Set theory and logic throughout mathematics

Chris Lambie-Hanson

February 18, 2024

Contents

Chapter 1

Lecture 1: Ordinals and the hydra

1.1 Well-orders

Let us begin by briefly reviewing the definition of *partial order*, *linear order*, and *well-order*.

Definition 1.1.1. Suppose that X is a set and $\leq$ is a binary relation on X. Then $\leq$ is a *partial order* on X if it is

1. *Reflexive*: $x \leq x$ for all $x \in X$;

2. *Transitive*: for all $x, y, z \in X$, if $x \leq y$ and $y \leq z$, then $x \leq z$; and

3. *Anti-symmetric*: for all $x, y \in X$, if $x \leq y$ and $y \leq x$, then $x = y$.

A partial order $\leq$ on a set X is a *linear order* if, in addition, it is *total*, i.e., for all $x, y \in X$, we have $x \leq y$ or $y \leq x$.

If $\leq$ is a partial order on a set X and $x, y \in X$, then we will write $x < y$ to mean that $x \leq y$ and $x \neq y$. The relation $<$ is then referred to as the *strict* part of $\leq$.

Definition 1.1.2. Suppose that X is a set and R is a binary relation on X. Then R is *well-founded* if every nonempty subset of X has an R-minimal element. In other words, for every nonempty $Y \subseteq X$, there is $y \in Y$ such that, for all $x \in Y$ with $x \neq y$, $\neg(xRy)$. A well-founded linear order is called a *well order*.

Exercise 1.1.3. Suppose that $\leq$ is a linear order on a set X. Prove that the following are equivalent.

1. $\leq$ is a well order.

2. There are no infinite, strictly decreasing sequences with respect to $\leq$. In other words, there does not exist a sequence $\langle x_0, x_1, x_2, \ldots \rangle$ of elements of X such that, for all n, we have $x_{n+1} < x_n$.

There is a natural way to assert that two partial orders are "essentially" the same, i.e., *isomorphic*.

Definition 1.1.4. Suppose that $\leq_0$ is a partial order on X_0 and $\leq_1$ is a partial order on X_1. Then we say that $\leq_0$ and $\leq_1$ are *isomorphic* if there is a bijection $F : X_0 \to X_1$ such that, for all $x, y \in X_0$, we have

$$x \leq_0 y \iff F(x) \leq_1 F(y).$$

Example 1.1.5. The following are some examples and non-examples of isomorphic partial orders.

1. The open interval $(0, 1)$ and the open interval $(0, 2)$, both with the usual ordering of real numbers, are isomorphic via the bijection $x \mapsto 2x$.
2. The open interval $(0, 1)$ and the closed interval $[0, 1]$ are not isomorphic. One way to see this is to note that $[0, 1]$ has a maximal element and $(0, 1)$ does not, so any order-preserving map from $(0, 1)$ to $[0, 1]$ could not include 1 in its range.
3. Let Y be any nonempty set. Let $X_0 = \mathscr{P}(Y)$ be the *power set* of Y, i.e., the collection of all subsets of Y. Let $\leq_0$ be the partial order on X_0 defined by letting $u \leq v$ if and only if $u \leq v$.

   Let X_1 be the collection of all functions $f : Y \to \{0, 1\}$, and let $\leq_1$ be the partial order on X_1 defined by letting $f \leq g$ if and only if $f(y) \leq g(y)$ for all $y \in Y$.

   Then $\leq_0$ and $\leq_1$ are isomorphic via the bijection $F : X_0 \to X_1$ that sends each $u \in X_0$ to the *characteristic function* of u, i.e., the function $f_u : Y \to \{0, 1\}$ that takes value 1 on all elements in u and value 0 on all elements of Y that are not in u.

1.2 Ordinal numbers

Roughly speaking, an ordinal number can be thought of as a description of the order type of a well-order. In other words, to each well-order, we assign an ordinal, and two well-orders are isomorphic if and only if they are assigned the same ordinal.

Example 1.2.1. For each natural number n, all well-orders of size n are isomorphic; their order type is itself referred to as "n".

