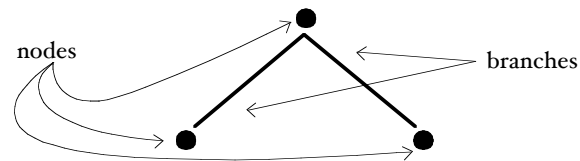


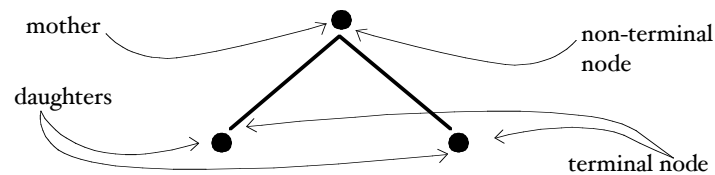
Tree Structures

A Tree structure (or 'tree' for short) is a way of displaying a hierarchical structure. A lot of language structures are hierarchically organized, and we will make use of trees to display this structure.

Trees are made from *nodes* that are connected by *branches*.



A node can have child nodes ('daughters'). If it does, it is called a *non-terminal node*. If a node does not have daughters, it is called a *terminal node* (or 'leaf').



We will only need a limited class of trees. For us, every tree will have exactly one *root node*. A root node is the node that has no mother node.

Every node in a tree has at most one mother node.

The linear order of the terminal elements will be fixed in all cases we consider, and no tree will have crossing branches.

In order to make it easier for us to talk about different nodes, we will allow nodes to have names, so called node labels. (On this handout letters like A, B, C, ..., but later in class more useful things like words, or labels like 'Noun Phrase', 'Verb Phrase', ...)

Some terminology pertaining relations between nodes

Dominance:

A node A *immediately dominates* a node B if and only if A is the mother node of B

A node A *dominates* a node B if

- (a) A immediately dominates B, or
- (b) A dominates a node C which dominates B and not otherwise

A node A is a *sister* of a node B if and only if A and B have the same mother node.

Precedence:

A node A *precedes* a node B if and only if

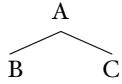
- all terminal nodes dominated by A (if A is non-terminal) or A itself (if A is terminal)
- appear to the left of
- all terminal nodes dominated by B (if B is non-terminal) or B itself (if B is terminal).

A node A *immediately precedes* a node B if and only if there is no node C that precedes B and is preceded by A.

A node A is *adjacent* to a node B if

- (a) A immediately precedes B, or
- (b) B immediately precedes A and not otherwise.

Other ways of writing trees:



Bracket Structure:

[_A B C]

From trees to bracket structures, the 'island' procedure:

Start at the root node and circle the tree counterclockwise, drawing the 'shore of the island'.

- 1) If you encounter a non-terminal node that you have not encountered before (the 'left-hand side' of it), write an opening bracket with the node label.
- 2) If you encounter a terminal node, write the node label.
- 3) If you encounter a terminal node that you have encountered before (the 'right-hand side' of it), write a closing bracket.

(Quick sanity check: Do you have as many opening as closing brackets?)

Structure Generating Rule:

$A \rightarrow B C$ ('A can be expanded to B and C.')

(more general: *mother* \rightarrow *daughter*₁ *daughter*₂ ...)