Group B
Mario Poire, Peter Alvarez, Bernardo Garza, Lionel Hernandez

Notes for Monday 2/6/17

What is a game?

We want our definition to capture:

- Board games (Chess, Checkers, Go, etc.)
- Card games (Poker, etc.)
- One-player games (Puzzles)
- Zero-player games (Automata, Computation)
    - Running a program is like an unfun zero-player game with the computer (Conway's Game of Life http://www.bitstorm.org/gameoflife/)

Four main features of games we are covering are:

- Positions
    - Finite board configurations
    - Finite amount of information
    - Bounded state (does not mean that the game is bounded, chess can repeat)
- Players
    - Players take turns (No Hungry Hungry Hippos)
    - Players have clear goals
    - Players have a clear list of allowable moves and must pick one
- Moves - Takes one game position (state) to another position
- Goals - Game positions with some property (may be different for each player)

We assume optimal play, also called rational play - The assumption that players play to win or to do the best possible.

Combinatorial Games

*Notable books on combinatorial games: *Lessons in Play*, *Winning Ways for your Mathematical Plays*, *On Numbers and Games*

We assume 2 players and perfect information (nothing hidden, no randomness).

Algorithmic Combinatorial Game Theory - finding strategies in large games

Surreal Numbers - all combinatorial games can be described by surreal numbers (Knuth - *Surreal Numbers*)

Definition: A combinatorial game is a 2-player game played between Louise (L/Left) and Richard (R/Right). The game consists of the following:

- A set of possible positions (The states of the game)
- A move rule indicating for each position what positions Louise can move to and what positions Richard can move to.
- A win rule indicating a set of terminal positions where the game ends. Each terminal position has an associated outcome.
    - -+ Richard wins
    - +- Louise wins
    - 00 Draw

To play: Choose a starting position and which player goes first. Then, take turns until a terminal position is reached.

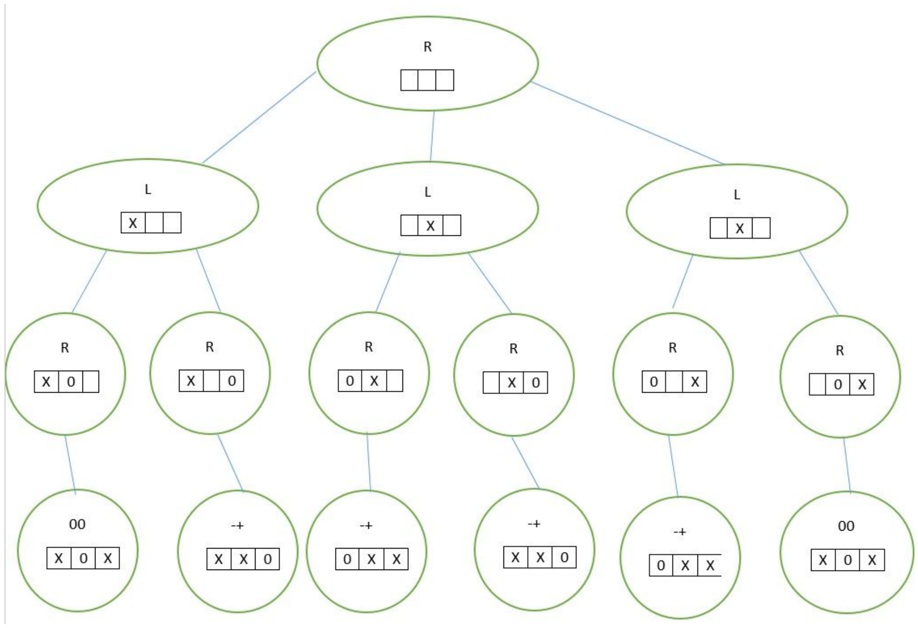
Normal Play Game - a game where there is a win rule that the last player to make a move wins, i.e., the first player that can't move loses.

Examples of normal play games:

- Pick Up Bricks - A player can pick up 1 or 2 bricks. The last player to pick up a brick wins.
- Domineering - Played on an arbitrary board of squares. One player plays horizontal dominoes, the other plays vertical ones. Last player to place a domino wins.

