

Structural Induction Cont.

CSE 311: Foundations of
Computing I
Lecture 18

Announcements

- HW5 solutions are at the front.
- HW6 will be released today, due next Wednesday at 11:59pm on Gradescope.
- The Midterm has been graded, and scores will be published on Gradescope this afternoon

Midterm Grades

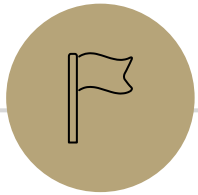
Overall we were very impressed with your work.

Stats:

- Median: 72 / 80 (90%)
- Mean: 66.58 / 80 (83.23%)

Course Grades

- The Midterm is only 15% of your overall course grade.
- Course grades are based on historical benchmarks for CSE 311.
 - If scores are lower than in the past, we will round grades up
 - If scores are higher than in the past, then we won't adjust down!
- The typical median GPA for this course is around a 3.5
- If you are concerned about your grade in the course, please feel free to reach out to anjalia@uw.edu to schedule an appointment.



Midterm Corrections

Midterm Corrections

- The purpose of an exam is to demonstrate learning
- We are offering the opportunity for you to correct your Midterm to earn a better score

Midterm Corrections

Detailed instructions will be posted on the course website under “Assignments” after class.

To submit corrections:

1. For each problem part that you wish to correct, rewrite your complete solution.
2. For each problem part that you wish to correct, write 2-4 sentences reflecting on **what** was incorrect in the original attempt, **why** the original error was made, and **how** you have fixed the error in your solutions.
3. Submit your corrections to Gradescope under “Midterm Corrections” by Wednesday, August 9th at 11:59pm.

Midterm Corrections

You **do not need** to submit corrections to every problem.

Your effective midterm score will be calculated as follows:

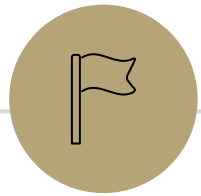
$$\text{midterm_grade} = 0.75 \cdot \text{original_grade} + 0.25 \cdot \max(\text{corrected_grade}, \text{original_grade})$$

Student A: 60% on the midterm, 95% after corrections. Effective grade is 68.75%.

Student B: 80% on the midterm, 70% after corrections. Effective grade is 80%.

Midterm Corrections

- You may not use Late Days on the Midterm corrections.
 - We intend to pass out Midterm solutions in class next Friday, August 11th
- Submissions without reflections will not be graded.
- The collaboration policy is the same as homework.
 - That means you may discuss with others and ask questions in OH, but you must write up solutions on your own
- Corrections will not be offered for the Final exam.



