# Combinatorial Games: Math 454 Lecture 20 (8/01/2017)

Cole Franks

August 1, 2017

## Contents



## 1 What is a combinatorial game?

**Definition 1.1.** A *combinatorial game* is usually defined as a game with

**Two players:** Two players, generally referred to as *Left* and *Right*, who alternate making moves.

**Positions:** Some positions; usually finitely many. We can think of positions as possible boards in chess or TTT.

**Available moves:** Some clearly defined moves (usually finite) available to Left and Right at each position. Moves are changes to the position.

**Winning/tying positions:** A subset of winning positions for Left, winning positions for Right, and some subset of tying positions. If a player moves to a winning position, they win (and hence the other player loses), and if they move to a tying position, well, they tie.

**Finite length:** some finite maximum number of moves. That is, the game must end.

**Perfect information:** Both players know the position and their available moves.

**No element of chance:** None of the players have to use chance to decide where to move. For example, you never need to role a dice and only make a move if you get at least some number.

**Fact 1.0.1.** Combinatorial games with no ties either have a winning strategy for the first player or a winning strategy for the second player.

*Proof.* Assume the first player is Left. Make a tree in which the root is labeled by the starting position and Left, denoting that it is Left's turn. Label the nodes of the children the positions that Left can move to, and Right, to denote that it is Right's turn. The children of those nodes should be the positions Right player can move to, and so on. The leaves of the tree will be winning positions for Left or Right. Color each leaf blue if it is a Left win and red if it is a Right win. Define the coloring of a node recursively as follows:

**Left node:** This means it is Left's turn. If some child of the node is colored blue, color the node blue. Else, color it red.

**Right node:** This means it is Left's turn. If some child of the node is colored red, color the node red. Else, color it blue.

If the root is red, the game is a Right (second player) win. Else it is a Left (first player) win. □

**Definition 1.2.** In theory, Left has a winning strategy if the root in the game tree is his color, and Right has a winning strategy otherwise. However, in practice, a winning strategy for Left means a function assigning an available move to each position such that if Left plays the assigned move each time, Left will win. Vice versa for right.

## 2 Strategy stealing

Consider tic-tac-toe. It is a game with the following important properties:

**Extra Moves Don't Hurt:** If a position is a winning position for Left (meaning Left can win with perfect play from that position), then any position obtained by making an additional Left move is also a win. *Vice versa for Right.*

**Symmetry:** From the starting position, any strategy employed by Left could also be employed by Right, and vice versa.

**Theorem 2.1.** *In such games, the second player cannot have a winning strategy.*

*Proof.* Suppose, for the sake of contradiction, that the second player had a winning strategy. The intuition of the argument is as follows.

- Begin with an arbitrary move.
- After second player moves, first player ignores his own first move and plays by second player's winning strategy. Essentially, first player pretends to be second player.

