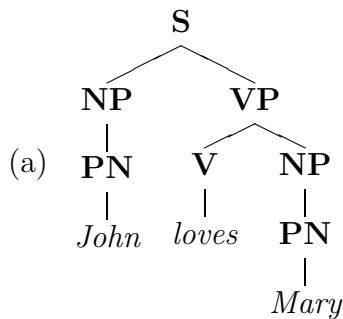


Harvey Mudd College
 Computer Science 80
 Logic for Computer Science
 Fall Semester 1999

Assignment #2 – Lumberjacking
Sample Solution

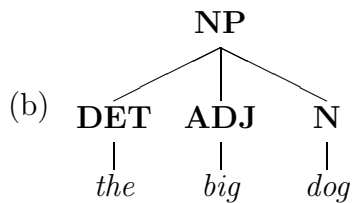
1. For each of the following trees give the tree domain and the graph of the labeling function:



• • •

Tree domain: $\{\epsilon, 1, 2, 11, 21, 22, 111, 211, 221, 2211\}$

Graph: $\{(\epsilon, \mathbf{S}), (1, \mathbf{NP}), (2, \mathbf{VP}), (11, \mathbf{PN}), (21, \mathbf{V}), (22, \mathbf{NP}), (111, \textit{John}), (211, \textit{loves}), (221, \mathbf{PN}), (2211, \textit{Mary})\}$



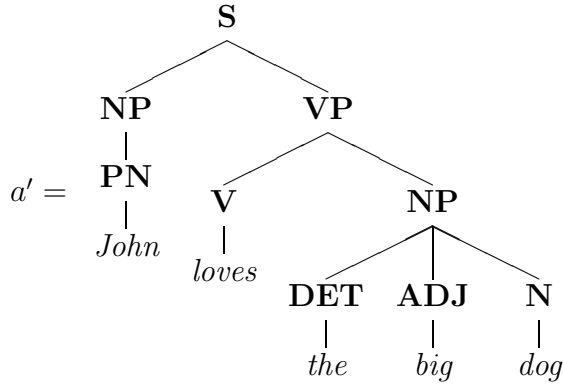
• • •

Tree domain: $\{\epsilon, 1, 2, 3, 11, 21, 31\}$

Graph: $\{(\epsilon, \mathbf{NP}), (1, \mathbf{DET}), (2, \mathbf{ADJ}), (3, \mathbf{N}), (11, \textit{the}), (21, \textit{big}), (31, \textit{dog})\}$

2. Assuming the trees above are called a and b , let the tree $a' = a[22 \leftarrow b]$. Draw the tree a' and give its domain and the graph of its labeling function.

•••



Tree domain: $\{\epsilon, 1, 2, 11, 21, 22, 111, 211, 221, 222, 223, 2211, 2221, 2231\}$

Graph: $\{(\epsilon, \mathbf{S}), (1, \mathbf{NP}), (2, \mathbf{VP}), (11, \mathbf{PN}), (21, \mathbf{V}), (22, \mathbf{NP}),$
 $(111, \textit{John}), (211, \textit{loves}), (221, \mathbf{DET}), (222, \mathbf{ADJ}), (223, \mathbf{N}),$
 $(2211, \textit{the}), (2221, \textit{big}), (2231, \textit{dog})\}$

3. Assuming a restriction to binary trees, give the specification of the total order $\leq_{inorder}$ on the nodes of a tree t corresponding to the ordering generated by an inorder depth-first traversal of the tree.

• • •

The following definition can be justified by appealing to the notion of *greatest common ancestor*. It can probably be expressed in other slightly different, but equivalent, forms.

