Computing Science (CMPUT) 657

Algorithms for Combinatorial Games

Martin Müller

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

Winter 2022

Comparing and Simplifying Games

Names and Notation for Outcome Classes

Game property	Meaning	Outcome class	Notation
$G > 0$	Left = Black wins	$\mathcal{L}$	L-position
$G < 0$	Right = White wins	$\mathcal{R}$	R-position
$G = 0$	Second player wins	$\mathcal{P}$	P-position
$G \not\gtrless 0$	First player wins	$\mathcal{N}$	N-position

- Why these names? Tradition in CGT
- Left = positive
- Right = negative
- P = Previous player wins (same as our second player)
- N = Next player wins (same as our first player)

Examples of Using Names and Notation for Outcome Classes

- $0 \in \mathcal{P}$, 0 is a P-position
- Clobber: $\texttt{BW} \in \mathcal{N}$, $\texttt{BW}$ is a N-position
- $\texttt{BBW} \in \mathcal{L}$, $\texttt{BBW}$ is a L-position
- $\texttt{BWW} \in \mathcal{R}$, $\texttt{BWW}$ is a R-position

When are Two Games Equal?

- What does it mean to be equal?
- There could be many different definitions
- Same game tree?
- Same winner?
- In CGT, it is something else
- An **equivalence relation**, as in arithmetic

Example - Equal Arithmetic Expressions

- Example, equality relation in arithmetic:
- 3 + 5 = (10 - 6) * (5 - 3)
- Proof: simplify each side to a *canonical form*, then compare
- 8 = 8

When are Two Games Equal?

- Same kind of defined equality relation for games
- We also have math operations to compute a canonical form
- Practical problem
  - Unlike arithmetic, canonical form of a game can be very complex
- Fortunately, we have another, search based algorithmic way to test equality
- Works directly on the games as given
- Avoids computing the canonical form
- It can still be slow if the games are large

More on Comparing Games

- Last time: we can compare games by playing the difference game $G - H$ = $G + (-H)$
- $-H$ is the inverse of H (switch colors Left $\leftrightarrow$ Right))
- Example: to prove that $G = H$:
- Show that $G - H = 0$, a second player win
- We can do that with two searches
  - B going first loses
  - W going first also loses
- These two results together prove that $G = H$

More on Comparing Games (2)

- Last time: distinguish 4 outcome classes by 2 searches
- How about $G \geq 0$, $G \leq 0$?
- Good news! Can do these with a **single** search!
- $G \geq 0 \quad \Leftrightarrow$ If White goes first, Black wins
- $G \leq 0 \quad \Leftrightarrow$ If Black goes first, White wins

Black first	White first	Compare to 0
Black wins	Black wins	$G > 0$
White wins	White wins	$G < 0$
White wins	Black wins	$G = 0$
Black wins	White wins	$G \not\gtrless 0$
(don't care)	**Black wins**	$G \geq 0$
White wins	(don't care)	$G \leq 0$

The Special Game 0

- In arithmetic, 0 is the neutral element for addition
- $x + 0 = 0$ for all x
- The same is true for games:
- $G + 0 = G$ for all games G
- **Any** second player win is *equal to* 0
- So $G + 0 = G$ means:
  $G + H = G$ for all games G,
  and for all 2nd player wins H

All Second Player Wins are equal to 0

- $G + H = G$ for all games G, and all 2nd player wins H
- Proof: play the difference game!
- $G + H = G \Leftrightarrow G + H - G = 0 \Leftrightarrow (G - G) + H = 0$
- Note: game addition is associative,
  a + (b + c) = (a + b) + c (from definition of sum)
- Last time: $G - G$ is a second player win by mimicking
- Assumption: H is 2nd player win
- $(G - G) + H = 0$ is also 2nd player win
- Proof: follow 2nd player win strategies in both $G - G$ and H

Using 0 for Pruning

- $G + 0 = G$
- This is extremely useful for pruning in search
- We can just remove any zero subgames
- We can remove any subsets of several subgames that add up to zero...
- ... such as $G + (-G)$
- Example: $1 + (-1) + \{0|1\} + \{-1|0\} = 0$
  - Why? 1 + -1 = 0
  - $-\{0|1\} = \{-1|0\}$, so $\{0|1\} + \{-1|0\} = \{0|1\} - \{0|1\} = 0$