However, there are many non-isomorphic countably infinite well-orders. The ordinal describing the order type of the natural numbers,

$$0 < 1 < 2 < 3 < \ldots$$

is denoted "ω". But we can form a new order type by adding a new element (call it ∞) that is larger than all of the natural numbers:

$$0 < 1 < 2 < 3 < \ldots < \infty.$$

The ordinal describing this order type is denoted "$\omega + 1$". Or we can form yet another order type by placing two copies of the natural numbers one after the other:

$$0 < 1 < 2 < 3 < \ldots < 0' < 1' < 2' < 3' < \ldots.$$

The ordinal describing this order type is denoted "$\omega + \omega$".

There is a natural way to order the ordinal numbers themselves. To make this precise, we need the following definition.

Definition 1.2.2. Suppose that $\leq$ is a well-order of a set X. Then an *initial segment* of $(X, \leq)$ is a subset $Y \subseteq X$ such that, for all $y \in Y$ and all $x \in X$, if $x \leq y$, then $x \in Y$. In other words, if $y \in Y$, then Y also contains all elements of X that are smaller than y in the ordering $\leq$.

Exercise 1.2.3. Suppose that $\leq$ is a well-order of a set X and Y is an initial segment of $(X, \leq)$. Then either

- $Y = X$; or
- there is $x \in X$ such that $Y = \{y \in X \mid y < x\}$.

Exercise 1.2.4. Suppose that $(X_0, \leq_0)$ and $(X_1, \leq_1)$ are two well-orders. Then either

1. $(X_0, \leq_0)$ is isomorphic to an initial segment of $(X_1, \leq_1)$; or
2. $(X_1, \leq_1)$ is isomorphic to an initial segment of $(X_0, \leq_0)$.

If both 1. and 2. hold, then $(X_0, \leq_0)$ and $(X_1, \leq_1)$ are isomorphic.

With Exercise 1.2.4 in mind, we can make the following definition.

Definition 1.2.5. Suppose that α and β are ordinals. Then we say that $\alpha \leq_{\text{ord}} \beta$ if, whenever $(X_\alpha, \leq_\alpha)$ is a well-order of type α and $(X_\beta, \leq_\beta)$ is a well-order of type β, then $(X_\alpha, \leq_\alpha)$ is isomorphic to an initial segment of $(X_\beta, \leq_\beta)$.

Exercise 1.2.6. The class of ordinals is well-ordered by $\leq_{\text{ord}}$.

One can perform arithmetic on ordinal numbers. We will make this more precise later, but let us first give an informal description. Let α and β be ordinal numbers, and let $(X_\alpha, \leq_\alpha)$ and $(X_\beta, \leq_\beta)$ be well-orders of type α and β, respectively.

We first describe ordinal addition. The ordinal $\alpha+\beta$ is the ordinal describing the well-ordering formed by placing a copy of $(X_\beta, \leq_\beta)$ after a copy of $(X_\alpha, \leq_\alpha)$ (i.e., every element of X_β is declared to be larger than every element of X_α).

Note that ordinal addition is not commutative: it may not be the case that $\alpha + \beta = \beta + \alpha$. To see this, consider $2 + \omega$ and $\omega + 2$. Represent the ordinal 2 by the order $0' < 1'$, and represent ω by the usual natural numbers. Then $2 + \omega$ is the order type of the order

$$0' < 1' < 0 < 1 < 2 < 3 < \dots$$

This is isomorphic to the usual ordering of the natural numbers, via the map

$$\begin{aligned} 0' &\mapsto 0 \\ 1' &\mapsto 1 \\ n &\mapsto n + 2 \text{ for every natural number } n. \end{aligned}$$

Thus, $2 + \omega = \omega$. On the other hand, $\omega + 2$ is the order type of the order

$$0 < 1 < 2 < \ldots < 0' < 1'$$

This is clearly *not* isomoprhic to the natural numbers; for example, it has a maximal element, whereas that natural numbers do not. Thus, $\omega + 2 \neq \omega$, and in fact $\omega <_{\mathrm{ord}} \omega + 2$.