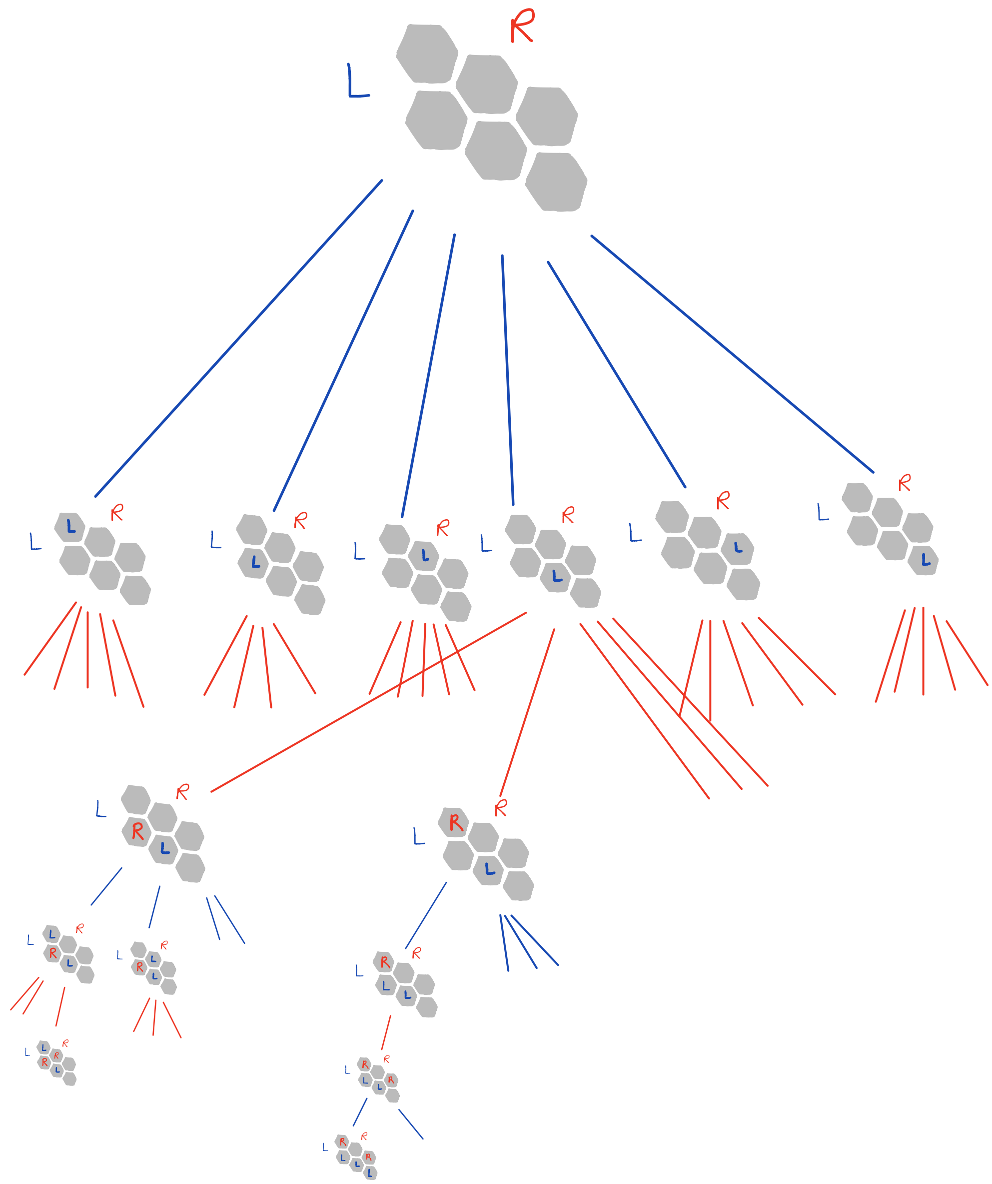

Figure 1: A partially filled out game tree for Left playing first in Hex, in which the objective of Left is to make a path from the side marked $L$ to the opposite side, and the objective of Right is to make a path from the side marked $R$ to the opposite side. Both players may claim any spot when it is their turn.

