Lenguajes y Compiladores

2017

Estructura de la materia a grandes rasgos:

Primera Parte: Lenguaje imperativo

Segunda Parte: Lenguaje aplicativo puro, y lenguaje aplicativo con referencias y asignación

Ejes de contenidos de la segunda parte

Semántica Denotacional del Cálculo Lambda

Requisito inicial para su semántica:

un conjunto C tal que C es isomorfo a $C \to C$

¿Paradoja? Sea f cualquier función, y definamos $p_f\, x = f(xx)$.

Entonces $p_f\, p_f$ es punto fijo de f.

Scott-Strachey(† 1975) Definen la semántica utilizando dominios:

$$D_\infty \;\simeq\; [\, D_\infty \to D_\infty]$$

Semántica Denotacional del Cálculo Lambda

Requisito inicial para su semántica:

un conjunto C tal que C es isomorfo a $C \to C$

¿Paradoja? Sea f cualquier función, y definamos $p_f\, x = f(xx)$.

Entonces $p_f\, p_f$ es punto fijo de f.

Scott-Strachey(† 1975) Definen la semántica utilizando dominios:

$$D_\infty \;\simeq\; [\, D_\infty \to D_\infty]$$

Semántica Denotacional del Cálculo Lambda

Requisito inicial para su semántica:

un conjunto C tal que C es isomorfo a $C \to C$

¿Paradoja? Sea f cualquier función, y definamos $p_f\, x = f(xx)$.

Entonces $p_f\, p_f$ es punto fijo de f.

Scott-Strachey(† 1975) Definen la semántica utilizando dominios:

$$D_\infty \quad \simeq \quad [\, D_\infty \to D_\infty]$$

Semántica Denotacional del Cálculo Lambda

Asumimos la existencia de un dominio D_∞, junto con un isomorfismo:

$$\phi \in D_\infty \to [D_\infty \to D_\infty]$$
$$\psi \in [D_\infty \to D_\infty] \to D_\infty$$

es decir

$$\phi \circ \psi = Id_{[D_\infty \to D_\infty]}$$

y

$$\psi \circ \phi = Id_{D_\infty}$$

Semántica Denotacional del Cálculo Lambda

Ambientes (Entornos):

$$\eta \in Env = \langle var \rangle \to D_\infty$$

Función semántica:

$$[\![\_]\!] \in \langle exp \rangle \to Env \to D_\infty$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \eta v$$

$$[\![e_0 e_1]\!]\eta = \phi([\![e_0]\!]\eta)\ [\![e_1]\!]\eta$$

$$[\![\lambda v.e]\!]\eta = \psi(\lambda d \in D_\infty.[\![e]\!][\eta|v:d])$$

Se puede probar que $[\![\Delta\,\Delta]\!]\eta = \bot$.

Semántica Denotacional del Cálculo Lambda

Ambientes (Entornos):

$$\eta \in Env = \langle var \rangle \to D_\infty$$

Función semántica:

$$[\![\_]\!] \in \ \langle exp \rangle \to Env \to D_\infty$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \eta v$$

$$[\![e_0 e_1]\!]\eta = \phi([\![e_0]\!]\eta)\ [\![e_1]\!]\eta$$

$$[\![\lambda v.e]\!]\eta = \psi(\lambda d \in D_\infty.[\![e]\!][\eta|v:d])$$

Se puede probar que $[\![\Delta\,\Delta]\!]\eta = \bot$.

Semántica Denotacional del Cálculo Lambda

Ambientes (Entornos):

$$\eta \in Env = \langle var \rangle \to D_\infty$$

Función semántica:

$$[\![\_]\!] \in \ \langle exp \rangle \to Env \to D_\infty$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \eta v$$

$$[\![e_0 e_1]\!]\eta = \phi([\![e_0]\!]\eta)\ [\![e_1]\!]\eta$$

$$[\![\lambda v.e]\!]\eta = \psi(\lambda d \in D_\infty.[\![e]\!][\eta|v:d])$$

Se puede probar que $[\![\Delta\,\Delta]\!]\eta = \bot$.

