Computing Science (CMPUT) 657

Algorithms for Combinatorial Games

Martin Müller

Department of Computing Science
University of Alberta
mmueller@ualberta.ca

Winter 2022

Part II

Introduction to Combinatorial Games

Topics

- Introduction to combinatorial games and how to solve them
- Try to take it easy on the mathematical theory
- Focus on computing side - algorithms and solving
- More sample games

Combinatorial Games

Two-player games

- Core of the theory is for games which are:
- Complete information
- Turn-based
- Finite
- Break down into subgames
- *Normal play* convention:
  last player that can make a move wins the game
- Many extensions to other game types exist

Combinatorial Game Theory (CGT)

- Developed by Conway, Berlekamp, Guy, $\ldots$
- Abstract definition of two-player games
- Game position defined by its *options*:
  sets of follow-up positions for both players
- Main application: games that are a *sum* of independent subgames
- Lots of interesting and unusual mathematics

Combinatorial Game Algorithms

- How to solve combinatorial games efficiently by using computation?
- Basic concepts shared with CGT
- Main applications:
  - Solve games that are a *sum* of independent subgames
  - Play well when we cannot solve:
    - Sum of too many games
    - Sum of too-complicated subgames
- Lots of interesting algorithms
- Not a lot of research yet

Some Highlights of Combinatorial Game Algorithms Research at Alberta

- World's top Hex-playing programs (Hayward group)
- Solving Hex up to 9x9 size board (Hayward group)
- Solving 5x5 and 5x6 Amazons (Müller; Jiaxing Song MSc thesis)
- Solving 10x10 Domineering (Nathan Bullock MSc thesis)
- Decomposition Search and solving Go endgame puzzles (Müller)
- Search algorithms for playing sums of hot games (Zhichao Li)

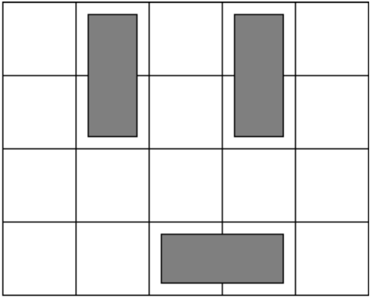
More Highlights from Alberta

- First implementation of generalized thermography
- Developing Temperature Discovery Search (TDS) and TDS+ (Yeqin Zhang MSc thesis)
  - First general forward search algorithms for temperature of complex (sub)games
- **Recent:** CGTSolver - Solver for Linear NoGo based on Combinatorial Game Ideas (Henry Du)
- **Recent:** MCGS - Minimax-based Combinatorial Game Solver (Taylor Folkersen, Henry Du)
- **Recent:** Multiple course projects for 2017, 2022 Combinatorial Game algorithm courses - will review later in class
- **Recent:** SEGClobber - A Linear Clobber Solver. (Taylor Folkersen, Zahra Bashir, Fatemeh Tavakoli)

Examples of Combinatorial Games

Example game: Domineering

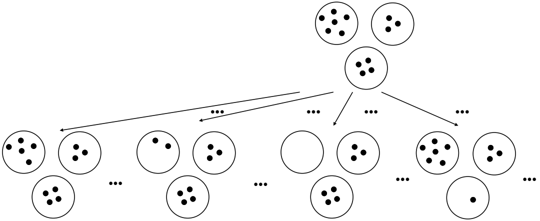
- Two players put dominoes on a grid
- One places dominoes vertically, the other horizontally
- Last player wins
- Let's play a 6×6 game



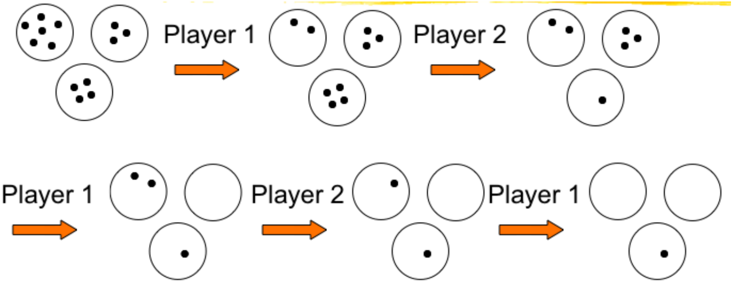
Example: 4×5 Domineering

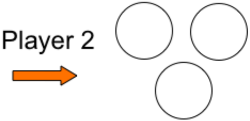
Example game: Nim

- Several heaps of tokens
- Move: take *any* number of tokens from a *single* heap
- Win: remove last token



Nim Sample Game: 3 Heaps

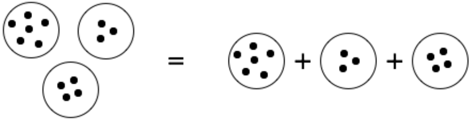




- Player 2 wins!
- Q: How to play well?

