Hercules, the Hydra, and terminating processes

How do we guarantee that processes terminate?

Hera Brown

The Oxford Compsoc

April 27, 2026

A dreadful battle...

- Hercules is doing battle with the dastardly Lernean Hydra!
- Every time he cuts a head off, more grow back.
- How can Hercules win this battle?

... and a surprising mathematical result

- We can help hercules by doing some mathematics; we can show that Hercules can always win!

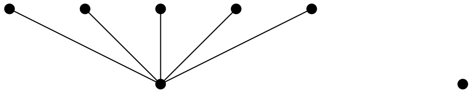


Figure 1: Hercules (right) and the Hydra (left).

- Perhaps more surprisingly, we can show that Hercules **can always win**. Regardless of what he does—regardless of how terrible his strategy is—the hydra will always eventually perish.
- This is surprising! But let's see the proper formal definition.

A zoo of hydras

Let's start with some definitions.

Definition

The Hydra is a finitely-branching rooted tree with finitely many nodes.

So some hydras would be as follows:

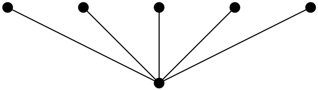


Figure 2: The hydra we saw earlier.

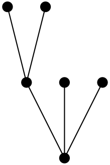


Figure 3: A tall hydra.

Figure 4: A very small hydra.

The rules of the game

Let's also formalise the game:

Definition

The Hydra game is as follows:

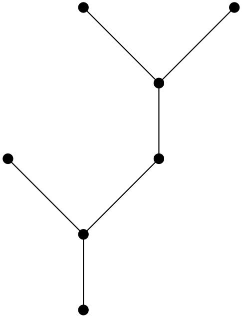
- Once every turn, Hercules chooses a head to cut off.
- If the head that was cut off is immediately above the root of the hydra, then nothing happens.
- If the head that was cut off isn't immediately above the root, then navigate to the grandparent node of the head that was just cut off, and add n copies of the tree rooted at the parent node to the grandparent node.
  Here n is the turn number of the game.[1]

That's dense, so let's look at an example game.

[1] My particular interpretation of the game is taken from [1].

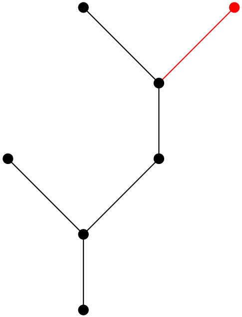
A worked example

- Here's our hydra.



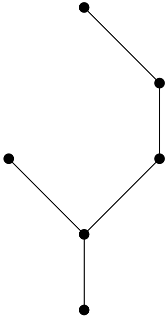
A worked example

- Here's our hydra.



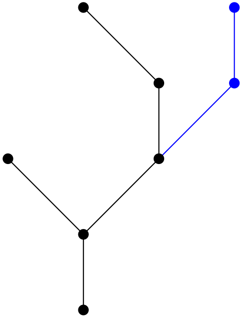
A worked example

- Here's our hydra.



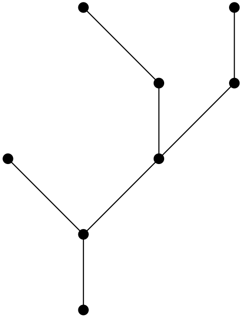
A worked example

- Here's our hydra.



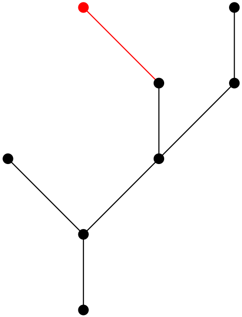
A worked example

- Here's our hydra.



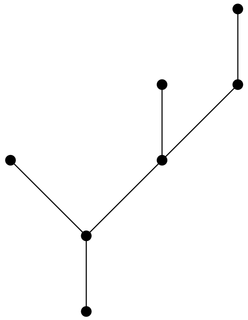
A worked example

- Here's our hydra.



