Discrete Mathematics

Induction Principles 2

Aleksandar Dimovski

adim@itu.dk

23/10/2014 [week 8]

Today's lecture

1. Defining Sequences Recursively
2. Solving Recurrence Relations by Iteration
3. Solving Recurrence Relations of Special Form
4. General Recursive Definitions and Structural Induction

Outline

1. Defining Sequences Recursively
2. Solving Recurrence Relations by Iteration
3. Solving Recurrence Relations of Special Form
4. General Recursive Definitions and Structural Induction

Defining Sequences?

A sequence $(a_0, a_1, a_2, \ldots)$ can be defined in a variety of different ways:

- informal (ambiguous) way: write the first few terms explicitly with the expectation that the general pattern will be obvious, e.g. "3, 5, 7, ..."
- formal (direct) way: give an explicit formula for the $n-$th term of the sequence, e.g. $a_n = \frac{(-1)^n}{n+1}$ for all $n \geq 0$
- formal (recursive) way: 1). give a recurrence relation that defines each later term in the sequence by reference to earlier terms; 2) give one or more initial values for the sequence

Defining Sequences Recursively

Definition

A recursive definition of a sequence $a_0, a_1, a_2, \ldots$ consists of:

- **recurrence relation** that relates each term a_k to certain of its predecessors $a_{k-1}, a_{k-2}, \ldots, a_0$
- **initial conditions** that specify the values of the first m terms of the sequence $a_0, a_1, \ldots, a_m (m \geq 0)$

Example

Consider the following recursively defined sequence:

(1) $c_k = c_{k-1} + k \cdot c_{k-2} + 1$ for $k \geq 2$ (**recurrence relation**)

(2) $c_0 = 1$ and $c_1 = 2$ (**initial conditions**)

Find $c_2, c_3, c_4, \ldots$?

$c_2 = c_1 + 2 \cdot c_0 + 1 = \ldots = 5$

$c_3 = c_2 + 3 \cdot c_1 + 1 = \ldots = ?$

Defining Sequences Recursively

Note that a recursively defined sequence is determined by **both**: recurrence relation and initial conditions.

Let $a_1, a_2, a_3, \ldots$ and $b_1, b_2, b_3, \ldots$ are two sequences defined by the **same** recurrence relation but the initial conditions are **different**.
Then $a_1, a_2, a_3, \ldots$ and $b_1, b_2, b_3, \ldots$ are **two different sequences**.

Example

Consider $a_1, a_2, a_3, \ldots$ and $b_1, b_2, b_3, \ldots$ recursively defined as:

(1) $a_k = 3 \cdot a_{k-1}$ and $b_k = 3 \cdot b_{k-1}$ for $k \geq 2$ (**recurrence relation**)

(2) $a_1 = 2$ and $b_1 = 3$ (**initial conditions**)

Find $a_2, a_3, a_4, \ldots ?$ and $b_2, b_3, b_4, \ldots ?$
$a_2 = 3 \cdot a_1 = 6 \quad a_3 = 3 \cdot a_2 = ? \quad a_4 = ?$
$b_2 = 3 \cdot b_1 = 9 \quad b_3 = 3 \cdot b_2 = ? \quad b_4 = ?$

Defining Sequences Recursively

We can show that a sequence given by an explicit formula satisfies a certain recurrence relation.

Let $a_0, a_1, a_2, \ldots$ be defined by the formula $a_n = 3 \cdot n + 1$ for $n \geq 0$. Show that this sequence satisfies the recurrence relation $a_k = a_{k-1} + 3$ for $k \geq 1$.

We first obtain a_k and a_{k-1} from the direct formula:
$a_k = 3 \cdot k + 1$
$a_{k-1} = 3 \cdot (k-1) + 1$

Then we start from the right-hand side of the recurrence relation and transform it into the left-hand side. For any $k \geq 1$, we have:

$$\begin{aligned} & a_{k-1} + 3 \\ & \quad = 3 \cdot (k-1) + 1 + 3 \\ & \quad = 3 \cdot k - 3 + 1 + 3 \\ & \quad = 3 \cdot k + 1 \\ & \quad = a_k \end{aligned}$$

Defining Sequences Recursively

The Fibonacci Sequence

A single pair of rabbits is born at the beginning of a year. Assume:

- Rabbit pairs are not fertile during their first month of life but thereafter give birth to one new pair at the end of every month.
- No rabbits die.

