COMP2022: Formal Languages and Logic

2018, Semester 2, Week 2

Assignment Project Exam Help

Joseph Godbehere

https://powcoder.com

9th August, 2018

Add WeChat powcoder

THE UNIVERSITY OF SYDNEY

COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

Assignment Project Exam Help

WARNING

This material has been reproduced and communicated to you by or on behalf of the University of Sydney pursuant to part VB of the Copyright Act 1968 (**the Act**).

https://powcoder.com

The material in this communication may be subject to copyright under the Act. Any further copying or communication of this material by you may be subject of copyright protect under the Act.

Add WeChat powcoder

Do not remove this notice.

OUTLINE

Assignment Project Exam Help

- Revision - Lambda Calculus

https://powcoder.com

- Currying

- Encodings

Add WeChat powcoder

- Functional Programming: LISP

Operations

Assignment Project Exam Help

- Application
  - Notation: $A \cdot B$
  - Expression B is applied to expression A

https://powcoder.com

Add WeChat powcoder

Operations

Assignment Project Exam Help

- Application
  - Notation: $A \cdot B$
  - Expression B is applied to expression A

https://powcoder.com

- Abstraction
  - $\lambda x.M$
  - Variable x is abstracted in expression M

Add WeChat powcoder

Rewriting

Assignment Project Exam Help

- $M[x := N]$

https://powcoder.com

- Expression M, but all free occurrences of x are replaced with N

Add WeChat powcoder

- e.g.
  - $(xyz\lambda x.(zxz))[x := A] =$
  - $(xyz\lambda x.(zxz))[y := B] =$
  - $(xyz\lambda x.(zxz))[z := C] =$

Rewriting

Assignment Project Exam Help

- $M[x := N]$

https://powcoder.com

- Expression M, but all free occurrences of x are replaced with N

Add WeChat powcoder

- e.g.
  - $(xyz\lambda x.(zxz))[x := A] = (Ayz\lambda x.(zxz))$
  - $(xyz\lambda x.(zxz))[y := B] =$
  - $(xyz\lambda x.(zxz))[z := C] =$

Rewriting

Assignment Project Exam Help

- $M[x := N]$

https://powcoder.com

- Expression M, but all free occurrences of x are replaced with N

Add WeChat powcoder

- e.g.
  - $(xyz\lambda x.(zxz))[x := A] = (Ayz\lambda x.(zxz))$
  - $(xyz\lambda x.(zxz))[y := B] = (xBz\lambda x.(zxz))$
  - $(xyz\lambda x.(zxz))[z := C] =$

Rewriting

Assignment Project Exam Help

- $M[x := N]$

https://powcoder.com

- Expression M, but all free occurrences of x are replaced with N

Add WeChat powcoder

- e.g.
  - $(xyz\lambda x.(zxz))[x := A] = (Ayz\lambda x.(zxz))$
  - $(xyz\lambda x.(zxz))[y := B] = (xBz\lambda x.(zxz))$
  - $(xyz\lambda x.(zxz))[z := C] = (xyC\lambda x.(CxC))$

α-reduction

Assignment Project Exam Help

- Rename a λ to remove a name conflict

https://powcoder.com

- $\lambda x.M \equiv \lambda y.(M[x := y])$

Add WeChat powcoder

- y must be a *new* variable
  - You must not choose a symbol that is already in use

β-reduction

Assignment Project Exam Help

- Solve an abstraction

https://powcoder.com

- $(\lambda x.M) \cdot N = M[x := N]$

Add WeChat powcoder

- Note: the free occurrences of x in M is exactly the set of occurrences which bound to the $\lambda x.$ in $(\lambda x.M)$

Outline

Assignment Project Exam Help

- Revision - Lambda Calculus

https://powcoder.com

- **Currying**
- Encodings

Add WeChat powcoder

- Functional Programming

Two arguments

- Suppose we have a function $f(x, y)$ which requires two arguments.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Two arguments

Assignment Project Exam Help

- Suppose we have a function $f(x, y)$ which requires two arguments.

https://powcoder.com

- Let $F_x = \lambda y.f(x, y)$ and $F = \lambda x.F_x$

Add WeChat powcoder

Two arguments

Assignment Project Exam Help

- Suppose we have a function $f(x, y)$ which requires two arguments.

https://powcoder.com

- Let $F_x = \lambda y.f(x, y)$ and $F = \lambda x.F_x$

- F is a function which takes one input, and returns a function F_x, which will take the *next* input.

Add WeChat powcoder

Two arguments

Assignment Project Exam Help

- Suppose we have a function $f(x, y)$ which requires two arguments.

https://powcoder.com

- Let $F_x = \lambda y.f(x, y)$ and $F = \lambda x.F_x$

Add WeChat powcoder

- F is a function which takes one input, and returns a function F_x, which will take the *next* input.
- The output of the second function will be $f(x, y)$.

Example

Normal arithmetic: $f(x, y) = (x + y)/2$

Assignment Project Exam Help

Lambda calculus: $(\lambda x.(\lambda y.(x + y)/2))$

https://powcoder.com

$$\big((\lambda x.(\lambda y.(x + y)/2)) \cdot 5\big) \cdot 7$$

Add WeChat powcoder

Example

Normal arithmetic: $f(x, y) = (x + y)/2$

Assignment Project Exam Help

Lambda calculus: $(\lambda x.(\lambda y.(x + y)/2))$

https://powcoder.com

$$\big((\lambda x.(\lambda y.(x + y)/2)) \cdot 5\big) \cdot 7$$

$$= (\lambda y.(5 + y)/2) \cdot 7$$

Add WeChat powcoder

Example

Normal arithmetic: $f(x, y) = (x + y)/2$

Assignment Project Exam Help

Lambda calculus: $(\lambda x.(\lambda y.(x + y)/2))$

https://powcoder.com

$$\big((\lambda x.(\lambda y.(x+y)/2)) \cdot 5\big) \cdot 7$$
$$= (\lambda y.(5+y)/2) \cdot 7$$
$$= (5+7)/2$$

Add WeChat powcoder

Example

Normal arithmetic: $f(x, y) = (x + y)/2$

Assignment Project Exam Help

Lambda calculus: $(\lambda x.(\lambda y.(x + y)/2))$

https://powcoder.com

$$
\begin{aligned}
& \big((\lambda x.(\lambda y.(x + y)/2)) \cdot 5\big) \cdot 7 \\
&= (\lambda y.(5 + y)/2) \cdot 7 \\
&= (5 + 7)/2 \\
&= 6
\end{aligned}
$$

Add WeChat powcoder

Currying

Assignment Project Exam Help

- A n-ary parameter function can be represented in the lambda calculus through *Currying*

https://powcoder.com

Add WeChat powcoder

Currying

Assignment Project Exam Help

- A n-ary parameter function can be represented in the lambda calculus through *Currying*

https://powcoder.com

- An n argument function returns an $(n-1)$ argument function, which returns an $(n-2)$ argument function, ...

Add WeChat powcoder

Currying

Assignment Project Exam Help

- A n-ary parameter function can be represented in the lambda calculus through *Currying*

https://powcoder.com

- An n argument function returns an $(n-1)$ argument function, which returns an $(n-2)$ argument function, ...

