udf Induction

ind.1 Introduction

mth:ind:int:sec

Induction is an important proof technique which is used, in different forms, in almost all areas of logic, theoretical computer science, and mathematics. It is needed to prove many of the results in logic.

Induction is often contrasted with deduction, and characterized as the inference from the particular to the general. For instance, if we observe many green emeralds, and nothing that we would call an emerald that's not green, we might conclude that all emeralds are green. This is an inductive inference, in that it proceeds from many particlar cases (this emerald is green, that emerald is green, etc.) to a general claim (all emeralds are green). *Mathematical* induction is also an inference that concludes a general claim, but it is of a very different kind that this "simple induction."

Very roughly, an inductive proof in mathematics concludes that all mathematical objects of a certain sort have a certain property. In the simplest case, the mathematical objects an inductive proof is concerned with are natural numbers. In that case an inductive proof is used to establish that all natural numbers have some property, and it does this by showing that (1) 0 has the property, and (2) whenever a number n has the property, so does $n + 1$. Induction on natural numbers can then also often be used to prove general about mathematical objects that can be assigned numbers. For instance, finite sets each have a finite number n of elements, and if we can use induction to show that every number n has the property "all finite sets of size n are ..." then we will have shown something about all finite sets.

Induction can also be generalized to mathematical objects that are *inductively defined*. For instance, expressions of a formal language suchh as those of first-order logic are defined inductively. *Structural induction* is a way to prove results about all such expressions. Structural induction, in particular, is very useful—and widely used—in logic.

ind.2 Induction on $\mathbb{N}$

mth:ind:inN:sec

In its simplest form, induction is a technique used to prove results for all natural numbers. It uses the fact that by starting from 0 and repeatedly adding 1 we eventually reach every natural number. So to prove that something is true for every number, we can (1) establish that it is true for 0 and (2) show that whenever it is true for a number n, it is also true for the next number $n+1$. If we abbreviate "number n has property P" by $P(n)$, then a proof by induction that $P(n)$ for all $n \in \mathbb{N}$ consists of:

1. a proof of $P(0)$, and

2. a proof that, for any n, if $P(n)$ then $P(n + 1)$.

To make this crystal clear, suppose we have both (1) and (2). Then (1) tells us that $P(0)$ is true. If we also have (2), we know in particular that if $P(0)$ then $P(0+1)$, i.e., $P(1)$. (This follows from the general statement "for any n, if $P(n)$ then $P(n+1)$" by putting 0 for n. So by modus ponens, we have that $P(1)$. From (2) again, now taking 1 for n, we have: if $P(1)$ then $P(2)$. Since we've just established $P(1)$, by modus ponens, we have $P(2)$. And so on. For any number k, after doing this k steps, we eventually arrive at $P(k)$. So (1) and (2) together establish $P(k)$ for any $k \in \mathbb{N}$.

Let's look at an example. Suppose we want to find out how many different sums we can throw with n dice. Although it might seem silly, let's start with 0 dice. If you have no dice there's only one possible sum you can "throw": no dots at all, which sums to 0. So the number of different possible throws is 1. If you have only one die, i.e., $n = 1$, there are six possible values, 1 through 6. With two dice, we can throw any sum from 2 through 12, that's 11 possibilities. With three dice, we can throw any number from 3 to 18, i.e., 16 different possibilities. 1, 6, 11, 16: looks like a pattern: maybe the answer is $5n + 1$? Of course, $5n + 1$ is the maximum possible, because there are only $5n+1$ numbers between n, the lowest value you can throw with n dice (all 1's) and $6n$, the highest you can throw (all 6's).

Theorem ind.1. *With n dice one can throw all $5n+1$ possible values between n and $6n$.*

Proof. Let $P(n)$ be the claim: "It is possible to throw any number between n and $6n$ using n dice." To use induction, we prove:

1. The *induction basis* $P(1)$, i.e., with just one die, you can throw any number between 1 and 6.

2. The *induction step*, for all k, if $P(k)$ then $P(k+1)$.

(1) Is proved by inspecting a 6-sided die. It has all 6 sides, and every number between 1 and 6 shows up one on of the sides. So it is possible to throw any number between 1 and 6 using a single die.

