COMP9020 Lecture 6
Session 2, 2017
Induction and Recursion

- Textbook (R & W) - Ch. 4, Sec. 4.2, 4.4, 4.6
- Problem set 7
- Supplementary Exercises Ch. 4 (R & W)

Inductive Reasoning

Suppose we would like to reach a conclusion of the form

P(x) *for all* x (of some type)

Inductive reasoning (as understood in philosophy) proceeds from examples.

E.g. From "This swan is white, that swan is white, in fact every swan I have seen so far is white"

Conclude: "Every Swan is white"

NB

This may be a good way to discover hypotheses.
But it is not a valid principle of reasoning!

Mathematical induction *is a variant that is valid.*

Inductive Reasoning

Suppose we would like to reach a conclusion of the form

P(x) *for all* x (of some type)

Inductive reasoning (as understood in philosophy) proceeds from examples.

E.g. From "This swan is white, that swan is white, in fact every swan I have seen so far is white"

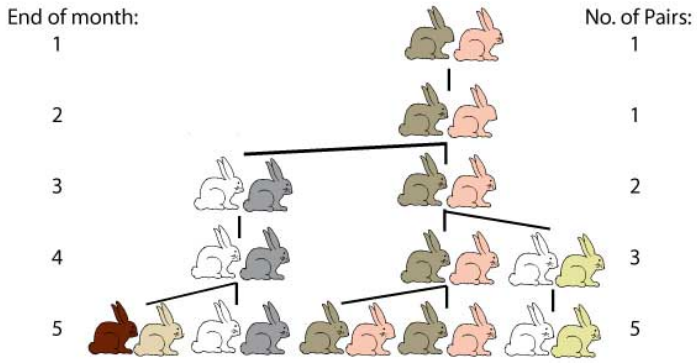
Conclude: "Every Swan is white"

NB

This may be a good way to discover hypotheses.
But it is not a valid principle of reasoning!

Mathematical induction *is a variant that is valid.*

Example



Fibonacci Numbers:

$$\text{FIB}(1) = 1$$
$$\text{FIB}(2) = 1$$
$$\text{FIB}(n) = \text{FIB}(n-1) + \text{FIB}(n-2) \quad \text{for all } n > 2$$

$\text{FIB}(1)$	1
$\text{FIB}(2)$	1
$\text{FIB}(3)$	2
$\text{FIB}(4)$	3

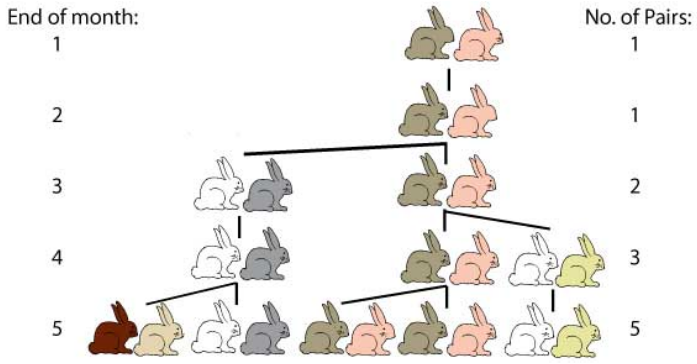
$\text{FIB}(5)$	5
$\text{FIB}(6)$	8
$\text{FIB}(7)$	13
$\text{FIB}(8)$	21

$\text{FIB}(9)$	34
$\text{FIB}(10)$	55
$\text{FIB}(11)$	89
$\text{FIB}(12)$	144

Example

Fibonacci Numbers:

$$\text{FIB}(1) = 1$$
$$\text{FIB}(2) = 1$$
$$\text{FIB}(n) = \text{FIB}(n-1) + \text{FIB}(n-2) \quad \text{for all } n > 2$$



