Introduction To Game Theory: Two-Person Games of Perfect Information and Winning Strategies

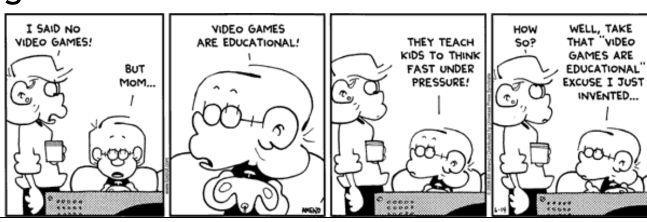
Wes Weimer, University of Virginia

Lecture Outline

- Introduction
- Properties of Games
- Tic-Toe
- Game Trees
- Strategies
- Impartial Games
  - Nim
  - Hackenbush
- Sprague-Grundy Theorem

Game Theory

- **Game Theory** is a branch of applied math used in the social sciences (econ), biology, compsci, and philosophy. Game Theory studies *strategic* situations in which one agent's success depends on the choices of other agents.



Broad Applicability

- Finding equilibria (Nash) – sets of strategies where agents are unlikely to change behavior.
- Econ: understand and predict the behavior of firms, markets, auctions and consumers.
- Animals: (Fisher) communication, gender
- Ethics: normative, good and proper behavior
- PolySci: fair division, public choice. Players are voters, states, interest groups, politicians.
- PL: model checking interfaces can be viewed as a two-player game between the program and the environment (e.g., Henzinger, ...)

Game Properties

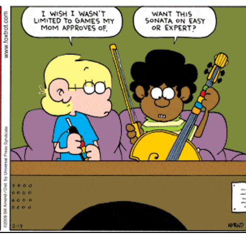
- *Cooperative* (binding contracts, coalitions) or ***non-cooperative***
- ***Symmetric*** (chess, checkers: changing identities does not change strategies) or *asymmetric* (Axis and Allies, Soulcalibur)
- *Zero-sum* (poker: your wins exactly equal my losses) or ***non-zero-sum*** (prisoner's dilemma: gain by me does not necessarily correspond to a loss by you)

Game Properties II

- *Simultaneous* (rock-paper-scissors: we all decide what to do before we see other actions resolve) or ***sequential*** (your turn, then my turn)
- ***Perfect information*** (chess, checkers, go: everyone sees everything) or *imperfect information* (poker, Catan: some hidden state)
- *Infinitely long* (relates to set theory) or ***finite*** (chess, checkers: add a "tie" condition)

Game Properties III

- ***Deterministic*** (chess, checkers, rock-paper-scissors, tic-tac-toe: the “game board” is deterministic, even if the players are not) vs *non-deterministic* (Yahtzee, Monopoly, poker: you roll dice or draw lots)
- More later ...

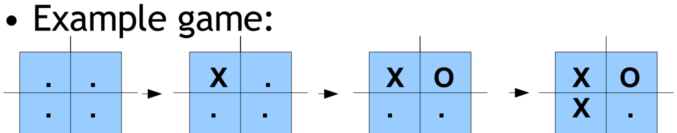


Game Representation

- We will represent games as **trees**
  - Tree of all possible game instances
- There is one **node** for every possible state of the game (e.g., every game board configuration)
  - **Initial Node:** we start here
  - **Decision Node:** I have many moves
  - **Terminal Node:** who won? what's my score?

Introducing: Tic-Toe

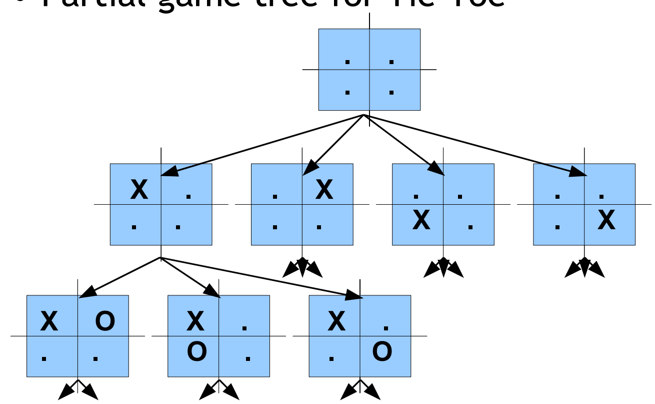
- ***Tic-Toe*** is like Tic-Tac-Toe, but on a 2x2 board where two-in-a-row wins (<u>not diagonal</u>).
  - X goes first
  - Resolutions: X wins, tie , X loses
- Example game:



  - Later: Does X always win?
  - Later: Does X always win if X is smart?

