Handling Delimited Continuations with Dependent Types (Technical Appendix)

YOUYOU CONG, Ochanomizu University, Japan
KENICHI ASAI, Ochanomizu University, Japan

This is a technical appendix of the authors' ICFP 2018 submission, containing:

- Proofs of metatheoretic properties of $\lambda_\Pi^{s/r}$ (Section 1);
- Detailed discussion of the multi-arity extension (Section 2);
- Complete specification of λ_Π^{k} (Section 3);
- Proofs of type preservation of the CPS translation (Section 4); and
- CPS translation of the multi-ality language (Section 5).

3

1 METATHEORETIC PROPERTIES OF $\lambda_\Pi^{s/r}$

This section proves a seriese of metatheoretic properties of $\lambda_\Pi^{s/r}$, including confluence, preservation, and progress.

1.1 Reduction

We first define parallel reduction, following Takahashi [1995]:

$$\frac{}{\Gamma \vdash t \rhd_p t}\ (\text{P-Refl})$$

$$\frac{\Gamma \vdash A \rhd_p A'}{\Gamma \vdash \Pi\, x : A.\, * \rhd_p \Pi\, x : A'.\, *}\ (\text{P-PiK})$$

$$\frac{\Gamma \vdash e \rhd_p e'}{\Gamma \vdash D\, e \rhd_p D\, e'}\ (\text{P-Data})$$

$$\frac{\Gamma \vdash A \rhd_p A' \quad \Gamma \vdash B \rhd_p B'}{\Gamma \vdash \Pi\, x : A.\, B \rhd_p \Pi\, x : A'.\, B'}\ (\text{P-PiT1})$$

$$\frac{\Gamma \vdash A \rhd_p A' \quad \Gamma \vdash \alpha \rhd_p \alpha' \quad \Gamma \vdash B \rhd_p B' \quad \Gamma \vdash \beta \rhd_p \beta'}{\Gamma \vdash \Pi\, x : A.\, \alpha \,\|\, B \,\|\, \beta \rhd_p \Pi\, x : A'.\, \alpha' \,\|\, B' \,\|\, \beta'}\ (\text{P-PiT2})$$

$$\frac{x = v : A \in \Gamma}{\Gamma \vdash x \rhd_p v}\ (\text{P-VarDelta})$$

$$\frac{\Gamma \vdash e \rhd_p e'}{\Gamma \vdash \lambda x : A.\, e \rhd_p \lambda x : A.\, e'}\ (\text{P-Abs})$$

$$\frac{\Gamma \vdash e \rhd_p e'}{\Gamma \vdash \text{rec}\ f\,(x : A).\, e \rhd_p \text{rec}\ f\,(x : A).\, e'}\ (\text{P-Rec})$$

Authors' addresses: Youyou Cong, Department of Computer Science, Ochanomizu University, Japan, so.yuyu@is.ocha.ac.jp; Kenichi Asai, Department of Computer Science, Ochanomizu University, Japan, asai@is.ocha.ac.jp.

2018. 2475-1421/2018/9-ART3
https://doi.org/10.1145/3234592

$$\frac{\Gamma \vdash e_0 \triangleright_p e_0' \quad \Gamma \vdash e_1 \triangleright_p e_1'}{\Gamma \vdash e_0\ e_1 \triangleright_p e_0'\ e_1'} \text{(P-App)}$$

$$\frac{\Gamma \vdash e_0 \triangleright_p e_0' \quad \Gamma \vdash v_1 \triangleright_p v_1'}{\Gamma \vdash (\lambda x : A.\, e_0)\ v_1 \triangleright_p e_0' [v_1'/x]} \text{(P-AppBeta)}$$

$$\frac{\Gamma \vdash e_0 \triangleright_p e_0' \quad \Gamma \vdash v_1 \triangleright_p v_1'}{\Gamma \vdash (\mathsf{rec}\ f\,(x : A).\, e_0)\ v_1 \triangleright_p e_0' [\mathsf{rec}\ f\,(x : A).\, e_0'/f, v_1'/x]} \text{(P-AppMu)}$$

$$\frac{\Gamma \vdash e_1 \triangleright_p e_1' \quad \Gamma, x : e_1 \vdash e_2 \triangleright_p e_2'}{\Gamma \vdash \mathsf{let}\ x = e_1 : A\ \mathsf{in}\ e_2 \triangleright_p \mathsf{let}\ x = e_1' : A\ \mathsf{in}\ e_2'} \text{(P-Let)}$$

$$\frac{\Gamma \vdash v_1 \triangleright_p v_1' \quad \Gamma, x : v_1 \vdash e_2 \triangleright_p e_2'}{\Gamma \vdash \mathsf{let}\ x = v_1 : A\ \mathsf{in}\ e_2 \triangleright_p e_2' [v_1'/x]} \text{(P-LetZeta)}$$

$$\frac{\Gamma \vdash e \triangleright_p e'}{\Gamma \vdash c_i\ e \triangleright_p c_i\ e'} \text{(P-Const)}$$

$$\frac{\Gamma \vdash e \triangleright_p e' \quad \Gamma \vdash e_i \triangleright_p e_i'}{\begin{array}{l}\Gamma \vdash \mathsf{match}\ e\ \mathsf{as}\ x\ \mathsf{in}\ D\ a\ \mathsf{return}\ P\ \mathsf{with}\ \{c_i\ y_i \to e_i\} \triangleright_p \\ \quad \mathsf{match}\ e'\ \mathsf{as}\ x\ \mathsf{in}\ D\ a\ \mathsf{return}\ P\ \mathsf{with}\ \{c_i\ y_i \to e_i'\}\end{array}} \text{(P-Match)}$$

$$\frac{\Gamma \vdash v \triangleright_p v' \quad \Gamma \vdash e_i \triangleright_p e_i'}{\Gamma \vdash \mathsf{match}\ c_i\ v\ \mathsf{as}\ x\ \mathsf{in}\ D\ a\ \mathsf{return}\ P\ \mathsf{with}\ \{c_i\ y_i \to e_i\} \triangleright_p e_i' [v'/y_i]} \text{(P-MatchIota)}$$

$$\frac{\Gamma \vdash e \triangleright_p e'}{\Gamma \vdash \mathcal{S}k : A.\, e \triangleright_p \mathcal{S}k : A.\, e'} \text{(P-Shift)}$$

$$\frac{\Gamma \vdash e \triangleright_p e'}{\Gamma \vdash \langle e \rangle \triangleright_p \langle e' \rangle} \text{(P-Reset)}$$

$$\frac{\Gamma \vdash F[x] \triangleright_p F'[x] \quad \Gamma \vdash e \triangleright_p e'}{\Gamma \vdash \langle F[\mathcal{S}k : A \to \alpha.\, e] \rangle \triangleright_p \langle e' [\lambda x : A.\, F'[x]/k] \rangle} \text{(P-ResetS)}$$

$$\frac{\Gamma \vdash v \triangleright_p v'}{\Gamma \vdash \langle v \rangle \triangleright_p v'} \text{(P-ResetR)}$$

Lemma 1.1 (Substitution and Parallel Reduction). *If* $\Gamma \vdash t \triangleright_p t'$ *and* $\Gamma \vdash v \triangleright_p v'$, *then* $\Gamma \vdash t\,[v/x] \triangleright_p t'\,[v'/x]$.

The proof is by induction on the derivation of $\Gamma \vdash t \triangleright_p t'$.

Case (Refl)

Sub-Case $t = x$

Our goal is to show:

$$\Gamma \vdash x\,[v/x] \triangleright_p x\,[v'/x]$$

This immediately follows by the premise.

Sub-Case $t = y$ where $y \neq x$
Trivial.
Case (AppRedBeta)
Our goal is to show:

$$\Gamma \vdash ((\lambda x : A.\, e_0)\ v_1)\,[v/x'] \triangleright_p (e_0'\,[v_1'/x])\,[v'/x']$$

By the induction hypothesis, we have

$$\Gamma \vdash e_0\,[v/x'] \triangleright_p e_0'\,[v'/x']$$

and

$$\Gamma \vdash v_1\,[v/x'] \triangleright_p v_1'\,[v'/x']$$

The goal follows by (AppRedBeta).

Lemma 1.2 (Confluence of $\triangleright_p$). *If $\Gamma \vdash e \triangleright_p e_1$ and $\Gamma \vdash e \triangleright_p e_2$, then there exists some e' such that $\Gamma \vdash e_1 \triangleright_p e'$ and $\Gamma \vdash e_2 \triangleright_p e'$.*

The proof is by induction on the structure of e.
Case $e = x$
Sub-Case One reduction is (P-Refl)
Trivial.
Sub-Case Both reduction is (P-VarDelta)
This case is also trivial.
Case $e = e_0\ e_1$
Sub-Case One reduction is (P-Refl)
Trivial.
Sub-Case Both reductions are (P-App)
In this case, we have

$$e_0\ e_1 \triangleright_p e_{00}\ e_{10}$$

and

$$e_0\ e_1 \triangleright_p e_{01}\ e_{11}$$

By the induction hypothesis, there exists some e_0' such that $e_{00} \triangleright_p e_0'$ and $e_{01} \triangleright_p e_0'$. We have a similar e_1'. By (P-App), we obtain $e_{00}\ e_{10} \triangleright_p e_0'\ e_1'$ and $e_{01}\ e_{11} \triangleright_p e_0'\ e_1'$ as desired.
Sub-Case One reduction is (P-AppBeta)
In this case, e must have the form $(\lambda x : A.\, e_0)\ v_1$.
Sub-Sub-Case The other reduction is (P-App)
We have

$$(\lambda x : A.\, e_0)\ v_1 \triangleright_p e_{00}\,[v_{10}/x]$$

and

$$(\lambda x : A.\, e_0)\ v_1 \triangleright_p (\lambda x : A.\, e_{01})\ v_{11}$$

By the induction hypothesis, there is a e_0' such that $e_{00} \triangleright_p e_0'$ and $e_{01} \triangleright_p e_0'$. We also have a similar v_1'. By Lemma 1.1, we have $e_0\,[v_1/x] \triangleright_p e_0'\,[v_1'/x]$. By (P-AppBeta), we also have $(\lambda x : A.\, e_{01})\ v_{11} \triangleright_p e_0'\,[v_1'/x]$. These imply the goal.
Sub-Sub-Case The other reduction is (P-AppBeta)
We have

$$(\lambda x : A.\, e_0)\ v_1 \triangleright_p e_{00}\,[v_{10}/x]$$

and

$$(\lambda x : A.\, e_0)\ v_1 \ \triangleright_p\ e_{01}\,[v_{11}/x]$$

By the induction hypothesis, there is a e'_0 such that $e_{00} \triangleright_p e'_0$ and $e_{01} \triangleright_p e'_0$. We also have a similar v'_1. By Lemma 1.1, we have $e_{00}\,[v_{10}/x] \triangleright_p e'_0\,[v'_1/x]$ and $e_{01}\,[v_{11}/x] \triangleright_p e'_0\,[v'_1/x]$. The goal follows by (P-AppBeta).

Sub-Case One reduction is (P-AppMu)

This case is analogous to the previous case.

Case $e = \text{let } x = e_1 : A \text{ in } e_2$

Sub-Case One reduction is (P-Refl)

Trivial.

Sub-Case Both reductions are (P-Let)

In this case, we have

$$\text{let } x = e_1 : A \text{ in } e_2 \ \triangleright_p\ \text{let } x = e_{10} : A \text{ in } e_{20}$$

and

$$\text{let } x = e_1 : A \text{ in } e_2 \ \triangleright_p\ \text{let } x = e_{11} : A \text{ in } e_{21}$$

By the induction hypothesis, there is a e'_1 such that $e_{10} \triangleright_p e'_1$ and $e_{11} \triangleright_p e'_1$. We also have a similar e'_2. The goal follows by (P-Let).

Sub-Case One reduction is (P-LetZeta)

In this case, e must have the form $\text{let } x = v_1 : A \text{ in } e_2$.

Sub-Sub-Case The other reduction is (P-Let)

We have

$$\text{let } x = v_1 : A \text{ in } e_2 \ \triangleright_p\ e_{20}\,[v_{10}/x]$$

and

$$\text{let } x = v_1 : A \text{ in } e_2 \ \triangleright_p\ \text{let } x = v_{11} : A \text{ in } e_{21}$$

By the induction hypothesis, there is a e'_1 such that $e_{10} \triangleright_p e'_1$ and $e_{11} \triangleright_p e'_1$. We also have a similar e'_2. By Lemma 1.1, we have $e_2\,[v_1/x] \triangleright_p e'_2\,[v'_1/x]$. By (P-LetZeta), we also have $\text{let } x = v_{11} : A \text{ in } e_{21} \triangleright_p e'_2\,[v'_1/x]$. These imply the goal.

Sub-Sub-Case The other reduction is (P-LetZeta)

$$\text{let } x = v_1 \text{ in } Ae_2 \ \triangleright_p\ e_{20}\,[v_{10}/x]$$

and

$$\text{let } x = v_1 \text{ in } Ae_2 \ \triangleright_p\ e_{21}\,[v_{11}/x]$$

By the induction hypothesis, there is a e'_1 such that $e_{10} \triangleright_p e'_1$ and $e_{11} \triangleright_p e'_1$. We also have a similar e'_2. By Lemma 1.1, we have $e_{20}\,[v_{10}/x] \triangleright_p e'_2\,[v'_1/x]$ and $e_{21}\,[v_{11}/x] \triangleright_p e'_2\,[v'_1/x]$. These imply the goal.

Case $e = \text{match } e \text{ as } x \text{ in } D\ a \text{ return } P \text{ with } \{c_i\ y_i \to e_i\}$

Sub-Case One reduction is (P-Refl)

Trivial.

Sub-Case Both reduction is (P-Match)

We have

$$\text{match } e \text{ as } x \text{ in } D\ a \text{ return } P \text{ with } \{c_i\ y_i \to e_i\} \ \triangleright_p\ \text{match } e_0 \text{ as } x \text{ in } D\ a \text{ return } P \text{ with } \{c_i\ y_i \to e_{i0}\}$$

and

$$\text{match } e \text{ as } x \text{ in } D\ a \text{ return } P \text{ with } \{c_i\ y_i \to e_i\} \ \triangleright_p\ \text{match } e_1 \text{ as } x \text{ in } D\ a \text{ return } P \text{ with } \{c_i\ y_i \to e_{i1}\}$$

By the induction hypothesis, there is a e' such that $\mathsf{e}_0 \rhd_p \mathsf{e}'$ and $\mathsf{e}_1 \rhd_p \mathsf{e}'$. We have a similar e_i'. The goal follows by (P-Match).

Sub-Case One reduction is (P-MatchIota)

In this case, e must have the form $\mathsf{match}\ \mathsf{c_i}\ \mathsf{v}\ \mathsf{as}\ \mathsf{x}\ \mathsf{in}\ \mathsf{D}\ \mathsf{a}\ \mathsf{return}\ \mathsf{P}\ \mathsf{with}\ \{\mathsf{c_i}\ \mathsf{y_i} \rightarrow \mathsf{e_i}\}$.

Sub-Sub-Case The other reduction is (P-Match)

We have

$$\mathsf{match}\ \mathsf{c_i}\ \mathsf{v}\ \mathsf{as}\ \mathsf{x}\ \mathsf{in}\ \mathsf{D}\ \mathsf{a}\ \mathsf{return}\ \mathsf{P}\ \mathsf{with}\ \{\mathsf{c_i}\ \mathsf{y_i} \rightarrow \mathsf{e_i}\} \rhd_p \mathsf{e_{i0}}\ [\mathsf{v_0}/\mathsf{y_i}]$$

and

$$\mathsf{match}\ \mathsf{c_i}\ \mathsf{v}\ \mathsf{as}\ \mathsf{x}\ \mathsf{in}\ \mathsf{D}\ \mathsf{a}\ \mathsf{return}\ \mathsf{P}\ \mathsf{with}\ \{\mathsf{c_i}\ \mathsf{y_i} \rightarrow \mathsf{e_i}\} \rhd_p \mathsf{match}\ \mathsf{c_i}\ \mathsf{v_1}\ \mathsf{as}\ \mathsf{x}\ \mathsf{in}\ \mathsf{D}\ \mathsf{a}\ \mathsf{return}\ \mathsf{P}\ \mathsf{with}\ \{\mathsf{c_i}\ \mathsf{y_i} \rightarrow \mathsf{e_{i1}}\}$$

By the induction hypothesis, there is a v' such that $\mathsf{v_0} \rhd_p \mathsf{v}'$ and $\mathsf{v_1} \rhd_p \mathsf{v}'$. We have a similar e_i'. By Lemma 1.1, we have $\mathsf{e_{i0}}\ [\mathsf{v_0}/\mathsf{y_i}] \rhd_p \mathsf{e}_i'\ [\mathsf{v}'/\mathsf{y_i}]$. By (P-MatchIota), we also have $\mathsf{match}\ \mathsf{c_i}\ \mathsf{v_1}\ \mathsf{as}\ \mathsf{x}\ \mathsf{in}\ \mathsf{D}\ \mathsf{a}\ \mathsf{return}\ \mathsf{P}\ \mathsf{with}\ \{\mathsf{c_i}\ \mathsf{y_i} \rightarrow \mathsf{e_{i1}}\} \rhd_p \mathsf{e}_i'\ [\mathsf{v}'/\mathsf{y_i}]$. These imply the goal.