To prove (2), we assume the antecedent of the conditional, i.e., $P(k)$. This assumption is called the *inductive hypothesis*. We use it to prove $P(k+1)$. The hard part is to find a way of thinking about the possible values of a throw of $k+1$ dice in terms of the possible values of throws of k dice plus of throws of the extra $k+1$-st die—this is what we have to do, though, if we want to use the inductive hypothesis.

The inductive hypothesis says we can get any number between k and $6k$ using k dice. If we throw a 1 with our $(k+1)$-st die, this adds 1 to the total. So we can throw any value between $k+1$ and $6k+1$ by throwing 5 dice and then rolling a 1 with the $(k+1)$-st die. What's left? The values $6k+2$ through $6k+6$. We can get these by rolling k 6s and then a number between 2 and 6 with our $(k+1)$-st die. Together, this means that with $k+1$ dice we can throw any of the numbers between $k+1$ and $6(k+1)$, i.e., we've proved $P(k+1)$ using the assumption $P(k)$, the inductive hypothesis. □

Very often we use induction when we want to prove something about a series of objects (numbers, sets, etc.) that is itself defined "inductively," i.e., by defining the $(n+1)$-st object in terms of the n-th. For instance, we can define the sum s_n of the natural numbers up to n by

$$\begin{aligned} s_0 &= 0 \\ s_{n+1} &= s_n + (n+1) \end{aligned}$$

This definition gives:

$$\begin{aligned} s_0 &= 0, \\ s_1 &= s_0 + 1 && = 1, \\ s_2 &= s_1 + 2 && = 1 + 2 = 3 \\ s_3 &= s_2 + 3 && = 1 + 2 + 3 = 6, \text{ etc.} \end{aligned}$$

Now we can prove, by induction, that $s_n = n(n+1)/2$.

Proposition ind.2. $s_n = n(n+1)/2$.

Proof. We have to prove (1) that $s_0 = 0 \cdot (0+1)/2$ and (2) if $s_n = n(n+1)/2$ then $s_{n+1} = (n+1)(n+2)/2$. (1) is obvious. To prove (2), we assume the inductive hypothesis: $s_n = n(n+1)/2$. Using it, we have to show that $s_{n+1} = (n+1)(n+2)/2$.

What is s_{n+1}? By the definition, $s_{n+1} = s_n + (n+1)$. By inductive hypothesis, $s_n = n(n+1)/2$. We can substitute this into the previous equation, and then just need a bit of arithmetic of fractions:

$$\begin{aligned} s_{n+1} &= \frac{n(n+1)}{2} + (n+1) = \\ &= \frac{n(n+1)}{2} + \frac{2(n+1)}{2} = \\ &= \frac{n(n+1) + 2(n+1)}{2} = \\ &= \frac{(n+2)(n+1)}{2}. \end{aligned}$$

□

The important lesson here is that if you're proving something about some inductively defined sequence a_n, induction is the obvious way to go. And even if it isn't (as in the case of the possibilities of dice throws), you can use induction if you can somehow relate the case for $n+1$ to the case for n.

ind.3 Strong Induction

mth:ind:str: sec

In the principle of induction discussed above, we prove $P(0)$ and also if $P(n)$, then $P(n+1)$. In the second part, we assume that $P(n)$ is true and use

this assumption to prove $P(n+1)$. Equivalently, of course, we could assume $P(n-1)$ and use it to prove $P(n)$—the important part is that we be able to carry out the inference from any number to its successor; that we can prove the claim in question for any number under the assumption it holds for its predecessor.

There is a variant of the principle of induction in which we don't just assume that the claim holds for the predecessor $n-1$ of n, but for all numbers smaller than n, and use this assumption to establish the claim for n. This also gives us the claim $P(k)$ for all $k \in \mathbb{N}$. For once we have established $P(0)$, we have thereby established that P holds for all numbers less than 1. And if we know that if $P(l)$ for all $l < n$ then $P(n)$, we know this in particular for $n = 1$. So we can conclude $P(2)$. With this we have proved $P(0)$, $P(1)$, $P(2)$, i.e., $P(l)$ for all $l < 3$, and since we have also the conditional, if $P(l)$ for all $l < 3$, then $P(3)$, we can conclude $P(3)$, and so on.