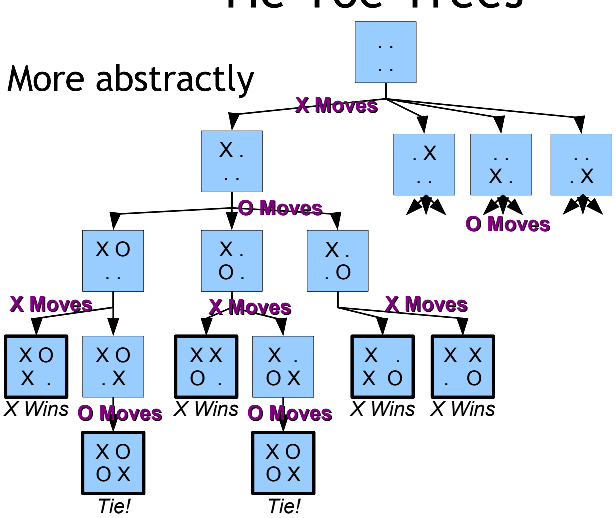
Tic-Toe Trees

- Partial game tree for Tic-Toe



Tic-Toe Trees

- More abstractly



More Definitions

- An ***instance of a game*** is a path through a game tree starting at the initial node and ending in a terminal node.
- ***X's moves*** in a game instance P are the set of edges along that path P taken from decision nodes labeled “X moves”.
- A ***strategy for X*** is a function mapping decision each node labeled “X moves” to a single outgoing edge from that node.

Still Going!

- A deterministic strategy for X, a deterministic strategy for O, and a deterministic game lead deterministically to a single game instance
  - Example: if you always play tic-tac-toe by going in the uppermost, leftmost available square, and I always play it by going in the lowermost, rightmost available square, every time we play we'll have the same result.
- Now we can study various strategies and their outcomes!

#13

Winning Strategies

- A ***winning strategy for X*** on a game G is a strategy S1 for X on G such that, **for all** strategies S2 for O on G, the result of playing G with S1 and S2 is a win for X.
- Does X have a winning strategy for Tic-Toe?
- Does O have a winning strategy for Tic-Toe?
- **Fact:** If the first player in a turn-based deterministic game has a winning strategy, the second player cannot have a winning strategy.
  - Why?

#14

Impartial Games

- An ***impartial*** game has (1) allowable moves that depend only on the position and not on which player is currently moving, and (2) symmetric win conditions (payoffs).
  - Only difference between Player1 and Player2 is that Player1 goes first.
- This is not the case for Chess: White cannot move Black's pieces
  - So I need to know which turn it is to categorize the allowable moves.
- A game that is not impartial is *partisan*.

#15

Nim

- ***Nim*** is a two-player game in which players take turns removing objects from distinct heaps.
  - Non-cooperative, symmetric, sequential, perfect information, finite, **impartial**
- One each turn, a player **must remove** at least one object, and may remove **any number** of objects provided they all come **from the same heap**.
- If you cannot take an object, **you lose**.
- Similar to Chinese game "Jianshizi" ("picking stones"); European refs in 16^{th} century

#16

Example Nim

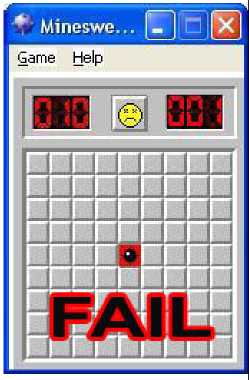
- Start with heaps of 3, 4 and 5 objects:
  - AAA, BBBB, CCCCC
- Here's a game:

