Recurrence Relations

CSE2315, Chapter 3-2

Properties of recurrence relations

- Recurrence relation is an equation that defines a sequence recursively
  - Each term is defined as a function of the preceding terms

- A linear recurrence relation can be written as

  $S(n) = f_1(n)S(n\text{-}1) + f_2(n)S(n\text{-}2) + \ldots\ldots + f_k(n)S(n\text{-}k) + g(n)$

  where f's and g are or can be expressions involving *n.*

Linearity, Homogeneity, and Orders

- A linear relation is when the earlier values in the definition of $S(n)$ as shown below have power 1.
  - The term "linear" means that each term of the sequence is defined as a linear function of the preceding terms.
  - Example: $F(n) = F(n-1) + F(n-2)$
- A nonlinear relation is the one that has earlier values in the definition as powers other than 1.
  - Example: $F(n+1) = 2nF(n-1)(1-F(n-1))$
  - Solutions are quite complex.

- Homogenous relation is a relation that has $g(n) = 0$ for all n
  - Example: $S(n) = 2S(n-1)$
- Inhomogenous relation:
  - Example: $a(n) - a(n-1) = 2n$

$$S(n) = f_1(n)S(n-1) + f_2(n)S(n-2) + \dots + f_k(n)S(n-k) + g(n)$$

Linearity, Homogeneity, and Orders

- A recurrence relation is said to have constant coefficients if the f's are all constants.
  - Fibonacci relation is homogenous and linear:
    - $F(n) = F(n-1) + F(n-2)$
  - Non-constant coefficients: $T(n) = 2nT(n-1) + 3n^2T(n-2)$

- Order of a relation is defined by the number of previous terms in a relation for the n^{th} term.
  - First order: $S(n) = 2S(n-1)$
    - n^{th} term depends only on term n-1
  - Second order: $F(n) = F(n-1) + F(n-2)$
    - n^{th} term depends only on term n-1 and n-2
  - Third Order: $T(n) = 3nT(n-2) + 2T(n-1) + T(n-3)$
    - n^{th} term depends only on term n-1 and n-2 and n-3

$$S(n) = f_1(n)S(n-1) + f_2(n)S(n-2) + \ldots + f_k(n)S(n-k) + g(n)$$

Solving recurrence relations

- Solving a recurrence relation employs finding a closed-form solution for the recurrence relation.
- An equation such as $S(n) = 2^n$, where we can substitute a value for n and get the output value back directly, is called a closed-form solution.

- Two methods used to solve a recurrence relation:
  - Expand, Guess, Verify
    - Repeatedly uses the recurrence relation to expand the expression for the n^{th} term until the general pattern can be guessed.
    - Finally the guess is verified by mathematical induction.
  - Solution from a formula
    - Known solution formulas can be derived for some types of recurrence relations.

Expand, Guess, and Verify

- Show that $S(n) = 2^n$ for the following recurrence relation:

  $S(1) = 2$

  $S(n) = 2S(n-1)$ for $n \geq 2$

- Expansion: Using the recurrence relation over again every time
  - $S(n) = 2S(n-1)$

  $\Rightarrow S(n) = 2(2S(n-2)) = 2^2S(n-2)$

  $\Rightarrow S(n) = 2^2(2S(n-3)) = 2^3S(n-3)$

- Looking at the developing pattern, we guess that after k such expansions, the equation has the form
  - $S(n) = 2^k S(n-k)$
- This should stop when $n-k = 1$, hence $k = n-1$,
  - As the base case provided is $S(1)$
- $S(n) = 2^{n-1}S(1) \Rightarrow S(n) = 2.2^{n-1} = 2^n$
- Do the verification step by assuming the closed form solution for $S(k)$ and proving $S(k+1)$

Verification step for Expand, Guess, and Verify

- Confirm derived closed-form solution by induction on the value of n.
  - Statement to prove: $S(n) = 2^n$ for $n \geq 2$.
- For the basis step, $S(1) = 2^1$. This is true since $S(1)$ is provided in the problem.
- Assume that $S(k) = 2^k$.
- Then $S(k+1) = 2S(k)$ (by using the recurrence relation definition)
  - $S(k+1) = 2(2^k)$ (by using the above inductive hypothesis)

  $\Rightarrow S(k+1) = 2^{k+1}$
- This proves that our closed-form solution is correct.

Class Exercise

- Find the solution for the following recurrence relation:
  - $T(1) = 1$
  - $T(n) = T(n\text{-}1) + 3$ for $n \geq 2$

Solution from a formula

- Solution formula for linear first order constant coefficient relation

  $S(n) = f_1(n)S(n-1) + f_2(n)S(n-2) + \ldots + f_k(n)S(n-k) + g(n)$

  For the relation $S(n) = 2S(n-1)$, we have $f_1(n) = 2$ and $g(n) = 0$

  So, $S(n) = cS(n-1) + g(n)$ ------------------------ (1)

  General form of linear first order recurrence relation with constant coefficient

  $S(n) = c[cS(n-2)+g(n-1)] + g(n) = c[c[cS(n-3)+g(n-2)] + g(n-1)] + g(n)$.

  $S(n) = c^kS(n-k) + c^{k-1}g(n-(k-1)) + \ldots + cg(n-1) + g(n)$

  The lowest value of $n-k$ is 1 (n-k=1, so k=n-1)

  Hence, $S(n) = c^{n-1}S(1) + c^{n-2}g(2) + c^{n-3}g(3) + \ldots + g(n)$

$$S(n) = c^{n-1}S(1) + \sum_{i=2}^{n} c^{n-i}g(i)$$

- For $S(n) = 2S(n-1)$, $c = 2$ and $g(n) = 0$

  Hence, $S(n) = 2^{n-1}S(1) = 2.2^{n-1} = 2^n$ since $S(1) = 2$

Solution Formula

<table>
<tr><th colspan="2">To solve recurrence relations of the form S(n) = cS(n-1)+g(n)
subject to basis S(1)</th></tr>
<tr><td>Method</td><td>Steps</td></tr>
<tr><td>Expand, guess, and verify</td><td>1. Repeatedly use the recurrence relation until you can guess a pattern
2. Decide what that pattern will be when n-k=1
3. Verify the resulting formula by induction</td></tr>
<tr><td>Solution formula</td><td>1. Match your recurrence relation to the form S(n)=cS(n-1) + g(n) to find c and g(n)
2. Use c, g(n), and S(1) in the formula
$$S(n) = c^{n-1}S(1) + \sum_{i=2}^{n} c^{n-i} g(i)$$
3. Evaluate the resulting summation to get the final expression</td></tr>
</table>

Class Exercise

- Find a closed-form solution to the recurrence relation
  - $S(n) = 2S(n-1) + 3$ for $n \geq 2$ and given $S(1) = 4$
  - Here, c=2 and g(n) = 3
- given by

$$S(n) = 2^{n+1} + 3\left[2^{n-1} - 1\right]$$

- Find a closed-form solution to the recurrence relation
  - $T(n) = T(n-1) + (n+1)$ for $n \geq 2$ and given $T(1) = 2$
  - Here, c = 1 and g(n) = n+1
- given by

$$T(n) = \frac{n(n+3)}{2}$$

- Solutions for these exercises is in the text (pg. 184-186, Example 17 & 18)