In fact, if we can establish the general conditional "for all n, if $P(l)$ for all $l < n$, then $P(n)$," we do not have to establish $P(0)$ anymore, since it follows from it. For remember that a general claim like "for all $l < n$, $P(l)$" is true if there are no $l < n$. This is a case of vacuous quantification: "all As are Bs" is true if there are no As, $\forall x\,(\varphi(x) \to \psi(x))$ is true if no x satisfies $\varphi(x)$. In this case, the formalized version would be "$\forall l\,(l < n \to P(l))$"—and that is true if there are no $l < n$. And if $n = 0$ that's exactly the case: no $l < 0$, hence "for all $l < 0$, $P(0)$" is true, whatever P is. A proof of "if $P(l)$ for all $l < n$, then $P(n)$" thus automatically establishes $P(0)$.

This variant is useful if establishing the claim for n can't be made to just rely on the claim for $n-1$ but may require the assumption that it is true for one or more $l < n$.

ind.4 Inductive Definitions

mth:ind:idf: sec

In logic we very often define kinds of objects *inductively*, i.e., by specifying rules for what counts as an object of the kind to be defined which explain how to get new objects of that kind from old objects of that kind. For instance, we often define special kinds of sequences of symbols, such as the terms and formulas of a language, by induction. For a simple example, consider strings of consisting of letters a, b, c, d, the symbol ∘, and brackets [and], such as "[[c ∘ d][", "[a[]∘]", "a" or "[[a ∘ b] ∘ d]". You probably feel that there's something "wrong" with the first two strings: the brackets don't "balance" at all in the first, and you might feel that the "∘" should "connect" expressions that themselves make sense. The third and fourth string look better: for every "[" there's a closing "]" (if there are any at all), and for any ∘ we can find "nice" expressions on either side, surrounded by a pair of parenteses.

We would like to precisely specify what counts as a "nice term." First of all, every letter by itself is nice. Anything that's not just a letter by itself should be of the form "$[t \circ s]$" where s and t are themselves nice. Conversely, if t and s are nice, then we can form a new nice term by putting a ∘ between them and

surround them by a pair of brackets. We might use these operations to *define* the set of nice terms. This is an *inductive definition*.

Definition ind.3 (Nice terms)**.** The set of *nice terms* is inductively defined as follows:

1. Any letter a, b, c, d is a nice term.
2. If s and s' are nice terms, then so is $[s \circ s']$.
3. Nothing else is a nice term.

This definition tells us that something counts as a nice term iff it can be constructed according to the two conditions (1) and (2) in some finite number of steps. In the first step, we construct all nice terms just consisting of letters by themselves, i.e.,

$$\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}$$

In the second step, we apply (2) to the terms we've constructed. We'll get

$$[\mathrm{a} \circ \mathrm{a}], [\mathrm{a} \circ \mathrm{b}], [\mathrm{b} \circ \mathrm{a}], \ldots, [\mathrm{d} \circ \mathrm{d}]$$

for all combinations of two letters. In the third step, we apply (2) again, to any two nice terms we've constructed so far. We get new nice term such as [a∘[a∘a]]—where t is a from step 1 and s is [a∘a] from step 2—and [[b∘c]∘[d∘b]] constructed out of the two terms [b ∘ c] and [d ∘ b] from step 2. And so on. Clause (3) rules out that anything not constructed in this way sneaks into the set of nice terms.

Note that we have not yet proved that every sequence of symbols that "feels" nice is nice according to this definition. However, it should be clear that everything we can construct does in fact "feel nice:" brackets are balanced, and ∘ connects parts that are themselves nice.

The key feature of inductive definitions is that if you want to prove something about all nice terms, the definition tells you which cases you must consider. For instance, if you are told that t is a nice term, the inductive definition tells you what t can look like: t can be a letter, or it can be $[r \circ s]$ for some other pair of nice terms r and s. Because of clause (3), those are the only possibilities.

