Demand Unchanged Variable Analysis Technical Report

Chahyun Kang and Kimball Germane

Brigham Young University, Provo UT 84606, USA
chi2eut@byu.edu kimball@cs.byu.edu

1 Semantics

In this section, we first introduce several semantics for lemmas and theorems. We define both Demand Evaluation and Demand m-CFA with and without Unchanged Variable Analysis.

1.1 Demand Evaluation

$$\textsc{Bind-Rat}\quad \frac{x, \rho \vdash C[(e_0\, e_1)] \Uparrow_{b/d} v}{x, \rho \vdash C[([e_0]\, e_1)] \Uparrow_{b/d} v}$$

$$\textsc{Bind-Ran}\quad \frac{x, \rho \vdash C[(e_0\, e_1)] \Uparrow_{b/d} v}{x, \rho \vdash C[(e_0\, [e_1])] \Uparrow_{b/d} v}$$

$$\textsc{Bind-Zero}\quad \frac{}{x, \rho \vdash C[\lambda x.[e]] \Uparrow_{b/d} C[\lambda x.[e]], \rho}$$

$$\textsc{Bind-Succ}\quad \frac{x \neq y \qquad x, \rho \vdash C[\lambda y.e] \Uparrow_{b/d} v}{x, \Sigma :: \rho \vdash C[\lambda y.[e]] \Uparrow_{b/d} v}$$

$$\textsc{Ref}\quad \frac{\begin{array}{c} x, \rho_x \vdash C_x[x] \Uparrow_{b/d} (C_\lambda[\lambda x.[e]], (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda) \\ \rho_\lambda \vdash C_\lambda[\lambda x.e] \Rightarrow_d (C[([e_0]\, e_1)], \Sigma :: \rho_0) \\ \Sigma :: \rho_0 \vdash C[(e_0\, [e_1])] \Downarrow_d v \end{array}}{\rho_x \vdash C_x[x] \Downarrow_d v}$$

$$\textsc{Lam}\quad \frac{}{\rho \vdash C[\lambda x.e] \Downarrow_d C[\lambda x.e], \rho}$$

$$\textsc{App}\quad \frac{\begin{array}{c} \Sigma :: \rho \vdash C[([e_0]\, e_1)] \Downarrow_d (C_\lambda[\lambda x.e], \rho_\lambda) \\ \Sigma :: \rho \vdash C[(e_0\, [e_1])] \Downarrow_d v_1 \\ (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda \vdash C_\lambda[\lambda x.[e]] \Downarrow_d v \end{array}}{\Sigma :: \rho \vdash C[(e_0\, e_1)] \Downarrow_d v}$$

$$\textsc{Find-Rat}\quad \frac{x, \rho \vdash C[([e_0]\, e_1)] \Uparrow_{f/d} (C_x[x], \rho_x)}{x, \rho \vdash C[(e_0\, e_1)] \Uparrow_{f/d} (C_x[x], \rho_x)}$$

$$\textsc{Find-Ran}\quad \frac{x, \rho \vdash C[(e_0\, [e_1])] \Uparrow_{f/d} (C_x[x], \rho_x)}{x, \rho \vdash C[(e_0\, e_1)] \Uparrow_{f/d} (C_x[x], \rho_x)}$$

$$\textsc{Find-Bod}\quad \frac{\begin{array}{c} x \neq y \qquad \rho_\lambda \vdash C_\lambda[\lambda y.e] \Rightarrow_d (C[([e_0]\, e_1)], \Sigma :: \rho) \\ x, (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda \vdash C_\lambda[\lambda y.[e]] \Uparrow_{f/d} (C_x[x], \rho_x) \\ x, \rho \vdash C[(e_0\, [e_1])] \Uparrow_{f/d} (C_x[x], \rho_x) \end{array}}{x, \rho_\lambda \vdash C_\lambda[\lambda y.e] \Uparrow_{f/d} (C_x[x], \rho_x)}$$

$$\textsc{Find-Ref}\quad \frac{}{\rho \vdash C[x] \Uparrow_{f/d}\ (C[x], \rho)}$$

