IMPARTIAL GAMES

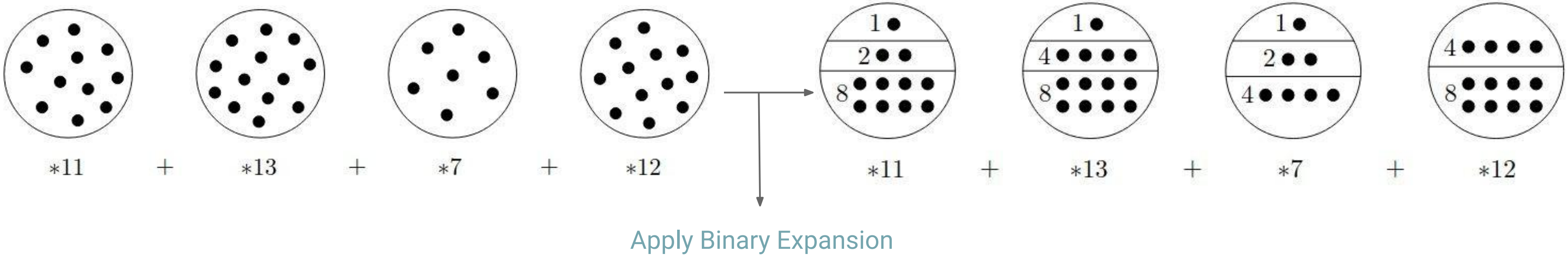
- Normal-Game
- Same moves
  - Identity of the player does no matter
  - There are no positions type L o R

Binary Expansion

- Representation of n as sum of distinct powers of two
  - 45 = 32+13 = 32+8+5 = 32+8+4+1

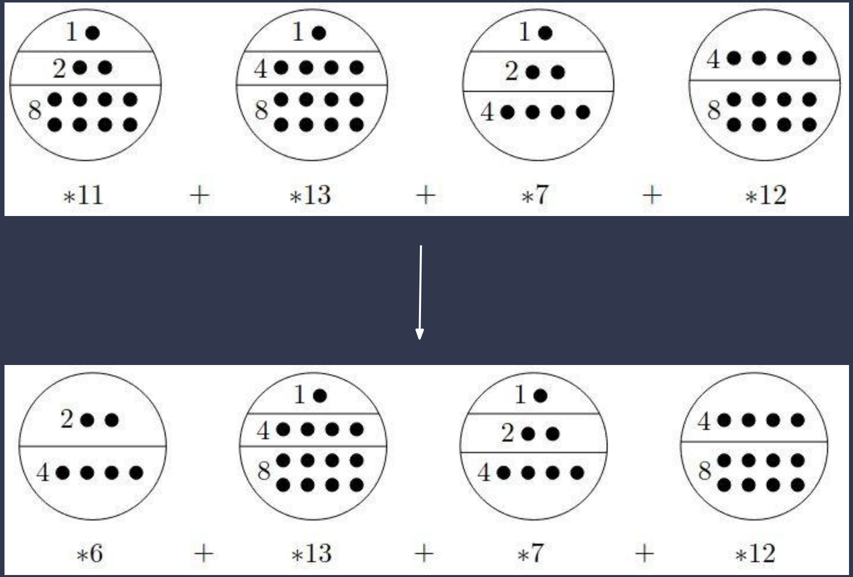
Type of Positions

- Let a be a non-negative integer and let *a denote a Nim position of a single pile of a stones



- A position *a1+...+*ak is balanced if, for every power of two, the total number of subpiles of that sizes is even

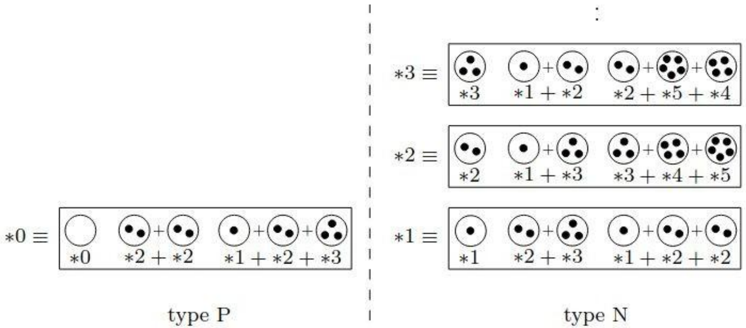
Balancing a Position



KEY → Take 2^m the largest power of 2 which odd number of subpiles

- Balanced position → Type P
- Unbalanced position → Type N

- The Nim-Sum of nonnegative integers a1⊕...⊕ak is a non-negative integer b s.t. if 2^j appears in the binary expansion of b ↔ 2^j appears an odd numer of time in the expansion a1,...,ak.
  - 13⊕19⊕10 = (8+4+1)⊕(16+2+1)⊕(8+2) = = 4+16 = 20
- If b = a1⊕...⊕ak, then *a1+...+*ak = b, and b is called Nimber.



