, t_n)$ such that its cumulative effects are a transversal of P_n. (Next time I'll just call this a *good algorithm.*) Additionally assume that $t_i \in P_{n-1}^2$ for each i. Then there is a good algorithm for P_n^2, which runs in the same number of steps.
2. Define $S_0' = \{\mathbf{0}\}' = P_n^2$ and let S_i' be as above for $i \geq 1$. If there is a good algorithm for S_i' running in k steps, then there is a good algorithm for S_{i+1}' running in $2k + 1$ steps.

Small note: for the phrase 'good algorithm for S_i'' to make sense, the notion of a transversal of S_i' has to make sense, and this requires S_i' to be closed under complements. Luckily, it is:

Lemma. *S' is closed under complements for any subset $S \subseteq P_n$.*

Proof. $xy \in S' \implies x - y \in S \implies \neg x - \neg y = (1 - x) - (1 - y) \in S \implies \neg(xy) = (\neg x)(\neg y) \in S'$. ∎

Proof of 1. Let $(t_1, \ldots, t_k)$ be a good algorithm for P_n, where each t_i is in P_{n-1}^2. Then I claim $(t_1^2, \ldots, t_k^2)$ is a good algorithm for P_n^2. Each cumulative effect has the form $C_i = \sum_i r_i(t_i^2)$ where each r_i is a random rotation of a length 2^{n+1} bitstring. It is not too hard to see that $r_i(t_i^2) = r_i'(t_i)^2$ for some rotation r_i' of length 2^n bitstrings (since $t_i \in P_{n-1}^2$), so we can rewrite $C_i = \sum_i r_i'(t_i)^2$. Since the cumulative effects $\sum_i r_i'(t_i)$ are a transversal of P_n, the C_i's clearly are a transversal of P_n^2. ∎

Proof of 2. Note that S_{i+1} is twice as big as S_i, so S_{i+1}' is twice as big as S_i'. For any $h \in S_{i+1}' \setminus S_i'$, addition by h defines a bijection between $S_i' \to S_{i+1}' \setminus S_i'$, and additionally it preserves pairs of complementary bitstrings. For example, take the case $n = 2$, $S_i' = \{00\}'$, $h = 1100 \in \{00, 11\}' \setminus \{00\}'$. Addition by h defines the correspondence

$$(0000, 1111) \mapsto (1100, 0011), \quad (0101, 1010) \mapsto (1001, 0110).$$

Both properties are straightforward to prove: addition by h is bijective since it is its own inverse, and complementary bitstrings are preserved since $\neg(x + h) = 1 - (x + h) = (1 - x) + h = \neg x + h$.

If $A = (t_1, \ldots, t_k)$ is a good algorithm for S_i', I claim that (A, h, A) is a good algorithm for S_{i+1}'. Let $C_i = \sum_i r_i(t_i)$ be the ith cumulative effect of A. Then the cumulative effects of (A, h, A) are

$$\big(C_1, C_2, \ldots, C_k, C_k + r(h), C_k + r(h) + s_1(t_1), \ldots, C_k + r(h) + s_k(t_k)\big),$$

where r is a random rotation and s_i are a new set of random rotations different from r_i. Since $(C_1, C_2, \ldots, C_k)$ are known to be a transversal of S_i', we need only show that $\big(C + s_1(t_1), \ldots, C + s_k(t_k)\big)$ is a transversal of $S_{i+1}' \setminus S_i'$, where $C = C_k + r(h)$. (Note $S_{i+1}' \setminus S_i'$ is indeed closed under complements.)

Before continuing a few facts are needed.