Recursively Defined Sets

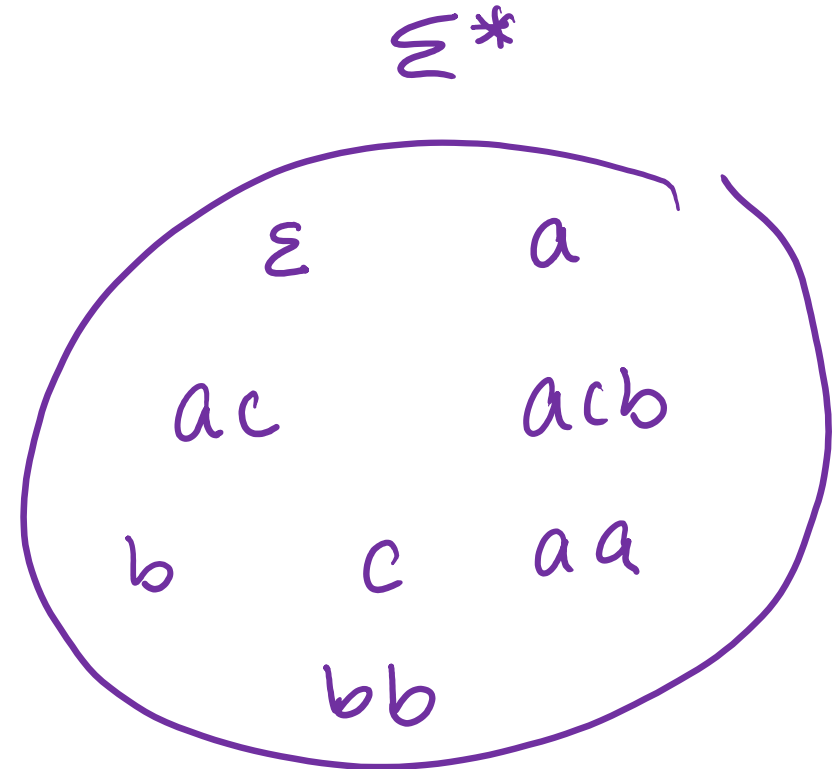
Recursive Definition for Strings

Σ is the alphabet
 Σ^* is the set of all strings
 ε is the empty string

Basis Step: $\varepsilon \in \Sigma^*$

Recursive Step: If $w \in \Sigma^*$ and $a \in \Sigma$, then $wa \in \Sigma^*$

$\Sigma = \{a, b, c\}$



Recursive Definition for Lists of Integers

Basis Step: [] \in List

Recursive Step: If L \in List and a $\in \mathbb{Z}$ then a :: L \in List

int :: list
list

1 :: 2 :: []

[1, 2]

Recursive Definition for Binary Trees of Integers

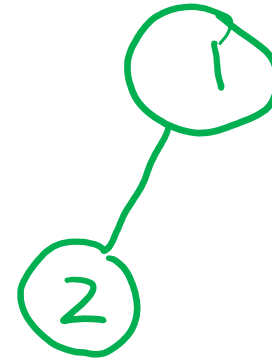
Basis Step: null \in Tree

Recursive Step: If L, R \in Tree, and a \in \mathbb{Z} then (L, a, R) \in Tree

null



(null, 1, null)



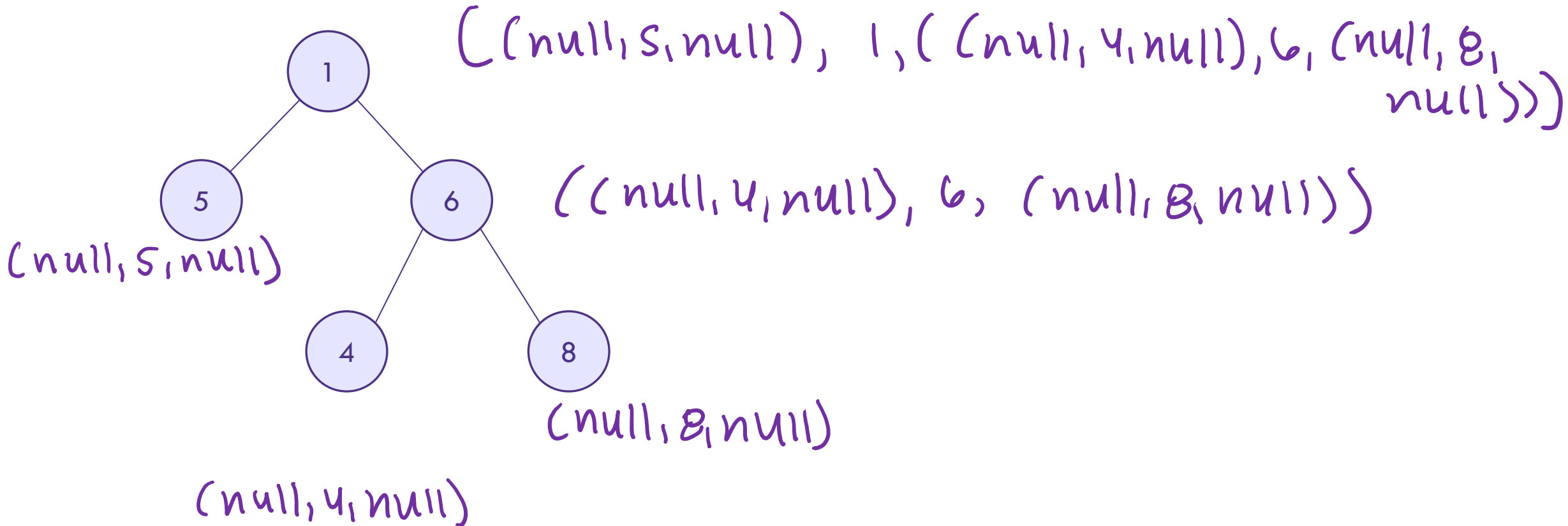
((null, 2, null), 1, null)