AAA	BBBB	CCCCC	I take 2 from A
A	BBBB	CCCCC	You take 3 from C
A	BBBB	CC	I take 1 from B
A	BBB	CC	You take 1 from B
A	BB	CC	I take all of A
	BB	CC	You take 1 from C
	BB	C	I take 1 from B
	B	C	You take all of C
	B		I take all of B
			You lose! (you cannot go)

#17

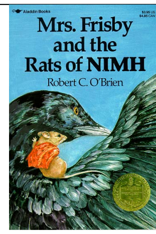
Real-Life Nim Demo

- I will now play Nim against audience members.
- Starting Board: 3, 4, 7
  - AAA, BBBB, CCCCCCC
- You go first ...



#18

The Rats of NIM



- How did I win every time?
  - Did I win every time? If not, pick on me mercilessly.
- Nim can be mathematically solved for any number of initial heaps and objects.
- There is an easy way to determine which player will win and what winning moves are available.
  - Essentially, a way to evaluate a board and determine its payoff / goodness / winning-ness.

Analysis

- You lose on the empty board.
- Working backwards, you also lose on two identical singleton heaps (A, B)
  - You take one, I take the other, you're left with the empty board.
- By **induction**, you lose on two identical heaps *of any size* (A^n, B^n)
  - You take *x* from heap A. I also take *x* from heap B, reducing it to a smaller instance of “two identical heaps”.

Analysis II

- On the other hand, you win on a board with a singleton heap (C).
  - You take C, leaving me with the empty board.
- You win with a single heap of any size (C^n).
- What if we add these insights together?
  - (AA, BB) is a loss for the current player
  - (C) is a win for the current player
  - (AA, BB, C) is what?

Analysis III

- (AA, BB, C) is a win for the current player.
  - You take C, leaving me with (AA, BB) – which is just as bad as leaving me with the empty board.
- When you take a turn, it becomes my turn
  - So leaving me with a board that would be a loss for you, if it were your turn
  - ... becomes a win for you!
- (AAA, BBB, C) – also a win for Player1.
- (AAAA, BBBB, CCCC) – also a win for Player1.

Generalize

- We want a way of evaluating nim heaps to see who is going to win (if you play optimally).
- Intuitively ...
- Two equal subparts cancel each other out
  - (AA, BB) is the same as the empty board (,)
- Win plus Loss is Win
  - (CC) is a win for me, (A,B) is a loss for me, (A,B,CC) is a win for me.
- What do we know that's kind of like addition but cancels out equal numbers?

The Trick!

- ***Exclusive Or***
  - XOR, ⊕, vector addition over GF(2), or ***nim-sum***
- If the XOR of all of the heaps is 0, you lose!
  - empty board = 0 = lose
  - (AAA,BBB) = $3 \oplus 3 = 0$ = lose
- Otherwise, goal is to leave opponent with a board that XORs to zero
  - (AAA,BBB,C) = $3 \oplus 3 \oplus 1 = 1$, so move to
    - (AAA,BBB) or (AA,BBB,C) or (AAA,BB,C)

Real-Life Nim Demo II

- I played Nim against audience members.
- Starting Board: 3, 4, 7
  - AAA, BBBB, CCCCCCC
- The nim sum is $3 \oplus 4 \oplus 7 = 0$
  - A loss for the first player!
- This time, I'll go first.

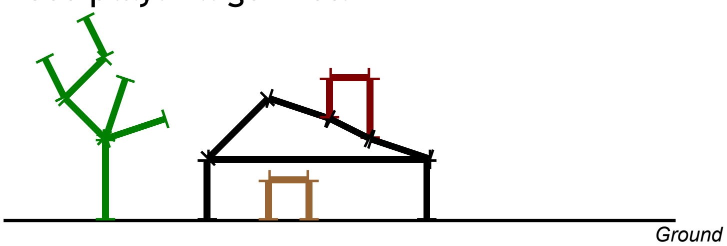
- You, the audience, must beat me. Muahaha!

