Game Theory and Nim

Dr. Michael Canjar
Department of Mathematics, Computer Science and Software Engineering
University of Detroit Mercy

26 January 2010

Background

- Nim is a simple game, easy to play.
- It illustrates basic principles of Game Theory.
- Very Elegant solution to it easily accessible to Math and CS students. (Just need Binary Numbers.)
- Makes a Nice "Math-CS Club" Talk
- I first learned about it in 1983, from Dr. Richard Stark at St Mary's College of Maryland. http://www.smcm.edu/catalog/directory/faculty.html
- There is a Wikipedia article on it, which has gotten better, but still not as clear as Stark's treatment of it. http://en.wikipedia.org/wiki/Nim

The Game of Nim

- *Nim* is a two player game, in which players take turns on alternate moves .
- A *Nim* Board consists of several piles of objects, such as

A	B	C	D
x x x x x	x x	x x x	x

- I will describe this Board as [5 2 3 1]
- In each turn, players must pick up some (≥1) of the objects, but from only one pile.
- In Example, we have 11 possible moves. We could pick up
  From A : 1,2,3,4 or 5
  From B: 1 or 2 objects
  From C : 1, 2 or 3 objects.
  From D : 1 object

Game is Finite

Nim is a game in which players take alternate turns. On each turns they must pick up some objects but from only one pile.

- Obviously, the game will eventually ends.
- No ∞ loops.
- In my Example:

A	B	C	D
x x x x x	x x	x x x	x
5	2	3	1

The game can't last beyond 11 moves, because there are 11 objects, and that number goes down each turn.

Object of the Game

Nim is a game in which players take alternate turns. On each turns they must pick up some objects but from only one pile.

- Obviously, the game ends when the last piece is picked up. What is the Object of the Game ? Who Wins ?
- Two Variations on the Rules :

1. The Last Player wins
   sometimes called *Normal* game
2. The Last Player loses
   sometimes called a *Misere* game.

- Version 2 (*Misere*) is the more common.
- Note that *No Ties* (unlike Tic Tac Toe).

1/27/2011 11:59:22 AM 5

Demonstration

Via Computer

1/27/2011 11:59:22 AM 6

Demonstration

- Simple Computer Game
- From Most Boards the computer usually wins.
- That is, the computer can "force" a win, being able to cleverly counter any move that we make.
- For some Boards, we can force a win, if we play properly and make no errors.
- Question: Determine the Winning Strategy
  Characterize the Winning Positions and the proper responses.
- Interesting Feature: That for most of the game, we don't need to know the rules to make the "correct" play.

1/27/2011 11:59:22 AM 7

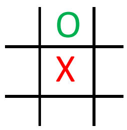
A Little Game Theory

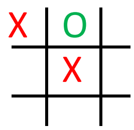
- We will say that a Board is a *Winning Board (WB)* or *Winning Position* if the current player can force a win from that position.
- We will say that a Board is *Losing Board (LB)* or *Losing Position* if the opponent can force a win
  (Opponent can counter to every move we make.)
- Say that a Board is non-determined (ND) if is neither a Winning Board or a Losing Board.
- *Theorem :* Every Nim Board is either a WB or a LB
  i.e. there are no ND Boards.

(in any Nim game, one of the players can force a win.)

1/27/2011 11:59:22 AM 8

Tic Tac Toe Example





X's turn:
X has a Winning Position

O's turn after X has made a winning move.
O has a Losing Position

Recursive Definition

- A Board is a LB if for every move we can make gives our opponent a WB
- A Board is a WB is we have some move to a LB.
- The terminal state (Empty Board) is specified by the Rules

    Final Board is a WB for Nim-Misere Game

    Final Board is a LB for Nim-Normal game
- This quasi-circular definition is an example of recursion.

Sketch of Proof

Theorem : Every Nim Board is either a Winning Position or a Losing Position
i.e. There are no Non-Determined (ND) Nim Boards.

- For contradiction, let B_0 be an ND Board.
- This means that there is a move M : $B_0 \rightarrow B_1$ which keeps the player alive. [B_1 is not WB]
- But B_1 is not a LB, since otherwise B_0 would be WB.

    So B_1 is an ND Board.
- Same argument shows that

    there is move from B_1 to an ND Board B_2.
- We get a chain of ND Boards $B_0 \rightarrow B_1 \rightarrow B_2 \rightarrow .. B_N \rightarrow$
- But this is an infinite loop, contradiction.

Elements of the Proof

- Properties of *Nim* that were used

1. There are no ∞ loops. All Games end. ("Tree is well-founded").
2. Each game ends with a Winner (and a Loser). That is No Ties.

- The Theorem applies to such Games in the same *genre*: Perfect Information with No Random Elements.

The Issues

Given a Nim Board :

- Determine if it is a Winning Board or a Losing Board
- If is a WB, find (all possible) winning moves.
- This function is called a Winning Strategy.

Brute Force Recursive Solution

- Given a Board 𝔅

  If it is empty (terminal) Apply Game Rules.
  Otherwise: Consider each possible Move
  Look at the Board Produced.
  - Apply the Algorithm to that Board
  - If any such Board is a LB then 𝔅 is WB
    else (all Boards are WB) then 𝔅 is LB
- Can Visualize with a Tree.
- This is a Top Down Algorithm.

