August 29, 2026 @ LOPSTR+PPDP

A Typed and Unified Reflection for shift and shift0

Yui Tamura and Kenichi Asai

Ochanomizu University, Japan

Delimited continuations & Computational effects

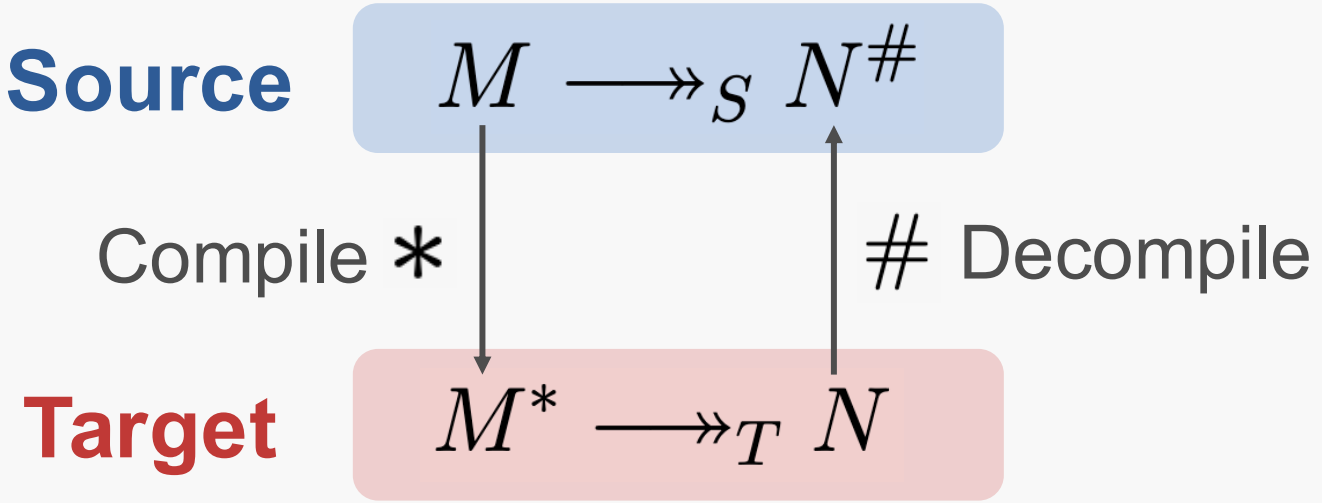
- **Delimited continuations**
  - Rest of computation within a limited scope
  - Four operators: `shift`, `shift0`, `control`, `control0`
- **Computational effects**
  - Behavior beyond pure functions
  - Algebraic effects and handlers (AEH): deep, shallow
- **AEH theory**
  - A foundation via control operators

Toward practical AEH theory

- **A typed setting**
  - Modern languages with static typing
  - Important for safety guarantees
- **A unified framework**
  - Both deep & shallow handlers are available
  - Accommodate them in a single foundation
- **Our focus: reasoning about program transformations**

Reduction correspondence between two calculi

- **Compiler correctness**
  - Target optimizations hold in source calculus
  - Optimize directly to the source program



To establish a typed & unified reflection for AEH

- **Stepping stone: reflection for all four operators**
  - Extend the techniques for delimited control to AEH
- **Connections between control operators & AEH**
  - Equally expressive
    - Deep handlers ≒ shift0
    - Shallow handlers ≒ control0
  - Common framework
    - Interpreter for effect handlers [Asai and Fujii, 2025]

Previous work & Our Position

Pure λ_v [Sabry & Wadler '97]

Delimited control

`shift/reset` [Biernacki+ '20]
Continuation-composing style

`shift/reset` [Honda+ '23]
Typed setting & Proof in Agda

This work: `shift` & `shift0`

All four operators

`shift0/dollar` [Biernacki+ '21]
Stack of continuations

AEH

deep handler [Cong & Asai+ '23]
In progress

deep & shallow handler

Introduce meta-continuation

- **A list of continuations outside reset ⟨⟩**

$$\langle \langle 3 + \langle 4 * 2 + 1 \rangle \rangle - 5 \rangle$$

$$\text{current continuation} : [\cdot] + 1$$

$$\text{meta-continuation} : 3 + [\cdot], [\cdot] - 5$$

- **Calling shift0 inside itself captures outer continuations**
  - shift S: leaves reset
  - shift0 S_0: removes reset

$$\langle J[Sk.e] \rangle \longrightarrow \langle e[\lambda x.\langle J[x] \rangle / k] \rangle$$

$$\langle J[S_0 k.e] \rangle \longrightarrow e[\lambda x.\langle J[x] \rangle / k]$$

