The Ackermann Function

A function which witnesses the inclusion $PR \subsetneq \mathcal{R} \cap Tot$ is the Ackermann function (the Greek letter below is "Psi" and an animation of the function here and my general summary on its point here)

The function is an example of a total recursive function that is not primitive recursive (so, not primitive recursive but computable, only *not using the set of primitive recursive operations,* as it has more cases). Its definition involves unbounded recursion, which is not guaranteed to terminate.

The Ackermann's function is $\psi : \mathbb{N}^2 \to \mathbb{N}$ defined as

$$\begin{cases} \psi(0, y) = y + 1 \\ \psi(x + 1, 0) = \psi(x, 1) \\ \psi(x + 1, y + 1) = \psi(x, \psi(x + 1, y)) \end{cases}$$

The function takes as argument two non-negative integers, so x and y. Specifically:

- Base case: the result is simply one more than the second argument y, which resembles the behavior of a successor function.

- In another base case where the second argument y is 0, the function behaves as follows: $\psi(x, 0) = \psi(x - 1, 1)$. Here, when y is 0, the function recursively calls itself with x decreased by 1 and y set to 1. So, first argument simply diminishes.

- In the most complex case, when both x and y are greater than 0, the Ackermann function proceeds with deep nested recursion. It evaluates as $\psi(x, y) = \psi(x - 1, \psi(p, y - 1))$. This case involves two levels of recursion. Initially, it calculates $\psi(x, y - 1)$, effectively decreasing the second argument y. Then, it calculates $\psi(x - 1, \psi(x, y - 1))$, resulting in a nested recursive structure.
    - So, in this case, we have a case in which the first argument gets smaller, another when the second argument gets smaller

In other terms, the arguments of the function diminish in a *lexicographical order* on N^2 (two-dimensional, because it depends on both x and y) inside sequences of numbers: $(N^2, \leq_{lex}), (x, y) \leq_{lex} (x', y') \; if \; x < x' or \; (x = x') \; and \; (y \leq y')$. We can show $(N^2, \leq_{lex})$ does not allow for infinite descending sequences.

This means that given two pairs (x, y) and (x', y'), the lexicographical order $\leq_{lex}$ dictates that (x, y) is considered less than or equal to (x', y') if either x is less than x' or, in the case of equal x values, y is less than or equal to y'.

$$(1000, 1000000) <_{lex} (1001, 0)$$

$$(1000, 1000000) >_{lex} (1001, 0)$$

Here, when you compare $(1000, 1000000)$ and $(1001, 0)$:

- The lexicographical order first compares the first elements: 1000 and 1001. Here, 1000 is less than 1001, so $(1000, 1000000) <_{lex} (1001, 0)$ because the first element x in the first pair is smaller than the first element x in the second pair.

- If the first elements were equal, the lexicographical order would then compare the second elements y. In this case, the second elements are 1000000 and 0. Because the first elements x have already determined the order ($1000 < 1001$), there's no need to further compare the second elements y. In lexicographical order, if the first elements are different, the comparison stops at that point.

Concretely: the function grows enormously, but the argument diminish according to an order.

An example:

$$f: \mathbb{Z} \to \mathbb{Z}$$

$$f(z) = \begin{cases} 0, & z \geq 0 \\ f(z-1), & z < 0 \end{cases}$$

Simply, we say that this has a finite recursion → $f(-1) \to f(-2) \to f(-3)$ …

In the example of $f(z)$, when the input is initially negative ($z < 0$), the function repeatedly reduces the value of z by 1. This process continues until z reaches a non-negative value.

Ackermann function has a logically sound recursion and to show so, we use need more notions.

Partially Ordered/Well Founded Posets

Definition (partially ordered set)

We define then a partially ordered set (abbreviated as "poset", elements defined in an increasing order) as:

$$(D, \leq) \qquad \leq reflexive \qquad x \leq x$$

$$\leq antisymmetric \qquad x \leq y \; and \; y \leq x \Rightarrow x = y$$

$$\leq transitive \qquad x \leq y \; and \; y \leq z \Rightarrow x \leq z$$

So basically, we analyze a binary relation in which elements are ordered (reflexive), there are no circular relations between elements because elements are different (antisymmetric) and elements have a predictable order (transitive).

