Discrete Mathematics	08/08/2018

Lecture 6: Recursion

Instructor: Sourav Chakraborty *Scribe: Pinaki Chakraborty*

1 Recursion

In this lecture we shall study the methods of solving equations of the following form:

$$T(n) = 2T(\lceil \frac{n}{2} \rceil) + n,\ T(0) = 1\ and\ T(1) = 0 \quad \forall n \in \mathbb{Z}^+ \tag{1}$$

$$F(n) = F(n-1) + F(n-2),\ F(0) = 0\ and\ F(1) = 1 \quad \forall n \in \mathbb{Z}^+ \tag{2}$$

$$S(n) = S(n-1) + n\ andS(0) = 2 \quad \forall n \in \mathbb{Z}^+ \tag{3}$$

$$C(n) = C_0C_{n-1} + C_1C_{n-2} + ... + C_{n-2}C_1 + C_{n-1}C_0\ andC_0 = 1 \quad \forall n \in \mathbb{Z}^+ \tag{4}$$

These kind of equations are known as ***Recurrence relations***. In general the method of solving these equation involves guessing a viable solution and then proving its validity by method of induction.

1.1 Examples

Example 1: This example illustrates the approach for solving (3).
Expanding the R.H.S of (3) we obtain $S(n) = S(n-1) + n = S(n-2) + (n-1) + n$. Hence we can guess $S(n) = S(n-i) + (n-(i-1)) + \cdots + n$ for some $i \leq n$. Putting $i = n$ and using $S(0) = 2$ we guess that $S(n) = 2 + (1 + 2 + 3 + \cdots + n = 2 + n(n+1)/2)$. Now the method of induction is used to prove that $S(n) = 2 + n(n+1)/2$ where S(0) = 2 formally, which is left as an exercise.

Example 2: In this example we attempt to find an exact solution of (1). If $n = 2^l$ for some l $\in \mathbb{Z}^+$ then $T(n) = T(\lceil \frac{n}{2} \rceil) + n = 2T(\frac{n}{2}) + n = 4T(\frac{n}{4}) + 2n = \cdots$ $= 2^iT(\frac{n}{2^i}) + in$ for some i $\in \mathbb{Z}^+$. When i = l, $t(n) = 2^lT(\frac{n}{2^l} + ln) = nT(1) + n\log n = n + n\log n$ as T(1) = 1. More generally, we guess $T(n) = n + n\lceil \log n \rceil$ and prove this closed form solution by Method of Induction. It is noteworthy that, we could have also guessed $T(n) = 2^{\lfloor \log n \rfloor} + n\lceil \log n \rceil$ i.e our guess may not be unique or viable solution always.

2 Asymptotic Notation

From the foregoing discussion it is evident that deriving exact solution for a recurrence is in general not feasible. Yet we can derive upper bonds and lower bounds for many recurrences. To notationally represent these we introduce the notion of asymptotic behavior.

2.1 Asymptotic Similarity

Two functions f(n) and g(n) are said to be asymptotically similar if $\lim_{n\to\infty} f(n)/g(n) = 1$. This is notationally written as $f(n) \sim g(n)$.

2.2 Θ-notation

A function f(n) is said to belong to Θ(g(n)) if $\exists\, c_1, c_2, N_1, N_2$ such that $\forall n > N_1$, $f(n) \leq c_1 g(n)$ and $\forall n > N_2$, $g(n) \leq c_2 f(n)$. This is notationally written as $f(n) = \Theta(g(n))$.

2.3 $\mathcal{O}$-notation

A function f(n) is said to belong to $\mathcal{O}$(g(n)) if $\exists\, c_1, N_1$ such that $\forall n > N_1$, $f(n) \leq c_1 g(n)$. This is notationally written as $f(n) = \mathcal{O}(g(n))$.

2.4 Ω-notation

A function f(n) is said to belong to Ω(g(n)) if $\exists\, c_1, N_1$ such that $\forall n > N_1$, $f(n) \geq c_1 g(n)$. This is notationally written as $f(n) = \Omega(g(n))$.

2.5 o-notation

A function f(n) is said to belong to o(g(n)) if $\lim_{n\to\infty} f(n)/g(n) = 0$. This is notationally written as $f(n) = o(g(n))$.

2.6 ω-notation

A function f(n) is said to belong to Ω(g(n)) if $\lim_{n\to\infty} g(n)/f(n) = 0$. This is notationally written as $f(n) = \omega(g(n))$.

2.7 Examples

The following are some simple examples of usage of these notations:

- $T(n) \geq \frac{1}{10} n \log n \Rightarrow$ T(n) = Ω(n log n)
- $T(n) \leq 10 n \log n \Rightarrow$ T(n) = $\mathcal{O}$(n log n)

It follows that in the above scenario $T(n) = \Theta(n \log n)$. Now we prove that for relation (1), $T(n) = \mathcal{O}(n \log n)$

Proof. It is given that $\exists\ c, N_0$ *such that* $\forall n > N_0$, T(n) $\leq 2T(\lceil \frac{n}{2} \rceil) + cn$. WTS, $\exists\ d, N_1$ *such that* $\forall n > N_1$, $T(n) \leq n \log n$. Let $T(n) \leq dn \log n$. By Induction Hypothesis $T(n) \leq 2(d\lceil \frac{n}{2} \rceil \log \lceil \frac{n}{2} \rceil) + cn$. Since, $d(n+1)\log \frac{2n}{3} + cn \geq 2(d\lceil \frac{n}{2} \rceil \log \lceil \frac{n}{2} \rceil) + cn$ as $\lceil \frac{n}{2} \rceil \leq \frac{2n}{3}$, we have $2(d\lceil \frac{n}{2} \rceil \log \lceil \frac{n}{2} \rceil) + cn \leq d(n+1)\log n - d9n+1)\log \frac{3}{2} + cn = dn \log n + [d \log n - d(n+1)\log \frac{3}{2} + cn] \leq dn \log n$ whenever $d \log n - d(n+1)\log \frac{3}{2} + cn < 0$. Hence, for $d > cn/[(n+1)\log \frac{3}{2} - \log n] \geq 3c$ the condition $T(n) \leq dn \log n$ is satisfied. Thus the Induction Hypothesis holds. $\square$