We next describe ordinal multiplication. The ordinal $\alpha \cdot \beta$ is the ordinal describing the well-ordering formed by starting with a copy of $(X_\beta, \leq_\beta)$ and replacing every element of X_β with a copy of $(X_\alpha, \leq_\alpha)$.

Again, ordinal multiplication is not commutative. For example, $2 \cdot \omega$ is the order type of the following order:

$$0 < 0' < 1 < 1' < 2 < 2' < 3 < 3' < \ldots$$

formed by replacing every natural number n with a copy $n < n'$ of the two-element order. It is not too hard to show that this order is isomorphic to the natural numbers, so $2 \cdot \omega = \omega$. On the other hand, $\omega \cdot 2$ is the order type of the following order:

$$0 < 1 < 2 < 3 < \ldots < 0' < 1' < 2' < 3' < \ldots$$

formed by replacing each element of the two-element order $* < *'$ by a copy of the natural numbers. This is not isomorphic to the set of natural numbers; for instance, it contains elements that are larger than infinitely many other elements, whereas the natural numbers do not. Thus, $\omega \cdot 2 \neq \omega$, and in fact $\omega <_{\mathrm{ord}} \omega \cdot 2$.

We finally describe ordinal exponentiation. If $\alpha = 0$, then $\alpha^\beta = 0$. Otherwise, first let 0_α denote the *minimal* element of $(X_\alpha, \leq_\alpha)$. This must exist, since $\leq_\alpha$ is a well-order. We say that a function $f : X_\beta \to X_\alpha$ is *finitely supported* if the set $\{y \in X_\beta \mid f(y) \neq 0_\alpha\}$ is finite. The ordinal α^β is now defined as follows. Let Z be the set of all finitely-supported functions from X_β to X_α. Now describe an ordering $\preceq$ on Z as follows. Given $f, g \in Z$, set $f \preceq g$ if and only if either

- $f = g$; or
- $f \neq g$ and, letting $y \in X_\beta$ be the $\leq_\beta$-maximal element such that $f(y) \neq g(y)$, we have $f(y) \leq_\alpha g(y)$.

Then let α^β be the ordinal describing the order type of $(Z, \preceq)$.

Exercise 1.2.7. Prove that the order $(Z, \preceq)$ described in the preceding paragraph is indeed a well-order.

A concrete representation of the ordinals.

In practice we often work with a particular concrete realization of the ordinals, and we think of an ordinal α as the set of all ordinals that are strictly less than α (with respect to the ordering $\leq_{\mathrm{ord}}$ introduced above. At first glance, this may appear like a circular definition, but it is not, due to the fact that $\leq_{\mathrm{ord}}$ is itself a well-ordering. In particular, there is a least ordinal, 0. Since there are no ordinals strictly less than 0, we represent 0 as the empty set, $\emptyset$. The next smallest ordinal is 1. It only has one ordinal less than it, namely, 0, so 1 is represented as $\{0\} = \{\emptyset\}$. The first few ordinals are thus represented as follows:

$$
\begin{aligned}
0 &= \emptyset \\
1 &= \{0\} = \{\emptyset\} \\
2 &= \{0,1\} = \{\emptyset, \{\emptyset\}\} \\
3 &= \{0,1,2\} = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} \\
4 &= \{0,1,2,3\} = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}\} \\
&\dots \\
\omega &= \{0,1,2,3,4,\dots\} \\
\omega + 1 &= \omega \cup \{\omega\} = \{0,1,2,3,4,\dots\} \cup \{\omega\} \\
\omega + 2 &= \omega \cup \{\omega, \omega+1\} \\
&\dots \\
\omega + \omega &= \omega \cdot 2 = \{0,1,2,3,4,\dots\} \cup \{\omega, \omega+1, \omega+2, \omega+3, \omega+4, \dots\}
\end{aligned}
$$