Sub-Sub-Case The other reduction is (P-MatchIota)

We have

$$\mathsf{match}\ \mathsf{c_i}\ \mathsf{v}\ \mathsf{as}\ \mathsf{x}\ \mathsf{in}\ \mathsf{D}\ \mathsf{a}\ \mathsf{return}\ \mathsf{P}\ \mathsf{with}\ \{\mathsf{c_i}\ \mathsf{y_i} \rightarrow \mathsf{e_i}\} \rhd_p \mathsf{e_{i0}}\ [\mathsf{v_0}/\mathsf{y_i}]$$

and

$$\mathsf{match}\ \mathsf{c_i}\ \mathsf{v}\ \mathsf{as}\ \mathsf{x}\ \mathsf{in}\ \mathsf{D}\ \mathsf{a}\ \mathsf{return}\ \mathsf{P}\ \mathsf{with}\ \{\mathsf{c_i}\ \mathsf{y_i} \rightarrow \mathsf{e_i}\} \rhd_p \mathsf{e_{i1}}\ [\mathsf{v_1}/\mathsf{y_i}]$$

By the induction hypothesis, there is a v' such that $\mathsf{v_0} \rhd_p \mathsf{v}'$ and $\mathsf{v_1} \rhd_p \mathsf{v}'$. We have a similar e_i'. By Lemma 1.1, we have $\mathsf{e_{i0}}\ [\mathsf{v_0}/\mathsf{y_i}] \rhd_p \mathsf{e}_i'\ [\mathsf{v}'/\mathsf{y_i}]$ and $\mathsf{e_{i1}}\ [\mathsf{v_1}/\mathsf{y_i}] \rhd_p \mathsf{e}_i'\ [\mathsf{v}'/\mathsf{y_i}]$. These imply the goal.

Case $\mathsf{e} = \langle \mathsf{e} \rangle$

Sub-Case One reduction is (P-Refl)

Trivial.

Sub-Case Both reductions are (P-Reset)

We have

$$\langle \mathsf{e} \rangle \rhd_p \langle \mathsf{e_0} \rangle$$

and

$$\langle \mathsf{e} \rangle \rhd_p \langle \mathsf{e_1} \rangle$$

By the induction hypothesis, there is a e' such that $\mathsf{e_0} \rhd_p \mathsf{e}'$ and $\mathsf{e_1} \rhd_p \mathsf{e}'$. The goal follows by (P-Reset).

Sub-Case One reduction is (P-ResetS)

In this case, e must have the form $\langle \mathsf{F}[\mathcal{S}\mathsf{k} : \mathsf{A} \rightarrow \alpha.\, \mathsf{e}] \rangle$.

Sub-Sub-Case The other reduction is (P-Reset)

We have

$$\langle \mathsf{F}[\mathcal{S}\mathsf{k} : \mathsf{A} \rightarrow \alpha.\, \mathsf{e}] \rangle \rhd_p \langle \mathsf{e_0}\ [\lambda\, \mathsf{x} : \mathsf{A}.\, \langle \mathsf{F_0}[\mathsf{x}] \rangle / \mathsf{k}] \rangle$$

and

$$\langle \mathsf{F}[\mathcal{S}\mathsf{k} : \mathsf{A} \rightarrow \alpha.\, \mathsf{e}] \rangle \rhd_p \langle \mathsf{F_1}[\mathcal{S}\mathsf{k} : \mathsf{A} \rightarrow \alpha.\, \mathsf{e_1}] \rangle$$

By the induction hypothesis, there is a F' such that $\mathsf{F_0}[\mathsf{x}] \rhd_p \mathsf{F}'[\mathsf{x}]$ and $\mathsf{F_1}[\mathsf{x}] \rhd_p \mathsf{F}'[\mathsf{x}]$. We have a similar e'. By Lemma 1.1, we have $\langle \mathsf{e_0}\ [\lambda\, \mathsf{x} : \mathsf{A}.\, \langle \mathsf{F_0}[\mathsf{x}] \rangle / \mathsf{k}] \rangle \rhd_p \langle \mathsf{e}'\ [\lambda\, \mathsf{x} : \mathsf{A}.\, \langle \mathsf{F}'[\mathsf{x}] \rangle / \mathsf{k}] \rangle$ (note that $\mathsf{F}[\mathsf{e}] = \mathsf{F}[\mathsf{x}]\ [\mathsf{e}/\mathsf{x}]$). By (P-ResetS), we also have $\langle \mathsf{F_1}[\mathcal{S}\mathsf{k} : \mathsf{A} \rightarrow \alpha.\, \mathsf{e_1}] \rangle \rhd_p \langle \mathsf{e}'\ [\lambda\, \mathsf{x} : \mathsf{A}.\, \langle \mathsf{F}'[\mathsf{x}] \rangle / \mathsf{k}] \rangle$. These imply the goal.

Sub-Case The other reduction is (P-ResetS)

We have

$$\langle F[\mathcal{S}k : A \to \alpha.\, e]\rangle \rhd_p \langle e_0 [\lambda x : A.\, \langle F_0[x]\rangle/k]\rangle$$

and

$$\langle F[\mathcal{S}k : A \to \alpha.\, e]\rangle \rhd_p \langle e_1 [\lambda x : A.\, \langle F_1[x]\rangle/k]\rangle$$

By the induction hypothesis, there is a F' such that $F_0[x] \rhd_p F'[x]$ and $F_1[x] \rhd_p F'[x]$. We have a similar e'. By Lemma 1.1, we have $\langle e_0 [\lambda x : A.\, \langle F_0[x]\rangle/k]\rangle \rhd_p \langle e' [\lambda x : A.\, \langle F'[x]\rangle/k]\rangle$ and $\langle e_1 [\lambda x : A.\, \langle F_1[x]\rangle/k]\rangle \rhd_p \langle e' [\lambda x : A.\, \langle F'[x]\rangle/k]\rangle$. This implies the goal.

Sub-Case One reduction is (P-ResetR)

In this case, e must have the form $\langle v\rangle$.

Sub-Sub-Case The other reduction is (P-Reset)

We have

$$\langle v\rangle \rhd_p v_0$$

and

$$\langle v\rangle \rhd_p \langle v_1\rangle$$

By the induction hypothesis, there is a v' such that $v_0 \rhd_p v'$ and $v_1 \rhd_p v'$. The goal follows by (P-ResetR).

Sub-Sub-Case The other reduction is (P-ResetR)

We have

$$\langle v\rangle \rhd_p v_0$$

and

$$\langle v\rangle \rhd_p v_1$$

By the induction hypothesis, there is a v' such that $v_0 \rhd_p v'$ and $v_1 \rhd_p v'$. This implies the goal.

Lemma 1.3 (Confluence of $\rhd^\star$). *If* $\Gamma \vdash e \rhd^\star e_1$ *and* $\Gamma \vdash e \rhd^\star e_2$*, then there exists some* e' *such that* $\Gamma \vdash e_1 \rhd^\star e'$ *and* $\Gamma \vdash e_2 \rhd^\star e'$*.*

This is the consequence of the confluence lemma (Lemma 1.2); note that $\rhd^\star$ is the transitive closure of $\rhd_p$.

Lemma 1.4 (Transitivity of Equivalence). *If* $t \equiv t'$ *and* $t' \equiv t''$*, then* $t \equiv t''$*.*

Suppose we have (i) $t \equiv t''$ where $t \rhd^\star t_0$ and $t \rhd^\star t_0$; and (ii) $t' \equiv t''$ where $t' \rhd^\star t_1$ and $t' \rhd^\star t_1$. By Lemma 1.3, there is a t_2 such that $t_0 \rhd_p t_2$ and $t_1 \rhd_p t_2$. This implies $t \rhd^\star t''$.

1.2 Substitution

Lemma 1.5 (Substitution).

(1) *If* $\vdash \Gamma, x : A', \Gamma'$ *and* $\Gamma \vdash_p e' : A'$*, then* $\vdash \Gamma, \Gamma' [e'/x]$.
(2) *If* $\Gamma, x : A', \Gamma' \vdash A : *$ *and* $\Gamma \vdash_p e' : A'$*, then* $\Gamma, \Gamma' [e'/x] \vdash A [e'/x] : *$.
(3) *If* $\Gamma, x : A', \Gamma' \vdash_p e : A$ *and* $\Gamma \vdash_p e' : A'$*, then* $\Gamma, \Gamma' [e'/x] \vdash_p e [e'/x] : A [e'/x]$.
(4) *If* $\Gamma, x : A', \Gamma'; \alpha \vdash e : B; \beta$ *and* $\Gamma \vdash_p e' : A'$*,*
then $\Gamma, \Gamma' [e'/x]; \alpha [e'/x] \vdash e [e'/x] : B [e'/x]; \beta [e'/x]$.

The proof is by mutual induction on the derivation of A and e.

Case Part 1

Sub-Case (Empty) Immediate.
Sub-Case (Extend1) Our goal is to show:

$$\vdash \Gamma, \Gamma'\,[e'/x'], x : A\,[e'/x']$$

By Parts 1 and 2 of the induction hypothesis, we have

$$\vdash \Gamma, \Gamma'\,[e'/x']$$

and

$$\Gamma, \Gamma'\,[e'/x'] \vdash A\,[e'/x'] : *$$

The goal follows by (Extend1).
Sub-Case (Extend2) Our goal is to show:

$$\vdash \Gamma, \Gamma'\,[e'/x'], x = e\,[e'/x'] : A\,[e'/x']$$

By Parts 1 and 3 of the induction hypothesis, we have

$$\vdash \Gamma, \Gamma'\,[e'/x]$$

and

$$\Gamma, \Gamma'\,[e'/x] \vdash_p e\,[e'/x] : A\,[e'/x]$$

The goal follows by (Extend2).
Case Part 2
Sub-Case (Data) Our goal is to show:

$$\Gamma, \Gamma'\,[e'/x] \vdash (D\ e)\,[e'/x] : *$$

By Part 3 of the induction hypothesis, we have

$$\Gamma, \Gamma'\,[e'/x] \vdash_p e\,[e'/x] : A\,[e'/x]$$

The goal follows by (Data).
Case Part 3
Sub-Case (Var)
Sub-Sub-Case $e = x$ Our goal is to show:

$$\Gamma, \Gamma'\,[e'/x] \vdash_p x\,[e'/x] : A\,[e'/x]$$

By the definition of substitution and well-formed typing environments, the goal reduces to:

$$\Gamma, \Gamma'\,[e'/x] \vdash_p e' : A$$

which follows by the premise.
Sub-Sub-Case $e = y$ where $y \neq x$ Our goal is to show:

$$\Gamma, \Gamma'\,[e'/x] \vdash_p y\,[e'/x] : A\,[e'/x]$$

By the induction hypothesis, we have either of the following:

$$\Gamma, \Gamma'\,[e'/x] \vdash A\,[e'/x] : *$$

$$\Gamma, \Gamma'\,[e'/x] \vdash_p e\,[e'/x] : A\,[e'/x]$$

The goal follows by (Var).

Sub-Case (App1) Our goal is to show:

$$\Gamma, \Gamma' [e'/x'] \vdash_p (e_0\ e_1) [e'/x'] : (B [e_1/x]) [e'/x']$$

By the induction hypothesis, we have

$$\Gamma, \Gamma' [e'/x'] \vdash_p e_0 [e'/x'] : (\Pi\, x : A.\ B) [e'/x']$$

and

$$\Gamma, \Gamma' [e'/x'] \vdash_p e_1 [e'/x'] : A [e'/x']$$

By the definition of substitution, we have

$$(e_0\ e_1) [e'/x'] = (e_0 [e'/x']) (e_1 [e'/x'])$$

We also know that

$$(B [e_1/x]) [e'/x'] = (B [e_1 [e'/x]/x]) [e'/x']$$

These imply the goal.

Sub-Case (Reset2) Our goal is to show:

$$\Gamma, \Gamma' [e'/x] \vdash_p \langle e \rangle [e'/x] : A [e'/x]$$

By Part 4 of the induction hypothesis, we have

$$\Gamma, \Gamma' [e'/x]; B [e'/x] \vdash e : B [e'/x]; A [e'/x]$$

The goal follows by (Reset2).

Case Part 4

Sub-Case (App5) Our goal is to show:

$$\Gamma, \Gamma' [e'/x']; (\alpha [e_1/x]) [e'/x'] \vdash (e_0\ e_1) [e'/x'] : (B [e_1/x]) [e'/x']; (\beta [e_1/x]) [e'/x']$$

By Part 3 of the induction hypothesis, we have

$$\Gamma, \Gamma' [e'/x'] \vdash_p e_0 [e'/x'] : (\Pi\, x : A.\ \alpha \parallel B \parallel \beta) [e'/x']$$

and

$$\Gamma, \Gamma' [e'/x'] \vdash_p e_1 [e'/x'] : A [e'/x']$$

The goal follows by (App5).

Sub-Case (Exp) Our goal is to show:

$$\Gamma, \Gamma' [e'/x']; \alpha [e'/x'] \vdash e [e'/x'] : A [e'/x']; \alpha [e'/x']$$

By Parts 3 and 2 of the induction hypothesis, we have

$$\Gamma, \Gamma' [e'/x'] \vdash_p e [e'/x'] : A [e'/x']$$

$$\Gamma, \Gamma' [e'/x'] \vdash \alpha [e'/x'] : *$$

The goal follows by (Exp).

1.3 Regularity

Lemma 1.6 (Context Regularity).

(1) *If* $\Gamma \vdash \kappa$ *then* $\vdash \Gamma$.
(2) *If* $\Gamma \vdash A : *$ *then* $\vdash \Gamma$.
(3) *If* $\Gamma \vdash_p e : A$ *or* $\Gamma; \alpha \vdash e : A; \beta$ *then* $\vdash \Gamma$.

The proof is by mutual induction on the derivation of κ, A, and e. We show some representative cases:

Case Part 1
Sub-Case (Star) Trivial.
Sub-Case (PiK) The goal follows by Part 2 of the induction hypothesis.
Case Part 2
Sub-Case (Data) The goal follows by Part 3 of the induction hypothesis.
Sub-Case (Pi1) The goal follows by the induction hypothesis.
Case Part 3
Sub-Case (Var) Trivial.
Sub-Case (Conv) The goal follows by the induction hypothesis.

Lemma 1.7 (Context Inversion). *If* $\vdash \Gamma$ *and* $x : A \in \Gamma$ *or* $x = e \in \Gamma$, *then* $\Gamma \vdash A : *$.

The proof is by induction on the derivation of Γ.

Case (Extend1)
Sub-Case $\Gamma = \Gamma', x : A$
The goal follows by the premise of (Extend1).
Sub-Case $\Gamma = \Gamma', y : B$
The goal follows by the induction hypothesis applied to Γ'.
Case (Extend2)
Sub-Case $\Gamma = \Gamma', x = e : A$
The goal follows by the premise of (Extend2).
Sub-Case $\Gamma = \Gamma', y : B$
The goal follows by the induction hypothesis applied to Γ'.

Lemma 1.8 (Regularity).

(1) *If* $\Gamma \vdash_p e : A$, *then* $\Gamma \vdash A : *$.
(2) *If* $\Gamma; \alpha \vdash e : A; \beta$, *then* $\Gamma \vdash A : *$.

The proof is by mutual induction on the derivation of e.