Example of difference between shift and shift0

Green Area Current continuation (bound to k_1) Orange Area Meta-continuation (bound to k_2)

shift $\langle J[Sk.e]\rangle \longrightarrow \langle e[\lambda x.\langle J[x]\rangle/k]\rangle$

$$\langle (\lambda x.\,1)\,\langle\ (\lambda y.\,2)\ S k_1.\,S k_2.\,(k_1\,0)\ \rangle\rangle$$

Empty meta-continuation

shift0 $\langle J[S_0k.e]\rangle \longrightarrow e[\lambda x.\langle J[x]\rangle/k]$

$$\langle\ (\lambda x.\,1)\,\langle\ (\lambda y.\,2)\ S_0 k_1.\,S_0 k_2.\,(k_1\,0)\ \rangle\ \rangle$$

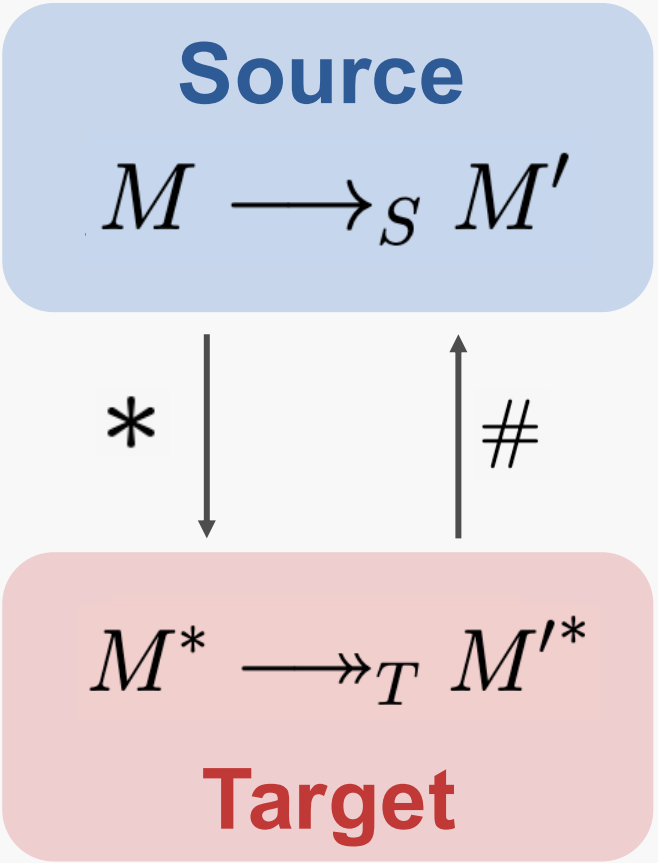
Non-empty meta-continuation

First typed and unified reflection for shift & shift0

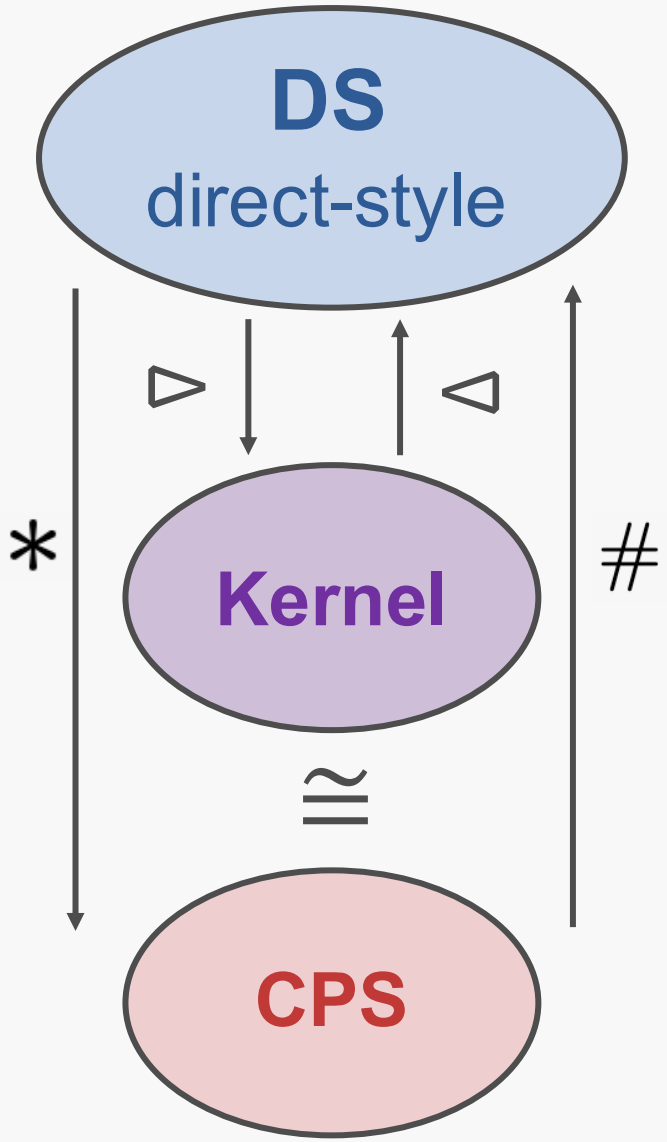
- **Employed 2CPS (two continuations-passing style)**
  - Handle both continuations and meta-continuations
- **Formalization in Agda for both typed and untyped versions**
  - Postulated a few variable-binding properties
- **The Key: annotations for continuations**
  - Manage the respective states of both continuations
  - Identify the location of k to substitute correctly

