EVALUATING HACKENBUSH

AGASTYA GOEL

1. Introduction to Hackenbush

The game of *hackenbush* is played by two players, which we will call Left and Right, on a collection of nodes, some of which may be touching the ground, and a collection of edges, which connect two nodes. The game proceeds in turns, with players alternating play. On one's turn, they may remove an edge of the appropriate color; Left may remove blue or green edge, and Right may remove red or green edges. After removing an edge, any 'floating' edges–edges which do not have a path to the ground–are removed. The player who is unable to make a move loses.

As with other combinatorial games, there are two important operations we can do to combine hackenbush games. The first is game *addition*, which we will refer to with the '+' symbol. If we have hackenbush games G and H, the game $G + H$ is the game formed by placing G and H on the floor separately, as demonstrated in Figure 2. The other is the *ordinal sum*, in which we place one game 'on top' of the other. Formally, if we have games G and H, $G : H$ represents the game such that if we play in H to an option H', we get the game $G : H'$, and if we play in G to an option G', H disappears and we get G'. We will discuss the ordinal sum in greater detail when we discuss simplifying hackenbush positions.

Red-blue hackenbush, where all edges are red or blue, is a fairly well-understood game with respect to addition. Specifically, we can assign each game a number value, where positive values are good for Left

Date: December 7, 2024.

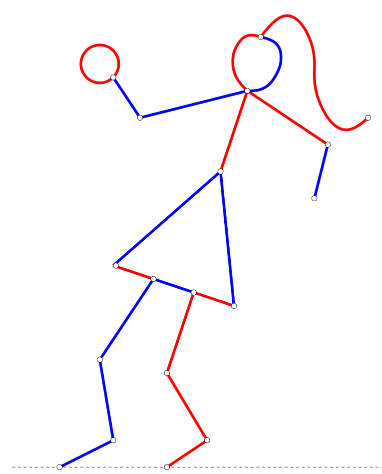

Figure 1. A red-blue hackenbush position.

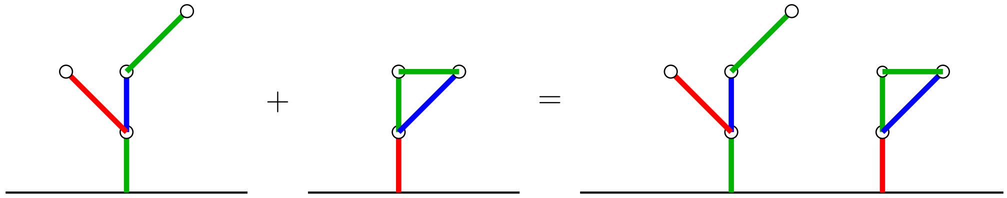

Figure 2. Hackenbush addition.

and negative values are good for Right, and when we take the game $G + H$, the corresponding number is the number of G plus hte number of H. In this expository paper, we will focus on the 'green' aspect of hackenbush, starting with games of only green edges, then discussing games with green edges as the base, and finally discussing how the green jungle slides down the purple mountain.

2. Green Hackenbush

A game of *green hackenbush* is a hackenbush game with no red or blue edges. Since such games are impartial and perfect information, their values can be described with nimbers. In this section, we will discuss how to calculate the nimber value of any green hackenbush position.

Definition 2.1. We say a game of green hackenbush is a *forest* if each node has a unique sequence of edges connecting it to the ground. We say a game is a *tree* if it is a forest with only one edge touching the ground.

Lemma 2.2. *Consider a green hackenbush forest G. If G has value $*n$, the game $1 : G$ has value $*(n + 1)$.*

Proof. We induct on the birthday of G. Now, we find the options of $1 : G$. If we play in 1, we get the game 0, which has value 0. If we play in G to G', we get a nimber value of $*(m + 1)$ where $*m$ is the nimber value of G' by induction. Let $*n$ be the nimber value of G. Then, this means that the options of G' contain all nimbers from 0 to $*(n - 1)$ but not $*n$ itself. Hence, the set of nimber values of $1 : G'$ contains nimbers $*$ through $*n$ but not $*(n + 1)$. When combined with the option of value 0 obtained by playing in 1, we find that $1 : G$ contains options with values 0 through $*n$ but not $*(n + 1)$, so its nimber value is $*(n + 1)$ by the MEX rule. □

Lemma 2.2 gives a natural way to compute the value of a green hackenbush forest via recursion. Namely, we can write the forest as the sum of individual trees. To calculate the value of a tree, we can write it as $1 : G$ for some game G, calculate the value of G recursively, and determine the value of the tree. We can then determine the value of the forest by taking the XOR of all of the tree values. Applying these principles also yields the following result.