$\text{FIB}(1)$	1
$\text{FIB}(2)$	1
$\text{FIB}(3)$	2
$\text{FIB}(4)$	3

$\text{FIB}(5)$	5
$\text{FIB}(6)$	8
$\text{FIB}(7)$	13
$\text{FIB}(8)$	21

$\text{FIB}(9)$	34
$\text{FIB}(10)$	55
$\text{FIB}(11)$	89
$\text{FIB}(12)$	144

Example

$$\text{FIB}(1) = 1$$
$$\text{FIB}(2) = 1$$
$$\text{FIB}(n) = \text{FIB}(n-1) + \text{FIB}(n-2) \quad \text{for all } n > 2$$

FIB(1)	1
FIB(2)	1
FIB(3)	2
FIB(4)	3

FIB(5)	5
FIB(6)	8
FIB(7)	13
FIB(8)	21

FIB(9)	34
FIB(10)	55
FIB(11)	89
FIB(12)	144

Claim: Every 4th Fibonacci number is divisible by 3
How can we prove this?

Mathematical Induction

Mathematical Induction is based not just on a set of examples, but also a rule for deriving new cases of $P(x)$ from cases for which P is known to hold.
General structure of reasoning by mathematical induction:

Base Case [B]: $P(a_1), P(a_2), \ldots, P(a_n)$ for some small set of examples $a_1 \ldots a_n$ (often $n = 1$)
Inductive Step [I]: A general rule showing that if $P(x)$ holds for some cases $x = x_1, \ldots, x_k$ then $P(y)$ holds for some new case y, constructed in some way from $x_1, \ldots, x_k$.

Conclusion: Starting with $a_1 \ldots a_n$ and repeatedly applying the construction of y from existing values, we can eventually construct all values in the domain of interest.

Example

Suppose we start with $x = 0$ and repeatedly apply the construction $x \mapsto x + 1$.

Then we construct values

$0, 0 + 1 = 1, 1 + 1 = 2, 2 + 1 = 3, 3 + 1 = 4, \ldots$

In the limit, this is all of $\mathbb{N}$

The corresponding principle of Mathematical Induction on $\mathbb{N}$:

Base Case [B]: $P(0)$

Inductive Step [I]: $\forall k \geq 0\,(P(k) \to P(k+1))$

Conclusion: $\forall n \in \mathbb{N}\ P(n)$

Example

Suppose we start with $x = 0$ and repeatedly apply the construction $x \mapsto x + 1$.

Then we construct values

$0, 0 + 1 = 1, 1 + 1 = 2, 2 + 1 = 3, 3 + 1 = 4, \ldots$

In the limit, this is all of $\mathbb{N}$

The corresponding principle of Mathematical Induction on $\mathbb{N}$:

Base Case [B]: $P(0)$

Inductive Step [I]: $\forall k \geq 0\,(P(k) \to P(k+1))$

Conclusion: $\forall n \in \mathbb{N}\ P(n)$

Example

Suppose we start with $x = 0$ and repeatedly apply the construction $x \mapsto x + 1$.

Then we construct values

$0, 0 + 1 = 1, 1 + 1 = 2, 2 + 1 = 3, 3 + 1 = 4, \ldots$

In the limit, this is all of $\mathbb{N}$

The corresponding principle of Mathematical Induction on $\mathbb{N}$:

Base Case [B]: $P(0)$

Inductive Step [I]: $\forall k \geq 0\,(P(k) \rightarrow P(k+1))$

Conclusion: $\forall n \in \mathbb{N}\ P(n)$

Inductive Hypothesis

To prove the Inductive Step, $P(k) \to P(k+1)$ for $k \geq 0$, we typically proceed as follows:

Assume $P(k)$, for an arbitrary $k \geq 0$

$\vdots$ (steps of reasoning, often using the assumption that $P(k)$)

Conclude $P(k+1)$.

Here $P(k)$ is called the *Inductive Hypothesis*

Example

Theorem. For all $n \in \mathbb{N}$, we have

$$P(n): \quad \sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

Proof.

