PPL – Recitation Four

Abdall

February 19, 2019

Goals for this Recitation

Lots of Random Things

- Python discussion on Duck Typing
- Rewrite Systems (not covered in lecture)
- Start some Lambda Calculus

Python Duck Typing

Forgiveness not Permission

- The idea that we should just try things
- If it looks like a duck, swims like a duck, and quacks like a duck, then it probably is a duck

Different from Java

- When talking about error handling in Java we often want to throw an error as soon as possible

Different from Java

- When talking about error handling in Java we often want to throw an error as soon as possible
- Functions don't really work when they do not get the parameters specified

Different from Java

- When talking about error handling in Java we often want to throw an error as soon as possible
- Functions don't really work when they do not get the parameters specified
- This is dissimilar than that of Python

Rewrite Systems

Working our way up

- This is moreso just a way to practice with the logic we will eventually need to use on Lambda Calc
- Gets us into the mindset of rewriting things and how replacement work

An example to get us started

- Let us say we have the following rules:

$$\$1 \Rightarrow 1\&$$
$$\$0 \Rightarrow 0\$$$
$$\&1 \Rightarrow 1\$$$
$$\&0 \Rightarrow 0\&$$
$$\$\# \Rightarrow A$$
$$\&\# \Rightarrow B$$

- Let us say we want to translate the string: $\$0101\#$

Attempting to make our own

- We want to make a rewrite system that tells us if a binary number is odd or even
- Again we can expect input such as \$0101\#

Lambda Calculus

The Basics

- We have three main items that we should be concerned with:
  - Variables
  - Abstractions
  - Applications
- For instance when we have $\lambda x.x$ this is a function that takes and returns x
- But we can actually apply this by saying something like $(\lambda x.x)y$

Wow I hate Parenthesis

- We nee to make something clear:
  THIS: $(\lambda x.y)z$
  AND THIS: $\lambda x.yz$
  NOT THE SAME.

Honestly why...

- We also should note that our arguments can be functions too...

Honestly why...

- We also should note that our arguments can be functions too...
- (who let this happen)
- So something of the form: $(\lambda x.x)(\lambda x.y)$ is totally valid

Honestly why...

- We also should note that our arguments can be functions too...
- (who let this happen)
- So something of the form: $(\lambda x.x)(\lambda x.y)$ is totally valid
- But this gets even better :)

Honestly why...

- We also should note that our arguments can be functions too...
- (who let this happen)
- So something of the form: $(\lambda x.x)(\lambda x.y)$ is totally valid
- But this gets even better :)
- We can have something like: $(\lambda x.xz)(\lambda x.y)$