Add WeChat powcoder

- e.g. $(\lambda x.(\lambda y.(\lambda z.f(x,y,z)))) \cdot 1 = (\lambda y.(\lambda z.f(1,y,z)))$

Evaluation

Recall the example from earlier:

Assignment Project Exam Help

$$
\begin{aligned}
&\big((\lambda x.(\lambda y.(x + y)/2)) \cdot 5\big) \cdot 7 \\
&= (\lambda y.(5 + y)/2) \cdot 7 \\
&= (5 + 7)/2 \\
&= 6
\end{aligned}
$$

https://powcoder.com

The function is *partially evaluated* at each step

Add WeChat powcoder

Evaluation

Recall the example from earlier:

$$
\begin{aligned}
&\big((\lambda x.(\lambda y.(x+y)/2)) \cdot 5\big) \cdot 7 \\
&= (\lambda y.(5+y)/2) \cdot 7 \\
&= (5+7)/2 \\
&= 6
\end{aligned}
$$

Assignment Project Exam Help
https://powcoder.com

The function is *partially evaluated* at each step

- The first function returns $(\lambda y.(5+y)/2)$

Add WeChat powcoder

Evaluation

Assignment Project Exam Help

Recall the example from earlier:

$$
\begin{aligned}
&\big((\lambda x.(\lambda y.(x+y)/2)) \cdot 5\big) \cdot 7 \\
&= (\lambda y.(5+y)/2) \cdot 7 \\
&= (5+7)/2 \\
&= 6
\end{aligned}
$$

https://powcoder.com

The function is *partially evaluated* at each step

Add WeChat powcoder

- The first function returns $(\lambda y.(5+y)/2)$
- 7 is then applied to the new function

Evaluation

Assignment Project Exam Help

Recall the example from earlier:

$$
\begin{aligned}
&\big((\lambda x.(\lambda y.(x+y)/2)) \cdot 5\big) \cdot 7 \\
&= (\lambda y.(5+y)/2) \cdot 7 \\
&= (5+7)/2 \\
&= 6
\end{aligned}
$$

https://powcoder.com

The function is *partially evaluated* at each step

Add WeChat powcoder

- The first function returns $(\lambda y.(5+y)/2)$
- 7 is then applied to the new function
- $(5+7)/2$ is evaluated and returned

Notation

- Too many parentheses. Let's make it simpler:

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Notation

Assignment Project Exam Help

- Too many parentheses. Let's make it simpler:
- We can write $(A \cdot B)$ as $A\ B$

https://powcoder.com

Add WeChat powcoder

Notation

Assignment Project Exam Help

- Too many parentheses. Let's make it simpler:

- We can write $(A \cdot B)$ as $A\ B$

https://powcoder.com

- For function application we use association to the *left*:

$$ABCDEF \equiv (((((AB)C)D)E)F)$$

Add WeChat powcoder

Notation

Assignment Project Exam Help

- Too many parentheses. Let's make it simpler:

- We can write $(A \cdot B)$ as $A\ B$

https://powcoder.com

- For function application we use association to the *left*:

$$ABCDEF \equiv (((((AB)C)D)E)F)$$

Add WeChat powcoder

- i.e. the leftmost application happens *first*

Notation

Assignment Project Exam Help

- For function abstraction we use association to the *right*

$$\lambda x_1 x_2 x_3 \ldots x_k . M$$

https://powcoder.com

Add WeChat powcoder

Notation

Assignment Project Exam Help

- For function abstraction we use association to the *right*

$$
\begin{aligned}
&\lambda x_1 x_2 x_3 \ldots x_k . M \\
&= \lambda x_1 . \lambda x_2 . \lambda x_3 \ldots \lambda x_k . M
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Notation

Assignment Project Exam Help

- For function abstraction we use association to the *right*

$$\begin{aligned} &\lambda x_1 x_2 x_3 \ldots x_k . M \\ = &\lambda x_1 . \lambda x_2 . \lambda x_3 . \ldots \lambda x_k . M \\ = &(\lambda x_1 . (\lambda x_2 . (\lambda x_3 . (\ldots(\lambda x_k . M)\ldots)))) \end{aligned}$$

https://powcoder.com

Add WeChat powcoder

Notation

Assignment Project Exam Help

- For function abstraction we use association to the *right*

$$
\begin{aligned}
&\lambda x_1 x_2 x_3 \ldots x_k . M \\
&= \lambda x_1 . \lambda x_2 . \lambda x_3 . \ldots \lambda x_k . M \\
&= (\lambda x_1 .(\lambda x_2 .(\lambda x_3 .(\ldots(\lambda x_k . M)\ldots))))
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

- This means the leftmost λ will match with the first input applied to the function

Notation

Assignment Project Exam Help

- Abstraction is right associative
- Application is left associative
- The abstractions and applications match up nicely:

https://powcoder.com

$$(\lambda xyz.((z - x) \times y))\ 4\ 2\ 3$$

Add WeChat powcoder

Notation

Assignment Project Exam Help

- Abstraction is right associative
- Application is left associative
- The abstractions and applications match up nicely:

https://powcoder.com

$$
\begin{aligned}
& (\lambda xyz.((z - x) \times y))\ 4\ 2\ 3 \\
= {} & (\lambda yz.((z - 4) \times y))\ 2\ 3
\end{aligned}
$$

Add WeChat powcoder

Notation

Assignment Project Exam Help

- Abstraction is right associative
- Application is left associative
- The abstractions and applications match up nicely:

https://powcoder.com

$$
\begin{aligned}
& \big(\lambda xyz.((z - x) \times y)\big)\ 4\ 2\ 3 \\
= & \big(\lambda yz.((z - 4) \times y)\big)\ 2\ 3 \\
= & \big(\lambda z.((z - 4) \times 2)\big)\ 3
\end{aligned}
$$

Add WeChat powcoder

Notation

Assignment Project Exam Help

- Abstraction is right associative
- Application is left associative
- The abstractions and applications match up nicely:

https://powcoder.com

$$
\begin{aligned}
& \big(\lambda xyz.((z-x)\times y)\big)\ 4\ 2\ 3 \\
= & \big(\lambda yz.((z-4)\times y)\big)\ 2\ 3 \\
= & \big(\lambda z.((z-4)\times 2)\big)\ 3 \\
= & (3-4)\times 2 \\
= & -2
\end{aligned}
$$

Add WeChat powcoder

Notation

- Abstraction is right associative
- Application is left associative
- If we wrote it out in full…

$$\big(\lambda xyz.((z - x) \times y)\big)\ 4\ 2\ 3$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Notation

- Abstraction is right associative
- Application is left associative
- If we wrote it out in full...

$$\left(\lambda xyz.((z - x) \times y)\right)\ 4\ 2\ 3$$

$$= \left(\lambda x.(\lambda y.(\lambda z.((z - x) \times y)))\right)\ 4\ 2\ 3$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Notation

- Abstraction is right associative
- Application is left associative
- If we wrote it out in full...

$$\left(\lambda xyz.((z-x)\times y)\right)\ 4\ 2\ 3$$

$$=\left(\lambda x.\left(\lambda y.\left(\lambda z.((z-x)\times y)\right)\right)\right)\ 4\ 2\ 3$$

$$=\left(\left(\left(\lambda x.\left(\lambda y.\left(\lambda z.((z-x)\times y)\right)\right)\right)\cdot 4\right)\cdot 2\right)\cdot 3$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Notation

- Abstraction is right associative
- Application is left associative
- If we wrote it out in full...

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$
\begin{aligned}
& \big(\lambda xyz.((z-x)\times y)\big)\ 4\ 2\ 3 \\
=\ & \Big(\lambda x.(\lambda y.(\lambda z.((z-x)\times y)))\Big)\ 4\ 2\ 3 \\
=\ & \left(\left(\left(\lambda x.(\lambda y.(\lambda z.((z-x)\times y)))\right)\cdot 4\right)\cdot 2\right)\cdot 3 \\
=\ & \Big(\big(\lambda y.(\lambda z.((z-4)\times y))\big)\cdot 2\Big)\cdot 3 \\
=\ & \big(\lambda z.((z-4)\times 2)\big)\cdot 3 \\
=\ & (3-4)\times 2 \\
=\ & -2
\end{aligned}
$$

Notation

Assignment Project Exam Help

- Question:
  1. Is $\lambda x.xy = (\lambda x.(xy))$, or
  2. is $\lambda x.xy = (\lambda x.x)y$?

https://powcoder.com

Add WeChat powcoder

Notation

Assignment Project Exam Help

- Question:
  1. Is $\lambda x.xy = (\lambda x.(xy))$, or
  2. is $\lambda x.xy = (\lambda x.x)y$?