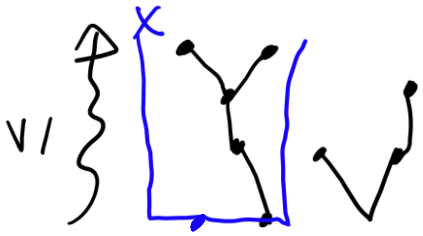
In a partially ordered set, some pairs of elements may be related, while others may not be related at all.

Definition (Well-founded posets)

$$(D, \leq) \; is \; well - founded \; if \; \forall x \subseteq D \; , x \neq 0, has \; a \; minimal \; element$$

In other words, within any non-empty subset, you can always find an element that is minimal with respect to the given partial order (so, we have no infinite descending chains) – this is what happens in Ackermann.

- This means there's no other element in X that is strictly smaller than d in terms of the order relation ($\leq$).
- If the computation is well-founded, there is always a step which leads to a smaller value and eventually the program will terminate, given *we will always find a minimal* element.

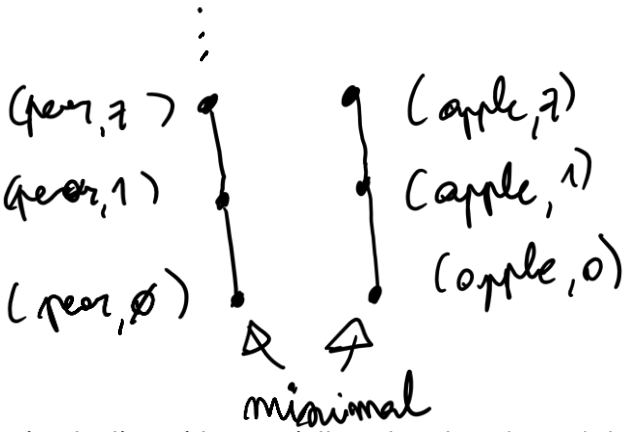


$$d \in X \; minimal \; s.t. if \; d' \leq d \; then \; d' = d$$

(given the partial order, “we can’t mix different things together” aka “you can’t mix pears with apples”).

$$D = \{(pear, n), (apple, n) \mid n \in N\}$$

$$(x, y) \leq (x', y') \; if \; (x = x') \; and \; (y \leq y')$$



In this context, we’re dealing with a partially ordered set (poset) that consists of pairs of elements, where each element is associated with a label (such as "$pear$" or "$apple$") and a natural number (n).

The partial order on this set is defined such that elements are ordered first by their labels and then, within the same label, by their associated natural numbers. The key relationship here is "≤," which denotes the partial order on this set.

- An element (x, y) is considered less than or equal to (x', y') if and only if both the labels are the same ($x = x'$) and the natural number associated with the first element is less than or equal to the natural number associated with the second element ($y \leq y'$).

Here's the explanation for the example in the context of the partial order:

- Suppose you have two elements, (x, y) and (x', y') in D. These elements represent items labeled "$pear$" and "$apple$," along with natural numbers, respectively.

- The partial order specifies that you can't mix items of different labels, meaning you can't compare "$pears$" with "$apples$" in this order. So, comparing (x, y) and (x', y') only makes sense when x and x' are the same (both "$pear$" or both "$apple$").

- Once you've established that the labels match ($x = x'$), you can compare the natural numbers (y and y'). The element (x, y) is considered "minimal" if there is no other element (x', y') in D with the same label ($x = x'$) where y' is less than or equal to y.