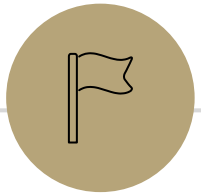
Exercise

Basis: $\text{null} \in \text{Tree}$

Recursive: If $L, R \in \text{Tree}$, and $a \in \mathbb{Z}$ then $(L, a, R) \in \text{Tree}$

Write out the following tree using our notation for trees.





Claim 1



Functions on Binary Trees

Basis: $\text{null} \in \text{Tree}$

Recursive: If $L, R \in \text{Tree}$, and $a \in \mathbb{Z}$ then $(L, a, R) \in \text{Tree}$

To prove interesting facts about trees, we need functions on trees.

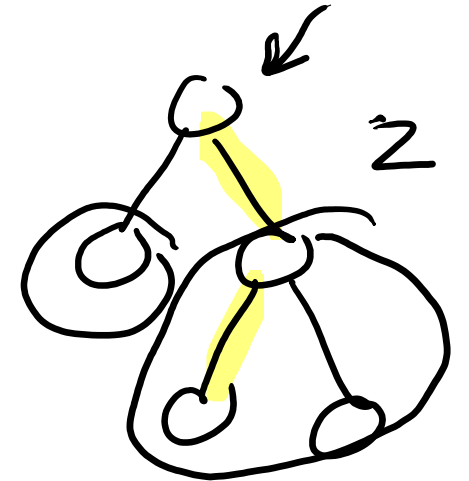
Size:

$$\text{size}(\text{null}) = 0$$

$$\text{size}((L, a, R)) = 1 + \text{size}(L) + \text{size}(R)$$

Height:

$$\text{height}(\text{null}) = \underline{-1}$$



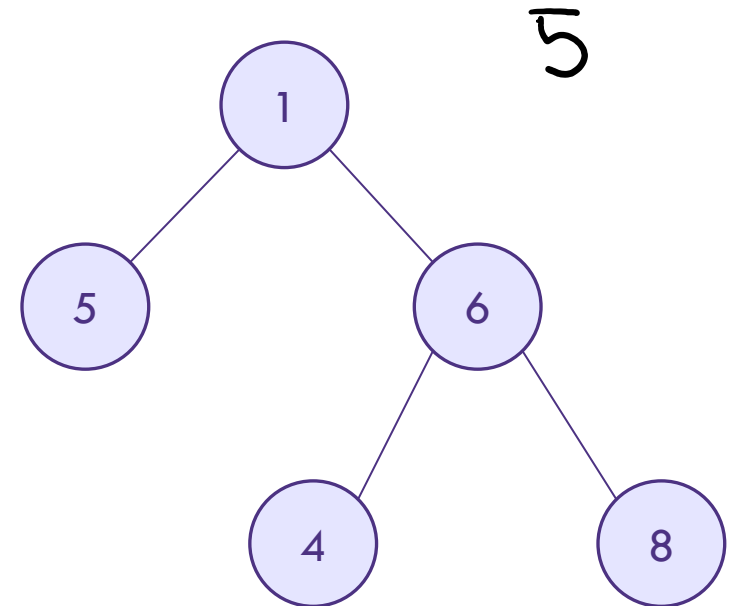
$$\text{height}((L, a, R)) = 1 + \max(\text{height}(L), \text{height}(R))$$
$$1 + \max(\text{height}(\text{null}), \text{height}(\text{null}))$$

Functions on Binary Trees

$$\text{size}(\text{null}) = 0$$

$$\text{size}((L, a, R)) = 1 + \text{size}(L) + \text{size}(R)$$

What's the size the tree $((\text{null}, 5, \text{null}), 1, ((\text{null}, 4, \text{null}), 6, (\text{null}, 8, \text{null})))$?