Four conditions for translations

- **Reflection**
  (1) $M \longrightarrow\!\!\!\!\!\rightarrow_S M^{*\#}$
  (2) $N^{\#*} \equiv_T N$
  (3) $M \longrightarrow_S M'$ implies $M^* \longrightarrow\!\!\!\!\!\rightarrow_T M'^*$
  (4) $N \longrightarrow_T N'$ implies $N^\# \longrightarrow\!\!\!\!\!\rightarrow_S N'^\#$
- **Order isomorphism**
  - Condition (1) to a syntactic identity
  - Strict one-to-one correspondence



Our calculi & Proof strategy



- STLC + let, `shift`, `shift0`, `reset`
- e.g. $f\,(y\,x)$

Reflection

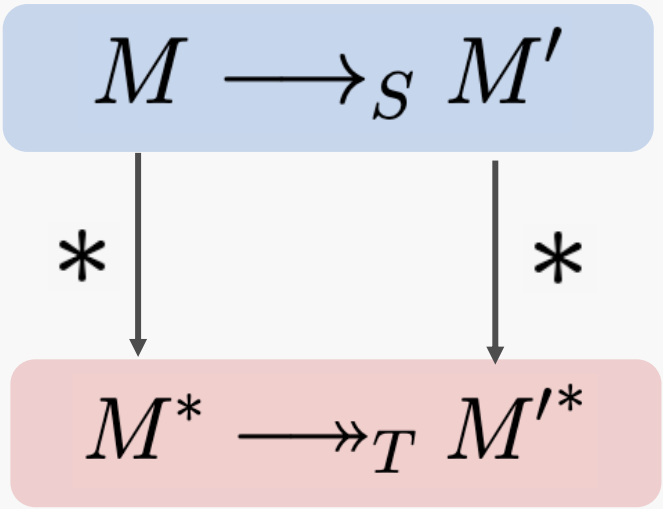
- A-normal form subset of DS calculus
- Identical to CPS calculus
- e.g. $[(\mathsf{let}\ b = y\,x\ \mathsf{in}\ [[f\,b]_k]_g)]_g$

Order isomorphism

- STLC + list abstraction & application
- e.g. $\lambda k.\,\lambda g.\,y\,x\,(\lambda b.\,\lambda g.\,f\,b\,k\,g)\,g$

Overview

- **Order isomorphism between kernel and CPS**
  - Kernel design with the same structure as CPS calculus
- **Reflection between DS and Kernel : Key lemmas**
  - Substitution for continuations
  - Continuation existence
- **Technical challenge in Agda**



Kernel design with the same structure as CPS calculus

CPS

$$
\begin{array}{rlll}
\text{terms} & M_{(\Delta{+}{+}\Theta)} & ::= & \boxed{K_\Delta\, V\, G_\Theta} \mid V\, W\, K_\Delta\, G_\Theta \\
\text{values} & V, W & ::= & x \mid \lambda x.\, \lambda k.\, \lambda g.\, M_k \mid S \mid S_0 \\
\text{continuations} & J_\Delta, K_\Delta & ::= & {}_{(\Delta=k)}\ k \mid {}_{(\Delta=\bullet)}\ k_{id} \mid \lambda x.\, \lambda g.\, M_\Delta \\
\text{meta continuations} & G_\Theta & ::= & {}_{(\Theta=g)}\ g \mid {}_{(\Theta=\Delta)}\ G_\Delta \\
\text{non-empty metaconts} & G_{(\Delta{+}{+}\Theta)} & ::= & K_\Delta :: G_\Theta
\end{array}
$$

