Sequences and Mathematical Induction

Lecture 16 / Spring 2015
State University of New York, Korea
Instructor: Dr. Ilchul Yoon

Administrivia

- Mid-Term #2 on May 7
  - Study Chapter 4 and Chapter 5

- Expect Homework #4 this weekend!!

- Makeup class scheduled on 5/8 (Friday)
Tower of Hanoi

- Question worth 10,000 francs (about $34,000 US today) to anyone who could move a tower of 64 disks by hand while following the rules of the game:
  - Rule 1: Move one by one
  - Rule 2: never place a larger one on top of a smaller one.

Assuming that you transferred the disks as efficiently as possible, how many moves would be required to win the prize?

Think recursively…

**somehow or other**, assume you found the most efficient way possible to transfer a tower of $k - 1$ disks one by one from one pole to another, obeying the restriction that you never place a larger disk on top of a smaller one.

Then, what is the most efficient way to transfer a tower of $k$ disks from one pole to another?
Tower of Hanoi

Step 1
- Transfer the top $k - 1$ disks from pole $A$ to pole $B$. If $k > 2$, execution of this step will require a number of moves of individual disks among the three poles. But the point of thinking recursively is not to get caught up in imagining the details of how those moves will occur.

Step 2
- Move the bottom disk from pole $A$ to pole $C$.

Step 3
- Transfer the top $k - 1$ disks from pole $B$ to pole $C$. (Again, if $k > 2$, execution of this step will require more than one move.)
Tower of Hanoi

- Recurrence relation

\[ m_k = m_{k-1} + 1 + m_{k-1} \]
\[ = 2m_{k-1} + 1 \quad \text{for all integers } k \geq 2. \]

initial condition \( m_1 = 1 \)
\( m_2 = 2m_1 + 1 = 2 \cdot 1 + 1 = 3 \)
\( m_3 = 2m_2 + 1 = 2 \cdot 3 + 1 = 7 \)
\( m_4 = 2m_3 + 1 = 2 \cdot 7 + 1 = 15 \)
\( m_5 = 2m_4 + 1 = 2 \cdot 15 + 1 = 31 \)
\( m_6 = 2m_5 + 1 = 2 \cdot 31 + 1 = 63 \)

- \( m_{64} \cong 1.844674 \times 10^{19} \) moves.
- Assuming 1 move per second, it takes 584.5 billion years.

Tower of Hanoi - Correct of Solution to a Recurrence Relation

- Let \( m \) be the minimum number of moves needed to transfer a tower of \( k \) disks from one pole to another. Prove the following statement.

If \( m_1, m_2, m_3, \ldots \) is the sequence defined by
\[ m_k = 2m_{k-1} + 1 \quad \text{for all integers } k \geq 2, \text{ and } m_1 = 1, \]
then \( m_n = 2^n - 1 \) for all integers \( n \geq 1. \)
Let $m_1, m_2, m_3, \ldots$ be the sequence defined by specifying that $m_1 = 1$ and $m_k = 2m_{k+1} + 1$ for all integers $k \geq 2$, and let the property $P(n)$ be the equation

$$m_n = 2^n - 1 \quad \leftarrow P(n)$$

We will use mathematical induction to prove that for all integers $n \geq 1$, $P(n)$ is true.

**Base step:** $m_1 = 2^1 - 1. \quad \leftarrow P(1)$
- $m_1 = 1$ from the definition of the sequence, and is equal to the value of the RHS of $P(1)$

**Inductive step:**
- Show that for all integers $k \geq 1$, if $P(k)$ is true then $P(k + 1)$ is also true:
  - Inductive hypothesis: [Suppose that $P(k)$ is true for a particular but arbitrarily chosen integer $k \geq 1$, That is:]
  - Suppose that $k$ is any integer with $k \geq 1$ such that
    $$m_k = 2^k - 1. \quad \leftarrow P(k)$$
  - [We must show that $P(k + 1)$ is true, That is:]
    $$m_{k+1} = 2^{k+1} - 1. \quad \leftarrow P(k + 1)$$
Tower of Hanoi - Correct of Solution to a Recurrence Relation