https://powcoder.com

- Answer: (1), it's $(\lambda x.(xy))$

Add WeChat powcoder

Notation

Assignment Project Exam Help

- Question:
  1. Is $\lambda x.xy = (\lambda x.(xy))$, or
  2. is $\lambda x.xy = (\lambda x.x)y$?

https://powcoder.com

- Answer: (1), it's $(\lambda x.(xy))$

Add WeChat powcoder

- Use parentheses to limit the scope of the λ if needed

Currying

- Suppose we wanted to abstract a function with k arguments:

$$(\lambda x_1 x_2 x_3 \ldots x_k . N)$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Currying

- Suppose we wanted to abstract a function with k arguments:

Assignment Project Exam Help

$$(\lambda x_1 x_2 x_3 \ldots x_k . N)$$

- If we apply k arguments, $v_1 \ldots v_k$, we get this:

https://powcoder.com

$$(\lambda x_1 x_2 x_3 \ldots x_k . N)\, v_1 v_2 v_3 \ldots v_k$$

Add WeChat powcoder

Currying

- Suppose we wanted to abstract a function with k arguments:

Assignment Project Exam Help

$$(\lambda x_1 x_2 x_3 \ldots x_k . N)$$

- If we apply k arguments, $v_1 \ldots v_k$, we get this:

https://powcoder.com

$$(\lambda x_1 x_2 x_3 \ldots x_k . N)\, v_1 v_2 v_3 \ldots v_k$$

Add WeChat powcoder

- Each β-reduction partially evaluates the function:
  - v_1 replaces x_1. The resulting function takes $k - 1$ arguments:

$$(\lambda x_2 x_3 \ldots x_k . N[x_1 : v_1])\, v_2 v_3 \ldots v_k$$

  - ... then v_2 would replace x_2, etc.

OUTLINE

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

- Revision - Lambda Calculus
- Currying
- **Encodings**
- Functional Programming: LISP

Assignment Project Exam Help

But... How do we actually *do* anything?

https://powcoder.com

Add WeChat powcoder

Untyped Lambda Calculus

Assignment Project Exam Help

- Lambda calculus does not have primitives
  - No numbers
  - No arithmetic operators
  - No aggregated data types (classes etc.)
  - No control flow (only recursion!)

https://powcoder.com

Add WeChat powcoder

Untyped Lambda Calculus

Assignment Project Exam Help

- Lambda calculus does not have primitives
  - No numbers
  - No arithmetic operators
  - No aggregated data types (classes etc.)
  - No control flow (only recursion!)

https://powcoder.com

- However, I'm claiming that it is computationally equivalent to a Turing Machine!

Add WeChat powcoder

Untyped Lambda Calculus

- Lambda calculus does not have primitives
  - No numbers
  - No arithmetic operators
  - No aggregated data types (classes etc.)
  - No control flow (only recursion!)
- However, I'm claiming that it is computationally equivalent to a Turing Machine!
- So, how can we represent data types?

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Untyped Lambda Calculus

- Lambda calculus does not have primitives
  - No numbers
  - No arithmetic operators
  - No aggregated data types (classes etc.)
  - No control flow (only recursion!)
- However, I'm claiming that it is computationally equivalent to a Turing Machine!
- So, how can we represent data types?
  - They must be expressed as functions, known as *encodings*

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

- Boolean constants:
  - TRUE $:= \lambda xy.x$
  - FALSE $:= \lambda xy.y$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

- Boolean constants:
  - TRUE $:= \lambda xy.x$
  - FALSE $:= \lambda xy.y$

https://powcoder.com

- Now we can do conditional logic:
  - IFELSE $:= \lambda fxy.fxy$ has semantics similar to:
    - if <cond> then <x> else <y>
    - If <cond> is true, return result of <x>, otherwise <y>

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$IFELSE\ TRUE\ A\ B$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$IFELSE\ TRUE\ A\ B$$

$$= (\lambda fxy.fxy)\ (\lambda xy.x)\ A\ B \qquad \text{(macro substitution)}$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{IFELSE}\ \mathit{TRUE}\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.x)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.x)\ A\ B && \text{(α-reduction)}
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

$$
\begin{aligned}
& \mathit{IFELSE}\ \mathit{TRUE}\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.x)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.x)\ A\ B && \text{(α-reduction)} \\
&= (\lambda fab.fab)\ (\lambda xy.x)\ A\ B && \text{(α-reduction)}
\end{aligned}
$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& IFELSE\ TRUE\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.x)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.x)\ A\ B && \text{(α-reduction)} \\
&= (\lambda fab.fab)\ (\lambda xy.x)\ A\ B && \text{(α-reduction)} \\
&= (\lambda ab.(\lambda xy.x)ab)\ A\ B && \text{(β-reduction)}
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$\begin{aligned}
& \mathit{IFELSE}\ \mathit{TRUE}\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.x)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.x)\ A\ B && (\alpha\text{-reduction}) \\
&= (\lambda fab.fab)\ (\lambda xy.x)\ A\ B && (\alpha\text{-reduction}) \\
&= (\lambda ab.(\lambda xy.x)ab)\ A\ B && (\beta\text{-reduction}) \\
&= (\lambda b.(\lambda xy.x)Ab)\ B && (\beta\text{-reduction})
\end{aligned}$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{IFELSE}\ \mathit{TRUE}\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.x)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.x)\ A\ B && \text{(α-reduction)} \\
&= (\lambda fab.fab)\ (\lambda xy.x)\ A\ B && \text{(α-reduction)} \\
&= (\lambda ab.(\lambda xy.x)ab)\ A\ B && \text{(β-reduction)} \\
&= (\lambda b.(\lambda xy.x)Ab)\ B && \text{(β-reduction)} \\
&= (\lambda xy.x)AB && \text{(β-reduction)}
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& IFELSE\ TRUE\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.x)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.x)\ A\ B && (\alpha\text{-reduction}) \\
&= (\lambda fab.fab)\ (\lambda xy.x)\ A\ B && (\alpha\text{-reduction}) \\
&= (\lambda ab.(\lambda xy.x)ab)\ A\ B && (\beta\text{-reduction}) \\
&= (\lambda b.(\lambda xy.x)Ab)\ B && (\beta\text{-reduction}) \\
&= (\lambda xy.x)AB) && (\beta\text{-reduction}) \\
&= (\lambda y.A)B) && (\beta\text{-reduction})
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& IFELSE\ TRUE\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.x)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.x)\ A\ B && (\alpha\text{-reduction}) \\
&= (\lambda fab.fab)\ (\lambda xy.x)\ A\ B && (\alpha\text{-reduction}) \\
&= (\lambda ab.(\lambda xy.x)ab)\ A\ B && (\beta\text{-reduction}) \\
&= (\lambda b.(\lambda xy.x)Ab)\ B && (\beta\text{-reduction}) \\
&= (\lambda xy.x)\ A B) && (\beta\text{-reduction}) \\
&= (\lambda y.A)B) && (\beta\text{-reduction}) \\
&= A && (\beta\text{-reduction})
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

