THE THEORY OF TYPES FOR SUMS OF NORMAL-PLAY GAMES

ADRIAN BAEZ AND DANIELLE HARRINGTON

ABSTRACT. Game theory can be described as a series of methods which are used in order to determine the outcome of a game decisively without needing to complete said game. Within game theory, the genre of games which we discuss are those classified under normal-play games. Normal-play games are a subgenre of combinatorial games, and they have an outcome of either a win or a loss, and with this comes classifications for these results known as types, which specifies the player with the winning strategy. In our paper, we provide tools and formalism which allow us to determine the types of various games.

CONTENTS

Date: May 21, 2023.

1. Introduction

Game theory is the study of patterns and strategies through games with the objective to say as much about the game with as little information as possible. We can further categorize specific games by their given qualities to assume their outcomes with accuracy and, later in the paper, celerity, so let's try to do so using the following definitions:

Definition 1.1. A *combinatorial games* is a two-player game consisting of the following information:

- a set of positions to be played on/with,
- a moving rule for both Louise (for the Left player) and Richard (for the Right player) when at a specific position, and
- a win rule which is a set of positions we designate as "terminating", such that, when reached, indicates that one of the players won the game.

Definition 1.2. *Normal-Play Games*: Combinatorial games that do not end in a draw as the winner is oftentimes the last player with an available move that corresponds with their given moving rule.

Example 1.3. *Pick-Up-Bricks*: With the starting position of a specific number of bricks held in a circle, Richard and Louise can pick up either one or two bricks at the same time as whoever is able to pick up the last brick wins.

Example 1.4. *Chomp*: A rectangular grid consisting of $m \times n$ units with an " $\times$ " drawn in the leftmost-corner unit where, at the beginning of the game, Richard and Louise will be assigned to either remove any row(s) or column(s) they choose in as many turns they wish. In this case, the an " $\times$ " unit cannot be removed as the winner of Chomp will be the last one to remove either a row or column of the figure. For extra clarity, Richard and Louise cannot remove both rows in columns in the same game as either Richard will be in charge of removing columns, Louise rows, and vice versa.

1.1. Game Trees.

Game Trees are used to organize all possible moves taken from a specific game position in a way that can eventually show all terminating moves. Let's take a game tree for Pick-Up-Bricks with four bricks as an example, for when Richard moves first.

The labels on the edges reflect whose turn it is, and inside each node, we see the position of the game at that state. The notation for the terminating nodes on the bottom later are the following:

- A node holding " $+-$" describes a winning outcome for Louise as the plus is on the Left side of the node.
- A node holding " $-+$" describes a winning outcome for Richard as the plus is on the Right side of the node.
- A node holding "00" describes an outcome of a drawing strategy between both Richard and Louise as no one is gaining anything; we will not find this in the game tree of any normal-play games.

Although game trees are a nice, visual representation of game outcomes, they can be quite tedious. Imagine attempting to create a game tree for a game of Pick-Up-Bricks with an initial position

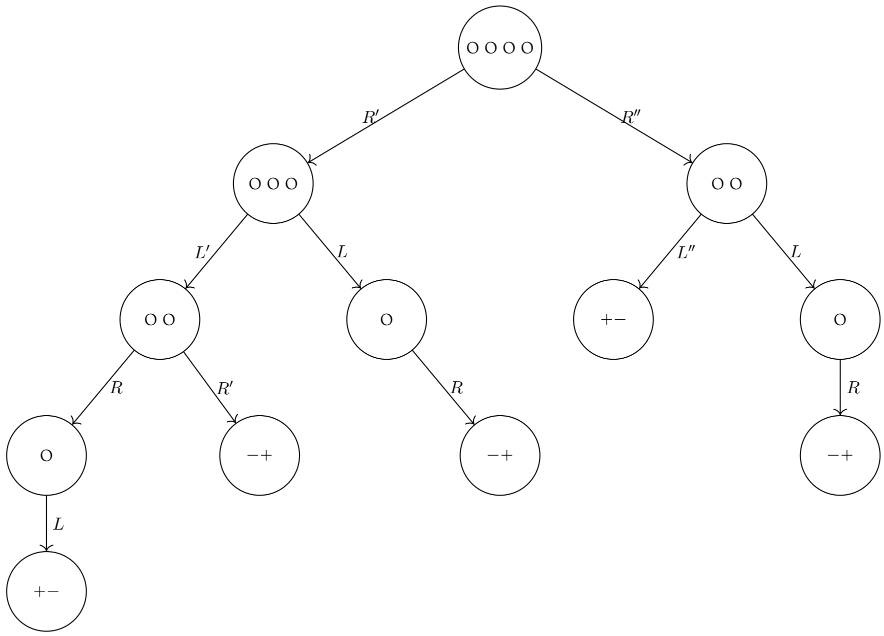

FIGURE 1. An example game tree

of one-million bricks, Chomp with a 300×300 grid, or even worse, an entire game of chess. To many, hopefully, this is not a simple task and is the reason in which we have game positions and types.

