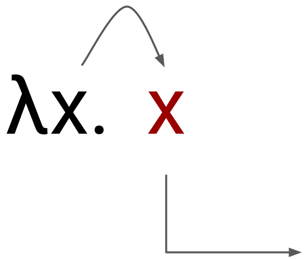
Free and Bound Variables

Ken Q Pu, Faculty of Science, Ontario Tech University

Occurrences of variables: bound vs free

A variable is bound in an expression if it refers to a parameter of some function.



This occurrence of the name **x** is bound because it refers to the parameter.

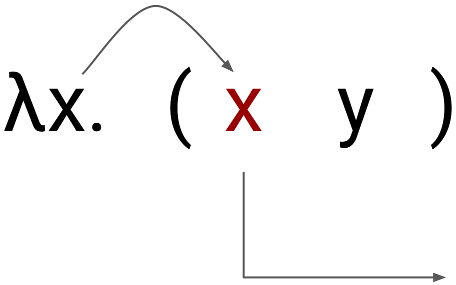
Occurrences of variables: bound vs free

Some occurrences are free in an expression

$$\lambda x. \; (\; x \quad y \;)$$

This occurrence of the name **y** is is free.

Occurrences of variables: bound vs free



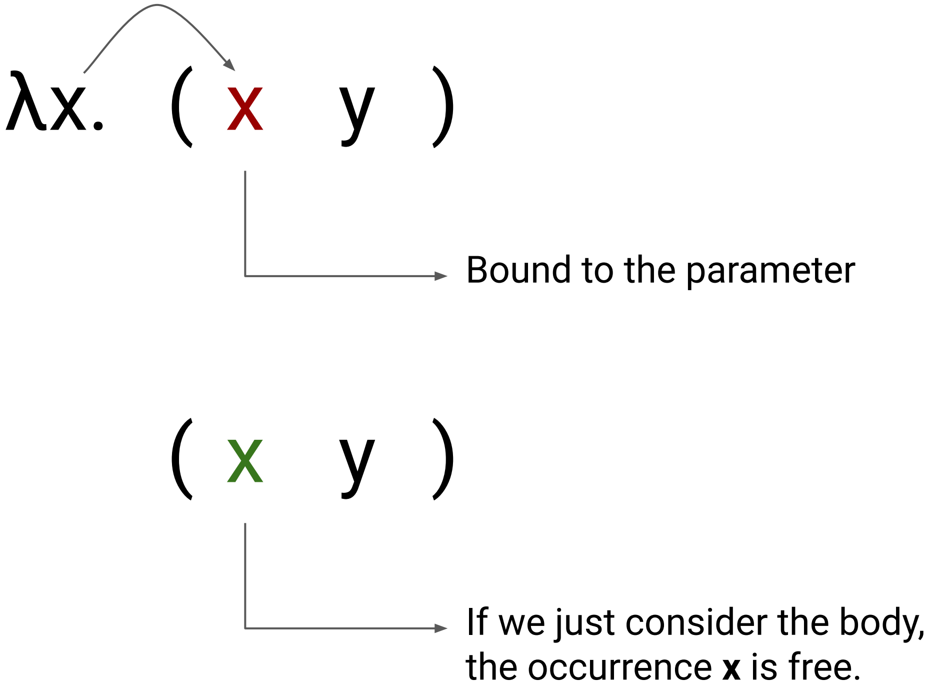
$\lambda x.\ (\ x\quad y\ )$

However, this occurrence of **x** is still bound to the parameter.

Occurrences of variables: bound vs free

Remember:

- An occurrence is bound in a specific expression.
- If we change the expression, its status will change.