The Sprague-Grundy Theorem

- For normal-play games, a position $\underline{\alpha} = \{\beta 1,\ldots,\beta k \mid \gamma 1,\ldots,\gamma m\}$ (Louise | Richard)
  - Redundant notation for impartial games.

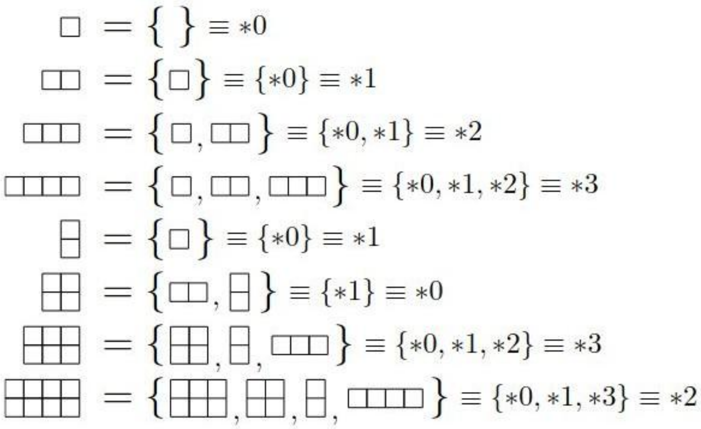
- For S = {a1,...,an} of non-negative integers, we define Minimal EXclued values (MEX) of S as the smallest integer b which is not in S (i.e. {0,1,2,5.8} is 3)
- The MEX Principle → Let $\alpha = \{\alpha 1,\ldots,\alpha k\}$ be a position in an impartial game and let $\alpha i \equiv {*}ai$, $1 \le i \le k$, then, $\alpha \equiv b$, where b is the MEX of the set $\{\alpha 1,\ldots,\alpha k\}$

- The Sprague-Grundy Theorem → Every position in an impartial game is equivalent to a nimber.

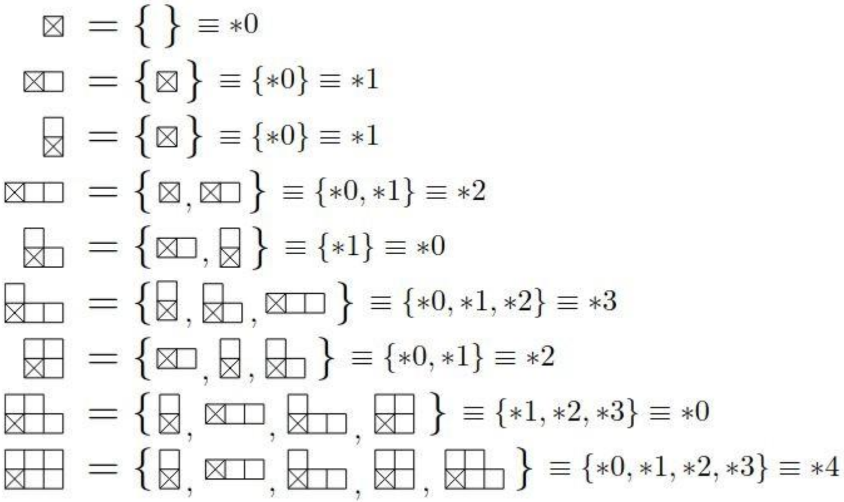
- We define a winning move in an impartial game to be any move to a position of type P. Note that winning moves only exists from positions of type N
  - In Nim, balancing always provides a winning move.

MEX Principle Examples

Example 3.14 (Chop). Here we work out the nimber equivalents for some small positions in Chop.



Example 3.15 (Chomp). Here we work out the nimber equivalents for some small positions in Chomp.