$$IFELSE\ FALSE\ A\ B$$

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$IFELSE\ FALSE\ A\ B$$

$$= (\lambda fxy.fxy)\ (\lambda xy.y)\ A\ B \qquad \text{(macro substitution)}$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{IFELSE}\ \mathit{FALSE}\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.y)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.y)\ A\ B && \text{(α-reduction)}
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{IFELSE}\ \mathit{FALSE}\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.y)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.y)\ A\ B && (\alpha\text{-reduction}) \\
&= (\lambda fab.fab)\ (\lambda xy.y)\ A\ B && (\alpha\text{-reduction})
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{IFELSE}\ \mathit{FALSE}\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.y)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.y)\ A\ B && \text{(α-reduction)} \\
&= (\lambda fab.fab)\ (\lambda xy.y)\ A\ B && \text{(α-reduction)} \\
&= (\lambda ab.(\lambda xy.y)ab)\ A\ B && \text{(β-reduction)}
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{IFELSE}\ \mathit{FALSE}\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.y)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.y)\ A\ B && \text{(α-reduction)} \\
&= (\lambda fab.fab)\ (\lambda xy.y)\ A\ B && \text{(α-reduction)} \\
&= (\lambda ab.(\lambda xy.y)ab)\ A\ B && \text{(β-reduction)} \\
&= (\lambda b.(\lambda xy.y)Ab)\ B && \text{(β-reduction)}
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{IFELSE}\ \mathit{FALSE}\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.y)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.y)\ A\ B && (\alpha\text{-reduction}) \\
&= (\lambda fab.fab)\ (\lambda xy.y)\ A\ B && (\alpha\text{-reduction}) \\
&= (\lambda ab.(\lambda xy.y)ab)\ A\ B && (\beta\text{-reduction}) \\
&= (\lambda b.(\lambda xy.y)Ab)\ B && (\beta\text{-reduction}) \\
&= (\lambda xy.y)AB) && (\beta\text{-reduction})
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& IFELSE\ FALSE\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.y)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.y)\ A\ B && (\alpha\text{-reduction}) \\
&= (\lambda fab.fab)\ (\lambda xy.y)\ A\ B && (\alpha\text{-reduction}) \\
&= (\lambda ab.(\lambda xy.y)ab)\ A\ B && (\beta\text{-reduction}) \\
&= (\lambda b.(\lambda xy.y)Ab)\ B && (\beta\text{-reduction}) \\
&= (\lambda xy.y)AB) && (\beta\text{-reduction}) \\
&= (\lambda y.y)B) && (\beta\text{-reduction})
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{IFELSE}\ \mathit{FALSE}\ A\ B \\
&= (\lambda fxy.fxy)\ (\lambda xy.y)\ A\ B && \text{(macro substitution)} \\
&= (\lambda fay.fay)\ (\lambda xy.y)\ A\ B && (\alpha\text{-reduction}) \\
&= (\lambda fab.fab)\ (\lambda xy.y)\ A\ B && (\alpha\text{-reduction}) \\
&= (\lambda ab.(\lambda xy.y)ab)\ A\ B && (\beta\text{-reduction}) \\
&= (\lambda b.(\lambda xy.y)Ab)\ B && (\beta\text{-reduction}) \\
&= (\lambda xy.y)AB) && (\beta\text{-reduction}) \\
&= (\lambda y.y)B) && (\beta\text{-reduction}) \\
&= B && (\beta\text{-reduction})
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

- Boolean constants:
  - TRUE := $\lambda xy.x$
  - FALSE := $\lambda xy.y$

https://powcoder.com

- IFELSE := $\lambda fxy.fxy$

Add WeChat powcoder

- Boolean operators:
  - NOT := $\lambda fxy.fyx$
  - OR := $\lambda xy.xxy$
  - AND := $\lambda xy.xyx$

Encodings: NOT

Assignment Project Exam Help

- NOT $:= \lambda fxy.fyx$

https://powcoder.com

- NOT is a function which takes 3 arguments
  - Suppose f was a function which takes 2 arguments
  - x, y would be those arguments

Add WeChat powcoder

Encodings: NOT

Assignment Project Exam Help

- NOT $:= \lambda fxy.fyx$

https://powcoder.com

- NOT is a function which takes 3 arguments
  - Suppose f was a function which takes 2 arguments
  - x, y would be those arguments

Add WeChat powcoder

- i.e. NOT outputs f, except its arguments have swapped around!

Encodings: truth

Assignment Project Exam Help

NOT TRUE

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{NOT}\ \mathit{TRUE} \\
&= (\lambda fxy.fyx)(\lambda xy.x) && \text{(macro substitution)}
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{NOT}\ \mathit{TRUE} \\
&= (\lambda fxy.fyx)(\lambda xy.x) && \text{(macro substitution)} \\
&= (\lambda fxy.fyx)(\lambda ay.a) && \text{(α-reduction)}
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& NOT\ TRUE \\
&= (\lambda fxy.fyx)(\lambda xy.x) && \text{(macro substitution)} \\
&= (\lambda fxy.fyx)(\lambda ay.a) && (\alpha\text{-reduction}) \\
&= (\lambda fxy.fyx)(\lambda ab.a) && (\alpha\text{-reduction})
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& NOT\ TRUE \\
&= (\lambda fxy.fyx)(\lambda xy.x) && \text{(macro substitution)} \\
&= (\lambda fxy.fyx)(\lambda ay.a) && (\alpha\text{-reduction}) \\
&= (\lambda fxy.fyx)(\lambda ab.a) && (\alpha\text{-reduction}) \\
&= \lambda xy.(\lambda ab.a)yx && (\beta\text{-reduction})
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{NOT}\ \mathit{TRUE} \\
&= (\lambda fxy.fyx)(\lambda xy.x) && \text{(macro substitution)} \\
&= (\lambda fxy.fyx)(\lambda ay.a) && \text{(α-reduction)} \\
&= (\lambda fxy.fyx)(\lambda ab.a) && \text{(α-reduction)} \\
&= \lambda xy.(\lambda ab.a)yx && \text{(β-reduction)} \\
&= \lambda xy.(\lambda b.y)x && \text{(β-reduction)}
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& NOT\ TRUE \\
&= (\lambda fxy.fyx)(\lambda xy.x) && \text{(macro substitution)} \\
&= (\lambda fxy.fyx)(\lambda ay.a) && (\alpha\text{-reduction}) \\
&= (\lambda fxy.fyx)(\lambda ab.a) && (\alpha\text{-reduction}) \\
&= \lambda xy.(\lambda ab.a)yx && (\beta\text{-reduction}) \\
&= \lambda xy.(\lambda b.y)x && (\beta\text{-reduction}) \\
&= \lambda xy.y && (\beta\text{-reduction})
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: truth

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{NOT}\ \mathit{TRUE} & \\
&= (\lambda fxy.fyx)(\lambda xy.x) & \text{(macro substitution)} \\
&= (\lambda fxy.fyx)(\lambda ay.a) & \text{(α-reduction)} \\
&= (\lambda fxy.fyx)(\lambda ab.a) & \text{(α-reduction)} \\
&= \lambda xy.(\lambda ab.a)yx & \text{(β-reduction)} \\
&= \lambda xy.(\lambda b.y)x & \text{(β-reduction)} \\
&= \lambda xy.y & \text{(β-reduction)} \\
&= \mathit{FALSE} & \text{(macro substitution)}
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Encodings: numbers

- The natural numbers can be thought of as a sequence, starting from 0, and successively increasing by one.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Encodings: numbers

- The natural numbers can be thought of as a sequence, starting from 0, and successively increasing by one.
- More formally, we can define them inductively:

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Encodings: numbers

Assignment Project Exam Help

- The natural numbers can be thought of as a sequence, starting from 0, and successively increasing by one.

https://powcoder.com

- More formally, we can define them inductively:
  - Basic clause: 0 is a number and is in the set

Add WeChat powcoder

Encodings: numbers

Assignment Project Exam Help

- The natural numbers can be thought of as a sequence, starting from 0, and successively increasing by one.

https://powcoder.com

- More formally, we can define them inductively:
  - Basic clause: 0 is a number and is in the set
  - Inductive clause: for any element x in the natural numbers, $x + 1$ is an element of the natural numbers

Add WeChat powcoder

Encodings: numbers

Assignment Project Exam Help

- The natural numbers can be thought of as a sequence, starting from 0, and successively increasing by one.