We note also:

- Is $\mathbb{Z}$ well-founded? No (we can’t always find a minimal element)
- Is $\mathbb{N}$ well-founded? Yes (given it’s a well-ordered set, we can always find a minimal element)

Note: $(D, \leq)$ is well-founded if and only if there is no infinite descending chain $d_0 > d_1 > d_2 \ldots$ in D. This way our computation descends a decreasing sequence of values, which is necessarily finite.

(In words: This fact can be useful when dealing with termination problems. If we can conclude that the set of configurations is well-founded, we simply need to prove that inductively this is all defined. This works also here with Ackermann: given the computation is based on smaller values, at some point it will end)

Remember from before that $(\mathbb{N}^2 \leq_{lex})$ is well-founded.

Let $x \subseteq \mathbb{N}^2, x \neq 0$:

$$x_0 = \min\{x \mid \exists y.\, (x, y) \in X\}$$

$$y_0 = \min\{y \mid (x_0, y) \in X\} \rightarrow (x_0, y_0) = \min\ X$$

Essentially, what we just said simply means there is always a smallest element according to the lexicographical order and there is always a well-defined order.

To explain the further concepts, we need to properly define induction.

Given $P(n), n \in \mathbb{N}$, $P(0)$ and assuming $P(n)$ you can deduce $P(n+1)$

$$\Downarrow$$

$$P(n)\ holds\ \forall n$$

(essentially, given a case, if it holds for a base case, it will hold for all natural numbers). Let's give a simple reasoning by induction: a *binary tree* formation.

Statement: "A binary tree with height h has at most $2^{n+1} - 1$ nodes"

- Base case $(n = 0) \rightarrow$ number of nodes $= 1 \leq 2^{0+1} - 1 = 2 - 1 = 1$
- Recursive case $(n \rightarrow n + 1)$

$n_1, n_2 < n+1$
the inductive step is only on n...
you "can't" conclude
↓
actually, you "could" with a clever hypothesis formulation

Complete/Well-founded Induction and Ackermann Proof

As shown here, normal induction reaches cases where it can't conclude (basically, the height of a binary tree can vary, because it would involve proving that if statement holds for trees of different height using always the same k; this is not linear, and the proof would require *bounding* the height to a value and inductively show the thing).

We then need the complete induction, in which this can be applied to any well-founded poset.

Specifically, to prove that $P(n)$ holds $\forall n \in \mathbb{N}$, show

$\forall n$, assuming $P(n')\ \forall n' < n$ then $P(n)$

All of this is a well-founded induction. We define here $(D, \leq)$ well-founded order, $P(x)$ property over D, if $\forall d \in D$, assuming $\forall d' < d, P(d')$, we can conclude $P(d)$ "holds everywhere":

$$\Downarrow$$

$$\forall d \in D\ P(d)$$

All this tour leads to a conclusion: the Ackermann function is total and if a property holds for all numbers before, then it will hold for all those after.

Formally:

1) Ψ is total

- $\forall (x, y) \in N^2, \Psi(x, y) \downarrow$ proceed by well-founded induction of $(N^2, \leq_{lex})$

Proof

Let $(x, y) \in \mathbb{N}^2$, assume $\forall (x', y') <_{lex} (x, y), \Psi(x', y') \downarrow$, we want to show $\Psi(x, y) \downarrow$

We have three cases:

1) $(x = 0) \rightarrow \Psi(x, y) = \Psi(0, y) = y + 1 \downarrow$

2) $(x > 0, y = 0) \rightarrow \Psi(x, 0) = \Psi(x - 1), 1)$
   - $(x - 1) <_{lex} (x, y)$ *hence* $\Psi(x - 1, 1) \downarrow$ by inductive hypothesis

3) $(x > 0, y > 0) \rightarrow \Psi(x, y) = \Psi(x - 1), \Psi(x, y - 1) = \Psi(x - 1, u) \downarrow$ (by ind. hyp.)
   - $<_{lex} (x, y) \rightarrow \Psi(x, y - 1) \downarrow = u$ by inductive hypothesis

Essentially, we prove Ackermann is total given two non-negative numbers which are well-defined in their order. We consider three cases based on the values of x and y:

- *Case 1* $(x = 0)$*:* In this case, if x is 0, we know that $\Psi(x, y)$ is $\Psi(0, y)$. This leads to a straightforward result, which is y + 1. The function Ψ(0, y) is guaranteed to terminate, so this case is covered.