* But the left-hand side of $P(k + 1)$ is

\[
m_{k+1} = 2m_{k+1-1} + 1
= 2m_k + 1
= 2(2^k - 1) + 1
= 2^{k+1} - 2 + 1
= 2^{k+1} - 1
\]

by definition of $m_1, m_2, m_3, \ldots$

by substitution from the inductive hypothesis

by the distributive law and the fact that $2 \cdot 2^k = 2^{k+1}$

by basic algebra

* which equals the right-hand side of $P(k + 1)$. (Since the basis and inductive steps have been proved, it follows by mathematical induction that the given formula holds for all integers $n \geq 1$.)

---

Recursive Definitions of Sum and Product

* The summation from $i=1$ to $n$ of a sequence is defined using recursion:

\[
\sum_{i=1}^{n} a_i = a_1 \quad \text{and} \quad \sum_{i=1}^{n} a_i = \left( \sum_{i=1}^{n-1} a_i \right) + a_n, \quad \text{if } n > 1.
\]

* The product from $i=1$ to $n$ of a sequence is defined using recursion:

\[
\prod_{i=1}^{n} a_i = a_1 \quad \text{and} \quad \prod_{i=1}^{n} a_i = \left( \prod_{i=1}^{n-1} a_i \right) \cdot a_n, \quad \text{if } n > 1.
\]
Sum of Sums

- For any positive integer \( n \), if \( a_1, a_2, \ldots, a_n \) and \( b_1, b_2, \ldots, b_n \) are real numbers, then
  \[
  \sum_{i=1}^{n} (a_i + b_i) = \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i.
  \]

- Proof by induction
  \[
  \sum_{i=1}^{n} (a_i + b_i) = \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i. \quad \leftarrow P(n)
  \]
  - base step: \( \sum_{i=1}^{1} (a_i + b_i) = a_1 + b_1 = \sum_{i=1}^{1} a_i + \sum_{i=1}^{1} b_i \)
  - inductive hypothesis: \( \sum_{i=1}^{k} (a_i + b_i) = \sum_{i=1}^{k} a_i + \sum_{i=1}^{k} b_i. \quad \leftarrow P(k) \)

- Cont.: We must show that:
  \[
  \sum_{i=1}^{k+1} (a_i + b_i) = \sum_{i=1}^{k+1} a_i + \sum_{i=1}^{k+1} b_i. \quad \leftarrow P(k+1)
  \]
  \[
  \sum_{i=1}^{k+1} (a_i + b_i) = \sum_{i=1}^{k} (a_i + b_i) + (a_{k+1} + b_{k+1})
  \]
  \[
  = \left( \sum_{i=1}^{k} a_i + \sum_{i=1}^{k} b_i \right) + (a_{k+1} + b_{k+1})
  \]
  \[
  = \left( \sum_{i=1}^{k} a_i + a_{k+1} \right) + \left( \sum_{i=1}^{k} b_i + b_{k+1} \right)
  \]
  \[
  = \sum_{i=1}^{k+1} a_i + \sum_{i=1}^{k+1} b_i
  \]
  \[
  \quad \quad \text{Q.E.D.}
  \]
Recurrence Relations by Iteration

- **Arithmetic sequence**: there is a constant $d$ such that
  \[ a_k = a_{k-1} + d \text{ for all integers } k \geq 1 \]
  It follows that, $a_n = a_0 + dn$ for all integers $n \geq 0$.

- **Iteration**
  \[ a_0 = 1 \]
  \[ a_1 = a_0 + 2 = 1 + 2 \]
  \[ a_2 = a_1 + 2 = (1 + 2) + 2 = 1 + 2 + 2 \]
  \[ a_3 = a_2 + 2 = (1 + 2 + 2) + 2 = 1 + 2 + 2 + 2 \]

- **Geometric sequence**: there is a constant $r$ such that
  \[ a_k = r \cdot a_{k-1} \text{ for all integers } k \geq 1 \]
  It follows that, $a_n = r^n \cdot a_0$ for all integers $n \geq 0$.