Hackenbush

- ***Hackenbush*** is a two-player impartial game played on any configuration of line segments connected to one another by their endpoints and to a **ground**.
- On your turn, you "cut" **(erase) a line segment** of your choice. Line segments no longer connected to the ground are erased.
- If you cannot cut anything (empty board) **you lose**.

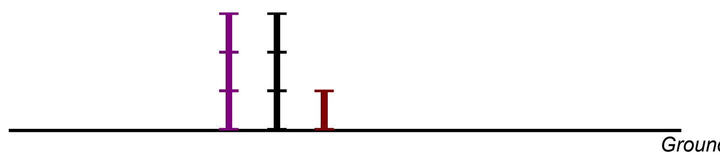
Hackenbush Example

- Each ⊢⊣ is a line segment. Ignore color.
- Let's play! I'll go first.



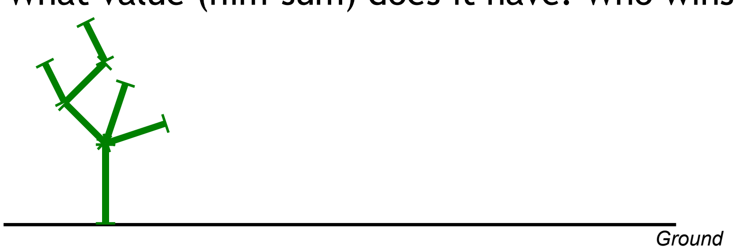
Hackenbush Subsumes Nim

- Consider (AAA, BBB, C) = (3,3,1) in Nim
- Who wins this *completely unrelated* Hackenbush game?



A Thorny Problem

- What about that Hackenbush tree?
- What value (nim-sum) does it have? Who wins?

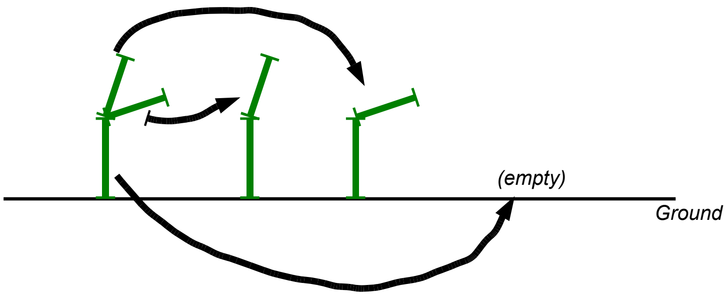


A Simple Twig

- Consider a simpler tree ...
- What moves do you have?

Twig Analysis

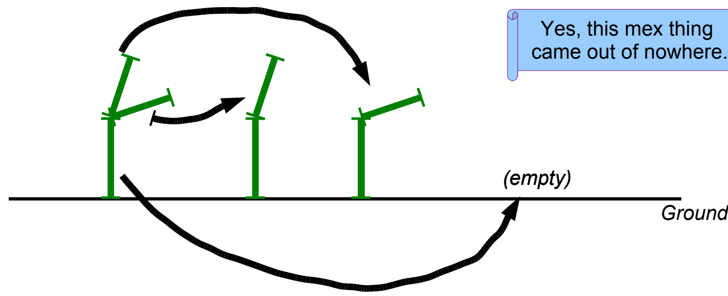
- Consider a simpler tree ...
- What moves do you have?



Maximum Excluded

- You can move to "2", "2" or "0".
- The ***minimal excluded*** of (2,2,0) is 1
  - mex(2,2,0) = 1 = value of that twig

Yes, this mex thing came out of nowhere.



