MATH230: Tutorial Eight

Curry-Howard Correspondence

Key ideas

- Write context dependent typing derivations.
- Understand the connection between natural deductions and typing derivations.
- Write proof-terms witnessing theorems of minimal logic.

Relevant lectures: Typed Lambda Calculus Slides

Relevant reading: Type Theory and Functional Programming, Simon Thompson

Hand in exercises: 1a, 1d, 1e, 1k, 3

Due Friday @ 5pm to the submission box on Learn.

Discussion Questions

- Write a program of the specified type in the given context:

$$p : A \times (B \times C) \ \vdash \ (A \times B) \times C$$

- For a fixed typed A, prove that the type $(A \to A) \to A$ is uninhabited i.e. there is no term t of simple type theory that has this type.

Tutorial Exercises

1. For each $\Sigma \vdash \alpha$ provide a term of type α from the given Σ context.

   (a) $f : A \to (B \to C) \vdash B \to (A \to C)$ [Handin exercise]

   (b) $t : A \times B \vdash B \times A$

   (c) $t : A + B \vdash B + A$

   (d) $f : (A \times B) \to C \vdash A \to (B \to C)$ [Handin exercise]

   (e) $f : A \to (B \to C) \vdash (A \times B) \to C$ [Handin exercise]

   (f) $f : A \to B \vdash A \to (B + C)$

   (g) $f : A \to B,\ g : B \to C \vdash A \to C$

   (h) $t : A + B,\ f : A \to C,\ g : B \to D \vdash C + D$

   (i) $f : A \to B \vdash (C \to A) \to (C \to B)$

   (j) $t : (A \to B) \times (A \to C) \vdash A \to (B \times C)$

   (k) $t : A \times (B + C) \vdash (A \times B) + (A \times C)$ [Handin exercise]

   (l) $t : (A \times B) + (A \times C) \vdash A \times (B + C)$

   (m) $t : A + (B \times C) \vdash (A + B) \times (A + C)$

   (n) $t : (A + B) \times (A + C) \vdash A + (B \times C)$

   Extras: For these extra problems consider $\perp$ to be type with no constructor or destructors. Furthermore, consider $\neg P$ to be shorthand for the function type: $\neg P := P \to \perp$.

   (a) $f : \neg A \vdash (C \to A) \to \neg C$

   (b) $t : A \times \neg B \vdash \neg(A \to B)$

   (c) $f : A \to C,\ g : B \to D,\ t : \neg C + \neg D \vdash \neg A + \neg B$

   (d) $t : A,\ f : \neg A \vdash \neg B$

   (e) $f : A \to B,\ g : A \to \neg B \vdash \neg A$

   (f) $f : A \to \neg B \vdash B \to \neg A$

   (g) $f : \neg(A \times B) \vdash A \to \neg B$

   (h) $t : A \vdash \neg\neg A$

   (i) $f : \neg\neg\neg A \vdash \neg A$

   (j) $t : \neg A + \neg B \vdash \neg(A \times B)$

   (k) $f : \neg A \times \neg B \vdash \neg(A + B)$

   (l) $f : \neg(A + B) \vdash \neg A \times \neg B$

   (m) $f : A \to \neg B \vdash \neg(A \times B)$

   (n) $\vdash \neg\neg(A + \neg A)$

2. Revisit Lab 1 and Lab 2. For each derivation in those labs, provide a proof-object witnessing a natural deduction of the sequent. You don't need to do any more derivations at this point!

3. This exercise shows you an example of a general observation first made by William Tait, relating the simplifications of proofs and the process of computation in the λ-calculus.

   Consider the following proof of the theorem

$$\vdash\ A \wedge B \to B$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{A \wedge B}^{\,1}}{B}\,\wedge E_R \qquad \dfrac{\overline{A \wedge B}^{\,1}}{A}\,\wedge_L}{B \wedge A}\,\wedge I}{B}\,\wedge E_L}{A \wedge B \to B}\,\to, 1$$

   (a) Determine the corresponding proof-object for this proof.

   (b) Why does the proof-object have a redex in it?

   (c) Perform the β-reduction on the proof object from (a).

   (d) What proof does the reduced proof-object correspond to?

4. Prove that the type $A + B \to A$ is uninhabited i.e. there is no term t of simple type theory that has this type. Your proof should be an informal reason for why no such program can exist. You might refer to the corresponding minimal logic sequent to help your justification.