[B] $P(0)$, i.e.

$$\sum_{i=0}^{0} i = \frac{0(0+1)}{2}$$

[I] $\forall k \geq 0\,(P(k) \to P(k+1))$, i.e.

$$\sum_{i=0}^{k} i = \frac{k(k+1)}{2} \quad \to \quad \sum_{i=0}^{k+1} i = \frac{(k+1)(k+2)}{2}$$

(proof?) □

Example

Theorem. For all $n \in \mathbb{N}$, we have

$$P(n): \quad \sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

Proof.

[B] $P(0)$, i.e.

$$\sum_{i=0}^{0} i = \frac{0(0+1)}{2}$$

[I] $\forall k \geq 0\,(P(k) \to P(k+1))$, i.e.

$$\sum_{i=0}^{k} i = \frac{k(k+1)}{2} \quad \to \quad \sum_{i=0}^{k+1} i = \frac{(k+1)(k+2)}{2}$$

(proof?) □

Example

Theorem. For all $n \in \mathbb{N}$, we have

$$P(n): \quad \sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

Proof.

[B] $P(0)$, i.e.

$$\sum_{i=0}^{0} i = \frac{0(0+1)}{2}$$

[I] $\forall k \geq 0\,(P(k) \to P(k+1))$, i.e.

$$\sum_{i=0}^{k} i = \frac{k(k+1)}{2} \quad \to \quad \sum_{i=0}^{k+1} i = \frac{(k+1)(k+2)}{2}$$

(proof?) □

Example (cont'd)

Proof.

Inductive step [I]:

$$\sum_{i=0}^{k+1} i = \left(\sum_{i=0}^{k} i\right) + (k+1)$$

$$= \frac{k(k+1)}{2} + (k+1) \qquad \text{(by the inductive hypothesis)}$$

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

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

Variations

1. Induction from m upwards
2. Induction steps >1
3. Strong induction
4. Backward induction
5. Forward-backward induction
6. Structural induction

Induction From m Upwards

If

[B] $P(m)$

[I] $\forall k \geq m\,(P(k) \rightarrow P(k+1))$

then

[C] $\forall n \geq m\,(P(n))$

Example

Theorem. For all $n \geq 1$, the number $8^n - 2^n$ is divisible by 6.

[B] $8^1 - 2^1$ is divisible by 6
[I] if $8^k - 2^k$ is divisible by 6, then so is $8^{k+1} - 2^{k+1}$, for all $k \geq 1$

Prove [I] using the "trick" to rewrite 8^{k+1} as $8 \cdot (8^k - 2^k + 2^k)$ which allows you to apply the Ind. Hyp. on $8^k - 2^k$

Exercise

Consider an **increasing** function $f : \mathbb{N} \longrightarrow \mathbb{N}$
i.e., $\forall m, n\,(m \leq n \;\rightarrow\; f(m) \leq f(n))$
and a function $g : \mathbb{N} \longrightarrow \mathbb{N}$ such that

- $f(0) < g(0)$
- $f(1) = g(1)$
- if $f(k) \geq g(k)$ then $f(k+1) \geq g(k+1)$, for all $k \in \mathbb{N}$

Always true, false or could be either?
(a) $f(n) > g(n)$ for all $n \geq 1$ — false
(b) $f(n) > g(n)$ for some $n \geq 1$ — could be either
(c) $f(n) \geq g(n)$ for all $n \geq 1$ — true
(d) g is decreasing $(m \leq n \;\rightarrow\; g(m) \geq g(n))$ — could be either

Exercise

Consider an **increasing** function $f : \mathbb{N} \longrightarrow \mathbb{N}$
i.e., $\forall m, n\,(m \leq n \;\rightarrow\; f(m) \leq f(n))$
and a function $g : \mathbb{N} \longrightarrow \mathbb{N}$ such that

- $f(0) < g(0)$
- $f(1) = g(1)$
- if $f(k) \geq g(k)$ then $f(k+1) \geq g(k+1)$, for all $k \in \mathbb{N}$