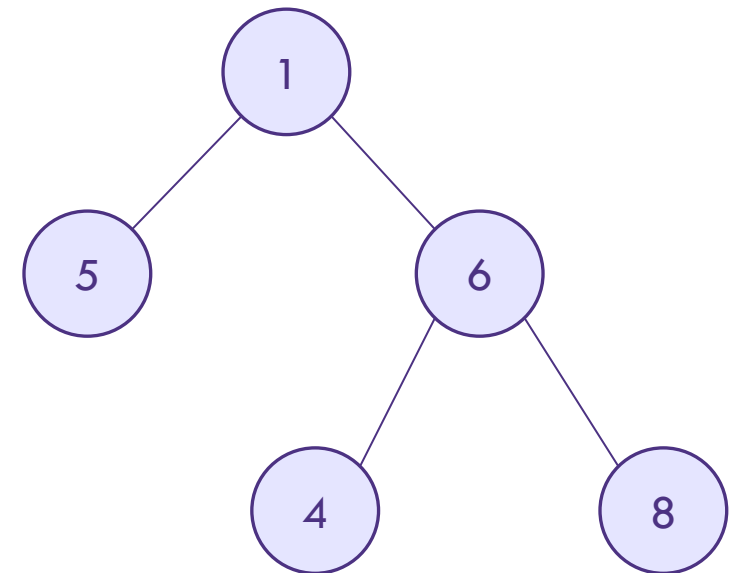


Functions on Binary Trees

$$\text{height}(\text{null}) = -1$$

$$\text{height}((L, a, R)) = 1 + \max(\text{height}(L), \text{height}(R))$$

What's the height of the tree $((\text{null}, 5, \text{null}), 1, ((\text{null}, 4, \text{null}), 6, (\text{null}, 8, \text{null})))$?

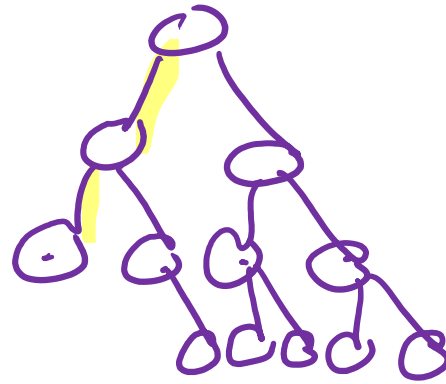


Claim 1

$$\begin{aligned} \text{height}(\text{null}) &= -1 \\ \text{height}((L, a, R)) &= 1 + \max(\text{height}(L), \text{height}(R)) \end{aligned}$$

$$\begin{aligned} \text{size}(\text{null}) &= 0 \\ \text{size}((L, a, R)) &= 1 + \text{size}(L) + \text{size}(R) \end{aligned}$$

Claim 1: For every binary tree, $\text{size}(T) \leq 2^{\text{height}(T)+1} - 1$.



size : 15
height : 3

$$2^{\text{height}(T)+1} - 1 = 2^4 - 1 = 16 - 1 = 15$$

$$\text{size}(\text{null}) = 0$$

$$\text{size}((L, a, R)) = 1 + \text{size}(L) + \text{size}(R)$$

$$\text{height}(\text{null}) = -1$$

$$\text{height}((L, a, R)) = 1 + \max(\text{height}(L), \text{height}(R))$$

Basis: $\text{null} \in \text{Tree}$

Recursive: If $L, R \in \text{Tree}$, and $a \in \mathbb{Z}$ then $(L, a, R) \in \text{Tree}$

Claim 1 Proof

- Let $P(T)$ be " $\text{size}(T) \leq 2^{\text{height}(T)+1} - 1$ ". We prove $P(T)$ holds for all $T \in \text{Tree}$ by structural induction.
- Base Case: ($T = \text{null}$). Then:

$$\text{size}(\text{null}) = 0 = 1 - 1 = 2^0 - 1 = 2^{\text{height}(\text{null})+1} - 1.$$

- IH: Suppose $P(L)$ and $P(R)$ for arbitrary trees L, R . Then $\text{size}(L) \leq 2^{\text{height}(L)+1} - 1$, and $\text{size}(R) \leq 2^{\text{height}(R)+1} - 1$.