Semántica Denotacional del Cálculo Lambda

Ambientes (Entornos):

$$\eta \in Env = \langle var \rangle \to D_\infty$$

Función semántica:

$$[\![\_]\!] \in \ \langle exp \rangle \to Env \to D_\infty$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \eta v$$

$$[\![e_0 e_1]\!]\eta = \phi([\![e_0]\!]\eta)\ [\![e_1]\!]\eta$$

$$[\![\lambda v.e]\!]\eta = \psi(\lambda d \in D_\infty.[\![e]\!][\eta | v : d])$$

Se puede probar que $[\![\Delta\ \Delta]\!]\eta = \bot$.

Semántica Denotacional del Cálculo Lambda

Ambientes (Entornos):

$$\eta \in Env = \langle var \rangle \to D_\infty$$

Función semántica:

$$[\![\_]\!] \in \ \langle exp \rangle \to Env \to D_\infty$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \eta v$$

$$[\![e_0 e_1]\!]\eta = \phi([\![e_0]\!]\eta)\ [\![e_1]\!]\eta$$

$$[\![\lambda v.e]\!]\eta = \psi(\lambda d \in D_\infty.[\![e]\!][\eta|v:d])$$

Se puede probar que $[\![\Delta\ \Delta]\!]\eta = \bot$.

Semántica Denotacional del Cálculo Lambda

Ambientes (Entornos):

$$\eta \in Env = \langle var \rangle \to D_\infty$$

Función semántica:

$$[\![\_]\!] \in \ \langle exp \rangle \to Env \to D_\infty$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \eta v$$

$$[\![e_0 e_1]\!]\eta = \phi([\![e_0]\!]\eta)\ [\![e_1]\!]\eta$$

$$[\![\lambda v.e]\!]\eta = \psi(\lambda d \in D_\infty.[\![e]\!][\eta|v:d])$$

Se puede probar que $[\![\Delta\,\Delta]\!]\eta = \bot$.

Propiedades de la semántica del CL

Teorema de Coincidencia:
Si $\eta w = \eta' w$ para todo $w \in FV\ e$, entonces $[\![e]\!]\eta = [\![e]\!]\eta'$.

Teorema de Renombre: Si $v_{new} \notin FV\ e - \{v\}$,
entonces $[\![\lambda v_{new}.(e/v \mapsto v_{new})]\!] = [\![\lambda v.e]\!]$.

Sustituciones: $\Delta\ =\ \langle var \rangle \to \langle exp \rangle$

Teorema de Sustitución: Si $[\![\delta w]\!]\eta = \eta' w$ para todo $w \in FV\ e$, entonces $[\![e/\delta]\!]\eta = [\![e]\!]\eta'$.

Propiedades de la semántica del CL

Teorema de Coincidencia:
Si $\eta w = \eta' w$ para todo $w \in FV\ e$, entonces $[\![e]\!]\eta = [\![e]\!]\eta'$.

Teorema de Renombre: Si $v_{new} \notin FV\ e - \{v\}$,
entonces $[\![\lambda v_{new}.(e/v \mapsto v_{new})]\!] = [\![\lambda v.e]\!]$.

Sustituciones: $\Delta\ =\ \langle var \rangle \to \langle exp \rangle$

Teorema de Sustitución: Si $[\![\delta w]\!]\eta = \eta' w$ para todo $w \in FV\ e$, entonces $[\![e/\delta]\!]\eta = [\![e]\!]\eta'$.

Propiedades de la semántica del CL

Teorema de Coincidencia:
Si $\eta w = \eta' w$ para todo $w \in FV\ e$, entonces $[\![e]\!]\eta = [\![e]\!]\eta'$.

Teorema de Renombre: Si $v_{new} \notin FV\ e - \{v\}$,
entonces $[\![\lambda v_{new}.(e/v \mapsto v_{new})]\!] = [\![\lambda v.e]\!]$.