Nim as a Sum Game

- Each move changes exactly one heap
- All other heaps unchanged
- Moves in one heap independent of other heaps
- Each heap is a subgame
- Overall game is sum of all heaps



Comments on Nim and Impartial Games

- Nim was solved in 1908 by Bouton
- Nim is an example of an *impartial* game: both players have exactly the same moves
- There is a complete (and much older) theory of impartial games, starting in the 1930's (Sprague, Grundy)
- We will discuss it soon
- Current CGT is much more general and powerful
  - Partizan games: Both players may have different moves
  - Example: move with pieces of different color

Comparing Classical vs Combinatorial Games

Classical Board Game

- Single, monolithic game state
- Full board evaluation
- Single game tree, minimax backup
- Central question: what is the minimax score?

Combinatorial

- Partition game into sum of subgames
- Local analysis
- Combination of local results
- Central question: which sums of games are wins?

Resources for Combinatorial Games

CGT Resources

- Also see our resources web page
- Undergrad CGT textbook: Lessons in Play by Albert, Nowakowski and Wolfe.
- Graduate level CGT textbook: Siegel, Combinatorial Game Theory (all the math precisely defined and proven)
- Aaron Siegel's Combinatorial Game Suite (CGSuite) software package for the math side of CGT
- MCGS - our minimax combinatorial game solver

CGT Classic Books

- All these focus more on the mathematical theory and game examples, not as much on algorithms
- Conway, On numbers and games, “ONAG”
  A mathematical theory where numbers are defined as special cases of games!
- Berlekamp, Conway and Guy, Winning Ways (4 volumes), “WW”
- Berlekamp and Wolfe: Mathematical Go
  Go endgame puzzles that top human players cannot solve, but mathematicians can

Software - CGSuite and MCGS

- Download CGSuite from `http://cgsuite.sourceforge.net/`
- Go through the first two built-in tutorials in the help window.
  - Tutorials/Getting Started/Worksheet Basics
  - Tutorials/Getting Started/Using the Explorer
- Download MCGS from `https://github.com/ualberta-mueller-group/MCGS/`
- Read the MCGS paper

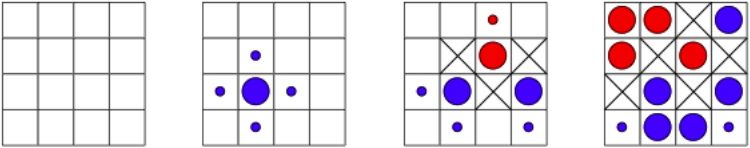
Main Ideas in Combinatorial Games

CGT - High-level Ideas

- Define sums of games - what does it mean to play two games at once?
- Use sums: split a complex game into sum of local subgames
- Mathematical theory precisely describes the *value* of a game, and the value of making a move
- Important: tradeoff between local gain and the right to play first in another subgame - which is worth more?
- Important applications in Go endgames
- We will study *search-based algorithms* related to these questions

Example Game: Snort

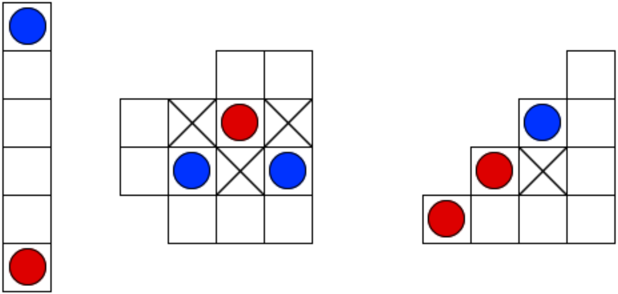
- Can play on any graph (here: a grid of squares)
- Move: color one square with your color
- Cannot color a square adjacent to opponent
- Last move wins
- Snort is a *partizan* game - move choices are different for both players



