# Explicit Substitutions Part 2: Explicit Substitutions

slides: https://github.com/donovancrichton/Talks

Donovan Crichton

August 2025

■ Introduce explicit substitution extensions to the STLC.

- Introduce explicit substitution extensions to the STLC.
- Provide examples on what the new substitutions mean. between types, terms, contexts, and substitutions.

- Introduce explicit substitution extensions to the STLC.
- Provide examples on what the new substitutions mean.
- Present an equational theory that determines relationships between types, terms, contexts, and substitutions.

- Introduce explicit substitution extensions to the STLC.
- Provide examples on what the new substitutions mean.
- Present an equational theory that determines relationships between types, terms, contexts, and substitutions.
- Scale up to a dependent type theory

# Untyped LC

## Untyped Lambda Calculus

$$V ::= x, y, z, \dots$$
  $M, N ::= V$  Variable. 
$$\mid M N \qquad \qquad \text{Application.}$$
 
$$\mid \lambda V. M \qquad \qquad \text{Abstraction.}$$

## The Simply Typed Lambda Calculus: Grammar

V ::= x, y, z, ...

Variable Set

## The Simply Typed Lambda Calculus: Grammar

$$V ::= x, y, z, \dots$$

$$\alpha, \beta ::= * \mid \alpha \to \beta$$

Variable Set

Types

## The Simply Typed Lambda Calculus: Grammar

$$V ::= x, y, z, ...$$
  
 $\alpha, \beta ::= * \mid \alpha \to \beta$ 

 $M, N ::= V \mid M N \mid \lambda(V : \alpha).M$ 

Variable Set

Types

**Terms** 

#### The Simply Typed Lambda Calculus: Grammar

$$V::=x,y,z,\dots$$
 Variable Set  $\alpha,\beta:=*\mid \alpha \to \beta$  Types  $M,N::=V\mid MN\mid \lambda(V:\alpha).M$  Terms  $\Gamma::=\diamond\mid \Gamma,x_m:\alpha$  Contexts

#### The Simply Typed Lambda Calculus: Grammar

$$V::=x,y,z,\dots$$
 Variable Set  $\alpha,\beta:=*\mid \alpha \to \beta$  Types  $M,N::=V\mid MN\mid \lambda(V:\alpha).M$  Terms  $\Gamma::=\diamond\mid \Gamma,x_m:\alpha$  Contexts

Why  $x_m$ :  $\alpha$  in the Contexts set, and not V:  $\alpha$ ?

#### The Simply Typed Lambda Calculus: Grammar

$$V::=x,y,z,\dots$$
 Variable Set  $\alpha,\beta:=*\mid \alpha \to \beta$  Types  $M,N::=V\mid MN\mid \lambda(V:\alpha).M$  Terms  $\Gamma::=\diamond\mid \Gamma,x_m:\alpha$  Contexts

Why  $x_m$ :  $\alpha$  in the Contexts set, and not V:  $\alpha$ ?

 $x_m$  represents a variable in our meta language

### STLC Rules

## STLC: Typing Rules

$$\frac{\Gamma \vdash \alpha \text{ Type} \qquad \Gamma \vdash \beta \text{ Type}}{\Gamma \vdash \alpha \rightarrow \beta \text{ Type}} (\rightarrow \text{-}Form)$$

#### STLC Rules

## STLC: Typing Rules

$$\frac{\Gamma \vdash \alpha \; \mathsf{Type} \qquad \Gamma \vdash \beta \; \mathsf{Type}}{\Gamma \vdash \alpha \to \beta \; \mathsf{Type}} (\to \mathsf{-Form})$$

$$\frac{\Gamma \vdash \alpha \to \beta \; \mathsf{Type} \qquad \Gamma, x_m : \alpha \vdash M : \beta}{\Gamma \vdash \lambda (x : \alpha).M : \alpha \to \beta} (\to \mathsf{-Intro})$$