Sustituciones: $\Delta\ =\ \langle var \rangle \to \langle exp \rangle$

Teorema de Sustitución: Si $[\![\delta w]\!]\eta = \eta' w$ para todo $w \in FV\ e$, entonces $[\![e/\delta]\!]\eta = [\![e]\!]\eta'$.

Corrección de las reglas β y η

Son válidas en el modelo D_∞ las reglas del cálculo lambda:

$$[\![(\lambda v.e)e']\!]\eta = [\![e/v \mapsto e']\!]\eta$$

$$[\![\lambda v.e\, v]\!]\eta = [\![e]\!]\eta, \text{ si } v \notin FV\, e$$

Corolario: Si $e \to^* e'$, entonces $[\![e]\!] = [\![e']\!]$.

Corrección de las reglas β y η

Son válidas en el modelo D_∞ las reglas del cálculo lambda:

$$[\![(\lambda v.e)e']\!]\eta = [\![e/v \mapsto e']\!]\eta$$

$$[\![\lambda v.e\, v]\!]\eta = [\![e]\!]\eta, \text{ si } v \notin FV\, e$$

Corolario: Si $e \to^* e'$, entonces $[\![e]\!] = [\![e']\!]$.

Corrección de las reglas β y η

Son válidas en el modelo D_∞ las reglas del cálculo lambda:

$$[\![(\lambda v.e)e']\!]\eta = [\![e/v \mapsto e']\!]\eta$$

$$[\![\lambda v.e\, v]\!]\eta = [\![e]\!]\eta, \text{ si } v \notin FV\, e$$

Corolario: Si $e \to^* e'$, entonces $[\![e]\!] = [\![e']\!]$.

¿Representa D_∞ la evaluación?

La semántica denotacional dada **no representa adecuadamente** la evaluación de los lenguajes funcionales.

Puesto que no distingue los valores de términos que divergen.

Mientras que $\lambda x.\Delta\,\Delta$ es un valor, en D_∞ tenemos $[\![\lambda v.\Delta\Delta]\!] = \bot$.

Si queremos construir un modelo que represente adecuadamente la evaluación (cualquiera de ellas), qué vamos a romper necesariamente? Repuesta: la regla η.

¿Representa D_∞ la evaluación?

La semántica denotacional dada **no representa adecuadamente** la evaluación de los lenguajes funcionales.

Puesto que no distingue los valores de términos que divergen.

Mientras que $\lambda x.\Delta\,\Delta$ es un valor, en D_∞ tenemos $[\![\lambda v.\Delta\Delta]\!] = \bot$.

Si queremos construir un modelo que represente adecuadamente la evaluación (cualquiera de ellas), qué vamos a romper necesariamente? Repuesta: la regla η.

¿Representa D_∞ la evaluación?

La semántica denotacional dada **no representa adecuadamente** la evaluación de los lenguajes funcionales.

Puesto que no distingue los valores de términos que divergen.

Mientras que $\lambda x.\Delta\,\Delta$ es un valor, en D_∞ tenemos $[\![\lambda v.\Delta\Delta]\!] = \bot$.

Si queremos construir un modelo que represente adecuadamente la evaluación (cualquiera de ellas), qué vamos a romper necesariamente?
Repuesta: la regla η.

¿Representa D_∞ la evaluación?

La semántica denotacional dada **no representa adecuadamente** la evaluación de los lenguajes funcionales.

Puesto que no distingue los valores de términos que divergen.

Mientras que $\lambda x.\Delta\,\Delta$ es un valor, en D_∞ tenemos $[\![\lambda v.\Delta\Delta]\!] = \bot$.

Si queremos construir un modelo que represente adecuadamente la evaluación (cualquiera de ellas), qué vamos a romper necesariamente? Repuesta: la regla η.