2. TYPES

In order to gain a further understanding of normal-play games, we must know their positions. Positions can be described as the configuration of the normal-play game at that moment as a result of both players actions, in other words, the current layout of the game.

FIGURE 2. Example of two positions which we will call A and B, respectively.

The *type* of a position in a normal-play game is an integral part of normal-play games as they show who will win said game, in a sense they capture the information of which player has a winning strategy in a normal-play game. Since there are no draws in normal-play games, there are four different types:

- *L type* means Louise has a winning strategy,
- *R type* means Richard has a winning strategy.
- *N type* means that the First/Next player in the game has a winning strategy,
- *P type* means that the Second/Previous player in the game wins.

So how can we be so sure that all normal-play game positions are of one of these four types? This can be established by the proof of Zermelo's Theorem.

Theorem 2.1. *In a combinatorial game, either one of the two players have a winning strategy, or they both have a drawing strategy.*

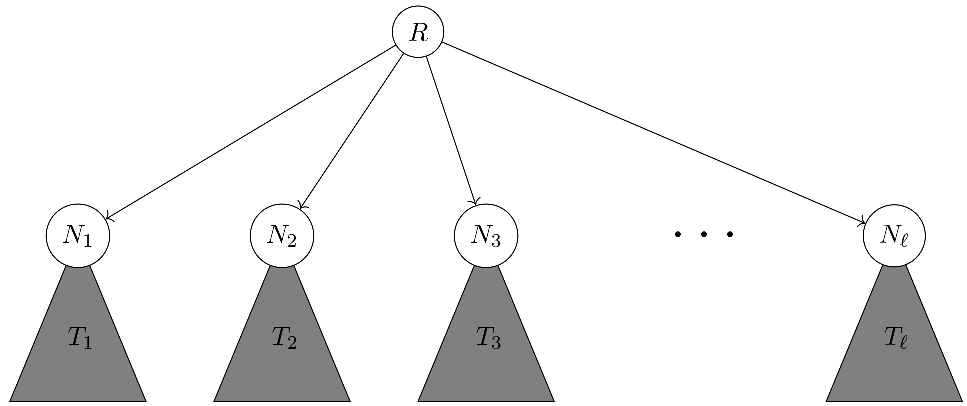

FIGURE 3. Using induction to prove Zermelo's theorem

Proof. **Base Case.** If a game tree were to have a depth, or the maximum number of possible moves to be made, of zero, our game would be trivial and the result of the game to already have been determined.

Inductive Step. In our game tree T, assuming that its depth, n, is greater than zero, if Richard were to move first he would be able to move to any N_i node on the game tree. It is completely possible for these N_i nodes to be the root of a new hypothetical game tree considering that game trees can be started using any game positions. Since every T_i tree has a depth $< n$, for every $1 \leqslant i \leqslant \ell$ tree T_i knowing that either player has a winning strategy or both are drawing, we can describe these strategies as L_i for Louise and R_i for Richard, noted through L_1, L_2, ..., L_i and R_1, R_2, ..., R_i. From here we form two cases in regards to our root nodes:

Case 1. At least one of $T_i, \ldots, T_\ell$ is type $-+$.

Let T_i be type -+, indicating that Richard has a winning strategy in position R_i. This winning strategy will be followed through if Richard plays to the N_i node.

Case 2. All of $T_1, \ldots, T_\ell$ are type $+-$.

Considering that every strategy in this case, L_i is winning, L is a winning strategy for Louise.

Case 3 None of $T_1, \ldots, T_\ell$ is type $-+$, but least one is type 00.

Suppose T_i is type 00. In this case, since Richard cannot win, his strategy R_i can only be a drawing strategy; formed by making Richard play on the corresponding N_i node. Knowing that $T_1, \ldots, T_e$ has both drawing strategies and no winning strategies for Richard, Louise's strategies $L_1, \ldots, L_\ell$ are either drawing or winning, L being a drawing strategy for Louise.

□

Now that we have proved that all positions of normal-play games each have their own types, this means that position A and position B should be either the same or different types. Referring back to Figure 2, the types of the positions are

- *Position A*: Type P,
- *Position B*: Type N.

The proof of Zermelo's theorem sheds light on how one can find a winning strategy: the way the type of a position in a normal-play game can be found is through the use of a game tree. Within the game tree, the positions can be seen as the result after a players turn has been enacted. Through following all possible moves which can be done from the position in question, it is possible to figure out the type of that position due to the fact that a normal-play game must end, you will eventually reach then final turn which stems off of that position, granting you the type of said position.