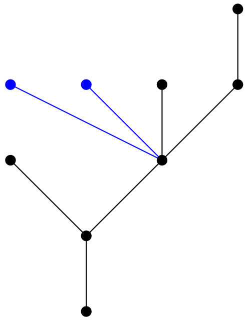
A worked example

- Here's our hydra.



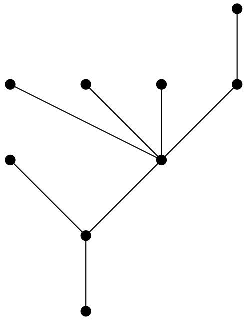
A worked example

- Here's our hydra.



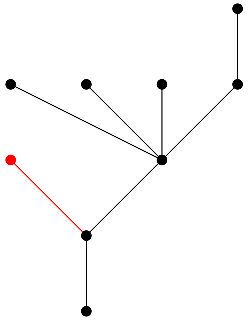
A worked example

- Here's our hydra.



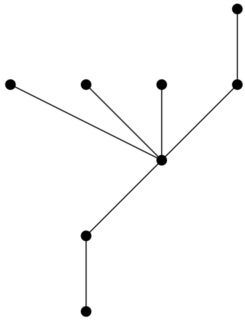
A worked example

- Here's our hydra.



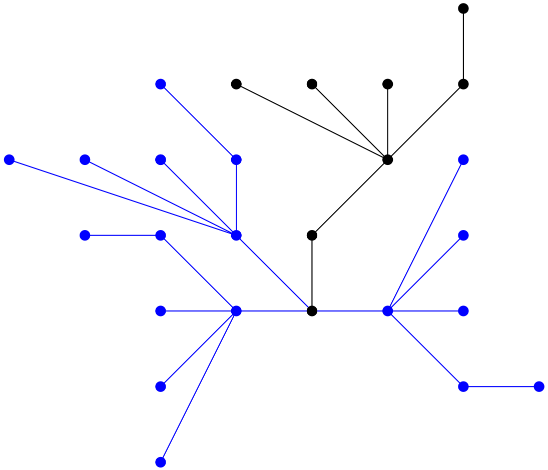
A worked example

- Here's our hydra.



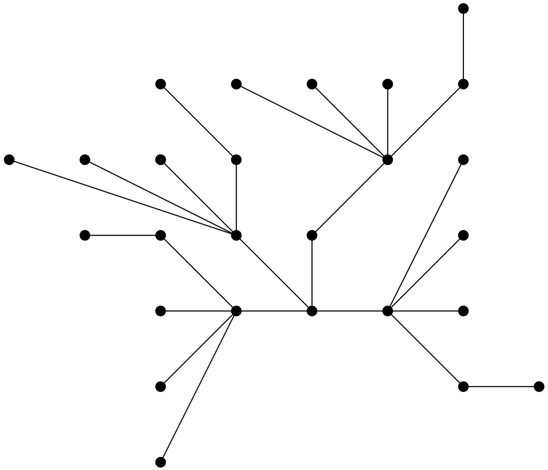
A worked example

- Here's our hydra.



A worked example

- Here's our hydra.



Oh dear.

A proof of termination

- It looks like the game will never end; it looks like the hydra will grow faster and faster as the game goes on.
- But the game does terminate.

Proposition

No matter which heads Hercules cuts off in whatever order, the hydra is eventually defeated.

- Why? The proof is deceptively short:

Proof.

We assign an **ordinal** to each node of the tree. In particular, we assign the ordinal 0 to each of the heads, and for a node with n children, assigned ordinals $\alpha_1 \geq \cdots \geq \alpha_n$, we assign the ordinal $\sum_{i=1}^{n} \omega^{\alpha_i}$.
Then each head Hercules cuts off reduces the ordinal assigned to the root. Since the ordering $\leq$ on the ordinals is a **well-quasi order**, this process must eventually terminate. □

For the rest of the talk, we'll go through what all this means.