#### STLC Rules

## STLC: Typing Rules

$$\frac{\Gamma \vdash \alpha \; \mathsf{Type} \qquad \Gamma \vdash \beta \; \mathsf{Type}}{\Gamma \vdash \alpha \to \beta \; \mathsf{Type}} (\to -\mathit{Form})$$

$$\frac{\Gamma \vdash \alpha \to \beta \; \mathsf{Type} \qquad \Gamma, x_m : \alpha \vdash M : \beta}{\Gamma \vdash \lambda (x : \alpha).M : \alpha \to \beta} (\to -\mathit{Intro})$$

$$\frac{\Gamma \vdash \alpha \to \beta \; \mathsf{Type} \qquad \Gamma \vdash M : \alpha \to \beta \qquad \Gamma \vdash N : \alpha}{\Gamma \vdash M \; N : \beta} (\to -\mathit{Elim})$$

#### STLC with Explicit Substitutions: Grammar

$$\alpha,\beta ::= \ * \mid \alpha \to \beta$$

Types

## STLC with Explicit Substitutions: Grammar

$$\begin{array}{ll} \alpha,\beta::=\ *\mid \alpha \to \beta & \text{Types} \\ \textit{M},\textit{N}::=1\ \mid \textit{M}\;\textit{N}\mid \lambda\alpha.\textit{M}\mid \textit{M}[\gamma] & \text{Terms} \end{array}$$

## STLC with Explicit Substitutions: Grammar

$$\begin{array}{lll} \alpha,\beta::= *\mid \alpha \rightarrow \beta & \text{Types} \\ \textit{M},\textit{N}::= 1\mid \textit{M}\;\textit{N}\mid \lambda\alpha.\textit{M}\mid \textit{M}[\gamma] & \text{Terms} \\ \gamma,\delta::= \textit{id}\mid \uparrow\mid \gamma,\textit{M}:\alpha\mid \gamma\circ\delta & \text{Substitutions} \end{array}$$

## STLC with Explicit Substitutions: Grammar

$$\begin{array}{lll} \alpha,\beta::= \ * \mid \alpha \to \beta & \text{Types} \\ \textit{M},\textit{N}::= 1 \mid \textit{M} \; \textit{N} \mid \lambda \alpha. \textit{M} \mid \textit{M}[\gamma] & \text{Terms} \\ \gamma,\delta::= \textit{id} \mid \uparrow \mid \gamma,\textit{M}:\alpha \mid \gamma \circ \delta & \text{Substitutions} \\ \Gamma::= \diamond \mid \Gamma,\alpha & \text{Contexts} \end{array}$$

#### **Explicit Substitution Terms**

 $M, N ::= 1 \mid M N \mid \lambda \alpha.M \mid M[\gamma]$ 

Terms

#### Explicit Substitution Terms

$$M, N ::= 1 \mid M N \mid \lambda \alpha. M \mid M[\gamma]$$

Terms

#### Intuition

 $\blacksquare$  *M N* is our standard application.

#### **Explicit Substitution Terms**

$$\mathit{M}, \mathit{N} ::= 1 \ | \ \mathit{M} \ \mathit{N} \ | \ \lambda \alpha. \mathit{M} \ | \ \mathit{M}[\gamma]$$
 Terms

- $\blacksquare$  *M N* is our standard application.
- $\lambda \alpha.M$  abstraction refers to the type, but no longer the variable.

#### **Explicit Substitution Terms**

$$\mathit{M},\mathit{N} ::= 1 \mid \mathit{M} \; \mathit{N} \mid \lambda \alpha.\mathit{M} \mid \mathit{M}[\gamma]$$
 Terms

- $\blacksquare$  *M N* is our standard application.
- $\lambda \alpha.M$  abstraction refers to the type, but no longer the variable.
- 1 is the most recently bound De-Bruijn index variable.

#### **Explicit Substitution Terms**

$$\mathit{M},\mathit{N} ::= 1 \mid \mathit{M} \; \mathit{N} \mid \lambda \alpha.\mathit{M} \mid \mathit{M}[\gamma]$$
 Terms

- $\blacksquare$  *M N* is our standard application.
- $\lambda \alpha.M$  abstraction refers to the type, but no longer the variable.
- 1 is the most recently bound De-Bruijn index variable.
- $M[\gamma]$  is term M with substituion  $\gamma$  applied.