$$\textsc{Rat}\quad \frac{}{\rho \vdash C[([e_0]\, e_1)] \Rightarrow_d (C[([e_0]\, e_1)], \rho)}$$

$$\textsc{Ran}\quad \frac{\begin{array}{c} \Sigma :: \rho \vdash C[([e_0]\, e_1)] \Downarrow_d (C_\lambda[\lambda x.e], \rho_\lambda) \\ x, (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda \vdash C_\lambda[\lambda x.[e]] \Uparrow_{f/d} C_x[x], \rho_x \\ \rho_x \vdash C_x[x] \Rightarrow_d c \end{array}}{\Sigma :: \rho \vdash C[(e_0\, [e_1])] \Rightarrow_d c}$$

$$\textsc{Bod}\quad \frac{\rho_\lambda \vdash C_\lambda[\lambda x.e] \Rightarrow_d (C[([e_0]\, e_1)], \rho) \qquad \rho \vdash C[(e_0\, e_1)] \Rightarrow_d c}{(C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda \vdash C_\lambda[\lambda x.[e]] \Rightarrow_d c}$$

1.2 Demand Evaluation With UVA

Bind-Rat
$$\frac{x, \rho \vdash C[(e_0\, e_1)] \Uparrow^{uva}_{b/d} v, \psi}{x, \rho \vdash C[([e_0]\, e_1)] \Uparrow^{uva}_{b/d} v, \psi}$$

Bind-Ran
$$\frac{x, \rho \vdash C[(e_0\, e_1)] \Uparrow^{uva}_{b/d} v, \psi}{x, \rho \vdash C[(e_0\, [e_1])] \Uparrow^{uva}_{b/d} v, \psi}$$

Bind-Zero
$$\frac{}{x, \rho \vdash C[\lambda x.[e]] \Uparrow^{uva}_{b/d} (C[\lambda x.[e]], \rho), \emptyset}$$

Bind-Succ
$$\frac{x \neq y \qquad x, \rho \vdash C[\lambda y.e] \Uparrow^{uva}_{b/d} v, \psi}{x, \Sigma :: \rho \vdash C[\lambda y.[e]] \Uparrow^{uva}_{b/d} v, \psi \cup \{x\}}$$

Ref
$$\frac{\begin{array}{c} x, \rho_x \vdash C_x[x] \Uparrow^{uva}_{b/d} (C_\lambda[\lambda x.[e]], (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda), \psi \\ \rho_\lambda \vdash C_\lambda[\lambda x.e] \Rightarrow^{uva}_{d} (C[([e_0]\, e_1)], \Sigma :: \rho_0), \psi' \\ \Sigma :: \rho_0 \vdash C[(e_0\, [e_1])] \Downarrow^{uva}_{d} v, \psi'' \end{array}}{\rho_x \vdash C_x[x] \Downarrow^{uva}_{d} v, \psi \cup \psi' \cup \psi'' \cup \{x\}}$$

Lam
$$\frac{}{\rho \vdash C[\lambda x.e] \Downarrow^{uva}_{d} C[\lambda x.e], \rho, \emptyset}$$

App
$$\frac{\begin{array}{c} \Sigma :: \rho \vdash C[([e_0]\, e_1)] \Downarrow^{uva}_{d} (C_\lambda[\lambda x.e], \rho_\lambda), \psi \\ \Sigma :: \rho \vdash C[(e_0\, [e_1])] \Downarrow^{uva}_{d} v_1, \psi' \\ (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda \vdash C_\lambda[\lambda x.[e]] \Downarrow^{uva}_{d} v\ \psi'' \end{array}}{\Sigma :: \rho \vdash C[(e_0\, e_1)] \Downarrow^{uva}_{d} v, \psi \cup \psi''}$$

Find-Rat
$$\frac{x, \rho \vdash C[([e_0]\, e_1)] \Uparrow^{uva}_{f/d} (C_x[x], \rho_x), \psi}{x, \rho \vdash C[(e_0\, e_1)] \Uparrow^{uva}_{f/d} (C_x[x], \rho_x), \psi}$$