Game Equivalence

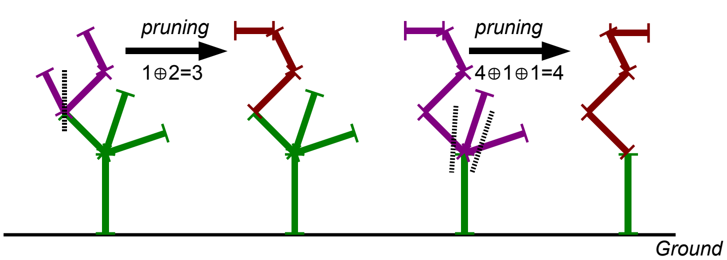
- I've claimed that the twig has nim-sum 1
- How to prove that? When are games equal?
- We write G ≈ G' when **G is equivalent to G'**.
- **Lemma 1.** Iff G≈G' then for all H, G⊕H ≈ G'⊕H.
- **Lemma 2.** G⊕G ≈ 0.
- **Lemma 3.** G ≈ G' if and only if G⊕G' ≈ 0.
  - Restated: G ≈ G' iff G⊕G' is a loss for Player 1.
  - If G ≈ G', then G⊕G ≈ G⊕G' (by Lemma 1).
  - Since G⊕G≈0 (by Lemma 2), we have 0≈G⊕G'.

A Simple Twig

- So twig≈1 if twig⊕1≈0
- twig⊕1≈0 means twig⊕1 is a first-player loss
  - You go first; two trials against me to verify ...

Generalized Pruning

- Can replace any subtree above a single branch point with the XOR of those branches
  - Via similar game-equivalence argument



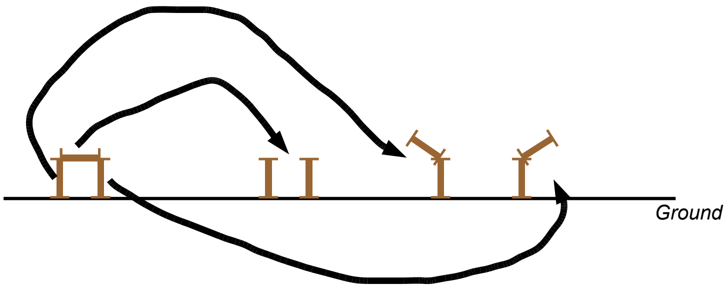
The whole tree has value "5".

Door Analysis

- What about cycles?
- What is the value (nim-sum) of this door?

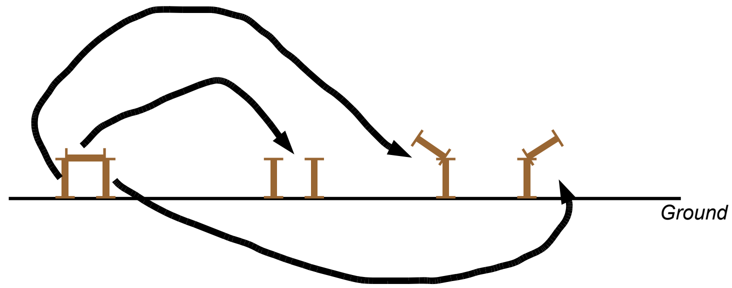
Door Analysis

- Well, what moves can you take from here?



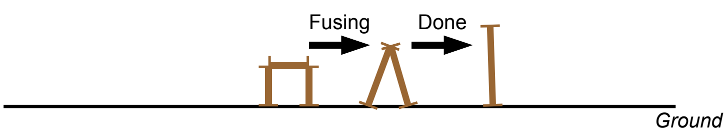
Door Analysis

- You can move to “0”, “2” or “2”.
  - mex(2,2,0) = 1 (recall: minimal excluded)
  - Value of door = 1



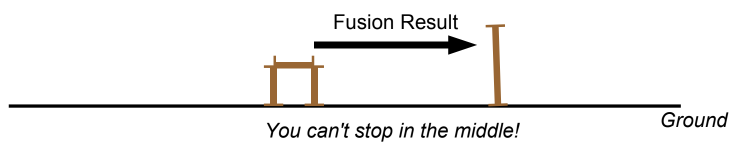
Fusion Principle

- We may replace any cycle with an equivalent subgraph where all of the non-ground vertices of that cycle are fused into one vertex and all of the ground vertices of that cycle are fused into another vertex.



