INDUCTION AND RECURSION

OUTLINE:

(1) Basic induction
(2) Basic recursion
(3) Variations
(4) Structural induction and recursion
(5) More examples

1. BASIC INDUCTION

Mathematical induction

- The 2 most basic properties which define the set of natural numbers are:
  1) The set has a minimum element 0
  2) Each element n has a successor $n+1$
- Mathematical induction is a proof technique which exploits the construction of the set of natural numbers. In its basic formulation, it says:
- In order to prove that a certain statement holds for any natural number, it is sufficient to
  1) Prove the statement for 0 ("base case");
  2) Prove that, assuming the statement holds for a generic natural number k (this assumption is called "inductive hypothesis", IH), then it also holds for $k+1$ ("induction step").

Example

- Prove that, for any natural number *n*, the sum of the natural numbers from *0* to *n* is

$$0+1+2+...+n=\frac{n(n+1)}{2}$$

- Base case: for *n = 0*, the sum of the natural numbers from *0* to *0*, that is just *0*, equals *0(0+1)/2 = 0*.
- Induction step: assume that for an arbitrary *k* we have

$$0+1+2+...+k=\frac{k(k+1)}{2}$$

(this is our inductive hypothesis, IH). We want to show that

$$0+1+...+k+(k+1)=\frac{(k+1)((k+1)+1)}{2}=\frac{(k+1)(k+2)}{2}$$

Example

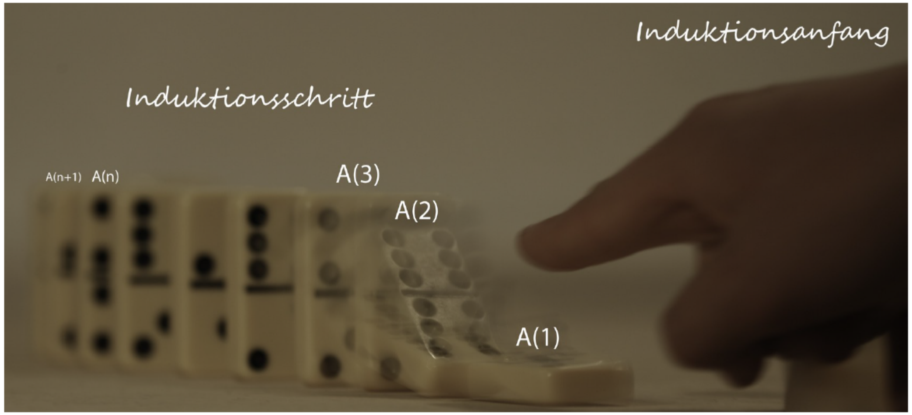
$$0+1+\ldots+k+(k+1) \overset{IH}{=} \frac{k(k+1)}{2}+(k+1) = \frac{k(k+1)+2(k+1)}{2} = \frac{(k+1)(k+2)}{2}$$

Note that I have actively used IH as a key ingredient to get to the formula I wanted.

Why does this work?

- The intuitive idea behind the induction principle is the same as the idea of domino show: if we can be sure that
  - the first tile falls (base case)
  - provided the k^{th} tile falls (IH), then the $(k+1)^{th}$ tile also falls (inductive step)

  then all tiles fall.



2. BASIC RECURSION

Recursion

- Recursion is mathematical induction's twin sibling.
- Induction is a proof strategy, exploiting the structure of natural numbers to prove statements.
- Recursion is a definition method, exploiting the structure of natural numbers to define objects (usually functions).
- In its basic form, the recursive definition of an object depending on the natural numbers involves:

  1) Defining the object for the natural 0 ("base case");

  2) Defining the object for an arbitrary natural number $k+1$ in terms of the definition of the object for the natural number k ("induction step").

Example: the factorial.

- The factorial of a natural number *n*, denoted n!, is a natural number defined recursively as follows:
- Base case: $0! = 1$ (direct, explicit definition).
- Inductive step: for a generic natural *k*, we define $(k+1)! = k! \cdot (k+1)$ (the definition of the factorial of $k+1$ is given in terms of the factorial of the smaller number *k*).