An overview, then:

- first we'll look at what ordinals are,
- then we'll look at what a well-quasi order is, and how this relates to the ordinals, and
- then we'll see why this makes our proof work.

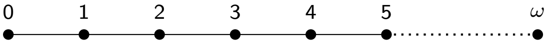
Let's get on with it, then.

To infinity...

- In thinking about ordinals, it helps to think about what it is to count.
- Normally, when counting, we think of the natural numbers:

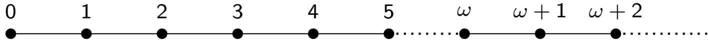
But what comes after all of the natural numbers? It's infinity!

- Here we denote infinity by ω, rather than ∞.
- So, when we're counting things that are really big, we count like so:

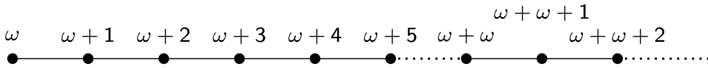


... and beyond

- Why stop at infinity? Of course, infinity plus one is bigger than infinity. And infinity plus two is bigger than that. So we get more numbers:



- There's no reason to stop there, though; what's bigger than $\omega, \omega+1, \omega+2, \omega+3, \ldots$ is just $\omega+\omega$. But we can keep adding to that as well, and so we get:



- We can keep going on like this; you get the idea.

Up to multiplication

- To make life easier for us, we can introduce some notation. Instead of writing $\omega + \omega$, we can write $\omega \cdot 2$. Instead of writing $\omega + \omega + \omega$, we can write $\omega \cdot 3$. And so on.

- (For technical reasons, we have to write $\omega \cdot 2$ instead of $2 \cdot \omega$, because this kind of multiplication isn't commutative!)
- We can count up these in the same way and so we get $\omega \cdot \omega$, then $\omega \cdot \omega \cdot \omega$, and so on.

And off to exponentiation

- Again, we'd expect that we can write $\omega \cdot \omega$ as ω^2, and $\omega \cdot \omega \cdot \omega$ as ω^3. And indeed we can.
- Generalising this (yet again!) we get up to ω^ω, then ω^{ω^ω}, and so on. So we've ended up counting really very high!

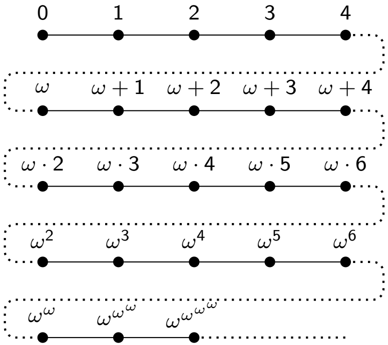


Figure 5: Some of the very big infinities we've seen so far.

- At this point we'll stop trying to think of bigger and bigger numbers; we can absolutely go higher (much, much higher) but for the purposes of this talk we've counted high enough.

These are the ordinals!

- All the numbers we've counted up to in this way—from 0, through all the usual finite numbers, up to infinity, and far off into the distance with exponentials of infinities—are what's called **ordinals**.
- Ordinals are called "ordinals" because they can be used to order things; for instance the ordinals $0, 1, 2, 3, \ldots$ denote the zeroth, first, second, third elements of a sequence and so on.
- Stretching this metaphor, we can think that ω denotes the infinitieth element of a sequence (but don't think about this too hard).
- For those who are interested, here's a formal definition of ordinals in terms of sets:

Definition

An ordinal is a set of one of the following types:

- The zero ordinal, formally $\emptyset$,
- The successor ordinal α^+ of an ordinal α, formally $\alpha \cup \{\alpha\}$, and
- The limit ordinal η of a set of ordinals Γ, where $\bigcup \eta = \eta$.

- So 0 is the zero ordinal, $1, 2, 3, \ldots$ and $\omega + 1, \omega + 2, \omega + 3, \ldots$ are successor ordinals, and ω and ω^2 are limit ordinals.
- But for the purposes of this talk we don't need to know all the technical details.