Case Part 1
Sub-Case (Var)
The goal follow by the context inversion lemma (Lemma 1.7).
Sub-Case (Abs1)
By Part 1 of the induction hypothesis, we have $\Gamma, x : A \vdash B : *$. By context regularity (Lemma 1.6), we also have $\vdash \Gamma, x : A$, and by context inversion (Lemma 1.7), we obtain $\Gamma \vdash A : *$. By (Pi1), we can conclude that $\Gamma \vdash \Pi\, x : A.\, B : *$.
Sub-Case (Let1)
By Part 1 of the induction hypothesis, we have $\Gamma, x = e_1 : A \vdash B : *$. The substitution lemma (Lemma 1.5) gives us $\Gamma \vdash B\,[e_1/x] : *$, which is what we want.
Case Part 2

Sub-Case (Shift)

By the typing rule, we have $\Gamma, k : A \to \alpha;\ B \vdash e : B;\ \beta$. The context regularity lemma (Lemma 1.6) gives us $\vdash \Gamma, k : A \to \alpha$, and by applying the context inversion lemma (Lemma 1.7), we obtain $\Gamma \vdash A \to \alpha : *$, which implies $\Gamma \vdash A : *$ and $\Gamma \vdash \alpha : *$. Together with the premise $\Gamma \vdash \beta : *$, we obtain the expected property.

Sub-Case (Exp)

By Part 1 of the induction hypothesis, we have $\Gamma \vdash A : *$. The premise of (Exp) also gives us $\Gamma \vdash \alpha : *$. These complete the proof.

1.4 Preservation

Lemma 1.9 (Injectivity).

(1) *If* $\Pi\, x : A.\, B \equiv \Pi\, x : A'.\, B'$, *then* $A \equiv A'$ *and* $A' \equiv B'$.
(2) *If* $D\ e \equiv D\ e'$, *then* $e \equiv e'$.

Lemma 1.10 (Inversion for Abstraction). *If* $\Gamma \vdash_p \lambda x : A.\, e : C$, *then either (1) or (2) holds.*

(1) $C \equiv \Pi\, x : A.\, B$ *for some* B, *and* $\Gamma, x : A \vdash_p e : B$.
(2) $C \equiv \Pi\, x : A.\, \alpha \parallel B \parallel \beta$ *for some* B, α, *and* β, *and* $\Gamma, x : A;\ \alpha \vdash e : B;\ \beta$.

The proof is by induction on the derivation of $\lambda x : A.\, e$. We only show the proof of Part 1; Part 2 can be proven in the exactly same way.

Case Part 1

Sub-Case (Abs1)

Immediate.

Sub-Case (Conv1)

We have $\Gamma \vdash_p \lambda x : A.\, e : C'$ where $C \equiv C'$. The induction hypothesis gives us $\equiv C\Pi\, x : A.\, B$ and $\Gamma, x : A \vdash_p e : B$. These imply the goal.

Lemma 1.11 (Inversion for Recursive Functions). *If* $\Gamma \vdash_p \mathrm{rec}\ f\,(x : A).\, e : C$, *then either (1) or (2) holds.*

(1) $C \equiv \Pi\, x : A.\, B$ *for some* B, *and* $\Gamma, x : A \vdash_p e : B$.
(2) $C \equiv \Pi\, x : A.\, \alpha \parallel B \parallel \beta$ *for some* B, α, *and* β, *and* $\Gamma, x : A;\ \alpha \vdash e : B;\ \beta$.

Lemma 1.12 (Inversion for Inductive Data). *If* $\Gamma \vdash_p c_i\ e : C$ *or* $\Gamma;\ \alpha \vdash c_i\ e : C;\ \beta$, *then* $C \equiv D\ u_i\,[e/y_i]$, *where* $\mathrm{Ind}(D : \kappa, \{c_i : \Pi\, y_i : B_i.\, D\ u_i\}) \in \Psi$, *and* $\Gamma \vdash_p e : B_i$ *or* $\Gamma;\ \alpha \vdash e : B_i;\ \beta$.

Lemma 1.13 (Inversion for Shift). *If* $\Gamma;\ \alpha \vdash \mathcal{S}k : K.\, e : C;\ \beta$, *then* $C \equiv A$, $K \equiv A \to \alpha'$, $\Gamma, k : A \to \alpha';\ B \vdash e : B;\ \beta'$ *for some* A, B, α', *and* β' *such that* $\alpha \equiv \alpha'$ *and* $\beta \equiv \beta'$.

The proof is by induction on the derivation of $\mathcal{S}k : K.\, e$.

Case (Shift) Immediate.

Case (Conv2) We have $\Gamma;\ \alpha_1 \vdash \mathcal{S}k : K.\, e : C_1;\ \beta_1$ where $C \equiv C_1$, $\alpha \equiv \alpha_1$, and $\beta \equiv \beta_1$. By the induction hypothesis, $C \equiv A$, which, together with transitivity of equivalence (Lemma 1.4), implies $C_1 \equiv A$. The induction hypothesis also tells us $K \equiv A \to \alpha'$ where $\alpha \equiv \alpha'$. Since $\alpha \equiv \alpha_1$, transitivity gives us $\alpha_1 \equiv \alpha'$. The last thing the induction hypothesis gives us is $\Gamma, k : A \to \alpha';\ B \vdash e : B;\ \beta'$ where $\beta \equiv \beta'$. The goal follows by $\beta' \equiv \beta_1$.

Lemma 1.14 (Purity of Values). *If* $\Gamma;\ \alpha \vdash v : A;\ \beta$, *then* $\alpha \equiv \beta$ *and* $\Gamma \vdash_p v : A$.

Since all the typing rules for values conclude with a $\vdash_p$-judgment, we know that the two answer types are introduced by rule (Exp), and that $\alpha \equiv \beta$. This implies the goal.

Theorem 1.15 (Preservation).

(1) *If* $\Gamma \vdash_p \mathsf{e} : \mathsf{A}$ *and* $\mathsf{e} \triangleright \mathsf{e}'$, *then* $\Gamma \vdash_p \mathsf{e}' : \mathsf{A}$.
(2) *If* $\Gamma; \alpha \vdash \mathsf{e} : \mathsf{A}; \beta$ *and* $\mathsf{e} \triangleright \mathsf{e}'$, *then* $\Gamma; \alpha \vdash \mathsf{e}' : \mathsf{A}; \beta$.

The proof is by induction on the derivation of e. We show several representative cases.

Case Part 1

Sub-Case (Var)

The only reduction rule that we can apply to a variable is the δ rule. When this rule is applicable, we know that the typing environment has a binding of the form $\mathsf{x} = \mathsf{v} : \mathsf{A}$. The goal immediately follows.

Sub-Case (App1)

Sub-Sub-Case $\mathsf{e}_0\ \mathsf{e}_1 \triangleright \mathsf{e}_0\ \mathsf{e}_1'$

Our goal is to show:

$$\Gamma \vdash_p \mathsf{e}_0\ \mathsf{e}_1' : \mathsf{B}\,[\mathsf{e}_1/\mathsf{x}]$$

By Part 1 of the induction hypothesis, we have $\Gamma \vdash_p \mathsf{e}_1' : \mathsf{A}$. By (App1), we also have $\Gamma \vdash_p \mathsf{e}_0\ \mathsf{e}_1' : \mathsf{B}\,[\mathsf{e}_1'/\mathsf{x}]$. Since $\mathsf{e}_1 \triangleright \mathsf{e}_1'$, we have $\mathsf{B}\,[\mathsf{e}_1'/\mathsf{x}] \equiv \mathsf{B}\,[\mathsf{e}_1/\mathsf{x}]$. The goal follows by (Conv1).

Sub-Sub-Case $(\lambda\,\mathsf{x} : \mathsf{A}.\,\mathsf{e})\ \mathsf{v} \triangleright \mathsf{e}\,[\mathsf{v}/\mathsf{x}]$

Our goal is to show:

$$\Gamma \vdash_p \mathsf{e}\,[\mathsf{v}/\mathsf{x}] : \mathsf{B}\,[\mathsf{v}/\mathsf{x}]$$

By Lemma 1.10, we have $\Gamma, \mathsf{x} : \mathsf{A} \vdash_p \mathsf{e} : \mathsf{B}$. The goal follows by the substitution lemma (Lemma 1.5).

Sub-Case (Match1)

Sub-Sub-Case $\mathsf{match\ e\ as\ x\ in\ D\ a\ return\ P\ with}\ \{\mathsf{c_i\ y_i} \rightarrow \mathsf{e_i}\} \triangleright$ $\mathsf{match\ e'\ as\ x\ in\ D\ a\ return\ P\ with}\ \{\mathsf{c_i\ y_i} \rightarrow \mathsf{e_i}\}$

Our goal is to show:

$$\Gamma \vdash_p \mathsf{match\ e'\ as\ x\ in\ D\ a\ return\ P\ with}\ \{\mathsf{c_i\ y_i} \rightarrow \mathsf{e_i}\} : \mathsf{P}\,[\mathsf{u}/\mathsf{a}, \mathsf{e}/\mathsf{x}]$$

By Part 1 of the induction hypothesis, we have $\Gamma \vdash_p \mathsf{e}' : \mathsf{D\ u}$. By (Match1), we also have $\Gamma \vdash_p \mathsf{match\ e'\ as\ x\ in\ D\ a\ return\ P\ with}\ \{\mathsf{c_i\ y_i} \rightarrow \mathsf{e_i}\} : \mathsf{P}\,[\mathsf{u}/\mathsf{a}, \mathsf{e}'/\mathsf{x}]$. Since $\mathsf{e} \triangleright \mathsf{e}'$, the goal follows by (Conv1).

Sub-Sub-Case $\mathsf{match\ c_i\ v\ as\ x\ in\ D\ a\ return\ P\ with}\ \{\mathsf{c_i\ y_i} \rightarrow \mathsf{e_i}\} \triangleright \mathsf{e_i}\,[\mathsf{v}/\mathsf{y_i}]$

Our goal is to show:

$$\Gamma \vdash_p \mathsf{e_i}\,[\mathsf{v}/\mathsf{y_i}] : \mathsf{P}\,[\mathsf{u}/\mathsf{a}, \mathsf{c_i\ v}/\mathsf{x}]$$

By inversion for inductive data and (Match1), we have $\Gamma \vdash_p \mathsf{c_i\ v} : \mathsf{D\ u}$ and $\Gamma, \mathsf{y_i} : \mathsf{A_i} \vdash_p \mathsf{e_i} : \mathsf{D\ u_i}$. The goal follows by Lemma 1.5.

Sub-Case $\langle \mathsf{e} \rangle \triangleright \langle \mathsf{e}' \rangle$

Our goal is to show:

$$\Gamma \vdash_p \langle \mathsf{e}' \rangle : \mathsf{A}$$

By (Reset), we have $\Gamma; \mathsf{B} \vdash \mathsf{e} : \mathsf{B}; \mathsf{A}$. By Part 2 of the induction hypothesis, we also have $\Gamma; \mathsf{B} \vdash \mathsf{e}' : \mathsf{B}; \mathsf{A}$. This implies the goal.

Sub-Case $\langle \mathsf{v} \rangle \triangleright \mathsf{v}$

Our goal is to show:

$$\Gamma \vdash_p \mathsf{v} : \mathsf{A}$$

By (Reset), we have $\Gamma; \mathsf{B} \vdash \mathsf{v} : \mathsf{B}; \mathsf{A}$. By Lemma 1.14, we must have a derivation $\Gamma \vdash_p \mathsf{v} : \mathsf{B}$, and A and B must be equivalent. These imply the goal.

Sub-Case $\langle F[\mathcal{S}k : A' \to \alpha .\, e]\rangle \triangleright \langle e\,[\lambda\, x : A' .\, \langle F[x]\rangle / k]\rangle$
Our goal is to show:

$$\Gamma \vdash_p \langle e\,[\lambda\, x : A' .\, \langle F[x]\rangle / k]\rangle : A$$

Following Asai and Kameyama [2007], we prove this case by decomposing the reduction into small reductions, which captures one context frame at a time. That is, we refine the reduction rule of `shift` as follows:

$$\begin{array}{rcl}
(\mathcal{S}k : \Pi\, x : A.\, B \to \alpha .\, e)\ e_1 & \triangleright & \mathcal{S}k' : B\,[e_1/x] \to \alpha .\, e\,[\lambda\, v : \Pi\, x : A.\, B.\, \langle k'\ (v\ e_1)\rangle / k] \\
v_0\ (\mathcal{S}k : A \to \alpha .\, e) & \triangleright & \mathcal{S}k' : B \to \alpha .\, e\,[\lambda\, v : A.\, \langle k'\ (v_0\ v)\rangle / k] \\
\text{let } x = \mathcal{S}k : A \to \alpha .\, e : A \text{ in } e_2 & \triangleright & \mathcal{S}k' : B \to \alpha .\, e\,[\lambda\, v : A.\, k'\ (\text{let } x = v : A \text{ in } e_2) / k] \\
c_i\ (\mathcal{S}k : A \to \alpha .\, e) & \triangleright & \mathcal{S}k' : D\ u \to \alpha .\, e\,[\lambda\, v : A.\, k'\ (c_i\ v) / k] \\
\begin{array}{r}\text{match } \mathcal{S}k : D\ u \to \alpha .\, e \text{ as \_} \\ \text{in } D \text{ \_ return } P \\ \text{with } \{c_i\ y_i \to e_i\}\end{array} & \triangleright & \begin{array}{l}\mathcal{S}k' : P \to \alpha . \\ \quad e\,[\lambda\, v : D\ u.\, k'\ (\text{match } v \text{ as \_ in } D \text{ \_ return } P \text{ with } \{c_i\ y_i \to e_i\}) / k]\end{array} \\
\langle \mathcal{S}k : A \to \alpha .\, e\rangle & \triangleright & e\,[\lambda\, v : A.\, v / k]
\end{array}$$

Sub-Sub-Case $(\mathcal{S}k : \Pi\, x : A'.\, B' \to \alpha' .\, e)\ e_1 \triangleright \mathcal{S}k' : B'\,[e_1/x] \to \alpha' .\, e\,[\lambda\, v : \Pi\, x : A'.\, B'.\, \langle k'\ (v\ e_1)\rangle / k]$
The application must be derived by (App3), (App4), or (Conv2). We consider the first case. Our goal is to show:

$$\Gamma;\, \alpha \vdash \mathcal{S}k' : B'\,[e_1/x] \to \alpha' .\, e\,[\lambda\, v : \Pi\, x : A'.\, B'.\, \langle k'\ (v\ e_1)\rangle / k] : B\,[e_1/x];\, \beta$$

By the typing rule, we have $\Gamma;\, \alpha \vdash \mathcal{S}k : \Pi\, x : A'.\, B' \to \alpha' .\, e : \Pi\, x : A.\, B;\, \beta$ and $\Gamma \vdash_p e_1 : A$. By inversion for `shift` (Lemma 1.13), we know that $\Pi\, x : A.\, B \equiv A_1$, $\Pi\, x : A'.\, B' \to \alpha' \equiv A_1 \to \alpha_1$, $\Gamma,\, k : A_1 \to \alpha_1;\, B \vdash e : B;\, \beta_1$ where $\alpha \equiv \alpha_1$ and $\beta \equiv \beta_1$. We must show that the function we substitute for k, namely $\lambda\, v : \Pi\, x : A'.\, B'.\, \langle k'\ (v\ e_1)\rangle$, has the correct type. By the definition of reduction, we know A_1 has the form $\Pi\, x : A_2.\, B_2$. The equivalence $\Pi\, x : A.\, B \equiv A_1$ implies $A \equiv A_2$, and similarly, $\Pi\, x : A'.\, B' \to \alpha' \equiv A_1 \to \alpha_1$ implies $A' \equiv A_2$. From these equivalences, we obtain $A \equiv A'$, which allows us to conclude that application $v\ e_1$ has type $B'\,[e_1/x]$, and $k'\ (v\ e_1)$ has type α'. Since the body of the function is pure, $\langle k'\ (v\ e_1)\rangle$ also has type α'. The equivalence information we obtain by the inversion lemma also gives us $B \equiv B'$ and $\alpha \equiv \alpha'$. These imply the goal.