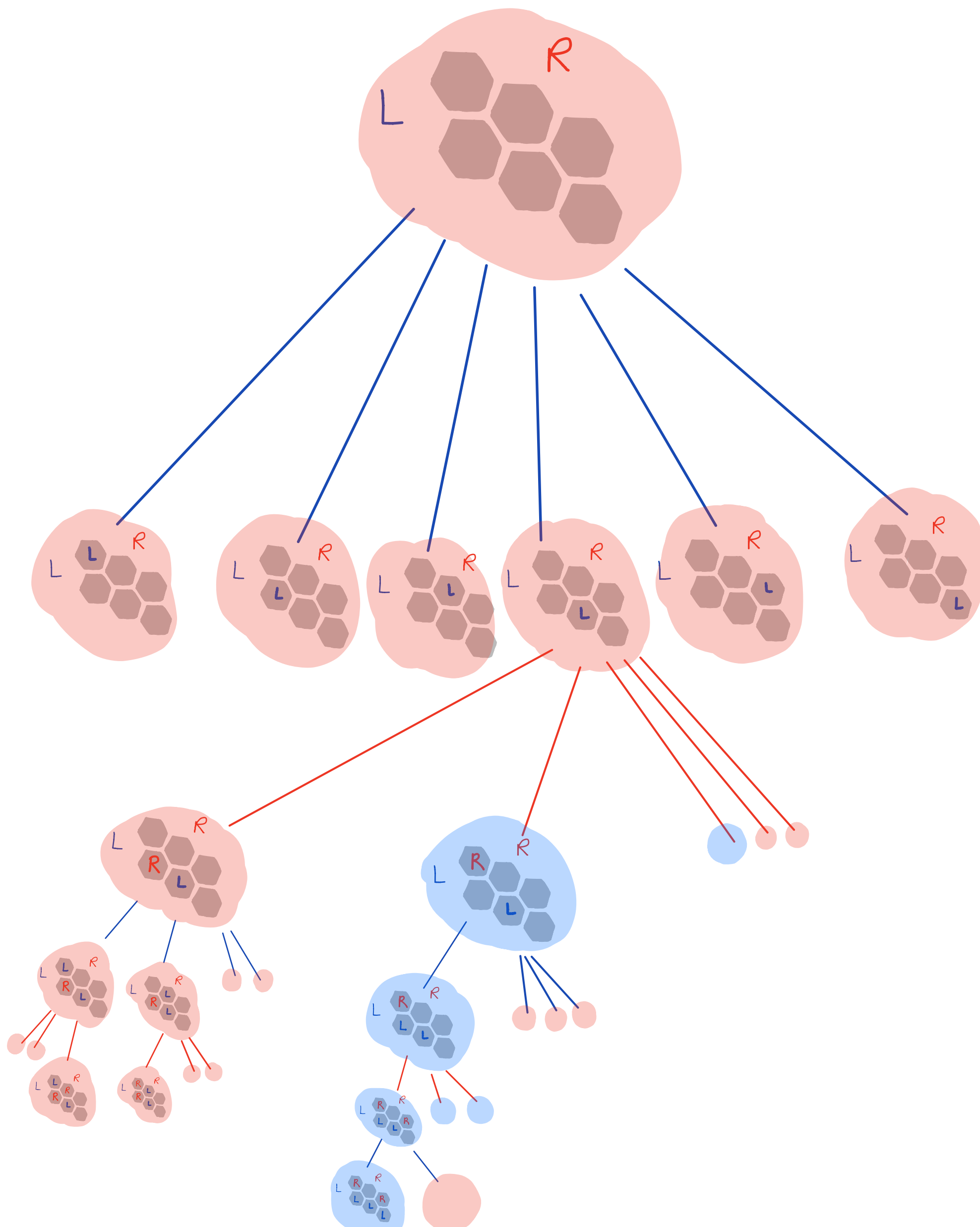

Figure 2: The same partially filled out game tree for Left playing first in Hex, except this time colored recursively as in the proof of Fact 1.0.1.

- If ever the strategy dictates first player should make the move first player already made, then first player makes an arbitrary move.

These extra moves never hurt first player, so pretending to be second player should make first player win, a contradiction.

A more formal argument would again be by contradiction: Assume Left is first, which makes no difference for the argument by the symmetry assumption. Suppose the starting position is a win for Right, which means no matter which move Left makes, the new position is a win for Right as first player. However, each of these positions is one Left move away from the starting position. By our symmetry assumption and our assumption that second player has a winning strategy, the starting position was a loss for Right if he went first. This means that a position that was a loss for Right as first player became a win for Right as first player by adding one Left move. However, this means that a win for Left as second player became a loss for Left as second player by performing a Left move, a contradiction to our "extra moves don't hurt" hypothesis. □

# 3 Nim

Nim is a game consisting of some number of stacks of coins, in which players alternate taking some number of coins from one of the stacks. The last player to take a coin wins. That is, if there are no coins left, a player loses.