Second-order Linear Homogeneous Recurrence Relation

- A **second-order linear homogeneous recurrence relation** with constant coefficients is a recurrence relation of the form:
  \[ a_k = A \cdot a_{k-1} + B \cdot a_{k-2} \]
  for all integers $k \geq$ some fixed integer
  where $A$ and $B$ are fixed real numbers with $B \neq 0$.

- Second-order: $a_k$ depends on $a_{k-1}$ and $a_{k-2}$;
- Linear: all exponents of the $a_k$'s are 1;
- Homogeneous: all the terms have the same exponent
Recursively Defined Sets

1. Identify a few core objects as belonging to the set AND
2. Give rules showing how to build new set elements from old

- A recursive definition for a set consists of:
  - I. BASE: A statement that certain objects belong to the set.
  - II. RECURSION: A collection of rules indicating how to form new set objects from those already known to be in the set.
  - III. RESTRICTION: A statement that no objects belong to the set other than those coming from I and II.

Recursive Definition of Boolean Expressions

- The set of Boolean expressions over a general alphabet is 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:
    (a) $P \land Q$ and
    (b) $P \lor Q$ and
    (c) $\sim P$.
  III. RESTRICTION: There are no Boolean expressions over the alphabet other than those obtained from I and II.
Recursive Definition of Boolean Expressions

- Example: the following is a Boolean expression over the English alphabet \{a, b, c, \ldots, x, y, z\}:
  \[(\neg(p \land q) \lor (\neg r \land p))\]

1. By I, p, q, and r are Boolean expressions.
2. By (1) and II(a) and (c), \((p \land q)\) and \(\neg r\) are Boolean expressions.
3. By (2) and II(c) and (a), \(\neg(p \land q)\) and \((\neg r \land p)\) are Boolean expressions.
4. By (3) and II(b), \((\neg(p \land q) \lor (\neg r \land p))\) is a Boolean expression.

Recursive String Definitions

- Definition:
  - Let \(S\) a finite set with at least one element. A string over \(S\) is a finite sequence of elements from \(S\).
  
  - The elements of \(S\) are called characters of the string.
  - The length of a string is the number of characters it contains.
  - The null string over \(S\) is defined to be the “string” with no characters.
    - It is usually denoted \(\epsilon\) (epsilon) and is said to have length 0.
Recursive String Definitions

- Example: the Set of Strings over an Alphabet:
  - Consider the set S of all strings in a’s and b’s - S is defined recursively as:
    I. BASE: $\epsilon$ is in S, where $\epsilon$ is the null string.
    II. RECURSION: If $s \in S$, then
      (a) $sa \in S$ and (b) $sb \in S$,
      where $sa$ and $sb$ are the concatenations of $s$ with a and b.
    III. RESTRICTION: Nothing is in S other than objects defined in I and II above.

Derive the fact that $ab \in S$.

(1) By I, $\epsilon \in S$.
(2) By (1) and II(a), $\epsilon a \in S$. But $\epsilon a$ is the concatenation of the null string $\epsilon$ and a, which equals a. So $a \in S$.
(3) By (2) and II(b), $ab \in S$. 

Recursive String Definitions

Derive the fact that $ab \in S$.

(1) By I, $\epsilon \in S$.
(2) By (1) and II(a), $\epsilon a \in S$. But $\epsilon a$ is the concatenation of the null string $\epsilon$ and a, which equals a. So $a \in S$.
(3) By (2) and II(b), $ab \in S$. 