Find-Ran
$$\frac{x, \rho \vdash C[(e_0\, [e_1])] \Uparrow^{uva}_{f/d} (C_x[x], \rho_x), \psi}{x, \rho \vdash C[(e_0\, e_1)] \Uparrow^{uva}_{f/d} (C_x[x], \rho_x), \psi}$$

Find-Bod
$$\frac{\begin{array}{c} x \neq y \qquad \rho_\lambda \vdash C_\lambda[\lambda y.e] \Rightarrow^{uva}_{d} (C[([e_0]\, e_1)], \Sigma :: \rho), \psi \\ x, (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda \vdash C_\lambda[\lambda y.[e]] \Uparrow^{uva}_{f/d} (C_x[x], \rho_x), \psi' \\ x, \rho \vdash C[(e_0\, [e_1])] \Uparrow^{uva}_{f/d} (C_x[x], \rho_x), \psi'' \end{array}}{x, \rho_\lambda \vdash C_\lambda[\lambda y.e] \Uparrow^{uva}_{f/d} (C_x[x], \rho_x), \psi'}$$

Find-Ref
$$\frac{}{\rho \vdash C[x] \Uparrow^{uva}_{f/d} (C[x], \rho), \emptyset}$$

Rat
$$\frac{}{\rho \vdash C[([e_0]\, e_1)] \Rightarrow^{uva}_{d} (C[([e_0]\, e_1)], \rho), \emptyset}$$

Ran
$$\frac{\begin{array}{c} \Sigma :: \rho \vdash C[([e_0]\, e_1)] \Downarrow^{uva}_{d} (C_\lambda[\lambda x.e], \rho_\lambda), \psi \\ x, (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda \vdash C_\lambda[\lambda x.[e]] \Uparrow^{uva}_{f/d} (c_x[x], \rho_x), \psi' \\ \rho_x \vdash C_x[x] \Rightarrow^{uva}_{d} c, \psi'' \end{array}}{\Sigma :: \rho \vdash C[(e_0\, [e_1])] \Rightarrow^{uva}_{d} c, \psi \cup \psi' \cup \psi'' \cup \{x\}}$$

Bod
$$\frac{\rho_\lambda \vdash C_\lambda[\lambda x.e] \Rightarrow^{uva}_{d} (C[([e_0]\, e_1)], \rho), \psi \qquad \rho \vdash C[(e_0\, e_1)] \Rightarrow^{uva}_{d} c, \psi'}{(C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda \vdash C_\lambda[\lambda x.[e]] \Rightarrow^{uva}_{d} c, \psi \cup \psi' \cup \{x\}}$$

1.3 Demand m-CFA

$$\textsc{Bind-Rat}\quad \frac{x, \rho \vdash C[(e_0\, e_1)] \Uparrow_{b/m} v}{x, \rho \vdash C[([e_0]\, e_1)] \Uparrow_{b/m} v} \qquad \textsc{Bind-Ran}\quad \frac{x, \rho \vdash C[(e_0\, e_1)] \Uparrow_{b/m} v}{x, \rho \vdash C[(e_0\, [e_1])] \Uparrow_{b/m} v}$$

$$\textsc{Bind-Zero}\quad \frac{}{x, \rho \vdash C[\lambda x.[e]] \Uparrow_{b/m} (C[\lambda x.[e]], \rho)} \qquad \textsc{Bind-Succ}\quad \frac{x \neq y \qquad x, \rho \vdash C[\lambda y.e] \Uparrow_{b/m} v}{x, \Sigma :: \rho \vdash C[\lambda y.[e]] \Uparrow_{b/m} v}$$

$$\textsc{Ref}\quad \frac{\begin{array}{c} x, \rho_x \vdash C_x[x] \Uparrow_{b/m} (C_\lambda[\lambda x.[e]], (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda) \\ \rho_\lambda \vdash C_\lambda[\lambda x.e] \Rightarrow_m (C[([e_0]\, e_1)], \Sigma :: \rho_0) \qquad \Sigma :: \rho_0 \vdash C[(e_0\, [e_1])] \Downarrow_m v \end{array}}{\rho_x \vdash C_x[x] \Downarrow_m v}$$