4. IS: Goal $P((L, a, R))$

Let $a \in \mathbb{Z}$ be arbitrary.

$$\text{size}((L, a, R)) = 1 + \text{size}(L) + \text{size}(R) \quad \text{Def of size}$$

$$\leq 1 + 2^{\text{height}(L)+1} + 2^{\text{height}(R)+1} - 1 \quad \text{IH}$$

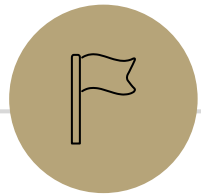
$$= 2^{\text{height}(L)+1} + 2^{\text{height}(R)+1} - 1$$

$$\leq 2^{\max(\text{height}(L), \text{height}(R))+1} + 2^{\max(\text{height}(L), \text{height}(R))+1} - 1$$

5. Conclusion

$$= 2 \cdot 2^{\max(\text{height}(L), \text{height}(R))+1} - 1 = 2^{\text{height}((L, a, R))+1} - 1$$

$$2^a \leq 2^{\max(a, b)}$$



Claim 2



Functions on Lists

Basis: $[] \in \text{List}$

Recursive: If $L \in \text{List}$ and $a \in \mathbb{Z}$ then $a :: L \in \text{List}$

To prove interesting facts about lists, we need functions on lists.

Length:

$$\text{len}([]) = 0$$

$$\text{len}(a :: L) = 1 + \text{len}(L)$$

Concatenation:

$$\text{concat}([], R) = R$$

$$\text{concat}(a :: L, R) = a :: \text{concat}(L, R)$$

$$\text{len}(1 :: 2 :: [])$$

$$= 1 + \text{len}(2 :: [])$$

$$= 1 + 1 + \text{len}([])$$

$$= 2$$

$$\text{concat}(1 :: [], 2 :: [])$$

$$= 1 :: \text{concat}([], 2 :: [])$$

$$= 1 :: 2 :: []$$

Claim 2

$$\begin{aligned} \text{len}([\]) &= 0 \\ \text{len}(a :: L) &= 1 + \text{len}(L) \end{aligned}$$

$$\begin{aligned} \text{concat}([\], R) &= R \\ \text{concat}(a :: L, R) &= a :: \text{concat}(L, R) \end{aligned}$$

Claim 2: For all lists L, R , $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$.

$\forall L \forall R$

How do we prove a nested forall?

Let $P(L)$ be " $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$
for all lists $R \in \text{List}$."

We prove $P(L)$ for all $L \in \text{List}$ by struct. ind.

Claim 2 Proof

Basis: $[\] \in \text{List}$

Recursive: If $L \in \text{List}$ and $a \in \mathbb{Z}$ then
 $a :: L \in \text{List}$

$\text{concat}([\], R) = R$

$\text{concat}(a :: L, R) = a :: \text{concat}(L, R)$

$\text{len}([\]) = 0$

$\text{len}(a :: L) = 1 + \text{len}(L)$

$L \in \text{List}$

1. Let $P(L)$ be " $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ for all lists $R \in \text{List}$ ". We prove $P(L)$ for all lists $L \in \text{List}$ by structural induction. $R \in \text{List}$

2. Base Case: $P([\])$. Let $R \in \text{List}$ be arbitrary. Then LHS is $\text{len}(\text{concat}([\], R)) = \text{len}(R)$. The RHS is $\text{len}([\]) + \text{len}(R) = 0 + \text{len}(R) = \text{len}(R)$. So base case holds.

3. IH: Suppose that $P(L)$ holds for some arbitrary $L \in \text{List}$. That means for all $R \in \text{List}$, $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$.

4. IS: **Goal: $P(a :: L)$**

Let $a \in \mathbb{Z}$ be an arb integer. Let $R \in \text{List}$ be arbitrary.

$$\begin{aligned} \text{len}(\text{concat}(a :: L, R)) &= \text{len}(a :: \text{concat}(L, R)) && \text{Def of concat} \\ &= 1 + \text{len}(\text{concat}(L, R)) \\ &= 1 + \text{len}(L) + \text{len}(R) && \leftarrow \text{By IH} \end{aligned}$$

5. Conclusion

$$= \text{len}(a :: L) + \text{len}(R) \quad \text{Def of length}$$