Always true, false or could be either?
(a) $f(n) > g(n)$ for all $n \geq 1$ — false
(b) $f(n) > g(n)$ for some $n \geq 1$ — could be either
(c) $f(n) \geq g(n)$ for all $n \geq 1$ — true
(d) g is decreasing ($m \leq n \;\rightarrow\; g(m) \geq g(n)$) — could be either

Induction Steps $\ell > 1$

If

[B] $P(m)$

[I] $P(k) \to P(k+\ell)$ for all $k \geq m$

then

[C] $P(n)$ for every ℓ'th $n \geq m$

Example

$$\begin{aligned}\textsc{fib}(1) &= 1\\ \textsc{fib}(2) &= 1\\ \textsc{fib}(n) &= \textsc{fib}(n-1) + \textsc{fib}(n-2)\end{aligned}$$

Every 4th Fibonacci number is divisible by 3.

[B] $\textsc{fib}(4) = 3$ is divisible by 3

[I] if $3 \mid \textsc{fib}(k)$, then $3 \mid \textsc{fib}(k+4)$, for all $k \geq 4$

Prove [I] by rewriting $\textsc{fib}(k+4)$ in such a way that you can apply the Ind. Hyp. on $\textsc{fib}(k)$

Strong Induction

This is a version in which the inductive hypothesis is stronger. Rather than using the fact that $P(k)$ holds for a single value, we use *all* values up to k.

If

[B] $P(m)$

[I] $[P(m) \wedge P(m+1) \wedge \ldots \wedge P(k)] \rightarrow P(k+1)$ for all $k \geq m$

then

[C] $P(n)$, for all $n \geq m$

Example

Claim: All integers ≥ 2 can be written as a product of primes.

[B] 2 is a product of primes

[I] If all x with $2 \leq x \leq k$ can be written as a product of primes, then $k+1$ can be written as a product of primes, for all $k \geq 2$

Proof for [I]?

Negative Integers, Backward Induction

NB

Induction can be conducted over any subset of $\mathbb{Z}$ with least element. Thus m can be negative; eg. base case $m = -10^6$.

NB

One can apply induction in the 'opposite' direction $p(m) \to p(m-1)$. It means considering the integers with the opposite ordering where the next number after n is $n-1$. Such induction would be used to prove some $p(n)$ for all $n \leq m$.

NB

Sometimes one needs to reason about all integers $\mathbb{Z}$. This requires two separate simple induction proofs: one for $\mathbb{N}$, another for $-\mathbb{N}$. They both would start form some initial values, which could be the same, e.g. zero. Then the first proof would proceed through positive integers; the second proof through negative integers.

Forward-Backward Induction

Idea

To prove $P(n)$ *for all* $n \geq k_0$

- *verify* $P(k_0)$
- *prove* $P(k_i)$ *for infinitely many* $k_0 < k_1 < k_2 < k_3 < \ldots$
- *fill the gaps*
  $P(k_1) \to P(k_1 - 1) \to P(k_1 - 2) \to \ldots \to P(k_0 + 1)$
  $P(k_2) \to P(k_2 - 1) \to P(k_2 - 2) \to \ldots \to P(k_1 + 1)$
  $\ldots\ldots\ldots$

Example

Claim

Binary search in an (ordered) list of $n-1$ elements requires no more than $\lceil \log_2 n \rceil$ comparisons.

Proof.

(i) it holds for $n = 1$

(ii) if it holds for k then it holds for $2k$,
thus true for $2,\ 4,\ 8,\ 16,\ \ldots$

(iii) if it holds for 2^i then it holds for $2^i - 1,\ 2^i - 2, \ldots, 2^{i-1} + 1$,
thus true for all n.

□

Forward-Backward Induction: Formalisation

[B] $P(k_0)$

[I] if $P(k)$ then $P(k')$ for *some* $k' > k$

[I] and [B] alone imply $P(k_i)$ for infinitely many $k_0 < k_1 < k_2 < \ldots$

[D] $P(k) \to P(k-1)$ for all k between k_i's and k_{i+1}'s (downward step)

[C] $\forall n \geq k_0\,(P(n))$

NB

This form of induction is extremely important for the analysis of algorithms.

