Semantiek Tutorial: Lambdas

Exercise 1

Give an informal description of the following functions

Example:
$\lambda x.\lambda y.y : e(ee)$ is a function that takes any two objects of type e and returns the second one as its result.

---

1. $\lambda x.\lambda y.x : t(et)$
2. $\lambda x.\lambda y.\,(\,x(y)\,)(y) : (e(et))(et)$

Exercise 2

Now do the opposite: give lambda terms with types corresponding to the following informal descriptions:

1. A function that takes an object of any type and returns that same object
2. A function that takes any two objects of type e and returns 1
3. A function that takes an object of type e and a function of type $e(et)$, applies the function to the object twice, and returns the result.
4. A function that takes a char. function of a binary relation over entities, and returns the char. function of the reverse of that relation.

Exercise 3

Derive the type of the following terms given the information on the types of variables and constants

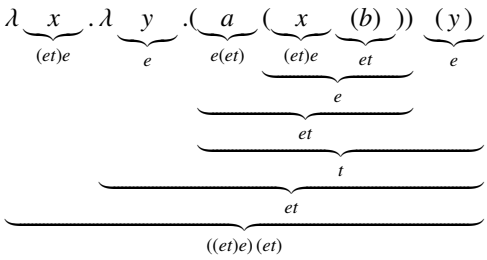
Example: given the following type assignment:

$x : (et)e, y : e, a : e(et), b : et$

the type of the following term:

$\lambda x.\lambda y.(a\ (x(\ b\ )))\ (y)$

can be derived as follows:



1. given the assignment
   $$x : (et)t, y : e(et), z : e$$
   find the type of
   $$\lambda x.\lambda y.\lambda z.x\ (\lambda w.y(\ w(\ z\ )\ )\ )$$
2. given the assignment
   $$y : (et)t$$
   find the type of
   $$\lambda x.y\ (\lambda z.x)$$

Exercise 4

In this exercise we give a new account of the familiar "Tina is tall and thin" ⇔ "Tina is tall and Tina is thin" equivalence using lambda's.
To do this, we will need to give denotations for two different uses of "and", as the first is a coördination between adjectives, and the second is a coördination between sentences. As sentences have truthvalues as denotations, the first case is simple: it denotes logical conjunction. The following types and denotations are given:

Word	Type	Denotation
"Tina"	e	**t'**
"tall"	et	**tall'**
"thin"	et	**thin'**
"and$_{sent}$"	$t(tt)$	$\lambda x.\lambda y.(\wedge_{ttt}(x))\ (y)$
"is"	$(et)(et)$	???
"and$_{adj}$"	???	???

Here we assume $\wedge_{ttt}$ is just propositional conjunction in the form of a function. It is defined as:

$$\wedge_{ttt} = \{\, 0 \mapsto \{0 \mapsto 0, 1 \mapsto 0\}, 1 \mapsto \{0 \mapsto 0, 1 \mapsto 1\}\,\}$$

1. Given that "is" denotes the identity function, what would be the the lambda term for "is"?

2. Give the structure for "Tina is tall and Tina is thin" as a tree and decorate each node with the appropriate types and denotations. Start by adding the lexicon entries at the leaves, then work your way up by repeatedly using function application. Use beta reduction where possible to keep the terms simple.

3. At the top you should now have a big lambda term of type t that represents the denotation of the whole sentence. Simplify it as much as possible if you haven't already. What are the truth conditions of the sentence? (that is: under which conditions will this term be true, and under which conditions will it be false?

4. Give a structure for "Tina is tall and thin", decorate it with the appropriate types and lambda terms where possible and use this information to infer the missing type for and_{adj}.

5. Give the denotation for and_{adj} as a lambda term. There are several pieces of information that we can use to solve this puzzle:

   - We know its type.
   - Because the sentences are equivalent, we know that the truth conditions of both sentences should be the same. What does this imply for the lambda term of the whole sentence?
   - If you know what the term at the top of the structure is and what the lambda terms at the other nodes are, how can you use this information to work your way down the tree to the missing entry?