Kernel

$$
\begin{array}{rlll}
\text{terms} & M_{(\Delta{+}{+}\Theta)} & ::= & \boxed{G_\Theta[K_\Delta[V]]} \mid G_\Theta[K_\Delta[V\, W]] \\
\text{values} & V, W & ::= & x \mid \lambda x.\, M_k \mid S \mid S_0 \\
\text{contexts} & J_\Delta, K_\Delta & ::= & {}_{(\Delta=k)}\ [\,]_k \mid {}_{(\Delta=\bullet)}\ [\,]_\bullet \mid \underline{\mathsf{let}}\ x = [\,]\ \underline{\mathsf{in}}\ M_\Delta \\
\text{meta contexts} & G_\Theta & ::= & {}_{(\Theta=g)}\ [\,]_g \mid {}_{(\Theta=\Delta)}\ G_\Delta \\
\text{non-empty metacontexts} & G_{(\Delta{+}{+}\Theta)} & ::= & G_\Theta[K_\Delta[\langle [\,] \rangle]]
\end{array}
$$

Key Lemma 1 : Substitution for continuations

- **Commutativity of substitution & translation**
- **Case split by the location of the default k**
  - In the continuation (no delimiters) $1^* = \underline{\lambda} k.\, \underline{\lambda} g.\, (\, k \,\underline{@}\, 1 \,\underline{@}\, g \,)$

$$(M : K_k : G_g)[J_\Delta/k] \equiv M : (K_k[J_\Delta/k]) : G_g$$

  - In the meta-continuation (with delimiters) $\underline{\langle 1 \rangle}^* = \underline{\lambda} k.\, \underline{\lambda} g.\, (\, k_{id} \,\underline{@}\, 1 \,\underline{@}\, (k :: g) \,)$

$$(M : K_\bullet : G_k)[J_\Delta/k] \equiv M : K_\bullet : (G_k[J_\Delta/k])$$

more details later

Substitution for continuations (no delimiter)

DS $h\,((\lambda x.\, f\,(y\,x))\;v) \longrightarrow h\,(f\,(y\;v))$

In continuation

CPS $\lambda k.\,\lambda g.\,(\lambda x.\,\lambda k.\,\lambda g.\,y\,x\,(\lambda b.\,\lambda g.\,f\,b\,k\,g)\,g)\;v\;(\lambda a.\,\lambda g.\,h\,a\,k\,g)\;g$

$\longrightarrow$ $\lambda k.\,\lambda g.\,y\;v\;(\lambda b.\,\lambda g.\,f\,b\;(\lambda a.\,\lambda g.\,h\,a\,k\,g)\;g)\;g$ flat

Kernel $[\,(\text{let } a = (\lambda x.\,[(\text{let } b = y\,x \text{ in } [[f\,b]_k]_g)]_g)\;v \text{ in } [[h\,a]_k]_g)\,]_g$

only v $\not\longrightarrow$ $[\,(\text{let } a = [(\text{let } b = y\;v \text{ in } [[f\,b]_k]_g)]_g \text{ in } [[h\,a]_k]_g)\,]_g$ nested

Substitute into $[\,]_k, [\,]_g$ $\longrightarrow$ $[\,(\text{let } b = y\;v \text{ in } [\,(\text{let } a = f\,b \text{ in } [[h\,a]_k]_g)\,]_g)\,]_g$ flat

Substitution for continuations (with delimiter)

DS $h\,((\lambda x.\,\langle f\,(y\,x)\rangle)\;v) \longrightarrow h\,\langle f\,(y\;v)\rangle$

reset

identity continuation

In meta-continuation

CPS $\lambda k.\,\lambda g.\,(\lambda x.\,\lambda k.\,\lambda g.\,y\,x\,(\lambda b.\,\lambda g.\,f\,b\,k_{id}\,g)\,(k :: g))\;v\;(\lambda a.\,\lambda g.\,h\,a\,k\,g)\;g$

$\longrightarrow$ $\lambda k.\,\lambda g.\,y\;v\;(\lambda b.\,\lambda g.\,f\,b\,k_{id}\,g)\,((\lambda a.\,\lambda g.\,h\,a\,k\,g) :: g)$

Kernel $[\,(\mathsf{let}\;a = (\lambda x.\,[[\langle \mathsf{let}\;b = y\,x\;\mathsf{in}\;[[f\,b]_\bullet]_g\rangle]_k]_g)\;v\;\mathsf{in}\;[[h\,a]_k]_g)\,]_g$

