UPPSALA UNIVERSITET
Matematiska institutionen
Lars Lindqvist

FACIT ÖVNINGAR 1
Tillämpad Logik DV1, ht-00
2000-09-07

Facit till övningsuppgifter i Tillämpad Logik DV1

1. Vi definierar mängden av strängar i S som förlänger u:

   $$S_u = \{v \in S : u \text{ är initialsegment av } v\}.$$

   Enligt antagande är $S_\varepsilon = S$ oändlig. Vidare, om S_u är oändlig, så är åtminstone en av S_{u0} eller S_{u1} oändlig. Nu kan vi konstruera den önskade följden $b_1 b_2 b_3 \cdots$ induktivt. Antag att för $u = b_1 b_2 \cdots b_n$ så är S_u oändlig. Välj nu $b_{n+1} \in \{0, 1\}$ sådan att $S_{ub_{n+1}}$ är oändlig. Eftersom $u \in S_u \subseteq S$, när $S_u \neq \emptyset$, gäller att $b_1 b_2 \cdots b_n \in S$ för alla n. (Om man ser strängarna som vägbeskrivningar i ett binärt träd, kan resultatet betraktas som en version av Königs lemma för träd. Detta är ett typiskt icke-konstruktivt resultat.)

2. $add2 = \lambda x \to \mathsf{rec}(x, \mathsf{S}(\mathsf{S}(0)), \lambda y \to \lambda z \to \mathsf{S}(\mathsf{z}))$

   $\mathsf{apply}(add2, \mathsf{S}(0)) =$
   $\mathsf{rec}(\mathsf{S}(0), \mathsf{S}(\mathsf{S}(0)), \lambda y \to \lambda z \to \mathsf{S}(\mathsf{z})) =$
   $\mathsf{S}(\mathsf{rec}(0, \mathsf{S}(\mathsf{S}(0)), \lambda \mathsf{y} \to \lambda \mathsf{z} \to \mathsf{S}(\mathsf{z}))) =$
   $\mathsf{S}(\mathsf{S}(\mathsf{S}(0)))$

3. $add = \lambda x \to \lambda y \to \mathsf{rec}(x, y, \lambda y \to \lambda z \to \mathsf{S}(\mathsf{z}))$

   $\mathsf{apply}(\mathsf{apply}(add, \mathsf{S}(0)), \mathsf{S}(0)) =$
   $\mathsf{apply}(\lambda y \to \mathsf{rec}(\mathsf{S}(0), y, \lambda y \to \lambda z \to \mathsf{S}(\mathsf{z})), \mathsf{S}(0)) =$
   $\mathsf{apply}(\lambda y \to \mathsf{S}(\mathsf{rec}(0, y, \lambda y \to \lambda z \to \mathsf{S}(\mathsf{z}))), \mathsf{S}(0)) =$
   $\mathsf{apply}(\lambda y \to \mathsf{S}(y), \mathsf{S}(0)) =$
   $\mathsf{S}(\mathsf{S}(0))$

4. (a) $\lambda x \to x : A \to A$
   (b) $\lambda x \to \langle \#_2(x), \#_1(x) \rangle : A \wedge B \to B \wedge A$
   (c) $\lambda x \to \mathsf{when}(x, \lambda y \to \mathsf{inr}(y), \lambda y \to \mathsf{inl}(y)) : A \vee B \to B \vee A$
   (d) $\lambda x \to \lambda y \to \mathsf{apply}(\#_2(x), \mathsf{apply}(\#_1(x), y)) : (A \to B) \wedge (B \to C) \to (A \to C)$
   (e) $\lambda x \to \lambda y \to \mathsf{apply}(\mathsf{apply}(x, y), y) : (A \to \neg A) \to \neg A$
   (f) $\lambda x \to \lambda y \to \mathsf{apply}(y, x) : A \to \neg\neg A$

5. En funktion som för varje certifikat för $\neg\neg A$ ger ett certifikat för A. Ett certifikat för $\neg\neg A$ är en funktion som till varje certifikat för $\neg A$ ger ett certifikat för $\bot$. Så det finns ingen möjlighet att utifrån ett certifikat för $\neg\neg A$ konstruera ett certifikat för A.

6. Ett certifikat för $\bot \to A$ är en total funktion som för varje (alla) certifikat för $\bot$ ger ett certifikat för A. Eftersom det inte finns några certifikat för $\bot$ är villkoret 'för varje' trivialt uppfyllt. Mängdteoretiskt är $\emptyset$ ett certifikat för $\bot \to A$.