$$\textsc{Lam}\quad \frac{}{\rho \vdash C[\lambda x.e] \Downarrow_m (C[\lambda x.e], \rho)} \qquad \textsc{App}\quad \frac{\begin{array}{c} \Sigma :: \rho \vdash C[([e_0]\, e_1)] \Downarrow_m (C_\lambda[\lambda x.e], \rho_\lambda) \\ \lfloor (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda \rfloor_m \vdash C_\lambda[\lambda x.[e]] \Downarrow_m v \end{array}}{\Sigma :: \rho \vdash C[(e_0\, e_1)] \Downarrow_m v}$$

$$\textsc{Find-Rat}\quad \frac{x, \rho \vdash C[([e_0]\, e_1)] \Uparrow_{f/m} (C_x[x], \rho_x)}{x, \rho \vdash C[(e_0\, e_1)] \Uparrow_{f/m} (C_x[x], \rho_x)} \qquad \textsc{Find-Ran}\quad \frac{x, \rho \vdash C[(e_0\, [e_1])] \Uparrow_{f/m} (C_x[x], \rho_x)}{x, \rho \vdash C[(e_0\, e_1)] \Uparrow_{f/m} (C_x[x], \rho_x)}$$

$$\textsc{Find-Bod}\quad \frac{\begin{array}{c} x \neq y \\ x, \lfloor (x?) :: \rho_\lambda \rfloor_m \vdash C_\lambda[\lambda y.[e]] \Uparrow_{f/m} (C_x[x], \rho_x) \end{array}}{x, \rho_\lambda \vdash C_\lambda[\lambda y.e] \Uparrow_{f/m} (C_x[x], \rho_x)} \qquad \textsc{Find-Ref}\quad \frac{}{\rho \vdash C[x] \Uparrow_{f/m} (C[x], \rho)}$$

$$\textsc{Rat}\quad \frac{}{\rho \vdash C[([e_0]\, e_1)] \Rightarrow_m (C[([e_0]\, e_1)], \rho)}$$

$$\textsc{Ran}\quad \frac{\begin{array}{c} \Sigma :: \rho \vdash C[([e_0]\, e_1)] \Downarrow_m (C_\lambda[\lambda x.e], \rho_\lambda) \\ x, \lfloor (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda \rfloor_m \vdash C_\lambda[\lambda x.[e]] \Uparrow_{f/m} C_x[x], \rho_x \qquad \rho_x \vdash C_x[x] \Rightarrow_d c \end{array}}{\Sigma :: \rho \vdash C[(e_0\, [e_1])] \Rightarrow_m c}$$

$$\textsc{Bod}\quad \frac{\begin{array}{c} \rho_\lambda \vdash C_\lambda[\lambda x.e] \Rightarrow_m (C[([e_0]\, e_1)], \rho) \\ C[(e_0\, e_1)] \sqsubset Ctx \qquad \rho \vdash C[(e_0\, e_1)] \Rightarrow_m c \end{array}}{Ctx :: \rho_\lambda \vdash C_\lambda[\lambda x.[e]] \Rightarrow_m c}$$

The following is the relation between the environments and values for Demand Evaluation and Demand m-CFA.

$$\frac{}{\epsilon \sim \epsilon} \qquad \frac{\Sigma_d \subseteq \Sigma \qquad \rho_d \sim \rho}{\Sigma_d :: \rho_d \sim \Sigma :: \rho} \qquad \frac{v_d \sqsubseteq v}{v_d \sim v}$$

Theorem 1 (Demand Evaluation and Demand m-CFA Soundness). *If $\rho \sim \rho_d$ then $\rho_d \ \vdash \ e \ \Downarrow_d \ v_d, \ \sigma \rightarrow \rho_m \ \vdash \ e \ \Downarrow_m \ v$ where $v_d \sim v$*

We use the proof from the Demand m-CFA work [1].

