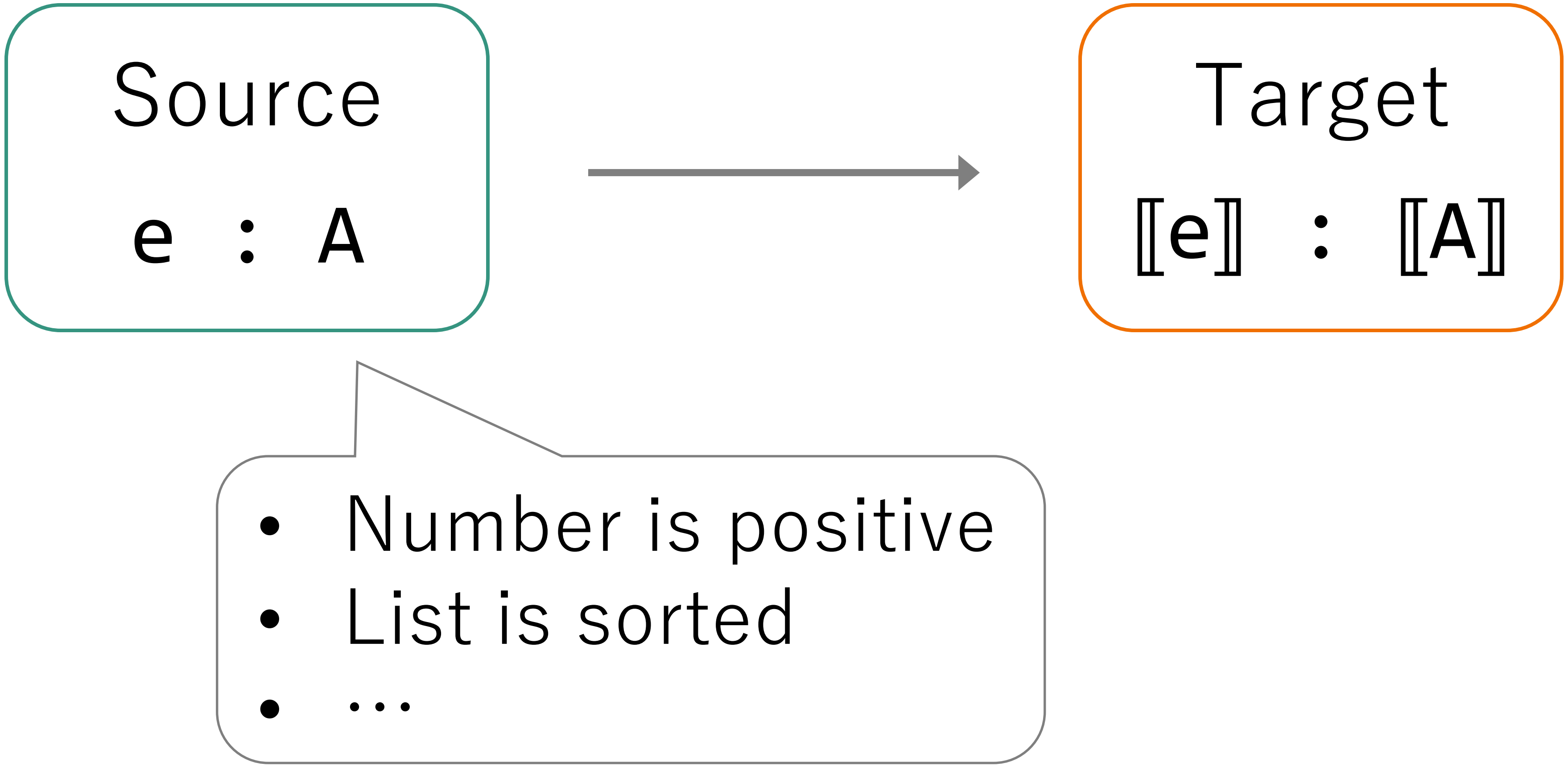
One-Pass CPS Translation of Dependent Types

Youyou Cong (Tokyo Institute of Technology)

Dependent type preserving compilation



Results about (call-by-name) CPS

Barthe+ ’99	CPS of Π is **possible**
Barthe & Uustalu ’02	CPS of Σ is **not possible**
Bowman+ ’18	CPS of Σ is **not not possible**
This work	One-pass CPS of Σ is possible?

