Samos Summer School on Combinatorics and Computer science

Combinatorial Game Theory

Part 1 : Impartial games and the Sprague-Grundy theorem

UNIVERSITY OF THE AEGEAN

ERUA European Reform University Alliance

Let's play games !

- In this course, we will play games... but more importantly, we will study some games from **mathematical** and **algorithmic** perspectives.

- Our main question will be:

Given a position of a game, what is the best move to play?

- We will focus on games with the following properties:
  - two players that alternate moves;
  - with perfect information, e.g. nothing is unknown to the players;
  - a winning and a losing outcome (with possible draws in some games).

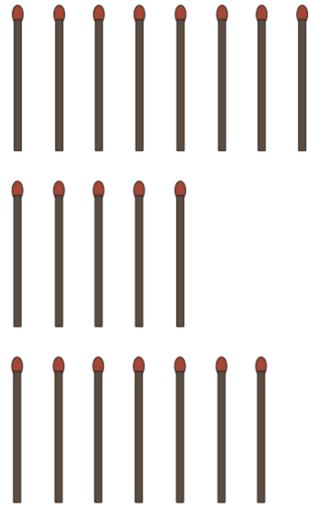
Examples

1. Subtraction Game

Remove a given number of tokens.
The player taking the last token wins.

Examples

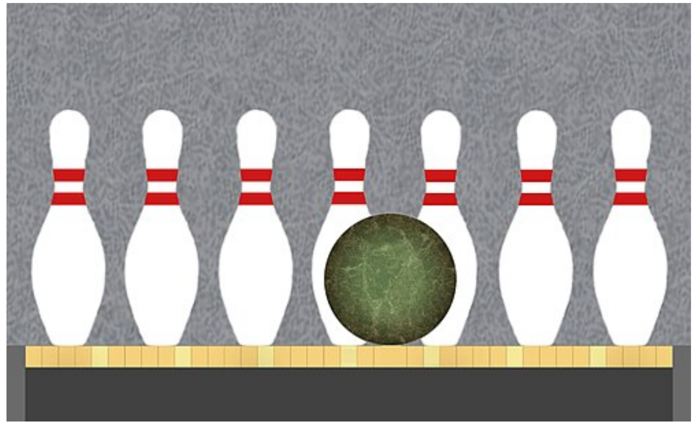
1. Subtraction Game
2. Nim



Choose one heap and remove any positive number of tokens.

Examples

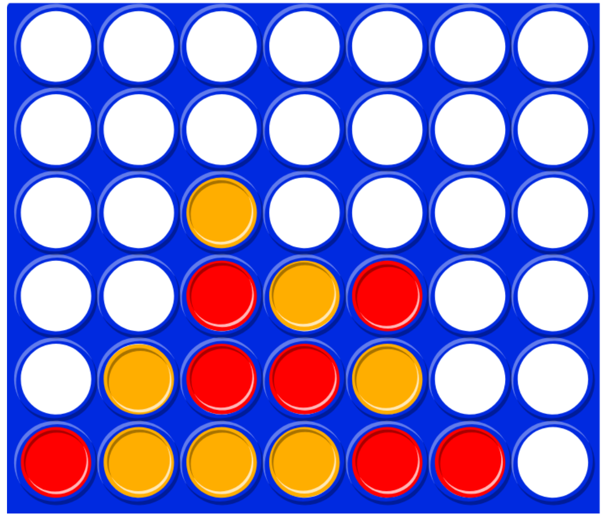
1. Subtraction Game
2. Nim
3. Kayles



Remove one pin or two adjacent pins.

Examples

1. Subtraction Game
2. Nim
3. Kayles
4. Connect 4



Align four pieces before your opponent.

Examples

1. Subtraction Game
2. Nim
3. Kayles
4. Connect 4
5. Chess

Checkmate your opponent's king.

Examples

1. Subtraction Game
2. Nim
3. Kayles
4. Connect 4
5. Chess
6. Go

Capture territory and surround your opponent.

Examples

1. Subtraction Game
2. Nim
3. Kayles
4. Connect 4
5. Chess
6. Go

- We will make a distinction between these games. Some of them are
  - **impartial**, that is both players have the same set of feasible moves for any position of the games. These will be studied in Part 1.
  - **partizan**, that is the set of moves depend on which player has to play. These will be studied in Part 2.

Combinatorial Game

- A proper definition of a **combinatorial game** G is the following : it is a game in which

  (1) There are 2 players that alternate moves.

  (2) There is a finite state of possible positions/configurations of the game.

  (3) The rules of the game specify for both players and each position which moves to other positions are legal. In an impartial game, the set of reachable positions is the same for the two players; whereas it is not for partizan games.

  (4) The game ends when a position is reached where no other move is possible for the player who has to move. The winner depends on the chosen rule, as specified below.

  (5) The game always ends in a finite number of moves, e.g there is no way to go from a position to the same position and cycle forever.

Combinatorial Game

- A proper definition of a **combinatorial game** G is the following : it is a game in which

  (1) There are 2 players that alternate moves.

  (2) There is a finite state of possible positions/configurations of the game.

  (3) The rules of the game specify for both players and each position which moves to other positions are legal. In an impartial game, the set of reachable positions is the same for the two players; whereas it is not for partizan games.

  (4) The game ends when a position is reached where no other move is possible for the player who has to move. The winner depends on the chosen rule, as specified below.

  (5) The game always ends in a finite number of moves, e.g there is no way to go from a position to the same position and cycle forever.

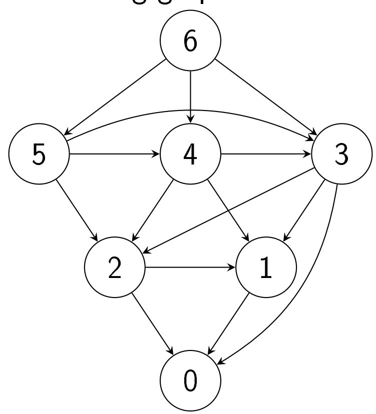
- An impartial combinatorial game (ICG) can be played following different rules :
  - **normal rules**, in which the last player to make a move is the winner.
  - **misère rules**, in which the last player to make a move is the loser.
  - Unless specified mention, we will use normal rules in the sequel of the course !

Graph Representation

- Every ICG can be represented as a finite directed graph (V, E), without any cycles, as follows :
  - the set of vertices V is the set of possible positions p of the game.
  - for any position p, denote by $H(p)$ the set of positions that can be reached from p using a legal move. Then for each p' in $H(p)$, there is a directed edge $p \to p'$ in the graph.

- **Exemple :** A **subtraction game** is a game in which there is a certain amount $n \in \mathbb{N}_{>0}$ and at each turn, the player can remove a quantity s of chips belonging to a certain fixed set S.

  For $n = 6$ and $S = \{1, 2, 3\}$, we obtain the following graph :

  where, instead of drawing 6 chips, we just represent the position of the game by the integer 6.

Subtraction game

- Let us analyze a subtraction game with $n = 21$ chips and $S = \{1, 2, 3\}$.
  - At each turn, a player can remove 1, 2 or 3 chips.
  - The player who takes the last chip wins !
- Assume Player 1 has to play :
  - with 1, 2 or 3 chips : he wins.
  - with 4 chips : he leaves either 1, 2 or 3 chips so the next player wins. It is a losing position!
  - with 5, 6 or 7 chips, you can move so that you leave only 4 and the next player is in a losing position, so you win !
  - with 8 chips, the next player will have 5, 6 or 7 and thus will win. So it is also a losing position.