When proving claims about all of an inductively defined set, the strong form of induction becomes particularly important. For instance, suppose we want to prove that for every nice term of length n, the number of [in it is $< n/2$. This can be seen as a claim about all n: for every n, the number of [in any nice term of length n is $< n/2$.

Proposition ind.4. *For any n, the number of [in a nice term of length n is $< n/2$.*

Proof. To prove this result by (strong) induction, we have to show that the following conditional claim is true:

If for every $k < n$, any parexpression of length k has $k/2$ ['s, then any parexpression of length n has $n/2$ ['s.

To show this conditional, assume that its antecedent is true, i.e., assume that for any $k < n$, parexpressions of length k contain $< k/2$ ['s. We call this assumption the inductive hypothesis. We want to show the same is true for parexpressions of length n.

So suppose t is a nice term of length n. Because parexpressions are inductively defined, we have three two cases: (1) t is a letter by itself, or t is $[r \circ s]$ for some nice terms r and s.

1. t is a letter. Then $n = 1$, and the number of [in t is 0. Since $0 < 1/2$, the claim holds.

2. t is $[s \circ s']$ for some nice terms s and s'. Let's let k be the length of s and k' be the length of s'. Then the length n of t is $k + k' + 3$ (the lengths of s and s' plus three symbols [, $\circ$,]). Since $k + k' + 3$ is always greater than k, $k < n$. Similarly, $k' < n$. That means that the induction hypothesis applies to the terms s and s': the number m of [in s is $< k/2$, and the number of [in s' is $< k'/2$.

   The number of [in t is the number of [in s, plus the number of [in s', plus 1, i.e., it is $m + m' + 1$. Since $m < k/2$ and $m' < k'/2$ we have:

$$m + m' + 1 < \frac{k}{2} + \frac{k'}{2} + 1 = \frac{k + k' + 2}{2} < \frac{k + k' + 3}{2} = n/2.$$

In each case, we've shown that the number of [in t is $< n/2$ (on the basis of the inductive hypothesis). By strong induction, the proposition follows. □

Problem ind.1. Define the set of supernice terms by

1. Any letter a, b, c, d is a supernice term.
2. If s is a supernice term, then so is $[s]$.
3. If t and s are supernice terms, then so is $[t \circ s]$.
4. Nothing else is a supernice term.

Show that the number of [in a supernice term s of length n is $\leq n/2 + 1$.

ind.5 Structural Induction

mth:ind:sti:sec

So far we have used induction to establish results about all natural numbers. But a corresponding principle can be used directly to prove results about all elements of an inductively defined set. This often called *structural* induction, because it depends on the structure of the inductively defined objects.

Generally, an inductive definition is given by (a) a list of "initial" elements of the set and (b) a list of operations which produce new elements of the set

from old ones. In the case of nice terms, for instance, the initial objects are the letters. We only have one operation: the operations are

$$o(s, s') = [s \circ s']$$

You can even think of the natural numbers $\mathbb{N}$ themselves as being given be an inductive definition: the initial object is 0, and the operation is the successor function $x + 1$.

In order to prove something about all elements of an inductively defined set, i.e., that every element of the set has a property P, we must:

1. Prove that the initial objects have P
2. Prove that for each operation o, if the arguments have P, so does the result.

For instance, in order to prove something about all nice terms, we would prove that it is true about all letters, and that it is true about $[s \circ s']$ provided it is true of s and s' individually.

Proposition ind.5. *The number of* [*equals the number of*] *in any nice term* t*.*

Proof. We use structural induction. Nice terms are inductively defined, with letters as initial objects and the operations o for constructing new nice terms out of old ones.

1. The claim is true for every letter, since the number of [in a letter by itself is 0 and the number of] in it is also 0.
2. Suppose the number of [in s equals the number of], and the same is true for s'. The number of [in $o(s, s')$, i.e., in $[s \circ s']$, is the sum of the number of [in s and s'. The number of] in $o(s, s')$ is the sum of the number of] in s and s'. Thus, the number of [in $o(s, s')$ equals the number of] in $o(s, s')$.

□

Problem ind.2. Prove by structural induction that no nice term starts with].