1.4 Demand m-CFA With UVA

Bind-Rat

$$\frac{x, \rho \vdash C[(e_0\, e_1)] \Uparrow^{uva}_{b/m} v, \psi}{x, \rho \vdash C[([e_0]\, e_1)] \Uparrow^{uva}_{b/m} v, \psi}$$

Bind-Ran

$$\frac{x, \rho \vdash C[(e_0\, e_1)] \Uparrow^{uva}_{b/m} v, \psi}{x, \rho \vdash C[(e_0\, [e_1])] \Uparrow^{uva}_{b/m} v, \psi}$$

Bind-Zero

$$\frac{}{x, \rho \vdash C[\lambda x.[e]] \Uparrow^{uva}_{b/m} (C[\lambda x.[e]], \rho), \emptyset}$$

Bind-Succ

$$\frac{x \neq y \qquad x, \rho \vdash C[\lambda y.e] \Uparrow^{uva}_{b/m} v, \psi}{x, \Sigma :: \rho \vdash C[\lambda y.[e]] \Uparrow^{uva}_{b/m} v, \psi \cup \{x\}}$$

Ref

$$\frac{\begin{array}{c} x, \rho_x \vdash C_x[x] \Uparrow^{uva}_{b/m} (C_\lambda[\lambda x.[e]], (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda), \psi \\ \rho_\lambda \vdash C_\lambda[\lambda x.e] \Rightarrow^{uva}_{m} (C[([e_0]\, e_1)], \Sigma :: \rho_0), \psi' \\ \Sigma :: \rho_0 \vdash C[(e_0\, [e_1])] \Downarrow^{uva}_{m} v, \psi'' \end{array}}{\rho_x \vdash C_x[x] \Downarrow^{uva}_{m} v, \psi \cup \psi' \cup \psi'' \cup \{x\}}$$

Lam

$$\frac{}{\rho \vdash C[\lambda x.e] \Downarrow^{uva}_{m} (C[\lambda x.e], \rho), \psi}$$

App

$$\frac{\begin{array}{c} \Sigma :: \rho \vdash C[([e_0]\, e_1)] \Downarrow^{uva}_{m} (C_\lambda[\lambda x.e], \rho_\lambda), \psi \\ \lfloor (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda \rfloor_m \vdash C_\lambda[\lambda x.[e]] \Downarrow^{uva}_{m} v, \psi' \end{array}}{\Sigma :: \rho \vdash C[(e_0\, e_1)] \Downarrow^{uva}_{m} v, \psi \cup \psi'}$$

Find-Rat

$$\frac{x, \rho \vdash C[([e_0]\, e_1)] \Uparrow^{uva}_{f/m} (C_x[x], \rho_x), \psi}{x, \rho \vdash C[(e_0\, e_1)] \Uparrow^{uva}_{f/m} (C_x[x], \rho_x), \psi}$$

Find-Ran

$$\frac{x, \rho \vdash C[(e_0\, [e_1])] \Uparrow^{uva}_{f/m} (C_x[x], \rho_x), \psi}{x, \rho \vdash C[(e_0\, e_1)] \Uparrow^{uva}_{f/m} (C_x[x], \rho_x), \psi}$$

Find-Bod

$$\frac{\begin{array}{c} x \neq y \\ x, \lfloor (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda \rfloor_m \vdash C_\lambda[\lambda y.[e]] \Uparrow^{uva}_{f/m} (C_x[x], \rho_x), \psi \end{array}}{x, \rho_\lambda \vdash C_\lambda[\lambda y.e] \Uparrow^{uva}_{f/m} (C_x[x], \rho_x), \psi}$$

Find-Ref

$$\frac{}{\rho \vdash C[x] \Uparrow^{uva}_{f/m} (C[x], \rho), \emptyset}$$

Rat

$$\frac{}{\rho \vdash C[([e_0]\, e_1)] \Rightarrow^{uva}_{m} (C[([e_0]\, e_1)], \rho), \emptyset}$$