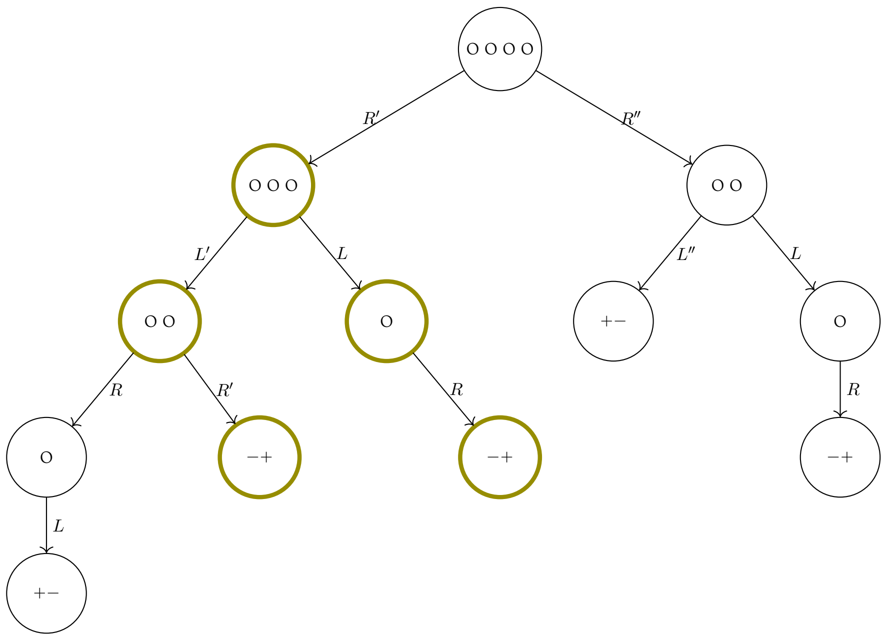

FIGURE 4. The winning strategy for Richard

As shown the in Figure 4, we see how the R' node can be labeled as type R since from this point on the player R is able to position the game in such a way to where they will always win, as shown by the olive paths.

3. Sums of Types

We can make new normal-play games through a process of putting together pieces from simpler games; we call this a sum of games, and we define this now.

Definition 3.1. Suppose we have two normal-play games, A and B, which are not necessarily the same game. Then, the *sum of games* $A + B$ is defined to be the game with

(1) the positions being the union of the positions of A and positions of B, each component inheriting the move-rule,

(2) in each turn, the player can choose to play in one of the components, but not both, and

(3) the game ends when there are no more bricks left.

Notice that by definition, the sum is also a normal-play game.

Example 3.2. Consider the sum of the positions in Figure 5. One possible play on this sum is as

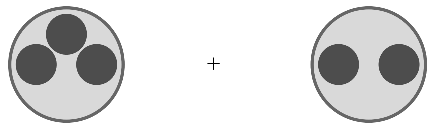

FIGURE 5. Sum of position A and B.

follows: Richard goes first and chooses to play in A, taking away 2 bricks and leaving 1, then Louise goes in B and takes 1 brick. Then Richard takes 1 bricks from A, Louise takes the remaining brick from B and wins the game.

As stated previously, every position in a normal-play game has its own type. Due to each position having its own type, this means that positions can be added together in order to find a result of said positions, this is known as Sums of Types. The reason for the name sums of types is due to the fact that when two types are added together, their result, also known as their sum will be effected by said types. We will denote the type as the sums of games as the sum of their types, i.e. we will denote the type of the sum $A + B$ as $P + N$. But, as we will see later, this is not a real sum, as it is not well-defined.

Here are a few results about the types of sums of games.

Proposition 3.3. *For normal-play games,*

$$L + L = L$$

and

$$R + R = R.$$

Proof. To view these equation under the lens of a normal-play game, if for example, player L, has a winning strategy within both positions of a normal-play game, then to follow both of those positions within a game tree will lead to a result where L will be left as the winner since both positions have been tampered in such a way where they lead to this result, this explanation applies for player R as well. □

The type P for Previous/Second is acts differently in comparison to the other types: the results of type P match that of the type which it is added with. In the perspective of a normal-play game, the reason for why type P behaves this way is because if two players are playing normal-play games, the way to win is to be the last player which makes a move in said game.

Proposition 3.4. *Suppose a normal-play game C has type X and another normal play game D has type P. Then, the type of $C + D$ is*

$$X + P = P$$

.

