CS 4110
Programming Languages & Logics

Lecture 18

λ-calculus

λ-calculus: History

One of the oldest programming languages...

λ-calculus: History

One of the oldest programming languages...

...even older than general-purpose computers!

λ-calculus: History

One of the oldest programming languages...

...even older than general-purpose computers!

Invented by Alonzo Church and Steven Cole Kleene in the 1930s to describe functions in an unambiguous way

λ-calculus: History

One of the oldest programming languages...

...even older than general-purpose computers!

Invented by Alonzo Church and Steven Cole Kleene in the 1930s to describe functions in an unambiguous way

The "λ" comes from the symbol to denote functions

λ-calculus: History

One of the oldest programming languages...

...even older than general-purpose computers!

Invented by Alonzo Church and Steven Cole Kleene in the 1930s to describe functions in an unambiguous way

The "λ" comes from the symbol to denote functions

The "calculus" comes from it being a system for calculuating by symbolic manipulation

λ-calculus: Why?

Provides the foundation of functional programming

λ-calculus: Why?

Provides the foundation of functional programming

Plays an out-sized role in PL research

λ-calculus: Why?

Provides the foundation of functional programming

Plays an out-sized role in PL research

Good for studying encodings (also known as semantics by translation)

What is a function?

We're all familiar with functions from mathematics...

$$f(x) = x^3$$
$$g(y) = y^3 - 2y^2 + 5y - 6.$$

What is a function?

We're all familiar with functions from mathematics...

$$f(x) = x^3$$
$$g(y) = y^3 - 2y^2 + 5y - 6.$$

What can you do with a function?

What is a function?

We're all familiar with functions from mathematics...

$$
\begin{aligned}
f(x) &= x^3 \\
g(y) &= y^3 - 2y^2 + 5y - 6.
\end{aligned}
$$

What can you do with a function?

Apply it to an argument!

What is a function?

We're all familiar with functions from mathematics...

$$f(x) = x^3$$
$$g(y) = y^3 - 2y^2 + 5y - 6.$$

What can you do with a function?

Apply it to an argument!

For example, $f(2) = 2^3 = 8$

λ-calculus

$$
\begin{array}{rcll}
e & ::= & x & \textit{Variable} \\
& | & \lambda x.\, e & \textit{Abstraction} \\
& | & e\ e & \textit{Application}
\end{array}
$$

Example: Identity function

$$\lambda x.\, x$$

Example: Composition of f and g

$$\lambda x.\, f\,(g\,x)$$

Example: Constant function returning 42

$$\lambda x.\, 42$$

Example: Application of identity function to 42

$$(\lambda x.\, x)\; 42$$

Example: Higher-order function

$$\lambda y.\, \lambda x.\, x$$

Syntactic Conventions

- λ expressions extend to the right as far as possible

Syntactic Conventions

- λ expressions extend to the right as far as possible

Example

$\lambda x.\, x\ \lambda y.\, y$ is the same as $\lambda x.\, x\ (\lambda y.\, y)$ and not the same as $(\lambda x.\, x)\ (\lambda y.\, y)$

Syntactic Conventions

- λ expressions extend to the right as far as possible

Example

$\lambda x.\, x\ \lambda y.\, y$ is the same as $\lambda x.\, x\ (\lambda y.\, y)$ and not the same as $(\lambda x.\, x)\ (\lambda y.\, y)$

- Application is left associative

Syntactic Conventions

- λ expressions extend to the right as far as possible

Example

$\lambda x.\, x\ \lambda y.\, y$ is the same as $\lambda x.\, x\ (\lambda y.\, y)$ and not the same as $(\lambda x.\, x)\ (\lambda y.\, y)$

- Application is left associative

Example

$e_1\ e_2\ e_3$ is the same as $(e_1\ e_2)\ e_3$ and not the same as $e_1\ (e_2\ e_3)$

Variable Binding

Definition (Free and Bound Variables)

An occurrence of a variable x is bound if there is an encosing $\lambda x.\, e$.

Otherwise the occurrence is free

Variable Binding

Definition (Free and Bound Variables)

An occurrence of a variable x is bound if there is an encosing $\lambda x.\, e$.

Otherwise the occurrence is free

Definition (Closed Expressions)

An expression is closed if all of its variables are bound

Example: free and bound variables

Example

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

Example: free and bound variables

Example

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

- Both occurrences of x bound

Example: free and bound variables

Example

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

- Both occurrences of x bound
- The occurrence of a is free

Example: free and bound variables

Example

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

- Both occurrences of x bound
- The occurrence of a is free
- The last occurrence of y is free

α-equivalence

The name of bound variables is not important.

Example

$\lambda x.\, x$ is the same function as $\lambda y.\, y$

Expressions e_1 and e_2 that differ only in the name of bound variables are called α-equivalent, written $e_1 \equiv_\alpha e_2$

Higher-order functions

In λ-calculus, functions are values: functions can take functions as arguments and return functions as results

In the pure λ-calculus, every value is a function, and every result is a function!

Example

$$\lambda f.\, f\,(\lambda x.\, x)$$

Intuition: Takes a function *f* and applies it to the identity

β-equivalence

When we apply a function $\lambda x.\, e_1$ to an argument e_2, we would like to regard it as equivalent to e_1 where every free occurrence of x has been replaced by e_2

This notion of equivalence is called β-equivalence, often written $e_1 \equiv_\beta e_2$

Notation: We write $e_1\{e_2/x\}$ to mean expression e_1 with all free occurrences of x replaced with e_2.