Sample Snort game:
starting position, after 1 and 3 moves, end of game

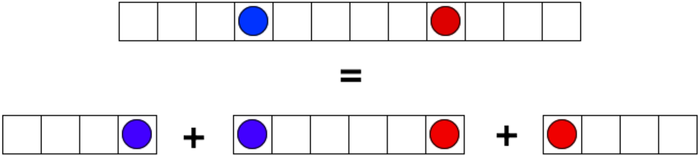
Snort as a Sum Game

- Play several subgames at the same time
- Move: make one move in one subgame
- Last player who can move *anywhere* wins



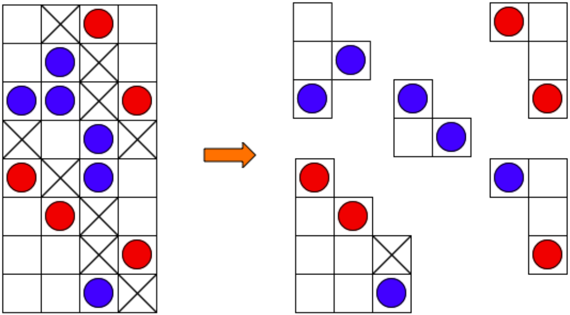
Split Game into Subgames

- Split game:
- Sum of independent subgames
- Move in one subgame does not affect others



Splitting one board into three subgames. Copy the neighboring colored squares into each subgame

Split into Subgames (2)



One 4x8 Snort board, split into sum of five subgames

Mathematical Notation for Combinatorial Games

Combinatorial Games According to Conway

- Theory invented by Berlekamp, Conway, and Guy (WW)
- Mathematical theory published by Conway (ONAG)
- Two players, **Left** and **Right**
- Game G defined by **move options** of both players
- Recursively, **each move leads to a game**
- A player who cannot move loses (normal play)
- That's all!

Combinatorial Games Definition

Recursive definition of game:

$$G = \{L_1, \cdots, L_n \mid R_1, \cdots, R_m\}$$

- $L_1, \cdots, L_n$ are the Left options
  - Each option is a game that Left can move to
  - $n = 0$ is possible - then Left has no options (no moves)
- Similarly, $R_1, \cdots, R_m$ are the Right options
  - If $m = 0$ then Right has no options (no moves)
- The vertical bar $\mid$ separates the left from the right options
- This notation looks almost like a set, but it is not: a game always consists of **two** sets

Combinatorial Games Definition (2)

Equivalent notation for writing a game:

$$G = \{G^L | G^R\}$$

- Here, G^L and G^R are *sets* of games
- $G^L = \{L_1, \cdots, L_n\}$
- $G^R = \{R_1, \cdots, R_m\}$
- CGT uses both notations - either write all options one by one, or just write them as a single set
- Choose whichever way is simplest/clearest

Whose Turn is it?

- In $G = \{G^L | G^R\}$,
  there is **no** concept of “whose turn” it is
- A game always contains the options for **both** players.
  - This will make more sense when we look at sums of games.
  - For solving games, of course we must consider whose turn it is
  - Left-to-play and Right-to-play can give very different results
- We will soon talk about “outcome classes” of games
  - Who can win, going first, or going second?
  - Four possible answers:
    Left, Right, Next player, Previous player

Finite vs Infinite Games

- $G = \{G^L | G^R\}$
- Amazingly, this one definition is enough to build a huge theory for both numbers and games.
- We focus on *finite* games in this class
- Each game ends after a finite number of moves

Numbers and Games

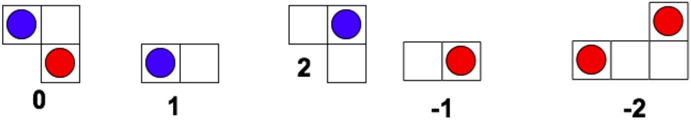
- Some finite numbers can be seen as a special case of finite games, e.g. 1, -5, 42, 123/256
- Other finite numbers cannot be represented by finite games, e.g. 1/3 can only be represented by an infinite game
- In this course, we do not discuss infinite numbers, or how to construct real numbers
  - See ONAG for this theory
  - Also discussed in the Maitzen video after minute 35 (optional to watch)