Subtraction game

- Let us analyze a subtraction game with $n = 21$ chips and $S = \{1, 2, 3\}$.
  - At each turn, a player can remove 1, 2 or 3 chips.
  - The player who takes the last chip wins !
- Assume Player 1 has to play :
  - with 1, 2 or 3 chips : he wins.
  - with 4 chips : he leaves either 1, 2 or 3 chips so the next player wins. It is a losing position!
  - with 5, 6 or 7 chips, you can move so that you leave only 4 and the next player is in a losing position, so you win !
  - with 8 chips, the next player will have 5, 6 or 7 and thus will win. So it is also a losing position.
- If we keep going, we notice that all the positions n such that $n = 0 \pmod 4$ result in a losing position for the player.
- Your goal is then to put your opponent in one of these losing positions. With 21 chips, you can remove one to reach one of these target positions and win.

$$21 \xrightarrow{1,1} 20 \xrightarrow{2,3} 17$$

Subtraction game

- Let us analyze a subtraction game with $n = 21$ chips and $S = \{1, 2, 3\}$.
  - At each turn, a player can remove 1, 2 or 3 chips.
  - The player who takes the last chip wins !
- Assume Player 1 has to play :
  - with 1, 2 or 3 chips : he wins.
  - with 4 chips : he leaves either 1, 2 or 3 chips so the next player wins. It is a losing position!
  - with 5, 6 or 7 chips, you can move so that you leave only 4 and the next player is in a losing position, so you win !
  - with 8 chips, the next player will have 5, 6 or 7 and thus will win. So it is also a losing position.
- If we keep going, we notice that all the positions n such that $n = 0 \pmod 4$ result in a losing position for the player.
- Your goal is then to put your opponent in one of these losing positions. With 21 chips, you can remove one to reach one of these target positions and win.

$$21 \xrightarrow{1,1} 20 \xrightarrow{2,3} 17 \xrightarrow{1,1} 16$$

Subtraction game

- Let us analyze a subtraction game with $n = 21$ chips and $S = \{1, 2, 3\}$.
  - At each turn, a player can remove 1, 2 or 3 chips.
  - The player who takes the last chip wins !
- Assume Player 1 has to play :
  - with 1, 2 or 3 chips : he wins.
  - with 4 chips : he leaves either 1, 2 or 3 chips so the next player wins. It is a losing position!
  - with 5, 6 or 7 chips, you can move so that you leave only 4 and the next player is in a losing position, so you win !
  - with 8 chips, the next player will have 5, 6 or 7 and thus will win. So it is also a losing position.
- If we keep going, we notice that all the positions n such that $n = 0 \pmod 4$ result in a losing position for the player.
- Your goal is then to put your opponent in one of these losing positions. With 21 chips, you can remove one to reach one of these target positions and win.

$$21 \xrightarrow{1,1} 20 \xrightarrow{2,3} 17 \xrightarrow{1,1} 16 \xrightarrow{2,1} 15$$

Subtraction game

- Let us analyze a subtraction game with $n = 21$ chips and $S = \{1, 2, 3\}$.
  - At each turn, a player can remove 1, 2 or 3 chips.
  - The player who takes the last chip wins !
- Assume Player 1 has to play :
  - with 1, 2 or 3 chips : he wins.
  - with 4 chips : he leaves either 1, 2 or 3 chips so the next player wins. It is a losing position!
  - with 5, 6 or 7 chips, you can move so that you leave only 4 and the next player is in a losing position, so you win !
  - with 8 chips, the next player will have 5, 6 or 7 and thus will win. So it is also a losing position.
- If we keep going, we notice that all the positions n such that $n = 0 \pmod 4$ result in a losing position for the player.
- Your goal is then to put your opponent in one of these losing positions. With 21 chips, you can remove one to reach one of these target positions and win.

$$21 \xrightarrow{1,1} 20 \xrightarrow{2,3} 17 \xrightarrow{1,1} 16 \xrightarrow{2,1} 15 \xrightarrow{1,3} 12$$

Subtraction game

- Let us analyze a subtraction game with $n = 21$ chips and $S = \{1, 2, 3\}$.
  - At each turn, a player can remove 1, 2 or 3 chips.
  - The player who takes the last chip wins !
- Assume Player 1 has to play :
  - with 1, 2 or 3 chips : he wins.
  - with 4 chips : he leaves either 1, 2 or 3 chips so the next player wins. It is a losing position!
  - with 5, 6 or 7 chips, you can move so that you leave only 4 and the next player is in a losing position, so you win !
  - with 8 chips, the next player will have 5, 6 or 7 and thus will win. So it is also a losing position.
- If we keep going, we notice that all the positions n such that $n = 0 \pmod 4$ result in a losing position for the player.
- Your goal is then to put your opponent in one of these losing positions. With 21 chips, you can remove one to reach one of these target positions and win.

$$21 \xrightarrow{1,1} 20 \xrightarrow{2,3} 17 \xrightarrow{1,1} 16 \xrightarrow{2,1} 15 \xrightarrow{1,3} 12 \xrightarrow{2,2} 10$$

Subtraction game

- Let us analyze a subtraction game with $n = 21$ chips and $S = \{1, 2, 3\}$.
  - At each turn, a player can remove 1, 2 or 3 chips.
  - The player who takes the last chip wins !
- Assume Player 1 has to play :
  - with 1, 2 or 3 chips : he wins.
  - with 4 chips : he leaves either 1, 2 or 3 chips so the next player wins. It is a losing position!
  - with 5, 6 or 7 chips, you can move so that you leave only 4 and the next player is in a losing position, so you win !
  - with 8 chips, the next player will have 5, 6 or 7 and thus will win. So it is also a losing position.
- If we keep going, we notice that all the positions n such that $n = 0 \pmod 4$ result in a losing position for the player.
- Your goal is then to put your opponent in one of these losing positions. With 21 chips, you can remove one to reach one of these target positions and win.

$$21 \xrightarrow{1,1} 20 \xrightarrow{2,3} 17 \xrightarrow{1,1} 16 \xrightarrow{2,1} 15 \xrightarrow{1,3} 12 \xrightarrow{2,2} 10 \xrightarrow{1,2} 8$$

Subtraction game

- Let us analyze a subtraction game with $n = 21$ chips and $S = \{1, 2, 3\}$.
  - At each turn, a player can remove 1, 2 or 3 chips.
  - The player who takes the last chip wins !
- Assume Player 1 has to play :
  - with 1, 2 or 3 chips : he wins.
  - with 4 chips : he leaves either 1, 2 or 3 chips so the next player wins. It is a losing position!
  - with 5, 6 or 7 chips, you can move so that you leave only 4 and the next player is in a losing position, so you win !
  - with 8 chips, the next player will have 5, 6 or 7 and thus will win. So it is also a losing position.
- If we keep going, we notice that all the positions n such that $n = 0 \pmod 4$ result in a losing position for the player.
- Your goal is then to put your opponent in one of these losing positions. With 21 chips, you can remove one to reach one of these target positions and win.

$$21 \xrightarrow{1,1} 20 \xrightarrow{2,3} 17 \xrightarrow{1,1} 16 \xrightarrow{2,1} 15 \xrightarrow{1,3} 12 \xrightarrow{2,2} 10 \xrightarrow{1,2} 8 \xrightarrow{2,3} 5$$

Subtraction game

- Let us analyze a subtraction game with $n = 21$ chips and $S = \{1, 2, 3\}$.
  - At each turn, a player can remove 1, 2 or 3 chips.
  - The player who takes the last chip wins !
- Assume Player 1 has to play :
  - with 1, 2 or 3 chips : he wins.
  - with 4 chips : he leaves either 1, 2 or 3 chips so the next player wins. It is a losing position!
  - with 5, 6 or 7 chips, you can move so that you leave only 4 and the next player is in a losing position, so you win !
  - with 8 chips, the next player will have 5, 6 or 7 and thus will win. So it is also a losing position.
- If we keep going, we notice that all the positions n such that $n = 0 \pmod 4$ result in a losing position for the player.
- Your goal is then to put your opponent in one of these losing positions. With 21 chips, you can remove one to reach one of these target positions and win.

$$21 \xrightarrow{1,1} 20 \xrightarrow{2,3} 17 \xrightarrow{1,1} 16 \xrightarrow{2,1} 15 \xrightarrow{1,3} 12 \xrightarrow{2,2} 10 \xrightarrow{1,2} 8 \xrightarrow{2,3} 5 \xrightarrow{1,1} 4$$

