Type Theory (CM0859) – Exercises

Andreas Abel

March 2017

1 Natural deduction

Exercise 1 (Natural deduction derivations)**.** Give derivations of the following propositions:

1. $A \Rightarrow A$.
2. $(A \wedge (A \Rightarrow B)) \Rightarrow B$.
3. $(A \wedge (B \vee C)) \Rightarrow (A \wedge B) \vee (A \wedge C)$.
4. $(\neg A \vee B) \Rightarrow (A \Rightarrow B)$.

Exercise 2 (De Morgan laws)**.** Which of the following de Morgan laws are constructively valid?

1. $\neg(A \vee B) \Rightarrow (\neg A \wedge \neg B)$.
2. $\neg(A \wedge B) \Rightarrow (\neg A \vee \neg B)$.
3. $(\neg A \wedge \neg B) \Rightarrow \neg(A \vee B)$.
4. $(\neg A \vee \neg B) \Rightarrow \neg(A \wedge B)$.

For the ones you consider valid, give natural deduction proofs. For the ones you consider constructively invalid, offer an argument why you think so.

Exercise 3 (Local soundness and completeness)**.**

1. Invent an elimination rule for conjunction which is too strong and argue that it lacks local soundness.
2. Give a set of elimination rules for conjunction which is too weak and argue that it lacks local completeness.
3. Do the same (1. and 2.) for disjunction.

Exercise 4 (Rules with explicit assumptions)**.** Write down the natural deduction rules for judgement $\Gamma \vdash A \ \mathit{true}$.

2 Classical logic

Exercise 5 (Equivalent formulations of classical logics)**.** Consider the following abbreviations for "classical" formulæ:

$\mathsf{EM}(A)$	:=	$A \vee \neg A$	excluded middle
$\mathsf{RAA}(A)$	:=	$(\neg A \Rightarrow \bot) \Rightarrow A$	*reductio ad absurdum*
$\mathsf{RAA}'(A)$	:=	$(\neg A \Rightarrow A) \Rightarrow A$	*reductio ad absurdum* (variant)
$\mathsf{Pierce}(A, B)$	:=	$((A \Rightarrow B) \Rightarrow A) \Rightarrow A$	Pierce's formula

Assuming one of these formulas categorically, for instance, assuming that $\mathsf{EM}(A)$ holds for all formulæA, makes the logic classical.

Prove constructively, either by drawing natural deduction derivations, by giving typed λ-terms, or by writing appropriate functions in Agda:

1. $\neg\neg\mathsf{EM}(A)$ holds for any formula A, constructively.

2. All four versions of classical logic are equivalent.
   This can be proven by a chain of implications, for instance:

   - RAA implies EM.
   - EM implies Pierce.
   - Pierce implies RAA'.
   - RAA' implies RAA.

Exercise 6 (Direct proof)**.** Using Pythagoras' theorem, the statement

> In any non-degenerate right triangle the hypothenuse is shorter than the sum of the catheti.

can be formally expressed as "$a, b, c > 0$ and $a^2 + b^2 = c^2$ imply $a + b > c$". Here is a proof by contradiction:

> Assume the contrary, $a+b \leq c$. Then $(a+b)^2 = a^2+2ab+b^2 \leq c^2$, thus, $2ab \leq 0$. This contradicts $a, b > 0$.

Transform this into a direct, constructive proof!

3 Lambda-calculus

Exercise 7 (Lambda terms)**.** Find closed lambda terms of the following types:

1. $(S \to T) \to ((T \to U) \to (S \to U))$.
2. $S \to (T \to (S \times T))$.
3. $(S \times (T + U)) \to ((S \times t) + (S \times U))$.
4. $((S \to 0) + T) \to (S \to T)$.

Exercise 8 (Substitution and free variables)**.** Consider the untyped lambda-calculus with tuples and variants.

1. $\mathsf{FV}(t)$ computes the set of free variables of t. Write out the full definition of FV!
2. $t[s/x]$ substitutes term s for all free occurrences of variable x in term t. Write out the full definition of $t[s/x]$.
3. Prove that $\mathsf{FV}(t[s/x]) \subseteq \mathsf{FV}(s) \cup (\mathsf{FV}(t) \setminus \{x\})$.
4. Give an example for $\mathsf{FV}(t[s/x]) \subsetneq \mathsf{FV}(s) \cup (\mathsf{FV}(t) \setminus \{x\})$.

Exercise 9 (Scoping)**.** Prove: If $\Gamma \vdash t : T$ then $\mathsf{FV}(t) \subseteq \mathsf{dom}(\Gamma)$.

Exercise 10 (Inversion of typing)**.** If $\Gamma \vdash \lambda x.t : U$, then $U = S \to T$ for some types S, T with $\Gamma, x{:}S \vdash t : T$.

1. Prove this inversion law!
2. Find inversion law for all other term constructors!

Exercise 11 (Capturing substitution)**.** Consider substitution defined with $(\lambda y.t)[s/x] = \lambda y.t[s/x]$ regardless of whether $x = y$ or $y \in \mathsf{FV}(s)$. Show by example that subject reduction is broken, i. e., find Γ, t, t', and T such that $\Gamma \vdash t : T$ and $t \longrightarrow t'$, but not $\Gamma \vdash t' : T$.

Exercise 12 (Subject reduction)**.** Prove the subject reduction theorem for simply typed lambda-calculus: If $\Gamma \vdash t : T$ and $t \longrightarrow t'$ then $\Gamma \vdash t' : T$.

Exercise 13 (Progress)**.** Prove the progress theorem: If $\Gamma \vdash t : T$ and $t \longrightarrow t'$ then $\Gamma \vdash t' : T$.

4 Logical Framework

Exercise 14 (HOAS representation of the untyped lambda-calculus)**.** We wish to encode untyped lambda terms

$$\mathsf{Tm} \ni t ::= x \mid \lambda x.t \mid t\,t'$$

via higher-order abstract syntax in the Logical Framework. Terms are represented as inhabitans of a new type Tm.

1. Give constants with their type that act as constructors for untyped lambda terms.

2. Represent the following untyped lambda terms using your constructors in LF:

   - $\lambda x.x$
   - $\lambda f.\lambda x.\, f\, x$
   - $\lambda f.(f\,(\lambda x.\, f\, x))$

Exercise 15 (HOAS representation of the type assignment judgement)**.** (Continues the previous exercise.) Consider simple types:

$$\mathsf{Ty} \ni T ::= \bigstar \mid T \to T'$$

1. Represent Ty with its constructors in LF!

2. Represent the typing judgement $\Gamma \vdash t : T$ in LF and give one constant for each typing rule of the simply typed lambda calculus.

References