Ran

$$\frac{\begin{array}{c} \Sigma :: \rho \vdash C[([e_0]\, e_1)] \Downarrow^{uva}_{m} (C_\lambda[\lambda x.e], \rho_\lambda), \psi \\ x, (C[(e_0\, e_1)] :: \Sigma) :: \rho_\lambda \vdash C_\lambda[\lambda x.[e]] \Uparrow^{uva}_{f/m} (C_x[x], \rho_x), \psi' \\ \rho_x \vdash C_x[x] \Rightarrow_d c, \psi'' \end{array}}{\Sigma :: \rho \vdash C[(e_0\, [e_1])] \Rightarrow^{uva}_{m} c, \psi \cup \psi' \cup \psi'' \cup \{x\}}$$

Bod

$$\frac{\begin{array}{c} \rho_\lambda \vdash C_\lambda[\lambda x.e] \Rightarrow^{uva}_{m} (C[([e_0]\, e_1)], \rho), \psi \\ C[(e_0\, e_1)] \sqsubset Ctx \qquad \rho \vdash C[(e_0\, e_1)] \Rightarrow^{uva}_{m} c, \psi' \end{array}}{Ctx :: \rho_\lambda \vdash C_\lambda[\lambda x.[e]] \Rightarrow^{uva}_{m} c, \psi \cup \psi' \cup \{x\}}$$

2 Fundamental Unchanged Variable Analysis Theorems

In this section, we define and prove lemmas and theorems regarding the soundness of unchanged variable analysis for both Demand Evaluation and Demand m-CFA. We first prove the soundness of Demand Evaluation with UVA. To prove the fundamental Demand Evaluation UVA theorem, we first define and prove lemmas for other relations: Bind, Find, and Trace. The following is a series of lemmas for the fundamental Demand Evaluation UVA theorem.

Lemma 1 (Fundamental DE UVA Bind). *If* $\rho,\ x \vdash C[e] \Uparrow^{uva}_{b/d} (C_x[\lambda x.[e_0]],\ \rho_0),\ X$ *then* $\forall y \in [free(e) \cap free(e_0)] \setminus X,\ \rho(y) = \rho_0(y)$

Proof. Proof by induction on the evaluation derivation. This is a trivial proof since the bind operator only moves up the AST. If the collected variable is not in X, then the binding of y will be equivalent since two environments share the same environment prefix.

Lemma 2 (Fundamental DE UVA Find). *If* $\rho,\ x \vdash e \Uparrow^{uva}_{f/d} (x,\ \rho_0),\ X$ *then* $\forall y \in [free(e) \cap free(x)] \setminus X,\ \rho(y) = \rho_0(y)$

Proof. Proof by induction on evaluation derivation. This is also a trivial proof, similar to the bind operator. The find operator only moves down the AST. The result's scope shares the same environment prefix as the initial scope.

Lemma 3 (Fundamental DE UVA Trace). *If* $\rho \vdash e \Rightarrow^{uva}_{d} (c,\ \rho_0),\ X$ *then* $\forall y \in [free(e) \cap free(c)] \setminus X,\ \rho(y) = \rho_0(y)$

Proof. Proof by induction on evaluation derivation.

Case Rat (Base case): The initial environment is returned as a result environment, and the changed variable set is empty.