Subtraction game

- Let us analyze a subtraction game with $n = 21$ chips and $S = \{1, 2, 3\}$.
  - At each turn, a player can remove 1, 2 or 3 chips.
  - The player who takes the last chip wins !
- Assume Player 1 has to play :
  - with 1, 2 or 3 chips : he wins.
  - with 4 chips : he leaves either 1, 2 or 3 chips so the next player wins. It is a losing position!
  - with 5, 6 or 7 chips, you can move so that you leave only 4 and the next player is in a losing position, so you win !
  - with 8 chips, the next player will have 5, 6 or 7 and thus will win. So it is also a losing position.
- If we keep going, we notice that all the positions n such that $n = 0 \pmod 4$ result in a losing position for the player.
- Your goal is then to put your opponent in one of these losing positions. With 21 chips, you can remove one to reach one of these target positions and win.

$$21 \xrightarrow{1,1} 20 \xrightarrow{2,3} 17 \xrightarrow{1,1} 16 \xrightarrow{2,1} 15 \xrightarrow{1,3} 12 \xrightarrow{2,2} 10 \xrightarrow{1,2} 8 \xrightarrow{2,3} 5 \xrightarrow{1,1} 4 \xrightarrow{2,1} 3$$

Subtraction game

- Let us analyze a subtraction game with $n = 21$ chips and $S = \{1, 2, 3\}$.
  - At each turn, a player can remove 1, 2 or 3 chips.
  - The player who takes the last chip wins !
- Assume Player 1 has to play :
  - with 1, 2 or 3 chips : he wins.
  - with 4 chips : he leaves either 1, 2 or 3 chips so the next player wins. It is a losing position!
  - with 5, 6 or 7 chips, you can move so that you leave only 4 and the next player is in a losing position, so you win !
  - with 8 chips, the next player will have 5, 6 or 7 and thus will win. So it is also a losing position.
- If we keep going, we notice that all the positions n such that $n = 0 \pmod 4$ result in a losing position for the player.
- Your goal is then to put your opponent in one of these losing positions. With 21 chips, you can remove one to reach one of these target positions and win.

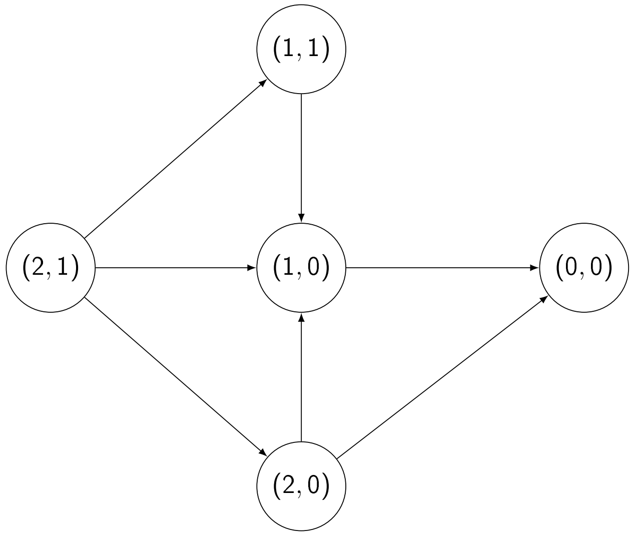
$$21 \xrightarrow{1,1} 20 \xrightarrow{2,3} 17 \xrightarrow{1,1} 16 \xrightarrow{2,1} 15 \xrightarrow{1,3} 12 \xrightarrow{2,2} 10 \xrightarrow{1,2} 8 \xrightarrow{2,3} 5 \xrightarrow{1,1} 4 \xrightarrow{2,1} 3 \xrightarrow{1,3} 0$$

Nim game

- **Notation :** The Nim game with 3 heaps having respectively 9, 7 and 5 stones will be represented by the 3-tuple $(9, 7, 5)$.

Nim game

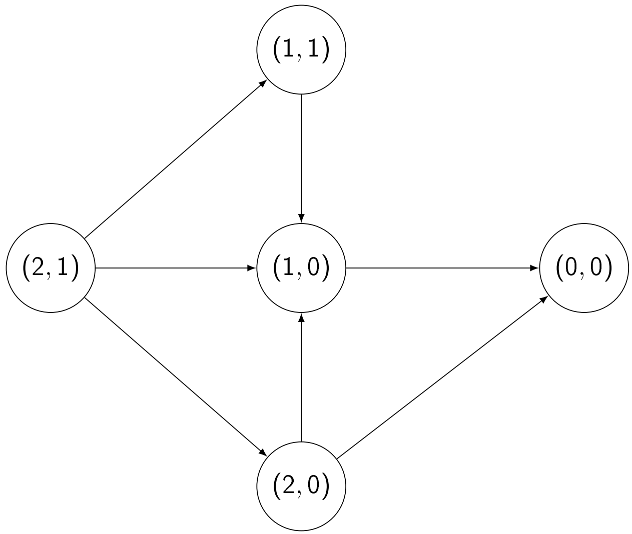
- **Notation :** The Nim game with 3 heaps having respectively 9, 7 and 5 stones will be represented by the 3-tuple $(9, 7, 5)$.
- **Exemple :** The situation of the game of Nim $(2, 1)$ is the following :



- **Question :** What should Player 1 do ?

Nim game

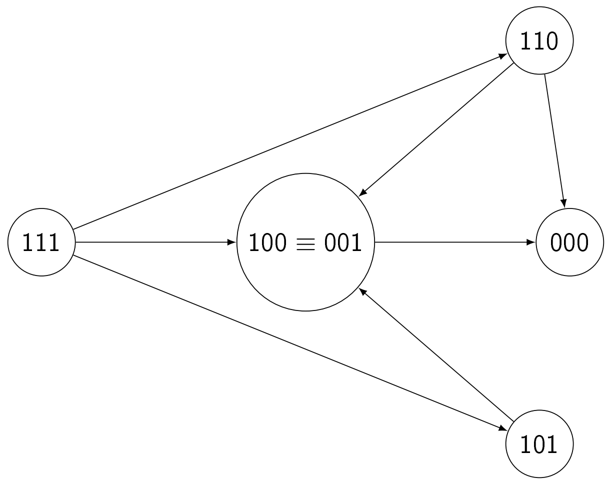
- **Notation** : The Nim game with 3 heaps having respectively 9, 7 and 5 stones will be represented by the 3-tuple $(9, 7, 5)$.
- **Exemple** : The situation of the game of Nim $(2, 1)$ is the following :



- **Question** : What should Player 1 do ? **Move to** $(1, 1)$ **!**

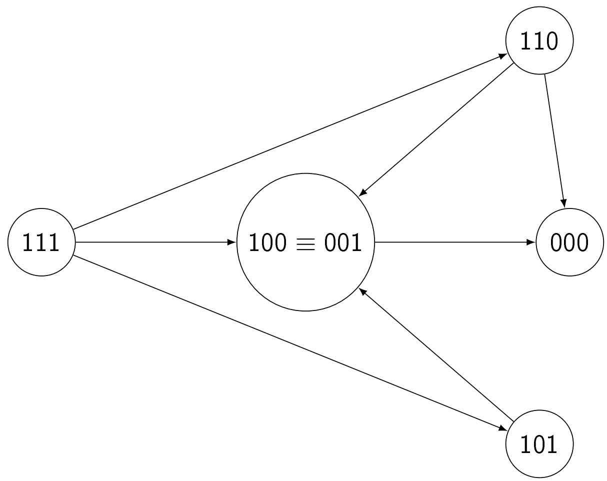
Kayles game

- In the Kayles game, we start with a row of n bowling pins for some integer n. A move consists in either :
  - knock down one pin,
  - knock down two adjacent pins.
- One of these moves might cause the game to split into several smaller parts. We will discuss that later !
- **Exemple :** The situation of the game of Kayles with 3 pins is the following :