We call it $n$-heap Nim if there are $n$ heaps. Nim is what's called an *impartial* game, meaning the set of moves available to Left and Right are the same. In this case we can forget about Left and Right, and there are only two possibilities for each position:

**Definition 3.1.** An $\mathcal{N}$-position is a position in which the next player wins. A $\mathcal{P}$-position is a position in which the previous player wins.

Thus, an impartial game is a first-player win if the starting position is an $\mathcal{N}$-position.

**Example 1.** First consider 1-heap nim - clearly an $\mathcal{N}$-position.

How about 2-heap nim? There are two cases:

- If the stacks are equal, then previous player can employ the *mirror argument* - whatever the next player takes, previous player takes the same number from the other stack. There will always be a coin left for him to take, so he wins. Thus, equal stacks is a $\mathcal{P}$-position.
- On the other hand, if the heaps are not equal, then next player can make them equal by removing enough from the larger stack. As he has converted it to a $\mathcal{P}$-position, he wins! Thus, unequal stacks is a $\mathcal{P}$-position.

Let's generalize this bizarrely. First note that two numbers being equal means that if their binary expansions are the same. For example $7 = 111$ and $5 = 101$. Another way to say this is that their binary expansions when treated as vectors sum to 0 mod 2: $7 = 7$, so $[1,1,1] + [1,1,1] = 0$. Now suppose we have multiple Nim stacks - represent each number as a binary vector. This is called a *nimber*. Add them up - the new vector represents a number in binary; this number is called the *nim-sum*. There are two cases:

**The nim-sum is not 0, or unbalanced case:** We claim next player can always subtract from a stack so that it the nim-sum becomes 0. To do this, *pick a stack $S$ that has a nonzero entry in the largest unbalanced bit.* We can replace this by any nimber where this bit is zero (this means the n**I**mber corresponds to a smaller n**U**mber). We can replace it by a heap size that causes the nim-sum to be come zero. In other words, we can turn this into a balanced game.

In particular, suppose the nim-sum is $N$. If we add the nim-sum to the nimber of the stack $S$ (adding as nimbers; vectors $\bmod 2$) we will get a new nimber for $S$. That nimber will have a 0 in the largest unbalanced bit, so it will be a smaller number, and the new nimber will be $N + N = 0$.

If the first player sets the nim-sum to 0 each time and decreases the total number of coins, eventually the total number of coins will be 0 after the first player's move. Thus, *the unbalanced case is an $\mathcal{N}$-position.*

**The nim-sum is 0, or balanced case:** The next player's move will cause the nim-sum to be nonzero. This is because the nim-sum is replaced by $0 + x + y = x + y$, where $x$ was the old nimber for the stack and $y$ was the new nimber for the stack - since $x$ and $y$ are not equal, the new nim-sum isn't zero. Thus, the next player's move converts the game to an $\mathcal{N}$-position no matter what, so *the balanced case is a $\mathcal{P}$-position.*

**Example 2.** If our position is given by stacks of size $108 = 1101100, 90 = 1011010, 57 = 0111001$, then the nim-sum is

$$\begin{array}{cccccccc} & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ + & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ + & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 1 & 1 & 1 & 1. \end{array}$$

Thus, this position is an $\mathcal{N}$-position because it is unbalanced. The next player should pick a stack where the third bit from right is nonzero, because that's the largest unbalanced bit. In this case, the only choice is the first stack of size 108. He adds the nim-sum to that nimber to obtain

$$\begin{array}{cccccccc} & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ + & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ + & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 0 & 1 & 1 & 1. \end{array} \longrightarrow \begin{array}{cccccccc} & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ + & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ + & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 0 & 0 & 0 & 0, \end{array}$$

which is a balanced game with stack sizes 107, 90, 57, so it is now a $\mathcal{P}$-position. This means the new next player (former previous player) can't do much - no matter what he does he will lose, and in particular he will leave an unbalanced position for the previous player. This will continue until all the stacks are 0.