Case Ran: Ran Case has three premises, and we can derive relations of the environment from each premise. From the first premise–the evaluation of the function–we derive $\forall y \in ([free(e_0) \cap free(\lambda x.e)] \setminus \psi), \Sigma :: \rho(y) = \rho_\lambda(y)$ From the second premise–the find of x in the body of the function-we derive $\forall y \in ([free(e) \cap free(x)] \setminus \psi'), (C[(e_0\ e_1)] :: \Sigma) :: \rho_\lambda(y) = \rho_x(y)$ From the third premise–the trace of the x-we derive $\forall y \in ([free(x) \cap free((e_2\ e_3)] \setminus \psi''), \Sigma :: \rho(y) = \rho_0(y)$. If we combine all of them, we get $\forall y \in [free(e_0) \cap free(\lambda x.e) \cap free((e_2\ e_3))] \setminus (\psi \cup \psi' \cup \psi'')$. Because $free(\lambda x.e)$ considers the extended environment from the result's starting environment and final environment, we can safely assume that the result changed variable set $(\psi \cup \psi' \cup \psi''$ already traversed the variables of $free(\lambda x.e)$. Thus, we can derive $\forall y \in [free(e_0) \cap free((e_2\ e_3))] \setminus (\psi \cup \psi' \cup \psi'')$. We demonstrate this assumption with an example. Consider a program $(\lambda f.\lambda x.((f\ \lambda h.(h\ 10))\ \lambda z.z)\ \lambda g.\lambda w.(g\ w))$ Say we run a UVA trace on $\lambda z.z$. For the first premise, we evaluate $(f\ \lambda h.(h\ 10))$, which results in $\lambda w.(g\ w), \{x, f, g\}$. For the second premise, we find w in $\lambda w.(g\ w)$, which reults with $w, \{w\}$. For the third premise, we trace w, which brings another inference for Ran: The first inner premise is to evaluate g, which results in $\lambda h.(h\ 10), \{w, g, f, x\}$. The second inner premise is to find h in $\lambda h.(h\ 10)$ which results to $h, \{h\}$. Third inner premise is trace h which results to $(h\ 10), \emptyset$. We can see that $free(\lambda w.(g\ w))$ is irrelevant because the result changed variable set removes the variable in scope for $\lambda w.(g\ w)$.

Case Bod: The Bod case has two premises, and we can likewise derive relations from them. From the first premise–the trace of the whole lambda expression–we derive $\forall y \in ([free(\lambda x.e) \cap free(e_0)] \setminus \psi), \Sigma :: \rho_\lambda(y) = \rho(y)$. From the second premise–the trace of the first result–we derive $\forall y \in ([free((e_0\ e_1)) \cap free(\lambda x'.e')] \setminus \psi'), \Sigma :: \rho(y) = \rho_0(y)$. From these derivation, we get $\forall y \in [free(\lambda x.e) \cap free(e_0) \cap free(\lambda x'.e')] \setminus \psi \cup \psi' \cup \{x\}$. We use the same argument from the Ran case that $free(e_0)$ becomes irrelevant because of the evaluation traversal. We get $\forall y \in [free(\lambda x.e) \cap free(\lambda x'.e')] \setminus \psi \cup \psi' \cup \{x\}$.

With the lemmas for bind, find, and trace, we now prove the soundness theorem for evaluation.

Theorem 2 (Fundamental DE UVA Theorem). *If* $\rho \vdash e \Downarrow_d^{uva} (\lambda x.e,\ \rho_0),\ X$ *then* $\forall y \in [free(e) \cap free(\lambda x.e)] \setminus X,\ \rho(y) = \rho_0(y)$

Proof. Proof by induction on the evaluation derivation.

Case Lam (Base Case): The initial environment is returned as a result environment, and the changed variable set is empty. Therefore, the environments are equivalent.

Case App: The evaluation of the application expression involves 3 premises: evaluation of the function, the argument, and the body of the function. From each of these premises, we derive relations of environments. We use these relations to build a bridge to prove the binding equivalence. From the first premise–the evaluation of the function–we derive $\forall y \in ([free(e_0) \cap free(\lambda x'.e')] \setminus \psi), \Sigma :: \rho(y) = \rho_\lambda(y)$ From the third premise–the evaluation of the body of the function-we derive $\forall y \in ([free(e') \cap free(\lambda x.e)] \setminus \psi'), \Sigma :: \rho(y) = \rho_0(y)$ We can now aggregate relation into $\forall y \in [free((e_0)) \cap free(\lambda x'.e') \cap free(\lambda x.e)] \setminus (\psi \cup \psi'), \Sigma :: \rho(y) = \rho_0(y)$tc We use the same argument from UVA Theorem's Ran case of Trace to show that $free(\lambda x'.e')$ is irrelevant by the changed variable set because evaluating the body expression collects all relevant variables within the scope.