Kayles game

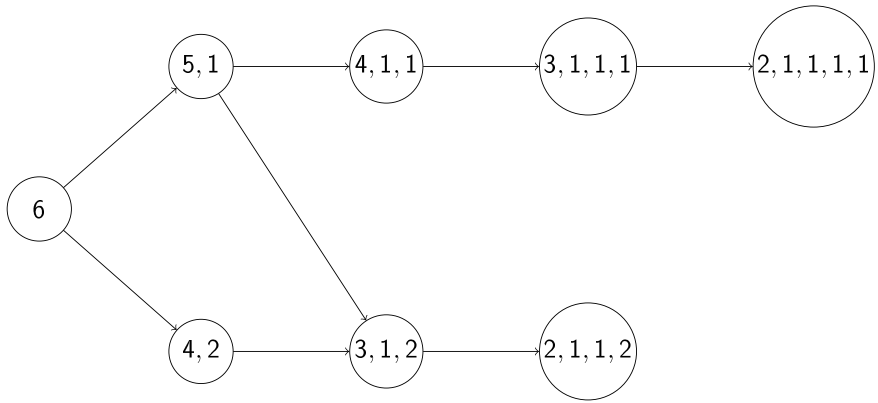
- In the Kayles game, we start with a row of n bowling pins for some integer n. A move consists in either :
  - knock down one pin,
  - knock down two adjacent pins.
- One of these moves might cause the game to split into several smaller parts. We will discuss that later !
- Exemple : The situation of the game of Kayles with 3 pins is the following :



so a winning move is to remove the middle pin.

Grundy's game

- In Grundy's game, as in Nim, there is a number of chips distributed in several heaps. A turn is to divide one heap into two **different-sized** heaps. The player who can not divide any pile loses.
- **Example :** You can divide for instance a heap of 6 chips into two heaps of 5 and 1 (or 4 and 2) but not 3 and 3.
- The situation for Grundy's game with $n = 6$ is the following :



- The winning strategy is thus to move from 6 to $(4, 2)$, then your opponent has to move to $(3, 1, 2)$ and you win.

P-positions and N-positions

- **Question :** how to determine whether we are in a winning/losing position and what is the best move to do in general ?
- For ICG, we can always determine which one of the two players is winning, and also calculate the optimal strategy for both players (provided they play optimally).
- We call a position of a game a :
  - **P-position** if it is a winning position for the **P**revious player. In other terms, it is a losing position for the player who has to play.
  - **N-position** if it is a winning position for the **N**ext player. In other terms, it is a winning position for the player who has to play.

Labeling procedure for P/N-positions

- **Theorem :**

  Every position of an ICG is either a P-position or a N-position.

Labeling procedure for P/N-positions

- **Theorem :**

  > Every position of an ICG is either a P-position or a N-position.

- The following inductive procedure gives a way to determine the positions which are P-positions and N-positions, starting from the terminal ones :

  Step 1. Label every terminal position as a P-position.

  Step 2. Label every position that can be reached as a labelled P-position **in one move** as a N-position.

  Step 3. Find positions whose **only** moves are to labelled N-positions, and label them as P-positions.

  Step 4. If no new P-position were found, the procedure is finished. Otherwise, go back to **Step 2.**

Labeling procedure for P/N-positions

- **Theorem :**

  > Every position of an ICG is either a P-position or a N-position.

- The following inductive procedure gives a way to determine the positions which are P-positions and N-positions, starting from the terminal ones :

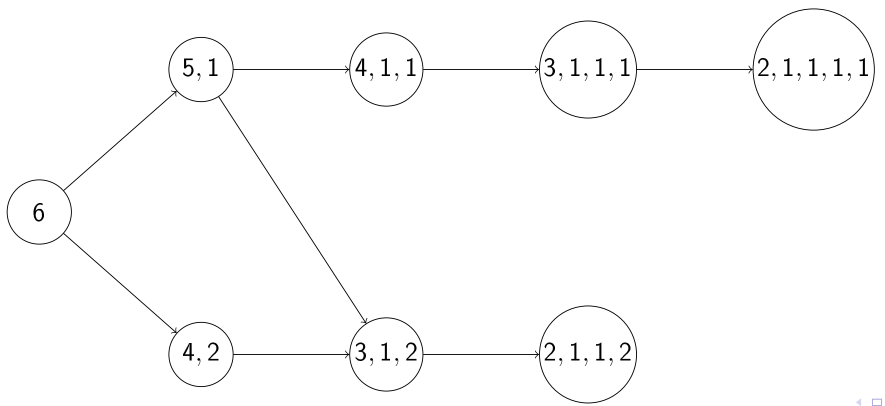
  Step 1. Label every terminal position as a P-position.

  Step 2. Label every position that can be reached as a labelled P-position **in one move** as a N-position.

  Step 3. Find positions whose **only** moves are to labelled N-positions, and label them as P-positions.

  Step 4. If no new P-position were found, the procedure is finished. Otherwise, go back to **Step 2.**

- **Exemple :**



Labeling procedure for P/N-positions

- **Theorem :**

  > Every position of an ICG is either a P-position or a N-position.

- The following inductive procedure gives a way to determine the positions which are P-positions and N-positions, starting from the terminal ones :

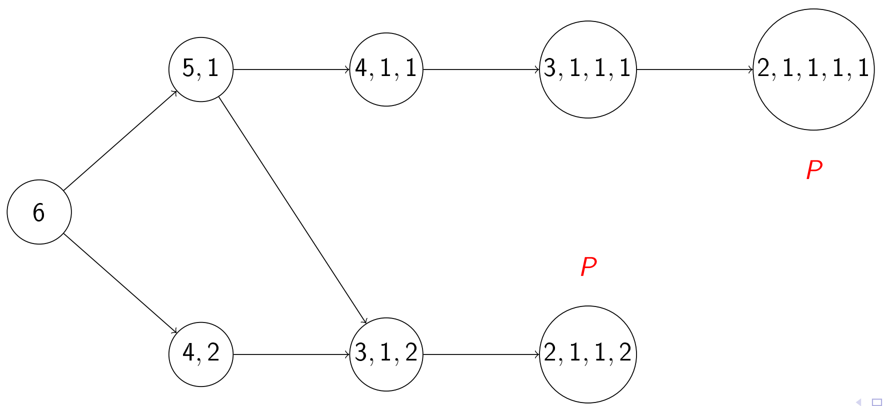
  Step 1. Label every terminal position as a P-position.

  Step 2. Label every position that can be reached as a labelled P-position **in one move** as a N-position.

  Step 3. Find positions whose **only** moves are to labelled N-positions, and label them as P-positions.

  Step 4. If no new P-position were found, the procedure is finished. Otherwise, go back to **Step 2.**

- **Exemple :**



Labeling procedure for P/N-positions

- **Theorem :**

  Every position of an ICG is either a P-position or a N-position.

- The following inductive procedure gives a way to determine the positions which are P-positions and N-positions, starting from the terminal ones :

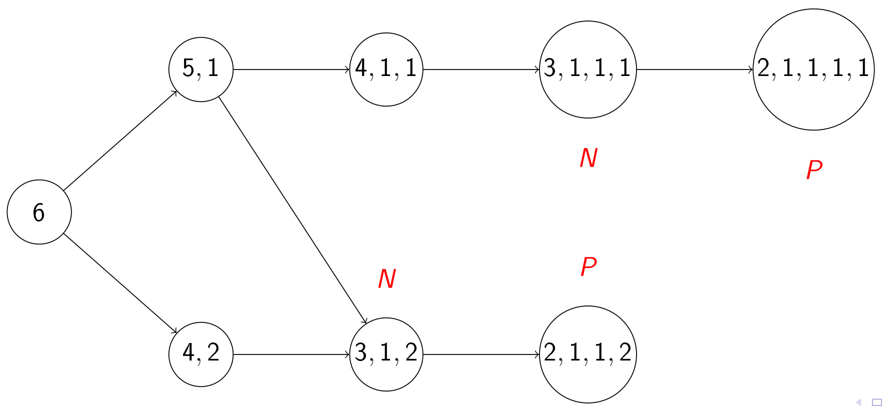
  Step 1. Label every terminal position as a P-position.