What do we do with these?

- The ordinals give us a nice general way to count really big things.
- Now, we'll look at how we can use ordinals in our proof that Hercules' battle always terminates.

Ordering the elements of sets

- Consider a decreasing chain of natural numbers; that is, a decreasing sequence of natural numbers.
- Some examples would be:

10	7	5	3	2	1	0

5	4	3	2	1	0

8192	4096	2048	...	4	2	1	0

- One thing to note about all these chains is that they're finite.
- Indeed it doesn't make sense to think of an infinite decreasing chain of natural numbers; there are only ever finitely many numbers strictly less than the first element of the chain.

An intuitive enough result

- For reasons that'll become clear later, this is worth writing down:

Proposition

Every descending chain of natural numbers is finite.

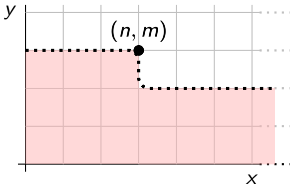
- What other order-able sets have this property? Let's find out.

Pairs of naturals

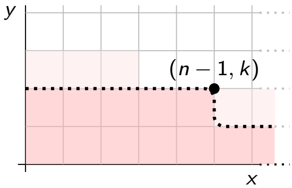
- Another collection of objects we can order are pairs of natural numbers, based on their lexicographic ordering.
- For instance, we say that $(1, 4) < (5, 6)$ and $(4, 0) < (4, 5)$.
- Is every descending chain of pairs of natural numbers finite? Yes.

Finite chains of naturals

- Suppose you start the chain off and give me a pair (n, m). Then I can only pick finitely many elements before I reach $(n, 0)$. At this point, I have to pick the next element to be less than $(n, 0)$.



- Of course, there are infinitely many elements less than $(n, 0)$. But I have to pick one of them; let's call it $(n-1, k)$.



- But k is a natural number, so it's finite! So I can only pick finitely many elements of my chain before I see $(n-1, 0)$.
- Since n is finite, I can only repeat this whole process a finite number of times. So all descending chains of pairs of naturals are finite.

Well-quasi orders

- These orders we've been looking at have a name: they're called "**well-quasi orders**". Formally, they're defined as follows.

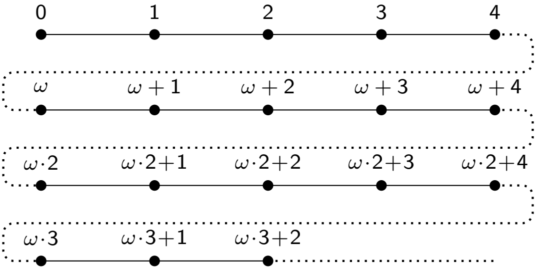
Definition

A well-quasi order is a linear order where every descending chain is finite.

- Well-quasi orders are useful in computer science because, in some sense, their descending chains always terminate. We like to know that things always terminate, and so we like well-quasi orders.
- As a side note, if you've done the Imperative Programming 1 course, then this is the property that lets us guarantee our programs terminate there!

Applying it to ordinals

- As it happens, the usual ordering on the ordinals is a well-quasi order as well.
- That's certainly true for the ordinals that are natural numbers, since there are finitely many of those.
- It's also true for any descending chain starting with ω, because we have to pick some finite number that's less than it.
- Indeed, we can view ordinals of the form $(\omega \cdot n) + m$ as being ordered precisely in the same way as the pairs (n, m) are. We know these are well-quasi ordered, and so those ordinals are too.



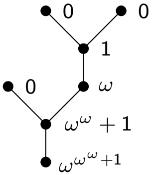
- This generalises, though we don't have to go through the specific details here. For those who know set theory, you can convince yourself this by thinking about the axiom of Foundation hard enough.

