Lecture 20: September 27

Lecturer: Samar *Scribes: Kanika Gupta, Khushi Gupta*

20.1 Structural induction for recursive data types

Lemma 1. $|s.t| = |s| + |t| \forall s, t \in\ A*$

By structural induction on the definition of $s \in\ A$. The induction hypothesis is

$$P(s) :: \forall t \in\ A*, \mid s.t \mid = \mid s \mid + \mid t \mid$$

Base Case ($s = \lambda$):

$$\begin{aligned} \mid s.t \mid = \mid s \mid + \mid t \mid &= \mid t \mid \\ &= 0 + \mid t \mid \\ &= \mid s \mid + \mid t \mid \end{aligned}$$

Constructor Case:($s ::= < a, r >$) **Base Case** ($s = \lambda$):

$$\begin{aligned} \mid s.t \mid = \mid < a, r > .t \mid &= \mid t \mid \\ &= 0 + \mid t \mid \\ &= \mid s \mid + \mid t \mid \end{aligned}$$

This proves that P(s) holds, completing the constructor case. By structural induction, we conclude that P(s) holds $\forall$ strings $s \in\ A*$

20.2 Recursive Functions on Nonnegative Integers

The nonnegative integers can be understood as a recursive data type.

Definition 3.5.8. The set, $\mathbb{N}$, is a data type defined recursivly as:

- **Base case:** $0 \in\ \mathbb{N}$
- **Constructor Case:** If $n \in\ \mathbb{N}$, then the successor , n + 1 , of n is in $\mathbb{N}$.

Any function defined on recursive data types is recursive.

$Example$: $f(n) = 2f(n-1)$

$Solution$: This is not a valid function , since we don't have a base case.

$Example$:

$$f(n) = \begin{cases} 0 & \text{if } n = 0; \\ f(n+1) & \text{if } n > 0. \end{cases}$$

$Solution$: We can't find $f(1)$, so it is not a recursive function.

$Example$:

$$f(n) = \begin{cases} 0 & \text{if } n \text{ is divisible by 2;} \\ 1 & \text{if } n \text{ is divisible by 3;} \\ 2 & \text{otherwise.} \end{cases}$$

$Solution$: Every multiple of 6 will have 2 values. Therefore not a recursive function.

$Example$:

$$f(n) = \begin{cases} 1 & \text{if } n <= 1 \text{ ;} \\ f(n/2) & \text{if } n > 1 \text{ and even;} \\ f(3n+1) & \text{if } n > 1 \text{ and odd.} \end{cases}$$

$Solution$: Suppose $n = 12$

$f(12) = f(6) = f(3) = f(10) = f(5) = f(16) = f(8) = f(4) = f(2) = f(1) = 1$

$f(13) = f(40) = f(20) = f(10) = f(5) = f(16) = f(8) = f(4) = f(2) = f(1) = 1$

The last function was Collatz conjecture or Ulam conjecture or Syracuse problem or Hasse's Algorithm, 3n +1 conjecture.

The sequence 12, 6, 3, 10, 5... 1 is called Hailstone sequence.

20.2.1 Ackermann function

It is an extremely fast growing function.

$$A(m,n) = \begin{cases} 2n & \text{if } m = 0, n <= 1 \text{ ;} \\ A(m-1, A(m, n-1)) & \text{otherwise.} \end{cases}$$

- We can't use induction here but it is a well defined function.
- It's inverse grows very slowly.
- During algo analysis, it occurs as a pre-factor(linear(n)).

It is a total computable function that is not primitive recursive.