  Step 2. Label every position that can be reached as a labelled P-position **in one move** as a N-position.

  Step 3. Find positions whose **only** moves are to labelled N-positions, and label them as P-positions.

  Step 4. If no new P-position were found, the procedure is finished. Otherwise, go back to **Step 2.**

- **Exemple :**



Labeling procedure for P/N-positions

- **Theorem :**

  $\boxed{\text{Every position of an ICG is either a P-position or a N-position.}}$

- The following inductive procedure gives a way to determine the positions which are P-positions and N-positions, starting from the terminal ones :

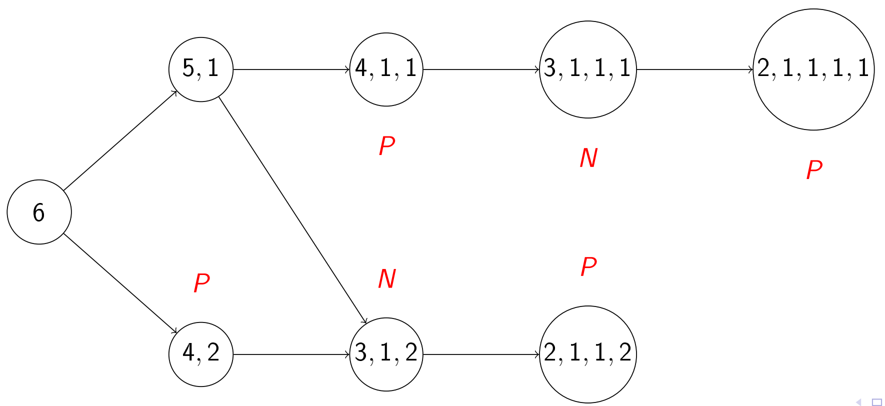
  Step 1. Label every terminal position as a P-position.

  Step 2. Label every position that can be reached as a labelled P-position **in one move** as a N-position.

  Step 3. Find positions whose **only** moves are to labelled N-positions, and label them as P-positions.

  Step 4. If no new P-position were found, the procedure is finished. Otherwise, go back to **Step 2.**

- **Exemple :**



Labeling procedure for P/N-positions

- **Theorem :**

  > Every position of an ICG is either a P-position or a N-position.

- The following inductive procedure gives a way to determine the positions which are P-positions and N-positions, starting from the terminal ones :

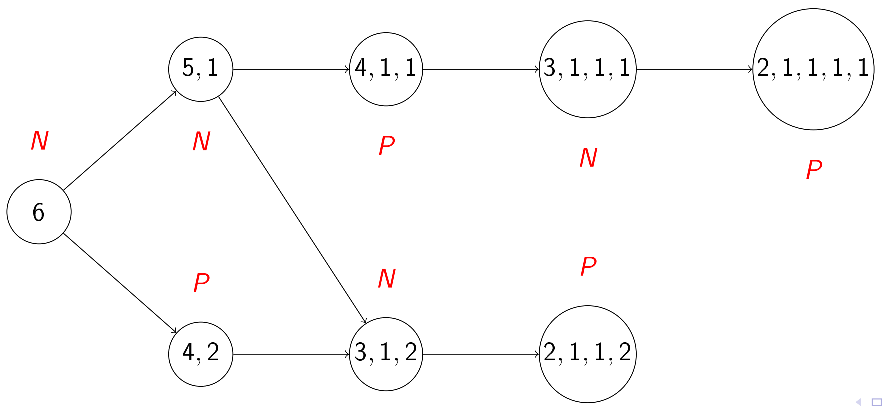
  Step 1. Label every terminal position as a P-position.

  Step 2. Label every position that can be reached as a labelled P-position **in one move** as a N-position.

  Step 3. Find positions whose **only** moves are to labelled N-positions, and label them as P-positions.

  Step 4. If no new P-position were found, the procedure is finished. Otherwise, go back to **Step 2.**

- **Exemple :**



Let's go back to subtraction games

- Consider a subtraction game with removing set $S = \{1, 3, 4\}$, and number of chips n.
  - If $n = 0$, we are in a terminal position, so **P-position**.
  - If $n = 1, 3$ or 4, we can go to 0 in one move so there are **N-positions**.
  - The only move from 2 is to 1 which is N, so 2 is a **P-position**.
  - If $n = 5$ or 6, we can reach 2 (P) in move move so this is a **N-position**.
  - If $n = 7$, the only moves lead to $6, 4$ and 3 which are all N, so 7 is a **P-position**.

Let's go back to subtraction games

- Consider a subtraction game with removing set $S = \{1, 3, 4\}$, and number of chips n.
  - If $n = 0$, we are in a terminal position, so **P-position**.
  - If $n = 1, 3$ or 4, we can go to 0 in one move so there are **N-positions**.
  - The only move from 2 is to 1 which is N, so 2 is a **P-position**.
  - If $n = 5$ or 6, we can reach 2 (P) in move move so this is a **N-position**.
  - If $n = 7$, the only moves lead to $6, 4$ and 3 which are all N, so 7 is a **P-position**.
- If we continue similarly, we obtain the following labelling on positions :

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Pos	P	N	P	N	N	N	N	P	N	P	N	N	N	N	P

- **Conjecture :** P-positions are given by integers n such that $n = 0 \pmod 7$ or $n = 2 \pmod 7$.
- **Question :** Who would win if we start with $n = 100$ chips ?

$$100 = 2 \pmod 7 \rightsquigarrow \textbf{P-position} \rightsquigarrow \text{Player 2 wins !}$$

Nim game with 2 heaps

- Let us try to identify the P-positions and N-positions for the game of Nim with two heaps.
  - $(0,0)$ is a **P-position**,
  - $(1,0)$ and $(0,1)$ are **N-positions** since they go to $(0,0)$ (P).
  - Similarly, $(2,0)$, $(0,2)$ and even $(x,0)$ and $(0,x)$ for any $x \in \mathbb{N}$ are **N-positions** !
  - $(1,1) \to (1,0),(0,1)$ which are N-positions, so it is a **P-position**.
  - $(2,1) \to (1,1)$ (P), so it is a **N-position**.
- **Conjecture** P-positions are exactly the pairs (x,x) for $x \in \mathbb{N}$!
- If you start with two heaps of different sizes, you can always win the game by coming back to a configuration where the two heaps have the same number of chips.

Nim game with 3 heaps

- Let us now do the same for the case of Nim with 3 heaps.
- When a heap is empty, the situation $(x, y, 0)$ is exactly like the case of a 2-heap Nim.
  - $(0, 0, 0)$ is a **P-position**,
  - $(1, 1, 0)$, $(x, x, 0)$, $(0, x, x)$ or $(x, 0, x)$ are **P-positions** for any integer x.
  - $(1, 1, x)$ is a **N-position** for any x since it goes to $(1, 1, 0)$.
  - $(1, 2, 2)$ is a **N-position** since it goes to $(0, 2, 2)$ (P).
  - $(1, 2, 3) \rightarrow (0, 2, 3), (1, 1, 3), (1, 0, 3), (1, 2, 2), (1, 2, 1), (1, 2, 0)$ which are all N-positions, so it is a **P-position**.
- The next simplest P-positions to find are $(1, 4, 5)$ and $(2, 4, 6)$.
- Question: What is the common point between all P-positions ??

$$(0, 0, 0), \quad (x, x, 0), \quad (1, 2, 3), \quad (2, 4, 6), \quad (1, 4, 5).$$

Nim game with 3 heaps

- Let us now do the same for the case of Nim with 3 heaps.
- When a heap is empty, the situation $(x, y, 0)$ is exactly like the case of a 2-heap Nim.
  - $(0, 0, 0)$ is a **P-position**,
  - $(1, 1, 0)$, $(x, x, 0)$, $(0, x, x)$ or $(x, 0, x)$ are **P-positions** for any integer x.
  - $(1, 1, x)$ is a **N-position** for any x since it goes to $(1, 1, 0)$.
  - $(1, 2, 2)$ is a **N-position** since it goes to $(0, 2, 2)$ (P).
  - $(1, 2, 3) \to (0, 2, 3), (1, 1, 3), (1, 0, 3), (1, 2, 2), (1, 2, 1), (1, 2, 0)$ which are all N-positions, so it is a **P-position**.
- The next simplest P-positions to find are $(1, 4, 5)$ and $(2, 4, 6)$.
- Question: What is the common point between all P-positions ??

$$(0, 0, 0), \quad (x, x, 0), \quad (1, 2, 3), \quad (2, 4, 6), \quad (1, 4, 5).$$

Difficult !