Game Trees

Each branch models a choice for a player and terminal nodes indicate an outcome.
Example: Tic



To build a game tree:

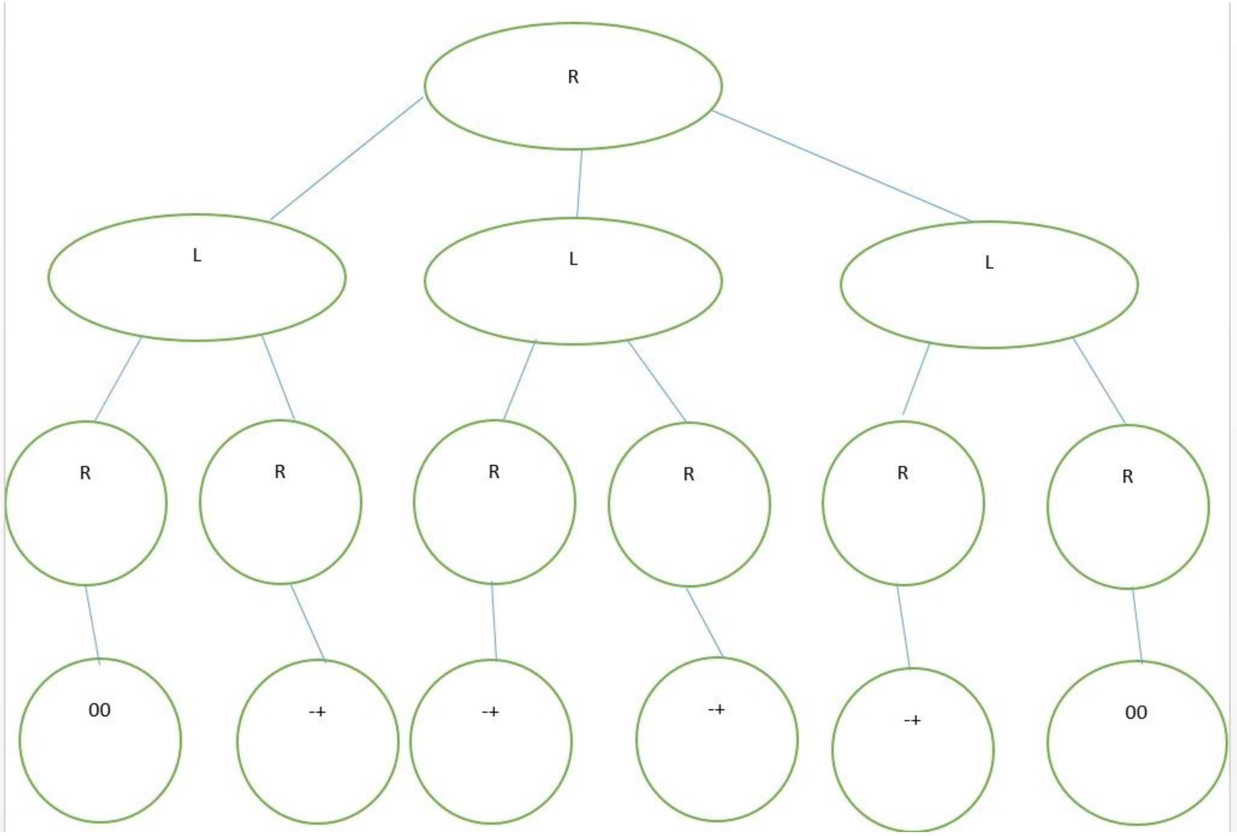
- Start at position α (alpha) with player L moving first.
- The root node has L and α.
- If L can move to $\alpha_1 \dots \alpha_k$ then join k new nodes, each containing one of $\alpha_1 \dots \alpha_k$ and R if nonterminal. If terminal, put the outcome instead (-+, +-, 00)

WLD Trees

- Win-lose-Draw tree
- Makes positional information superflows
- Really only need whose turn and outcome
- unified way to think combinatorial games
    - Trivial Trees - single nodes that are either -+, +-, or 00.
    - Trees can be unbounded in depth (like chess) but for now, limit ourselves to bounded games with bounded depth trees.