One-pass: yields no administrative redexes

Naive CPS of `snd` (simply typed)

`snd e : B` where `e : A × B`

↓

$[\![B]\!] \to \bot$

$\lambda k.\ [\![e]\!]\ (\lambda v.\ \mathtt{snd}\ v\ k) : ([\![B]\!] \to \bot) \to \bot$

$([\![B]\!] \to \bot) \to \bot$

Naive CPS of **snd** (dependently typed)

$\texttt{snd e : B[fst e / x]}$ where $\texttt{e : } \Sigma \texttt{x : A. B}$

$[\![\texttt{B}]\!][[\![\texttt{fst e}]\!] / \texttt{x}] \to \bot$

$\lambda \texttt{k}. [\![\texttt{e}]\!] \ (\lambda \texttt{v}. \ \texttt{snd v k}) : ([\![\texttt{B}]\!][[\![\texttt{fst e}]\!] / \texttt{x}] \to \bot) \to \bot$

$([\![\texttt{B}]\!][\texttt{fst v} / \texttt{x}] \to \bot) \to \bot$

Proving equivalence

`snd e : B[fst e / x]`   where `e : Σx : A. B`

unique if **e** is pure

`λk. ⟦e⟧ (λv. snd v k) : (⟦B⟧[⟦fst e⟧ / x] → ⊥) → ⊥`

`fst v = fst val-of(⟦e⟧) = fst (⟦e⟧ id) = ⟦fst e⟧`

Solution of Bowman et al. '18

1. Polymorphic answer type

$$e : A$$

↓

$$\lambda\alpha.\ \lambda k.\ e' : \Pi\alpha.(\llbracket A \rrbracket \to \alpha) \to \alpha$$

$$\lambda\alpha.\lambda k.\llbracket e \rrbracket\ \alpha\ (\lambda v.\text{snd}\ v\ \alpha\ k)\ :\ \Pi\alpha.(\llbracket B \rrbracket[\llbracket \text{fst}\ e \rrbracket / x] \to \alpha) \to \alpha$$

$$\text{fst}\ v = \text{fst}\ \text{val-of}(\llbracket e \rrbracket) = \text{fst}\ (\llbracket e \rrbracket\ C\ \textbf{id}) = \llbracket \text{fst}\ e \rrbracket$$

where $C = \Sigma x : \llbracket A \rrbracket.\ \llbracket B \rrbracket$

Solution of Bowman et al. ’18

2. New typing rule

$$\frac{\Gamma \vdash e_1 : \forall\alpha.(A \to \alpha) \to \alpha \qquad \Gamma, v = e_1\ A\ \mathrm{id} \vdash e_2 : B}{\Gamma \vdash e_1\ B\ (\lambda v.\ e_2) : B}$$

$$\lambda\alpha.\lambda k.[\![e]\!]\ \alpha\ (\lambda v.\mathrm{snd}\ v\ \alpha\ k) \ :\ \Pi\alpha.([\![B]\!][[\![\mathrm{fst}\ e]\!] / x] \to \alpha) \to \alpha$$

$$\mathrm{fst}\ v = \mathrm{fst}\ \text{val-of}([\![e]\!]) = \mathrm{fst}\ ([\![e]\!]\ C\ \mathrm{id}) = [\![\mathrm{fst}\ e]\!]$$

where $C = \Sigma x : [\![A]\!].\ [\![B]\!]$

Solution of Bowman et al. ’18

3. New equivalence rule

$$\frac{}{\texttt{e B k} \equiv \texttt{k (e A id)}}$$

$$\lambda\alpha.\lambda k.[\![e]\!]\ \alpha\ (\lambda v.\mathrm{snd}\ v\ \alpha\ k)\ :\ \Pi\alpha.([\![B]\!][[\![\mathrm{fst}\ e]\!]\ /\ x] \to \alpha) \to \alpha$$

$$\mathrm{fst}\ v = \mathrm{fst}\ \mathrm{val\text{-}of}([\![e]\!]) = \mathbf{fst}\ \mathbf{([\![e]\!]\ C\ id)} = \mathbf{[\![fst\ e]\!]}$$

where $C = \Sigma x : [\![A]\!].\ [\![B]\!]$

Type preservation

If $\Gamma \vdash e : A$ in the source,

then $[\![\Gamma]\!] \vdash [\![e]\!] : \Pi\alpha.\ ([\![A]\!] \to \alpha) \to \alpha$ in the target.

Proof: By induction on the typing derivation.