Hint: it uses binary decompositions...

Binary Representation

- Every positive integer can be written uniquely as a sum of powers of 2:

$$N = \sum_{i=0}^{k(N)} b_i 2^i, \qquad b_i \in \{0, 1\}.$$

  The coefficients b_i are called the **bits** of N.

- **Notation:** $N = (N_{k(N)} \dots N_1 N_0)_2$

- **Example:**

$$42 = 32 + 8 + 2 = 2^5 + 2^3 + 2^1 = (101010)_2.$$

Binary Representation

- Every positive integer can be written uniquely as a sum of powers of 2:

$$N = \sum_{i=0}^{k(N)} b_i 2^i, \qquad b_i \in \{0, 1\}.$$

  The coefficients b_i are called the **bits** of N.

- **Notation:** $N = (N_{k(N)} \dots N_1 N_0)_2$

- **Example:**

$$42 = 32 + 8 + 2 = 2^5 + 2^3 + 2^1 = (101010)_2.$$

- The binary representation is obtained by repeated divisions by 2:

  1. Divide the number by 2.
  2. Record the remainder (0 or 1). It gives you the **rightmost** bit of N.
  3. Replace the number N by its quotient by 2.
  4. Repeat until the quotient becomes 0.

The magical operation : bitwise XOR / Nim sum

- For two integers with their base 2 decompositions

$$N = (N_{k(N)} \dots N_1 N_0)_2 \qquad M = (M_{k(M)} \dots M_1 M_0)_2$$

we define the integer $N \oplus M$, called the **bitwise XOR** of N and M, as the integer whose binary decomposition $(x_i)_{i \in \mathbb{N}}$ is given by:

$$x_i = N_i + M_i \pmod 2$$

that is

$$\begin{cases} 0 \oplus 0 = 0 \\ 0 \oplus 1 = 1 \\ 1 \oplus 0 = 1 \\ 1 \oplus 1 = 0 \end{cases}$$

- The $\oplus$ operation is associative and commutative, so we can use it as a usual group operation addition.
  - For any integer n, $n \oplus 0 = 0 = 0 \oplus n$.
  - For any integer n, $n \oplus n = 0$.
  - For any integers n, m and p, $n \oplus p = m \oplus p \Rightarrow n = m$.

The magical operation : bitwise XOR / Nim sum

- The non-trivial P-positions were $(1, 2, 3)$, $(1, 4, 5)$ and $(2, 4, 6)$.

$$1 = (001)_2, \quad 2 = (010)_2, \quad 3 = (011)_2, \quad 4 = (100)_2, \quad 5 = (101)_2, \quad 6 = (110)_2.$$

$1 \oplus 2 \oplus 3$	$1 \oplus 4 \oplus 5$	$2 \oplus 4 \oplus 6$
001	001	010
010	100	100
011	101	110
000	000	000

- **Conjecture:** The P-positions are exactly the 3-tuples (x, y, z) such that

$$x \oplus y \oplus z = 0.$$

  This is why the bitwise XOR is also called the **Nim sum** in that case.

- This actually works for any number of heaps. Let us now discover why !

Sum of Games

- We've understood everything for a subtraction game, for instance with $S = \{1, 2\}$. But let's imagine we want now to play two such games G_1 and G_2 in parallel, with two distinct set of chips !
  - At each turn, the player has to chose if he wants to play in G_1 or G_2, and remove 1 or 2 chip(s) in that game **only**.
  - This is formally called a **sum** of games.
  - The game of Nim exactly looks like a sum of subtraction games with different subtracting sets.

Sum of Games

- We've understood everything for a subtraction game, for instance with $S = \{1, 2\}$. But let's imagine we want now to play two such games G_1 and G_2 in parallel, with two distinct set of chips !
  - At each turn, the player has to chose if he wants to play in G_1 or G_2, and remove 1 or 2 chip(s) in that game **only**.
  - This is formally called a **sum** of games.
  - The game of Nim exactly looks like a sum of subtraction games with different subtracting sets.

- The **sum** of two games G and H, denoted

$$G + H,$$

  is played according to the following rule:
  - on each turn, a player chooses **one** of the two games;
  - the player makes a legal move in the chosen game only;
  - the other game remains unchanged.

P-positions and N-positions with sums of games

- Before, we've described a function $\{\text{positions of a game } G\} \mapsto \{N, P\}$ that labels a position as a P-position or N-position.
- However, the classification into **P-positions** and **N-positions** is no longer sufficient to analyze sums of games. Indeed:
  - **P + P = P**: if we are in a losing position in both games, no matter the move we make in one of the two games, the opponent can answer in that same game.
  - **P + N = N = N + P**. Indeed, suppose you are in a sum $G + H$ of two games with G being in a N-position and H in a P-position, you can play your winning move in G and the position is now P + P, which is losing for your opponent as we've seen.
  - **N + N = ??**. It can be either P or N.
- **Exemple:** For example, consider a sum G + G where G is a subtraction game with $S = \{1, 2\}$.
  - The positions 1 and 2 are P-positions in G since they can go to 0.
  - The position $(1, 1)$ in $G + G$ is a P-position since it can only go to $(1, 0)$,$(0, 1)$ which are N.
  - The position $(1, 2)$ in $G + G$ is a N-position since it can move to $(1, 1)$ which is a P-position.

Grundy functions

- Instead of having a function that associates to a position of a game G a label P/N, we will consider a function

$$g : \{\text{positions of G}\} \to \mathbb{N}$$

that associates to each position p an integer, called the **Grundy number** of p which satisfies

$$g(p) = 0 \quad \Leftrightarrow \text{p is a P-position.}$$

Grundy functions

- Instead of having a function that associates to a position of a game G a label P/N, we will consider a function

$$g : \{\text{positions of G}\} \to \mathbb{N}$$

that associates to each position p an integer, called the **Grundy number** of p which satisfies

$$g(p) = 0 \quad \Leftrightarrow \text{p is a P-position.}$$

- The **Grundy number** of a position p of a game is

$$\begin{aligned} g(p) &= \min\{m \in \mathbb{N}_{\geq 0} \mid m \neq g(q), \quad \forall q \in H(p)\} \\ &:= \operatorname{mex}\{g(q) \mid q \in H(p)\} \end{aligned}$$

where $H(p)$ is the set of possible positions that we can reach from p in one turn, and mex stands for *minimum excluded value*.

- **Example:**
  - $\operatorname{mex}(\varnothing) = 0$.
  - $\operatorname{mex}(\{0, 1, 2, 3\}) = 4$.
  - $\operatorname{mex}(\{0, 2\}) = 1$.

Examples of Grundy numbers

- For the subtraction game with $S = \{1, 2\}$.
  - $g(0) = \text{mex}(\varnothing) = 0$.
  - $g(1) = \text{mex}(\{0\}) = 1$.
  - $g(2) = \text{mex}(\{0, 1\}) = 2$.
  - $g(3) = \text{mex}(\{1, 2\}) = 0$!
- For the game of Nim with only one heap:

$$g(x) = \text{mex}(\{0, 1, 2, \ldots, x - 1\}) = x.$$

- The Sprague-Grundy function does the same as the P/N labelling:
  - each position p that can only reach N-positions (with value $\neq 0$) gets value 0;
  - eache position p that can reach at least one P-position (with value 0) gets a non-zero value.