- *Case 2* $(x > 0\ and\ y = 0)$*:* When x is greater than 0 and y is 0, we have $\Psi(x, 0) = \Psi(x - 1, 1)$. We know that $\Psi(x - 1, 1)$ terminates because it's part of our induction hypothesis. This means that $\Psi(x, 0)$ also terminates.

- *Case 3* $(x > 0\ and\ y > 0)$*:* In the most complex case, where both x and y are greater than $0, \Psi(x, y)$ involves a nested recursion. It's defined as $\Psi(x - 1, \Psi(x, y - 1))$. Our induction hypothesis ensures that $\Psi(x, y - 1)$ terminates (denoted as "u"). Since $(x - 1, u)$ is smaller than (x, y), and we've assumed that for all smaller pairs, Ψ terminates, we can conclude that $\Psi(x, y)$ also terminates.

If you want to discuss infinite things:

$$(N^2, \leq_{lex})$$

$$(0,0)\ \ (0,1)\ \ (0,2)\ \ \ldots.\quad (1,0)\ \ (1,1)\ \ (1,2)\ \ \ldots.\quad (2,0)\ \ (2,1)\ \ (2,2)\ \ \ldots.$$

$\mathbb{N}$ $\mathbb{N}$ $\mathbb{N}$

We are highlighting that, even though this set contains an infinite number of pairs, they are ordered in a systematic and predictable way. As you move through this set, the values in each pair follow a pattern, allowing us to compare and order them consistently.

Within this well-ordered set, the Ackermann function operates by moving through these pairs in a specific manner. It doesn't "jump out" of this structured order.

- The function goes through a process of descending values within this well-ordered set
- This way, it can compute values within the natural numbers (N) without running into infinite or unbounded operations.

One could argue by using the Church-Turing thesis: the computation of $\Psi(x, y)$ is always reduced to the computation of ψ on smaller input values until we reach a base case where the successor is used. The above is unsatisfactory. Given it is always defined, it is total.

2) $\Psi \in \mathbb{R} = \mathcal{C}$

$$\Psi(1,1) = \Psi\big(0, \Psi(1,0)\big) = \Psi(0,2) = 3$$

$\Psi(1,0)$

1

2

$$(1,1,3) \quad (0,2,3) \quad (1,0,2) \quad (0,1,2)$$

In words: we reach sets of values defined by recursion, in this case triples defined inside these sets. Especially, we characterize:

$$(x, y, z) \in N^3 \qquad \rightarrow Z = \Psi(x, y)$$

$$\rightarrow S \; contains \; all \; triples \; needed \; to \; compute \; \Psi(x, y)$$

A set $S \subseteq \mathbb{N}^3$ is considered *valid* if, for all $(x, y, z) \in S$, it satisfies two conditions:

1. z equals $\Psi(x, y)$, ensuring that the results in the set are consistent with the Ackermann function.
2. S contains all the triples needed to compute $\Psi(x, y)$ for different values of x and y.

Formally, you just need to recall the function is defined (Ackermann system of equations definition), so $S \subseteq N^3$ is valid.