Case Ref: The evaluation of the reference involves 3 premises: bind of x, trace of the lambda expression, and evaluation of the argument. From the first premise–the bind of the x–we derive $\forall y \in ([free(x) \cap free(e)] \setminus \psi), \rho_x(y) = (C[(e_0\ e_1)] :: \Sigma) :: \rho_\lambda(y)$ From the second premise–the trace of the lambda expression–we derive $\forall y \in ([free(\lambda x.e) \cap free((e_0\ e_1))] \setminus \psi'), \rho_\lambda(y) = \Sigma :: \rho_0(y)$ From the third premise–the evaluation of the argument–we derive $\forall y \in ([free(e_1) \cap free(\lambda x'.e')] \setminus \psi''), \Sigma :: \rho_0(y) = \rho_v(y)$ We aggregate them into $\forall y \in [free(\lambda x.e) \cap free((e_0\ e_1)) \cap free(\lambda x'.e')] \setminus (\psi \cup \psi' \cup \psi'' \cup \{x\}), \rho_x(y) = \rho_v(y)$ We use the same argument from the App case to show $free((e_0\ e_1))$ is removed by the changed variables in the scope.

With the Fundamental UVA Theorem for Demand Evaluation with UVA proven, proving the same lemmas and proofs for Demand ∞-CFA can be proven with the same method. Demand ∞-CFA has nearly identical rule structure with

Demand Evaluation's rule, except for omitting a few constraints. $infty$-CFA can be achieved by setting m to ∞. Following is the series of Lemmas for bind, find, and trace for Demand ∞-CFA.

Lemma 4 (Fundamental CFA UVA Bind). *If* $\rho,\ x\ \vdash\ C[e]\ \Uparrow^{uva}_{b/\infty}\ (C_x[\lambda x.[e_0]],\ \rho_0),\ X$ *then* $\forall y \in [free(e) \cap free(e_0)] \setminus X,\ \ \rho(y) = \rho_0(y)$

Lemma 5 (Fundamental CFA UVA Find). *If* $\rho,\ x\ \vdash\ e\ \Uparrow^{uva}_{f/\infty}\ (x,\ \rho_0),\ X$ *then* $\forall y \in [free(e) \cap free(x)] \setminus X,\ \ \rho(y) = \rho_0(y)$

Lemma 6 (Fundamental CFA UVA Trace). *If* $\rho\ \vdash\ e\ \Rightarrow^{uva}_{\infty}\ (c,\ \rho_0),\ X$ *then* $\forall y \in [free(e) \cap free(c)] \setminus X,\ \ \rho(y) = \rho_0(y)$

Theorem 3 (Fundamental UVA Theorem For Demand ∞-CFA). *If* $\rho\ \vdash$ $e\ \Downarrow^{uva}_{\infty},\ (\lambda x.e,\ \rho_0),\ X_\infty$
then $\forall y \in [free(e) \cap free(\lambda x.e)] \setminus X,\ \ \rho(y) = \rho_0(y)$

Proof. Follows from the soundness of Demand ∞-CFA with respect to Demand Evaluation–in particular that $X_d \subseteq X_\infty$.

Theorem 4. *For all* $n \in \mathbb{N}$*, if* $\rho_a \vdash e \Downarrow^{uva}_{n+1} v_a, X_a$ *and* $\rho_a \sqsubseteq \rho_b$*, then* $\rho_b \vdash e \Downarrow^{uva}_{n}$ v_b, X_b *where* $v_a \sqsubseteq v_b$ *and* $X_a \sqsubseteq X_b$

Proof. Proof by induction on the evaluation derivation. The only difference between $\Downarrow^{uva}_{n+1}$ and $\Downarrow^{uva}_{n}$ is the restriction of the environment stack. Thus, the lower the n is, the more it overapproximates the result with less precision, but it still subsumes the more precise result. Otherwise, all relations are consistent between two semantics, which makes the proof straightforward.

References

[1] Tim Whiting and Kimball Germane. "Context-Sensitive Demand-Driven Control-Flow Analysis". en. In: *Programming Languages and Systems.* Ed. by Viktor Vafeiadis. Cham: Springer Nature Switzerland, 2025, pp. 374–401. ISBN: 978-3-031-91121-7. DOI: 10.1007/978-3-031-91121-7_15.