# Lambda Calculus

*Tutorial*

Dr S. S. Chandra

---

The Lambda Calculus is an encoding scheme for computation based on just a single concept: the pure function. In much the same way that Turing 'built' an abstract encoding machine that represented the simplest computer, we will follow in the footsteps of Alonzo Church, who constructed an abstract encoding scheme that can represent computation itself, without any assumption of a physical machine.

In this tutorial, you will practice simplifying $\lambda$ expressions and understanding how the encoding process is used to create the machinery of $\lambda$-Calculus.

## Problem 1

Use $\beta$-reduction on the following $\lambda$ expressions into $\beta$-normal form:

a) $(\lambda x.\lambda y.x + y)(3,5)$
b) $(\lambda x.\lambda y.yxy)ab$
c) $(\lambda x.xx)(\lambda y.y)$
d) $(\lambda x.(\lambda y.xy))y$
e) $(\lambda x.(\lambda y.x))y$

## Problem 2

Explain Church encoding and how Church numerals are derived. Derive the Church numerals up to 4 and describe how they can be used with the successor combinator. What is the consequence of developing these concepts within Lambda Calculus?

## Problem 3

Describe how Boolean logic can be defined in Lambda Calculus and therefore the three main operations of AND, OR and NOT can be encoded into it making sure to explain all the relevant combinators, their arguments and function bodies.

## Problem 4

Define and discuss the significance of the Composition Combinator in Lambda Calculus. Describe how this combinator contributes to the Turing completeness of Lambda Calculus.

## Problem 5

In a paragraph and using a few equations, describe how one can compute the factorial function using recursion in Lambda Calculus. Make sure to mention all the relevant combinators and theorems required to achieve your result.