Sub-Sub-Case $v_0\ (\mathcal{S}k : A' \to \alpha' .\, e) \triangleright \mathcal{S}k' : B' \to \alpha' .\, e\,[\lambda\, v : A'.\, \langle k'\ (v_0\ v)\rangle / k]$
The application must be derived by (App2), (App4), or (Conv2). We consider the first case. Our goal is to show:

$$\Gamma;\, \alpha \vdash \mathcal{S}k' : B \to \alpha' .\, e\,[\lambda\, v : A'.\, \langle k'\ (v_0\ v)\rangle / k] : B;\, \beta$$

By the typing rule, we have $\Gamma \vdash_p v_0 : A \to B$ and $\Gamma;\, \alpha \vdash \mathcal{S}k : A' \to \alpha' .\, e : A;\, \beta$. By inversion for `shift` (Lemma 1.13, we know that $A \equiv A_1$, $A' \to \alpha' \equiv A_1 \to \alpha_1$, $\Gamma,\, k : A_1 \to \alpha_1;\, B_1 \vdash e : B_1;\, \beta_1$ where $\alpha \equiv \alpha_1$ and $\beta \equiv \beta_1$. As in the previous case, we check whether the function we substitute for k is of the required type. By $A' \to \alpha' \equiv A_1 \to \alpha_1$ and transitivity of equivalence (Lemma 1.4), we know $A' \equiv A_1$, which, given $A \equiv A_1$, implies $A \equiv A'$. This gives us $v_0\ v : B$, and $k'\ (v_0\ v) : \alpha'$. Since the body of the function is pure, $\langle k'\ (v_0\ v)\rangle : \alpha'$. The equivalence information further gives us $\alpha \equiv \alpha'$, allowing us to derive the goal.

Sub-Sub-Case $\langle \mathcal{S}k : A' \to \alpha' .\, e\rangle \triangleright e\,[\lambda\, v : A'.\, v / k]$
By (Reset), we know that $\Gamma;\, B' \vdash \mathcal{S}k : A' \to \alpha' .\, e : B';\, A$. By inversion for `shift` (Lemma 1.13), we also know that $B' \equiv A_1$, $A' \to \alpha' \equiv A_1 \to \alpha$, $\Gamma,\, k : A_1 \to \alpha;\, B \vdash e : B;\, \beta$, $B' \equiv \alpha$, and $A \equiv \beta$.

It is easy to see that $\lambda\, v : A'.\, v$ has type $A' \rightarrow A'$. We must show this type is equivalent to $A_1 \rightarrow \alpha$. The equivalence $A' \rightarrow \alpha' \equiv A_1 \rightarrow \alpha$ implies $A' \equiv A_1$ and $\alpha' \equiv \alpha$. By transitivity of $\equiv$ (Lemma 1.4), we also have $A_1 \equiv \alpha$. This means substitution of $\lambda\, v : A'.\, v$ for k is type-safe. The goal now follows by $A \equiv \beta$.

1.5 Progress

Lemma 1.16 (Canonical Forms).

(1) *If* $\bullet \vdash_p v : \Pi\, x{:}A.\, B$ *or* $\bullet \vdash_p v : \Pi\, x{:}A.\, \alpha \parallel B \parallel \beta$*, then* v *is of the form* $\lambda\, x : A'.\, e$ *or* $\mathrm{rec}\, f\, (x : A').\, e$.
(2) *If* $\bullet \vdash_p v : D\, u$*, then* v *is of the form* $c_i\, v'$.

Theorem 1.17 (Progress).

(1) *If* $\bullet \vdash_p e : A$*, then either* e *is a value, or there is a* e' *such that* $e \triangleright e'$.
(2) *If* $\bullet; \alpha \vdash e : A; \beta$*, then* e *is a value, or there is a* e' *such that* $e \triangleright e'$*, or it is a stuck term of the form* $F[\mathcal{S}k : \_.\, e']$.

The proof is by induction on the derivation of e. The progress property in the usual sense holds only for pure terms, because impure terms are not executable in general (*e.g.*, a `shift` clause is a stuck term).

Case Part 1
Sub-Case (Var)
This case is impossible since no variable is well-typed under an empty context.
Sub-Case (Abs1), (Abs2), (Rec1), (Rec2)
These cases are trivial since functions are values.
Sub-Case (App1)
By Part 1 of the induction hypothesis, we know e_0 is either a value or there is a e_0' such that $e_0 \triangleright e_0'$, and similarly for e_1. If e_0 is a value, we know from Lemma 1.16 that it has the form $\lambda\, x : A'.\, e_0'$. If e_1 is also a value, the application is a β-redex. If e_1 is a non-value, $e_0\, e_1 \triangleright e_0\, e_1'$. If e_0 is a non-value, $e_0\, e_1 \triangleright e_0'\, e_1$.
Sub-Case (Match1)
By Part 1 of the induction hypothesis, we know e is either a value, or there is a e' such that $e \triangleright e'$. In the first case, we know from Lemma 1.16 that e is of the form $c_i\, v$. Therefore $\mathrm{match}\ e\ \mathrm{as}\ x\ \mathrm{in}\ D\ a\ \mathrm{return}\ P\ \mathrm{with}\ \{c_i\, y_i \rightarrow e_i\} \triangleright e_i\, [v/y_i]$. In the second case, $\mathrm{match}\ e\ \mathrm{as}\ x\ \mathrm{in}\ D\ a\ \mathrm{return}\ P\ \mathrm{with}\ \{c_i\, y_i \rightarrow e_i\} \triangleright \mathrm{match}\ e'\ \mathrm{as}\ x\ \mathrm{in}\ D\ a\ \mathrm{return}\ P\ \mathrm{with}\ \{c_i\, y_i \rightarrow e_i\}$.
Sub-Case (Reset)
By Part 2 of the induction hypothesis, we know e is either a value, or there is a e' such that $e \triangleright e'$, or it is a stuck term of the form $F[\mathcal{S}k : \_.\, e']$. In the first case, $\langle e \rangle \triangleright e$. In the second case, $\langle e \rangle \triangleright \langle e' \rangle$. In the last case, $\langle e \rangle \triangleright \langle e'\, [\lambda\, x : \_.\, \langle F[x] \rangle / k] \rangle$.
Case Part 2
Sub-Case (App6)
By Part 1 of the induction hypothesis, we know either e_0 is a value, or there is a e_0' such that $e_0 \triangleright e_0'$. By Part 2 of the induction hypothesis, we know the same holds for e_1, with an additional possibility that e_1 is a stuck term of the form $F[\mathcal{S}k : \_.\, e]$. Suppose e_0 is a value. If e_1 is also a value, the application is a β-redex and takes step. If e_1 is a non-value that evaluates to e_1', $e_0\, e_1 \triangleright e_0\, e_1'$. If e_1 is a stuck term, the whole application is also a stuck term $e_0\, F[\mathcal{S}k : \_.\, e]$.
Sub-Case (Shift)

By Part 2 of the induction hypothesis, we know e is either a value, or there is a e' such that $\mathsf{e} \triangleright \mathsf{e}'$, or it is a stuck term of the form $\mathsf{F}[\mathcal{S}\mathsf{k} : \_ .\, \mathsf{e}']$. In the first and third cases, the entire `shift` construct is a stuck term. In the second case, $\mathcal{S}\mathsf{k} : \mathsf{A} \to \alpha .\, \mathsf{e} \triangleright \mathcal{S}\mathsf{k} : \mathsf{A} \to \alpha .\, \mathsf{e}'$.

$$
\begin{array}{rcl}
\Psi & ::= & \bullet \mid \Psi,\, \mathsf{Ind}(\mathsf{D} : \kappa,\, \{\mathsf{c_i} : \mathsf{C_i}\}) \\
\Gamma & ::= & \bullet \mid \Gamma,\, \mathsf{x} : \mathsf{A} \mid \Gamma,\, \mathsf{x} = \mathsf{A} : \mathsf{e} \\
\kappa & ::= & * \mid \Pi\, \overline{\mathsf{x_i} : \mathsf{A_i}}.\, * \\
\mathsf{A}, \alpha & ::= & \mathsf{Unit} \mid \mathsf{D}\ \overline{\mathsf{e_i}} \mid \Pi\, \overline{\mathsf{x_i} : \mathsf{A_i}}.\, \mathsf{B} \mid \Pi\, \mathsf{x} : \mathsf{A}.\, \alpha \parallel \mathsf{B} \parallel \beta \\
\mathsf{v} & ::= & () \mid \mathsf{x} \mid \lambda\, \mathsf{x} : \mathsf{A}.\, \mathsf{e} \mid \mathsf{rec}\ \mathsf{f}\ (\mathsf{x} : \mathsf{A}).\, \mathsf{e} \mid \mathsf{c_i}\ \overline{\mathsf{v_i}} \\
\mathsf{e} & ::= & \mathsf{v} \mid \mathsf{e}\ \mathsf{e} \mid \mathsf{let}\ \mathsf{x} = \mathsf{e} : \mathsf{A}\ \mathsf{in}\ \mathsf{e} \mid \mathsf{c_i}\ \overline{\mathsf{e_i}} \\
 & & \mid\ \mathsf{match}\ \mathsf{e}\ \mathsf{as}\ \mathsf{D}\ \overline{\mathsf{a_i}}\ \mathsf{in}\ \mathsf{x}\ \mathsf{return}\ \mathsf{P}\ \mathsf{with}\ \{\mathsf{c_i}\ \overline{\mathsf{y_i}} \to \mathsf{e_i}\} \\
 & & \mid\ \mathcal{S}\mathsf{k} : \mathsf{A}.\, \mathsf{e} \mid \langle \mathsf{e} \rangle
\end{array}
$$

Fig. 1. $\lambda_{\Pi}^{s/r+}$ Syntax

2 λ_{Π}^{s/r^+}: EXTENDING $\lambda_{\Pi}^{s/r}$ WITH MULTI-ARITY INDUCTIVE DATATYPES

This section shows how to support multi-arity datatype constants and constructors.

2.1 Syntax

Figure 1 defines the syntax of the extended language $\lambda_{\Pi}^{s/r+}$. The functional kind $\Pi\, \overline{\mathsf{x_i} : \mathsf{A_i}}.\, *$ abbreviates $\Pi\, \mathsf{x_1} : \mathsf{A_1}.\, \mathsf{x_2} : \mathsf{A_2}.\, \ldots\, \mathsf{x_n} : \mathsf{A_n}.\, *$, where each $\mathsf{A_i}$ may depend on preceding variables $\mathsf{x_j}$ where $j < i$. The kind represents the type of datatype constants that takes in n indices. That is, if constant D has such a type, it must be applied to a sequence of n terms $\overline{\mathsf{e_i}}$. Function types $\Pi\, \overline{\mathsf{x_i} : \mathsf{A_i}}.\, \mathsf{B}$ are extended in a similar way, in order to support multi-argument constructors. Note that we do not extend the impure function type $\Pi\, \mathsf{x} : \mathsf{A}.\, \alpha \parallel \mathsf{B} \parallel \beta$, because constructor application causes no effect other than the effect associated with the arguments. This is in contrast to function application, where the effect of the whole term may come from the function's body.

2.2 Evaluation

Figure 2 shows changes in the rules for evaluation. The extended evaluation contexts tell us that when we have a constructor $\mathsf{c_i}$ applied to a sequence of arguments, we evaluate them from left to right. ι-reduction happens when the scrutinee of a pattern matching construct is a constructor application to values, and the reduction involves sequential substitution. We use the notation $\mathsf{e}\,[\overline{\mathsf{v_i}/\mathsf{x_i}}]$ to mean replacing $\mathsf{x_1}$ in e with $\mathsf{v_1}$, and then replacing $\mathsf{x_2}$ in the resulting term with $\mathsf{v_2}$, and so on.

2.3 Typing Rules

We now define typing rules for the new constructs. The formation rules of functional kinds and types are a straightforward extension of their $\lambda_{\Pi}^{s/r}$ counterpart: we simply type check each domain $\mathsf{A_i}$ (and the co-domain B) with a context extended with preceding variables. Rule (Data) requires all indices $\mathsf{e_i}$ to be a pure term. Notice that each $\mathsf{e_i}$ has a type of the form $\mathsf{A_i}\,[\overline{\mathsf{e_j}/\mathsf{x_j}}]_p^i$, which reads: for all $j < i$ where $\mathsf{e_j}$ is a pure term, substitute $\mathsf{e_j}$ for $\mathsf{x_j}$ in $\mathsf{A_i}$. We need this substitution because each $\mathsf{A_i}$ may depend on preceding variables. However, we should only substitute pure terms: remember that our language do not allow dependency on impure terms.

Rule (Const1) accounts for constructor application where all arguments are pure. The only change from $\lambda_{\Pi}^{s/r}$'s rule is that the result type is applied a sequential substitution. On the other

$\boxed{\text{Evaluation Contexts}}$

$$\begin{array}{rcl} \mathsf{E} & ::= & \ldots \mid \mathsf{c_i}\ \mathsf{v_1} \ldots \mathsf{E} \ldots \mathsf{e_n} \\ \mathsf{F} & ::= & \ldots \mid \mathsf{c_i}\ \mathsf{v_1} \ldots \mathsf{F} \ldots \mathsf{e_n} \end{array}$$

$\boxed{\text{Substitution}}$

$$\mathsf{e}\,[\overline{\mathsf{e_i}/\mathsf{x_i}}] \overset{\text{def}}{\equiv} (\ldots (\mathsf{e}\,[\mathsf{e_1}/\mathsf{x_1}])\,[\mathsf{e_2}/\mathsf{x_2}] \ldots)\,[\mathsf{e_n}/\mathsf{x_n}]$$

$\boxed{\text{Reduction Rules}}$

$$\Gamma \vdash \mathsf{match}\ \mathsf{c_i}\ \overline{\mathsf{v_i}}\ \mathsf{as}\ \mathsf{x}\ \mathsf{in}\ \mathsf{D}\ \overline{\mathsf{a_i}}\ \mathsf{return}\ \mathsf{P}\ \mathsf{with}\ \{\mathsf{c_i}\ \overline{\mathsf{y_i}} \rightarrow \mathsf{e_i}\} \quad \triangleright_\iota \quad \mathsf{e_i}\,[\overline{\mathsf{v_i}/\mathsf{y_i}}]$$

Fig. 2. $\lambda_{\Pi}^{s/r+}$ Evaluation

hand, rule (Const2), where we have impure arguments, needs some non-trivial reasoning about the answer types α_i, β_i, α, and β. Formally, these types must satisfy the following conditions:

- If $\mathsf{e_i}$ is not the last impure argument, $\alpha_i = \beta_j$, where $\mathsf{e_j}$ is the closest impure argument following $\mathsf{e_i}$.
- $\alpha = \alpha_i$, where $\mathsf{e_i}$ is the last impure argument.
- $\beta = \beta_i$, where $\mathsf{e_i}$ is the first impure argument.

To better understand the intuition behind these conditions, consider the following example, where we are applying the ::-constructor to a pure natural number and two impure lists:

$$\langle ::\ 0\ (\mathcal{S}\mathsf{k_1} : \mathbb{N} \rightarrow \mathsf{L}\ 2.\ ::\ 2\ 10\ (\mathsf{k_1}\ 30))\ (\mathcal{S}\mathsf{k_2} : \mathsf{L}\ 0 \rightarrow \mathsf{L}\ 3.\ ::\ 1\ 20\ (\mathsf{k_2}\ \mathsf{nil}))\rangle$$

This term reduces is the following way:

$$\begin{aligned} & \langle ::\ 0\ (\mathcal{S}\mathsf{k_1}.\ ::\ 2\ 10\ (\mathsf{k_1}\ 30))\ (\mathcal{S}\mathsf{k_2}.\ ::\ 1\ 20\ (\mathsf{k_2}\ \mathsf{nil}))\rangle \\ = & \langle ::\ 2\ 10\ \langle ::\ 0\ 30\ (\mathcal{S}\mathsf{k_2}.\ ::\ 1\ 20\ (\mathsf{k_2}\ \mathsf{nil}))\rangle\rangle \\ = & \langle ::\ 2\ 10\ \langle ::\ 1\ 20\ \langle ::\ 0\ 30\ \mathsf{nil}\rangle\rangle\rangle \\ = & [10;\ 20;\ 30] \end{aligned}$$

We first observe the chaining of argument answer types. In the above reduction sequence, we can see that the second `shift` happens when we call the continuation $\mathsf{k_1}$ captured by the first `shift`. According to the reduction rule of `shift`, we know that there is a `reset` surrounding the body of $\mathsf{k_1}$, and what this `reset` returns is determined by how the second `shift` changes the answer type. Thus, we have $\alpha_1 = \beta_2$.