Semántica Denotacional Normal

Hay que distinguir las semántica de $\lambda v.\Delta\Delta$ y $\Delta\Delta$

Los "valores" (conjunto V), representan a las formas canónicas:

$$V \approx [D \to D]$$

Definimos el conjunto de resultados posibles como :

$$D = V_\perp$$

Semántica Denotacional Normal

Hay que distinguir las semántica de $\lambda v.\Delta\Delta$ y $\Delta\Delta$

Los "valores" (conjunto V), representan a las formas canónicas:

$$V \quad \approx \quad [D \to D]$$

Definimos el conjunto de resultados posibles como :

$$D \;=\; V_{\perp}$$

Semántica Denotacional Normal

Hay que distinguir las semántica de $\lambda v.\Delta\Delta$ y $\Delta\Delta$

Los "valores" (conjunto V), representan a las formas canónicas:

$$V \quad \approx \quad [D \to D]$$

Definimos el conjunto de resultados posibles como :

$$D \; = \; V_{\perp}$$

Dominio para Semántica Normal

Asumimos la existencia de un dominio D, junto con un isomorfismo:

$$\phi \in V \to [D \to D]$$
$$\psi \in [D \to D] \to V$$

$$\phi \circ \psi = Id_{D \to D}$$
$$\psi \circ \phi = Id_V$$

Notación:

$$\iota_\perp \in V \to D$$

Semántica Denotacional del Cálculo Lambda Normal

Dominio Semántico: $D = V_\perp \qquad V \approx [D \to D]$

Ambientes: $Env = \langle var \rangle \to D$

Función semántica:

$$[\![\_]\!] \in \langle exp \rangle \to Env \to D$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \eta v$$
$$[\![e_0 e_1]\!]\eta = \phi_{\perp\!\perp}([\![e_0]\!]\eta)([\![e_1]\!]\eta)$$
$$[\![\lambda v.e]\!]\eta = \iota_\perp \circ \psi\ (\lambda d \in D.[\![e]\!][\eta | v : d])$$

Semántica Denotacional del Cálculo Lambda Normal

Dominio Semántico: $D = V_\perp \qquad V \approx [D \to D]$

Ambientes: $Env = \langle var \rangle \to D$

Función semántica:

$$[\![\_]\!] \in \langle exp \rangle \to Env \to D$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \eta v$$
$$[\![e_0 e_1]\!]\eta = \phi_{\perp\!\perp}([\![e_0]\!]\eta)([\![e_1]\!]\eta)$$
$$[\![\lambda v.e]\!]\eta = \iota_\perp \circ \psi\ (\lambda d \in D.[\![e]\!][\eta | v : d])$$

Semántica Denotacional del Cálculo Lambda Normal

Dominio Semántico: $D = V_\perp \qquad V \approx [D \to D]$

Ambientes: $Env = \langle var \rangle \to D$

Función semántica:

$$[\![\_]\!] \in \quad \langle exp \rangle \to Env \to D$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \eta v$$
$$[\![e_0 e_1]\!]\eta = \phi_{\perp\!\perp}([\![e_0]\!]\eta)([\![e_1]\!]\eta)$$
$$[\![\lambda v.e]\!]\eta = \iota_\perp \circ \psi\ (\lambda d \in D.[\![e]\!][\eta | v : d])$$

Semántica Denotacional del Cálculo Lambda Normal

Dominio Semántico: $D = V_\perp \qquad V \approx [D \to D]$

Ambientes: $Env = \langle var \rangle \to D$

Función semántica:

$$[\![\_]\!] \in \quad \langle exp \rangle \to Env \to D$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \eta v$$

$$[\![e_0 e_1]\!]\eta = \phi_{\perp\!\perp}([\![e_0]\!]\eta)([\![e_1]\!]\eta)$$

$$[\![\lambda v.e]\!]\eta = \iota_\perp \circ \psi\ (\lambda d \in D.[\![e]\!][\eta | v : d])$$

