MATH230: λ-Calculus

Church Encodings

Key ideas

- Practice β-reduction,
- Encode logic in λ-calculus,
- Encode natural numbers in λ-calculus.

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

Hand in exercises: 1c, 2, 3a, 5
Due following Friday @ 5pm.

Discussion Questions

- Logical connectives are encoded as functions in the λ-calculus. They're intended to take in the Boolean λ-expressions TRUE and FALSE. However, without any extra structure to the λ-calculus, there is nothing stopping us writing:

  NOT NOT NOT

  Perform β-reduction on the above until the expression is in normal form. The strange answer you get suggests that we should not do this! To avoid this, extra *type* structure is added to the λ-calculus; this amounts to saying only certain λ-expressions can be applied to others. If the language were typed sensibly, we would get a type error when trying to evaluate NOT NOT NOT.

Tutorial Exercises

1. By substituting the explicit λ-expressions (as necessary) and performing β-reduction, show that the expressions below are β-equivalent to the Boolean values we should expect from the logical connectives involved.

   (a) NOT FALSE

   (b) OR TRUE FALSE

   (c) AND FALSE TRUE

   (d) IMPLIES FALSE TRUE

   Only expand those expressions necessary for each step.

2. Write down λ-expressions that represent the propositional binary connectives XOR, NAND, and NOR. Recall that these have the following truth tables.

P	Q	XOR(P, Q)
T	T	F
T	F	T
F	T	T
F	F	F

P	Q	NAND(P, Q)
T	T	F
T	F	T
F	T	T
F	F	T

P	Q	NOR(P, Q)
T	T	F
T	F	F
F	T	F
F	F	T

3. By substituting the explicit λ-expressions (as necessary) and performing β-reduction, determine the normal forms of the following λ-expressions.

   (a) SUCC ONE

   (b) SUM ONE ZERO

   (c) MULT TWO ZERO

   Only expand those expressions necessary for each step.

Compound Data with PAIR

4. We have defined the following λ-expression to construct pairs of λ-expressions:

$$\text{PAIR} = \lambda x.\ \lambda y.\ \lambda f.\ f\ x\ y$$

The third input is a built-in place ready to take a selector:

$$\text{FirST} = \lambda x.\ \lambda y.\ x \qquad \text{SecoND} = \lambda x.\ \lambda y.\ y$$

Reduce these to normal form

(a) PAIR a b FST

(b) PAIR a b SND

(c) PAIR (PAIR a b) (PAIR c d) SND

You may wish to make use of the app linked on learn to write your answers to the following problems.

5. Integers are solutions to equations $x + b = a$. We can represent natural number solutions using Church numerals. However, further abstractions are required to represent the negative solutions to such equations.

Example The equation $x + 1 = 0$ has solution $x = -1$. One way to represent this in the λ-calculus is to use the pair $(0, 1)$ which we interpret as $-1 = (0, 1)$.

More generally $k = (a, b) = a - b$ represents the solution to $x + b = a$.

Such representations are not unique! e.g. $0 = (0, 0) = (1, 1) = \ldots$

Write λ-expressions for arithmetic on integers as pairs of Church numerals.

INT-SUM to calculate the sum of two integers.

INT-MULT to calculate the product of two integers.

INT-NEG(ative) to calculate the negative of an integer.

6. Rational numbers are solutions to equations of the form $bx = a$. Use PAIR to represent rational numbers in the λ-calculus and write λ-expressions to compute rational number arithmetic.

RAT-SUM to calculate the sum of two rational numbers.

RAT-MULT to calculate the product of two rational numbers.

RAT-REC(iprocal) to calculate the reciprocal of an integer.

7. Imaginary numbers can be represented as PAIRs of rational numbers.

8. Real numbers are a more complicated issue.