Trees

Problem 1) Suppose that you have a binary tree with 1000 nodes. What are the maximum and minimum heights? How many different structures can the maximum height tree have?

In a binary tree, each node will have 2 (possibly empty) children.

For the maximum height, each node will have exactly one child.

Node 1 will be at level 0
Node 2 will be at level 1
…
Node n will be at level n-1

So the max height would be 999.

For the minimum height, fill each level with the largest number of nodes.

Level 0 – 1 node (the root)
Level 1 – 2 nodes (left and right children)
Level 2 – 4 nodes
…
Level h - 2^h nodes

If every level is filled, then we can accommodate $1 + 2 + 4 + \cdots 2^h = 2^{h+1} - 1$ nodes. We can solve this for 1000 nodes

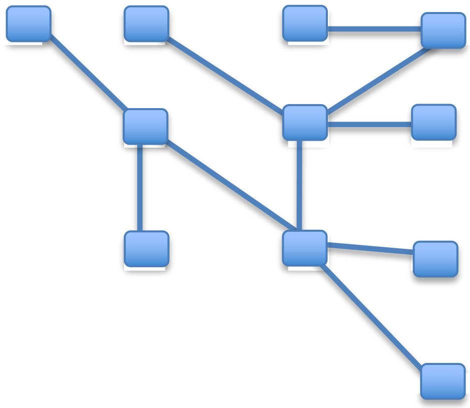
$$2^{h+1} - 1 = 1000$$
$$2^{h+1} = 1001$$
$$lg(2^{h+1}) = lg(1001)$$
$$h + 1 = lg(1001)$$
$$h = lg(1001) - 1$$
$$h \approx 9.97 - 1$$
$$h \approx 8.97$$

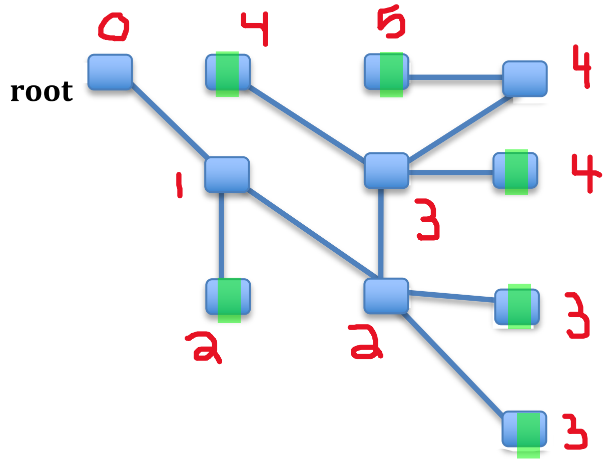
We need to take the ceiling so we find that h=9. If we add in just a few more nodes, we will need another level in the tree.

The number of different structures for a maximum height tree is easy to compute. Every child can either be on the left or right. So we have a sequence of 999 left/right choices (all but the root) for a total of 2^{999} different structures.

Problem 2) For the following tree, what is the height? How many leaf nodes are there? How many internal nodes are there? Is this tree a 4-ary tree?

root





The maximum level gives the height of 5.
There are 6 leaf nodes marked in green.
The remaining 5 nodes are internal nodes.

Note:

Regular grammar:

Only single variables on the LHS of replacement rules.
Only single variables on the RHS of replacement rules.
All terminals on RHS are left of the variable.

Context free grammar:

Only single variables on the LHS of replacement rules.

Context sensitive grammar:

Anything else

Derivation:

Sequence of strings of terminals and non-terminals where each step applies a single replacement rule.

Parse tree:

A tree where the children of a non-terminal are the symbols on the RHS of a replacement rule.

Language generated by a grammar:

The set of strings of terminals (words) that can be derived from the starting non-terminal.

Ambiguous grammar:

A grammar where some word in the language has two different parse trees. Every derivation has a unique parse tree, but one parse tree maybe associated with many derivations.

Problem 3) For the following grammar G

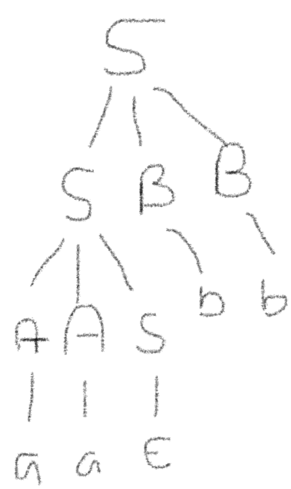
a) Give 3 different derivations of length at least 3. Sketch the parse tree for each derivation.
b) Give a general description of the strings that are generated.
c) Is the grammar regular?
d) Is the grammar context free?
e) Is the grammar ambiguous?

