Notes on Foundations of Programming Languages
Induction

Sandra Alves

September 17, 2023

Abstract

Induction is a fundamental proof technique for the topics discussed in this course. We will present several forms of induction: from induction on natural numbers, to structural induction, or induction on proofs. Although both structural induction and induction on proofs can be seen as special cases of induction on the natural numbers (considering as measure the size of the terms in the case of structural induction, and the lenght of the proofs for induction on proofs), we will present all these forms of induction, as instances of a more general notion, which is the notion of *well-founded induction*.

1 Induction on Natural Numbers

We start by defining the principal of mathematical induction in its most common form.

Definition 1.1 (Induction on natural numbers) *Given a certain property* $P(n)$*, with* $n \in \mathbb{N}$*. Suppose the following hold:*

- $P(0)$
- *For every* $k \in \mathbb{N}$*, if* $P(k)$*, then* $P(k+1)$*.*

Then $\forall n. P(n)$ *is true.*

We refer to the proof of $P(0)$ as the *base case*, and to the proof that $P(k)$ implies $P(k+1)$ as the *induction step*. We assume $P(k)$ to be true in order to proof $P(k+1)$, and refer to it as *induction hypothesis*.

It is clear that having proved the base case and the induction step, then from $P(0)$ and the induction step one can infer $P(1)$. Using $P(1)$ and the same argument, we obtain a proof of $P(2)$, and so on. Therefore, it is possible in this way to build a proof for any given n. We will now see an example of a simple proof by induction.

Example 1.2 *Consider the property "Every positive interger power of* 3 *is odd". We want to show that for every* $n \in \mathbb{N}^+$*, it holds that* $3^n = 2l + 1$*, for some* $l \in \mathbb{N}$*. We proceed by induction.*

- *Base case: for* $n = 0$ *we have* $1 = 2 \cdot 0 + 1$*, so the property holds.*
- *Induction step: suppose* 3^k *is odd (that is* $3^k = 2l + 1$*,* $l \in \mathbb{N}$*), we want to prove that* 3^{k+1} *is also odd.*
$$\begin{aligned} 3^{k+1} &= 3 \cdot 3^k \\ &= 3(2l+1) && \textit{(by induction hypothesis)} \\ &= 2(3l+1)+1 \end{aligned}$$
Therefore 3^{k+1} *is odd, which concludes the proof.*

Definition 1.3 (Strong induction on natural numbers) *Given a certain property* $P(n)$*, with* $n \in \mathbb{N}$*. Suppose the following hold:*

- $P(0)$
- *For every* $k \in \mathbb{N}$*, if* $P(0) \wedge P(1) \wedge \cdots \wedge P(k)$*, then* $P(k+1)$*.*

Then $\forall n. P(n)$ *is true.*

Example 1.4 *Consider the property "Any positive integer greater than one, can be written as a product of prime numbers". We proceed by strong induction.*

- *Base case: for* $n = 2$*, since* 2 *is a prime number the property holds. (Note that the base case is* 2 *and not* 0*, since we want to prove the claim for all the positive integer greater or equal than* 2*.)*
- *Induction step: Lets assume the claim for all the positive integers between* 2 *and* k*. If* $k+1$ *is a prime number, then the property holds trivially. Otherwise,* $k+1$ *has a positive divisor other than* 1 *and itself. Therefore* $k+1 = a \cdot b$*, with* $2 \le a, b \le k$*. By induction hypothesis, both* a *and* b *can be written as a product of primes, therefore so can their product, which concludes the proof.*

Note than, although we use the term *"strong"*, to refer to this second case of induction, in fact both forms are equivalent with respect to the properties one can prove. That is, one can express the first form using the second and vice versa.

Let $P(n)$ be a property on the natural numbers and assume the conditions of the induction principle:

1) $P(0)$

2) For every $k \in \mathbb{N}$, if $P(k)$, then $P(k+1)$.

Lets prove by strong induction that $\forall n.P(n)$. From (1) we get $P(0)$. Lets assume $P(0) \wedge P(1) \wedge \cdots \wedge P(k)$, which implies $P(k)$. Therefore, by (2), we get $P(k+1)$. Thus, by strong induction $\forall n.P(n)$.

