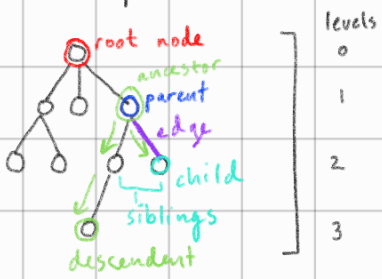
CS 173 - Lecture 16

Trees

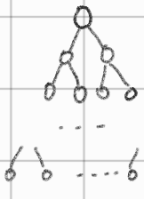
- Undirected graph w/ root node where every node is connected to root node by exactly one path

ex.



- ancestor/descendent - related through a direct or chain of parent/child relationships
- leaf - no children
- internal node - non-leaf, has child(ren)
- level - path length from that node to root
- height - maximum level of any node
- m-ary tree - each node in the tree has $\leq m$ children (eg. binary tree is 2-ary)
- full m-ary tree - each node has exactly 0 or m children
- complete tree - all leaves are at the same level
- subtree - take nodes/edges from the original tree

ex. Counting nodes: how many nodes in a full and complete binary tree of height h?



level	nodes	total
0	1	1
1	2	3
2	4	7
...		
h	2^h	

node at level $l = 2^l$

$$\sum_{k=0}^{h} 2^k = \boxed{2^{h+1} - 1}$$

Grammars

- Strings - finite length sequence of characters
- Length - # of characters
- ϵ - empty string, length zero, no characters

ex. "apple" length = 5

ϵ = "" length = 0

- we can concatenate strings by writing them next to each other

ex. α = cup β = cake

$\alpha\beta$ = cupcake αs = cups

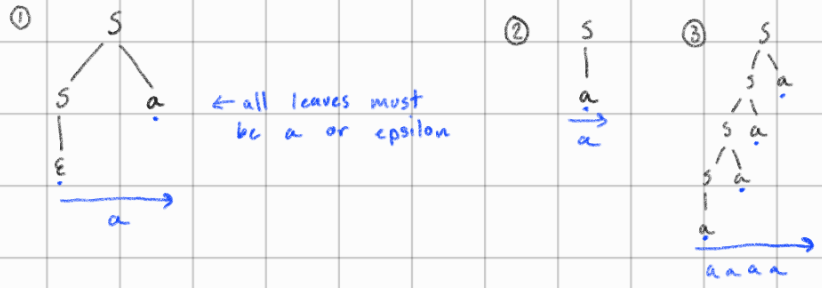
- bit strings - made up of only 0's and 1's
- Context free grammars - set of rules that define the of a tree, produces terminal sequence/string

ex. Define grammar G with start symbol S and terminal symbol a, with rules:

$S \to Sa \mid a \mid \epsilon$

starting node can branch into either 2 children S, a, 1 child a, or 1 child ϵ

possible trees:



Our grammar G generates the set of strings made up of 0 or more a's

↳ produces everything in the set and nothing outside the set

ex. Design a grammar that generates strings of the form $a^n bc^n$ where $n \geq 0$

basically saying terminal string has exactly 1 b, and equal number of a's and c's, and 0 or more a's and c's (eg. aabcc, abc, b, etc.)

Grammar tree G, start S, terminals a, b, c, rules: $S \to b \mid aSc$

eg.

Discussion Problems - Trees and Grammars Problems

1. Thinking About Trees

Suppose you have a binary tree consisting of 10 nodes.

(a) What is the **tallest** possible way to draw this tree?

(i) What is the height of this tree?

(i) How many internal nodes does this tree have? How many leaves?

(b) What is the **shortest** possible way to draw this tree?

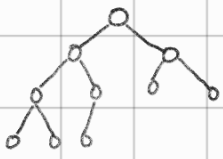
(i) What is the height of this tree?

(i) How many internal nodes does this tree have? How many leaves?

(c) Can you construct a full binary tree with 10 nodes? Explain informally why or why not.

(d) Write a generalized form to describe the number of nodes, n, that can form a full binary tree.

a)  i) height = 9 ii) internal nodes = 9 leaves = 1

b)  i) height = 3 ii) internal nodes = 5 leaves = 5

c) no, you always need an odd number of nodes to build a full binary tree

d) $n = 2k+1$, k = # internal nodes, $k \in \mathbb{N}$

2. Constructing a Grammar

A *palindrome* is a string that is the same forwards as it is backwards. For example, "racecar," "madam," and "level" are all examples of palindromes.

(a) Design a context-free grammar that generates all possible palindromes involving the symbols "a" and "b". *Explicitly state what start and terminal symbols you are using.*

(b) Modify the grammar from part (a) to only generate palindromes where any "a" symbols must be outside any "b" symbols. More specifically, the grammar should generate strings of the form $a^n b^m a^n$. *Try to use as few rules as possible!*

a) Grammar tree G with start S, terminals a, b, and rules: $S \to a \mid b \mid aSa \mid bSb \mid \epsilon$

ensures that even length palindromes can be generate (eg. aa, baab, etc.)

b) Grammar tree G with start S, terminals a, b, and rules: $S \to a \mid aSa \mid B$

$B \to b \mid bBb \mid \epsilon$