- More formally, we can define them inductively:

https://powcoder.com

  - Basic clause: 0 is a number and is in the set

Add WeChat powcoder

  - Inductive clause: for any element x in the natural numbers, $x + 1$ is an element of the natural numbers

  - Extremal clause: nothing is in the set of natural numbers unless it is obtained by the inductive clause and basis clause

Church numerals

- Natural numbers in lambda calculus have two constructors:

Assignment Project Exam Help

  - ZERO := $\lambda xy.y$
    - This represents 0
    - It is the same formula we used to encode FALSE

https://powcoder.com

Add WeChat powcoder

Church numerals

Assignment Project Exam Help

- Natural numbers in lambda calculus have two constructors:

https://powcoder.com

  - ZERO $:= \lambda xy.y$
    - This represents 0
    - It is the same formula we used to encode FALSE
  - SUCCESSOR $:= \lambda xyz.y(xyz)$
    - Returns the next number in the sequence

Add WeChat powcoder

Church numerals

Assignment Project Exam Help

- Natural numbers in lambda calculus have two constructors:

https://powcoder.com

  - ZERO := $\lambda xy.y$
    - This represents 0
    - It is the same formula we used to encode FALSE

Add WeChat powcoder

  - SUCCESSOR := $\lambda xyz.y(xyz)$
    - Returns the next number in the sequence

- We're now ready to start constructing the natural numbers!

Church numerals

Assignment Project Exam Help

$$
\begin{aligned}
& ONE \\
&= SUCCESSOR\ ZERO
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Church numerals

Assignment Project Exam Help

$$
\begin{aligned}
& ONE \\
&= SUCCESSOR\ ZERO \\
&= (\lambda xyz.y(xyz))(\lambda xy.y) \quad \text{(macro)}
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Church numerals

Assignment Project Exam Help

$$
\begin{aligned}
& ONE \\
&= SUCCESSOR\ ZERO \\
&= (\lambda xyz.y(xyz))(\lambda xy.y) && \text{(macro)} \\
&= (\lambda xyz.y(xyz))(\lambda ab.b) && (\alpha)
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Church numerals

Assignment Project Exam Help

$$
\begin{aligned}
& ONE \\
&= SUCCESSOR\ ZERO \\
&= (\lambda xyz.y(xyz))(\lambda xy.y) && \text{(macro)} \\
&= (\lambda xyz.y(xyz))(\lambda ab.b) && (\alpha) \\
&= \lambda yz.y((\lambda ab.b)yz) && (\beta)
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Church numerals

Assignment Project Exam Help

$$
\begin{aligned}
& ONE \\
&= SUCCESSOR\ ZERO \\
&= (\lambda xyz.y(xyz))(\lambda xy.y) && \text{(macro)} \\
&= (\lambda xyz.y(xyz))(\lambda ab.b) && (\alpha) \\
&= \lambda yz.y((\lambda ab.b)yz) && (\beta) \\
&= \lambda yz.y((\lambda b.b)z) && (\beta)
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Church numerals

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$
\begin{aligned}
& ONE \\
&= SUCCESSOR\ ZERO \\
&= (\lambda xyz.y(xyz))(\lambda xy.y) && \text{(macro)} \\
&= (\lambda xyz.y(xyz))(\lambda ab.b) && (\alpha) \\
&= \lambda yz.y((\lambda ab.b)yz) && (\beta) \\
&= \lambda yz.y((\lambda b.b)z) && (\beta) \\
&= \lambda yz.yz && (\beta)
\end{aligned}
$$

Church numerals

Assignment Project Exam Help

$$
\begin{aligned}
& TWO \\
= \ & SUCCESSOR\ ONE \\
= \ & (\lambda xyz.y(xyz))(\lambda yz.yz) \quad \text{(macro)}
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Church numerals

Assignment Project Exam Help

$$
\begin{aligned}
& TWO \\
&= SUCCESSOR\ ONE \\
&= (\lambda xyz.y(xyz))(\lambda yz.yz) && \text{(macro)} \\
&= (\lambda xyz.y(xyz))(\lambda ab.ab) && (\alpha)
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Church numerals

Assignment Project Exam Help

$$
\begin{aligned}
& TWO \\
&= SUCCESSOR\ ONE \\
&= (\lambda xyz.y(xyz))(\lambda yz.yz) && \text{(macro)} \\
&= (\lambda xyz.y(xyz))(\lambda ab.ab) && (\alpha) \\
&= \lambda yz.y((\lambda ab.ab)yz) && (\beta)
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Church numerals

Assignment Project Exam Help

$$
\begin{aligned}
& TWO \\
&= SUCCESSOR\ ONE \\
&= (\lambda xyz.y(xyz))(\lambda yz.yz) && \text{(macro)} \\
&= (\lambda xyz.y(xyz))(\lambda ab.ab) && (\alpha) \\
&= \lambda yz.y((\lambda ab.ab)yz) && (\beta) \\
&= \lambda yz.y((\lambda b.yb)z) && (\beta)
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Church numerals

Assignment Project Exam Help

$$
\begin{aligned}
& TWO \\
&= SUCCESSOR\ ONE \\
&= (\lambda xyz.y(xyz))(\lambda yz.yz) && \text{(macro)} \\
&= (\lambda xyz.y(xyz))(\lambda ab.ab) && (\alpha) \\
&= \lambda yz.y((\lambda ab.ab)yz) && (\beta) \\
&= \lambda yz.y((\lambda b.yb)z) && (\beta) \\
&= \lambda yz.y(yz) && (\beta)
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Church numerals

Assignment Project Exam Help

$$\begin{aligned} &THREE \\ &= SUCCESSOR\ TWO \\ &= \ldots \\ &= \lambda yz.y(y(yz)) \end{aligned}$$

https://powcoder.com

Add WeChat powcoder

Church numerals

Assignment Project Exam Help

$$\begin{aligned}
&THREE \\
&= SUCCESSOR\ TWO \\
&= \ldots \\
&= \lambda yz.y(y(yz))
\end{aligned}$$

https://powcoder.com

Add WeChat powcoder

$$\begin{aligned}
&FOUR \\
&= SUCCESSOR\ THREE \\
&= \ldots \\
&= \lambda yz.y(y(y(yz)))
\end{aligned}$$

Arithmetic?

Assignment Project Exam Help

- We have numbers. Do they work?

https://powcoder.com

Add WeChat powcoder

Arithmetic?

Assignment Project Exam Help

- We have numbers. Do they work?

https://powcoder.com

- Arithmetic:
  - ADD $:= \lambda xypq.xp(ypq)$
  - MULT $:= \lambda xyz.x(yz)$
  - EXP $:= \lambda xy.yx$

Add WeChat powcoder

Addition example

$$ADD\ TWO\ THREE$$

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Addition example

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{ADD}\ \mathit{TWO}\ \mathit{THREE} \\
&= (\lambda xypq.xp(ypq))\ (\lambda yz.y(yz))\ (\lambda yz.y(y(yz)))
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Addition example

Assignment Project Exam Help

$$
\begin{aligned}
&\mathit{ADD}\ \mathit{TWO}\ \mathit{THREE} \\
&= (\lambda xypq.xp(ypq))\ (\lambda yz.y(yz))\ (\lambda yz.y(y(yz))) \\
&= (\lambda xypq.xp(ypq))\ (\lambda ab.a(ab))\ (\lambda cd.c(c(cd))) \qquad (\alpha)
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Addition example

$$
\begin{aligned}
& ADD\ TWO\ THREE \\
&= (\lambda xypq.xp(ypq))\ (\lambda yz.y(yz))\ (\lambda yz.y(y(yz))) \\
&= (\lambda xypq.xp(ypq))\ (\lambda ab.a(ab))\ (\lambda cd.c(c(cd))) && (\alpha) \\
&= (\lambda ypq.(\lambda ab.a(ab))p(ypq))\ (\lambda cd.c(c(cd))) && (\beta)
\end{aligned}
$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Addition example

$$
\begin{aligned}
& \mathit{ADD}\ \mathit{TWO}\ \mathit{THREE} \\
&= \big(\lambda xypq.xp(ypq)\big)\ (\lambda yz.y(yz))\ (\lambda yz.y(y(yz))) \\
&= \big(\lambda xypq.xp(ypq)\big)\ (\lambda ab.a(ab))\ (\lambda cd.c(c(cd))) && (\alpha) \\
&= \big(\lambda ypq.(\lambda ab.a(ab))p(ypq)\big)\ (\lambda cd.c(c(cd))) && (\beta) \\
&= \big(\lambda ypq.(\lambda b.p(pb))(ypq)\big)\ (\lambda cd.c(c(cd))) && (\beta)
\end{aligned}
$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Addition example

$$
\begin{aligned}
& ADD\ TWO\ THREE \\
&= \big(\lambda xypq.xp(ypq)\big)\ (\lambda yz.y(yz))\ (\lambda yz.y(y(yz))) \\
&= \big(\lambda xypq.xp(ypq)\big)\ (\lambda ab.a(ab))\ (\lambda cd.c(c(cd))) && (\alpha) \\
&= \big(\lambda ypq.(\lambda ab.a(ab))p(ypq)\big)\ (\lambda cd.c(c(cd))) && (\beta) \\
&= \big(\lambda ypq.(\lambda b.p(pb))(ypq)\big)\ (\lambda cd.c(c(cd))) && (\beta) \\
&= \big(\lambda ypq.p(p(ypq))\big)\ (\lambda cd.c(c(cd))) && (\beta)
\end{aligned}
$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Addition example