Next, we turn our attention to the initial answer type of the entire application. Observe that before evaluation, we have a term of type $\mathsf{L}\ 1$ in the `reset` clause. In the above sequence, we obtain a singleton list when we call the continuation $\mathsf{k_2}$ captured by the second `shift`. This gives us $\alpha = \alpha_2$.

Lastly, we look at the final answer type. We find that elimination of the first `shift` changes the answer type from $\mathsf{L}\ 1$ to $\mathsf{L}\ 3$, and that the new answer type does not change in the subsequent steps. This is because elimination of the second `shift` happens in the application of $\mathsf{k_1}$: since the body of $\mathsf{k_1}$ is surrounded by a `reset`, the second `shift` cannot touch the $\mathsf{L}\ 3$-returning computation. Hence we have $\beta = \beta_1$.

$$\dfrac{\begin{array}{c}\vdash \Psi \quad \bullet \vdash \Pi\,\overline{a_i : A_i}.\,* \quad (\bullet \vdash \Pi\,\overline{y_i : B_i}.\,\mathsf{D}\ u_i : *)_{i=1\ldots k}\\ \mathsf{D}, c_i \text{ fresh} \quad \mathsf{safe}(\mathsf{D}, \overline{B_i})\end{array}}{\vdash \Psi, \mathsf{Ind}(\mathsf{D} : \Pi\,\overline{a_i : A_i}.\,*, \{c_i : \Pi\,\overline{y_i : B_i}.\,\mathsf{D}\ u_i\})}\ (\textsc{ExtendSig})$$

$$\dfrac{\Gamma \vdash A_1 : * \quad \Gamma, x_1 : A_1 \vdash A_2 : * \quad \ldots \quad \Gamma, \overline{x_i : A_i} \vdash A_n : *}{\Gamma \vdash \Pi\,\overline{x_i : A_i}.\,*}\ (\textsc{PiK})$$

$$\dfrac{\mathsf{Ind}(\mathsf{D} : \Pi\,\overline{x_i : A_i}.\,*, \{c_i : C_i\}) \in \Psi \quad (\Gamma \vdash_p e_i : A_i\,[\overline{e_j/x_j}]^i_p)_{i=1\ldots n}}{\Gamma \vdash \mathsf{D}\ \overline{e_i} : *}\ (\textsc{Data})$$

$$\dfrac{\Gamma \vdash A_1 : * \quad \Gamma, x_1 : A_1 \vdash A_2 : * \quad \ldots \quad \Gamma, \overline{x_i : A_i} \vdash B : *}{\Gamma \vdash \Pi\,\overline{x_i : A_i}.\,B : *}\ (\textsc{PiT1})$$

$$\dfrac{\vdash \Gamma \quad \mathsf{Ind}(\mathsf{D} : \kappa, \{c_i : \Pi\,\overline{y_i : B_i}.\,\mathsf{D}\ \overline{u_i}\}) \in \Psi \quad \Gamma \vdash_p e_i : B_i\,[\overline{e_j/x_j}]^i_p}{\Gamma \vdash_p c_i\ \overline{e_i} : \mathsf{D}\ \overline{u_i}\,[\overline{e_i/y_i}]}\ (\textsc{Const1})$$

$$\dfrac{\begin{array}{c}\vdash \Gamma \quad \mathsf{Ind}(\mathsf{D} : \kappa, \{c_i : \Pi\,\overline{y_i : B_i}.\,\mathsf{D}\ \overline{u_i}\}) \in \Psi\\ \Gamma \vdash_p e_i : B_i\,[\overline{e_j/x_j}]^i_p \text{ or}\\ (\Gamma; \alpha_i \vdash e_i : B_i\,[\overline{e_j/x_j}]^i_p; \beta_i \text{ and } y_i \notin FV(B_j \cup \mathsf{D}\ \overline{u_i}) \text{ where } i < j)\end{array}}{\Gamma; \alpha \vdash c_i\ \overline{e_i} : \mathsf{D}\ \overline{u_i}\,[\overline{e_j/y_j}]^i_p; \beta}\ (\textsc{Const2})$$

$$\dfrac{\begin{array}{c}\mathsf{Ind}(\mathsf{D} : \Pi\,\overline{a_i : A_i}.\,*, \{c_i : \Pi\,\overline{y_i : B_i}.\,\mathsf{D}\ \overline{u_i}\}) \in \Psi \quad \Gamma \vdash_p e : \mathsf{D}\ \overline{u}\\ \Gamma, \overline{a_i : A_i}, x : \mathsf{D}\ \overline{a_i} \vdash P : * \quad (\Gamma, \overline{y_i : B_i} \vdash_p e_i : P\,[\overline{u_i/a_i}]\,[c_i\ y_i/x])_{i=1\ldots k}\end{array}}{\Gamma \vdash_p \mathsf{match}\ e\ \mathsf{as}\ x\ \mathsf{in}\ \mathsf{D}\ \overline{a_i}\ \mathsf{return}\ P\ \mathsf{with}\ \{c_i\ \overline{y_i} \to e_i\} : P\,[\overline{u/a_i}]\,[e/x]}\ (\textsc{Match1})$$

$$\dfrac{\begin{array}{c}\mathsf{Ind}(\mathsf{D} : \Pi\,\overline{a_i : A_i}.\,*, \{c_i : \Pi\,\overline{y_i : B_i}.\,\mathsf{D}\ \overline{u_i}\}) \in \Psi \quad \Gamma \vdash_p e : \mathsf{D}\ \overline{u}\\ \Gamma, \overline{a_i : A_i}, x : \mathsf{D}\ a_i \vdash P : *\\ (\Gamma, \overline{y_i : B_i}; \alpha\,[\overline{u_i/a_i}]\,[c_i\ y_i/x] \vdash e_i : P\,[\overline{u_i/a_i}]\,[c_i\ y_i/x]; \beta\,[\overline{u_i/a_i}]\,[c_i\ y_i/x])_{i=1\ldots k}\end{array}}{\Gamma; \alpha\,[\overline{u/a_i}]\,[e/x] \vdash \mathsf{match}\ e\ \mathsf{as}\ x\ \mathsf{in}\ \mathsf{D}\ \overline{a_i}\ \mathsf{return}\ P\ \mathsf{with}\ \{c_i\ \overline{y_i} \to e_i\} : P\,[\overline{u/a_i}]\,[e/x]; \beta\,[\overline{u/a_i}]\,[e/x]}\ (\textsc{Match2})$$

$$\dfrac{\begin{array}{c}\mathsf{Ind}(\mathsf{D} : \Pi\,\overline{a_i : A_i}.\,*, \{c_i : \Pi\,\overline{y_i : B_i}.\,\mathsf{D}\ \overline{u_i}\}) \in \Psi \quad \Gamma; \beta \vdash e : \mathsf{D}\ \overline{u}; \gamma\\ \Gamma \vdash P : * \quad (\Gamma, \overline{y_i : B_i}; \alpha \vdash e_i : P; \beta)_{i=1\ldots k}\end{array}}{\Gamma; \alpha \vdash \mathsf{match}\ e\ \mathsf{as}\ \_\ \mathsf{in}\ \mathsf{D}\ \overline{\_}\ \mathsf{return}\ P\ \mathsf{with}\ \{c_i\ \overline{y_i} \to e_i\} : P; \beta}\ (\textsc{Match3})$$

Fig. 3. $\lambda_{\Pi}^{s/r+}$ Typing

3 λ_Π^K: TARGET LANGUAGE OF CPS TRANSLATION

This section presents the complete specification of λ_Π^k, the target language of the CPS translation.

3.1 Syntax

$$
\begin{array}{rcl}
\Psi & ::= & \{\} \mid \Psi, \mathsf{Ind}(\mathrm{D} : \kappa, \{\mathrm{c_i} : \mathrm{C_i}\}) \\
\Gamma & ::= & \bullet \mid \Gamma, \mathrm{x} : \mathrm{A} \mid \Gamma, \mathrm{x} = \mathrm{e} : \mathrm{A} \mid \Gamma, \mathrm{e} \equiv \mathrm{e} \\
\kappa & ::= & * \mid \Pi\, \mathrm{x} : \mathrm{A}.\, * \\
\mathrm{A} & ::= & \mathbf{Unit} \mid \alpha \mid \mathrm{D}\ \mathrm{e} \mid \Pi\, \mathrm{x} : \mathrm{A}.\, \mathrm{B} \mid \Pi\, \alpha : *.\, \mathrm{B} \\
\mathrm{e} & ::= & () \mid \mathrm{x} \mid \lambda\, \mathrm{x} : \mathrm{A}.\, \mathrm{e} \mid \lambda\, \alpha : *.\, \mathrm{e} \mid \mathbf{rec}\ \mathrm{f}\ (\mathrm{x} : \mathrm{A}).\, \mathrm{e} \\
 & & \mid \mathrm{e}\ \mathrm{e} \mid \mathrm{e}\ \mathrm{A} \mid \mathrm{e} \mathbin{@} \mathrm{A}\ \mathrm{e} \mid \mathbf{let}\ \mathrm{x} = \mathrm{e} : \mathrm{A}\ \mathbf{in}\ \mathrm{e} \\
 & & \mid \mathrm{c_i}\ \mathrm{e} \mid \mathbf{match}\ \mathrm{e}\ \mathbf{as}\ \mathrm{x}\ \mathbf{in}\ \mathrm{D}\ \mathrm{a}\ \mathbf{return}\ \mathrm{P}\ \mathbf{with}\ \{\mathrm{c_i}\ \mathrm{y_i} \to \mathrm{e_i}\}
\end{array}
$$

Fig. 4. λ_Π^k Syntax

3.2 Evaluation and Equivalence

$\boxed{\text{Evaluation Contexts}}$

$$
\mathrm{E} \quad ::= \quad \mathrm{E}\ \mathrm{e} \mid \mathrm{E}\ \mathrm{A} \mid \mathrm{E} \mathbin{@} \mathrm{A}\ \mathrm{e} \mid \mathbf{match}\ \mathrm{E}\ \mathbf{as}\ \mathrm{x}\ \mathbf{in}\ \mathrm{D}\ \mathrm{a}\ \mathbf{return}\ \mathrm{P}\ \mathbf{with}\ \{\mathrm{c_i}\ \mathrm{y_i} \to \mathrm{e_i}\}
$$

$\boxed{\text{Reduction Rules}}$

$$
\begin{array}{rcl}
\Gamma \vdash \mathrm{x} & \triangleright_\delta & \mathrm{e} \quad \text{if } \mathrm{x} = \mathrm{e} : \mathrm{A} \in \Gamma \\
\Gamma \vdash (\lambda\, \mathrm{x} : \mathrm{A}.\, \mathrm{e_0})\ \mathrm{e_1} & \triangleright_\beta & \mathrm{e_0}\,[\mathrm{e_1}/\mathrm{x}] \\
\Gamma \vdash (\lambda\, \alpha : *.\, \mathrm{e})\ \mathrm{A} & \triangleright_\beta & \mathrm{e}\,[\mathrm{A}/\alpha] \\
\Gamma \vdash \mathbf{let}\ \mathrm{x} = \mathrm{e_1}\ \mathbf{in}\ \mathrm{e_2} & \triangleright_\zeta & \mathrm{e_2}\,[\mathrm{e_1}/\mathrm{x}] \\
\Gamma \vdash (\lambda\, \alpha : *.\, \mathrm{e}) \mathbin{@} \mathrm{A}\ \mathrm{e_2} & \triangleright_@ & \mathrm{e}\,[\mathrm{A}/\alpha]\ \mathrm{e_2} \\
\Gamma \vdash \mathbf{match}\ \mathrm{c_i}\ \mathrm{v}\ \mathbf{as}\ \mathrm{x}\ \mathbf{in}\ \mathrm{D}\ \mathrm{a}\ \mathbf{return}\ \mathrm{P}\ \mathbf{with}\ \{\mathrm{c_i}\ \mathrm{y_i} \to \mathrm{e_i}\} & \triangleright_\iota & \mathrm{e_i}\,[\mathrm{v}/\mathrm{y_i}]
\end{array}
$$

Fig. 5. λ_Π^k Evaluation

$\boxed{\text{Equivalence Rules}}$

$$\frac{\Gamma \vdash \mathbf{t_0} \rhd^{\star} \mathbf{t} \qquad \Gamma \vdash \mathbf{t_1} \rhd^{\star} \mathbf{t}}{\Gamma \vdash \mathbf{t_0} \equiv \mathbf{t_1}}\ (\equiv)$$

$$\frac{\Gamma \vdash \mathbf{e} \rhd^{\star} \lambda\, \mathbf{x} : \mathbf{A}.\, \mathbf{e_0} \qquad \Gamma \vdash \mathbf{e'} \rhd^{\star} \mathbf{e_1} \qquad \Gamma, \mathbf{x} : \mathbf{A} \vdash \mathbf{e_0} \equiv \mathbf{e_1}\ \mathbf{x}}{\Gamma \vdash \mathbf{e} \equiv \mathbf{e'}}\ (\equiv\text{-}\eta_1)$$

$$\frac{\Gamma \vdash \mathbf{e} \rhd^{\star} \mathbf{v_0} \qquad \Gamma \vdash \mathbf{e'} \rhd^{\star} \lambda\, \mathbf{x} : \mathbf{A}.\, \mathbf{e_1} \qquad \Gamma, \mathbf{x} : \mathbf{A} \vdash \mathbf{e_0}\ \mathbf{x} \equiv \mathbf{e_1}}{\Gamma \vdash \mathbf{e} \equiv \mathbf{e'}}\ (\equiv\text{-}\eta_2)$$

$$\frac{}{\Gamma \vdash \mathbf{e_1}\ @\ \mathbf{A}\ (\lambda\, \mathbf{v} : \mathbf{B}.\, \mathbf{e_2}) \equiv (\lambda\, \mathbf{v} : \mathbf{B}.\, \mathbf{e_2})\ (\mathbf{e_1}\ \mathbf{B}\ \mathbf{id})}\ [\equiv\text{-}\textsc{Cont}]$$

Fig. 6. λ^k_Π Equivalence

3.3 Typing

$\boxed{\text{Signatures } \vdash \Psi}$

$$\frac{}{\vdash \{\}}\ [\textsc{EmptySig}] \qquad \frac{\vdash \Psi \quad \bullet \vdash \Pi\, a : A.\, * \quad (\bullet, D : \Pi\, a : A.\, * \vdash \Pi\, y_i : A_i.\, D\, u_i : *)_{i=1\ldots k} \quad D, c_i \text{ fresh} \quad \text{strictly-positive}(D, A_i)}{\vdash \Psi, \mathsf{Ind}(D : \Pi\, a : A.\, *, \{c_i : \Pi\, y_i : A_i.\, D\, u_i\})}\ [\textsc{ExtendSig}]$$

$\boxed{\text{Typing Environments } \vdash \Gamma}$

$$\frac{}{\vdash \bullet}\ [\textsc{Empty}] \qquad \frac{\vdash \Gamma \quad \Gamma \vdash A : *}{\vdash \Gamma, x : A}\ [\textsc{Extend1}] \qquad \frac{\vdash \Gamma \quad \Gamma \vdash e : A \quad \Gamma \vdash A : *}{\vdash \Gamma, x = e : A}\ [\textsc{Extend2}]$$

$$\frac{\vdash \Gamma}{\vdash \Gamma, \alpha : *}\ [\textsc{Extend3}] \qquad \frac{\vdash \Gamma \quad \Gamma \vdash e_1 : A \quad \Gamma \vdash e_2 : A}{\vdash \Gamma, e_1 \equiv e_2}\ [\textsc{Extend4}]$$

$\boxed{\text{Kinds } \Gamma \vdash \kappa}$

$$\frac{\vdash \Gamma}{\Gamma \vdash *}\ [\textsc{Star}] \qquad \frac{\Gamma \vdash A : *}{\Gamma \vdash \Pi\, x : A.\, *}\ (\textsc{PiK})$$

$\boxed{\text{Types } \Gamma \vdash A : *}$

$$\frac{\vdash \Gamma \quad \alpha : * \in \Gamma}{\Gamma \vdash \alpha : *}\ [\textsc{VarT}] \qquad \frac{\mathsf{Ind}(D : \Pi\, a : A.\, *, \{c_i : C_i\}) \in \Psi \quad \Gamma \vdash e : A}{\Gamma \vdash D\, e : *}\ [\textsc{Data}]$$

$$\frac{\Gamma, x : A \vdash B : *}{\Gamma \vdash \Pi\, x : A.\, B : *}\ [\textsc{PiT1}] \qquad \frac{\Gamma, \alpha : * \vdash B : *}{\Gamma \vdash \Pi\, \alpha : *.\, B : *}\ [\textsc{PiT2}]$$