How many rabbits are there at the end of the year?

Let F_n be the number of rabbit pairs at the end of month n; for $n \geq 1$.
Then we have:
$F_0 = 1$; $F_1 = 1$ (initial conditions); and
$F_k = F_{k-1} + F_{k-2}$ for $k \geq 2$ (recurrence relation) Why?

$F_2 = F_1 + F_0 = 2$
$F_3 = F_2 + F_1 = 3$
$F_4 = F_3 + F_2 = 5$
$\ldots$
$F_{12} = F_{11} + F_{10} = ?$

Outline

1. Defining Sequences Recursively
2. Solving Recurrence Relations by Iteration
3. Solving Recurrence Relations of Special Form
4. General Recursive Definitions and Structural Induction

Solving Recurrence Relations by Iteration

Suppose that a sequence $a_0, a_1, a_2, \ldots$ is defined by recurrence relation and initial conditions.

It is often helpful to find an **explicit formula** for the sequence, especially if we need to compute terms with very large subscripts.
Such an explicit formula is called a **solution** to the recurrence relation.

The most basic method for finding an explicit formula is **iteration**:
we start from the initial conditions and calculate successive terms of the sequence until we see a pattern developing.
At that point we guess the explicit formula.

Solving Recurrence Relations by Iteration – Example

Let $a_0, a_1, a_2, \ldots$ be defined recursively by:

- $a_k = a_{k-1} + 2$ for $k \geq 1$
- $a_0 = 1$

Here is how the process works for the given sequence:

$$\begin{aligned}
a_0 &= 1 = 1 + 0 \cdot 2 \\
a_1 &= a_0 + 2 = 1 + 0 \cdot 2 + 2 = 1 + 1 \cdot 2 \\
a_2 &= a_1 + 2 = 1 + 1 \cdot 2 + 2 = 1 + 2 \cdot 2 \\
a_3 &= a_2 + 2 = 1 + 2 \cdot 2 + 2 = 1 + 3 \cdot 2 \\
a_4 &= a_3 + 2 = 1 + 3 \cdot 2 + 2 = 1 + 4 \cdot 2 \\
&\ldots
\end{aligned}$$

Guess: $a_n = 1 + n \cdot 2 = 1 + 2 \cdot n$

Solving Recurrence Relations by Iteration

Definition

A sequence $a_0, a_1, a_2, \ldots$ is called an **arithmetic sequence** if there is a constant d such that

$$a_k = a_{k-1} + d \quad \text{for all} \quad k \geq 1$$

It follows that

$$a_n = a_0 + d \cdot n \quad \text{for all} \quad n \geq 0$$

Definition

A sequence $a_0, a_1, a_2, \ldots$ is called an **geometric sequence** if there is a constant r such that

$$a_k = r \cdot a_{k-1} \quad \text{for all} \quad k \geq 1$$

It follows that

$$a_n = a_0 \cdot r^n \quad \text{for all} \quad n \geq 0$$

Solving Recurrence Relations by Iteration

Checking the Correctness of a Solution to a Recurrence Relation.

Let $m_0, m_1, m_2, \ldots$ be a geometric sequence defined by:

$$m_k = 2 \cdot m_{k-1}, \text{ for } k \geq 2$$
$$m_1 = 2$$

then $m_n = 2^n$ for all $n \geq 1$.

Proof of Correctness:
Let the property $P(n)$ be the equation: $m_n = 2^n$.
We will use mathematical induction to prove that $P(n)$ is true for all $n \geq 1$.

Show that $P(1)$ is true.
We must show that $m_1 = 2^1$.
The left-hand side of $P(1)$ is: $m_1 = 2$
The right-hand side of $P(1)$ is: $2^1 = 2$
Thus the two sides of $P(1)$ are equal, and hence $P(1)$ is true.

Solving Recurrence Relations by Iteration

Checking the Correctness of a Solution to a Recurrence Relation.
The sequence is defined by: $m_k = 2 \cdot m_{k-1},\ \text{for } k \geq 2$

Proof of Correctness:
The property $P(n)$ is: $m_n = 2^n$.

