λ-Calculus-Boolean

Professor: Suman Saha

Bound vs. Free Variables

In $(\lambda x. t)$, the variable x in t is ***bound*** to λx

A variable is ***free*** if it is not bound to any λ

A variable is bound to the closest λ

Examples

$(\lambda x. x)\ x$ applies the identity function to x (i.e., the x after dot is bound to λ)

$\lambda x. \lambda x. x$ is a function that takes a parameter, and returns the identity function (i.e., the inner-most x is bound to the second λ)

Free Variables (Examples)

- ((λ (x) x) y)
- ((λ (x) (x x)) (λ (x) (x x)))
- ((λ (x) (z y)) x)

Useful Rule

We say a term t is ***closed*** if it contains no free variable

Examples: $(\lambda x.\, x)$ $\quad$ $(\lambda x.\, \lambda y.\, x\ y\ )$

Rule: for a capture avoiding substitution

$$t_1\{t_2/x\}$$

the subtle captured-variable check is vacuously true when t_2 is closed

Another Useful Rule

$(\lambda x_1\ x_2\ \ldots x_n. t)\ t_1\ t_2 \ldots\ t_n = t\{t_1/x_1\} \ldots \{t_n/x_n\}$
when $t_1\ t_2\ \ldots\ t_n$ are all closed terms

Evaluation

An ***evaluation*** of a λ term is a sequence

$$t_1 = t_2 = t_3 = \ \ldots$$

where each step is either an α-reduction or a β-reduction

Evaluation Order

No reduction order is specified in classical λ-Calculus

If evaluation terminates, any order gives same result

$$\begin{aligned} & (\lambda x. (\lambda y. x)\ z)\ u \\ = {} & (\lambda y.\ u)\ z \\ = {} & \ u \end{aligned}$$

$$\begin{aligned} & (\lambda x. (\lambda y. x)\ z)\ u \\ = {} & (\lambda x.\ x)\ u \\ = {} & \ u \end{aligned}$$

β-Reduction Example

- ((λ (y)

  ((λ (z) (λ (y) (z y)))

)y))

  (λ (x) x))

β-Reduction Example

- $(\lambda\ (y)\ ((\lambda\ (z)(\lambda\ (y)\ z))\ (\lambda\ (x)\ y)))$

Encoding: Boolean

Shorthand for $\lambda x. (\lambda y. x)$

Booleans

$\text{TRUE} \triangleq \lambda x\ y. x$ $\qquad$ $\text{FALSE} \triangleq \lambda x\ y. y$

Encoding of "if"?

Goal: $\text{IF}\ b\ t\ f = \begin{cases} t \text{ when } b \text{ is TRUE} \\ f \text{ when } b \text{ is FALSE} \end{cases}$

Definition $\text{IF} \triangleq \lambda b\ t\ f. (b\ t\ f)$

Encoding: Boolean

PennState

Booleans

$$\text{TRUE} \triangleq \lambda x\ y.\, x \qquad \text{FALSE} \triangleq \lambda x\ y.\, y$$

Encoding of “and”?

Goal: $\text{AND}\ b_1\ b_2 = \begin{cases} \text{TRUE when } b_1, b_2 \text{ are both TRUE} \\ \text{FALSE otherwise} \end{cases}$

Definition $\text{AND} \triangleq \lambda b_1\ b_2.\ (b_1\ (b_2\ \text{TRUE FALSE})\ \text{FALSE})$

Check that $\text{AND TRUE FALSE} = \text{FALSE}$ (Note 2)

Encoding: Boolean

Definition

$$\text{AND} \triangleq \lambda b_1\, b_2.\ (b_1\ (b_2\ \text{TRUE FALSE})\ \text{FALSE})$$

Check:

$$\text{AND TRUE FALSE} = \text{FALSE}$$