Lemma. *S'_i is a subgroup of P_{n+1}, i.e. it is closed under addition and inverses.*

Proof. I will prove the general statement that for all n, the abovementioned tower of subsets is in fact a tower of subgroups of P_n. It suffices to prove closure under addition, because every bitstring is its own inverse. This is done by induction on n. For the base case $n = 1$, it is obvious that $\{00, 11\}$ and $\{00, 11, 01, 10\}$ are subgroups of P_1. The inductive step is to show that if S is a subgroup of P_n, then S^2 and S' are subgroups of P_{n+1}. I leave this as an exercise for the reader. ∎

Lemma. *$S'_{i+1} \setminus S'_i$ is closed under rotation. In particular $r(h) \in S'_{i+1} \setminus S'_i$.*

Proof. First, we induct to show that for all n, every subset in the tower is closed under rotation. For the base case $n = 2$, clearly $\{00, 11\}$ and $\{00, 11, 01, 10\}$ are closed under rotation. The inductive step is to show that if S satisfy this property, then so do S^2 and S'. The former follows from the fact that for every rotation r there is some rotation r' such that $r(t^2) = r'(t)^2$. As for the latter, WLOG let r be a left shift by one digit. write $x = x_1 \dots x_n$ and $y = y_1 \dots y_n$, and define $z_i = x_i - y_i$. If $xy \in S'$, then $z_1 z_2 \dots z_n \in S$, so $z_2 \dots z_n z_1 \in S$, and therefore $r(xy) = r(x_1 \dots x_n y_1 \dots y_n) = (x_2 \dots x_n y_1)(y_2 \dots y_n x_1) \in S'$. This completes the induction.

Now, given $x \in S'_{i+1} \setminus S'_i$ we must have $r(x) \in S'_{i+1}$ (since S'_{i+1} is closed under rotation). If furthermore $r(x) \in S'_i$, then $r^{-1}(r(x)) = x \in S_i$ where r^{-1} is the inverse rotation, since S'_i is closed under rotation. Therefore, $r(x) \in S'_{i+1} \setminus S'_i$ as desired. ∎

Lemma. *$C \in S'_{i+1} \setminus S'_i$.*

Proof. Otherwise $r(h) = C - C_k \in S'_i$, which is not true. ∎

Lemma. *If $h \in S'_{i+1} \setminus S'_i$, then addition by h sends transversals of S'_i to transversals of $S'_{i+1} \setminus S'_i$.*

Proof. Let T be a transversal of S'_i; we want to show $h + T = \{h + t \mid t \in T\}$ is a transversal of $S'_{i+1} \setminus S'_i$.

For each $x \in S'_{i+1} \setminus S'_i$, we can write $x = h + x'$ for some $x' \in S'_i$, and also we have $\neg x = h + \neg x'$. Since addition by h is a bijection and T contains exactly one of $\{x', \neg x'\}$, it follows that $h + T$ contains exactly one of $\{x, \neg x\}$, as desired. ∎

The rest of the proof falls out these lemmas. We have that $\big(s_1(t_1), \dots, s_k(t_k)\big)$ is a transversal of S'_i. (This is different from $(C_1, \dots, C_k)$ since the rotations s_i are possibly different from the r_i's.) Setting $h = C$ in the last lemma, we can conclude that $\big(C + s_1(t_1), \dots, C + s_k(t_k)\big)$ is a transversal of $S'_{i+1} \setminus S'_i$ as desired. ∎

Theorem. *For all n, there exists a good algorithm for P_n that takes $2^{2^n - 1} - 1$ turns.*

Proof. The existence of the algorithm follows from the two steps. As for the number of turns, assume by induction that the algorithm for P_n takes $2^{2^n - 1} - 1$ turns.

Then the algorithm for P_{n+1} is obtained by running step 2 for 2^n times (since the tower for P_{n+1} has length 2^{n+1}), and each time the number of steps increases according to the map $f(k) = 2k + 1$. We have $f(2^a - 1) = 2^{a+1} - 1$ for all a, so $f^{2^n}(2^{2^n - 1} - 1) = 2^{2^n - 1 + 2^n} - 1 = 2^{2^{n+1} - 1} - 1$. ∎

Way Yan
14/9/2022