G:	$S \rightarrow AAS \mid SBB \mid \varepsilon$ $A \rightarrow a$ $B \rightarrow b$

$S \rightarrow AAS \rightarrow AA \rightarrow aA \rightarrow aa$

$S \rightarrow AAS \rightarrow AASBB \rightarrow aASBB \rightarrow aASbB \rightarrow aAbB \rightarrow aabB \rightarrow aabb$
$S \rightarrow SBB \rightarrow SBb \rightarrow AASBb \rightarrow AASbb \rightarrow AaSbb \rightarrow aaSbb \rightarrow aabb$

The strings generated are an even number of a's followed by an even number of b's. The grammar is not regular since it has multiple non-terminals on the RHS of some production rule. It is context free. It is ambiguous. The last two derivations end at the same word, but have different parse trees as you can see.



Problem 4) For the following grammar G

a) Give 3 different derivations of length at least 3. Sketch the parse tree for each derivation.

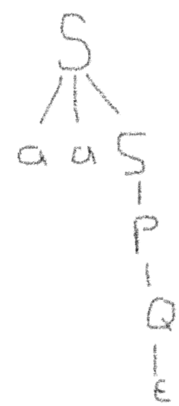
b) Give a general description of the strings that are generated.

c) Is the grammar regular?

d) Is the grammar context free?

e) Is the grammar ambiguous?

G:
$S \rightarrow aaS \mid P$
$P \rightarrow bbP \mid Q$
$Q \rightarrow \varepsilon$

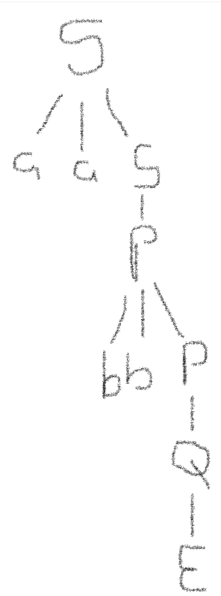


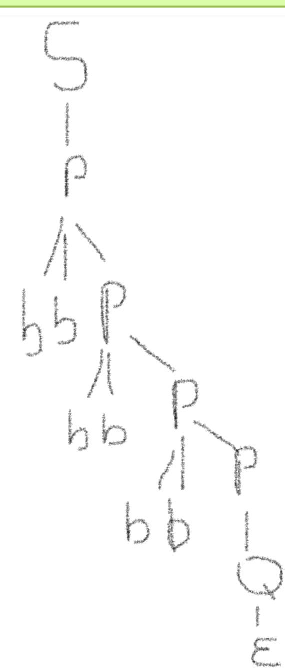
$S \rightarrow aaS \rightarrow aaP \rightarrow aaQ \rightarrow aa$

$S \rightarrow aaS \rightarrow aaP \rightarrow aabbP \rightarrow aabbP \rightarrow aabbQ \rightarrow aabb$

$S \rightarrow P \rightarrow bbP \rightarrow bbbbP \rightarrow bbbbbbP \rightarrow bbbbbbQ \rightarrow bbbbbb$

The strings generated are an even number of a's followed by an even number of b's. The grammar is regular. All the non-terminals are to the left of variables. It is also context free. The grammar is not ambiguous. The number of a's is determined by the number of times the first rule for S is applied. Then we switch to producing b's.





Note: Given a regular language we can always find an unambiguous grammar that accepts that language. This is not the case for context free laguages. There are some context free languages where every context free grammar that accepts that language must be ambiguous.

Problem 5) For the following grammar G

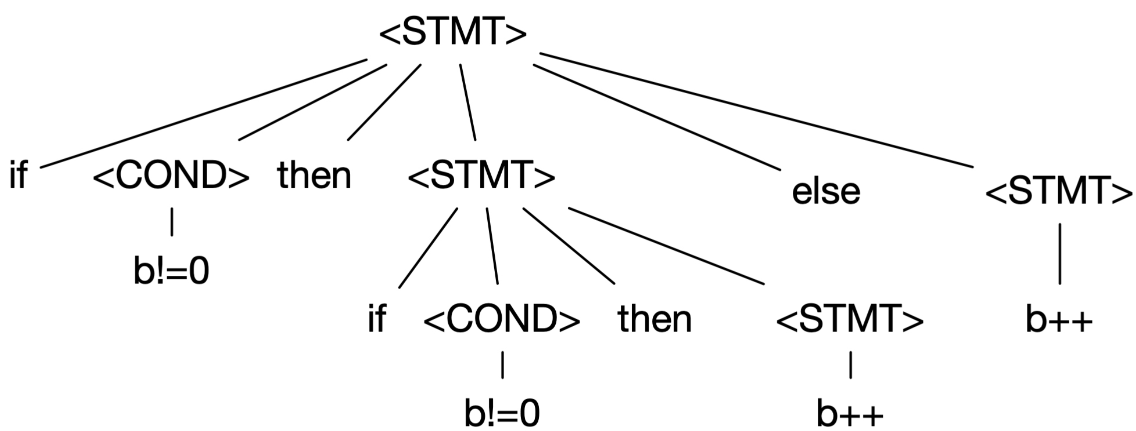
a) Give a parse tree for the string
`if b!=0 then if b!=0 then b++ else b++`

b) Is the grammar regular?

c) Is the grammar ambiguous?