Make a strategy (WLD Tree)

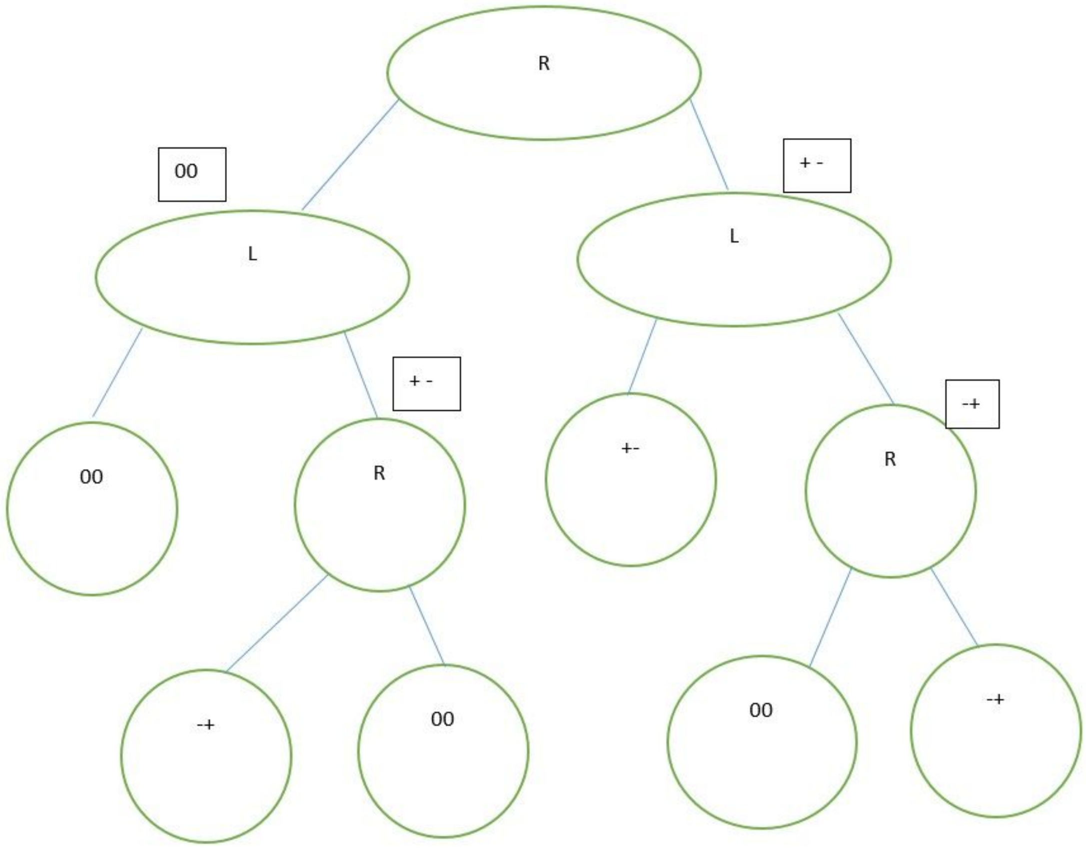
- Formulate a plan to win.
- A set of decisions indicating which move to make at each node where the player has a choice.
- Winning Strategy - guarantees you will win
- Drawing Strategy - guarantees you don't lose



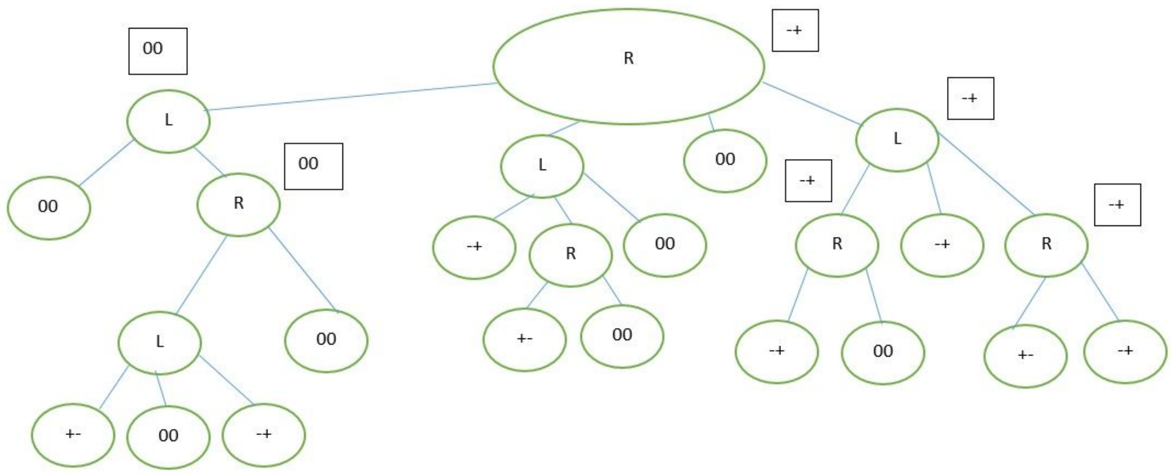
*If Richard always picks the right (directional) branch, it gives L a chance to pick a winning move - it is a bad strategy

Working Backwards

- If we could consider the entire tree at once, it may be difficult to choose the best strategy.