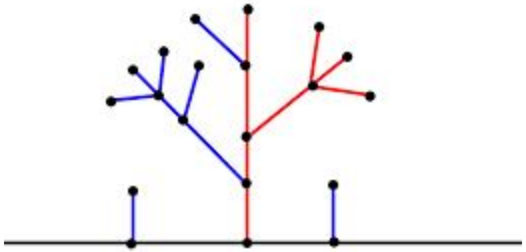


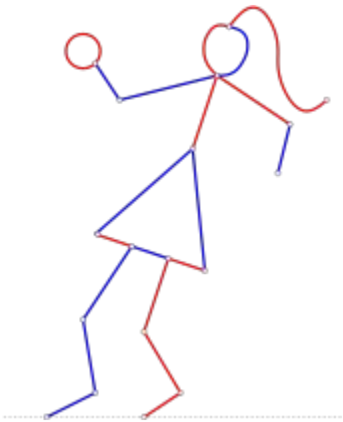
NIM Code in Python

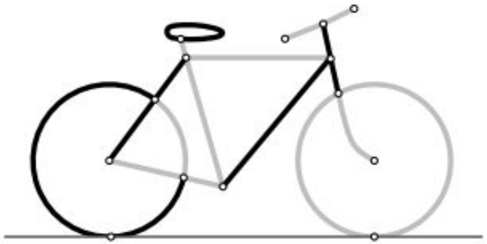
GitHub Code

1. Ask for players name
2. Use random number generator to generate between 2 to 5 piles and to generate between 1 to 8 stones in each pile.
3. Display the board in the following fashion: (Note: O is the letter capital O)
   Pile 1 : O O O
   Pile 2 : O
   …
4. Ask player 1 for the move – the pile number and the number of stones to remove. You need to make sure that if a user enters something invalid, whether the pile or stone number, you ask for input again until valid input is entered. Also, the program is to give a suggestion to player 1 as to the number of stones to remove and from which pile in order to win the game.
5. Repeat step 4
6. Repeat step 5 for player 2.
7. When the game terminates, you need to proclaim the winner.
8. You should ask, if the players want to play the game again; if so, repeat steps 1-9

Partizan games and Hackenbush

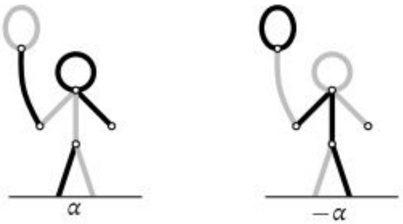






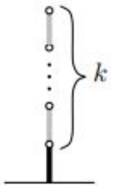
Algebra and integer positions

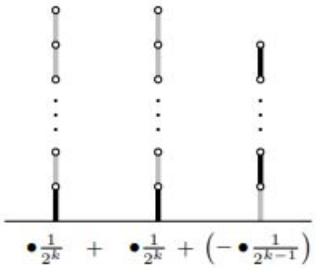
- We fundamentally consider the sum and negation operations when discussing about hackenbush
- Integer positions are representations of numbers as Hackenbush positions
- It exists a relation between the Hackenbush operator and sum and negation

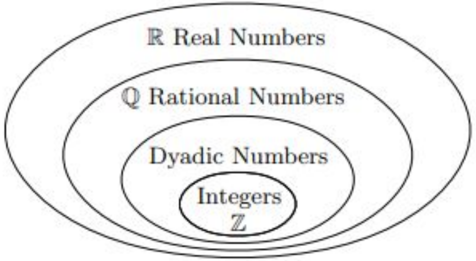


Fractional positions and dyadic numbers

- There exists certain positions which describe fractional numbers
- Algebra in these positions apply as usual
- Dyadic numbers are a special set which Will be used to explain these positions



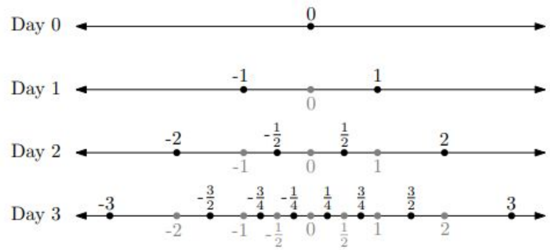


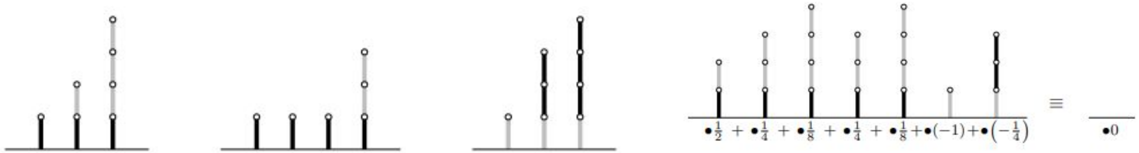


Birthday of a number and dyadic positions

- The concept of birthday of a dyadic number is key for later explanation of the simplicity principle

- Using the unique finite binary decomposition of dyadic number we can define dyadic positions.

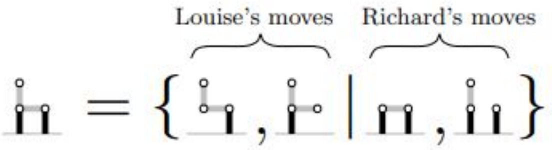




The Simplicity Principle

The simplicity principle is the most powerful tool available for us to decompose some Hacklebush positions in order to know which it is type. Given all the moves possible for a position in which we don't know its associated dyadic number, we can then in some cases obtain the actual dyadic number associated. So that we can easily obtain the position type inmediately

- q *is type* $\begin{cases} \text{L} & if \quad q > 0, \\ \text{P} & if \quad q = 0, \\ \text{R} & if \quad q < 0. \end{cases}$



Application of the simplicity principle

This is a step by step application of the simplicity principle. We begin constructing the figure step by step and obtaining the value of each construction so finally we can conclude which is the value o f the desired position.

Also it is interesting to mention that algebra here applies as usual, this means adding or negating positions will behave normally.