The **MIU-system**:

I. **BASE**: MI is in the MIU-system.

II. **RECURSION**:
   a. If $xI$ is in the MIU-system, where $x$ is a string, then $xIU$ is in the MIU-system = i.e., we can add a U to any string that ends in I. For example, since MI is in the system, so is MIU.
   b. If $Mx$ is in the MIU-system, where $x$ is a string, then $Mxx$ is in the MIU-system = i.e., we can repeat all the characters in a string that follow an initial M. For example, if MUI is in the system, so is MUIUI.
   c. If $xIIIy$ is in the MIU-system, where $x$ and $y$ are strings (possibly null), then $xUY$ is also in the MIU-system = i.e., we can replace $III$ by U. For example, if MIII is in the system, so are MUI and MII.
   d. If $xIIIy$ is in the MIU-system, where $x$ and $y$ are strings (possibly null), then $xUY$ is also in the MIU-system = i.e., we can replace $III$ by U. For example, if MIIIU is in the system, so is MIIU.

III. **RESTRICTION**: No strings other than those derived from I and II are in the MIU-system.

**Derive the fact that MUIU is in the MIU-system**:

1. By I, MI is in the MIU-system.
2. By (1) and II(b), M I I is in the MIU-system.
3. By (2) and II(b), M I I I I is in the MIU-system.
4. By (3) and II(c), MUI is in the MIU-system.
5. By (4) and II(a), MUIU is in the MIU-system.

**Legal Parenthesis Structures**:

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 derived from I and II above.

**Derive the fact that (( )) is in P**:

1. By I, ( ) is in P.
2. By (1) and II(a), (( )) is in P.
3. By (2), (1), and II(b), (( )) is in P.
Structural Introduction for Recursively Defined Sets

- Let S be a set that has been defined recursively, and consider a property that objects in S may or may not satisfy.

To prove that every object in S satisfies the 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 defined by the rule also satisfy the property.

Because no objects other than those obtained through the BASE and RECURSION conditions are contained in S, it must be the case that every object in S satisfies the property.

Legal Parenthesis Structures

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 derived from I and II above.

- Every configuration in P contains an equal number of left and right parentheses:
  Property: any parenthesis configuration has an equal number of left and right parentheses!

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

Show that for each rule in the RECURSION for P if the rule is applied to an object in P that satisfies the property, then the object defined by the rule also satisfies the property:

The recursion for P consists of two rules denoted II(a) and II(b).

Suppose E and F are parenthesis configurations that have equal numbers of left and right parentheses.

When rule II(a) is applied to E, the result is (E), so both the number of left parentheses and the number of right parentheses are increased by one: same number of parenthesis.

When rule II(b) is applied, the result is EF, which has an equal number, m(in E) + n(in F), of left and right parentheses.
Recursive Functions

- **McCarthy’s 91 Function:** \( M : \mathbb{Z}^+ \rightarrow \mathbb{Z} \)

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

\[
M(99) = M(M(110)) \quad \text{since } 99 \leq 100 \\
= M(100) \quad \text{since } 110 > 100 \\
= M(M(111)) \quad \text{since } 100 \leq 100 \\
= M(101) \quad \text{since } 111 > 100 \\
= 91 \quad \text{since } 101 > 100
\]

Recursive Functions

- **The Ackermann Function:**

\[A(0, n) = n + 1 \text{ for all nonnegative integers } n \] (1)
\[A(m, 0) = A(m-1, 1) \text{ for all positive integers } m \] (2)
\[A(m, n) = A(m-1, A(m, n-1)) \text{ for all positive integers } m \text{ and } n \] (3)

\[
A(1, 2) = A(0, A(1, 1)) \quad \text{by (3) with } m = 1 \text{ and } n = 2 \\
= A(0, A(0, A(1, 0))) \quad \text{by (3) with } m = 1 \text{ and } n = 1 \\
= A(0, A(0, A(0, 1))) \quad \text{by (2) with } m = 1 \\
= A(0, A(0, 2)) \quad \text{by (1) with } n = 1 \\
= A(0, 3) \quad \text{by (1) with } n = 2 \\
= 4 \quad \text{by (1) with } n = 3.
\]

\[A(n, n) \text{ increases with extraordinary rapidity: } A(4, 4) \approx 2^{2^{2^{2^{65536}}}}.\]