- Procedure: A player has a decision to make at node N. We've already determined the outcome under rational play for all possible nodes from node N. Choose the best possible outcome for this player. Indicate the choice in the tree and mark N. Continue until the root is marked.



Notes for Wednesday 2/8/17

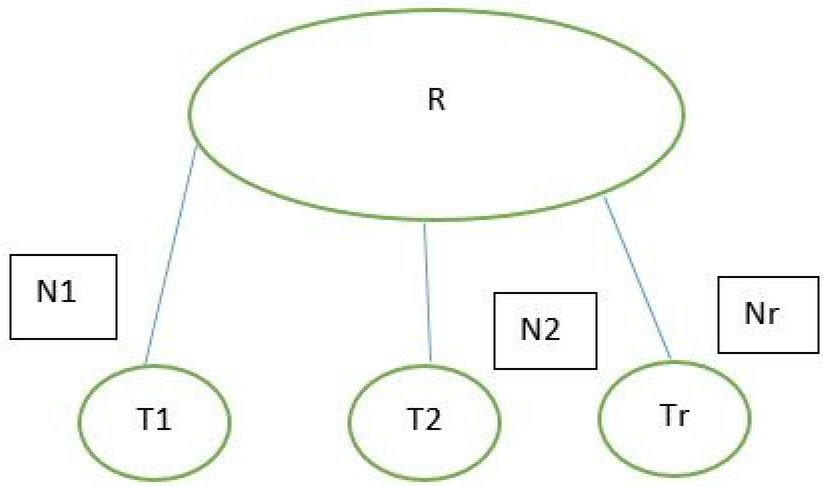
<u>Zermelo's Theorem</u>

Every WLD game tree is

Type	Description
+-	L has a winning strategy
-+	R has a winning strategy
00	Both players have a winning strategy

Proof sketch:

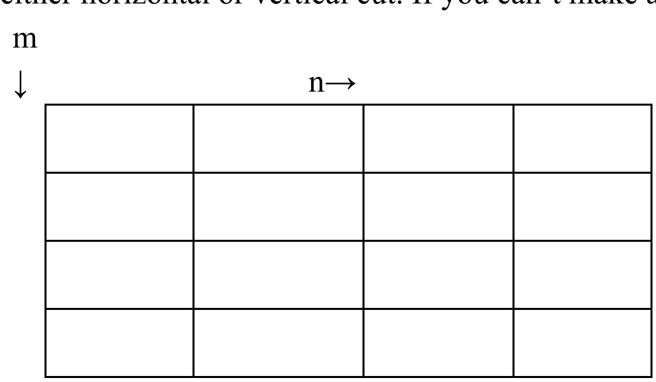
- Base case: +- -+ 00
- Induction



- Induction step:
  - If for any $N_1 \dots N_L$, there exists -+
    - Pick it (R WIN)
  - If all $N_1 \dots N_L$ are +-
    - Doesn't matter -> +- (L WIN)
  - If there does not exist -+ nodes $N_1 \dots N_L$
    - All nodes are either +- or 00
    - Pick 00 (R DRAW)

Strategy

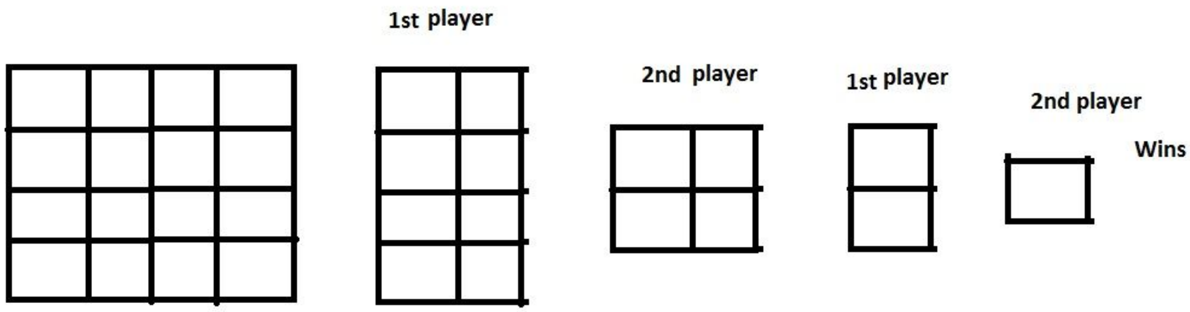
Chop - Players make either horizontal or vertical cut. If you can't make a move, you lose.