Shorthand Notation for Games

- Shorter form - often used in literature
- omit curly brackets $\{\}$: $\{3 \mid 2\} = 3 \mid 2$
- Indicate precedence by multiple |: ||, |||,...
  $\{3 \mid \{2 \mid 1\}\} = 3 \mid \mid \{2 \mid 1\} = 3 \mid\mid 2 \mid 1$
- Use special symbols for frequently occurring games
  Numbers, $* = \{0 \mid 0\}$, *n, up, down, many more...

Simplifying Games

- How can we simplify a combinatorial game?
- Simpler games are faster to search
- Some ideas:
  - Use symmetry
  - cancel games that add to 0
  - remove bad moves
  - stop early by using an endgame database
  - ...

Simplifying Games - the Theory

- Two simplification methods
    1. Remove dominated options
    2. Bypass reversible moves
- Simplify a game as long as possible
- Any order of simplification steps leads to same result
- Result is a unique *canonical form*

Simplifying Games - Algorithms

- In this class we will **NOT** purely follow the math approach
- Main problem: computational complexity
- Even sums of small games can have **HUGE** canonical forms
- We want algorithms that solve specific sum games quickly
- We will use some of the ideas, if they are fast to implement...

Simplifying Games - Clobber Examples

- $G =$ `.BW.BB.WW..WWWB.WBBB.BW.BBWB..`
- Simplify: strip dead parts, empty `.` at end
- $G =$ `BW.BB.WW..WWWB.WBBB.BW.BBWB`
- Simplify: break into subgames
- $G =$ `BW` + `BB` + `WW` + `WWWB` + `WBBB` + `BW` + `BBWB`
- Simplify: remove zero subgames
- `BB` = 0, `WW` = 0
- $G =$ `BW` + `WWWB` + `WBBB` + `BW` + `BBWB`

Simplifying Games - Clobber Examples

- $G =$ BW + WWWB + WBBB + BW + BBWB
- Simplify: recognize and add up simple games
- BW = *, BW + BW = * + * = 0
- $G =$ WWWB + WBBB + BBWB
- Simplify: recognize and remove game + inverse, $G - G = 0$
- WWWB = -WBBB
- $G =$ BBWB

Remove Dominated Options

- $L_1 \geq L_2$ means $L_1 > L_2$ or $L_1 = L_2$
- Two Left options L_1, L_2, with $L_1 \geq L_2$: can prune L_2
- Two Right options R_1, R_2, with $R_1 \leq R_2$: can prune R_2
- Example: $\{2, -5, 6, 3 | -2, 6, 13, -8\} = \{6 | -8\}$
- General case: decide if $L_1 \geq L_2$ by checking whether Left can win the difference game $L_1 - L_2$ as second player
- After removing dominated options, all remaining options for a player are incomparable

Remove Dominated Options - Example

- $G = \{2, 2*, -1| - 1, \{1| - 2\}, \{2|0\}\}$
- Prune left options:
  - $2 > -1$, so prune -1
  - $2, 2*$ incomparable (why?)
- Prune right options:
  - $-1 < \{2|0\}$ (why?), so prune $\{2|0\}$
  - $-1, \{1| - 2\}$ incomparable (why?)
  - $\{1| - 2\}$ is in canonical form
- Canonical form: $G = \{2, 2 * | - 1, \{1| - 2\}\}$

What About Bypassing Reversible Moves?

- To reduce a game to canonical form, need to apply both repeatedly:
  1. Remove dominated options
  2. Bypass reversible moves
- Bypass reversible moves is a bit more complicated
- I do not know an efficient algorithm (just "try all")
- We will discuss it in the CGT math lecture
- We can maybe develop a better algorithm once we discuss more concepts, such as temperature
- Note: We can always check if a simplification is valid by searching the difference game to verify if $G - GSimple = 0$

Summary

- More details on comparing games
- Use search(es)
- Use for pruning, removing dominated options
- Special role of 0 = any 2nd player win