3. VARIATIONS

Different base cases

- Sometimes, statements only hold from a certain natural number *b* onwards, or a recursive definition makes sense only from a certain natural number *b* onwards. In these situations, induction or recursion cannot be started at *0*, but rather we have to use *b* as our basis step.

Different base cases

- EX: prove that, for any $n \geq 4$, $n! > 2^n$.
- Note that the statement is false for $n = 0,1,2,3$, therefore, we start with the base case $n = 4$.
- Base case: for $n = 4$ we have $4! = 24 > 16 = 2^4$.
- Induction step: assume that, for a generic natural $k \geq 4$, $k! > 2^k$ (IH). Then

$$(k+1)! = k!\cdot(k+1) \overset{IH}{>} 2^k\cdot(k+1) > 2^k\cdot 2 = 2^{k+1}$$

Several base cases

- Sometimes, to inductively prove a statement or to recursively define something on the naturals, we need more than one base case.
- EX: The Fibonacci numbers F_n (a very interesting sequence of numbers with crazily deep properties) are defined recursively as follows:
  - Base case 1: $F_0 = 0$
  - Base case 2: $F_1 = 1$
  - Induction step: $F_{n+2} = F_{n+1} + F_n$

  Note that in this definition the inductive step requires 2 calls of the definition in 2 previous cases, and correspondingly there are 2 base cases to trigger the recursive process.

Strong induction

- Strong induction is a refined form of basic induction in which
  - The basis step works in the same way
  - For the inductive step, we prove that the statement for a generic natural $k+1$ holds if we assume that the statement holds for any natural $\leq k$ (not just for k).

An example of strong induction

- Prove the correctness of integer division, i.e., that for all integers $n \geq 0$ (the dividend) and $m > 0$ (the divisor) there are two integers q (the quotient) and r (the remainder), with $0 \leq r < m$, such that $n = mq + r$.
- We proceed by induction on n: to simplify the notation, let $A(n)$ denote the following sentence

  "For all integers $m>0$ there are two integers q and r, with $0 \leq r < m$, such that $n = mq + r$."
- Base case: if $n = 0$, then for all m the assertion is true with $q = r = 0$.

An example of strong induction

- Inductive step: let $k \geq 1$; we have to verify that $A(k)$ follows from the IH that $A(j)$ holds for all integers j between 0 and $k-1$ (included). [Note that, for cleanliness of notation, instead of proving $A(k+1)$ given $A(0),A(1),...,A(k)$, we prove $A(k)$ given $A(0),A(1),...,A(k-1)$.] Let's then pick a positive integer m.
  - Case 1: $m > k$. This is easy, just set $q = 0$ and $r = k$. (here we don't need IH).
  - Case 2: $m \leq k$. Then $k-m$ is a natural number strictly smaller than n (why strictly?), so, by IH, $A(k-m)$ holds, that is, there are integers q and r, with $0 \leq r < m$, such that $k-m = mq+r$. But then $k = m(q+1)+r$, as we required.
- This is an extreme case of induction: to make the inductive step work, we need to assume as IH that all previous instances $A(0)$, $A(1)$, ..., $A(k-1)$ hold; this is because $k-m$ can assume any value between 0 and k.

4. STRUCTURAL INDUCTION AND RECURSION

Recursively defined sets

- The idea of induction / recursion can be extended to sets other than the natural numbers, but present a similar structure.
- A recursively defined set is a set defined through:
  - The explicit specification of the “simplest” element(s) of the set (base case);
  - The specification of how “more complicated” elements of the set can be constructed from “simpler” elements (induction step).
  - The declaration (often tacitly implied) that nothing else belongs to the set.
- If a set is recursively defined, then structural induction and recursion can be applied to prove statements or define functions and attributes about the elements of the set.
- Structural induction and recursion work exactly as induction and recursion for natural numbers.