Semántica Denotacional del Cálculo Lambda Normal

Dominio Semántico: $D = V_\perp \qquad V \approx [D \to D]$

Ambientes: $Env = \langle var \rangle \to D$

Función semántica:

$$[\![\_]\!] \in \quad \langle exp \rangle \to Env \to D$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \eta v$$
$$[\![e_0 e_1]\!]\eta = \phi_{\perp\!\!\perp}([\![e_0]\!]\eta)([\![e_1]\!]\eta)$$
$$[\![\lambda v.e]\!]\eta = \iota_\perp \circ \psi\ (\lambda d \in D.[\![e]\!][\eta | v : d])$$

Semántica Denotacional del Cálculo Lambda Normal

Dominio Semántico: $D = V_\perp \qquad V \approx [D \to D]$

Ambientes: $Env = \langle var \rangle \to D$

Función semántica:

$$[\![\_]\!] \in \quad \langle exp \rangle \to Env \to D$$

Ecuaciones semánticas:

$$\begin{aligned}
[\![v]\!]\eta &= \eta v \\
[\![e_0 e_1]\!]\eta &= \phi_{\perp\!\!\perp}([\![e_0]\!]\eta)([\![e_1]\!]\eta) \\
[\![\lambda v.e]\!]\eta &= \ \iota_\perp \circ \psi\ (\lambda d \in D.[\![e]\!][\eta | v : d])
\end{aligned}$$

Una aplicación se cuelga si . . .

¿Cuáles son todas las posibilidades para que $e_0 e_1$ tenga semántica $\perp$?

Recordemos que $\phi_{\perp\!\!\!\perp} \perp \quad = \quad \perp_{D \to D}$.

Si bien la semántica denotacional no expresa un orden de evaluación ya que no es operacional, establece indirectamente el orden natural en que debe evaluarse. Por más que $[\![e_1]\!]\eta = \perp$, no necesariamente $[\![e_0 e_1]\!]\eta = \perp$.

Eso indica que e_1 no necesariamente debe evaluarse para evaluarse $e_0 e_1$.

Una aplicación se cuelga si . . .

¿Cuáles son todas las posibilidades para que $e_0 e_1$ tenga semántica $\perp$?
Recordemos que $\phi_{\perp\!\!\perp} \perp \;=\; \perp_{D \to D}$.

Si bien la semántica denotacional no expresa un orden de evaluación ya que no es operacional, establece indirectamente el orden natural en que debe evaluarse. Por más que $[\![e_1]\!]\eta = \perp$, no necesariamente $[\![e_0 e_1]\!]\eta = \perp$.

Eso indica que e_1 no necesariamente debe evaluarse para evaluarse $e_0 e_1$.

Una aplicación se cuelga si . . .

¿Cuáles son todas las posibilidades para que $e_0 e_1$ tenga semántica $\perp$? Recordemos que $\phi_{\perp\!\!\perp}\ \perp\ \ = \ \perp_{D\to D}$.

Si bien la semántica denotacional no expresa un orden de evaluación ya que no es operacional, establece indirectamente el orden natural en que debe evaluarse. Por más que $[\![e_1]\!]\eta = \perp$, no necesariamente $[\![e_0 e_1]\!]\eta = \perp$.

Eso indica que e_1 no necesariamente debe evaluarse para evaluarse $e_0 e_1$.

Una aplicación se cuelga si . . .

¿Cuáles son todas las posibilidades para que $e_0 e_1$ tenga semántica $\perp$? Recordemos que $\phi_{\perp\!\!\perp}\ \perp\ \ = \ \perp_{D\to D}$.

Si bien la semántica denotacional no expresa un orden de evaluación ya que no es operacional, establece indirectamente el orden natural en que debe evaluarse. Por más que $[\![e_1]\!]\eta = \perp$, no necesariamente $[\![e_0 e_1]\!]\eta = \perp$.

Eso indica que e_1 no necesariamente debe evaluarse para evaluarse $e_0 e_1$.

Una aplicación se cuelga si . . .