Symmetry

Proposition: Consider an mxn position of chop

1. If m = n, second player has a winning strategy
2. If m ≠ n, first player has a winning strategy

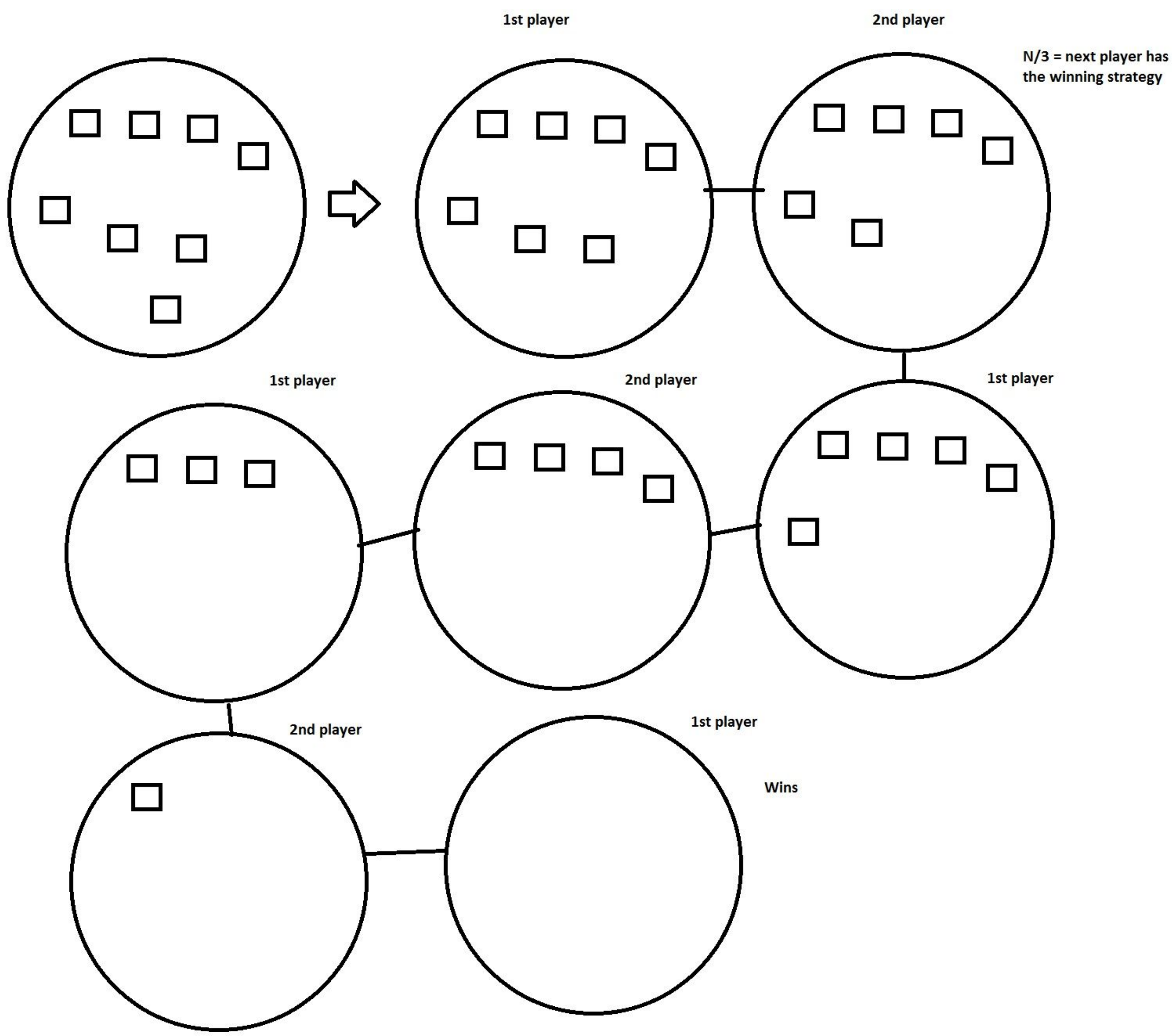


The game must end at a 1x1 square. One player is always making squares, the other is always making rectangles