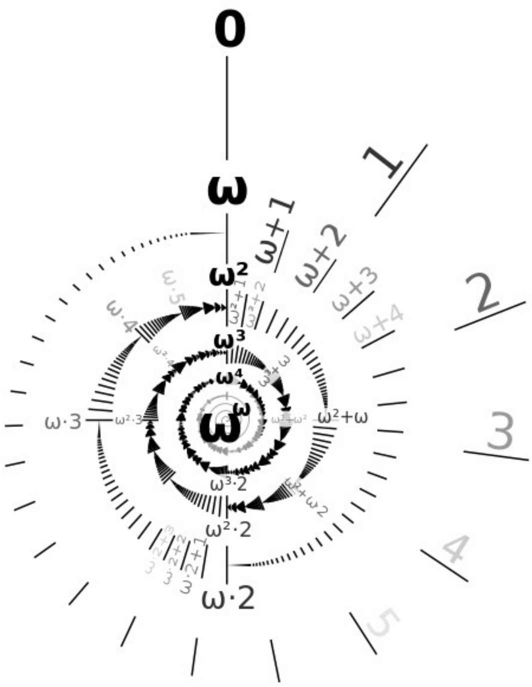

Figure 1.1: A stylized image of the ordinals up to ω^ω

With this concrete representation of the ordinals, we can easily be more precise about ordinal arithmetic. We first introduce the following notions.

Definition 1.2.8. Let X be a nonempty set of ordinals. Then the *supremum* of X, denoted $\sup(X)$, is the least ordinal that is greater than or equal to every element of X.

Exercise 1.2.9. Working with our concrete representation of the ordinals, prove that, for every nonempty set of ordinals X, the supremum of X is equal to the union of all of the elements of X, i.e.,

$$\sup(X) = \bigcup X.$$

Definition 1.2.10. Suppose that β is an ordinal.

1. We say that β is a *successor* ordinal if $\beta = \alpha + 1$ for some ordinal α.
2. If β is not a successor ordinal, we say that β is a *limit* ordinal.

We can now rigorously define ordinal arithmetic by recursion. We first deal with addition. For all ordinals α, we let:

- $\alpha + 0 = \alpha$;
- $\alpha + 1 = \alpha \cup \{\alpha\}$;
- for all ordinals β, we have $\alpha + (\beta + 1) = (\alpha + \beta) + 1$;
- if γ is a nonzero limit ordinal, then $\alpha + \gamma = \sup\{\alpha + \beta \mid \beta < \gamma\}$.

Next, multiplication:

- $\alpha \cdot 0 = 0$;
- $\alpha \cdot 1 = \alpha$;
- for all ordinals β, we have $\alpha \cdot (\beta + 1) = (\alpha \cdot \beta) + \alpha$;
- if γ is a nonzero limit ordinal, then $\alpha \cdot \gamma = \sup\{\alpha \cdot \beta \mid \beta < \gamma\}$.

Finally, exponentiation:

- $\alpha^0 = 1$;
- $\alpha^1 = \alpha$;
- for all ordinals β, we have $\alpha^{\beta+1} = (\alpha^\beta) \cdot \alpha$;
- if γ is a nonzero limit ordinal, then $\alpha^\gamma = \sup\{\alpha^\beta \mid \beta < \gamma\}$.

1.3 The hydra

We end this first lecture with a surprising demonstration of the utility of infinite ordinals: the hydra game. You may be familiar with the Hydra from Greek mythology. It is a fearsome water monster with many heads with the property that, whenever you chop off one of its heads, two heads will grow back in its place. Eventually, the hydra was slain by Heracles, with the assistance of his nephew Iolaus.