¿Cuáles son todas las posibilidades para que $e_0 e_1$ tenga semántica $\perp$? Recordemos que $\phi_{\perp\!\!\perp}\ \perp\ \ = \ \perp_{D\to D}$.

Si bien la semántica denotacional no expresa un orden de evaluación ya que no es operacional, establece indirectamente el orden natural en que debe evaluarse. Por más que $[\![e_1]\!]\eta = \perp$, no necesariamente $[\![e_0 e_1]\!]\eta = \perp$.

Eso indica que e_1 no necesariamente debe evaluarse para evaluarse $e_0 e_1$.

Propiedades de la Semántica Denotacional Normal

Los teoremas que vimos antes siguen valiendo.

También vale la regla β, que utiliza la igualdad:

$$\phi_{\perp\!\!\perp} \circ (\iota_{\perp} \circ \psi) = Id_{D \to D}$$

No vale la regla η. De hecho, empezamos queriendo distinguir $\lambda x.(\Delta\,\Delta)x$ de $\Delta\,\Delta$.

Propiedades de la Semántica Denotacional Normal

Los teoremas que vimos antes siguen valiendo.

También vale la regla β, que utiliza la igualdad:

$$\phi_{\perp\!\!\perp} \circ (\iota_{\perp} \circ \psi) = Id_{D \to D}$$

No vale la regla η. De hecho, empezamos queriendo distinguir $\lambda x.(\Delta\,\Delta)x$ de $\Delta\,\Delta$.

Propiedades de la Semántica Denotacional Normal

Los teoremas que vimos antes siguen valiendo.

También vale la regla β, que utiliza la igualdad:

$$\phi_{\perp\!\!\perp} \circ (\iota_{\perp} \circ \psi) = Id_{D \to D}$$

No vale la regla η. De hecho, empezamos queriendo distinguir $\lambda x.(\Delta\,\Delta)x$ de $\Delta\,\Delta$.

Modalidad EAGER

La semántica denotacional dada **no representa adecuadamente** la evaluación de los lenguajes funcionales.

Por ejemplo: $(\lambda x.\lambda y.y)(\Delta\Delta)$

La aplicación se efectúa cuando el operando se haya evaluado. Desde el punto de vista semántico, las funciones toman valores solamente.

Modalidad EAGER

La semántica denotacional dada **no representa adecuadamente** la evaluación de los lenguajes funcionales.

Por ejemplo: $(\lambda x.\lambda y.y)(\Delta\Delta)$

La aplicación se efectúa cuando el operando se haya evaluado. Desde el punto de vista semántico, las funciones toman valores solamente.

Semántica Denotacional Eager

Los "valores" (conjunto V), representan a las formas canónicas, pero ahora son funciones que sólo toman valores:

$$V \quad \approx \quad V \to D$$

Definimos el conjunto de resultados posibles como :

$$D \; = \; V_{\perp}$$

Dominios para Semántica Eager

Asumimos la existencia de un dominio D, junto con isomorfismos:

$$\phi \in V \to [V \to D]$$
$$\psi \in [V \to D] \to V$$

$$\phi \circ \psi = Id_{V \to D}$$
$$\psi \circ \phi = Id_V$$

Notación:

$$\iota_\perp \in V \to D$$

Semántica Denotacional del Cálculo Lambda Eager

Dominio Semántico: $D = V_\perp \qquad V \approx [V \to D]$

Ambientes: $Env = \langle var \rangle \to V$

Función semántica:

$$[\![\_]\!] \in \langle exp \rangle \to Env \to D$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \iota_\perp(\eta v)$$
$$[\![e_0 e_1]\!]\eta = (\phi_{\perp\!\!\perp}([\![e_0]\!]\eta))_{\perp\!\!\perp}([\![e_1]\!]\eta)$$
$$[\![\lambda v.e]\!]\eta = \iota_\perp \circ \psi\ (\lambda z \in V.[\![e]\!][\eta | v : z])$$