Integers are Simple Games

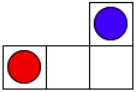
- Interpretation of integers:
- $0 = \{\,|\,\}$: no one can play, no moves
- $1 = \{0|\,\}$: Left can play one move, no moves for Right
- $-1 = \{\,|0\}$: Right can play one move, no moves for Left
- $2 = \{1|\,\}, -2 = \{\,|-1\}$ Two moves for one player, opponent cannot play
- $n = \{n-1|\,\}$: Left has n moves
- $-n = \{\,|-(n-1)\}$: Right has n moves

Integers in Snort

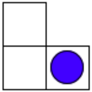
- Integers:
  - Free moves for one player
  - Opponent cannot play
- Color and sign conventions:
  - Left = Blue = Black = positive
  - Right = Red = White = negative
- Examples: numbers in Snort



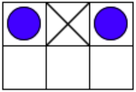
More Snort Examples - Some Games are Not Numbers



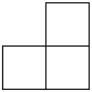
$* = \{ 0 \mid 0 \}$



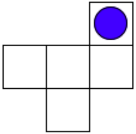
$\{1 \mid 0\}$



$\{ 2 \mid 0 \}$



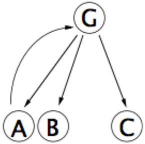
$\{ 2 \mid -2 \}$



$\{ 3 \mid -2 \}$

More Games Examples - Loopy Games

- Conway's definition allows cycles ("loopy games"), infinitely long play
- We focus on finite games
- In recent years, some progress on understanding "loopy" games as well
- Important application - "Ko" fights in Go



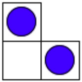
G = {A,B | C}, A = { | G}

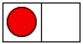
Inverse of a Game

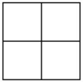
- Idea: switch the roles of Left and Right
  - Exchange Left and Right options
  - Keep doing that recursively
- For $G = \{L_1, \cdots, L_n | R_1, \cdots, R_m\}$
- Inverse of G: $-G = \{-R_1, \cdots, -R_m | -L_1, \cdots, -L_n\}$
- Snort: swap colors Red, Blue
- Clobber: swap colors Black, White
  - `BBW` is inverse of `WWB`, written `BBW = -WWB`

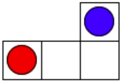
Outcome Classes of Games

- Four possible outcome classes $\mathscr{L}, \mathscr{R}, \mathscr{N}, \mathscr{P}$, with Snort examples
- Class $\mathscr{L}$, games $G > 0$: **L**eft wins (no matter who starts)
- Class $\mathscr{R}$, games $G < 0$ **R**ight wins (no matter who starts)
- Class $\mathscr{P}$, games $G = 0$ Second (**p**revious) player wins
- Class $\mathscr{N}$, games $G \not\gtrless 0$ First (**n**ext) player wins
  "*G* is *confused* with 0"
  "*G* and 0 are not comparable"









Examples of Outcome Classes in Clobber

G vs. 0	Class, Meaning	Clobber example
$G > 0$	$\mathscr{L}$, Left = Black wins	`BBW`
$G < 0$	$\mathscr{R}$, Right = White wins	`BWW`
$G = 0$	$\mathscr{P}$, Second player wins	`BWBWBW`
$G \not\gtrless 0$	$\mathscr{N}$, First player wins	`BW`

Finding Outcome Classes by Search

- Given a game G
- Use search to find the winner (e.g. boolean negamax, proof number search)
- Two searches: Left plays first, Right plays first
- $2 \times 2 = 4$ possible results
- Each result corresponds to one of the four outcome classes

Black first	White first	Outcome class, G vs. 0
Black wins	Black wins	$\mathscr{L}$, $G > 0$
White wins	White wins	$\mathscr{R}$, $G < 0$
White wins	Black wins	$\mathscr{P}$, $G = 0$
Black wins	White wins	$\mathscr{N}$, $G \gtrless 0$

Comments

- The case $G = 0$ may be strange at first (but it will make sense in sums!)
- All second player wins have the same value 0
- Games confused with 0 are sometimes “hot” - both players can gain additional moves from playing there
  - Clobber has no such hot games positions
  - Many other games such as Go, Snort have them

Sum of Games

- Choice: play either in G or in H
- Leave other subgame(s) unchanged
- $G + H = \{G + H^L, G^L + H | G + H^R, G^R + H\}$
- Remark: (slightly mis-)using the set notation here
  - Notation $G + H^L$ means: the set of games of the form $G + h$, where h is one of the games in set H^L
- Sum of more than two games: $G + H + K = (G + H) + K$
  - Math remark: addition of games is commutative, associative

Sum of Games - Clobber Example

- Clobber example: add games by putting them on the same board, with empty point(s) in between
- `BBWB` + `WWWWB` = `BBWB.WWWWB`
- Can you see that the games have the same moves and outcomes?

