Lecture Notes #14B

These are not full lecture notes. These are an outline for my lecture. I've omitted complicated LaTeX machinery like typing rules—I can remember these on the day of lecture.

#1 Natural deduction Rules of propositional logic

How to read natural deduction rules; Logical rules for:

1. $\top$, $\bot$, $\wedge$, $\vee$, and $\Rightarrow$.

#2 The syntax of the untyped lambda calculus

Grammar

$$x \in \Sigma^*$$
$$M ::= x \mid \lambda x.M \mid M\,M$$

#3 Natural deduction & Operational semantics

#3.1 Substitution

Define substitution of N over x in M, written $M[N/x]$, inductively as:

$$\begin{aligned} x[N/x] &= N \\ (\lambda y.M)[N/x] &= \lambda y.(M[N/x]) \\ (M_1\,M_2)[N/x] &= M_1[N/x]\,M_2[N/x] \end{aligned}$$

(We follow the Barendregt convention and ignore the very-real challenge of variable capture.)

#3.2 Beta reduction

For lambda terms M, N define $M \to_\beta N$ as:

$$\frac{}{(\lambda x.M)[N/x] \to_\beta M[N/x]}$$

and the other congruence rules.

#3.3 Reflexive, transitive closures

We can define $\to_\beta^*$ as the RTC of the $\to_\beta$ relation.

#3.4 A comparison to "traces" and "configuration histories"

We can think of $\to_\beta^*$ as expressing the same idea as the yields relation $\Rightarrow^*$ on TM configurations.

#3.5 Normal forms, normalization

We say a term M is in normal form if there does not exist N such that $M \to_\beta N$. We say that a term M diverges if it has no normal form. We say that a calculus is normalizing if all terms have normal forms.

To the students: do you remember this claim in Lab 13A?

Claim 1. *A Turing machine that halts does not repeat a configuration. OR: A Turing machine that repeats a configuration does not halt.*

Draw a comparison.

#4 The Simply Typed Lambda Calculus

Doubtful I'll get this far.

#4.1 Type systems: what are they good for?

Type errors and nonsensical terms, etc.

#4.1.1 Divergence

Define $\Omega = \lambda x.x\,x$ and then consider the reduction of $\Omega\Omega$.

$$(\lambda x.x\,x)(\lambda x.x\,x) \to_\beta (\lambda x.x\,x)(\lambda x.x\,x) \to_\beta \ldots \to_\beta (\lambda x.x\,x)(\lambda x.x\,x) \to_\beta \ldots$$

So this term diverges. We can think of this as a type error: In the term $\lambda x.x\,x$, x is behaving as both a function and an applicand. A type system can help us rule out this "type error".

#4.2 Typing Rules

Let's define the grammar of types and terms with types.

$$\begin{aligned} x &\in \Sigma^* \\ M &::= x \mid \lambda x{:}T.M \mid M\,M \\ T &::= \circ \mid T \to T \end{aligned}$$

And now we introduce a relation $\Gamma \vdash M : T$ where the environment Γ has the following rules:

$$\ldots$$

and $\Gamma \vdash M : T$ has the rules:

$$\ldots$$

#5 The Curry-Howard Correspondence: Minimal logic and the STLC with products and sums

Implication, products, sums, etc.