„Semantic Theory“ SS 06 Exercise 5 (23/05/2006)

`http://www.coli.uni-saarland.de/courses/semantics-06/`

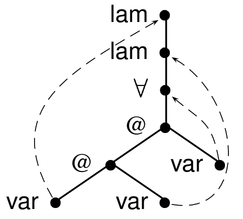
1 Lambda terms and lambda structures

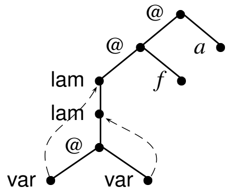
For each of the following lambda terms, give the corresponding lambda structure:

(a) $\forall z.\mathsf{dolphin}(z) \rightarrow (\mathsf{mammal}(z) \wedge \mathsf{live\_in\_sea}(z))$

(b) $\lambda x.(f(\lambda x.x)(x))$

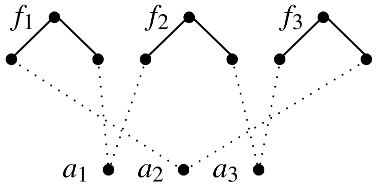
For each of the following lambda structures, give a corresponding lambda term:

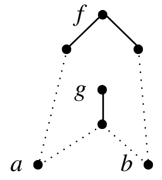
(c) 

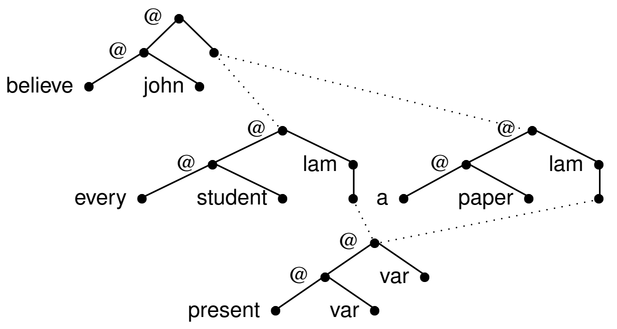
(d) 

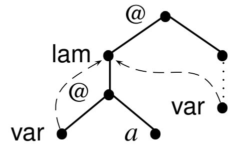
2 Solvability of dominance graphs

For each of the following dominance graphs, decide whether it is solvable or unsolvable. If it is solvable, give a lambda structure that solves it. If it is unsolvable, explain why you think it is unsolvable.

(a) 

(b) 

(c) 

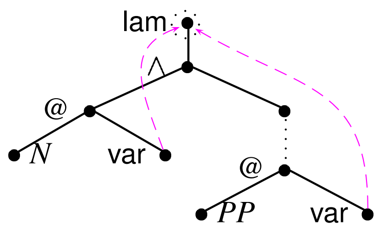
(d) 

3 Semantics construction

Derive a dominance graph that describes the five readings of the following sentence:

> Every researcher of a company sees a sample.

The lexicon entry for the words "of" and "sees" are dominance graphs consisting of a single node with labels of^* and see^*, respectively; of^* and see^* are both constants of type $\langle e, \langle e, t\rangle\rangle$. The semantic construction rule for PP → P NP introduces an application, in the same way as the rules for sentences and verb phrases from the lecture. The rule for N' → N PP looks as follows:



Assign a number to each node in the syntax tree, and mark the interface node that belongs to each syntax node with its number.

4 Solving dominance graphs

Use the dominance graph solver described in the lecture to enumerate the five solved forms of the graph from Question 3. It is sufficient to give the dominance graphs that occur as arguments to recursive calls of the solver procedure; you don't have to spell out applications of Parent Normalisation and Redundancy Elimination. You may abbreviate the tree fragments (i.e. the subgraphs that are connected via tree edges) by triangles as in the lecture, but make sure that the tree fragments and their holes can still be identified.

5 Type-raised noun phrases

In the lectures on underspecification, we deviated from our earlier analysis of transitive verbs as expressions of type $\langle\langle\langle e, t\rangle, t\rangle, \langle e, t\rangle\rangle$. Instead, we applied the verb representations directly to variables or constants of type e because this made for more readable analyses. This works well for transitive verbs with referential objects, but as you know, it will get us into trouble for verbs with non-referential objects, such as "seek".

So modify the semantics construction rules from the lecture in such a way that they work with transitive verbs analysed as expressions of type $\langle\langle\langle e,t\rangle,t\rangle,\langle e,t\rangle\rangle$. Your rule system should derive underspecified descriptions of exactly the same expressions that would have been derived by last week's Cooper Storage analysis. *Hint:* It may be helpful to maintain the invariant that the subtree below an NP interface node is always an expression of type $\langle\langle e,t\rangle,t\rangle$ (by contrast, the rules from the lecture assume is that the trees below an NP interface node are always of type e).

6 * Solved forms and solutions

In the lecture, we stated that a dominance graph without binding edges has a solution if and only if it has a solved form, but we didn't prove this. Convince yourself that it is in fact true as follows:

(a) Given a dominance graph in solved form, show how to construct a solution of this graph.

(b) Given a solution of a dominance graph, show how to construct a solved form of the graph.

Your algorithms should be correct for arbitrary graphs and solutions. Demonstrate them on small but non-trivial examples.

To be turned in by 06/06/2006, 11:15 am