Homework for Wednesday, October 14

1 Type theory and lambdas

- What are the types of the following expressions? (If you get stuck, think about what sorts of syntactic frames these expressions occur in.)

  1. devour
  2. fond
  3. part of New Brunswick
  4. show Porky
  5. white cat from Providence

- Evaluate the following claims.

  ▷ If the expressions are equal, show how to derive the latter from the former by applications of α-, β-, or η- equivalence.

  ▷ If not, say why not.

  1. $(\lambda x.\, \lambda y.\, \textit{licks}'\, x\, y)\, \mathrm{q}\, \mathrm{v} \equiv \textit{licks}'\, \mathrm{q}\, \mathrm{v}$
  2. $(\lambda x.\, f\, x\, y)\, y \equiv f\, y\, y$
  3. $(\lambda x.\, \lambda y.\, f\, x\, y)\, y \equiv \lambda z.\, f\, y\, z$
  4. $\lambda x.\, \textit{kiss}'\, x \equiv \lambda y.\, \lambda x.\, \textit{kiss}'\, y\, x$
  5. $\lambda x.\, \textit{kiss}'\, x \equiv \lambda x.\, \lambda y.\, \textit{kiss}'\, y\, x$

- Simplify the following expressions as much as possible.

  ▷ Show (and justify) each step in your calculation (you might have to do more than one β-reduction!).

  ▷ Exploit α-equivalences as needed to avoid variable capture.

  ▷ Be careful. Some of these are tricky.

  1. $(\lambda P.\lambda x.P\, x)\, \textit{run}'$
  2. $(\lambda R.R\, a\, b)\, (\lambda y.\, \lambda x.\, \textit{kiss}'\, y\, x)$
  3. $(\lambda f.f\, x)\, (\lambda y.\, \lambda x.\, g\, x\, y)$
  4. $(\lambda \mathcal{P}.\, \mathcal{P}\, (\lambda p.\, p))\, (\lambda k.\, k\, (\textit{meows}'\, x))$
  5. $(\lambda m.\, \lambda n.\, m\, (\lambda f.\, n\, (\lambda x.\, f\, x)))\, (\lambda k.\, k\, (\textit{left}'))\, (\lambda k.k\, x)$

- Any function f has a type that we can write as $\langle \sigma, \tau \rangle$ (for some type σ and some type τ). Can a function ever apply to itself? Why or why not?

2 Composition inside DP

- Calculate ⟦Uni is a big white cat from Providence⟧, labeling each node in the tree with its type and denotation (as in the Oct 5 handout).

  ▷ Assume the available combination operations are **Functional Application** (**FA**) and **Predicate Modification** (**PM**).

  ▷ Assume whatever semantics you like for is and a (so long as it works!).

  ▷ Indicate which composition operation (**FA** or **PM**) you used to interpret each binary-branching node.

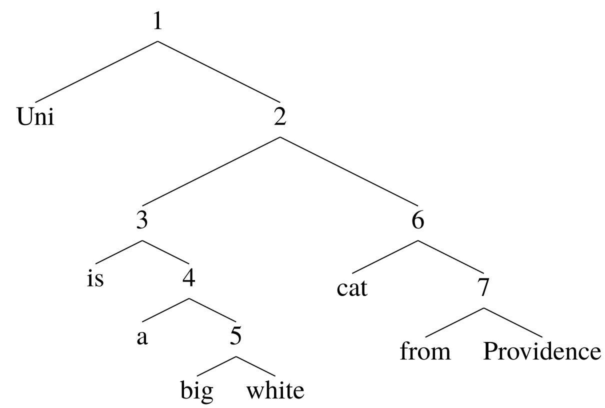
- Now, suppose that you only have **FA** in your toolbox.

  ▷ Devise a silent morpheme $\text{PRED}_{\emptyset}$ which allows you to give a meaning for ⟦Uni is a big white cat from Providence⟧ anyway.

  ▷ Give a derivation (again labeling each node).

  ▷ Did you get the same result as before?

- Both the grammar with **FA** and **PM** and the grammar with **FA** and $\text{PRED}_{\emptyset}$ can glue this sentence together in a way you might not have expected. For example, the tree below is interpretable using **FA** and **PM**.



  ▷ Assign an interpretation to the tree, giving a type and meaning to each numbered node.

  ▷ Does the interpretation differ from the previous examples?

  ▷ Do you find this structure plausible? Why or why not?