Show that if $P(k)$ is true then $P(k+1)$ is true for $k \geq 1$.
Suppose that $P(k)$ is true, i.e. $m_k = 2^k$.
We must show that $P(k+1)$ is true, i.e. $m_{k+1} = 2^{k+1}$.
The left-hand side of $P(k+1)$ is:

$$\begin{aligned} m_{k+1} =& 2 \cdot m_k \text{ (by definition of the sequence } m) \\ &= 2 \cdot 2^k \text{ (by using } P(k)) \\ &= 2^{k+1} \end{aligned}$$

which is equal to the right-hand side of $P(k+1)$.

Outline

1. Defining Sequences Recursively
2. Solving Recurrence Relations by Iteration
3. Solving Recurrence Relations of Special Form
4. General Recursive Definitions and Structural Induction

Solving Recurrence Relations of Special Form

Definition

A **second-order linear homogeneous recurrence relation with constant coefficients** is a recurrence relation of the form:

$$a_k = A \cdot a_{k-1} + B \cdot a_{k-2} \ \text{ for all } \ k \geq \ \text{some fixed int}$$

where A and B are fixed real numbers with $B \neq 0$.

Check whether the following relations are in the above form:

$$a.\ a_k = 3 \cdot a_{k-1} + 2 \cdot a_{k-2}$$
$$b.\ b_k = b_{k-1} + b_{k-2} + b_{k-3}$$
$$c.\ c_k = c_{k-1}^2 + c_{k-1} \cdot c_{k-2}$$

Solving Recurrence Relations of Special Form

Theorem (Distinct-Roots Theorem)

Suppose a sequence $a_0, a_1, a_2, \ldots$ satisfies a recurrence relation:

$$a_k = A \cdot a_{k-1} + B \cdot a_{k-2}$$

for some real numbers A and B with $B \neq 0$ and $k \geq 2$. If the characteristic equation:

$$t^2 - A \cdot t - B = 0$$

has two distinct roots r and s, then $a_0, a_1, a_2, \ldots$ is given by the explicit formula

$$a_n = C \cdot r^n + D \cdot s^n$$

where C and D are numbers whose values are determined by a_0 and a_1.

Solving Recurrence Relat. of Special Form – Example

Find a sequence that satisfies the recurrence relation

$$a_k = a_{k-1} + 2 \cdot a_{k-2} \text{ for all } k \geq 2$$

and also satisfies the initial conditions: $a_0 = 1$ and $a_1 = 8$.

Solution.
The characteristic equation is: $t^2 - t - 2 = 0$
Since $t^2 - t - 2 = (t-2) \cdot (t+1)$
it has two roots: 2 and -1.
Thus, the sequence $a_0, a_1, a_2, \ldots$ defined by

$$a_n = C \cdot 2^n + D \cdot (-1)^n$$

also satisfies the recurrence relation.
We need to find the values for C and D?
$a_0 = 1 = C \cdot 2^0 + D \cdot (-1)^0 = C + D$
$a_1 = 8 = C \cdot 2^1 + D \cdot (-1)^1 = 2 \cdot C - D$
The solution is: $C = 3$ and $D = -2$.

It follows that the sequence $a_0, a_1, a_2, \ldots$ is given by:

$$a_n = 3 \cdot 2^n - 2 \cdot (-1)^n$$

for $n \geq 0$.

Solving Recurrence Relations of Special Form

Theorem (Single-Root Theorem)

Suppose a sequence $a_0, a_1, a_2, \ldots$ *satisfies a recurrence relation:*

$$a_k = A \cdot a_{k-1} + B \cdot a_{k-2}$$

for some real numbers A *and* B *with* $B \neq 0$ *and* $k \geq 2$*. If the characteristic equation:*

$$t^2 - A \cdot t - B = 0$$

has a single root r*, then* $a_0, a_1, a_2, \ldots$ *is given by the explicit formula*

$$a_n = C \cdot r^n + D \cdot n \cdot r^n$$

where C *and* D *are numbers whose values are determined by* a_0 *and* a_1*.*

Outline

1. Defining Sequences Recursively
2. Solving Recurrence Relations by Iteration
3. Solving Recurrence Relations of Special Form
4. General Recursive Definitions and Structural Induction

Recursively Defined Sets

To define a set of objects recursively, we need to:
1) identify a few core objects of the set; and
2) give rules showing how to build new objects from the old.