$u \leq_{inorder} v$ if and only if one of the following mutually exclusive conditions holds:

- There is a (possibly empty) string $x \in \mathcal{N}_+^*$ such that $u = v.1.x$
That is, u is a left descendant of v .
- There is a (possibly empty) string $x \in \mathcal{N}_+^*$ such that $v = u.2.x$ That is, v is a right descendant of u .
- There are (possibly empty) strings $x, y, z \in \mathcal{N}_+^*$ such that $u = x.1.y$ and $v = x.2.z$
That is, the string x is the longest common prefix of u and v , and thus is the address of the common ancestor of u and v that is deepest in the tree, and u is a left-descendant of x while v is a right-descendant of x .
- $u = v$

4. Prove that though tree addresses are finite strings, trees may have both (or either) infinitely long branches and nodes with an infinite number of children.

• • •

Note that it is impossible to prove this result by a positive application of induction. Showing that for any tree there is one with a longer branch or a bushier node only shows that we can have an arbitrarily large finite length branch or finite branching node. It does not construct one with an infinite branch or infinitely bushy node. This is an important limitation of induction.

We can, however, easily give a proof-by-example of each.

- (a) For a tree with an infinitely long branch, choose a tree with domain:

$$\{1\}^* = \{\epsilon, 1, 11, 111, 1111, \dots\}$$

This is just the degenerate tree with only a single, infinite, path.

- (b) For a tree with an infinitely bushy node, choose a tree with domain:

$$\{\epsilon\} \cup \mathcal{N}_+ = \{\epsilon, 1, 2, 3, \dots\}$$

This is a two level tree with an infinite number of children of the root.

- (c) For a tree with an infinitely bushy node, and an infinitely long path, choose a tree with domain equal to the union of the previous two domains:

$$\{1\}^* \cup \mathcal{N}_+$$

In each case it is easy to see that the proposed tree domain is a legal and well-defined tree domain, yet specifies a tree with the desired properties. (You should show this. I omit the details here.)

It is also possible to prove each of these by contradiction. For example, assume that a tree cannot contain a node with an infinite number of children. Then, for a given tree t and any given node with address u there must be a maximum number, n , of children it can have. Add the addresses ui to t for all $1 \leq i \leq n + 1$. Assuming t was well-formed, then this tree domain is as well, and yet u has $n + 1 > n$ children, which contradicts our assumption.

5. We say that tree t is a subtree of tree s if there is a tree address u in the domain of s such that t is the subtree rooted at u (as defined in class). Prove that the relation “is a subtree of” is transitive.

• • •

Proof:

Suppose tree t is a subtree of tree s rooted at u , and tree s is a subtree of tree r rooted at v .

Claim: Tree t is a subtree of tree r rooted at $v.u$.

Proof of Claim:

We need to show that the domains of t and $r/v.u$ are the same, and that their labeling functions are the same on their domains:

$$\begin{aligned} \text{dom}(t) &= \{x \mid (v.u).x \in \text{dom}(r)\} = \text{dom}(r/v.u) \\ t(x) &= r/v.u(x) \quad \forall x \in \text{dom}(t) \end{aligned}$$

Since $t = s/u$ and $s = r/v$, we know

$$\begin{aligned} \text{dom}(t) &= \text{dom}(s/u) = \{y \mid u.y \in \text{dom}(s)\} \\ \text{dom}(s) &= \text{dom}(r/v) = \{z \mid v.z \in \text{dom}(r)\} \end{aligned}$$

Then, immediately,

$$\text{dom}(t) = \{y \mid v.(u.y) \in \text{dom}(r)\} = \{y \mid (v.u).y \in \text{dom}(r)\}$$

as needed. We also know

$$\begin{aligned} t(x) &= s/u(x) = s(u.x) \quad \forall x \in \text{dom}(t) \\ s(y) &= r/v(y) = r(v.y) \quad \forall y \in \text{dom}(s) \end{aligned}$$

Since $x \in \text{dom}(t)$ implies $u.x \in \text{dom}(s)$, we have

$$t(x) = s(u.x) = r(v.(u.x)) = r((v.u).x) = r/v.u(x) \quad \forall x \in \text{dom}(t)$$

as needed.

Now given t a subtree of s , and s a subtree of r , we can find t as a subtree of r , so the subtree relation must be transitive.

Q.E.D.