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{ADD}\ \mathit{TWO}\ \mathit{THREE} \\
&= \big(\lambda xypq.xp(ypq)\big)\ (\lambda yz.y(yz))\ (\lambda yz.y(y(yz))) \\
&= \big(\lambda xypq.xp(ypq)\big)\ (\lambda ab.a(ab))\ (\lambda cd.c(c(cd))) && (\alpha) \\
&= \big(\lambda ypq.(\lambda ab.a(ab))p(ypq)\big)\ (\lambda cd.c(c(cd))) && (\beta) \\
&= \big(\lambda ypq.(\lambda b.p(pb))(ypq)\big)\ (\lambda cd.c(c(cd))) && (\beta) \\
&= \big(\lambda ypq.p(p(ypq))\big)\ (\lambda cd.c(c(cd))) && (\beta) \\
&= \lambda pq.p\big(p((\lambda cd.c(c(cd)))pq)\big) && (\beta)
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Addition example

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$
\begin{aligned}
& \mathit{ADD}\ \mathit{TWO}\ \mathit{THREE} \\
&= \big(\lambda xypq.xp(ypq)\big)\ (\lambda yz.y(yz))\ (\lambda yz.y(y(yz))) \\
&= \big(\lambda xypq.xp(ypq)\big)\ (\lambda ab.a(ab))\ (\lambda cd.c(c(cd))) && (\alpha) \\
&= \big(\lambda ypq.(\lambda ab.a(ab))p(ypq)\big)\ (\lambda cd.c(c(cd))) && (\beta) \\
&= \big(\lambda ypq.(\lambda b.p(pb))(ypq)\big)\ (\lambda cd.c(c(cd))) && (\beta) \\
&= \big(\lambda ypq.p(p(ypq))\big)\ (\lambda cd.c(c(cd))) && (\beta) \\
&= \lambda pq.p(p((\lambda cd.c(c(cd)))pq)) && (\beta) \\
&= \lambda pq.p(p((\lambda d.p(p(pd)))q)) && (\beta)
\end{aligned}
$$

Addition example

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$
\begin{aligned}
& ADD\ TWO\ THREE \\
&= \big(\lambda xypq.xp(ypq)\big)\ (\lambda yz.y(yz))\ (\lambda yz.y(y(yz))) \\
&= \big(\lambda xypq.xp(ypq)\big)\ (\lambda ab.a(ab))\ (\lambda cd.c(c(cd))) && (\alpha) \\
&= \big(\lambda ypq.(\lambda ab.a(ab))p(ypq)\big)\ (\lambda cd.c(c(cd))) && (\beta) \\
&= \big(\lambda ypq.(\lambda b.p(pb))(ypq)\big)\ (\lambda cd.c(c(cd))) && (\beta) \\
&= \big(\lambda ypq.p(p(ypq))\big)\ (\lambda cd.c(c(cd))) && (\beta) \\
&= \lambda pq.p(p((\lambda cd.c(c(cd)))pq)) && (\beta) \\
&= \lambda pq.p(p((\lambda d.p(p(pd)))q)) && (\beta) \\
&= \lambda pq.p(p(p(p(pq)))) && (\beta)
\end{aligned}
$$

Addition example

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$
\begin{aligned}
& \mathit{ADD}\ \mathit{TWO}\ \mathit{THREE} & \\
&= \big(\lambda xypq.xp(ypq)\big)\ (\lambda yz.y(yz))\ (\lambda yz.y(y(yz))) & \\
&= \big(\lambda xypq.xp(ypq)\big)\ (\lambda ab.a(ab))\ (\lambda cd.c(c(cd))) & (\alpha) \\
&= \big(\lambda ypq.(\lambda ab.a(ab))p(ypq)\big)\ (\lambda cd.c(c(cd))) & (\beta) \\
&= \big(\lambda ypq.(\lambda b.p(pb))(ypq)\big)\ (\lambda cd.c(c(cd))) & (\beta) \\
&= \big(\lambda ypq.p(p(ypq))\big)\ (\lambda cd.c(c(cd))) & (\beta) \\
&= \lambda pq.p(p((\lambda cd.c(c(cd)))pq)) & (\beta) \\
&= \lambda pq.p(p((\lambda d.p(p(pd)))q)) & (\beta) \\
&= \lambda pq.p(p(p(p(pq)))) & (\beta) \\
&= \mathit{FIVE} &
\end{aligned}
$$

Multiplication example

Assignment Project Exam Help

$$MULT\ EIGHT\ THIRTEEN$$

https://powcoder.com

Add WeChat powcoder

Multiplication example

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{MULT}\ \mathit{EIGHT}\ \mathit{THIRTEEN} \\
= \ & (\lambda xyz.x(yz))(\lambda fx.f(f(f(f(f(f(f(fx)))))))) \\
& (\lambda fx.f(f(f(f(f(f(f(f(f(f(f(f(fx)))))))))))))
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Multiplication example

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{MULT}\ \mathit{EIGHT}\ \mathit{THIRTEEN} \\
= & (\lambda xyz.x(yz))(\lambda fx.f(f(f(f(f(f(f(fx))))))))) \\
& (\lambda fx.f(f(f(f(f(f(f(f(f(f(f(f(fx))))))))))))) \\
= & \ldots
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Just kidding

Multiplication example

$$MULT\ TWO\ THREE$$

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Multiplication example

$$MULT\ TWO\ THREE$$
$$= (\lambda xyz.x(yz))\ TWO\ THREE$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Multiplication example

$$\begin{aligned}
& MULT\ TWO\ THREE \\
&= (\lambda xyz.x(yz))\ TWO\ THREE \\
&= (\lambda yz.TWO\ (yz))\ THREE
\end{aligned}$$

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Multiplication example

$$\begin{aligned}
& MULT\ TWO\ THREE \\
&= (\lambda xyz.x(yz))\ TWO\ THREE \\
&= (\lambda yz.TWO\ (yz))\ THREE \\
&= \lambda z.TWO\ (THREE\ z)
\end{aligned}$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Multiplication example

$$
\begin{aligned}
& MULT\ TWO\ THREE \\
&= (\lambda xyz.x(yz))\ TWO\ THREE \\
&= (\lambda yz.TWO\ (yz))\ THREE \\
&= \lambda z.TWO\ (THREE\ z) \\
&= \lambda z.(\lambda fx.f(fx))(THREE\ z)
\end{aligned}
$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Multiplication example

$$
\begin{aligned}
& MULT\ TWO\ THREE \\
&= (\lambda xyz.x(yz))\ TWO\ THREE \\
&= (\lambda yz.TWO\ (yz))\ THREE \\
&= \lambda z.TWO\ (THREE\ z) \\
&= \lambda z.(\lambda fx.f(fx))(THREE\ z) \\
&= \lambda z.\Big(\lambda x.(THREE\ z)\big((THREE\ z)x\big)\Big)
\end{aligned}
$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Multiplication example