#### Explicit Substitutions: Subs

$$\gamma, \delta ::= id \mid \uparrow \mid \gamma, M : \alpha \mid \gamma \circ \delta$$
 Substitutions

#### Explicit Substitutions: Subs

$$\gamma, \delta ::= id \mid \uparrow \mid \gamma, M : \alpha \mid \gamma \circ \delta$$

Substitutions

#### Intuition

■ *id* is the identity substitution.

#### Explicit Substitutions: Subs

$$\gamma, \delta ::= id \mid \uparrow \mid \gamma, M : \alpha \mid \gamma \circ \delta$$

Substitutions

- *id* is the identity substitution, mapping the source context to itself.
- ↑ is the weakening or shifting substituion. We 'weaken' the context by one additional variable.

#### Explicit Substitutions: Subs

$$\gamma, \delta ::= id \mid \uparrow \mid \gamma, M : \alpha \mid \gamma \circ \delta$$
 Substitutions

- *id* is the identity substitution, mapping the source context to itself.
- ↑ is the weakening or shifting substituion. We 'weaken' the context by one additional variable.
- lacksquare  $\gamma, M$  is the substitution extension or snoc. We extend the substitution by a term.

#### Explicit Substitutions: Subs

$$\gamma, \delta ::= id \mid \uparrow \mid \gamma, M : \alpha \mid \gamma \circ \delta$$

Substitutions

- *id* is the identity substitution, mapping the source context to itself.
- ↑ is the weakening or shifting substituion. We 'weaken' the context by one additional variable.
- $\bullet$   $\gamma$ , M:  $\alpha$  is the substitution extension or snoc. We extend the substitution by a term.
- $\quad \bullet \quad \gamma \circ \delta$  is the composition substitution where we compose two substitutions together

## Example: Identity

$$\diamond, \alpha, \beta, \theta \vdash id : \diamond, \alpha, \beta, \theta$$

We move from source context to target context with no changes.

## Example: Identity

$$\diamond, \alpha, \beta, \theta \vdash id : \diamond, \alpha, \beta, \theta$$

We move from source context to target context with no changes.



# Example: Weakening (Shift)

$$\diamond, \alpha, \beta, \theta \vdash \uparrow : \diamond, \alpha, \beta$$

We add a new variable (weaken) the context, which shifts everything else up by one index.

# Example: Weakening (Shift)

$$\diamond, \alpha, \beta, \theta \vdash \uparrow : \diamond, \alpha, \beta$$

We add a new variable (weaken) the context, which shifts everything else up by one index.



# Example: Extension (Snoc)

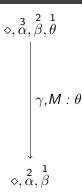
$$\diamond, \alpha, \beta \vdash (\gamma, M : \theta) : \diamond, \alpha, \beta, \theta$$

We extend substitution  $\gamma$  by a term  $M:\theta$  and drop  $\theta$  from our source context. This models the act of substituting in a term for a variable in context.

# Example: Extension (Snoc)

$$\diamond, \alpha, \beta \vdash (\gamma, M : \theta) : \diamond, \alpha, \beta, \theta$$

We extend substitution  $\gamma$  by a term  $M:\theta$  and drop  $\theta$  from our source context. This models the act of substituting in a term for a variable in context.



# Example: Composition

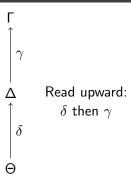
## Composition combines two substitutions

If  $\Delta \vdash \gamma : \Gamma$  and  $\Theta \vdash \delta : \Delta$ then  $\Theta \vdash \gamma \circ \delta : \Gamma$ 

## Example: Composition

### Composition combines two substitutions

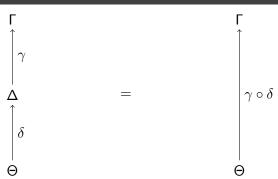
If 
$$\Delta \vdash \gamma : \Gamma$$
 and  $\Theta \vdash \delta : \Delta$  then  $\Theta \vdash \gamma \circ \delta : \Gamma$ 