Where we go from here

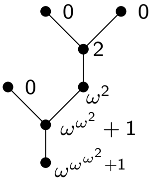
- We've shown that the ordinals have a well-quasi order over them. So any descending sequence of them must eventually hit 0.
- In the proof we saw at the start of the talk, we argued that we can assign an ordinal to the root of the hydra that always strictly decreases each time Hercules chops a head off.
- Hopefully, we're starting to see why this works as a proof. So let's go through it a bit more properly.

Back to the proof

- Let's recall what we do in our proof. Given a hydra, we assigned an ordinal to each node in the following fashion:
  - each head was assigned the ordinal 0, and
  - each node with children assigned ordinals $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$ was assigned the ordinal $\sum_{i=1}^{n\omega^{\alpha_i}}$.
- For instance, for the hydra on the right, we get the following labelling:

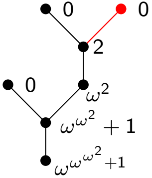


Back to the proof



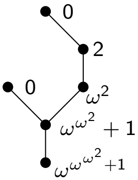
- Let's convince ourselves that the ordinals assigned to edges really do decrease, by playing through a game:

Back to the proof



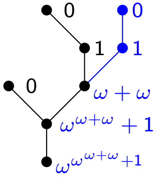
- Let's convince ourselves that the ordinals assigned to edges really do decrease, by playing through a game:

Back to the proof



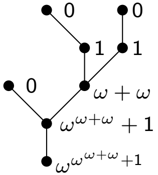
- Let's convince ourselves that the ordinals assigned to edges really do decrease, by playing through a game:

Back to the proof



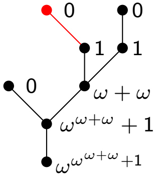
- Let's convince ourselves that the ordinals assigned to edges really do decrease, by playing through a game:

Back to the proof



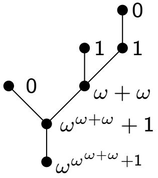
- Let's convince ourselves that the ordinals assigned to edges really do decrease, by playing through a game:

Back to the proof



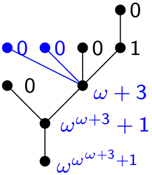
- Let's convince ourselves that the ordinals assigned to edges really do decrease, by playing through a game:

Back to the proof



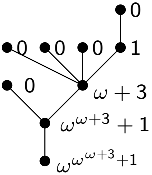
- Let's convince ourselves that the ordinals assigned to edges really do decrease, by playing through a game:

Back to the proof



- Let's convince ourselves that the ordinals assigned to edges really do decrease, by playing through a game:

Back to the proof



- Let's convince ourselves that the ordinals assigned to edges really do decrease, by playing through a game:
- And so on.

Why does the root ordinal always decrease?

- The trick is that, every time we remove a head, we paste a new tree down at the **grandparent** node.
- This has the effect of slowly flattening out the tree as the game progresses.
- Since a flatter tree gets us a smaller ordinal, it follows that cutting heads off always reduces the ordinal at the root of the tree.
- If we're so inclined we can prove this by an induction.

Where do we go from here?

- In this talk, we've showed that Hercules always defeats the hydra, regardless of what moves he makes.
- But this isn't just good new for Hercules; it's good news for us as well. The methods we've used and the results we have help us to show:
  - that certain classes of processes and functions—for instance, those that branch and fork in ways similar to how the hydra grows heads—can be guaranteed to terminate,

```
process forking-thread(n){
  [do some work]
  for(i = 1 to n){
    parent.parent.fork(parent.thread);
  }
  end;
}
```

Figure 6: A program that we now know terminates.

  - that we can use well-quasi orderings to guarantee termination more generally, and
  - that just because something is infinite (like the ordinals), it doesn't mean that we can't use that thing in showing that something else is finite.

References

Noah Schoem.
Well-foundedness of countable ordinals and the hydra game, 2014.

Goodbye!

So, the main takeaways:

- Even processes that branch heavily can be guaranteed to terminate.
- Well-quasi orderings are great ways to show that a process terminates.
- The ordinals are a helpful generalisation of the natural numbers.

Thanks for listening!

Any questions?