The Sprague-Grundy theorem

- The Grundy-number of a position p of a game G can always be computed recursively. However, for the majority of games it can take a very long time to compute !
- However, they have one advantage: they are easy to compute for simple games (e.g. Nim with one heap) and can be extended to sum of games.
- **Sprague-Grundy theorem:**

> Let $G_1, G_2, \dots, G_n$ be a set of n ICG with respective Grundy functions $(g_i)_{1 \leq i \leq n}$. Denote by G the sum of the games $G_1 + G_2 + \cdots + G_n$. Then, the Grundy function of the game G is given by:
>
> $$g(p_1, p_2, \dots, p_n) = g_1(p_1) \oplus g_2(p_2) \oplus \cdots \oplus g_n(p_n)$$
>
> where $\oplus$ denotes the **bitwise XOR** !

Proof of the Sprague-Grundy theorem

- According to the book *A course in Game Theory* by Ferguson.

- Let us focus on two games G_1, G_2 with respective Grundy functions g_1 and g_2. We want to prove that

$$g(p_1, p_2) = g_1(p_1) \oplus g_2(p_2).$$

- Denote by $a := g_1(p_1)$, $b := g_2(p_2)$ and $k := a \oplus b$. We then need to prove that $k = \mathrm{mex}\{g(z) \mid z \in H(p_1, p_2)\}$.

(Step 1) For every $h < k$, there is a position (p_1', p_2') that can be reached from (p_1, p_2) in one turn, with $g_1(p_1') \oplus g_2(p_2') = h$.

(Step 2) There is no position (p_1', p_2') that can be reached from (p_1, p_2) with $g_1(p_1') \oplus g_2(p_2') = k$.

Proof of the Sprague-Grundy theorem - Step 1

- Take an arbitrary value $h < k = g_1(p_1) \oplus g_2(p_2)$.
- Consider $\ell = k \oplus h$, and the binary decompositions $k = (k_i)_2$ and $h = (h_j)_2$. Denote by x the largest power of 2 that appears in ℓ, e.g. the number of bits of ℓ minus one.
  - For $i > x$, we have $k_i = h_i$, since $k_i \oplus h_i = 0$.
  - Since $k > h$, we know that $k_x = 1$ and $h_x = 0$.
  - Since $k_x = 1$ and $k = g_1(p_1) \oplus g_2(p_2)$, we know that either $g_1(p_1)_x = 1$ or $g_2(p_2)_x = 1$. We can assume without loss of generality that we are in the first case.
- Now, since $\ell_x = 1 = g_1(p_1)_x$ and $\ell_j = 0$ for $j > x$, we deduce that

$$\ell \oplus g_1(p_1) < g_1(x).$$

  By definition of the mex operator, we know that there is a position p_1' reachable from p_1 such that $g_1(p_1') = \ell \oplus g_1(p_1)$.
- Then consider the move $(p_1, p_2) \to (p_1', p_2)$. We have

$$g_1(p_1') \oplus g_2(p_2) = \ell \oplus g_1(p_1) \oplus g_2(p_2) = \ell \oplus k = k \oplus h \oplus k = h.$$

Proof of the Sprague-Grundy theorem - Step 2

- Let us assume that we can reach in one turn a position (p_1', p_2') such that

$$g_1(p_1') \oplus g_2(p_2') = k = g_1(p_1) \oplus g_2(p_2).$$

- Since it is reached in one move, either played in G_1 or G_2, we know that either $p_1 \neq p_1'$ or $p_2 \neq p_2'$. Without loss of generality, let us assume that $p_1' \neq p_1$ and $p_2' = p_2$.

- This means that

$$\begin{aligned} g_1(p_1') \oplus g_2(p_2') &= g_1(p_1') \oplus g_2(p_2) \text{ on the one hand} \\ &= g_1(p_1) \oplus g_2(p_2) \text{ on the other hand} \end{aligned}$$

  which implies that $g_1(p_1') = g_1(p_1)$ since the bitwise XOR is a group operation.

- This is impossible since, by definition of the mex operator, if p_1' is reachable from p_1, then $g(p_1) \neq g(p_1')$.

Optimal strategy of play

- The Grundy function of a game allows to describe the optimal strategy of play following this principle: if a game G is in a position p,
  - p is a P-position if and only if $g(p) = 0$, and in that case there is no optimal move for the next player;
  - p is a N-position if and only if $g(p) \neq 0$, and in that case the optimal move is to go to a P-position, that is a position p' with $g(p') = 0$. **There is an algorithm to determine the position p'.**

Input: $G = G_1 + \cdots + G_m$ is a sum of games, at position p

Step 1: Consider a subgame G_m in state p_m with

$$g_m(p_m) := x,$$

such that

$$x_{k(s)} = 1 \text{ with } s = g(p) \text{ and } k(s) = \text{largest power of 2 in } s.$$

Step 2: Define $\hat{x}$ by

$$\hat{x}_i = \begin{cases} x_i, & \text{if } i > k(s), \\ s \oplus x_i, & \text{if } i \leq k(s). \end{cases}$$

Step 3: Move the subgame G_m to a state $\hat{p}_m$ such that

$$g_m(\hat{p}_m) = \hat{x}.$$

Optimal strategy of play - Proof

- **Proposition:**

  For a game G in a position p such that $g(p) \neq 0$, the previous optimal move algorithm indeed gives an **optimal move**, that is a move to a position p' such that $g(p') = 0$.

- **Proof:** Let us show that the specified move is possible and leads to a P-position.

- First, there exists a subgame G_m such that $g_m(p_m) = x$ and $x_{k(s)} = 1$. Otherwise,

$$1 = s_{k(s)} = g_1(p_1)_{k(s)} \oplus g_2(p_2)_{k(s)} \oplus \cdots \oplus g_n(p_n)_{k(s)} = 0 \oplus 0 \oplus \cdots \oplus 0 = 0.$$

- Then, a move into the game G_m from p_m to $\hat{p}_m$ is feasible. In binary, we have $\hat{x}_i = x_i$ for $i > k(s)$ and $x_{k(s)} = 1$ whereas $\hat{x}_{k(s)} = 0$.
  This means that $x > \hat{x}$ and thus by definition of the mex, a move from p_m with $g_m(p_m) = x$ to $\hat{p}_m$ with $g(\hat{p}_m) = \hat{x}$ is doable.

Optimal strategy of play - Proof - II

- Finally, prove that the move is optimal. Consider $p' = (p_1, \dots, p_{m-1}, \hat{p}_m, p_{m+1}, \dots, p_n)$ the new position of the game G. We have

$$\begin{aligned} s' &:= g(p') = g_1(p_1) \oplus \dots \hat{x} \oplus \dots g_n(p_n) \\ &= g_1(p_1) \oplus \dots x \oplus \dots \oplus g_n(p_n) \oplus x \oplus \hat{x} \\ &= g(p) \oplus x \oplus \hat{x} \end{aligned}$$

- This means that

$$\begin{aligned} s'_i &= s_i \oplus \hat{x}_i \oplus x_i \\ &= \begin{cases} s_i \oplus x_i \oplus x_i, & \text{if } i > h(s), \\ s_i \oplus x_i \oplus s_i \oplus x_i, & \text{if } i \leq h(s), \end{cases} \\ &= \begin{cases} s_i, & \text{if } i > h(s), \\ 0, & \text{if } i \leq h(s), \end{cases} \\ &= 0. \end{aligned}$$

So $g(p') = 0$ and it gives an optimal move !

Optimal move in Nim

- Let us consider a Nim game with 3 heaps in the position $(11, 7, 5)$.

$$\begin{aligned} 11 &= 1011 \\ 7 &= 0111 \\ 5 &= 0101 \\ \hline \bigoplus &= 1001 \end{aligned}$$

so $s = (1001)_2$, $k(s) = 3$ and we have to play in the game with 11 chips.