Fig. 7. Well-formed Signatures, Typing Environments, Kinds, and Types

Note that $e \rhd^{\star}_{adm} e'$ in rule [Rec] means e reduces to e' after administrative reductions. The set of administrative redexes r are defined as follows:

$$\begin{array}{rcl}
r & ::= & m\ c \mid m\ A\ c \mid m\ @\ A\ c \mid c\ v \\
m & ::= & \lambda k : A.\, e \mid \lambda \alpha : *.\, \lambda k : A.\, e \\
c & ::= & k \mid id \mid \lambda x : A.\, r \mid \lambda x : A.\, x\, x'\, k \mid \lambda x : A.\, x\, x'\, A\, k \\
 & & \mid \lambda x : A.\, \mathbf{match}\ x\ \mathbf{as}\ x'\ \mathbf{in}\ D\ a\ \mathbf{return}\ P\ \mathbf{with}\ \{c_i\ y_i \rightarrow r\} \\
v & ::= & x \mid \lambda x : A.\, m \mid \mathbf{rec}\ f\ (x : A).\, m \mid c_i\ v
\end{array}$$

$\boxed{\text{Terms } \Gamma \vdash e : A}$

$$\frac{\vdash \Gamma \quad x : A \in \Gamma \text{ or } x = e : A \in \Gamma}{\Gamma \vdash x : A}\ [\textsc{Var}]$$

$$\frac{\Gamma, x : A \vdash e : B}{\Gamma \vdash \lambda x : A.\, e : \Pi x : A.\, B}\ [\textsc{Abs1}] \qquad \frac{\Gamma, \alpha : * \vdash e : B}{\Gamma \vdash \lambda \alpha : *.\, e : \Pi \alpha : *.\, B}\ [\textsc{Abs2}]$$

$$\frac{\Gamma, f : \Pi x : A.\, B, x : A \vdash e : B \quad e \triangleright^{\star}_{adm} e' \quad \text{guard}(f, x, e', \{\,\})}{\Gamma \vdash \text{rec } f\ (x : A).\, e : \Pi x : A.\, B}\ [\textsc{Rec}]$$

$$\frac{\Gamma \vdash e_0 : \Pi x : A.\, B \quad \Gamma \vdash e_1 : A}{\Gamma \vdash e_0\ e_1 : B\,[e_1/x]}\ [\textsc{App1}] \qquad \frac{\Gamma \vdash e : \Pi \alpha : *.\, B \quad \Gamma \vdash A : *}{\Gamma \vdash e\ A : B\,[A/\alpha]}\ [\textsc{App2}]$$

$$\frac{\Gamma \vdash e_1 : \Pi \alpha : *.\, (B \to \alpha) \to \alpha \quad \Gamma, v = e_1\ B\ \text{id} : B \vdash e_2 : A}{\Gamma \vdash e_1\ @\ A\ (\lambda v : B.\, e_2) : A}\ [\textsc{T-Cont}]$$

$$\frac{\Gamma \vdash e_1 : A \quad \Gamma, x = e_1 : A \vdash e_2 : B}{\Gamma \vdash \text{let } x = e_1 : A \text{ in } e_2 : B\,[e_2/x]}\ [\textsc{Let}]$$

$$\frac{\vdash \Gamma \quad \text{Ind}(D : \kappa, \{c_i : \Pi y_i : B_i.\, D\ u_i\}) \in \Psi \quad \Gamma \vdash e : B_i}{\Gamma \vdash c_i\ e : D\ u_i\,[e/y_i]}\ [\textsc{Const}]$$

$$\frac{\begin{array}{c}\text{Ind}(D : \Pi a : A.\, *, \{c_i : \Pi y_i : B_i.\, D\ u_i\}) \in \Psi \quad \Gamma \vdash e : D\ u \\ \Gamma, a : A, x : D\ a \vdash P : * \quad (\Gamma, y_i : B_i, u \equiv u_i, e \equiv c_i\ y_i \vdash e_i : P\,[u_i/a, c_i\ y_i/x])_{i=1\ldots k}\end{array}}{\Gamma \vdash \text{match } e \text{ as } x \text{ in } D\ a \text{ return } P \text{ with } \{c_i\ y_i \to e_i\} : P\,[u/a, e/x]}\ [\textsc{Match}]$$

$$\frac{e_1 \equiv e_2 \in \Gamma \quad e_1 \equiv c_i\ \overline{u} \quad e_2 \equiv c_j\ \overline{v} \quad c_i \neq c_j}{\Gamma \vdash e : A}\ [\textsc{InCon}]$$

Fig. 8. Typing Rules

4 TYPE PRESERVATION OF THE CPS TRANSLATION

Lemma 4.1 (CPS Computation η). *If* e *is a pure term, then* $e^{\div} \equiv \lambda\,\alpha : *.\,\lambda\,k : A \to \alpha.\,e^{\div} \mathbin{@} \alpha\;(\lambda\,v : B.\,k\,v)$

By the construction of the CPS translation, we know that any pure term e is translated to a polymorphic function $\lambda\,\alpha' : *.\,\lambda\,k' : A' \to \alpha'.\,e$. The equivalence follows by the η-rule and the semantics of $@$.

Lemma 4.2 (Compositionality). *Suppose* e' *is a pure term.*

(1) $(\kappa\,[e'/x])^{+} \equiv \kappa^{+}\,[e'^{\div}\;\_\;\mathbf{id}/x]$
(2) $(A\,[e'/x])^{+} \equiv A^{+}\,[e'^{\div}\;\_\;\mathbf{id}/x]$
(3) $(e\,[e'/x])^{\div} \equiv e^{\div}\,[e'^{\div}\;\_\;\mathbf{id}/x]$

The proof is by induction on the derivation of κ, A, and e. The only interesting case is when $e = x$, which uses the new equivalence rule:

$$\begin{aligned}
(x\,[e'/x])^{\div} &= e'^{\div} && \text{by substitution}\\
&\equiv \lambda\,\alpha.\,\lambda\,k.\,e'^{\div} \mathbin{@} \alpha\;(\lambda\,x.\,k\,x) && \text{by Lemma 4.1}\\
&\equiv \lambda\,\alpha.\,\lambda\,k.\,(\lambda\,x.\,k\,x)\;(e'^{\div}\;A'^{+}\;\mathbf{id}) && \text{by } [\equiv\text{-Cont}]\\
&= (\lambda\,\alpha.\,\lambda\,k.\,(\lambda\,x.\,k\,x)\;x)\,[e'^{\div}\;A'^{+}\;\mathbf{id}/x] && \text{by substitution}\\
&\rhd^{\star} (\lambda\,\alpha.\,\lambda\,k.\,k\,x)\,[e'^{\div}\;A'^{+}\;\mathbf{id}/x] && \text{by } \beta\text{-reduction}\\
&= x^{\div}\,[e'^{\div}\;A'^{+}\;\mathbf{id}/x] && \text{by definition of translation}
\end{aligned}$$

Notice that in the fifth step, we are substituting a non-value $e'^{\div}\;\_\;\mathbf{id}$. This is why the target language must be call-by-name.

Lemma 4.3 (Correctness).

(1) *If* $\Gamma \vdash \kappa$ *and* $\kappa \rhd^{\star} \kappa'$, *then* $\kappa^{+} \equiv \kappa'^{+}$.
(2) *If* $\Gamma \vdash A : *$ *and* $A \rhd^{\star} A'$, *then* $A^{+} \equiv A'^{+}$.
(3) *If* $\Gamma \vdash_{p} e : A$ *or* $\Gamma;\,\alpha \vdash e : A;\,\beta$ *and* $e \rhd^{\star} e'$, *then* $e^{\div} \equiv e'^{\div}$.

We first prove correctness with regard to single-step reduction, by cases on the $\rhd$-relation, and then extend the result to multi-step reduction, by induction on the length of the reduction sequence.

Case Part 3
Sub-Case $(\lambda\,x : A.\,e_0)\;v_1$ by (App5)

$$\begin{aligned}
((\lambda\,x.\,e_0)\;v_1)^{\div} &= \lambda\,k.\,(\lambda\,x.\,e_0)^{\div}\;(\beta\,[v_1/x])^{+}\;(\lambda\,v_0.\,v_1{}^{\div} \mathbin{@} (\beta\,[v_1/x])^{+}\;(\lambda\,v_1.\,v_0\;v_1\;k))\\
&= \lambda\,k.\,(\lambda\,\alpha.\,\lambda\,k.\,k\;(\lambda\,x.\,e_0{}^{\div}))\;(\beta\,[v_1/x])^{+}\;(\lambda\,v_0.\,v_1{}^{\div} \mathbin{@} (\beta\,[v_1/x])^{+}\;(\lambda\,v_1.\,v_0\;v_1\;k))\\
&\rhd^{\star}_{\beta} \lambda\,k.\,v_1{}^{\div} \mathbin{@} (\beta\,[v_1/x])^{+}\;(\lambda\,v_1.\,(\lambda\,x.\,e_0{}^{\div})\;v_1\;k)\\
&\rhd_{\beta} \lambda\,k.\,v_1{}^{\div} \mathbin{@} (\beta\,[v_1/x])^{+}\;(\lambda\,v_1.\,(e_0{}^{\div}\,[v_1/x])\;k)\\
&\equiv \lambda\,k.\,(\lambda\,v_1.\,(e_0{}^{\div}\,[v_1/x])\;k)\;(v_1{}^{\div}\;A^{+}\;\mathbf{id}) && \text{by } [\equiv\text{-Cont}]\\
&\rhd_{\beta} \lambda\,k.\,(e_0{}^{\div}\,[v_1{}^{\div}\;A^{+}\;\mathbf{id}/x])\;k\\
&\equiv e_0{}^{\div}\,[v_1{}^{\div}\;A^{+}\;\mathbf{id}/x] && \text{by } \eta\text{-equivalence}\\
&\equiv (e_0\,[v_1/x])^{\div} && \text{by Lemma 4.2}
\end{aligned}$$

Sub-Case $\mathsf{match}\ c_i\ v\ \mathsf{as}\ x\ \mathsf{in}\ D\ a\ \mathsf{return}\ P\ \mathsf{with}\ \{c_i\ y_i \to e_i\} \triangleright e_i\,[v/y_i]$ by (Match2)

$$
\begin{aligned}
&(\mathsf{match}\ c_i\ v\ \mathsf{as}\ x\ \mathsf{in}\ D\ a\ \mathsf{return}\ P\ \mathsf{with}\ \{c_i\ y_i \to e_i\})^{\div} \\
&= \lambda\, \mathbf{k}.\ (c_i\ v)^{\div}\ @\ \_\ (\lambda\, \mathbf{v}'.\ \mathbf{match}\ \mathbf{v}'\ \mathbf{as}\ \mathbf{x}\ \mathbf{in}\ \mathbf{D}\ a^{\div}\ \_\ \mathbf{id}\ \mathbf{return}\ (\beta\,[u/a,\, c_i\ v/x])^{+}\ \mathbf{with}\ \{\mathbf{c_i}\ \mathbf{y_i} \to e_i{}^{\div}\ \mathbf{k}\}) \\
&\equiv \lambda\, \mathbf{k}.\ \mathbf{match}\ \mathbf{c_i}\ (v^{\div}\ \_\ \mathbf{id})\ \mathbf{as}\ \mathbf{x}\ \mathbf{in}\ \mathbf{D}\ a^{\div}\ \_\ \mathbf{id}\ \mathbf{return}\ (\beta\,[u/a,\, c_i\ v/x])^{+}\ \mathbf{with}\ \{\mathbf{c_i}\ \mathbf{y_i} \to e_i{}^{\div}\ \mathbf{k}\} \quad \text{by Lemma 4.2, } \beta \\
&\triangleright_{\iota}\ \lambda\, \mathbf{k}.\ e_i{}^{\div}\ \mathbf{k}\,[\mathbf{c_i}\ (v^{\div}\ \_\ \mathbf{id})/\mathbf{y_i}]
\end{aligned}
$$

$$
\begin{aligned}
&(e_i{}^{\div}\,[c_i\ v/y_i])^{\div} \\
&\equiv e_i{}^{\div}\,[(c_i\ v)^{\div}\ \_\ \mathbf{id}/\mathbf{y_i}] && \text{by Lemma 4.2} \\
&\triangleright_{\beta}^{\star}\ e_i{}^{\div}\,[v^{\div}\ @\ \_\ (\lambda\, \mathbf{v}'.\ \mathbf{c_i}\ \mathbf{v}')/\mathbf{y_i}] \\
&\equiv e_i{}^{\div}\,[\mathbf{c_i}\ (v^{\div}\ \_\ \mathbf{id})/\mathbf{y_i}] \text{by Lemma 4.2} \\
&\equiv \lambda\, \mathbf{k}.\ e_i{}^{\div}\ \mathbf{k}\,[\mathbf{c_i}\ (v^{\div}\ \_\ \mathbf{id})/\mathbf{y_i}] \text{by } \eta \\
&= \lambda\, \mathbf{k}.\ e_i{}^{\div}\ \mathbf{k}\,[\mathbf{c_i}\ (v^{\div}\ \_\ \mathbf{id})/\mathbf{y_i}] && \text{by substitution}
\end{aligned}
$$

Sub-Case $\langle F[\mathcal{S} k : A \to \alpha.\, e]\rangle \triangleright \langle e\,[\lambda\, x : A.\, \langle F[x]\rangle/k]\rangle$
As in the preservation proof, we prove correctness with regard to smaller reductions.
Sub-Sub-Case $(\mathcal{S} c : \Pi\, x : A.\, B \to \alpha.\, e)\ e_1 \triangleright \mathcal{S} c' : B\,[e_1/x] \to \alpha.\, e\,[\lambda\, v : \Pi\, x : A.\, B.\, \langle c'\ (v\ e_1)\rangle/c]$
by (App3)

$$
\begin{aligned}
&((\mathcal{S} c : \Pi\, x : A.\, B \to \alpha.\, e)\ e_1)^{\div} \\
&= \lambda\, \mathbf{k}.\ ((\mathcal{S} c : \Pi\, x : A.\, B \to \alpha.\, e)^{\div}\ (\lambda\, \mathbf{v_0}.\ e_1{}^{\div}\ @\ \_\ (\lambda\, \mathbf{v_1}.\ \mathbf{v_0}\ \mathbf{v_1}\ \alpha^{+}\ \mathbf{k}))) \\
&= \lambda\, \mathbf{k}.\ ((\lambda\, \mathbf{k}.\ e^{\div}\ \mathbf{id}\,[\lambda\, \mathbf{v}.\, \lambda\, \alpha'.\, \lambda\, \mathbf{k}'.\, \mathbf{k}'\ (\mathbf{k}\ \mathbf{v})/\mathbf{c}])\ (\lambda\, \mathbf{v_0}.\ e_1{}^{\div}\ @\ \_\ (\lambda\, \mathbf{v_1}.\ \mathbf{v_0}\ \mathbf{v_1}\ \alpha^{+}\ \mathbf{k}))) \\
&\triangleright_{\beta}^{\star}\ \lambda\, \mathbf{k}.\ e^{\div}\ \mathbf{id}\,[\lambda\, \mathbf{v}.\, \lambda\, \alpha'.\, \lambda\, \mathbf{k}'.\, \mathbf{k}'\ (e_1{}^{\div}\ @\ \_\ (\lambda\, \mathbf{v_1}.\ \mathbf{v}\ \mathbf{v_1}\ \_\ \mathbf{k}))/\mathbf{c}]
\end{aligned}
$$