Bottom Up Iterative Algorithm

- Our analysis of *Nim* lends itself to an iterative approach:
- Loop for N=0,1,2,3, …

  Analyze all sub-boards with N elements.
  At each you store the Results.
  You use the results of the previous stage.
- In Tree, we are working from the Bottom Up
- Can Use this to prove Theorem by Induction.

* Easier with Odometer Order (001, 002, …00M, 010, 011 …)
  [Code as a Base M numeral]

Are we done ?

- Problem : Given a Nim Board :
  - Determine if it is a Winning Board or a Losing Board
  - If is a WB, find (all possible) winning moves.
  - This function is called a Winning Strategy.
- Our Brute Methods do “Work”

  But . . .

Disadvantages of Brute Force

- A Mathematician would like an *elegant* solution.
  Some "closed form" method that will "compute" the winning moves from a Board.
- A Computer Scientist is concerned with Complexity Issues- running time and memory.
  If we consider NxM Nim Games (N piles of < M objects), the # of sub-boards is M^N
  For 10x10 games, we have 10 Billion.
- Let us work on a direct method.

Example

- We will see that the Board
  { 73 69 57 21 20 }
- is a Winning Board, which admits **only one** winning move (out of 240 possibilities):
  { 73 69 13 21 20 }
- The number of sub-boards is
  73 x 69 x 57 x 21 x 20 = 120,585,780
- Before we are done, will put the computation of the winning move on one slide.
- Easy with Binary Number

Binary Numbers

- Our Solution will involve Base-2 Numbers.
- The decimal number 167 is
  $1 x 10^2 + 6 x 10^1 + 7 x 10^0$
- In Binary is represented as 1010011 1
  $1x2^7+0x2^6+1x2^5+0x2^4+0x2^3+1x2^2+1x2^1+1x2^0$
  128 32 4 2 1

Converting to Binary

- Example : Convert 13 decimal to Base-2
- Divide 13 by 2 : Quotient = 6 Remainder = 1
- Divide 6 by 2 : Quotient is 3 Remainder = 0
- Divide 3 by 2 Quotient is 1 Remainder = 1
- Divide 1 by 2 Quotient=0 Remainder = 1

Done : Quotient of 0 : 1 1 0 1

Nim Boards in Binary

Given a Board [X_1 X_2 . . . X_n]

- Express the numbers in Binary and list them in column, lining up their respective bits in columns.
- In each Column compute the Sum of Bits, and classify the column as to whether or not this is Even or Odd.
- Call Board an *Even Position* if every column is even.
- Call it an *Odd Position* if any column is Odd.

  (Of course, most positions are Odd.)

[Preview: Non-Trivial Odd Positions are Winning
Non-Trivial Even Positions are Losing]

Example [9 8 5 4]

If Every Col. Is Even, Board is Even
If Any Col. Is Odd, the Board is Odd.

	8	4	2	1
9	1	0	0	1
8	1	0	0	0
5	0	1	0	1
4	0	1	0	0
Parity:	Even	Even	Even	Even

- [9 8 5 4] is even
- [9 8 5 x] is odd (x ≠ 4)
- [9 8 x 4] is odd (x ≠ 5)
- [9 x 5 4] is odd (x ≠ 8)
- [x 8 5 4] is odd (x ≠9)

	8	4	2	1
9	1	0	0	1
8	1	0	0	0
5	0	1	0	1
2	0	0	1	0
Parity:	Even	Odd	Odd	Even

	8	4	2	1
1	0	0	0	1
8	1	0	0	0
5	0	1	0	1
4	0	1	0	0
	Odd	Even	Even	Even

E-Lemma

- E-Lemma: Every Move from an Even Position results in an Odd Position.
- Proof: Any move changes only one row and must change at least one bit, which changes he parity of that column from Even → Odd

- What about Odd Positions ? Next Lemma will be:
- O-Lemma: From an Odd Position there is at least one move that results in an Even Position.

Odd→Even Ex: [11 7 10]

	8	4	2	1
11	1	0	1	1
7	0	1	1	1
10	1	0	1	0
	Even	Odd	Odd	Even

Need to Change Col 4 & 2

3 Piles (Rows) to Try :
Change each bit in Odd Cols

	8	4	2	1
13	1	**1**	**0**	1
7	0	1	1	1
10	1	0	1	0
	Even	Even	Even	Even

Adds 2 ☹

	8	4	2	1
11	1	0	1	1
1	0	**0**	**0**	1
10	1	0	1	0
	Even	Even	Even	Even

Subt 6 ☺

	8	4	2	1
11	1	0	1	1
7	0	1	1	1
12	1	**1**	**0**	0
	Even	Even	Even	Even

Adds 2 ☹

Odd→ Even Summary

Given Odd Position

- In each Pile, there is a "move" to an Even Position, by changing the bits in odd columns.
- Not all are legal Nim Moves.
  Need the first bit change to be "1→0"
- But Must be at least one "1" in first Odd Col
  since a cols of all 0s would be even.

