

Exercises 3 — Applied Logic

1. *Lazy computation in Gödel's T.* Reduce each of terms $(s\ 0) + (s\ (s\ 0))$ and $(s\ 0) \cdot (s\ (s\ 0))$ to lazy normal form using the fewest possible number of steps/inferences. Use definition of addition and multiplication given in (Coquand, Dybjer, Palmgren: Type-theoretic foundations of constructive mathematics, 2007).
2. *Abstract realizability of arithmetic.* Find lambda term witnesses to the validity of the following propositions or judgements. Use Heyting Arithmetic with witnesses as described on page 28-30 of (Coquand *et al.* 2007).
 - (a) $A \supset \neg\neg A$,
 - (b) $\neg\neg\neg A \supset \neg A$,
 - (c) $A \wedge (B \vee C) \supset (A \wedge B) \vee (A \wedge C)$.
 - (d) $A \wedge B \supset C \vdash_X A \supset (B \supset C)$.
 - (e) $(\exists y)A(y) \vee (\exists y)B(y) \vdash_X (\exists y)(A(y) \vee B(y))$
 - (f) $(\forall y)A(y), (\forall y)(A(y) \supset B(y)) \vdash_X (\forall y)B(y)$.
3. The *Ackermann function* $\text{ack} : \mathbb{N} \rightarrow \mathbb{N} \rightarrow \mathbb{N}$ can be defined by the following recursion equations

$$\begin{aligned} \text{ack } 0\ n &= s\ n \\ \text{ack } (s\ m)\ 0 &= \text{ack } m\ (s\ 0) \\ \text{ack } (s\ m)\ (s\ n) &= \text{ack } m\ (\text{ack } (s\ m)\ n) \end{aligned}$$

It can be shown that the Ackermann function grows faster than any primitive recursive function. Show that it nevertheless may be defined in Gödel's system T with the help of the recursion operator *rec*. [Hint: expand the third line of the definition.]

Can you compute a lazy normal form of $\text{ack } 5\ 5$ using your definition?

4. *Propositions as ordinary sets* (p. 35, Coquand *et al.* 2007). The proposition $(\forall n \in \mathbb{N})[n = 0 \vee (\exists m \in \mathbb{N})n = m + 1]$ says that each natural number is 0 or a successor of another natural number. The corresponding set becomes

$$(\prod n \in \mathbb{N}) [E_{n,0} + (\sum m \in \mathbb{N}) E_{n,m+1}].$$

Exercise: show that it contains exactly one element.