Recursive definition of a set consists of:

(I) BASE: A few core objects that belong to the set

(II) RECURSION: A collection of rules indicating how to form new objects from those already known to be in the set.

(III) RESTRICTION: A statement that no objects belong to the set other than those generated from (I) and (II).

Recursively Defined Sets – Example 1

The set of Boolean expression over an alphabet defined recursively:

(I) BASE: Each symbol of the alphabet is a Boolean expression.

(II) RECURSION: If P and Q are Boolean expressions, then so are: $(P \wedge Q)$; $(P \vee Q)$; $\sim P$

(III) RESTRICTION: There are no Boolean expressions over an alphabet other than those obtained from (I) and (II).

Show that

$$(\sim (p \wedge q) \vee (\sim r \wedge p))$$

is a Boolean expression over the English alphabet.

Solution.
(1) By (I), p, q, and r are Boolean expressions.
(2) ...

Recursively Defined Sets – Example 2

The set P of legal configurations of parentheses is defined recursively as follows:

(I) BASE: () is in P.

(II) RECURSION:
  a. If E is in P, so is (E).
  b. If E and F are in P, so is EF.

(III) RESTRICTION: No configurations of parentheses are in P other than those obtained from (I) and (II).

Show that: (())() is in P.

Solution.
(1) By (I), () is in P.
(2) By (1) and (IIa), (()) is in P.
(3) By (2), (1) and (IIb), (())() is in P.

Structural Induction

Let S be a recursively defined set. To prove that every object in S satisfies a property:

1. Show that each object in the BASE for S satisfies the property.
2. Show that for each rule in the RECURSION, if the rule is applied to objects in S that satisfy the property, then the objects generated by the rule also satisfy the property.

Since all objects in S are obtained through BASE and RECURSION; it must be the case that every object in S satisfies the property.

Structural Induction – Example 1

Prove that every legal configuration of parentheses in P contains an equal number of left and right parentheses.

Proof by structural induction.
Let the property be that a given configuration of parentheses in P has an equal number of left and right parentheses.

Show that each object in BASE for P satisfies the property.
The only object in BASE for P is (), which has 1 left and 1 right parenthesis.

Structural Induction – Example 1

Proof by structural induction.
Let the property be that a given configuration of parentheses in P has an equal number of left and right parentheses.

Show that each object in BASE for P satisfies the property.
The only object in BASE for P is (), which has 1 left and 1 right parenthesis.

Show that for each rule in RECURSION for P, if the rule is applied to an object that satisfies the property, then the object generated by the rule also satisfies the property.
Suppose E has an equal number of left and right parentheses. When rule (IIa) is applied to E, the result is (E), so both the number of left and right parentheses is increased by one. Thus, they remain equal.
Suppose E and F have equal numbers of left and right parentheses. Say E has m left and right parentheses, and F has n left and right parentheses. When rule (IIb) is applied to E, the result is EF, which has an equal number, namely $n+m$, of left and right parentheses.
Therefore every object in P has an equal number of left and right parentheses.

Structural Induction – Example 2

Define a set of integers S recursively as follows:

(I) BASE: $0 \in P$.

(II) RECURSION: If $s \in S$, then
  a. $s + 3 \in S$
  b. $s - 3 \in S$

(III) RESTRICTION: Nothing is in S other than those integers obtained from (I) and (II).

Show that: $9 \in P$; and $-3 \in S$.

Use structural induction to show that every integer in S is divisible by 3.

Recursive Functions

A function is defined **recursively** if its rule of definition refers to itself.
Example: McCarthy's 91 Function

$$M(n) = \begin{cases} n - 10 & \text{if } n > 100 \\ M(M(n+11)) & \text{if } n \leq 100 \end{cases}$$

for all positive integers n.
Find $M(99)$? What about $M(100)$ and $M(98)$?