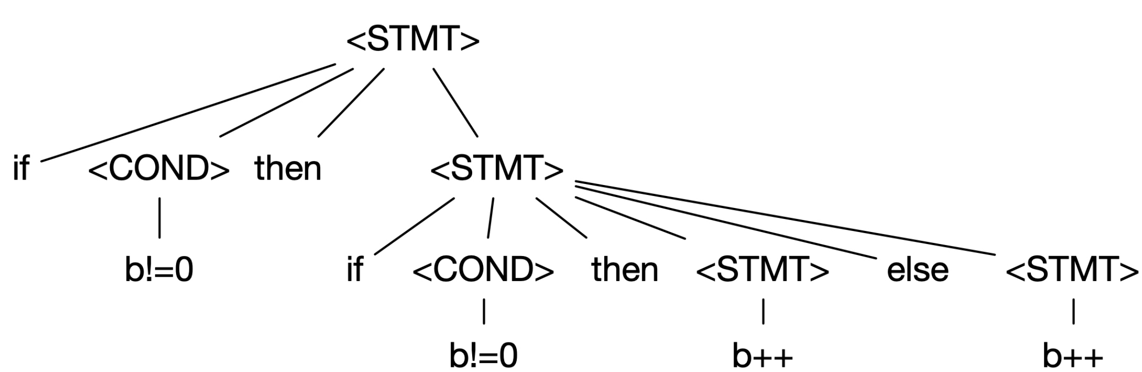
G: $\langle STMT\rangle \rightarrow$ if $\langle COND\rangle$ then $\langle STMT\rangle$
$\langle STMT\rangle \rightarrow$ if $\langle COND\rangle$ then $\langle STMT\rangle$ else $\langle STMT\rangle$
$\langle STMT\rangle \rightarrow$ b++
$\langle COND\rangle \rightarrow$ b! = 0

a) **Here is a parse tree.**



b) The grammar has multiple non-terminals on the RHS of rules. It is context free.

c) It is ambiguous. The string from part a also has the parse tree.



Problem 6) For the following grammar G show that every string that is derivable from S has equal numbers of a's and b's.

G: $S \rightarrow \varepsilon$
$S \rightarrow aSb$
$S \rightarrow bSa$
$S \rightarrow SS$

We can do a proof by induction on the length of a derivation. The shortest derivation is length one:

Base: The only rule that can derive a string of terminals directly is the first rule

Rule: $\boldsymbol{S \rightarrow \varepsilon}$

The empty string has equal numbers of a's and b's.

from S has equal numbers of a's and b's.

Inductive Case: We have three rules that can start the derivation and we need to give a proof for each one.

Rule: $\boldsymbol{S \rightarrow SS}$

Consider a derivation of length k+1. If it starts with this rule then we get two strings of terminals which we will call $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$**.**

$$\boldsymbol{S \rightarrow SS \rightarrow \ldots \rightarrow \alpha\, \beta}$$

We have two derivations that we can intermix to get the final string of terminals.

$$\boldsymbol{S \rightarrow \ldots \rightarrow \alpha}$$
$$\boldsymbol{S \rightarrow \ldots \rightarrow \beta}$$

Each of these must be at least length 1 and at most length k-1 and will be covered by induction. If both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ have the same number of a's and b's, then their concatenation will also have the same number of a's and b's.

Rule: $\boldsymbol{S \rightarrow aSb}$

Consider a derivation of length k+1. If it starts with this rule then we get a strings of terminals which we will call $\boldsymbol{\alpha}$**.**

$$\boldsymbol{S} \rightarrow \text{a}\boldsymbol{S}\text{b} \rightarrow \ldots \rightarrow \text{a}\boldsymbol{\alpha}\text{b}$$

We have a sub derivation.

$$\boldsymbol{S \rightarrow \ldots \rightarrow \alpha}$$

which must be of length k and is covered by induction. If $\boldsymbol{\alpha}$ has the same number of a's and b's, then a$\boldsymbol{\alpha}$b will also have the same number of a's and b's.

Rule: $\boldsymbol{S \rightarrow bSa}$

Similar to the previous rule.

Note: If you have other non-terminals, you will need to prove appropriate facts for them as well as part of your proof by induction. This can get quite complicated and challenging.