We will be examining a game played using a mathematical version of the Hydra introduced in the paper "Accessible independence results for Peano Arithmetic" by Laurie Kirby and Jeff Paris. For us, a *hydra* is a finite tree with a root. In other words, a hydra consists of finitely many nodes and edges. There is a root node at the bottom, which has finitely many edges coming out of it, each leading to another node. In turn, each of these nodes has finitely many edges coming out of it, each leading to a further node, and so on. For example, this is a hydra:

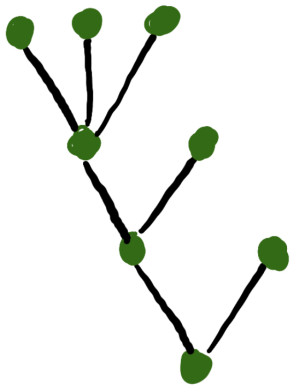

Figure 1.2: A hydra

We will always draw hydras with the root at the bottom. A *terminal node* of a hydra is a non-root node that is connected to only one other node. A *head* of a hydra consists of a terminal node and the single edge that leads to it. For example, the hydra pictured above has five heads:

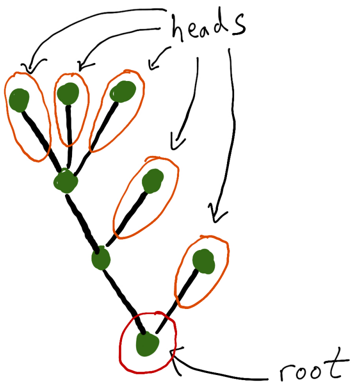

Figure 1.3: A hydra with its root and heads labeled

Given a head, the single node that it is attached to is called its *parent*. If its parent is not the root, then the node that is one step closer to the root from its parent is called its *grandparent*:

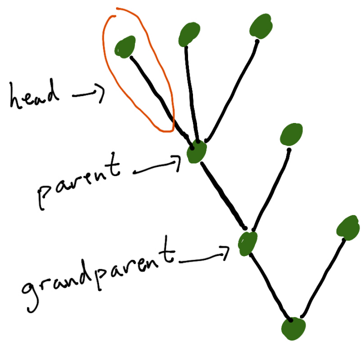

Figure 1.4: A hydra with a labeled head, its parent, and its grandparent

In the hydra game, we start with a hydra and, on each move (starting with Move 1), we chop off one of its heads. Our goal is to reduce the hydra to only a root node in a finite number of moves. However, like its mythological counterpart, the hydra regenerates, according to the following rules:

- If, on Move n, we chop off a head directly connected to the root, then the hydra does not create any new heads.

- If, on Move n, we chop off a head *not* directly connected to the root, then first delete the node and edge that make up that head. Then, move down one edge towards the root, to the edge connecting the parent and the grandparent of the head that was removed. The hydra makes n new copies of the subtree consisting of this edge and everything above it and attaches each of these new copies to the grandparent of the head that was removed.

This is best illustrated with a picture. Suppose that we are about to make Move 2 of a game, and we are confronted with the hydra pictured above. One option is to chop off the head in the bottom right, directly connected to the root:

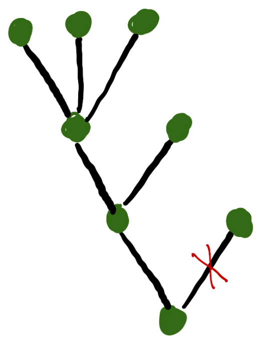

Figure 1.5: Chopping off a head directly connected to the root

Since this head is directly connected to the root, the hydra does not generate any new heads, so on our next move we see the following hydra:

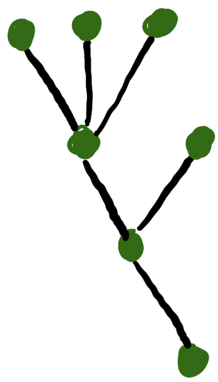

Figure 1.6: The result of the move in Figure 1.5

However, we could have done something different on Move 2 and instead chopped off the head on the upper left:

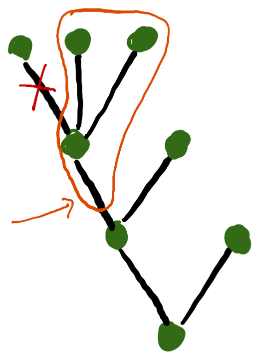

Figure 1.7: Chopping off a head not directly connected to the root

Now, to generate the hydra for the next move, we first remove the head. Then we consider the subtree consisting of the edge between the head's parent and grandparent and everything above it (circled in orange in Figure 1.6). Since we are on Move 2, we make 2 new copies of it and attach them to the grandparent of the removed head, resulting in the following hydra:

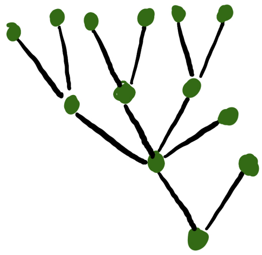

Figure 1.8: The result of making the move in Figure 1.7 on Move 2

Let us see now a complete play of the game, starting from a very simple hydra:

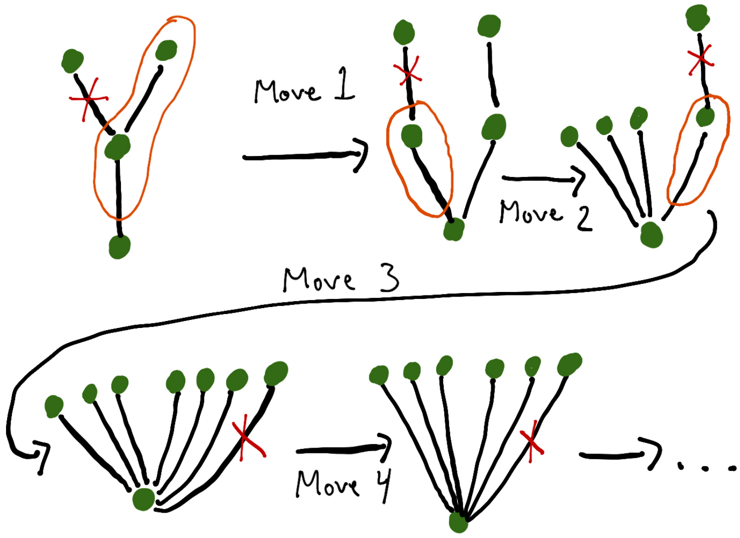

Figure 1.9: A play of the hydra game

We begin with a simple hydra with two heads. In Move 1, we chop off the left head. The head is not connected to the root, so we make one new copy of the circled region and connect it to the head's grandparent (which in this case is the root). The hydra we are left with still has two heads. In Move 2, we chop off the left head. Again, this is not connected to the root, so we make two new copies of the circled region and connect them them to the head's grandparent. In Move 3, we chop off the right head (the only one left that is not directly connected to the root). We make three new copies of the circled region and connect them to the head's grandparent. We are then left with a hydra that has seven heads, but each of them is directly connected to the root. We can thus chop them off one at a time, winning the game after seven more moves.

We thus won this round of the hydra game, but maybe that is only because we started with a very simple hydra. Consider the following diagram, taken from the paper by Kirby and Paris in which the hydra game was introduced, depicting the first three moves in a hydra game starting from a more complicated hydra:

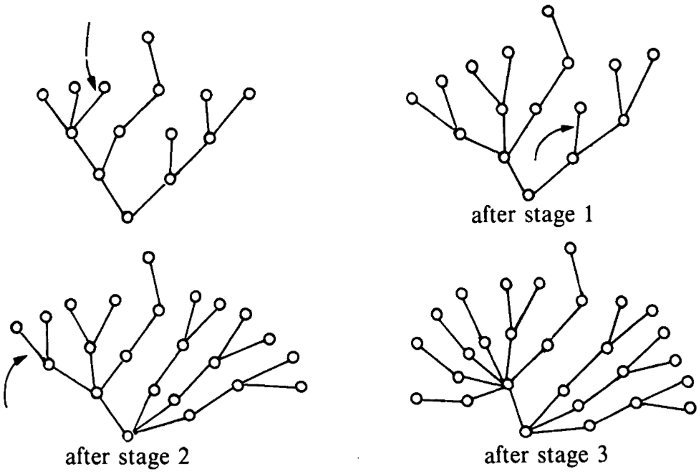

Figure 1.10: The first three moves of a more complicated hydra game

Here, the hydra we end up with after three moves looks, to the untrained eye, to be significantly larger and more complicated than the one we started with, and it seems conceivable that we will never win this hydra game. However, we will prove the following somewhat surprising theorem, showing that not only can we win *every* hydra game, but in fact we *cannot lose.*