Various Inductive Arguments

Induction by cases

Consider separately various subsets $S_1, S_2, \ldots \subset \mathbb{N}$, eg. odd and even numbers, making sure that they jointly cover all of $\mathbb{N}$.
Complete the proof (by induction) separately for each subset.

Example

Any amount $n \in \mathbb{N}$ greater than \$1 can be paid using units ('coins') of \$2 and \$3.
To prove it we conduct **two** inductive arguments: one over the even numbers, the other over the odd numbers.
Equivalently, we can split the proof into **three** cases: one for the numbers divisible by 3, one for those with remainder 1 and one for those with remainder 2.

NB

One can use the same type of argument for any two coin values $m and $n if $\gcd(m, n) = 1$ *and amount* $> \$(mn - m - n)$.
The proof splits into m *cases: one for numbers divisible by* m*, then one for those having remainder 1 mod* m*, then . . .*
Each case uses a similar inductive argument: if an amount $p can be paid using coins $m,$n, then so can be the amount $\$(p + m)$.

Infinite Descent

To prove that $Q(n)$, for all $n \geq m$, show

- $\neg Q(n) \to \neg Q(n')$ for some $n' < n$
- there cannot be arbitrarily small n s.t. $Q(n)$ is false; in particular the "base case" $Q(m)$ is *true*

This amounts to a proof by contradiction: to verify $\forall n\, Q(n)$ we assume (provisionally) its negation $\exists n \neg Q(n)$ and proceed to show that there would have to exist a smaller n' such that $\neg Q(n')$. Usually the conditions of the problem make it clear that no such infinite decreasing chain $\ldots < n'' < n' < n$ can possibly exist.

Example

Theorem

For a planar, connected graph let F be the number of faces (enclosures) including the exterior face, E the number of edges, and V the number of vertices.
Euler's formula *holds:*

$$V - E + F = 2 \qquad (EF)$$

Proof.

Suppose $G = (V, E)$ is a planar connected graph that **violates** (EF).

First observe that G must have an edge because it is connected and the graph with just one vertex satisfies (EF).

If G has an outside edge, that is, an edge separating the exterior face from an interior face, then removing that edge results in a smaller (planar, connected) graph, also violating (EF) because both E and F are reduced by 1.

If G has no outside edge then it has a vertex v of degree 1. Removing v reduces both V and E by 1 while F remains unchanged. It follows that we again found a smaller (planar, connected) graph violating (EF). □

Structural Induction

The induction schemes can be applied not only to natural numbers (and integers) but to any partially ordered set in general.

The basic approach is always the same — we need to verify that

- **[I]** for any given object, if the property in question holds for all its predecessors ('smaller' objects) then it holds for the object itself
- **[B]** the property holds for all minimal objects — objects that have no predecessors; they are usually very simple objects allowing immediate verification

Example: Induction on Rooted Trees

We write $T = \langle r;\ T_1, T_2, \ldots, T_k \rangle$ for a tree T with root r and k subtrees at the root $T_1, \ldots, T_k$

If

[B]	$p(\langle v; \rangle)$	for trees with only a root
[I]	$p(T_1) \wedge \ldots \wedge p(T_k) \rightarrow p(T)$	for all trees $T = \langle r;\ T_1, T_2, \ldots, T_k \rangle$

then

[C] $p(T)$ for every tree T

Example

Theorem

In any rooted tree the number of vertices is one more than the number of edges.

Proof.

[B] If $T = \langle v; \rangle$ then $v(T) = 1$ and $e(T) = 0$

Example

Theorem

In any rooted tree the number of vertices is one more than the number of edges.

Proof.