$$
\begin{aligned}
&(\mathcal{S} c' : B\,[e_1/x] \to \alpha.\, e\,[\lambda\, v : \Pi\, x : A.\, B.\, \langle c'\ (v\ e_1)\rangle/c])^{\div} \\
&= \lambda\, \mathbf{k}.\ (e\,[\lambda\, v : \Pi\, x : A.\, B.\, \langle c'\ (v\ e_1)\rangle/c])^{\div}\ \mathbf{id}\,[\lambda\, \mathbf{v}.\, \lambda\, \alpha'.\, \lambda\, \mathbf{k}'.\, \mathbf{k}'\ (\mathbf{k}\ \mathbf{v})/\mathbf{c}'] \\
&= \lambda\, \mathbf{k}.\ e^{\div}\,[\lambda\, \mathbf{v}.\, \lambda\, \alpha'.\, \lambda\, \mathbf{k}'.\, \mathbf{k}'\ ((c'\ (v\ e_1))^{\div}\ \mathbf{id})/\mathbf{c}]\ \mathbf{id}\,[\lambda\, \mathbf{v}.\, \lambda\, \alpha'.\, \lambda\, \mathbf{k}'.\, \mathbf{k}'\ (\mathbf{k}\ \mathbf{v})/\mathbf{c}'] \\
&\triangleright_{\beta}^{\star}\ \lambda\, \mathbf{k}.\ e^{\div}\,[\lambda\, \mathbf{v}.\, \lambda\, \alpha'.\, \lambda\, \mathbf{k}'.\, \mathbf{k}'\ (e_1{}^{\div}\ @\ \_\ (\lambda\, \mathbf{v}_1'.\ \mathbf{v}\ \mathbf{v}_1'\ (\lambda\, \mathbf{v_1}.\ \mathbf{c}'\ \mathbf{v_1}\ \_\ \mathbf{id})))/\mathbf{c}]\ \mathbf{id}\,[\lambda\, \mathbf{v}.\, \lambda\, \alpha'.\, \lambda\, \mathbf{k}'.\, \mathbf{k}'\ (\mathbf{k}\ \mathbf{v})/\mathbf{c}'] \\
&= \lambda\, \mathbf{k}.\ e^{\div}\,[\lambda\, \mathbf{v}.\, \lambda\, \alpha'.\, \lambda\, \mathbf{k}'.\, \mathbf{k}'\ (e_1{}^{\div}\ @\ \_\ (\lambda\, \mathbf{v}_1'.\ \mathbf{v}\ \mathbf{v}_1'\ (\lambda\, \mathbf{v_1}.\ (\lambda\, \mathbf{v}.\, \lambda\, \alpha'.\, \lambda\, \mathbf{k}'.\, \mathbf{k}'\ (\mathbf{k}\ \mathbf{v}))\ \mathbf{v_1}\ \_\ \mathbf{id})))/\mathbf{c}]\ \mathbf{id} \\
&\triangleright_{\beta}^{\star}\ \lambda\, \mathbf{k}.\ e^{\div}\,[\lambda\, \mathbf{v}.\, \lambda\, \alpha'.\, \lambda\, \mathbf{k}'.\, \mathbf{k}'\ (e_1{}^{\div}\ @\ \_\ (\lambda\, \mathbf{v}_1'.\ \mathbf{v}\ \mathbf{v}_1'\ \_\ (\lambda\, \mathbf{v_1}.\ (\mathbf{k}\ \mathbf{v_1}))))/\mathbf{c}]\ \mathbf{id} \\
&\equiv \lambda\, \mathbf{k}.\ e^{\div}\ \mathbf{id}\,[\lambda\, \mathbf{v}.\, \lambda\, \alpha'.\, \lambda\, \mathbf{k}'.\, \mathbf{k}'\ (e_1{}^{\div}\ @\ \_\ (\lambda\, \mathbf{v_1}.\ \mathbf{v}\ \mathbf{v_1}\ \_\ \mathbf{k}))/\mathbf{c}]
\end{aligned}
$$

Sub-Case $v_0\ (\mathcal{S} c : A \to \alpha.\, e) \triangleright \mathcal{S} c' : B \to \alpha.\, e\,[\lambda\, v : A'.\, \langle c'\ (v_0\ v)\rangle/c]$ by (App2)

$$
\begin{aligned}
&(v_0\ (\mathcal{S} c : A \to \alpha.\, e))^{\div} \\
&= \lambda\, \mathbf{k}.\ v_0{}^{\div}\ \_\ (\lambda\, \mathbf{v_0}.\ (\mathcal{S} c : A' \to \alpha'.\, e)^{\div}\ (\lambda\, \mathbf{v_1}.\ \mathbf{v_0}\ \mathbf{v_1}\ \alpha^{+}\ \mathbf{k})) \\
&= \lambda\, \mathbf{k}.\ v_0{}^{\div}\ \_\ (\lambda\, \mathbf{v_0}.\ (\lambda\, \mathbf{k}.\ e^{\div}\ \mathbf{id}\,[\lambda\, \mathbf{v}.\, \lambda\, \alpha'.\, \lambda\, \mathbf{k}'.\, \mathbf{k}'\ (\mathbf{k}\ \mathbf{v})/\mathbf{c}])\ (\lambda\, \mathbf{v_1}.\ \mathbf{v_0}\ \mathbf{v_1}\ \alpha^{+}\ \mathbf{k})) \\
&\triangleright_{\beta}^{\star}\ \lambda\, \mathbf{k}.\ v_0{}^{\div}\ \_\ (\lambda\, \mathbf{v_0}.\ e^{\div}\ \mathbf{id}\,[\lambda\, \mathbf{v}.\, \lambda\, \alpha'.\, \lambda\, \mathbf{k}'.\, \mathbf{k}'\ (\mathbf{v_0}\ \mathbf{v}\ \alpha^{+}\ \mathbf{k})/\mathbf{c}])
\end{aligned}
$$

$$\begin{aligned}
&(\mathcal{S}c' : B \to \alpha.\, e\,[\lambda\, v : A'.\, \langle c'\ (v_0\ v)\rangle / c])^{\div} \\
&= \lambda\, k.\ (e\,[\lambda\, v : A'.\, \langle c'\ (v_0\ v)\rangle / c])^{\div}\ \mathbf{id}\,[\lambda\, v.\, \lambda\, k'.\, \lambda\, \alpha'.\, k'\ (k\ v)/c'] \\
&= \lambda\, k.\ e^{\div}\,[\lambda\, v.\, \lambda\, \alpha'.\, \lambda\, k'.\, k'\ ((c'\ (v_0\ v))^{\div}\ \mathbf{id})/c]\ \mathbf{id}\,[\lambda\, v.\, \lambda\, k'.\, \lambda\, \alpha'.\, k'\ (k\ v)/c'] \\
&\triangleright_\beta^\star \lambda\, k.\ e^{\div}\,[\lambda\, v.\, \lambda\, \alpha'.\, \lambda\, k'.\, k'\ ((v_0{}^{\div}\ \_\ (\lambda\, v_0'.\, v_0'\ v\ \_\ (\lambda\, v_1.\, c'\ v_1\ \_\ \mathbf{id})))\ )/c]\ \mathbf{id}\,[\lambda\, v.\, \lambda\, k'.\, \lambda\, \alpha'.\, k'\ (k\ v)/c'] \\
&= \lambda\, k.\ e^{\div}\,[\lambda\, v.\, \lambda\, \alpha'.\, \lambda\, k'.\, k'\ (v_0{}^{\div}\ \_\ (\lambda\, v_0'.\, v_0'\ v\ \_\ (\lambda\, v_1.\, (\lambda\, v.\, \lambda\, k'.\, \lambda\, \alpha'.\, k'\ (k\ v))\ v_1\ \_\ \mathbf{id})))/c]\ \mathbf{id} \\
&\triangleright_\beta^\star \lambda\, k.\ e^{\div}\,[\lambda\, v.\, \lambda\, \alpha'.\, \lambda\, k'.\, k'\ (v_0{}^{\div}\ \_\ (\lambda\, v_0'.\, v_0'\ v\ \_\ (\lambda\, v_1.\, k\ v_1)))/c]\ \mathbf{id} \\
&\equiv \lambda\, k.\ v_0{}^{\div}\ \_\ (\lambda\, v_0.\, e^{\div}\ \mathbf{id}\,[\lambda\, v.\, \lambda\, \alpha'.\, \lambda\, k'.\, k'\ (v_0\ v\ \alpha^{+}\ k)/c])
\end{aligned}$$

Sub-Case $\langle \mathcal{S}c : A \to \alpha.\, e\rangle \triangleright \langle e\,[\lambda\, x : A.\, x/c]\rangle$

$$\begin{aligned}
&(\langle \mathcal{S}c : A \to \alpha.\, e\rangle)^{\div} \\
&= \lambda\, \alpha.\, \lambda\, k.\, k\ ((\mathcal{S}c : A \to \alpha.\, e)^{\div}\ \mathbf{id}) \\
&= \lambda\, \alpha.\, \lambda\, k.\, k\ ((\lambda\, k.\, e^{\div}\ \mathbf{id}\,[\lambda\, x.\, \lambda\, \alpha'.\, \lambda\, k'.\, k'\ (k\ x)/c])\ \mathbf{id}) \\
&\triangleright_\beta \lambda\, \alpha.\, \lambda\, k.\, k\ (e^{\div}\ \mathbf{id}\,[\lambda\, x.\, \lambda\, \alpha'.\, \lambda\, k'.\, k'\ x/c])
\end{aligned}$$

$$\begin{aligned}
&(\langle e\,[\lambda\, x : A.\, x/c]\rangle)^{\div} \\
&= \lambda\, \alpha.\, \lambda\, k.\, k\ ((e\,[\lambda\, x : A.\, x/c])^{\div}\ \mathbf{id}) \\
&\equiv \lambda\, \alpha.\, \lambda\, k.\, k\ ((e^{\div}\,[\lambda\, x.\, \lambda\, \alpha'.\, \lambda\, k'.\, k'\ x/c])\ \mathbf{id}) && \text{by Lemma 4.2} \\
&= \lambda\, \alpha.\, \lambda\, k.\, k\ (e^{\div}\ \mathbf{id}\,[\lambda\, x.\, \lambda\, \alpha'.\, \lambda\, k'.\, k'\ x/c]) && \text{by substitution}
\end{aligned}$$

Note that the domain of $\mathbf{id}$ in the second-last line is $(B\,[\lambda\, x : A.\, x/c])^{+}$, and the domain of $\mathbf{id}$ in the last line is $B^{+}\,[\lambda\, x : A^{+}.\, \lambda\, \alpha' : *.\, \lambda\, k : A^{+} \to \alpha'.\, k'\ x/c]$. These types are equivalent by Lemma 4.2.

Sub-Case $\langle v\rangle \triangleright v$

$$\langle v\rangle^{\div} = \lambda\, \alpha.\, \lambda\, k.\, k\ (v^{\div}\ \mathbf{id}) \qquad \equiv \lambda\, \alpha.\, \lambda\, k.\, k\ (v^{\div}\ \_\ \mathbf{id}) \qquad \text{by Lemma 1.14}$$

$$\begin{aligned}
v^{\div} &= \lambda\, \alpha.\, \lambda\, k.\, v^{\div}\ @\ \alpha\ (\lambda\, v.\, k\ v) && \text{by Lemma 1.14 and 4.1} \\
&\equiv \lambda\, \alpha.\, \lambda\, k.\ (\lambda\, v.\, k\ v)\ (v^{\div}\ \_\ \mathbf{id}) && \text{by Lemma 4.2} \\
&\triangleright_\beta \lambda\, \alpha.\, \lambda\, k.\, k\ (v^{\div}\ \_\ \mathbf{id})
\end{aligned}$$

Lemma 4.4 (Coherence).

(1) $\kappa \equiv \kappa' \Rightarrow \kappa^{+} \equiv \kappa'^{+}$ (2) $A \equiv A' \Rightarrow A^{+} \equiv A'^{+}$ (3) $e \equiv e' \Rightarrow e^{\div} \equiv e'^{\div}$

In both $\lambda_{\Pi}^{s/r}$ and λ_{Π}^{k}, equivalence is defined in terms of reduction. Since the translation is correct (Lemma 4.3), equivalent expressions in $\lambda_{\Pi}^{s/r}$ are mapped to equivalent expressions in λ_{Π}^{k}.

Theorem 4.5 (Type Preservation).

(1) $\vdash \Psi \Rightarrow \vdash \Psi^+$ (4) $\Gamma \vdash A : * \Rightarrow \Gamma^+ \vdash A^+ : *$
(2) $\vdash \Gamma \Rightarrow \vdash \Gamma^+$ (5) $\Gamma \vdash_p e : A \Rightarrow \Gamma^+ \vdash e^{\div} : \Pi\,\alpha : *.\,(A^+ \rightarrow \alpha) \rightarrow \alpha$
(3) $\Gamma \vdash \kappa \Rightarrow \Gamma^+ \vdash \kappa^+$ (6) $\Gamma;\alpha \vdash e : A;\beta \Rightarrow \Gamma^+ \vdash e^{\div} : (A^+ \rightarrow \alpha^+) \rightarrow \beta^+$

The proof is by mutual induction on the derivation. Here we give some interesting cases:

Case Part 1
Sub-Case (ExtendSig)
We must show

$$\vdash \Psi^+, \mathsf{Ind}(\mathbf{D} : (\Pi\, a : A.\, *)^+, \{\mathbf{c_i} : \Pi\, \mathbf{y_i} : B_i{}^+.\, (D\ u_i)^+\})$$

By the induction hypothesis, we have $\vdash \Psi^+$, $\bullet \vdash (\Pi\, a : A.\, *)^+$, and $\bullet \vdash \Pi\, \mathbf{y_i} : B_i{}^+.\, (D\ u_i)^+ : *$. What remains to check is the two additional restrictions in the premise of [ExtendSig]. The first one, freshness of names, is trivially satisfied. The second one, strict positivity, is also satisfied, thanks to the safety condition of $\lambda_\Pi^{s/r}$. Suppose B_i is of the form $\Pi\, x : B_1.\, \alpha \parallel B_2 \parallel \beta$. This type is converted into $\Pi\, \mathbf{x} : B_1{}^+.\, (B_2{}^+ \rightarrow \alpha^+) \rightarrow \beta^+$. We find that $B_1{}^+$, $B_2{}^+$, and α^+ appear in a negative position, but the safety condition safe(D, B_i) guarantees that D does not occur in these types. This implies that strictly-positive$(\mathbf{D}, B_i{}^{\div})$ holds.

Case (Extend2)
We must show

$$\vdash \Gamma^+, \mathbf{x} = e^{\div}\ A^+\ \mathbf{id} : A^+$$

By the induction hypothesis, we have $\vdash \Gamma^+$, $\Gamma^+ \vdash A^+ : *$, and $\Gamma^+ \vdash e^{\div} : \Pi\,\alpha : *.\,(A^+ \rightarrow \alpha) \rightarrow \alpha$. The last hypothesis implies $\Gamma^+ \vdash e^{\div}\ A^+\ \mathbf{id} : A^+$. The goal follows by [Extend2].

Case Part 4
Sub-Case (Data)
We must show

$$\Gamma^+ \vdash \mathbf{D}\ e_i{}^{\div}\ A_i{}^+\ \mathbf{id} : *$$

Since $D : \Pi\ y_i : A_i.\, *$, we have $\mathbf{D} : (\Pi\ y_i : A_i.\, *)^+ = \Pi\, \overline{\mathbf{y_i} : A_i{}^+}.\, *$. By [Data], we know that all e_i are pure, and the induction hypothesis gives us $e_i{}^{\div}\ A_i{}^+\ \mathbf{id} : A_i{}^+$. This implies that the translated type is well-formed.

Sub-Case (PiT2)
We must show

$$\Gamma^+ \vdash \Pi\, \mathbf{x} : A^+.\, (B^+ \rightarrow \alpha^+) \rightarrow \beta^+ : *$$

By the induction hypothesis, we have $\Gamma^+ \vdash A^+ : *$, $\Gamma^+, \mathbf{x} : A^+ \vdash B^+ : *$, and similarly for α^+ and β^+. The goal follows by [PiT1].

Case Part 5
Sub-Case (Var)
Our goal is to show

$$\Gamma^+ \vdash \lambda\,\alpha : *.\, \lambda\, \mathbf{k} : A^+ \rightarrow \alpha.\, \mathbf{k}\ \mathbf{x} : \Pi\,\alpha : *.\,(A^+ \rightarrow \alpha) \rightarrow \alpha$$

By the induction hypothesis, we have $\vdash \Gamma^+$ and $\mathbf{x} : A^+ \in \Gamma^+$ or $\mathbf{x} = e^{\div}\ A^+\ \mathbf{id} : A^+ \in \Gamma^+$. By [Var], we have $\Gamma^+ \vdash \mathbf{x} : A^+$, which means the application $\mathbf{k}\ \mathbf{x}$ is well-typed. Now the goal immediately follows.