## Example: Composition

### Composition combines two substitutions

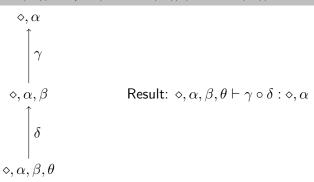
If  $\Delta \vdash \gamma : \Gamma$  and  $\Theta \vdash \delta : \Delta$  then  $\Theta \vdash \gamma \circ \delta : \Gamma$ 



## Example: Composition

#### Example: Specific Composition

Consider 
$$\diamond, \alpha, \beta \vdash \gamma : \diamond, \alpha \text{ and } \diamond, \alpha, \beta, \theta \vdash \delta : \diamond, \alpha, \beta$$



### Reading the diagram

Start from  $\diamond, \alpha, \beta, \theta$ , apply  $\delta$  to get  $\diamond, \alpha, \beta$ , then apply  $\gamma$  to reach  $\diamond, \alpha$ 

## A Note On Notation: Composition and Context

#### A Note On Notation: Composition and Context

#### Two notations for composition exist in the literature:

- Sequential (Abadi et al. original):  $\gamma \circ \delta$  means "first  $\gamma$ , then  $\delta$ "
- Categorical (most papers):  $\delta \circ \gamma$  means "first  $\gamma$ , then  $\delta$ "
- In this talk: We use  $\gamma \circ \delta$  meaning "first  $\delta$ , then  $\gamma$ "

#### Why the difference?

The original paper used sequential notation matching evaluation order.

Modern presentations use categorical notation matching function composition.

## A Note On Notation: Composition and Context

#### A Note On Notation: Composition and Context

Critical: Pay attention to context order in substitution typing!

In the judgment  $\Gamma \vdash \gamma : \Delta$ :

- $\blacksquare$   $\triangle$  is the **source** context (domain)
- $\Gamma$  is the **target** context (codomain)
- $\blacksquare$   $\gamma$  maps FROM  $\Delta$  TO  $\Gamma$

### Composition typing rule

$$\frac{\Gamma \vdash \gamma : \Delta \qquad \Delta \vdash \delta : \Theta}{\Gamma \vdash \gamma \circ \delta : \Theta}$$

The middle context  $\Delta$  must match!

Read:  $\gamma \circ \delta$  takes us from  $\Theta$  to  $\Gamma$  via  $\Delta$ 

# Explicit Subs: Context Typing

### Explicit Subs: Context Typing

$$\frac{}{|+ \diamond \operatorname{ctx}|} (\operatorname{\mathit{Ctx-Empty}})$$

# Explicit Subs: Context Typing

### Explicit Subs: Context Typing

$$\frac{}{\vdash \diamond \mathsf{ctx}}(\mathit{Ctx-Empty})$$

$$\frac{\Gamma \vdash \alpha \; \mathsf{Type}}{\vdash \Gamma, \alpha \; \mathsf{ctx}}(\mathit{Ctx-Extend})$$

$$\frac{\vdash \Gamma \ \mathsf{ctx}}{\Gamma \vdash \ast \ \mathsf{Type}} (\mathit{Type-Base})$$

$$\frac{\vdash \Gamma \ \mathsf{ctx}}{\Gamma \vdash \ast \ \mathsf{Type}} (\mathit{Type-Base})$$
 
$$\frac{\Gamma \vdash \alpha \ \mathsf{Type} \qquad \Gamma \vdash \beta \ \mathsf{Type}}{\Gamma \vdash \alpha \to \beta \ \mathsf{Type}} (\mathit{Type-Arrow})$$

$$\frac{\vdash \Gamma, \alpha \mathsf{ctx}}{\Gamma, \alpha \vdash 1 : \alpha} (\mathit{Var})$$

$$\frac{\vdash \Gamma, \alpha \mathsf{ctx}}{\Gamma, \alpha \vdash 1 : \alpha} (Var)$$
$$\frac{\Gamma, \alpha \vdash M : \beta}{\Gamma \vdash \lambda \alpha. M : \alpha \to \beta} (Abs)$$

$$\frac{\frac{\vdash \Gamma, \alpha \operatorname{ctx}}{\Gamma, \alpha \vdash 1 : \alpha}(Var)}{\frac{\Gamma, \alpha \vdash M : \beta}{\Gamma \vdash \lambda \alpha. M : \alpha \to \beta}(Abs)}$$