Assignment Project Exam Help

$$
\begin{aligned}
& MULT\ TWO\ THREE \\
&= (\lambda xyz.x(yz))\ TWO\ THREE \\
&= (\lambda yz.TWO\ (yz))\ THREE \\
&= \lambda z.TWO\ (THREE\ z) \\
&= \lambda z.(\lambda fx.f(fx))(THREE\ z) \\
&= \lambda z.\Big(\lambda x.(THREE\ z)\big((THREE\ z)x\big)\Big) \\
&= \lambda zx.(THREE\ z)\big((THREE\ z)x\big)
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Multiplication example

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{MULT}\ \mathit{TWO}\ \mathit{THREE} \\
&= (\lambda xyz.x(yz))\ \mathit{TWO}\ \mathit{THREE} \\
&= (\lambda yz.\mathit{TWO}\ (yz))\ \mathit{THREE} \\
&= \lambda z.\mathit{TWO}\ (\mathit{THREE}\ z) \\
&= \lambda z.(\lambda fx.f(fx))(\mathit{THREE}\ z) \\
&= \lambda z.\Big(\lambda x.(\mathit{THREE}\ z)\big((\mathit{THREE}\ z)x\big)\Big) \\
&= \lambda zx.(\mathit{THREE}\ z)\big((\mathit{THREE}\ z)x\big) \\
&= \lambda zx.\big((\lambda fx.f(f(fx)))\ z\big)\big(((\lambda fx.f(f(fx)))\ z)x\big)
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Multiplication example

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

$$
\begin{aligned}
& \mathit{MULT}\ \mathit{TWO}\ \mathit{THREE} \\
&= (\lambda xyz.x(yz))\ \mathit{TWO}\ \mathit{THREE} \\
&= (\lambda yz.\mathit{TWO}\ (yz))\ \mathit{THREE} \\
&= \lambda z.\mathit{TWO}\ (\mathit{THREE}\ z) \\
&= \lambda z.(\lambda fx.f(fx))(\mathit{THREE}\ z) \\
&= \lambda z.\Big(\lambda x.(\mathit{THREE}\ z)\big((\mathit{THREE}\ z)x\big)\Big) \\
&= \lambda zx.(\mathit{THREE}\ z)\big((\mathit{THREE}\ z)x\big) \\
&= \lambda zx.\big((\lambda fx.f(f(fx)))\ z\big)\big(((\lambda fx.f(f(fx)))\ z)x\big) \\
&= \lambda zx.\big(\lambda x.z(z(zx))\big)\big((\lambda x.z(z(zx)))x\big)
\end{aligned}
$$

Multiplication example

$$
\begin{aligned}
& \mathit{MULT}\ \mathit{TWO}\ \mathit{THREE} \\
&= (\lambda xyz.x(yz))\ \mathit{TWO}\ \mathit{THREE} \\
&= (\lambda yz.\mathit{TWO}\ (yz))\ \mathit{THREE} \\
&= \lambda z.\mathit{TWO}\ (\mathit{THREE}\ z) \\
&= \lambda z.(\lambda fx.f(fx))(\mathit{THREE}\ z) \\
&= \lambda z.\Big(\lambda x.(\mathit{THREE}\ z)\big((\mathit{THREE}\ z)x\big)\Big) \\
&= \lambda zx.(\mathit{THREE}\ z)\big((\mathit{THREE}\ z)x\big) \\
&= \lambda zx.\big((\lambda fx.f(f(fx)))\ z\big)\big(((\lambda fx.f(f(fx)))\ z)x\big) \\
&= \lambda zx.\big(\lambda x.z(z(zx))\big)\big((\lambda x.z(z(zx)))x\big) \\
&= \lambda zx.\big(\lambda x.z(z(zx))\big)\big(z(z(zx))\big)
\end{aligned}
$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Multiplication example

Assignment Project Exam Help

$$
\begin{aligned}
& \mathit{MULT}\ \mathit{TWO}\ \mathit{THREE} \\
={}& (\lambda xyz.x(yz))\ \mathit{TWO}\ \mathit{THREE} \\
={}& (\lambda yz.\mathit{TWO}\ (yz))\ \mathit{THREE} \\
={}& \lambda z.\mathit{TWO}\ (\mathit{THREE}\ z) \\
={}& \lambda z.(\lambda fx.f(fx))(\mathit{THREE}\ z) \\
={}& \lambda z.\Big(\lambda x.(\mathit{THREE}\ z)\big((\mathit{THREE}\ z)x\big)\Big) \\
={}& \lambda zx.(\mathit{THREE}\ z)\big((\mathit{THREE}\ z)x\big) \\
={}& \lambda zx.\big((\lambda fx.f(f(fx)))\ z\big)\big(((\lambda fx.f(f(fx)))\ z)x\big) \\
={}& \lambda zx.\big(\lambda x.z(z(zx))\big)\big((\lambda x.z(z(zx)))x\big) \\
={}& \lambda zx.\big(\lambda x.z(z(zx))\big)\big(z(z(zx))\big) \\
={}& \lambda zx.z(z(z(z(z(zx)))))
\end{aligned}
$$

https://powcoder.com

Add WeChat powcoder

Multiplication example

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$
\begin{aligned}
& \mathit{MULT}\ \mathit{TWO}\ \mathit{THREE} \\
&= (\lambda xyz.x(yz))\ \mathit{TWO}\ \mathit{THREE} \\
&= (\lambda yz.\mathit{TWO}\ (yz))\ \mathit{THREE} \\
&= \lambda z.\mathit{TWO}\ (\mathit{THREE}\ z) \\
&= \lambda z.(\lambda fx.f(fx))(\mathit{THREE}\ z) \\
&= \lambda z.\Big(\lambda x.(\mathit{THREE}\ z)\big((\mathit{THREE}\ z)x\big)\Big) \\
&= \lambda zx.(\mathit{THREE}\ z)\big((\mathit{THREE}\ z)x\big) \\
&= \lambda zx.\big((\lambda fx.f(f(fx)))\ z\big)\big(((\lambda fx.f(f(fx)))\ z)x\big) \\
&= \lambda zx.\big(\lambda x.z(z(zx))\big)\big((\lambda x.z(z(zx)))x\big) \\
&= \lambda zx.\big(\lambda x.z(z(zx))\big)\big(z(z(zx))\big) \\
&= \lambda zx.z(z(z(z(z(zx))))) \\
&= \mathit{SIX}
\end{aligned}
$$

Recursion

In imperative languages, we can easily write recursive code:

Assignment Project Exam Help

```
def factorial(x):
    if x == 1:
        return 1
    return x*factorial(x-1)
```

https://powcoder.com

... by referencing the method itself by name.

Add WeChat powcoder

So far, we haven't directly seen iteration or recursion in the lambda calculus.

Recursion

In the last tutorial you tried to reduce:

$$(\lambda x.xx)(\lambda x.xx)$$

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Recursion

In the last tutorial you tried to reduce:

Assignment Project Exam Help

$$(\lambda x.xx)(\lambda x.xx)$$

... and discovered that it looped forever.

https://powcoder.com

Add WeChat powcoder

Recursion

In the last tutorial you tried to reduce:

Assignment Project Exam Help

$$(\lambda x.xx)(\lambda x.xx)$$

... and discovered that it looped forever.

https://powcoder.com

This is related to a slightly more useful construct called the Y Combinator:

$$Y \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

Add WeChat powcoder

Recursion

In the last tutorial you tried to reduce:

Assignment Project Exam Help

$$(\lambda x.xx)(\lambda x.xx)$$

... and discovered that it looped forever.

https://powcoder.com

This is related to a slightly more useful construct called the Y Combinator:

Add WeChat powcoder

$$\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

Next week, we'll use this to compute recursive functions in the lambda calculus.