Proof. Referring back to Figure 5, when added together, the sum of positions A and B will be of type N. This is because to win in type N, you must be the first player to make a move in that position. The first player to move will eventually win due to having the last turn in said position. Due to making the last move in position B, their next move must be done in position A since it is the only position left. Since they just had their turn, this means that for position A they must move second. As we know, type P means the second player to make a move wins, therefore, the winner of type N will also be the winner of type P due to having to play second in a position in which the second player has a winning strategy. This scenario cooperates similarly for all other types when added with type P. □

We can summarize our findings in the following table.

+	L	R	N	P
L	L	?	?	L
R	?	R	?	R
N	?	?	?	N
P	L	R	N	P

FIGURE 6. Table for types of sums of games.

The table shown above displays all possible results which each type can have a sum of, we will be using this table as a guide which will aid in visualizing interactions between types of different positions. With regards towards "?", we will see that they are ill-defined in the next section.

4. SPACES IN TYPE TABLE WHICH ARE NOT WELL-DEFINED

Now that we've gone through types, it's time to address the "?" chart spaces. These are an indicator that, by knowing the two initial types, we cannot determine the type of the sum of said games. In order to prove this, we will be introduced to a new game called Domineering.

Definition 4.1. *Domineering* is a normal-play game with a set of positions consisting of multiple rectangular arrays fused together. Whoever is the last to remove a 2×1 or 1×2 domino (over two unoccupied unit squares of the figure) wins. Richard can eliminate 1×2 dominoes while Louise can only remove 2×1 dominoes.

Example 4.2. The following Domineering game is type N. This is because if Richard goes first, Louise cannot remove a vertical domino, and vice-versa. Hence the first player wins the game.

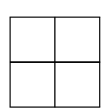

FIGURE 7. A type N game of Dominneering.

Now we will tackle the question of ill-definedness.

Example 4.3. An example of where a “?” is on our chart includes the sum of a type N and type L game of Domineering.

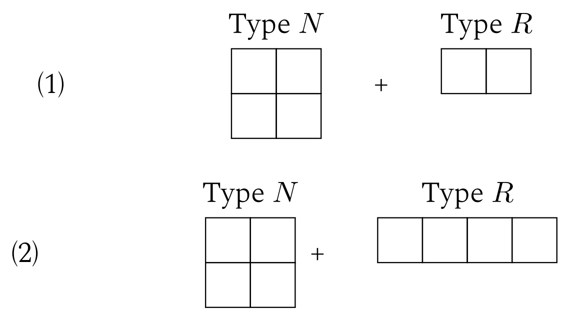

FIGURE 8. Two sums of games which look the same on the level of types.

In the first case, if Richard were to go first in the 2×2 figure, Louise would automatically be left without any more moves $(-+)$ while if he were to move first in the 1×2 figure, Louise would move in the 2×2 figure and leave Richard with no other moves $(+-)$; these results indicate that the sum of the given type N and R games are type N.

The second case is another example of the sums of type N and R games. If Richard were to move first in this case, he could move in the 2×2 position and leave Louise with no other moves $(-+)$. Richard could also move first in the 1×4 position going first as Louise would have no other option but to move to the position of type N. Despite such, with these two extra units in the type R position, Richard would still be able to take the last move $(-+)$. The only position Louise could play if she were to go first would be in the 2×2 type N position whereas Richard’s only option afterward would be the right-hand position. Afterward, Louise’s final possible move would be the remaining 1×2 position from the type N game although, once again due to the two additional units in the right-hand position, Richard will be able to take the final move and win $(-+)$. These results are able to prove that this sum between a type N and type R game is equal to type R.

For clarity:

(1) $(2 \times 2) + (1 \times 2)$: Type N + Type R = Type N
(2) $(2 \times 2) + (1 \times 4)$: Type N + Type R = Type R

As we know, type N describes the winning strategy of the next player while type R does equally for type R, though Richard is not always the next player, therefore, indicating that Type N + Type R; or the sums of type N and type R games are not always equivalent to one another. Knowing that multiple types can be produced from a sum between the same game types, “?” is a practical way to identify the fact that the types of game positions are not always enough to determine the types of their true sums.

5. Acknowledgement

This section marks the end of the document. The source used for the information gathered was the book *Game Theory, A Playful Introduction*[1]. All of this work was possible due to program coordinators Marisa Gaetz and Mary Stelow of MIT PRIMES Circle, our Game Theory mentor Yuyuan Luo, as well as everyone else involved in the program. Danielle would like to extend her thanks to Ms. Byrne for supporting her mathematical journey.

References

[1] Matt DeVos and Deborah Kent. *Game theory: a playful introduction*, volume 80. American Mathematical Society, 2016.

Email address: abaez50@bostonk12.org

Email address: daniilanaa00@gmail.com