$$\frac{\Gamma \vdash M : \alpha \to \beta \qquad \Gamma \vdash N : \alpha}{\Gamma \vdash M N : \beta}(App)$$

$$\frac{ \vdash \Gamma, \alpha \operatorname{ctx}}{\Gamma, \alpha \vdash 1 : \alpha} (Var)$$

$$\frac{\Gamma, \alpha \vdash M : \beta}{\Gamma \vdash \lambda \alpha. M : \alpha \to \beta} (Abs)$$

$$\frac{\Gamma \vdash M : \alpha \to \beta \qquad \Gamma \vdash N : \alpha}{\Gamma \vdash M N : \beta} (App)$$

$$\frac{\Delta \vdash M : \alpha \qquad \Gamma \vdash \gamma : \Delta}{\Gamma \vdash M[\gamma] : \alpha} (Subst)$$

$$\frac{\vdash \Gamma \operatorname{ctx}}{\Gamma \vdash id : \Gamma} (Sub\text{-}Id)$$

$$\frac{\vdash \Gamma \mathsf{ctx}}{\Gamma \vdash \mathsf{id} : \Gamma} (\mathsf{Sub}\text{-}\mathsf{Id})$$

$$\frac{\vdash \Gamma, \alpha \mathsf{ctx}}{\Gamma, \alpha \vdash \uparrow : \Gamma} (\mathsf{Sub}\text{-}\mathsf{Weak})$$

$$\frac{\vdash \Gamma \operatorname{ctx}}{\Gamma \vdash id : \Gamma}(Sub\text{-}Id)$$

$$\frac{\vdash \Gamma, \alpha \operatorname{ctx}}{\Gamma, \alpha \vdash \uparrow : \Gamma}(Sub\text{-}Weak)$$

$$\frac{\Gamma \vdash \gamma : \Delta \qquad \Gamma \vdash M : \alpha}{\Gamma \vdash \gamma, M : \alpha : \Delta, \alpha}(Sub\text{-}Ext)$$

$$\frac{\vdash \Gamma \operatorname{ctx}}{\Gamma \vdash \operatorname{id} : \Gamma}(\operatorname{Sub-Id})$$

$$\frac{\vdash \Gamma, \alpha \operatorname{ctx}}{\Gamma, \alpha \vdash \uparrow : \Gamma}(\operatorname{Sub-Weak})$$

$$\frac{\Gamma \vdash \gamma : \Delta \qquad \Gamma \vdash M : \alpha}{\Gamma \vdash \gamma, M : \alpha : \Delta, \alpha}(\operatorname{Sub-Ext})$$

$$\frac{\Gamma \vdash \gamma : \Delta \qquad \Delta \vdash \delta : \Theta}{\Gamma \vdash \gamma \circ \delta : \Theta}(\operatorname{Sub-Comp})$$

## Equational Theory: Category Laws

### Equational Theory: Category Laws

$$id \circ \gamma = \gamma$$
  $\gamma \circ id = \gamma$ 

$$\gamma \circ id = \gamma$$

#### **Identity Laws**

The identity substitution is both a left and right unit for composition

# Equational Theory: Category Laws

### Equational Theory: Category Laws

$$id \circ \gamma = \gamma$$
  $\gamma \circ id = \gamma$  
$$(\gamma \circ \delta) \circ \theta = \gamma \circ (\delta \circ \theta)$$

Identity and Associativity Laws

Substitution composition forms a category

#### Equational Theory: Closure Laws

$$1[id] = 1$$

Variable Identity

Applying id to 1 is 1.

#### Equational Theory: Closure Laws

$$1[id] = 1$$

$$1[id, M : \alpha] = M$$

Variable Extension of Identity is Substitution.

Applying id extended with M to 1 is M. This is our actual substitution being performed.

#### Equational Theory: Closure Laws

$$1[id, M : \alpha] = M$$
  
 $1[\uparrow] = 1[\uparrow \circ \gamma]$   
 $M[\gamma][\delta] = M[\delta \circ \gamma]$ 

Closure (Application) is composition

Note pay careful attention to the order of sequential closures here.