Strings

- An alphabet A is a non-empty finite set of symbols, called the characters.
- A string on the alphabet A is a finite sequence of characters of A.
- The set of strings on an alphabet can be recursively defined via the concatenation operator $\circ$ which joins 2 strings into one (e.g., $abc \circ cba = abccba$). Note that $\circ$ is associative: if S,T,U are strings, then $S\circ(T\circ U) = (S\circ T)\circ U$.

Strings

- Base case: the empty string on an alphabet A is the string made of no characters. It is denoted with a symbol not in A, say ε.
- Induction step: for any string S on A and any character c in A, $S \circ c$ is a string on A.
- Nothing else is a string on A.
- Now that we have a recursive definition of strings, we can use structural recursion to define string attributes and structural induction to prove statements on strings.

String length

- We want to define a function *length* which takes a string as input and gives the number of characters forming the string as output. By structural recursion, we can proceed as follows:
  - Base case: we define $length(\varepsilon) = 0$.
  - Inductive step: for any string S and any character c we define $length(S \circ c) = length(S) + 1$.

String length

- Now we want to prove that the length function satisfies the following property: for all strings S and T, $length(S \circ T) = length(S) + length(T)$.
- We can proceed by structural induction on T.
  - Base case: $T = \varepsilon$. Then, for any string S, $S \circ T = S$. Therefore, $length(S \circ T) = length(S) = length(S) + 0 = length(S) + length(T)$.
  - Inductive step: let $T = U \circ c$ for a suitable string U and a character c. Note that U is “simpler” than T. The IH is that for any string S and any string V simpler (shorter) than T (including U), $length(S \circ V) = length(S) + length(V)$. Then

$length\ (S \circ T) = length(S \circ (U \circ c)) = length((S \circ U) \circ c)$	[associativity of $\circ$]
$= length(S \circ U) + 1$	[recursive definition of length]
$= length(S) + length\ (U) + 1$	[IH]
$= length\ (S) + length\ (T)$	[recursive definition of length]

Well-formed formulas

- The propositional well-formed formulas are in fact formally defined via a recursive definition. We just have to make brackets parts of the formulas in the induction step in order to achieve correctness:
- Base case: every atom is a WFF
- Induction step: If A and B are WFFs, then so are $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ [and also $(A \leftrightarrow B)$, $(A \oplus B)$, $(A \downarrow B)$, $(A \uparrow B)$]
- Nothing else is a WFF

5. MORE EXAMPLES

Factorization into primes

- Prove that any integer $n \geq 2$ has a factorization into the product of prime numbers.
- Base case: 2 is prime, so it is its own factorization into primes.
- Induction step: let $n \geq 3$ be a natural number and assume (strong IH) that any natural between 2 and n has a factorization into primes.
  - If n is prime, $n = n$ is its factorization into primes. Done.
  - If n is not prime, then by definition n can be factored into the product of two integers $a, b \geq 2$, that is, $n = ab$. But since $a < n$ and $b < n$ [why?], by strong IH both a and b have a factorization into primes, say $a = p_1p_2...p_k$ and $b = q_1q_2...q_j$.
  - Therefore, $n = ab = p_1p_2...p_kq_1q_2...q_j$ is a factorization of n into primes.

What’s wrong?

- I will now “prove” that all natural numbers have the same parity (that is, the same remainder modulo 2: they are all even or all odd).
- Base case: trivially, 0 has the same parity as itself.
- Inductive step: assume by IH that any n natural numbers have the same parity. Consider a set consisting of $n+1$ natural numbers $\{a_0,a_1,...,a_n\}$.
  - First, remove one of the numbers, say a_0, and look at the subset of the other n numbers $\{a_1,...,a_n\}$: by IH they have the same parity.
  - Now remove another number, say a_n: by IH, the remaining numbers $a_0,a_1,...,a_{n-1}$ (among which is the previously removed number a_0) have again the same parity.
  - Therefore, the number a_0 has the same parity as all the other n numbers, that is, all $n+1$ numbers have the same parity.
- By induction, all natural numbers have the same parity.