

Final Project Proposal

Lambda Calculus for Quantum Computing

Classical Lambda Calculus

The lambda calculus is a minimal programming language which is turning complete. It has three components namely “variable”, “function” and “application”. We can build numbers and Boolean logic just using these three components. The following is a program in scheme to verify if a given expression is a valid classical lambda calculus expression.

```
λexpr ::= y           ; Variable
      / (λ (x) λexpr) ; Function
      / (λexpr λexpr) ; Application

(define parse
  (λ (e)
    (match e
      [ ,y #:when (symbol? y) y]
      [ (λ ( ,x) ,body) #:when (symbol? x) `(λ ( ,x) ,(parse body))]
      [ ( ,rator ,rand) `( , (parse-match rator) ,(parse rand))]
      [else (error "Given expression not a valid lambda calculus expression")])))
```

Quantum Lambda Calculus

To extend the same concept of minimal programming language for Quantum computing the minimal set above is not sufficient as in Quantum computing we must take into consider the following things which are particular to Quantum world.

- Reversibility of the computation
- No cloning theory
- Superposition
- Interference
- ...

To design such a system for Quantum computing we must extend the components lambda calculus to the following [1]

```
λexpr ::= y           ; Variable
      / (λ (x) λexpr) ; Function
      / (λexpr λexpr) ; Application
      / c             ; Constant
      / !t            ; Nonlinear term
      / (λ! (x) λexpr) ; Nonlinear function

c ::= Constants:
    0 / 1 / H / S / R3 / cnot / X / Y / Z / ...
```

Project proposal

As part of the class project, I would like to do the following:

1. Design a parser for Quantum lambda calculus
2. Convert the classic quantum algorithms (like Simon's algorithm, Shor's Algorithm, Grover's search, Amplitude amplification) to Quantum lambda calculus
3. Explore the possibility of building an interpreter for Quantum lambda calculus

Bibliography

[1] A. v. Tonder. [Online]. Available: <https://arxiv.org/pdf/quant-ph/0307150.pdf>.