[B] If $T = \langle v; \rangle$ then $v(T) = 1$ and $e(T) = 0$

[I] If $T = \langle r;\ T_1, T_2, \ldots, T_k \rangle$ then
$v(T) = 1 + \sum_{i=1}^{k} v(T_i)$ and $e(T) = k + \sum_{i=1}^{k} e(T_i)$

Example

Theorem

In any rooted tree the number of vertices is one more than the number of edges.

Proof.

[B] If $T = \langle v; \rangle$ then $v(T) = 1$ and $e(T) = 0$

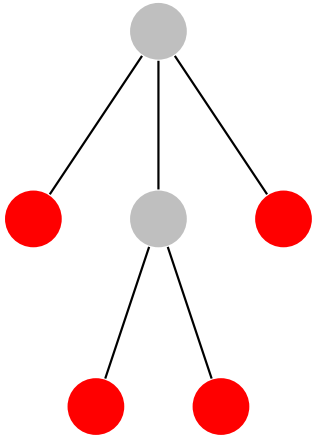
[I] If $T = \langle r;\ T_1, T_2, \ldots, T_k \rangle$ then
$v(T) = 1 + \sum_{i=1}^{k} v(T_i)$ and $e(T) = k + \sum_{i=1}^{k} e(T_i)$
From the Ind. Hyp. on $T_1, \ldots, T_k$ it follows that
$\sum_{i=1}^{k} v(T_i) = \sum_{i=1}^{k} (e(T_i) + 1) = (\sum_{i=1}^{k} e(T_i)) + k$
Therefore
$v(T) = 1 + (\sum_{i=1}^{k} e(T_i)) + k = 1 + e(T)$

Example

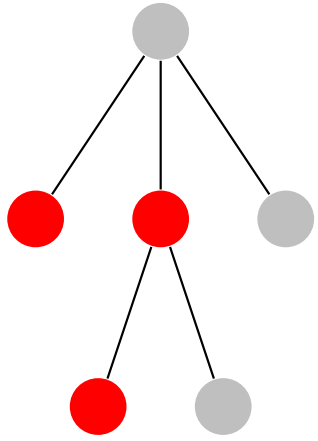
Theorem

In any rooted tree the number of leaves is one more than the number of vertices that have a right sibling.

Proof: exercise



4 leaves



3 vertices with right sibling

Recursive Definitions

They comprise basis (B) and recursive process (R).
A sequence is recursively defined when (typically)
(B) some initial terms are specified, perhaps only the first one;
(R) later terms stated as functional expressions of the earlier terms.

Examples

Factorial:

(B) $\quad 0! = 1$

(R) $\quad (n+1)! = (n+1) \cdot n!$

Fibonacci numbers:

(B) $\quad \text{FIB}(1) = 1$

(B) $\quad \text{FIB}(2) = 1$

(R) $\quad \text{FIB}(n) = \text{FIB}(n-1) + \text{FIB}(n-2)$

NB

(R) also called **recurrence formula**

Inductive Proofs About Recursive Definitions

Proofs about recursively defined function very often proceed by a mathematical induction following the structure of the definition.

Example

$\forall n \in \mathbb{N}\left(n! \geq 2^{n-1}\right)$

Proof.

[B] $0! = 1 \geq \frac{1}{2} = 2^{0-1}$

[I] Assume $n \geq 1$.

$$\begin{aligned} (n+1)! = n! \cdot (n+1) &\geq 2^{n-1} \cdot (n+1) && \text{by Ind. Hyp.} \\ &\geq 2^{n-1} \cdot 2 && \text{by } n \geq 1 \\ &= 2^n \end{aligned}$$

□

Exercise

4.4.2 Define $s_1 = 1$ and $s_{n+1} = \frac{1}{1+s_n}$ for $n \geq 1$
Then $s_1 = 1,\ s_2 = \frac{1}{2},\ s_3 = \frac{2}{3},\ s_4 = \frac{3}{5},\ s_5 = \frac{5}{8}, \ldots$
The numbers in numerator and denominator remind one of the Fibonacci sequence.
Prove by induction that

$$s_n = \frac{\mathrm{FIB}(n)}{\mathrm{FIB}(n+1)}$$

Example (continued)