Let's give another proof by structural induction: a proper initial segment of a string of symbols t is any string t' that agrees with t symbol by symbol, read from the left, but t' is longer. So, e.g., $[a \circ$ is a proper initial segment of $[a \circ b]$, but neither are $[b \circ$ (they disagree at the second symbol) nor $[a \circ b]$ (they are the same length).

mth:ind:sti: prop:initial

Proposition ind.6. *Every proper initial segment of a nice term* t *has more* [*'s than*]*'s.*

Proof. By induction on t:

1. t is a letter by itself: Then t has no proper initial segments.

2. $t = [s \circ s']$ for some nice terms s and s'. If r is a proper initial segment of t, there are a number of possibilities:

   a) r is just [: Then r has one more [than it does].

   b) r is $[r'$ where r' is a proper initial segment of s: Since s is a nice term, by induction hypothesis, r' has more [than] and the same is true for $[r'$.

   c) r is $[s$ or $[s \circ$: By the previous result, the number of [and] in s is equal; so the number of [in $[s$ or $[s \circ$ is one more than the number of].

   d) r is $[s \circ r'$ where r' is a proper initial segment of s': By induction hypothesis, r' contains more [than]. By the previous result, the number of [and of] in s is equal. So the number of [in $[s \circ r'$ is greater than the number of].

   e) r is $[s \circ s'$: By the previous result, the number of [and] in s is equal, and the same for s'. So there is one more [in $[s \circ s'$ than there are].

□

ind.6 Relations and Functions

mth:ind:rel:sec

When we have defined a set of objects (such as the natural numbers or the nice terms) inductively, we can also define *relations on* these objects by induction. For instance, consider the following idea: a nice term t is a subterm of a nice term t' if it occurs as a part of it. Let's use a symbol for it: $t \sqsubseteq t'$. Every nice term is a subterm of itself, of course: $t \sqsubseteq t$. We can give an inductive definition of this relation as follows:

Definition ind.7. The relation of a nice term t being a subterm of t', $t \sqsubseteq t'$, is defined by induction on s' as follows:

1. If t' is a letter, then $t \sqsubseteq t'$ iff $t = t'$.

2. If t' is $[s \circ s']$, then $t \sqsubseteq t'$ iff $t = t'$, $t \sqsubseteq s$, or $t \sqsubseteq s'$.

This definition, for instance, will tell us that a $\sqsubseteq$ [b ∘ a]. For (2) says that a $\sqsubseteq$ [b ∘ a] iff a = [b ∘ a], or a $\sqsubseteq$ *b*, or a $\sqsubseteq$ a. The first two are false: a clearly isn't identical to [b∘a], and by (1), a $\sqsubseteq$ b iff a = b, which is also false. However, also by (1), a $\sqsubseteq$ a iff a = a, which is true.

It's important to note that the success of this definition depends on a fact that we haven't proved yet: every nice term t is either a letter by itself, or there are uniquely determined nice terms s and s' such that $t = [s \circ s']$. "Uniquely determined" here means that if $t = [s \circ s']$ it isn't *also* $= [r \circ r']$ with $s \neq r$ or $s' \neq r'$. If this were the case, then clause (2) may come in conflict with itself: reading t' as $[s \circ s']$ we might get $t \sqsubseteq t'$, but if we read t' as $[r \circ r']$ we might get not $t \sqsubseteq t'$. Before we prove that this can't happen, let's look at an example where it *can* happen.

Definition ind.8. Define *bracketless terms* inductively by

1. Every letter is a bracketles term.
2. If s and s' are bracketless terms, then $s \circ s'$ is a bracketless term.
3. Nothing else is a bracketless term.

Bracketless terms are, e.g., a, b ∘ d, b ∘ a ∘ b. Now if we defined "subterm" for bracketless terms the way we did above, the second clause would read

$$\text{If } t' = s \circ s', \text{ then } t \sqsubseteq t' \text{ iff } t = t', t \sqsubseteq s, \text{ or } t \sqsubseteq s'.$$

Now b ∘ a ∘ b is of the form $s \circ s'$ with $s = \mathrm{b}$ and $s' = \mathrm{a} \circ \mathrm{b}$. It is also of the form $r \circ r'$ with $r = \mathrm{b} \circ \mathrm{a}$ and $r' = \mathrm{b}$. Now is a ∘ b a subterm of b ∘ a ∘ b? The answer is yes if we go by the first reading, and no if we go by the second.