Rewriting $(\lambda x.\, e_1)\ e_2$ into $e_1\{e_2/x\}$ is called a β-reduction.

But in general, there can be several ways to perform β-reductions...

Call-by-value

Let v range over functions, that is $\lambda x.\, e$.

$$\frac{e_1 \longrightarrow e_1'}{e_1\, e_2 \longrightarrow e_1'\, e_2} \qquad \frac{e \longrightarrow e'}{v\, e \longrightarrow v\, e'} \qquad \beta\, \frac{}{(\lambda x.\, e)\, v \longrightarrow e\{v/x\}}$$

Example reductions

$$(\lambda x.\, \lambda y.\, y\, x)\; (5 + 2)\; \lambda x.\, x + 1$$

Example reductions

$$(\lambda x.\, \lambda y.\, y\, x)\, (5 + 2)\, \lambda x.\, x + 1 \;\; \rightarrow (\lambda x.\, \lambda y.\, y\, x)\, 7\, \lambda x.\, x + 1$$

Example reductions

$$
\begin{aligned}
(\lambda x.\, \lambda y.\, y\, x)\, (5 + 2)\, \lambda x.\, x + 1 &\to (\lambda x.\, \lambda y.\, y\, x)\, 7\, \lambda x.\, x + 1 \\
&\to (\lambda y.\, y\, 7)\, \lambda x.\, x + 1
\end{aligned}
$$

Example reductions

$$\begin{aligned}(\lambda x.\, \lambda y.\, y\, x)\, (5+2)\, \lambda x.\, x+1 &\to (\lambda x.\, \lambda y.\, y\, x)\, 7\, \lambda x.\, x+1 \\ &\to (\lambda y.\, y\, 7)\, \lambda x.\, x+1 \\ &\to (\lambda x.\, x+1)\, 7\end{aligned}$$

Example reductions

$$\begin{aligned}
(\lambda x.\, \lambda y.\, y\, x)\; (5 + 2)\; \lambda x.\, x + 1 \;&\to (\lambda x.\, \lambda y.\, y\, x)\; 7\; \lambda x.\, x + 1 \\
&\to (\lambda y.\, y\; 7)\; \lambda x.\, x + 1 \\
&\to (\lambda x.\, x + 1)\; 7 \\
&\to 7 + 1
\end{aligned}$$

Example reductions

$$\begin{aligned}
(\lambda x.\, \lambda y.\, y\, x)\, (5 + 2)\, \lambda x.\, x + 1 &\to (\lambda x.\, \lambda y.\, y\, x)\, 7\, \lambda x.\, x + 1 \\
&\to (\lambda y.\, y\, 7)\, \lambda x.\, x + 1 \\
&\to (\lambda x.\, x + 1)\, 7 \\
&\to 7 + 1 \\
&\to 8
\end{aligned}$$

Example reductions

$$
\begin{aligned}
(\lambda x.\, \lambda y.\, y\, x)\, (5 + 2)\, \lambda x.\, x + 1 \;&\rightarrow (\lambda x.\, \lambda y.\, y\, x)\; 7\; \lambda x.\, x + 1 \\
&\rightarrow (\lambda y.\, y\, 7)\; \lambda x.\, x + 1 \\
&\rightarrow (\lambda x.\, x + 1)\; 7 \\
&\rightarrow 7 + 1 \\
&\rightarrow 8
\end{aligned}
$$

Note: this example uses an applied lambda calculus that also includes reduction rules for arithmetic expressions.

Call-by-name

$$\frac{e_1 \to e_1'}{e_1\, e_2 \to e_1'\, e_2}$$

$$\beta \; \frac{}{(\lambda x.\, e_1)\, e_2 \to e_1\{e_2/x\}}$$

Example reduction

$$(\lambda x.\, \lambda y.\, y\, x)\, (5 + 2)\, \lambda x.\, x + 1$$

Example reduction

$$(\lambda x.\, \lambda y.\, y\, x)\, (5 + 2)\, \lambda x.\, x + 1 \;\; \to (\lambda y.\, y\, (5 + 2))\, \lambda x.\, x + 1$$

Example reduction

$$\begin{aligned}(\lambda x.\, \lambda y.\, y\, x)\,(5+2)\,\lambda x.\, x+1 &\to (\lambda y.\, y\,(5+2))\,\lambda x.\, x+1 \\ &\to (\lambda x.\, x+1)\,(5+2)\end{aligned}$$

Example reduction

$$\begin{aligned}(\lambda x.\, \lambda y.\, y\, x)\,(5+2)\,\lambda x.\, x+1 &\to (\lambda y.\, y\,(5+2))\,\lambda x.\, x+1\\ &\to (\lambda x.\, x+1)\,(5+2)\\ &\to (5+2)+1\end{aligned}$$

Example reduction

$$\begin{aligned}(\lambda x.\, \lambda y.\, y\, x)\,(5+2)\,\lambda x.\, x+1 &\to (\lambda y.\, y\,(5+2))\,\lambda x.\, x+1\\ &\to (\lambda x.\, x+1)\,(5+2)\\ &\to (5+2)+1\\ &\to 7+1\end{aligned}$$

Example reduction

$$
\begin{aligned}
(\lambda x.\, \lambda y.\, y\, x)\, (5+2)\, \lambda x.\, x+1 &\rightarrow (\lambda y.\, y\, (5+2))\, \lambda x.\, x+1 \\
&\rightarrow (\lambda x.\, x+1)\, (5+2) \\
&\rightarrow (5+2)+1 \\
&\rightarrow 7+1 \\
&\rightarrow 8
\end{aligned}
$$