Furthermore,

$$\lim_{n\to\infty} s_n = \frac{2}{\sqrt{5}+1} = \frac{\sqrt{5}-1}{2} \approx 0.6$$

This is obtained by showing (using induction!) that

$$\text{FIB}(n) = \frac{1}{\sqrt{5}}(r_1^n - r_2^n)$$

where $r_1 = \frac{1+\sqrt{5}}{2}$ and $r_2 = \frac{1-\sqrt{5}}{2}$

Exercise

4.4.4 (a) Give a recursive definition for the sequence

$$(2,\ 4,\ 16,\ 256,\ \ldots)$$

To generate $a_n = 2^{2^n}$ use $a_n = (a_{n-1})^2$.
(The related "Fermat numbers" $F_n = 2^{2^n} + 1$ are used in cryptography.)

(b) Give a recursive definition for the sequence

$$(2,\ 4,\ 16,\ 65536,\ \ldots)$$

To generate a "stack" of n 2's use $b_n = 2^{b_{n-1}}$.
(These are *Ackermann's numbers*, first used in logic. The inverse function is extremely slow growing; it is important for the analysis of several data organisation algorithms.)

Exercise

4.4.4 (a) Give a recursive definition for the sequence

$$(2,\ 4,\ 16,\ 256,\ \ldots)$$

To generate $a_n = 2^{2^n}$ use $a_n = (a_{n-1})^2$.
(The related "Fermat numbers" $F_n = 2^{2^n} + 1$ are used in cryptography.)

(b) Give a recursive definition for the sequence

$$(2,\ 4,\ 16,\ 65536,\ \ldots)$$

To generate a "stack" of n 2's use $b_n = 2^{b_{n-1}}$.
(These are *Ackermann's numbers*, first used in logic. The inverse function is extremely slow growing; it is important for the analysis of several data organisation algorithms.)

Correctness of Recursive Definition

A recurrence formula is correct if the computation of any later term can be reduced to the initial values given in (B).

Example (Incorrect definition)

- Function $g(n)$ is defined recursively by

$$g(n) = g(g(n-1)-1)+1, \qquad g(0) = 2.$$

The definition of $g(n)$ is incomplete — the recursion may not terminate:
Attempt to compute $g(1)$ gives

$$g(1) = g(g(0)-1)+1 = g(1)+1 = \ldots = g(1)+1+1+1\ldots$$

When implemented, it leads to an overflow; most static analyses cannot detect this kind of ill-defined recursion.

Example (continued)

However, the definition could be repaired. For example, we can add the specification specify $g(1) = 2$.

Then $g(2) = g(2-1) + 1 = 3$,
$g(3) = g(g(2)-1) + 1 = g(3-1) + 1 = 4$,
$\ldots$

In fact, by induction $\ldots$ $g(n) = n + 1$

This illustrates a very important principle: the boundary (limiting) cases of the definition are evaluated *before* applying the recursive construction.

Example

Function $f(n)$ is defined by

$$f(n) = f(\lceil n/2 \rceil), \quad f(0) = 1$$

When evaluated for $n = 1$ it leads to

$$f(1) = f(1) = f(1) = \ldots$$

This one can also be repaired. For example, one could specify that $f(1) = 1$.
This would lead to a constant function $f(n) = 1$ for all $n \geq 0$.

Mutual Recursion

Several more sophisticated programs employ a technique of two procedures calling each other. Of course, it should be designed so that each consecutive call refers to ever smaller parameters, so that the entire process terminates. This method is often used in computer graphics, in particular for generating fractal images (basis of various imaginary landscapes, among others).

Summary

- Mathematical induction:
  base case(s), inductive hypothesis $P(k)$,
  inductive step $\forall k\,(P(k) \to P(k+1))$, conclusion
- Variations:
  strong ind., forward-backward ind., ind. by cases,
  structural ind.
- Recursive definitions