Combinatorial Number System(CNS) for Mapping Combinations to Natural Numbers

The Play family DAC creators can select whatever kind of rules they like, color counts, ball count, k to select, prize definition etc. The outer rule layer will also provide a simple API for inheritors to customize prize rules. Here is an example to demonstrate what a rule will look like.

On the blockchain, there are only ranked lucky numbers and winning number, no rule configuration and jackpots drawing information. So all transactions much the core layer and part of rule could be accepted.

The communication and cooperation between these two layers are using the combinatorial number system (CNS) to map them. CNS will be described below.

CNS is used to map the rule layer model to the core layer's RNG process This is especially useful in mapping the lottery combination inputs to natural numbers, which can help customize different rules with only some simple script language or configuration file.

In mathematics, and in particular in combinatorics, the **combinatorial number system** of degree *k* (for some positive integer *k*), also referred to as **combinadics**, is a correspondence between natural numbers (taken to include 0) *N* and *k*-combinations, represented as strictly decreasing sequences $c_k > \ldots > c_2 > c_1 \geq 0$. Since the latter are strings of numbers, one can view this as a kind of numerical system for representing *N*, although the main utility is representing a *k*-combination by *N* rather than the other way around. Distinct numbers correspond to distinct *k*-combinations, and produce them in a lexicographical order; moreover the numbers less than $\binom{k}{n}$ correspond to all *k*-combinations of { 0, 1, ..., *n* − 1}. The correspondence does not depend on the size *n* of the set that the *k*-combinations are taken from, so it can be interpreted as a map from **N** to the *k*-combinations taken from **N**; in this view the correspondence is a bijection.
The number *N* corresponding to $(c_k,\ldots,c_2,c_1)$ is given by

$$N = \binom{C_k}{k} + \cdots + \binom{C_2}{2} + \binom{C_1}{1}$$

The fact that a unique sequence corresponds to a number *N* was observed by D.H. Lehmer.[1] Indeed a greedy algorithm finds the *k*-combination corresponding to *N*: take c_k maximal with $\binom{C_k}{k} \leq N$, then take c_{k-1} maximal with $\binom{C_{k-1}}{k-1} \leq N - \binom{C_k}{k}$, and so forth. Finding the number *N*, using the formula above, from the *k*-combination $(c_k,\ldots,c_2,c_1)$ is also known as "ranking", and the opposite operation (given by the greedy algorithm) as "unranking"; these operations are known by those names in most Computer algebra systems, and in computational mathematics.

Ranking/Unranking algorithm for multi-color balls lottery:

1. Given that there are c kinds of colored balls, each kind of balls has B_i balls, numbered from 0, 1, … B_i, users have to choose K_i balls from each kind ball as the combination of this color ball's combinations.
2. The final user selected balls are actually ball combination groups, each color has a related combination group, each group i is a combination with K_i balls select, such as $(C_1, C_2, C_3, \ldots, C_{K_i})_{G_i}$.

3. First, mapping the G_i group combination to nature number using ranking algorithm of CNS, The number RG_i is the corresponding ranking number calculated according to $(C_1, C_2, C_3, \ldots, C_{K_i})_{G_i}$, which are ordered incrementally.

$$RG_i = \binom{C_{K_i}}{K_i} + \cdots + \binom{C_2}{2} + \binom{C_1}{1}$$

4. For each group, there is a probability space of

$$SG_i = \binom{B_i}{K_i}$$

5. Then the final ranking number R is calculated as following

$$R = \sum_{i=0}^{c-1} (RG_i \times \prod_{x=0}^{i-1} SG_x)$$

6. The unranking is the reverse of the 3 to 5 steps.

In the ticket purchase block, the user selected combination group is converted to its ranking number as the lucky number which will be broadcasted in the claim_ticket transaction, further to be stored in blockchain.

After the winning number is out, and before the jackpots matching process, the winning number will be converted to combination groups using the unranking algorithm, the winning combination groups are used as the input of draw jackpots in rule layer. By the same token, because nodes can only know the ranking of lucky combination groups, and can not generate the jackpot just according to ranking winning and lucky number, the lucky number also needs to be unranked before match.

The winning number is random number created by the RNG and is of type u64_t. However, before the unranking, we should notice that there is a maximum probability space to the possible combination groups of

$$TRG = \prod_{i=0}^{c-1} SG_i$$

The winning number need to do $(mod\ TRG)$ before unranking.