Now, lets assume the conditions of the strong induction principle:

1) $P(0)$

2) For every $k \in \mathbb{N}$, if $P(0) \wedge P(1) \wedge \cdots \wedge P(k)$, then $P(k+1)$.

Lets prove by weak induction that $\forall n.P(n)$.

Let $Q(m)$ be $P(n)$ holds for all n such that $0 \leq n \leq m$. Apply mathematical induction to $Q(m)$. Since $Q(0)$ is just $P(0)$, we have the base case. Now suppose $Q(m)$ is given and we wish to show $Q(m+1)$. Notice that $Q(m)$ is the same as $P(0) \wedge P(1) \wedge \cdots \wedge P(m)$. The hypothesis of strong induction tells us that this implies $P(m+1)$. If we add $P(m+1)$ to $Q(m)$, we get $P(0) \wedge P(1) \wedge \cdots P(m) \wedge P(m+1)$, which is just $Q(m+1)$. So using mathematical induction, we get that $Q(n)$ holds for all natural numbers n. But $Q(n)$ implies $P(n)$, so we have the conclusion of strong induction, namely that $P(n)$ holds for all natural numbers n.

We can also use induction on natural numbers to prove properties on other sets (for example properties over trees, or terms of a given language), by defining a suitable function on the natural numbers.

Example 1.5 *Consider the following data type defining binary trees:*

$$\texttt{data (Arv a)} = \texttt{Empty} \mid \texttt{Leaf a} \mid \texttt{No (Arv a) (Arv a)}$$

Consider the following property of binary trees:

$$P(t) \ = \textit{tree } t \textit{ has at most one more leaf than internal nodes.}$$

One can define a function height : Trees $\rightarrow \mathbb{N}$, *and rewrite* $P(t)$ *as the property* $Q(n)$ *on natural numbers:*

$$Q(n) = \forall \textit{ trees } t, \textit{ if } \mathsf{height}(t) = n \textit{ then } P(t).$$

2 Induction on Expressions and Proofs

We now look at two particular kinds of induction to prove properties on expressions generated by a grammar or over proofs defined by a particular proof system.

Definition 2.1 (Structural induction) *Let e be an expression generated by a particular grammar and $P(e)$ a property on e. If*

1. *the property holds for any atomic expression;*

2. *for any compound expression e', with imediate sub-expressions $e_1, \ldots, e_k$, if $P(e_i)$ for $i = 1, \ldots, k$ implies $P(e')$;*

Then $P(e)$ holds for every expression e.

A stronger version of the above principle requires that, for any non-atomic expression, $P(e')$ holds, if it holds for any sub-expression (imediate or not). These two principles are related, respectively, with the weak and strong principles of induction on natural numbers (note that any proof by structural induction can be writen as a proof of induction on the natural numbers over the size of the expression.

The other induction principle we will consider, will be induction on proofs over a Hilbert proof system.

Definition 2.2 *A Hilbert proof system consists of a set of axioms and a set of inference rules:*

- *an axiom is a statment that is provable by definition;*
- *an inference rule determines that, if a list of statments (called premises) is provable then so it is the conclusion of the rule.*

$$\frac{A_1 \ \cdots \ A_n}{B}$$

Definition 2.3 *Let π be a proof in some proof system, and $P(\pi)$ a property on π. If*

1. *the property holds for any axiom;*
2. *the property holds for proofs $\pi_1, \ldots, \pi_k$ (with shorter proofs), implies that the property holds for π', where π' ends by extending one or more of the proofs $\pi_1, \ldots, \pi_k$, with an inference rule.*

Then $P(\pi)$ holds for every proof π, of the proof system.

3 Well-founded Induction

We can see all these different kinds of induction as instances of a general form of induction on what are called "well-founded relations".

Definition 3.1 (Well-founded relation) *A well-founded relation on a set A is a binary relation $\prec$ on A, such that, there is no infinite descending sequence $a_0 \succ a_1 \succ a_2 \succ \cdots$.*

Lemma 3.2 *Let $\prec$ be a binary relation on A. Then $\prec$ is well-founded if and only if every nonempty subset of A has a minimal element.*

Proof: We prove the two directions separately.

($\Rightarrow$) Suppose $\prec$ is a well-founded relation on A and let $B \subseteq A$ be any nonempty subset. We will show, by contradiction, that B has a minimal element. If B does not have a minimal element, then for any $a \in B$ there exists $a' \in B$ such that $a' \prec a$. But then we can build the infinite sequence $a_0 \succ a_1 \succ a_2 \succ \cdots$ starting with any $a_0 \in B$ and using the fact that no a_i can be minimal since B has no minimal element.

