Psycholinguistics and Linguistic Theory, I Fall 2009
Short Answer Assignment, #1, "Representation and Process"

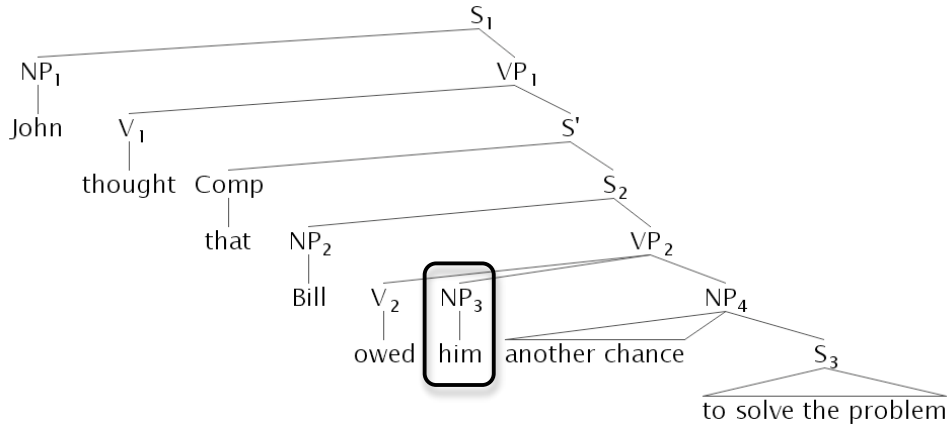
Due 1 October, 2009. Hand in your response, or email it to mwagers@ucsc.edu, by start of class. You may talk with other classmates about the assignment, but you must complete the numbered problems on your own.

Recall what Marr (1982) says about representations: "A representation is a formal system for making explicit certain types of information, together with a specification of how the system does this. And I shall call the result of using a representation to describe a given entity a *description* of the entity in that representation (Marr & Nishihara, 1978)."

Now, consider the following sentence: "John thought that Bill owed him another chance to solve the problem." A phrase structure tree is given below.

Imagine you are trying to understand this sentence, and you arrive at pronoun 'him.' You want to figure out who 'him' refers to, and, specifically, whether it refers to any entities named in the sentence. From your grammar, you know that (roughly) a pronoun cannot co-refer with a c-commanding antecedent in the same clause (~Principle B). Now, let's restate that in information processing terms.

Input: { The tree below, the pronoun 'him' (i.e., NP_3) }
Output: NP_1 [=='John']



1. Write a step-by-step procedure for mapping the **Input** above to the **Output.** What does Marr call such a procedure?

Note: I'm **not** asking for a formal program, but a description in plain English. But you'll probably want to use a numbered list to organize the steps and be able to refer to them. You are allowed latitude in what kind of concepts you appeal to (e.g. *top*, *bottom*, XP, X …). Just be clear and consistent.

Hint #1: You may have learned to solve 'Binding Theory problems' in your Linguistics coursework – start by introspecting on how it is you would know whether or not there's a legitimate binder for the pronoun in this sentence. Now imagine someone who doesn't know Binding Theory, or even syntactic theory, but can read a tree diagram. How would you tell them to solve the problem?

Hint #2: Think about the tree in terms of terms of its nodes (NP_1, V_1, etc.) and the branches that lead from one node to the next. Which node will your procedure start on? How will it decide which node to look at next? How will it know when it's finished?

2. A good procedure for applying Principle B should also handle the following input/output pairs. How would you amend your procedure from (1) to handle these cases? Draw the trees if it helps.

Pair 1
Input: (Tree for: "Beth thought that Bill owed [$_{NP3}$ **him**] another chance." , [$_{NP3}$ **him**])

Output: 0 [== no possible antecedent in this sentence]

Pair 2
Input: (Tree for: "Beth thought that Bill's brother owed [$_{NP3}$ **him**] another chance." , [$_{NP3}$ **him**])

Output: Bill

Pair 3
Input: (Tree for: "John thought that Bill's brother owed [$_{NP3}$ **him**] another chance." , [$_{NP3}$ **him**])

Output: John, Bill

3. Take your algorithm from (2). What does it output for the following input pairs:

Input 4 (Tree for: "The man who Robert met thought that Bill's brother owed [$_{NP3}$ **him**] another chance.", [$_{NP3}$ **him**])

Input 5 (Tree for: "Beth thought that Bill owed [$_{NP3}$ **her**] another chance.", [$_{NP3}$ **her**])

4. Marr writes: "… the choice of algorithm often depends rather critically on the particular representation that is employed" (1982, p. 71) What aspects of tree structure are well-suited to implementing the Principle B restriction? What aspects are ill-suited?

Extra Credit (choose one)

[a] Later in the term, we will read some papers on how humans determine pronominal co-reference (Sturt, 2003). For each of the inputs in 1-3, how many steps does your (final) algorithm require to arrive at an output? Suppose each step takes 50 ms (milliseconds) to execute; how long does the procedure require in total?

[b] How else could the example sentences be represented? Come up with an alternate representational format. How does it change the algorithm?