Theorem 1.3.1. *In every hydra game, no matter how we play, we will always win after a finite number of moves.*

Proof. We will denote runs of the hydra game by $\langle H_0, H_1, H_2, \ldots \rangle$, where H_0 is the initial hydra that starts the game, H_1 is the hydra resulting after Move 1, H_2 is the hydra resulting after Move 2, and, in general, H_k is the hydra resulting after Move k. If we ever reach an n such that H_n is the hydra consisting of just a root node, then we have won the game, so the complete run of the game is then $\langle H_0, H_1, H_2, \ldots, H_n \rangle$. We must show that every possible run of the hydra game is finite.

To do this, given an arbitrary hydra H, we will assign it an ordinal number, $\#(H)$ in the following way. Starting with the terminal nodes and working our way down to the root, we will assign an ordinal number to each node of the hydra. Each terminal node gets labeled with a 0. Now suppose that u is a non-terminal node of H and we have labeled all of the nodes that are directly above u (i.e., above u and connected to it by an edge). Suppose that there are m such nodes, and they are labeled with ordinal numbers $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_m$ (arranged in non-increasing order). Then label u with the ordinal

$$\omega^{\alpha_1} + \omega^{\alpha_2} + \cdots + \omega^{\alpha_m}.$$

Finally, let $\#(H)$ equal the ordinal number that is assigned to the root of H by this process. Here are a couple of simple examples to illustrate this.

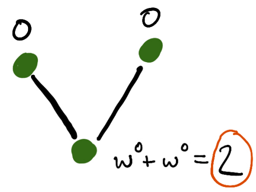

Figure 1.11: A hydra H with $\#(H) = 2$

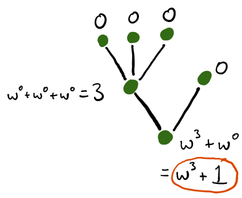

Figure 1.12: A hydra H with $\#(H) = \omega^3 + 1$

If we calculate $\#(H)$ for the first hydra presented above, we find that it is equal to $\omega^{\omega^3+1} + 1$:

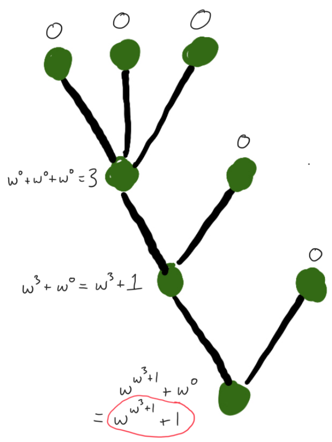

Figure 1.13: A hydra H with $\#(H) = \omega^{\omega^3+1} + 1$.

Now let's see what happens to the ordinal number assigned to this hydra if we make a play of the hydra game and chop off one of its heads. Let's suppose that we are at Move 2 and chop off the left head of this hydra, as depicted in Figure 1.7 above. The resulting hydra H' is depicted in Figure 1.8 above, and we can calculate $\#(H')$ to be $\omega^{\omega^2 \cdot 3+1} + 1$:

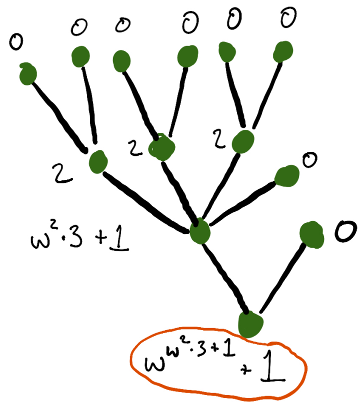

Figure 1.14: A hydra H' with $\#(H') = \omega^{\omega^2 \cdot 3+1} + 1$

Notice that $\omega^2 \cdot 3 < \omega^3$, so $\omega^{\omega^2 \cdot 3+1} + 1 < \omega^{\omega^3+1} + 1$, i.e., $\#(H') < \#(H)$. Thus, by making a move in the hydra game and chopping off a head of H, we created a new hydra H' such that, even though H' has more heads than H, its ordinal value is strictly *smaller*. This is not a coincidence.