Odd→Even
Ex: [11 7 10]

	8	4	2	1
11	1	0	1	1
7	0	1	1	1
10	1	0	1	0
	Even	Odd	Odd	Even

	8	4	2	1
13	1	1	0	1
7	0	1	1	1
10	1	0	1	0
	Even	Even	Even	Even

Adds 2 ☹

	8	4	2	1
11	1	0	1	1
1	0	0	0	1
10	1	0	1	0
	Even	Even	Even	Even

Subt 6 ☺

	8	4	2	1
11	1	0	1	1
7	0	1	1	1
12	1	1	0	0
	Even	Even	Even	Even

Adds 2 ☹

Given Odd Position

- In each Pile, can move to Even Pos : change the bits in odd cols.
- Not all are "legal" Nim Moves. Need the first bit changed is "1"
- Have at least one "1" in the first Odd Col, since all 0s is Even.

Example: [73 69 57 21 20]

	64	32	16	8	4	2	1
73	1	0	0	1	0	0	1
69	1	0	0	0	1	0	1
57	0	1	1	1	0	0	1
21	0	0	1	0	1	0	1
20	0	0	1	0	1	0	0
	Even	Odd	Odd	Even	Odd	Even	Even

Need "1" in First Odd Col : 32

Need to Change Col 32, 16, & 4

Example: [73 69 57 21 20]
Only 1 Move [73 69 13 21 20]

	64	32	16	8	4	2	1
73	1	0	0	1	0	0	1
69	1	0	0	0	1	0	1
13	0	0	0	1	1	0	1
21	0	0	1	0	1	0	1
20	0	0	1	0	1	0	0
	Even	Even	Even	Even	Even	Even	Even

Need "1" in First Odd Col : 32

Need to Change Col 32, 16, & 4

Odd→ Even Algorithm

O-Lemma : Given an Odd Position, there is at least one move to an Even Pos.

- In the first Odd Col, find one row (pile) in which there is a "1"

  Must be at least one "1" in the first Odd Col, since all 0s is Even
- In that Pile, change the bits in each Odd Col..

Winning Strategy (Even-Odd Strat)

OL : Every Move from an Odd Position produces an Even Position.
EL: From an Even Position, there is a move to an Odd Position.

- Always Give your opponents an Even Position
- They returns an Odd Position . . .
- Since the Empty Board is Even, this will win

  every Normal Game ☺
- But, it loses every *Misere* Game. ☹
- Need a little tweaking ! D-Lemma

Trivial Games

- Call a pile a *Singleton* if it has 1 elements.
- Call a board *Trivial* if all non-empty piles are singletons,

  let B = a trivial Board with N non-empty piles :

  [1 1 1 1 1 .... 1] <N – singletons >
- If N =1: we pick last object.

  Win Normal Game, Lose *Misere* Game.
- If N =2 : Opponent takes last object.

  Win *Misere* Game, Lose Normal Game.
- N =3: Wins Normal, Loses *Misere* Game
- N=4 Win *Misere* Game, Lose Normal Game.

Note that there is no real choice here.

Characterization of Trivial Boards

- L B be trivial board consisting of N objects. (N piles of 1 element each.)

- If N is Even

  Normal Game: B is LB

  Misere Game : B is WB

- If N is Odd

  Normal Game: B is WB

  Misere Game : B is LB

Decisive Positions

- Call a Board a Decisive Position if it contains exactly one non-singleton.
- Example: 17 1 1 1 . . . 1
- Pick from the "17"pile and either take it all or leave only 1 (depending on Rules)
- Result is a Trivial Board. Control whether we leave an Even or Odd number of Singletons.
- D-Lemma: A Decisive Position is a Winning Position for both types of Games.

DP Example

D-Lemma: A Decisive Position is a Winning Position for both types of Games.

- Given 4 piles: [13 1 1 1]
- If I want you to have an even number of singletons (Normal Game)

  I pick up 12 → [1 1 1 1]

If I want you to have an odd number of singletons (*Misere* Game)

I pick up all 13 →[0 1 1 1]

Existence of DP

- Any play of non-trivial Game will pass through a Decisive Position.
- There will be last non-trivial position and that must be Decisive.

  [Each move can only reduce the number on non-singletons by ≤ 1. Eventually get one with exactly 1 non-singleton]

Summary

- Even-Odd Plan: Given an Odd Position which is not decisive, return an even position.
  Opponent will return Odd Position
- The Decisive Pos is winning for both Rules.
- The Even-Odd Plan wins all Normal Games.

- The Decisive Position must be Odd.
  [Can Do this by direct computation – Exercise]

Complete Characterization

- A non-trivial Board is a WB iff it is Odd.
- Given Odd Board, we give our opponent an Even Board, until we have a Decisive Position.
- At the DP, we consult the rules and leave our opponent a trivial Board. Consult the Rule to decide whether to leave even/odd # of singletons.
- A Trivial Board is WB/LB depending on its parity (even/odd # of piles) N and the Rules of the Game.
- The move at Decisive Position is the *only* time we pay attention to the rules.

End Slides