10/23/2024

Recursively defined Sets

Recursively defined Structures

We are going to look at this because we want to use induction on above structure or set. (Trying to extend induction beyond natural numbers)

A recursively defined structure/set is a structure/set S defined by:

1) Its smallest element(s) (base case(s))
2) Rule(s) that construct elements out of smaller elements.
3) The structure/set includes only elements that can be constructed from the base case and the recursive case.

$S = \{x : x$ is either case (1) or x follows case (2)$\}$

Ex: The set of non-negative integers

1) 0
2) If k is a non-negative integer, $k+1$ is also a non-negative integer.

we can generate the smallest element

$\underline{0}$

I can make 1 using 0 and case 2

$0 + 1 = 1$

1 is a non-negative integer

$1 + 1 = 2$

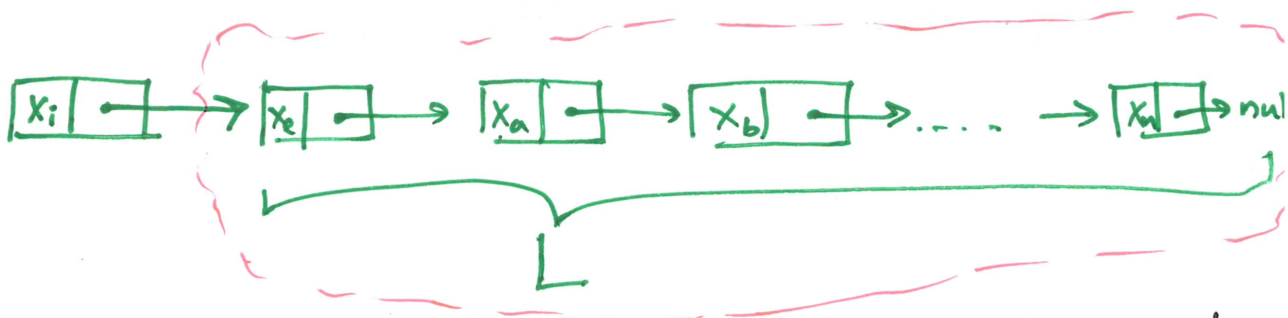
---

ex! A recursively defined structure.

Let's try to define a Linked List as a recursively defined structure.

1) Base case: Empty linked List [ ], null

2) Recursive case: $[x_i, L]$, where x_i is some data & L is a Linked List.

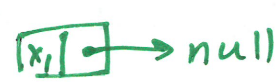


Smallest Linked List

[ ], null

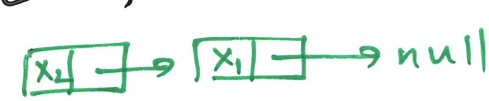
Linked List with 1- element

$[x_1, [\ ]]$



LL with 2-elements

$[x_2, [x_1, [\ ]]]$



Ex: set of ~~all~~ well formed statements of propositional logic, over a set of boolean variables X.

$X = \{t, q, r\}$

$t \Rightarrow q$, $t \vee q$, $r \vee q$, $r \Leftrightarrow q$

$(t \vee q) \Rightarrow (r \Leftrightarrow q)$

1. p, for some $p \in X$

2a) If p, s are well formed sentences of propositional logic;
$p \star s$ where $\star \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow, \oplus\}$

2b) If p is WFS of propositional logic, $\neg p$ is a well formed statement of propositional logic

$X = \{a, b, c, d\}$

Base case elements

a, b, c, d

$a \vee b$ $\quad a \oplus d$ $\quad \neg a$ $\quad \neg b$

$(a \vee b) \Leftrightarrow (a \oplus d)$

$\neg((a \vee b) \Leftrightarrow (a \oplus d))$

$S = \{a, b, d, c, a \vee b, a \oplus d, \neg a, \neg b, \ldots\ldots \}$

Ex: Binary Trees as a ~~set~~ recursively defined structure.

1) null (empty tree) or ___ ← nothing

2) root node r, left subtree T_ℓ, right subtree T_r

where r is an arbitrary value, & T_ℓ, T_r are both binary trees.