The property that the way a nice term is built up from other nice terms is unique is called *unique readability*. Since inductive definitions of relations for such inductively defined objects are important, we have to prove that it holds.

Proposition ind.9. *Suppose t is a nice term. Then either t is a letter by itself, or there are uniquely determined nice terms s, s' such that $t = [s \circ s']$.*

Proof. If t is a letter by itself, the condition issatisfied. So assume t isn't a letter by itself. We can tell from the inductive definition that then t must be of the form $[s \circ s']$ for some nice terms s and s'. It remains to show that these are uniquely determined, i.e., if $t = [r \circ r']$, then $s = r$ and $s' = r'$.

So suppose $t = [s \circ s']$ and $t = [r \circ r']$ for nice terms s, s', r, r'. We have to show that $s = r$ and $s' = r'$. First, s and r must be identical, for otherwise one is a proper initial segment of the other. But by Proposition ind.6, that is impossible if s and r are both nice terms. But if $s = r$, then clearly also $s' = r'$. □

We can also define functions inductively: e.g., we can define the function f that maps any nice term to the maximum depth of nested $[\dots]$ in it as follows:

mth:ind:rel: defn:depth **Definition ind.10.** The *depth* of a nice term, $f(t)$, is defined inductively as follows:

$$f(s) = 0 \text{ if } s \text{ is a letter}$$
$$f([s \circ s'] = \max(f(s), f(s')) + 1$$

For instance

$$\begin{aligned} f([\mathrm{a} \circ \mathrm{b}]) &= \max(f(\mathrm{a}), f(\mathrm{b})) + 1 = \\ &= \max(0, 0) + 1 = 1, \text{ and} \\ f([[\mathrm{a} \circ \mathrm{b}] \circ \mathrm{c}]) &= \max(f([\mathrm{a} \circ \mathrm{b}]), f(\mathrm{c})) + 1 = \\ &= \max(1, 0) + 1 = 2. \end{aligned}$$

Here, of course, we assume that s an s' are nice terms, and make use of the fact that every nice term is either a letter or of the form $[s \circ s']$. It is again important that it can be of this form in only one way. To see why, consider again the bracketless terms we defined earlier. The corresponding "definition" would be:

$$g(s) = 0 \text{ if } s \text{ is a letter}$$
$$g(s \circ s') = \max(g(s), g(s')) + 1$$

Now consider the bracketless term $\mathrm{a} \circ \mathrm{b} \circ \mathrm{c} \circ \mathrm{d}$. It can be read in more than one way, e.g., as $s \circ s'$ with $s = \mathrm{a}$ and $s' = \mathrm{b} \circ \mathrm{c} \circ \mathrm{d}$, or as $r \circ r'$ with $r = \mathrm{a} \circ b$ and $r' = \mathrm{c} \circ \mathrm{d}$. Calculating g according to the first way of reading it would give

$$\begin{aligned} g(s \circ s') &= \max(g(\mathrm{a}), g(\mathrm{b} \circ \mathrm{c} \circ \mathrm{d})) + 1 = \\ &= \max(0, 2) + 1 = 3 \end{aligned}$$

while according to the other reading we get

$$\begin{aligned} g(r \circ r') &= \max(g(\mathrm{a} \circ \mathrm{b}), g(\mathrm{c} \circ \mathrm{d})) + 1 = \\ &= \max(1, 1) + 1 = 2 \end{aligned}$$

But a function must always yield a unique value; so our "definition" of g doesn't define a function at all.

Problem ind.3. Give an inductive definition of the function l, where $l(t)$ is the number of symbols in the nice term t.

Problem ind.4. Prove by induction on nice terms t that $f(t) < l(t)$ (where $l(t)$ is the number of symbols in t and $f(t)$ is the depth of t as defined in Definition ind.10).

Photo Credits

Bibliography