Sub-Case (Abs2)
Our goal is to show

$$\lambda\,\alpha : *.\, \lambda\, \mathbf{k} : (\Pi\, x : A.\, \alpha \parallel B \parallel \beta)^+ \rightarrow \alpha.\, \mathbf{k}\ (\lambda\, \mathbf{x} : A^+.\, e^{\div})$$

has type

$$\Pi\,\alpha : *.\ ((\Pi\, x : A.\ \alpha \parallel B \parallel \beta)^+ \to \alpha) \to \alpha$$

under context Γ^+. By the induction hypothesis, $\Gamma^+, \mathbf{x} : A^+ \vdash e^{\div} : (B^+ \to \alpha^+) \to \beta^+$. The goal follows by [Abs1].

Sub-Case (App1)

Our goal is to show the term

$$\lambda\,\alpha : *.\ \lambda\,\mathbf{k} : (B\,[e_1/x])^+ \to \alpha.\ e_0{}^{\div}\ \alpha\ (\lambda\,\mathbf{v_0} : (\Pi\, x : A.\ B)^+.\ e_1{}^{\div}\ @\ \alpha\ (\lambda\,\mathbf{v_1} : A^+.\ \mathbf{v_0}\ \mathbf{v_1}\ \alpha\ \mathbf{k}))$$

has type $\Pi\,\alpha : *.\ ((B\,[e_1/x])^+ \to \alpha) \to \alpha$ under context Γ^+. By compositionality (Lemma 4.2), we have $(B\,[e_1/x])^+ \equiv B^+\,[e_1{}^{\div}\ A^+\ \mathbf{id}/\mathbf{x}]$. By Part 5 of the induction hypothesis, we have $e_0{}^{\div} : \Pi\,\alpha : *.\ ((\Pi\, x : A.\ B)^+ \to \alpha) \to \alpha$ and $e_1{}^{\div} : \Pi\,\alpha : *.\ (A^+ \to \alpha) \to \alpha$. These imply $\mathbf{v_0}\ \mathbf{v_1} : B^+\,[\mathbf{v_1}/\mathbf{x}]$. Since we type application of $e_1{}^{\div}$ to its continuation using [T-Cont], we can type $\mathbf{v_0}\ \mathbf{v_1}$ under the assumption that $\mathbf{v_1} = e_1{}^{\div}\ A^+\ \mathbf{id}$. This implies that $\mathbf{v_0}\ \mathbf{v_1}\ \mathbf{k}$ is well-typed.

Sub-Case (Match1)

Our goal is to show the term

$$\begin{array}{l}\lambda\,\alpha : *.\ \lambda\,\mathbf{k} : (P\,[u/a,\ e/x])^+ \to \alpha.\\ \quad e^{\div}\ @\ \alpha\ (\lambda\,\mathbf{v} : (D\ u)^+.\ \mathbf{match\ v\ as\ x\ in\ D\ a\ return}\ \alpha\ \mathbf{with}\ \{\mathbf{c_i}\ \overline{\mathbf{y_i}} \to e_i{}^{\div}\ \alpha\ \mathbf{k}\})\end{array}$$

has type

$$\Pi\,\alpha : *.\ ((P\,[u/a,\ e/x])^+ \to \alpha) \to \alpha$$

under context Γ^+. By [T-Cont] and [Match], it suffices to show that each branch $e_i{}^{\div}\ \alpha\ \mathbf{k}$ has type α under Γ^+ extended with the following information:

$$\alpha : *,\ \mathbf{k} : (P\,[u/a,\ e/x])^+ \to \alpha,\ \mathbf{v} = e^{\div}\ (D\ u)^+\ \mathbf{id} : (D\ u)^+,\ \mathbf{y_i} : B_i{}^+,\ u^{\div}\ A^+\ \mathbf{id} \equiv u_i{}^{\div}\ A^+\ \mathbf{id},\ \mathbf{v} \equiv \mathbf{c_i}\ \mathbf{y_i}$$

By the induction hypothesis, $e_i{}^{\div}$ has the following type:

$$\Pi\,\alpha : *.\ ((P\,[u_i/a,\ c_i\ y_i/x])^+ \to \alpha) \to \alpha$$

Now, let us prove that $(P\,[u_i/a,\ c_i\ y_i/x])^+$ is equivalent to the domain of the continuation $\mathbf{k}$:

$$\begin{array}{rll}
(P\,[u_i/a,\ c_i\ y_i/x])^+ & \equiv P^+\,[u_i{}^{\div}\ \_\ \mathbf{id}/\mathbf{a},\ (c_i\ y_i)^{\div}\ \_\ \mathbf{id}/\mathbf{x}] & \text{by Lemma 4.2}\\
& \triangleright_{\beta}^{\star}\ P^+\,[u_i{}^{\div}\ \_\ \mathbf{id}/\mathbf{a},\ \mathbf{c_i}\ \mathbf{y_i}/\mathbf{x}] & \\
& \equiv P^+\,[u_i{}^{\div}\ \_\ \mathbf{id}/\mathbf{a},\ \mathbf{v}/\mathbf{x}] & \text{by } \mathbf{v} \equiv \mathbf{c_i}\ \mathbf{y_i}\\
& \equiv P^+\,[u^{\div}\ \_\ \mathbf{id}/\mathbf{a},\ \mathbf{v}/\mathbf{x}] & \text{by } u^{\div}\ \_\ \mathbf{id} \equiv u_i{}^{\div}\ \_\ \mathbf{id}\\
& \triangleright_{\delta}\ P^+\,[u^{\div}\ \_\ \mathbf{id}/\mathbf{a},\ e^{\div}\ (D\ u)^+\ \mathbf{id}/\mathbf{x}] & \\
& \equiv (P\,[u/a,\ e/x])^+ & \text{by Lemma 4.2}
\end{array}$$

The rule [Match] gives us the information $u^{\div}\ \_\ \mathbf{id} \equiv u_i{}^{\div}\ \_\ \mathbf{id}$ and $\mathbf{v} \equiv \mathbf{c_i}\ \mathbf{y_i}$. Since (Match1) requires a pure scrutinee e, we further know that $\mathbf{v}$ is the result of running $e^{\div}$ with the identity continuation. This knowledge is made explicit by [T-Cont], allowing us to δ-reduce $\mathbf{v}$ to $e^{\div}\ (D\ u)^+\ \mathbf{id}$. Thus we conclude that the translation of the matching construct has the correct type.

Sub-Case (Reset)

Our goal is to show

$$\Gamma^+ \vdash e^{\div}\ (\lambda\,\mathbf{x} : B^+.\ \mathbf{x}) : \Pi\,\alpha : *.\ (A^+ \to \alpha) \to \alpha$$

By Part 5 of the induction hypothesis, we have

$$\Gamma^+ \vdash e^{\div} : (B^+ \to B^+) \to A^+$$

The goal immediately follows.

Sub-Case (CONV1)

Our goal is to show

$$\Gamma^+ \vdash e^{\div} : \Pi\, \alpha : *. \, (B^+ \to \alpha) \to \alpha$$

By Part 5 of the induction hypothesis, we have

$$\Gamma^+ \vdash e^{\div} : \Pi\, \alpha : *. \, (A^+ \to \alpha) \to \alpha$$

The goal follows by the coherence property (Lemma 4.4).

Case Part 6

Sub-Case (LET2)

Our goal is to show

$$\lambda\, \mathbf{k} : (B\,[e_1/x])^+ \to (\alpha\,[e_1/x])^+ .\, e_1{}^{\div} \mathbin{@} (\beta\,[e_1/x])^+ \; (\lambda\, \mathbf{x} : A^+ .\, e_2{}^{\div}\, \mathbf{k})$$

has type

$$((B\,[e_1/x])^+ \to (\alpha\,[e_1/x])^+) \to (\beta\,[e_1/x])^+$$

under context Γ^+. By compositionality, we have $(B\,[e_1/x])^+ \equiv B^+\,[e_1{}^{\div}\, A^+\, \mathbf{id}/\mathbf{x}]$ and similarly for the answer types. By Parts 5 and 6 of the induction hypothesis, we have

$$\Gamma^+ \vdash e_1{}^{\div} : \Pi\, \alpha : *. \, (A^+ \to \alpha) \to \alpha$$

and

$$\Gamma^+, \mathbf{x} = e_1{}^{\div}\, A^+\, \mathbf{id} : A^+ \vdash e_2{}^{\div} : (B^+ \to \alpha^+) \to \beta^+$$

Since we type the application of $e_1{}^{\div}$ using [T-CONT], we can type $e_2{}^{\div}\, \mathbf{k}$ under the assumption that $\mathbf{x} = e_1{}^{\div}\, A^+\, \mathbf{id}$. This implies that $e_2{}^{\div}\, \mathbf{k}$ is type-safe.

5 CPS TRANSLATION OF λ_Π^{s/r^+}

This section shows how to scale the CPS translation to multi-arity inductive datatypes. As in $\lambda_\Pi^{s/r}$, the key idea is to convert pure arguments using the polymorphic answer type translation so that we can use the free theorem to reason about their CPS images.

Extending the translation is straightforward in most cases; the only case that requires non-trivial reasoning is (CONST2). This rule concludes with a constructor application, where some arguments are pure and some are impure. Recall that a pure argument e_i may appear in the type of subsequent arguments as well as the result type. This means we have to give $e_i{}^{\div}$ a polymorphic type, and instantiate its answer type when applying $e_i{}^{\div}$ to its continuation. In our translation, we use a new application form $\mathbf{e} \bullet_{\mathbf{a}} \mathbf{k}$ to perform answer type instantiation. The application is defined as follows:

- When e_i is impure, $e_i{}^{\div} \bullet_{\mathbf{a}} \mathbf{k} = e_i{}^{\div}\, \mathbf{k}$.
- When e_i is pure and there is an impure argument following e_i, $e_i{}^{\div} \bullet_{\mathbf{a}} \mathbf{k} = e_i{}^{\div} \mathbin{@} \beta_j{}^+\, \mathbf{k}$ or $e_i{}^{\div}\, \beta_j{}^+\, \mathbf{k}$, where e_j is the closest impure argument following e_i.
- When e_i is pure and there is an impure argument preceding e_i, $e_i{}^{\div} \bullet_{\mathbf{a}} \mathbf{k} = e_i{}^{\div} \mathbin{@} \alpha_j{}^+\, \mathbf{k}$ or $e_i{}^{\div}\, \alpha_j{}^+\, \mathbf{k}$ where e_j is the closest impure argumet preceding e_i.

Since the typing rule applies when at least one argument is impure, one of the above three cases must hold. If e_i is impure, there is no need to instantiate the answer type, hence we simply apply $e_i{}^{\div}$ to $\mathbf{k}$. If there is an impure argument e_j following e_i, the application of $e_j{}^{\div}$ to its continuation must have type $\beta_j{}^+$, where β_j is the final answer type of e_j. This type will be the return type of the continuation passed to $e_i{}^{\div}$, therefore we instantiate the answer type of $e_i{}^{\div}$ to $\beta_j{}^+$. Note that we use $@$ when e_i appears in types, that is, when the constructor's i'th domain is $y_i : B_i$ and y_i is bound by Π. If there is an impure argument e_j preceding e_i, $e_j{}^{\div}$ requires a continuation that returns $\alpha_j{}^+$,

$$(\textsc{ExtendSig})\ \overset{+}{\rightsquigarrow}\ \Psi^+, \mathsf{Ind}(\mathbf{D} : (\Pi\,\overline{a_i : A_i}.\,*)^+, \{\mathbf{c_i} : \Pi\,\overline{\mathbf{y_i} : B_i{}^+}.\,(D\ \overline{u_i})^+\})$$

$$(\textsc{PiK})\ \overset{+}{\rightsquigarrow}\ \Pi\,\overline{\mathbf{x_i} : A_i{}^+}.\,*$$

$$(\textsc{Pi1})\ \overset{+}{\rightsquigarrow}\ \Pi\,\overline{\mathbf{x_i} : A_i{}^+}.\,B^+ \qquad (\textsc{Data})\ \overset{+}{\rightsquigarrow}\ \mathbf{D}\ \overline{(e_i{}^{\div}\ (A_i\ \overline{[e_j/x_j]}^{i}_{p})^+\ \mathbf{id})}$$

$$(\textsc{Const1})\ \overset{\div}{\rightsquigarrow}\ \begin{aligned}&\lambda\alpha:*.\,\lambda\mathbf{k}:(D\ (u_i\ [e/y_i]))^+\to\alpha.\\ &\quad e_1{}^{\div}\ @\ \alpha\ (\lambda\mathbf{v_1}:B_1{}^+.\ \dots\ e_n{}^{\div}\ @\ \alpha\ (\lambda\mathbf{v_n}:(B_n\ \overline{[e_j/x_j]}^{n}_{p})^+.\ \mathbf{k}\ (\mathbf{c_i}\ \overline{\mathbf{v_i}})))\end{aligned}$$

$$(\textsc{Const2})\ \overset{\div}{\rightsquigarrow}\ \begin{aligned}&\lambda\mathbf{k}:(D\ (u_i\ [e/y_i]))^+\to\alpha^+.\\ &\quad e_1{}^{\div}\ \bullet_a\ (\lambda\mathbf{v_1}:B_1{}^+.\ \dots\ e_n{}^{\div}\ \bullet_a\ (\lambda\mathbf{v_n}:(B_n\ \overline{[e_j/x_j]}^{n}_{p})^+.\ \mathbf{k}\ (\mathbf{c_i}\ \overline{\mathbf{v_i}})))\end{aligned}$$

$$(\textsc{Match1})\ \overset{\div}{\rightsquigarrow}\ \begin{aligned}&\lambda\alpha:*.\,\lambda\mathbf{k}:(P\ \overline{[u/a_i]}\ [e/x])^+\to\alpha.\\ &\quad e^{\div}\ @\ \alpha\ (\lambda\mathbf{v}:(D\ u)^+.\ \mathbf{match\ v\ as\ x\ in\ D}\ \overline{\mathbf{a_i}}\ \mathbf{return}\ \alpha\ \mathbf{with}\ \{\mathbf{c_i}\ \overline{\mathbf{y_i}}\to e_i{}^{\div}\ \alpha\ \mathbf{k}\})\end{aligned}$$

$$(\textsc{Match2})\ \overset{\div}{\rightsquigarrow}\ \begin{aligned}&\lambda\mathbf{k}:(P\ \overline{[u/a_i]}\ [e/x])^+\to(\alpha\ \overline{[u/a_i]}\ [e/x])^+.\\ &\quad e^{\div}\ @\ (\beta\ \overline{[u/a_i]}\ [e/x])^+\ (\lambda\mathbf{v}:(D\ u)^+.\ \mathbf{match\ v\ as\ x\ in\ D}\ \overline{\mathbf{a_i}}\ \mathbf{return}\ \beta^+\ \mathbf{with}\ \{\mathbf{c_i}\ \overline{\mathbf{y_i}}\to e_i{}^{\div}\ \mathbf{k}\})\end{aligned}$$

$$(\textsc{Match3})\ \overset{\div}{\rightsquigarrow}\ \begin{aligned}&\lambda\mathbf{k}:P^+\to\alpha^+.\\ &\quad e^{\div}\ (\lambda\mathbf{v}:(D\ u)^+.\ \mathbf{match\ v\ as\ \_\ in\ D\ \_\ return}\ \beta^+\ \mathbf{with}\ \{\mathbf{c_i}\ \overline{\mathbf{y_i}}\to e_i{}^{\div}\ \mathbf{k}\})\end{aligned}$$

Fig. 9. CPS Translation of $\lambda_{\Pi}^{s/r+}$

where α_j is the initial answer type of e_j. As e_i is pure and thus does not change the answer type, application $e_i{}^{\div}\ @\ \alpha_j{}^+\ \mathbf{k}$ has type $\alpha_j{}^+$, which guarantees that $e_j{}^{\div}\ (\lambda\,\mathbf{v_j}.\ e_i{}^{\div}\ @\ \alpha_j{}^+\ \mathbf{k})$ is well-typed.

The translation of $\lambda_{\Pi}^{s/r+}$ signatures, kindes, types, and terms are presented in Figure 9.

REFERENCES

Kenichi Asai and Yukiyoshi Kameyama. 2007. Polymorphic Delimited Continuations. In *Proceedings of the 5th Asian Conference on Programming Languages and Systems (APLAS '07)*. Springer-Verlag, Berlin, Heidelberg, 239–254. http://dl.acm.org/citation.cfm?id=1784774.1784797

Masako Takahashi. 1995. Parallel reductions in λ-calculus. *Information and computation* 118, 1 (1995), 120–127.