Pick up bricks

Proposition: Consider a pickup bricks position of n bricks

1) If 3 divides N, the 2nd player has the winning strategy
2) Otherwise, the 1st player has the winning strategy



Strategy Stealing

- Prove existence
- Proof by contradiction

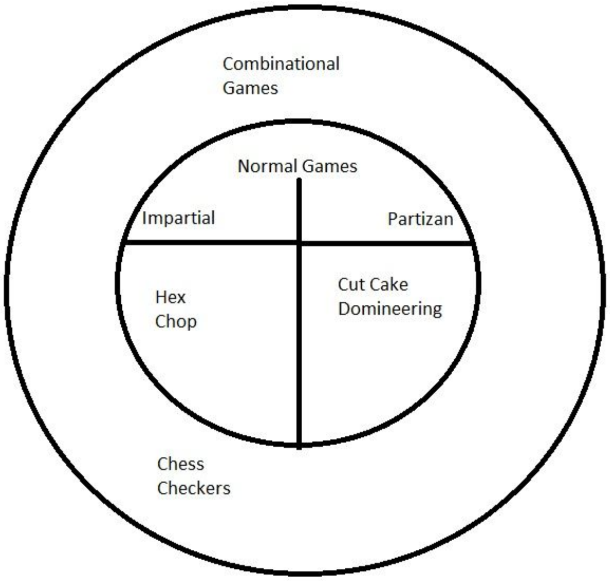
Proposition: First player has the winning strategy in Hex (starting from an empty board)

- Assume 2nd player has the winning strategy

N= { b_1...b_n}

Proof sketch

Pick random b_i and place piece. Then play as the second player. If the strategy says play b_i, pick a random b_j and play i+j