Substitute to $[\,]_k, [\,]_g$

$[\,(\mathsf{let}\;a = [[\langle \mathsf{let}\;b = y\;v\;\mathsf{in}\;[[f\,b]_\bullet]_g\rangle]_k]_g\;\mathsf{in}\;[[h\,a]_k]_g)\,]_g$

Substitution for continuations

In continuation

k in the continuation $(M : K_{k} : G_g)[J_\Delta/k] \equiv M : (K_{k}[J_\Delta/k)] : G_g$

$$h\,((\lambda x.\,(f\,(y\,x)))\,v) \xrightarrow{\beta} h\,(f\,(y\,v))$$

$$\rhd \downarrow \qquad\qquad \rhd \downarrow$$

$$[(\text{let } a = (\lambda x.\,[(\text{let } b = y\,x \text{ in } [[f\,b]_k]_g)]_g)v \text{ in } [[h\,a]_k]_g)]_g \xrightarrow{\beta} [(\text{let } b = y\,v \text{ in } [(\text{let } a = f\,b \text{ in } [[h\,a]_k]_g)]_g)]_g$$

In meta-continuation

k in the meta-continuation $(M : K_\bullet : G_{k})[J_\Delta/k] \equiv M : K_\bullet : (G_{k}[J_\Delta/k])$

$$h\,((\lambda x.\,\langle f\,(y\,x)\rangle)\,v) \xrightarrow{\beta} h\,\langle f\,(y\,v)\rangle$$

$$\rhd \downarrow \qquad\qquad \rhd \downarrow$$

$$[(\text{let } a = (\lambda x.\,[[\langle(\text{let } b = y\,x \text{ in } [[f\,b]_\bullet]_g)\rangle]_k]_g)\,v \text{ in } [[h\,a]_k]_g)]_g \xrightarrow{\beta} [(\text{let } a = \langle(\text{let } b = y\,v \text{ in } [[f\,b]_\bullet]_g)\rangle \text{ in } [[h\,a]_k]_g)]_g$$

Key Lemma 1 : Substitution for continuations

- **Commutativity of substitution & translation**
- **Case split by the location of the default k**
  - In the continuation (no delimiters) $1^* = \underline{\lambda} k.\, \underline{\lambda} g.\, (\, k \,\underline{@}\, 1 \,\underline{@}\, g \,)$

$$(M : K_k : G_g)[J_\Delta/k] \equiv M : (K_k[J_\Delta/k]) : G_g$$

Annotations

  - In the meta-continuation (with delimiters) $\underline{\langle 1 \rangle}^* = \underline{\lambda} k.\, \underline{\lambda} g.\, (\, k_{id} \,\underline{@}\, 1 \,\underline{@}\, (k :: g)\,)$

$$(M : K_\bullet : G_k)[J_\Delta/k] \equiv M : K_\bullet : (G_k[J_\Delta/k])$$

Key lemma 2: Continuation existence

- **Important lemma on `shift` and `shift0`**
  - Pushing the context into continuation (for non-value P)

$$J[P] : K_\Delta : G_\Theta = P : \hat{J}_\Delta : G_\Theta$$

  - Returning the context back (for value V)

$$V : \hat{J}_\Delta : G_\Theta \longrightarrow\!\!\!\!\rightarrow_{\lambda^{\triangleright}_{cs_0}} J[V] : K_\Delta : G_\Theta$$

- **Extend with meta-continuation**
  - Meta-continuation remains the same
  - Prove both cases using this single lemma

Technical challenge in Agda

- **Same structure of the kernel and CPS calculi**
  - Different notations
    - Kernel: binds only x
    - CPS: binds x, k, and g

$$\underline{\lambda} x.\ [[x \mathbin{\underline{@}} 1]_k]_g$$

$$\underline{\lambda} x.\ \underline{\lambda} k.\ \underline{\lambda} g.\ x \mathbin{\underline{@}} 1 \mathbin{\underline{@}} k \mathbin{\underline{@}} g$$

  - Treat k and g as constants (not bound variables)

- **Required standard postulates in PHOAS**
  - Functional extensionality $f(x) = g(x) \implies f = g$
  - Identity substitution $M_\Delta[x/x] \equiv M_\Delta$

Conclusion and Future Work

- **Goal: A typed and unified reflection for `shift` and `shift0`**
- **Contribution**
  - Employed 2CPS with annotations
  - Formalized in Agda for both typed and untyped versions
- **Key lemma: substitution for continuations**
- **Next step: extension to deep handler**
- **Future work: all four operator framework & shallow handler**