Lemma 2.3 (Parity Principle)**.** *The parity of the value of a green forest G equals the parity of the number of edges.*

Proof. We can prove this result via induction on birthdays. Consider a green forest. If the forest contains multiple trees, we can invoke the inductive hypothesis for each tree. Then, since the parity of $a + b$ equals the parity of $a \oplus b$, we are done by the XOR rule.

Considering a single tree $G = 1 : H$, the parity of the value of G is opposite that of H by Lemma 2.2 and the number of edges is opposite as well, concluding the proof. □

We may also compute the value of a green hackenbush forest using the *colon principle*, which states that if the value of G is the same as the value of H, then $X : G$ has the same value as $X : H$ for all games X. Hence, we can simplify a green hackenbush forest to a nimber by using the colon principle to recursively simplify upper branches until we are left with a single stalk. This approach is actually equivalent to the approach given by Lemma 2.2, except that it does not use recursion.

We will now consider non-forest graphs, which contain cycles.

Definition 2.4. A distinct sequence of nodes $a_1, a_2, \ldots, a_n$ in a game G forms a *cycle* if there are edges connecting a_i and a_{i+1} for $1 \leq i < n$ as well as an edge connecting a_1 and a_n.

Note that a forest never contains any cycles. Further, if we merge all ground nodes into one, a game is a forest iff it does not contain any cycles.

To evaluate positions with cycles, we can fuse certain nodes together to eliminate these cycles.

Definition 2.5. Consider a green hackenbush game G, as well as two nodes $x, y \in G$. Then, we may obtain a new game H by *fusing* x and y; the vertices of the new game are the vertices of $G \setminus \{x, y\}$ as well as a new vertex xy. Any edge with endpoint x or y in the original graph now has xy as that endpoint (which may lead to self-edges from xy to xy). Finally, the vertex xy is on the ground if either x or y was previously on the ground. We denote the game obtained by fusing x, y in G as G_{xy}.

Theorem 2.6 (Fusion Principle)**.** *If the vertices $a_1, a_2, \ldots, a_n$ form a cycle in G, the game H formed by fusing these vertices together has the same value as G.*

Note that a self-loop from node x to itself is equivalent to an edge from x to a new node x'. Thus, we may use the fusion principle to remove *all* cycles from a green hackenbush game, thus allowing us to compute its value using Lemma 2.2, as described previously. Whenever we invoke the fusion principle, we will also perform this conversion of loops to hanging edges implicitly.

Proof of theorem 2.6. For contradiction, consider the graph G with the fewest number of edges containing two nodes x, y which share a cycle but cannot be fused without changing the value. If there are multiple such graphs, consider one with the minimal number of nodes.

We can now show several properties G must obey:

(1) Only one node may touch the ground. If there were multiple, we could merge them together and obtain a smaller game.
(2) For every pair of nodes x, y which share a cycle, it must be the case that fusing x and y changes the value, since otherwise such a graph would have fewer nodes and the same number of edges.
(3) For every pair of nodes x, y, there can be at most two disjoint paths (paths sharing no edges) connecting them. For contradiction, let there be three or more paths. Then, let $H = G_{xy}$. We can show that $G = H$ by showing that the second player wins $G + H$. Since G and H have edges in correspondence, the second player may delete the corresponding edge in whichever game the first player does not play in. After these first two moves, x and y will still be in a cycle together, since there will still be at least two disjoint paths connecting them. Thus, since G was the minimal contradictory game, we can fuse x, y which yields two copies of the same game, which has value 0. Thus, the second player wins.
(4) All cycles must include the unique ground node. For contradiction, assume that G has a cycle not including the ground. Then, we consider two cases. First, consider when there are two distinct cycle nodes x, y which have paths to the ground not including cycle edges. In this case, let z be the first node where these paths meet, so that the paths from x to z and y to z are disjoint. Then, there are three paths from x to y: one through z and two along the two cycle directions, which is a contradiction.

    Alternatively, consider when there is only one cycle node x with a path to the ground. In this case, we may decompose the graph G into graphs A and B such that $G = A : B$ and B has a cycle. Since B has fewer edges than G, we may freely fuse all of its nodes creating B^* without changing its value, and then replace G with $G^* = A : B^*$, with, again, the same value by the colon principle.
(5) There must be exactly one cycle, or we may split the game into a sum of multiple distinct ones. If there are multiple cycles that cannot be split, they must be connected in some way, which would create two nodes with three disjoint paths between them: a contradiction.
(6) Any branches hanging off of this single cycle must be stalks. Otherwise, we could simplify them into a single nimber stalk via the colon principle.