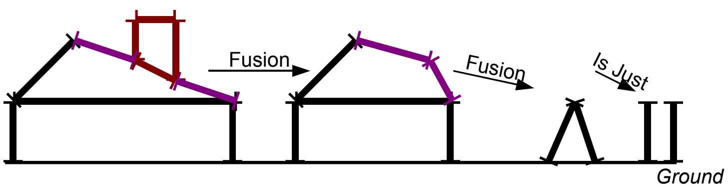
Fusion Principle

- We may replace any cycle with an equivalent subgraph where all of the non-ground vertices of that cycle are fused into one vertex and all of the ground vertices of that cycle are fused into another vertex.



Cold Fusion

- Let's boil the house down to something simple!

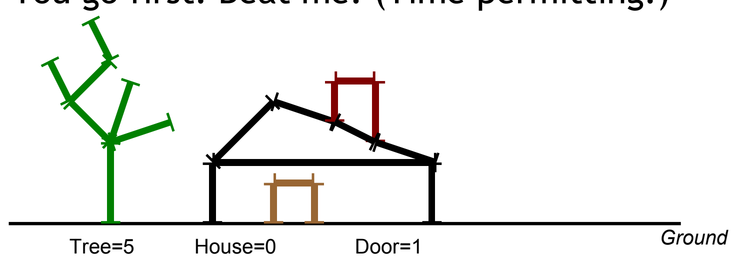


The whole house has value 1⊕1=0.

How would I check that?

Hackenbush Example

- This board has value 5⊕0⊕1=4.
- You go first. Beat me. (Time permitting.)



Why Do We Care?

- ... about Nim and Hackenbush?
- **Theorem (Sprague-Grundy, '35-'39).** ***Every impartial game is equivalent to a nim sum.***
- Proof: How?
  - Hint: what is the most important proof technique in computer science?

Why Do We Care?

- ... about Nim and Hackenbush?
- **Theorem (Sprague-Grundy, '35-'39).** ***Every impartial game is equivalent to a nim sum.***
- Proof: By structural induction on the set (tree) representing the game.
  - Proof not shown here
  - Proof sketch can be found at end of slide set

Old-School CS Work

- Explore a new formalism
- Define properties and categories
- Investigate a few popular instances
- Show that many interesting instances are in fact in the ***same equivalence class***
- ... and thus that your results about that equivalence class have broad applicability.
- **Today: all impartial games are just nim!**

Questions?

Sprague-Grundy Proof!

- **Theorem (Sprague-Grundy, '35-'39).** ***Every impartial game is equivalent to a nim sum.***
- Proof: By structural induction on the set (tree) representing the game.
  - Let $G = \{G_1, G_2, \ldots, G_k\}$. G_i is the game resulting if the current player takes move i.
  - By IH, each G_i is a nim sum, $G_i \approx N_i$.
  - Let $m = \text{mex}(N_1, N_2, \ldots, N_k)$. We'll show: $G \approx m$.

Sprague-Grundy Proof

- Let $G' = \{N_1, N_2, \ldots, N_k\}$. Then $G \approx G'$. Why?
  - Player 1 makes a move i in G to $G_i \approx N_i$. Then Player 2 can make a move equivalent to N_i in G'. So the resulting game is a first-player loss, so by Lemma 3, $G \approx G'$.
- To show $G \approx m$, we'll show $G+m$ is a first-player loss.
- We'll give an explicit strategy for the second player in the *equivalent* G'+m.

Sprague-Grundy Proof II

- To Show: P2 Wins in G'+m
- Suppose P1 moves in the m subpart to some option q with q<m. But since m was the minimal excluded number, P2 can move in G' to q as well.
- Suppose instead P1 moves in the G' subpart to the option N_i.
  - If N_i < m then P2 moves in the m subpart from m to N_i.
  - If N_i > m then P2, using the IH, moves to m in the G' subpart (which has been reduced to the smaller game N_i by P1's move). There must be such a move since N_i is the mex of options in N_i. If m<N_i were not a suboption, the mex would be m!
- Therefore, G'+m is a first-player loss. By Lemma 1, G+m is a first-player loss. So G≈m. QED.