- $(0, y, z) \in S \rightarrow z = y + 1$
- $(x + 1,0, z) \in S \rightarrow (x, 1, z) \in S$
- $(x + 1, y + 1, Z) \in S \rightarrow \exists u \; (x + 1, y, u) \in S \; and \; (x, u, z) \in S$

You can show:

$$\forall (x, y) \in N^2, z \in N \rightarrow \Psi(x, y) = z \;\; iff \; \exists \; valid \; set \; of \; triples \; S \in N^3 \; and \; S \; finite \; s.t. \; (x, y, z) \in S$$

(essentially, we have $\Psi(x, y) = z \; iff$ a valid finite set of triples is defined by complete induction, knowing the set is preserved under union)

Then (in words: every triple can be encoded as a set of numbers and then as a number using primes)

$$\Psi(x, y) = " \; \mu \; (S, z). (S \; \in N^3 \; finite \; valid \; set \; of \; triples \; and \; (x, y, z))"$$

encode as a number

$$\rightarrow \Psi \in \mathcal{R} = \mathcal{C}$$

(so, there exists the smallest finite number in which a valid set of triples minimizes correctly and gives values recursively enumerable inside the resulting function – aka computable and quantifiable).

3) $y \notin \mathcal{PR}$

This part of the proof wants to show that Ψ is not a primitive recursive function showing it grows faster than every other function in PR. We combine nested primitive recursion to show Ackermann cannot compute a finite number of nested primitive recursions.

By using primitive recursion you can define the sum via the successor:

- $x + y$

$$x + 0 = x$$

$$x + (y + 1) = (x + y) + 1$$

- $x * y$

$$x * 0 = 0$$

$$x + (y + 1) = (x * y) + x$$

- x^y

$$x^0 = 1$$

$$x^{(y+1)} = (x^y) * x$$

nesting primitive recursions for loops

(so, essentially, basic arithmetic operations under primitive recursion is defined).

Consider x as first parameter:

$$\Psi_x(y) = \Psi(x, y)$$

$$\Psi_{x+1}(y) = \Psi_x\big(\Psi_{x+1}(y-1)\big) = \Psi_x\Big(\Psi_x\big(\Psi_{x+1}(y-2)\big)\Big) = \underbrace{\Psi_x \dots \Psi_x}_{y \text{ times}}\, \Psi_{x+1}(0)$$

$$= \underbrace{\Psi_x \dots \Psi_x(1)}_{y+1} = \Psi_x^{y+1}(1)$$

Roughly: increasing $x \; to \; x + 1$ you need iterating Ψ_x , $y + 1$ times $\rightarrow additional \; for \; loop$

So, we need minimalization $\rightarrow \Psi \notin \mathcal{PR}$

(So, number of nested recursions is infinite and with primitive recursion we can't quantify it prior; it's necessary to use the unbounded minimalisation, hence Ackermann is not primitive recursive)

Yeah, that was quite a ride, wasn't it?

- Short explanation in words:

Intuitively, if x grows so does the level of nesting in the functions, which is equivalent to say that we need more nested for loops. Since x can grow to infinity and for loops cannot be nested to infinity, a

while loop is needed. The for-loops nesting level won't be able to catch up in a bounded way, so is not in PR.

- Longer explanation in words:

The discussion transitions to $\Psi_{x+1}(y)$, representing the Ackermann function for the next value of x. The key insight here is that when you increase x by 1, you need to iterate $\Psi_x(y)$ a certain number of times, specifically $y + 1$ times. This represents additional loops or iterations in the computation.

The key insight is that you need a form of "minimalization" to determine how many additional iterations are required when x increases by 1, *but primitive recursive functions lack this capacity*.

For the final part, to account for this additional looping when increasing x, a process of minimalization is introduced. This is because the Ackermann function doesn't fit neatly into the framework of primitive recursion, as it requires also *unbounded iterations*, giving we need additional loop and iteration "to try to reach a finite value".

To properly move ahead with the function, you need to iterate the function multiple times, which goes beyond what primitive recursion can handle. This leads to the conclusion that Ψ is not a primitive recursive function, as it necessitates unbounded iteration and minimalization, making it a more powerful and complex function.

- At the end of the day:

To be able to define all total functions you also need minimalization: otherwise, some functions might be too powerful to express traditionally, like happens here.

Mathematically, we have:

$$\psi \in \mathcal{R} \cap Tot, \Psi \notin \mathcal{PR}$$

and so:

$$PR \subsetneq \mathcal{R} \cap Tot$$