- To make the XOR be 0, change the heap 11 into $11 \oplus s = (0010)_2$.
The optimal move is thus to go from $(11, 7, 5)$ to $(2, 7, 5)$!

? Exercise (2).

1. Calculate the best move for the game of Nim with 3 heaps and in the position $(27, 18, 4)$.
2. Consider a Nim game with a position being given by a list of integers, e.g. `[11,7,5]` for the position $(11, 7, 5)$. Write a Python function that finds the best move to play !

- In Python, there is an operator to compute the bitwise XOR of two integers: $\wedge$.

Grundy numbers for Kayle's game

- **Remainder:** consider a row of n bowling pins, denoted by K_n. A move consists in removing either one pin or two adjacent pins.
- **Objective:** Compute $g(K_n)$ for any integer n.
  - $g(K_0) = 0$;
  - For K_1, the only move possible is $1 \to 0$ so $g(K_1) = \mathrm{mex}(\{0\}) = 1$.

Grundy numbers for Kayle's game

- **Remainder:** consider a row of n bowling pins, denoted by K_n. A move consists in removing either one pin or two adjacent pins.
- **Objective:** Compute $g(K_n)$ for any integer n.
  - $g(K_0) = 0$;
  - For K_1, the only move possible is $1 \to 0$ so $g(K_1) = \mathrm{mex}(\{0\}) = 1$.
  - For K_2,

$$\begin{tikzcd} & 11 \arrow[dl] \arrow[d] \arrow[dr] & \\ 01 & 10 & 00 \end{tikzcd}$$

Grundy numbers for Kayle's game

- **Remainder:** consider a row of n bowling pins, denoted by K_n. A move consists in removing either one pin or two adjacent pins.

- **Objective:** Compute $g(K_n)$ for any integer n.
  - $g(K_0) = 0$;
  - For K_1, the only move possible is $1 \to 0$ so $g(K_1) = \mathrm{mex}(\{0\}) = 1$.
  - For K_2,

$$
\begin{tikzcd}
 & 11 \arrow[dl] \arrow[d] \arrow[dr] & \\
01 \arrow[d, squiggly] & 10 \arrow[d, squiggly] & 00 \arrow[d, squiggly] \\
1 & 1 & 0
\end{tikzcd}
$$

Grundy numbers for Kayle's game

- **Remainder:** consider a row of n bowling pins, denoted by K_n. A move consists in removing either one pin or two adjacent pins.

- **Objective:** Compute $g(K_n)$ for any integer n.
  - $g(K_0) = 0$;
  - For K_1, the only move possible is $1 \to 0$ so $g(K_1) = \mathrm{mex}(\{0\}) = 1$.
  - For K_2,

$$
\begin{tikzcd}
& 11 \arrow[dl] \arrow[d] \arrow[dr] & \\
01 \arrow[d, squiggly] & 10 \arrow[d, squiggly] & 00 \arrow[d, squiggly] \\
1 & 1 & 0
\end{tikzcd}
$$

  - For K_3,

$$
\begin{tikzcd}
& & 111 \arrow[dll] \arrow[dl] \arrow[d] \arrow[dr] \arrow[drr] & & \\
011 & 110 & 001 & 100 & 101
\end{tikzcd}
$$

Grundy numbers for Kayle's game

- **Remainder:** consider a row of n bowling pins, denoted by K_n. A move consists in removing either one pin or two adjacent pins.
- **Objective:** Compute $g(K_n)$ for any integer n.
  - $g(K_0) = 0$;
  - For K_1, the only move possible is $1 \to 0$ so $g(K_1) = \mathrm{mex}(\{0\}) = 1$.
  - For K_2,

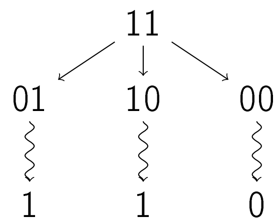
$$\begin{tikzcd}
& 11 \arrow[dl] \arrow[d] \arrow[dr] & \\
01 \arrow[d, squiggly] & 10 \arrow[d, squiggly] & 00 \arrow[d, squiggly] \\
1 & 1 & 0
\end{tikzcd}$$

  - For K_3,

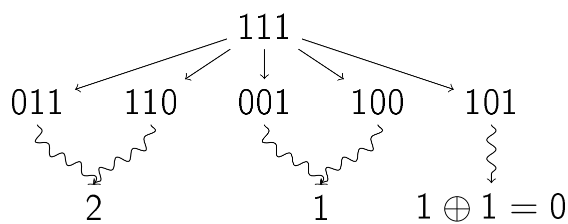
$$\begin{tikzcd}
& & & & 111 \arrow[dllll] \arrow[dll] \arrow[d] \arrow[drr] \arrow[drrrr] & & & & \\
011 \arrow[dr, squiggly] & & 110 \arrow[dl, squiggly] & & 001 \arrow[dr, squiggly] & & 100 \arrow[dl, squiggly] & & 101 \\
& 2 & & & & 1 & & &
\end{tikzcd}$$

Grundy numbers for Kayle's game

- **Remainder:** consider a row of n bowling pins, denoted by K_n. A move consists in removing either one pin or two adjacent pins.
- **Objective:** Compute $g(K_n)$ for any integer n.
  - $g(K_0) = 0$;
  - For K_1, the only move possible is $1 \to 0$ so $g(K_1) = \mathrm{mex}(\{0\}) = 1$.
  - For K_2,



  - For K_3,



Grundy numbers for Kayle's game - II

? Exercise (3 & 4).

1. Calculate the Grundy number $g(K_4)$ and $g(K_5)$.
2. (**Difficult.**) Write down a **recursive** Python function to compute the Grundy number $g(K_n)$ for any integer n. One has first to enumerate the list of all possible moves from a given position.

Grundy numbers for Kayle's game - II

? Exercise (3 & 4).

1. Calculate the Grundy number $g(K_4)$ and $g(K_5)$.
2. (**Difficult.**) Write down a **recursive** Python function to compute the Grundy number $g(K_n)$ for any integer n. One has first to enumerate the list of all possible moves from a given position.

- For K_4,

$$\begin{tikzcd}[column sep=small] & & 1111 \arrow[dll] \arrow[dl] \arrow[d] \arrow[dr] & \\ 0111, 1110 & 0011, 1100 & 1011, 1101 & 1001 \end{tikzcd}$$

$$g(K_4) = \mathrm{mex}(\{3, 2, 0\}) = 1!$$

Grundy numbers for Kayle's game - II

Exercise (3 & 4).

1. Calculate the Grundy number $g(K_4)$ and $g(K_5)$.
2. (**Difficult.**) Write down a **recursive** Python function to compute the Grundy number $g(K_n)$ for any integer n. One has first to enumerate the list of all possible moves from a given position.

- For K_4,

$$\begin{tikzcd}[column sep=small]
 & 1111 \arrow[dl] \arrow[d] \arrow[dr] \arrow[drr] & & \\
0111, 1110 \arrow[d, rightsquigarrow] & 0011, 1100 \arrow[d, rightsquigarrow] & 1011, 1101 \arrow[d, rightsquigarrow] & 1001 \arrow[d, rightsquigarrow] \\
3 & 2 & 1 \oplus 2 = 3 & 1 \oplus 1 = 0
\end{tikzcd}$$

$$g(K_4) = \mathrm{mex}(\{3, 2, 0\}) = 1!$$

Grundy numbers for Kayle's game - II

- Here are the first 20 values of the Grundy numbers $K(n)$ for $1 \leq n \leq 20$.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$g(K_n)$	1	2	3	1	4	3	2	3	1	4	2	6	4	1	2	7	1	3	2	4

? Exercise (4).

You're playing a Kayle's game with 13 pins. Of the 13 bowling pins in the row, the second has been knocked down.

1. Prove that this is an N-position. You may use the tabular above.
2. Find a winning move: which pin(s) should be knocked down ?