OUTLINE

Assignment Project Exam Help

- Revision - Lambda Calculus

https://powcoder.com

- Currying

- Encodings

Add WeChat powcoder

- **Functional Programming: LISP**

LISP

Assignment Project Exam Help

- LISP is the second oldest programming language in common use
- Invented in 1958 by John McCarthy
- Was very popular in the AI boom
- Is a functional programming language
- Is a practical implementation of the Lambda Calculus
- Has many dialects (e.g. Clojure, Common Lisp, Racket, Scheme, etc.)

https://powcoder.com

Add WeChat powcoder

LISP = LISt Processing

Assignment Project Exam Help

- LISP has atoms
  - Numbers, e.g. 10
  - Identifiers, e.g. Foo
  - Strings, e.g. "filename"

https://powcoder.com

- LISP has lists
  - can contain other lists
  - can contain atoms
  - can contain nothing (empty)

Add WeChat powcoder

- *very* small syntax:

```
<object> ::= <atoms> | <list>
<list> ::= "(" { <object> } ")"
```

List Examples in LISP

Assignment Project Exam Help

```
(1 2 3)
()
(+ 1 2)
(* (+ 1 2) (- 2 3))
(sq 1 2)
(setq a 100)
(defun sq (n) (* n n))
(let ((a 6)) a)
(if t 5 6)
(cons 5 6)
(cons (cons 6 7))
```

https://powcoder.com

Add WeChat powcoder

Concepts of LISP

Assignment Project Exam Help

- LISP has a data structure model
  - Lists
  - Atoms
- Even programs are written as lists.
- Even *LISP* is written as a list.
- No other data structures

https://powcoder.com

Add WeChat powcoder

Evaluation

Assignment Project Exam Help

- Prefix notation of function calls as lists
  - Operation is first element
  - Second and following elements are arguments
  - `(<operation> <arg1> ... <argn>)`

https://powcoder.com

- Examples:

```
(+ 4 2)
(+ 3 (- 3 2))
(sq (* 4 2))
```

Add WeChat powcoder

Numerical Functions

Assignment Project Exam Help

- Numerical operations:
  - Addition: `(+ 1 2)`
  - Subtraction: `(- 1 2)`
  - Multiplication: `(* 1 2)`
  - Division: `(/ 1 2)`

https://powcoder.com

- Square root: `(sqrt x)`
- Base Exponent: `(expt x y)`
- Trigonometric Functions: `(sin x)`

Add WeChat powcoder

- Absolute Value: `(abs x)`
- Modulo: `(mod x y)`
- Rounding: `(round x)`

Interaction

Assignment Project Exam Help

- Interaction with lisp is done in a *read-eval-print loop*
- Loop consists of the following steps:
  - Parse input and construct LISP object
  - Evaluate LISP object to produce output
  - Print output object
- Example:

https://powcoder.com

$>> (+ 1 2)$
3

Add WeChat powcoder

Variables

- Variables can be defined by:

```
(setq <var> <value>)
```

Assignment Project Exam Help

- Semantics

```
<var> := <value>
```

https://powcoder.com

- Occurrence of variable symbol replaces variable symbol by the value of the variable
- Example:

Add WeChat powcoder

```
>> (setq a (+ 5 3))
8
>> a
8
```

Quote

Assignment Project Exam Help

- If lists should not be evaluated, use function quote

```
>> (setq a (+ 1 2))
3
>> (setq a (quote (+ 1 2)))
(+ 1 2)
```

https://powcoder.com

- There is a short-hand form, using a single quotation mark

```
>> (setq a '(+ 1 2))
(+ 1 2)
```

Add WeChat powcoder

Condition Function

Assignment Project Exam Help

- Definition: (if <cond> <true-value> <false-value>)
- Boolean values in LISP are given by two symbols
  - Symbol `nil` (equal to the empty list) represents false
  - `T` represents true

https://powcoder.com

```
>> (if nil 1 2)
2
>> (if (= 10 10) 1 2)
1
>> (if () 1 2)
2
```

Add WeChat powcoder

Predicates

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

- Type checking predicates
  - `(atom x)` checks whether x is not a list
  - `(integerp x)` checks whether x is an integer
  - `(numberp x)` checks whether x is a number
  - `(stringp x)` checks whether x is a string
- Numerical predicates
  - `(oddp x)` checks whether x is integer and odd
  - `(evenp x)` checks whether x is integer and even
- Equality
  - `(equal x y)` checks structural equality
  - `(eql x y)` checks atom equality
  - `(eq x y)` checks identity
  - `(= x y)` checks numerical equality
- Logical operators
  - `(or x y)` logical OR
  - `(and x y)` logical AND

Functions

- Function declaration:
  ```
  (defun <name> (<arg1> ... <argn>) body)
  ```
- Translates to:
  ```
  (setq <name> '(lambda (<arg1> ... <argn>) body))
  ```

```
>> (defun factorial (x)
        (if (= x 0)
            1
            (* x (factorial (- x 1)))))
FACTORIAL
>> (factorial 4)
24
```

- Next week we'll do this in lambda calculus directly - without the impurity of defining variables

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Bindings

Assignment Project Exam Help

- Definition:

```
(let ((<name1> <value1>) ... (<namen> <valuen>))
body)
```

https://powcoder.com

```
>> (let
     ((a 3) (b 4) (c 5))
     (+ (* a b) c))
17
>> a
Error: variable A is unbound
```

Add WeChat powcoder

Bindings (2)

Assignment Project Exam Help

- Let allows local bindings of variables
- Bindings might be nested – innermost variable is taken

https://powcoder.com

```
>> (let
     ((a 3))
     (let
       ((a 5))
       a
     ))
   5
```

Add WeChat powcoder

List Construction

Assignment Project Exam Help

- Construction with cons: `(cons <element> <list>)`
- Cons returns a new list with `<element>` as first element, followed by elements in `<list>`
- Construction with list: `(list <elem1> ... <elemn>)`

https://powcoder.com

```
>> (cons 1 nil)
(1)
>> (cons 'a '(b c))
(a b c)
>> (list 1 2 3)
(1 2 3)
```

Add WeChat powcoder

List Access

Assignment Project Exam Help

- Access first element: `(first <list>)`
- Access all but first element: `(rest <list>)`

https://powcoder.com

```
>> (first '(a b c))
a
>> (rest '(a b c))
(b c)
```

Add WeChat powcoder

λ in LISP

Assignment Project Exam Help

```
>> ((lambda (x) (+ x 1)) 4)
5
>> ((lambda (x y z) (* (+ x x) z)) 1 3 5)
10
```

https://powcoder.com

Add WeChat powcoder

λ in Haskell

Assignment Project Exam Help

```
>> (\x -> x + 1) 4
5
>> (\x y z -> (x + x) * z) 1 3 5
10
```

https://powcoder.com

Add WeChat powcoder

λ in Python

Assignment Project Exam Help

```
>> (lambda x: x + 1) 4
5
>> (lambda x: lambda y: lambda z: (x + x) * z)(1)
10
>>> f = lambda x: lambda y: lambda z: (x + x) * z
>>> f
<function <lambda> at 0x02F66270>
>>> f(1)
<function <lambda>.<locals>.<lambda> at 0x02F6615
>>> f(1)(3)
<function <lambda>.<locals>.<lambda>.<locals>.<lar
>>> f(1)(3)(5)
10
```

https://powcoder.com

Add WeChat powcoder

λ in Python

Assignment Project Exam Help

```
>>> NOT = lambda f: lambda x: lambda y: f(y)(x)
>>> TRUE = lambda x: lambda y: x
>>> FALSE = lambda x: lambda y: y
>>> IF = lambda f: lambda x: lambda y: f(x)(y)
>>> IF (NOT(TRUE)) ("a") ("b")
'a'
>>> IF (NOT(NOT(TRUE))) ("a") ("b")
'b'
```

https://powcoder.com

Add WeChat powcoder

Review

- Lambda Calculus revision
  - Application, Abstraction
  - Rewriting
  - α and β reductions
- Currying
  - Multiple arguments
  - Associativity
- Encodings
  - Boolean logic
  - Church numerals, arithmetic
- Functional programming
  - Introduction to LISP
  - Brief look λ in other languages

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder