MATH230: Tutorial Twelve

Curry-Howard Correspondence

Key ideas

- Write recursive processes in λ-calculus,
- Write programs in simple type theory,
- Interpret programs as proofs of propositions,
- Prove that certain types are uninhabited.

Relevant lectures: Lambda Calculus and Typed Lambda Calculus Slides
Relevant reading: Type Theory and Functional Programming, Simon Thompson

Hand in exercises: 1b, 1d, 4a, 4b, 4c, 6
Due before the exam to the lecturer.

Discussion Questions

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

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

- Determine some steps towards writing a program (λ-term) representing the unary function, INT-SQRT, that returns the greatest natural number whose square is less than or equal to the input.

Tutorial Exercises

1. Write recursive λ-expressions that represent the following functions of natural numbers. For each function determine an appropriate helper-function GO to put through the Y combinator.
   (a) SUM of two natural numbers
   (b) MULTiply two natural numbers
   (c) EXPONentiation of a base to an exponent
   (d) FACTorial of a natural number
   (e) INT-SQRT the smallest integer whose square is greater than input
   (f) Calculate the nth FIBonacci number (Challenge!)
2. Write a λ-expression that can be used to compute the smallest natural number that satisfies a given unary-predicate $P?(x)$ that is represented by some λ-expression.
3. (Challenge!) Represent the following processes in the λ-calculus to get an expression that can be used to test whether a natural number is prime. For simplicity, assume the input is greater than TWO.
   (a) REMAINDER calculate the remainder of a division.
   (b) DIVIDES? binary predicate does second divide first?
   (c) Implement bounded-search to satisfy a predicate.
   (d) PRIME? Unary-predicate to detect primality.

Curry-Howard Correspondence

4. Each problem below is of the form:

$$\Sigma \ \vdash ? : \mathsf{TYPE}$$

To answer the question you must provide a λ-term of the specified TYPE from the context Σ stated in the problem. To ensure the term is of the specified type you must use the typing rules for the construction and destruction of types.

(a) $f : (A \times B) \to C \vdash ? : A \to (B \to C)$

(b) $f : A \to (B \to C) \ \vdash ? : (A \times B) \to C$

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

(d) $p : A + B, \ f : A \to C, \ g : B \to D \vdash ? : C + D$

Compare each of the typing derivations to the minimal logic proofs from Tutorial Two. What do you notice?

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

(a) $p : \neg A + \neg B \vdash ? : \neg(A \times B)$

(b) $p : \neg(A + B) \vdash ? : \neg A \times \neg B$

(c) $p : \neg A \times \neg B \vdash ? : \neg(A + B)$

(d) $f : A \to C, \ g : B \to D, \ p : \neg C + \neg D \vdash ? : \neg A + \neg B$

(e) $a : A, \ f : \neg A \vdash ? : \neg B$

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

5. 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

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

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

(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?

6. 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.

$$\nvdash t : (A \to A) \to A$$