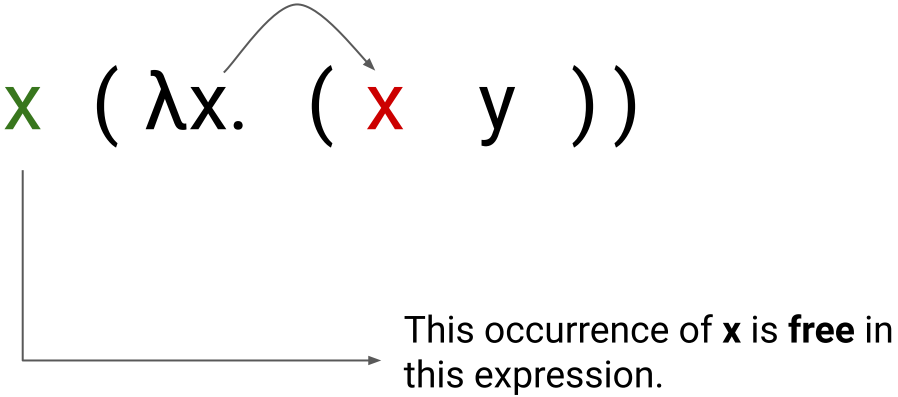
$\lambda x.\ (\ x\ \ y\ )$

Bound to the parameter

$(\ x\ \ y\ )$

If we just consider the body, the occurrence **x** is free.

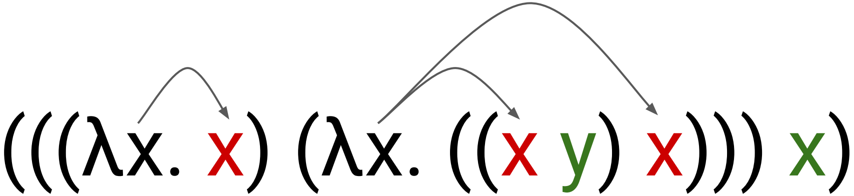
Occurrences of variables: bound vs free



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

This occurrence of **x** is **free** in this expression.

It can get complicated



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

Formal Definition

The **Free** function maps expressions to sets of names. It's defined as:

- $\mathbf{Free}(\mathbf{x}) = \{\mathbf{x}\}$ for all names $\mathbf{x}$
- $\mathbf{Free}(\lambda\ \mathbf{x}.\ e) = \mathbf{Free}(e) - \{\ \mathbf{x}\ \}$
- $\mathbf{Free}(\ e_1\ e_2\ ) = \mathbf{Free}(e_1) \cup \mathbf{Free}(e_2)$

Formal Definition

The **Bound** function maps expressions to sets of names. It's defined as:

- **Bound**(x) = { }
- **Bound**(λ **x**. e) = { **x** } ∪ Bound(**e**)
- **Bound**(e_1 e_2) = **Bound**(e_1) ∪ **Bound**(e_2)

Strange situations

It's possible that

$$\mathbf{x} \in \mathbf{Free}(e) \cap \mathbf{Bound}(e)$$

Consider the case:

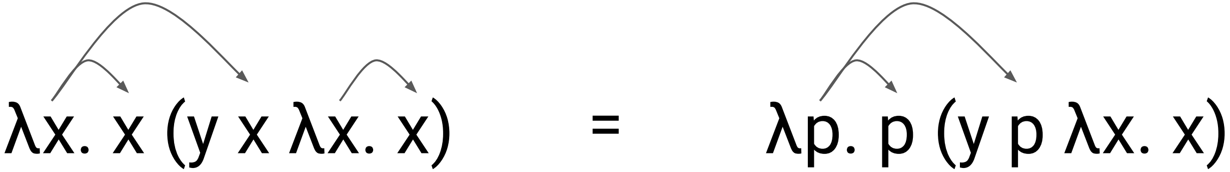
$$e = (\lambda x.\ xy)\ (\lambda y.y)$$

$\text{Free}(e) = \text{Free}(\lambda x.\ xy) \cup \text{Free}(\lambda y.y) = \{\ y\ \}$

$\text{Bound}(e) = \text{Bound}(\lambda x.\ xy) \cup \text{Bound}(\lambda y.y) = \{x, y\}$

Permitted Variable Renaming

$$\lambda x.\ x \quad = \quad \lambda p.\ p \quad = \quad \lambda u.\ u$$



$$\lambda x.\ x\ (y\ x\ \lambda x.\ x) \quad = \quad \lambda p.\ p\ (y\ p\ \lambda x.\ x)$$

The first rewrite rule: α-conversion

Alpha conversion allows us to rewrite an abstraction expression:

$$\lambda x.\ e \longrightarrow \lambda y.\ \underbrace{[y/x]}\ e$$

The substitution must satisfy the following:

1. Replace only free **x** in e.
2. The new parameter **y** must not appear in **e** as a free variable.

LC offers many familiar concepts

- Functions with parameter and body
- Bindings for names in a function body
- Renaming of function parameters