Adding a Game and its Inverse

- Theorem: $G + (-G) = 0$
- Proof: second player wins by *strategy stealing*
- Math remarks: Finite games behave a lot like numbers.
  - They form a group
  - Closed under addition
  - Commutative, associative
  - Have a zero element and an inverse

Strategy Stealing - Clobber Example

- $G + (-G) = 0$
- Second player win by strategy stealing
- G = `BBBWB`, -G = `WWWBW`
- G + (-G) = `BBBWB` + `WWWBW`
- Same as: `BBBWB.WWWBW`
- Any move the first player makes in one subgame, the other player can mimic in the other subgame
- The second player always has an answer, gets the last move
- Let's try it out!

Comparing Games

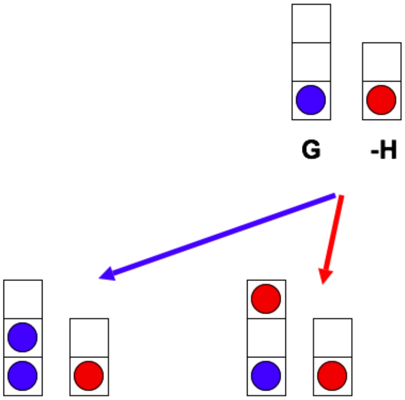
- How to compare games G, H?
- When can we say that two games are equal?
- When can we say that one game is better than another for a player?
- If all games were numbers it would be easy
- In general, we need to use search (or math) to figure it out
- How exactly?

Comparing Games by Playing the Difference Game

- How to compare games G, H?
- Play the *difference game* $G - H = G + (-H)$
- $G - H > 0$ means: $G > H$
- $G - H = 0$ means: $G = H$
- Same for $<$, $\gtrless$
- $G > H$: for the Left player, G is better than H

Comparing Games (2)

- Blue can win going first
- Red can also win going first
- $G - H$ is a first-player win, $G - H \not\gtrless 0$
- $G \not\gtrless H$, G and H are incomparable



Comparing Games for Pruning the Game Tree

- We can compare different options for a player
- Example: game $\{G, H \cdots | \cdots\}$
- Left has two options G, H
- Assume we can show $G > H$ by playing the difference game
- Then a rational player will always prefer G
- In mathematical CGT, this is the basis for pruning by *removing dominated options* (later)
- In algorithms for sum games, we can use this simple idea in many ways to simplify our search, prune moves
- More details next time

Incomparable Games

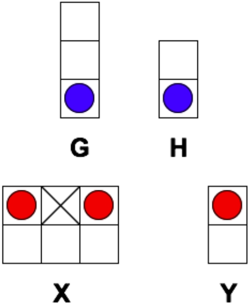
- What does it mean to say that G and H are *incomparable*?
- In some sums, it is better to have G than H
- In other sums, it is better to have H than G
- Games are only partially ordered
- Example: there are games X, Y such that:
  - $X + G$: Left can win going first
  - $X + H$: Left going first loses

  But:

  - $Y + G$: Left going first loses
  - $Y + H$: Left can win going first

Incomparable Games - Snort Example

- $G = \{1|0\}$, $H = 1$
- What if Left = Blue goes first?
- $X = \{0| - 2\}$
  - $X + G$: Blue **loses**
  - $X + H$: Blue **wins**
- $Y = -1$
  - $Y + G$: Blue wins
  - $Y + H$: Blue loses



Summary and Outlook

- Introduced basic ideas of CGT, with examples
- Most researchers who work on CGT are pure mathematicians
  - Not so many computing scientists
  - We have a small but successful CG algorithms research group here in Alberta
- Many efficient algorithms are still waiting to be discovered and applied to games