Combining all of these properties, G must consist of a single cycle attached to the ground, with a single stalk, or 'string' hanging off each non-ground node. To prove the fusion principle, we now divide into cases based on the number of edges in the cycle. If the number of edges in the cycle is even, the game given by applying the fusion principle is just the collection of strings. Call the game formed from these strings H. Then, we wish to show that the second player wins $G + H$. If the first player plays in either of the strings, the second player can match their move. Then, since G will be reduced, we can apply the fusion principle and conclude that the value is 0. If the first player moves in the cycle, they leave a green forest with an odd total number of edges. Thus, it must have a nonzero value by the parity principle, and the second player wins once again.

We may apply a similar argument if the number of edges in the cycle is odd. This time, we let the game H be the collection of strings with an additional single stalk. If the first player cuts a string, the second player can cut the corresponding string in the other of G, H and apply the fusion principle to win. If the first player cuts the cycle in G, they leave an odd number of edges in a green forest and the second player wins. □

There is actually one additional case, where the first player cuts down the lone stalk. It turns out that we may always cut a cycle edge playing second to win. However, this part of the proof is beyond the scope of this paper.

3. Hackenbush Gardens

We now turn our attention to *Red-Green-Blue*, or *RGB*, games of hackenbush which may include all three colors. We call connected subparts of this game with only green edges *greenery*.

The first type of RGB hackenbush game we will consider is a *flower*, consisting of a tall green stalk, potentially with red and blue edges on top. Note that, by the colon principle, we can replace a flower topped with r red edges and b blue edges with one topped by $r-b$ red or $b-r$ blue edges, whichever is nonnegative. Thus, we may consider a flower to be red, blue, or green depending on whether it is topped by a game with negative, positive, or zero value. A sum of multiple flowers is called a *flower garden*.

Lemma 3.1. *If Left has a blue flower and Red has none, then Left wins playing first. Similarly, if only Red has a flower, then they win playing first.*

Proof. We consider the case where Left has a flower, as the other case is identical. Ignore the blue edges, let the game have a value of $*n$. If $n > 0$, Left can pretend the blue edges do not exist and win the underlying game. Otherwise, Left can remove a blue edge, 'passing,' and giving Right a game with value 0. ☐

Corollary 3.2. *If either player has at least two more flowers of their color than the other, they win.*

Corollary 3.2 motivates us to introduce the *atomic weight*, which is just the number of blue flowers minus the number of red flowers. A precise, general definition of the atomic weight is beyond the scope of this paper, but in general, an atomic weight of n indicates that a game is equivalent to n blue flowers (or $-n$ red ones), modulo some greenery. We denote the atomic weight of G with $\mathrm{aw}(G)$. The atomic weight also satisfies $\mathrm{aw}(G+H) = \mathrm{aw}(G)+\mathrm{aw}(H)$. With the atomic weight, Corollary 3.2 becomes G is winning for Left if $\mathrm{aw}(G) \geq 2$, and winning for Right if $\mathrm{aw}(G) \leq -2$.

We can also calculate the atomic weight of a more general type of game, a *parted jungle* is a hackenbush game where no red edges touch blue edges, neither red nor blue edges touch the ground, and all green edges are connected to the ground using only green edges. For convenience, we will merge all nodes on the ground into one.

Definition 3.3. A *cut* between two sets of nodes S, T is a set of edges such that all paths between nodes in S and T pass through one of these edges. A *minimum cut* is a cut with the minimum number of edges.

Definition 3.4. A *flow* between two sets of nodes S, T is a collection of disjoint paths between the two. A *maximum flow* is a flow with the maximum number of disjoint edges.

There exists a famous connection between maximum flows and minimum cuts:

Theorem 3.5 (Min-Cut-Max-Flow)**.** *The size of the minimum cut is equal to the maximum flow.*

This theorem is a special case of *strong duality*, a discussion of which is beyond the scope of this book. We will, however, take a brief detour to discuss how to compute the maximum flow.

To this end, we introduce the *residual graph* V, generated from a graph U and an associated flow. Namely, the residual graph is the graph containing edges of U not in the flow, as well as edges in the flow only traversable in the opposite direction as they appear in the flow. The residual graph is important since it encodes information about the flow and associated cuts.

Lemma 3.6. *A flow is maximum if the associated residual graph contains no path from S to T.*

The proof of this fact is beyond the scope of this paper, but by repeatedly finding paths in the residual graph and adding them to our flow, we can extend it to a maximum flow.

Now, we will let the blue cluster B be the set of all nodes that are the endpoint of a blue edge, and the red cluster R be the set of all nodes that are the endpoint of a red edge. Note that no node can be in both sets. With this notation, we can compute the atomic weight.

Theorem 3.7. *Consider a parted garden G. Then, let the size of the maximum flow from B to R be m. Now, consider the size $n+m$ of the maximum flow from B to R or to the ground with the condition that the flow between B and R is still m. If $m > 0$, the atomic weight of G is m. Otherwise, consider the size $n+m$ of the maximum flow from R to B or the ground such that the flow between R and B is still m. Then, the atomic weight is $-m$.*

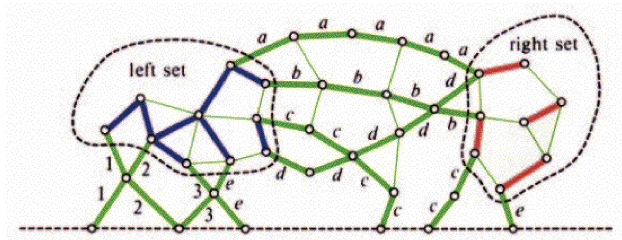

FIGURE 3. By Theorem 3.7, this parted jungle has atomic weight 3.

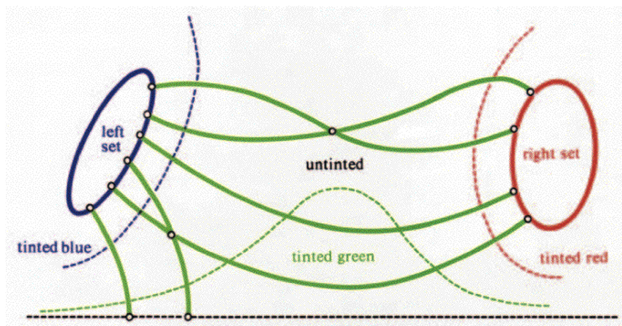

FIGURE 4. An example of how to tint nodes in a parted jungle.

Note that it cannot be the case that there exists a flow from B to R or the ground of size $n+m > n$ and a flow from R to B or the ground with the same property, since by appropriately merging the two flows, we could obtain a larger flow from B to R. Thus, assume WLOG that there is no larger flow from R to B or the ground, and the largest flow from B to R or the ground (with a flow of m from B to R) is of size $n+m$.

We now introduce a way to tint nodes in a parted jungle. As a rule of thumb, a node is tinted a certain color if you can reach that node from that color by only moving on unused green edges or backwards along green edges.

Definition 3.8. We define the *tint* of a node in a parted jungle as follows:

- A node is *tinted blue* if it can be reached from B by only traversing unused green edges or traversing used green edges in the direction opposing their flow.
- A node is *tinted red* if it can be reached from R by only traversing unused green edges or traversing used green edges along the direction of their flow.
- A node is *tinted green* if it can be reached from the ground by only traversing unused green edges or traversing edges carrying flow from B to the ground in the direction of their flow.

Note that not all nodes must be tinted.

We will also the green edges of our parted jungle.

Definition 3.9. We *shade* our edges as follows:

- An edge is *shaded blue* if it is on a path containing a green node and carries flow out of a blue tinted node.
- An edge is *shaded red* if it is on a path containing a green node and carries flow into a red tinted node.

All other edges are shaded green.

Lemma 3.10. *There are m more edges shaded blue than those shaded red.*

Proof. Consider the flow as a set of disjoint paths. Each path from B to the ground will consist of exactly one edge shaded green: the edge closest to B which ends at a node tinted green. On the other hand, each path from B to R will consist of either no edges shaded red or blue or exactly one edge of each type. Thus, in total, there are m more edges shaded blue than red. ☐

We are now ready to prove Theorem 3.7.

Proof of Theorem 3.7. Left's strategy will be to start by cutting down all edges shaded red. After this, there will be at least m edges shaded blue. At this point, Left's position is equivalent to one with atomic weight m. Specifically, since all red nodes cannot reach the ground without passing through a node tinted blue, the game consisting of all nodes not tinted green must have a positive value, as Left may play to win by removing all green edges and then all blue ones, after which Right will have no moves. Since the game atop the edges shaded blue is positive, Right must chop down all m edges shaded blue, giving the parted jungle an atomic weight of m. $\square$