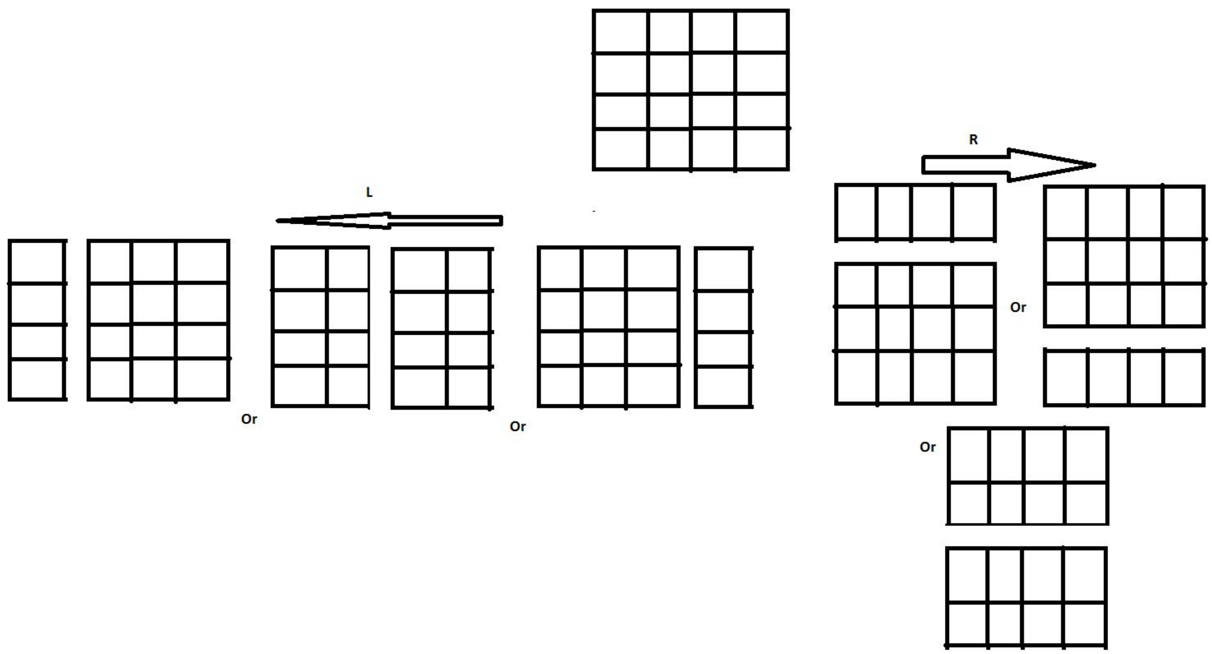


Impartial - Same moves available to both players

Partizan - Not impartial

Cut Cake

- L only makes vertical cuts
- R can only make horizontal cuts



Position

$A = \{a_1,\ldots,a_m \mid B_1,\ldots,B_n\}$

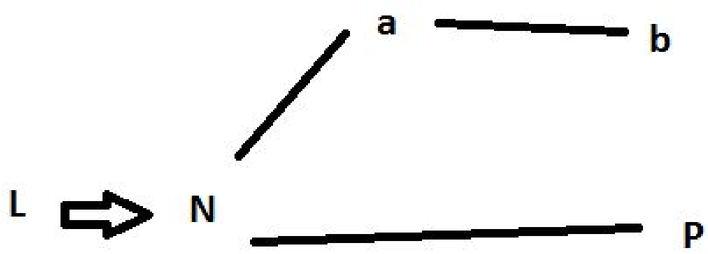
Types of positions

- Based on Zermelo's Theorem
- Cor. every position in a normal play game is one of these types

Type	Description
L	L has a winning strategy, whoever goes first
R	R has a winning strategy, whoever goes first
N	Next player to move has a winning strategy
P	Previous player has a winning strategy

Determining Type

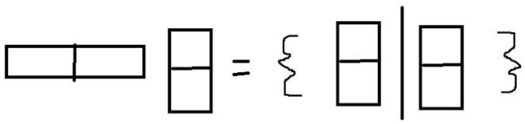
- Suppose L moves from some position and has >= 1 type B to move to.
- One of them is type Por L



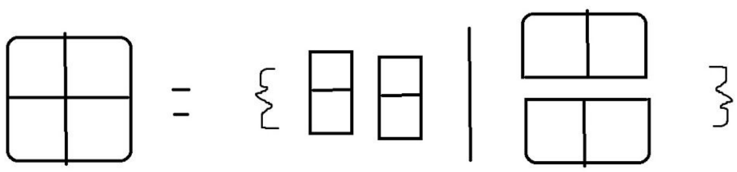
Proposition: If $A = \{a_1,...,a_m \mid B_1,...,B_n\}$, the type of A

	Some B_j is type R or P	All of $B_j...B_n$ are types of L or N	
Some a_i is type L or P	N	L	
all a are R or N	R	P	

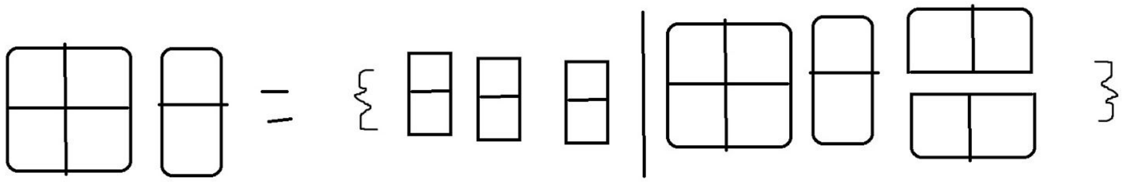
Simple Cut Cake Positions



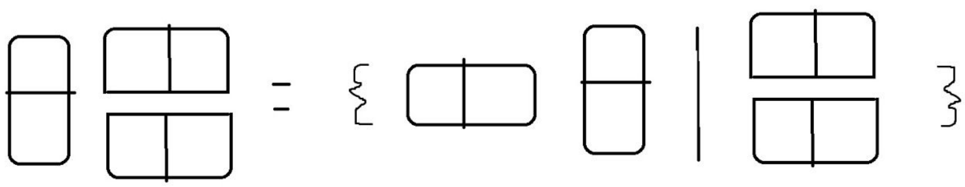
P



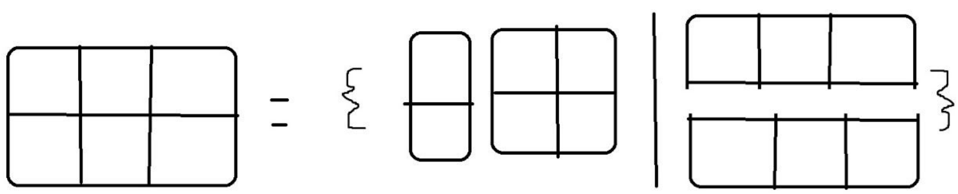
P



R



L



P