($\Leftarrow$) Suppose that any subset has a minimal element. Then there can be no infinite decreasing sequence $a_0 \succ a_1 \succ a_2 \succ \cdots$ since such a sequence would give us a set $\{a_0, a_1, a_2, \ldots\}$ without a minimal element. This completes the proof.

□

Proposition 3.3 (Well-founded induction) *Let $\prec$ be a well-founded binary relation on a set A and let P be some property on A. If $P(a)$ holds whenever we have $P(b)$ for all $b \prec a$, then $P(a)$ is true for all $a \in A$.*

Form of Induction	Well-founded relation
Natural number induction (weak)	$m \prec n$ if $m + 1 = n$
Natural number induction (strong)	$m \prec n$ if $m < n$
Structural induction (weak)	$e \prec e'$ if e is an immediate subexpression of e'
Structural induction (strong)	$e \prec e'$ if e is a subexpression of e'
Induction on proofs	$\pi \prec \pi'$ if π is the subproof for some antecedent of the last inference rule in proof π'

Table 1: Well founded relations for common forms of induction

Proof: Suppose $\forall a.(\forall b.(b \prec a \Rightarrow P(b)) \Rightarrow P(a))$. We will show, by contradiction, that $P(a)$ holds for $a \in A$. Suppose there exists $x \in A$, such that $\neg P(x)$. Therefore the set $B = \{a \in A \mid \neg P(a)\}$ is nonempty, which means B has a minimal element $a \in B$. But since we then have $P(b)$ for all $b \prec a$, this contradicts the assumption $\forall b.(b \prec a \Rightarrow P(b)) \Rightarrow P(a)$, which concludes the proof. □

Exercises

1 Consider the following data type defining binary trees:

$$\texttt{data (Arv a)} = \texttt{Empty} \mid \texttt{Leaf a} \mid \texttt{No (Arv a) (Arv a)}$$

Using induction prove that any tree of type `Arv a`, has at most one more leaf than internal nodes.

2 Let $\mathcal{V}$ be an infinite set of variables. Consider the following grammar for expressions:

$$e := 0 \mid 1 \mid v \mid e + e \mid e * e$$

Prove that, for any list of variables $v_0, \ldots, v_n$, containing all the variables in a given expression e, there exists a polinomial $p_n = cv_0^k v_1^k \ldots v_n^k$ such that, for all the possible values of $v_0, \ldots, v_n > 0$, the value of e is less than p_n.

3 Consider the following relation on $\mathbb{N}^2$:

$$(n, m) \prec (n', m') \text{ iff } n < n' \text{ or } (n = n' \text{ and } m < m')$$

Prove that $\prec$ is a well-founded relation.

4 Let L be the set of *expressions* containing only the symbols (and), defined the following way:

- () is an expression;
- if α is an expression, then (α) is an expression;
- if α and β are expressions, then $\alpha\beta$ is an expression.

where $\alpha\beta$ denotes the concatenation of the two expressions. Show that for every expression in L:

(a) the number of (is equal to the number of).

(b) in every prefix of an expression the number of) does not excede the number of (.

5 Consider the set of *words* containing only the letters M, I, U, and defined in the following way:

- MI is a word;
- if $x\mathsf{I}$ is a word, then $x\mathsf{IU}$ is a word;
- if $\mathsf{M}x$ is a word, then $\mathsf{M}xx$ is a word;
- if $x\mathsf{III}y$ is a word, then $x\mathsf{U}y$ is a word;
- if $x\mathsf{UU}y$ is a word, the xy is a word.

where x, y are any sequences of M's, I's and U's.

(a) Is MU a word?

(b) Prove by induction that the number if occurrences of I in any word is nerver a multiple of 3. What can you conclude?

6 Let L be the set of words defined in the following way:

- a and b are words;
- if β is a word then $aa\beta$ and $bb\beta$ are also words;

(a) Prove that any word of L has either an even number of a's and an odd number of b's, or an even number of b's and an odd number of a's.

(b) Show that L can be described as the set of words build of blocks of a's and/or b's, being even the numebr of letters in each block, except on the last block.