One-pass CPS of **snd** (simply typed)

`snd e : B` where `e : A × B`

↓

`λk. e | λv. snd v k : (⟦B⟧ → ⊥) → ⊥`

"colon" translation

One-pass CPS of **snd** (simply typed)

$\texttt{snd e : B}$ where $\texttt{e : A × B}$

↓

$[\![B]\!] \to \bot$

$\lambda k.\ e \mid \lambda v.\ \texttt{snd v k} : ([\![B]\!] \to \bot) \to \bot$

$([\![B]\!] \to \bot) \to \bot$

One-pass CPS of `snd` (dependently typed)

`snd e : B[fst e / x]` where `e : Σx : A. B`

↓

$[\![B]\!][\,[\![\mathsf{fst}\ e]\!] / x] \to \bot$

$\lambda k.\ e \mid \lambda v.\ \mathsf{snd}\ v\ k : ([\![B]\!][\,[\![\mathsf{fst}\ e]\!] / x] \to \bot) \to \bot$

$([\![B]\!][\mathsf{fst}\ v / x] \to \bot) \to \bot$

Applying Bowman et al.'s solution

1. Polymorphic answer type

$$e : A$$

$$\downarrow$$

$$\lambda\alpha.\ \lambda k.\ e' : \Pi\alpha.(⟦A⟧ \to \alpha) \to \alpha$$

$$\lambda\alpha.\lambda k.e \mid \alpha \mid \lambda v.\mathrm{snd}\ v\ \alpha\ k : \Pi\alpha.(⟦B⟧[⟦\mathrm{fst}\ e⟧ / x] \to \alpha) \to \alpha$$

$$\mathrm{fst}\ v = \mathrm{fst}\ \mathrm{val\text{-}of}(⟦e⟧) = \mathrm{fst}\ (e \mid C \mid \mathrm{id}) = ⟦\mathrm{fst}\ e⟧$$

where $C = \Sigma x : ⟦A⟧.\ ⟦B⟧$

Applying Bowman et al.’s solution

2. New typing rule

$$\frac{\Gamma \vdash e_1 : \forall\alpha.(A \to \alpha) \to \alpha \qquad \Gamma, v = e_1\ A\ id \vdash e_2 : B}{\Gamma \vdash e_1\ B\ (\lambda v.\ e_2) : B}$$

$$\lambda\alpha.\lambda k.e \mid \alpha \mid \lambda v.\mathrm{snd}\ v\ \alpha\ k\ :\ \Pi\alpha.(⟦B⟧[⟦\mathrm{fst}\ e⟧ / x] \to \alpha) \to \alpha$$

$$\mathrm{fst}\ v = \mathrm{fst}\ \text{val-of}(⟦e⟧) = \mathrm{fst}\ (e \mid C \mid id) = ⟦\mathrm{fst}\ e⟧$$

where $C = \Sigma x : ⟦A⟧.\ ⟦B⟧$

Applying Bowman et al.'s solution

2. New typing rule

$$\frac{\Gamma \vdash e_1 : \forall\alpha.(A \to \alpha) \to \alpha \qquad \Gamma, v = e_1\ A\ \mathrm{id} \vdash e_2 : B}{\Gamma \vdash e_1\ B\ (\lambda v.\ e_2) : B}$$

cannot apply

$$\lambda\alpha.\lambda k.e \mid \alpha \mid \lambda v.\mathrm{snd}\ v\ \alpha\ k \;:\; \Pi\alpha.(⟦B⟧[⟦\mathrm{fst}\ e⟧ / x] \to \alpha) \to \alpha$$

$$\mathrm{fst}\ v = \mathrm{fst}\ \text{val-of}(⟦e⟧) = \mathrm{fst}\ (e \mid C \mid \mathrm{id}) = ⟦\mathrm{fst}\ e⟧$$

where $C = \Sigma x : ⟦A⟧.\ ⟦B⟧$

Proposal: definition lambda (cf. Koronkevich+ ’22)

$$e ::= \ldots \mid \lambda x = e.\ e$$

$$\frac{\Gamma \vdash e_1 : A \qquad \Gamma, x = e_1 : A \vdash e_2 : B}{\Gamma \vdash \lambda x = e_1.\ e_2 : (e_1 : A) \to B}$$

$$\lambda\alpha.\lambda k.e \mid \alpha \mid \lambda v = e \mid C \mid id.\mathrm{snd}\ v\ \alpha\ k$$

$$\mathrm{fst}\ v = \mathrm{fst}\ \text{val-of}(\llbracket e \rrbracket) = \mathrm{fst}\ (e \mid C \mid id) = \llbracket \mathrm{fst}\ e \rrbracket$$

where $C = \Sigma x : \llbracket A \rrbracket.\ \llbracket B \rrbracket$

Challenge: type translation depends on inhabitants

```
e : A
```

↓

```
λα. λk. e | α | k : Πα. ((e | ⟦A⟧ | id) : ⟦A⟧ → α) → α
```

Need to translate terms and types in parallel?

Summary

- Goal: one-pass CPS translation of Σ types
- Idea: use definition lambda
- Challenge: type translation depends on inhabitants