#### Equational Theory: Closure Laws

$$1[id, M : \alpha] = M$$
$$1[\uparrow] = 1[\uparrow \circ \gamma]$$
$$M[\gamma][\delta] = M[\delta \circ \gamma]$$
$$(\delta \circ \uparrow), 1[\delta] = \delta$$

#### Weaken, Extension

A substitution is equal to its most recent element extended on to the rest.

## Equational Theory: Weakening Laws

### Equational Theory: Weakening Laws

$$\uparrow$$
, 1 = id

Weakening Extended with Var is Identity

Extending De-Bruijn index 0 with a new variable, and then removing it is the identity substitution.

## Equational Theory: Weakening Laws

#### Equational Theory: Weakening Laws

$$\uparrow$$
,  $1 = id$ 

$$\uparrow \circ \gamma, M: \alpha = \gamma$$

#### Weaken-Extend Cancel

Weakening an extended substituion recovers our original substitution.

## Equational Theory: Distribution Laws

### Equational Theory: Distribution Laws

$$(M N)[\gamma] = M[\gamma] N[\gamma]$$

Application Distribution

Substitution distributes over application.

## Equational Theory: Distribution Laws

#### Equational Theory: Distribution Laws

$$(M\ N)[\gamma] = M[\gamma]\ N[\gamma]$$

$$(\lambda \alpha. M)[\gamma] = \lambda \alpha. M[(\uparrow \circ \gamma), 1 : \alpha]$$

#### Abstraction Distribution

Substitution distributes under binders by extending a weakening with variable.

## Equational Theory: Extension Laws

#### Equational Theory: Extension Laws

$$\delta \circ (\gamma, M : \alpha) = (\delta \circ \gamma), M[\delta]$$

#### **Extension Composition**

Composition distributes into extensions.

## Equational Theory: Extension Laws

#### Equational Theory: Extension Laws

$$(\gamma, M : \alpha) \circ \delta = (\gamma \circ \delta, M[\delta] : \alpha)$$
 
$$id = (\uparrow, 1 : \alpha)$$

Identity Decomposition

Identity as shift extended with variable.

## Equational Theory: Computation Laws

#### Equational Theory: Computation Laws

$$(\lambda \alpha.M) N = M[id, N : \alpha]$$

#### Beta Reduction

Application reduces by substituting the argument

# Equational Theory: Computation Laws

#### Equational Theory: Computation Laws

$$(\lambda \alpha.M) \ N = M[id, N : \alpha]$$

$$M = \lambda \alpha.(M[\uparrow] \ 1)$$

#### Eta Expansion

Any term can be eta-expanded by shifting and applying to variable 1

## Equational Theory: Computation Laws

#### Equational Theory: Computation Laws

$$(\lambda \alpha.M) \ N = M[id, N : \alpha]$$

$$M = \lambda \alpha.(M[\uparrow] \ 1)$$

$$1[id, M : \alpha] = M$$

#### Variable Beta

Substituting for variable 1 with extension gives the substituted term

## **Equational Theory Intuition**

#### **Equational Theory Intuition**

- Equations let us rewrite expressions to canonical forms
- Substitutions can be simplified before application
- No need for complex variable renaming ( $\alpha$ -conversion)

#### Conclusion

We've seen how explicit substitutions:

- Make substitution a syntactic operation
- Use De Bruijn indices to avoid variable names
- Form a category with composition and identity
- Have a complete equational theory

#### Conclusion

We've seen how explicit substitutions:

- Make substitution a syntactic operation
- Use De Bruijn indices to avoid variable names
- Form a category with composition and identity
- Have a complete equational theory

#### **Benefits**

- Cleaner metatheory and proofs
- Direct implementation of theory
- Foundation for dependent type theory

### Next Steps

This framework extends naturally to:

- Dependent type theory (as hinted)
- Proof assistants and type checkers
- Abstract machines for evaluation
- Categorical semantics of type theory

#### Thank You!

Questions?

Slides available at:

https://github.com/donovancrichton/Talks