Exercise 1.3.2. Calculate the ordinal numbers assigned to the hydras appearing in the run of the hydra game depicted in Figure 1.9 above.

You should have found in the above exercise that the ordinal values assigned to the hydras were strictly decreasing throughout the run of the game. We can in fact prove that this is always the case. The following is the key step of this proof; for now, we leave it as an exercise.

Exercise 1.3.3. Suppose that $\langle H_0, H_1, H_2, H_3, \ldots \rangle$ is a run of the hydra game. Prove that $\#(H_0) > \#(H_1) > \#(H_2) > \#(H_3) > \ldots$. In other words, performing a move in the hydra game always strictly decreases the value of the ordinal assigned to the hydra. (**Hint.** First prove the following basic fact about ordinal arithmetic: for every ordinal α and every natural number n, we have $\omega^\alpha \cdot n < \omega^{\alpha+1}$.)

With the previous exercise, though, we can finish the proof of the theorem! For every run of the hydra game $\langle H_0, H_1, H_2, H_3, \ldots \rangle$, we obtain a strictly decreasing sequence of ordinals $\#(H_0) > \#(H_1) > \#(H_2) > \#(H_3) > \ldots$. Since the ordinals are themselves well-ordered, there can be no infinite strictly decreasing sequences of ordinals. Therefore, *every* run of the hydra game must be finite. In other words, no matter how you play the hydra game, you will always win after some finite number of moves. □

1.4 Peano arithmetic

Our use of infinite ordinals to prove that every hydra game must end after finitely many moves may seem counterintuitive and perhaps unnecessary. The theorem about hydra games is, after all, a statement that is entirely about finite objects. Why should we need to reason about infinite ordinals in order to prove it? However, a remarkable theorem of Kirby and Paris shows that something like this *is* indeed necessary to prove the theorem.

You may have seen before the axioms of Peano Arithmetic (PA), introduced by Giuseppe Peano in the 19th century. These axioms are meant to capture our intuition about the behavior of the natural numbers, and hence about finite discrete objects more broadly.

The language of Peano arithmetic consists of

- the equality sign $=$;
- a constant symbol 0;
- a unary function symbol S.

The intended interpretation of the function S is that it returns the *successor* of its input, i.e., $S(n) = n + 1$. Peano arithmetic has five axioms that are meant to describe the arithmetical properties of the *natural numbers*. These axioms can be stated as follows:

1. 0 is a natural number.
2. For every natural number n, $S(n)$ is also a natural number.
3. For all natural numbers m and n, if $S(m) = S(n)$, then $m = n$.
4. For every natural number n, $0 \neq S(n)$, i.e., 0 is not the successor of any natural number.
5. (Induction) If K is a set such that
   - 0 is in K; and
   - for every natural number n, if n is in K, then $S(n)$ is also in K,

   then K contains every natural number.

Peano Arithmetic captures much of our intuition about the natural numbers, and many theorems about the natural numbers or finite discrete objects can be proven using only PA. For example, much of number theory, as well as the finite Ramsey theorem, can be established in PA. However, Kirby and Paris proved that PA is *not* strong enough to prove that every hydra game must end in a finite number of moves. In fact, they proved the following (stated in a slightly imprecise way):

Theorem 1.4.1 (Kirby–Paris)**.** *If a set of axioms can prove that every hydra game must end in a finite number of moves, then it can also prove the consistency of* PA.

By Gödel's Second Incompleteness Theorem, PA cannot prove its own consistency. Therefore, the Kirby–Paris theorem implies that we cannot prove our theorem about hydra games using PA alone; we must use *something* that goes beyond it.