Semántica Denotacional del Cálculo Lambda Eager

Dominio Semántico: $D = V_\perp \qquad V \approx [V \to D]$

Ambientes: $Env = \langle var \rangle \to V$

Función semántica:

$$[\![\_]\!] \in \langle exp \rangle \to Env \to D$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \iota_\perp(\eta v)$$
$$[\![e_0 e_1]\!]\eta = (\phi_{\perp\!\!\perp}([\![e_0]\!]\eta))_{\perp\!\!\perp}([\![e_1]\!]\eta)$$
$$[\![\lambda v.e]\!]\eta = \iota_\perp \circ \psi\ (\lambda z \in V.[\![e]\!][\eta | v : z])$$

Semántica Denotacional del Cálculo Lambda Eager

Dominio Semántico: $D = V_\perp \qquad V \approx [V \to D]$

Ambientes: $Env = \langle var \rangle \to V$

Función semántica:

$$[\![\_]\!] \in \quad \langle exp \rangle \to Env \to D$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \iota_\perp(\eta v)$$
$$[\![e_0 e_1]\!]\eta = (\phi_{\perp\!\!\perp}([\![e_0]\!]\eta))_{\perp\!\!\perp}([\![e_1]\!]\eta)$$
$$[\![\lambda v.e]\!]\eta = \iota_\perp \circ \psi\ (\lambda z \in V.[\![e]\!][\eta|v : z])$$

Semántica Denotacional del Cálculo Lambda Eager

Dominio Semántico: $D = V_\perp \qquad V \approx [V \to D]$

Ambientes: $Env = \langle var \rangle \to V$

Función semántica:

$$[\![\_]\!] \in \quad \langle exp \rangle \to Env \to D$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \iota_\perp(\eta v)$$

$$[\![e_0 e_1]\!]\eta = (\phi_{\perp\!\!\perp}([\![e_0]\!]\eta))_{\perp\!\!\perp}([\![e_1]\!]\eta)$$

$$[\![\lambda v.e]\!]\eta = \iota_\perp \circ \psi\ (\lambda z \in V.[\![e]\!][\eta|v:z])$$

Semántica Denotacional del Cálculo Lambda Eager

Dominio Semántico: $D = V_\perp \qquad V \approx [V \to D]$

Ambientes: $Env = \langle var \rangle \to V$

Función semántica:

$$[\![\_]\!] \in \quad \langle exp \rangle \to Env \to D$$

Ecuaciones semánticas:

$$[\![v]\!]\eta = \iota_\perp(\eta v)$$
$$[\![e_0 e_1]\!]\eta = (\phi_{\perp\!\!\perp}([\![e_0]\!]\eta))_{\perp\!\!\perp}([\![e_1]\!]\eta)$$
$$[\![\lambda v.e]\!]\eta = \iota_\perp \circ \psi\ (\lambda z \in V.[\![e]\!][\eta | v : z])$$

Semántica Denotacional del Cálculo Lambda Eager

Dominio Semántico: $D = V_\perp \qquad V \approx [V \to D]$

Ambientes: $Env = \langle var \rangle \to V$

Función semántica:

$$[\![\_]\!] \in \quad \langle exp \rangle \to Env \to D$$

Ecuaciones semánticas:

$$\begin{aligned}
[\![v]\!]\eta &= \iota_\perp(\eta v) \\
[\![e_0 e_1]\!]\eta &= (\phi_{\perp\!\!\perp}([\![e_0]\!]\eta))_{\perp\!\!\perp}([\![e_1]\!]\eta) \\
[\![\lambda v.e]\!]\eta &= \ \iota_\perp \circ \psi\ (\lambda z \in V.[\![e]\!][\eta | v : z])
\end{aligned}$$

Propiedades de la Semántica Denotacional Eager

Los teoremas que vimos antes siguen valiendo.

Ya no vale la regla β, (para contra-ejemplo alcanza un $\perp$):

$$[\![(\lambda x\, y.y)\,(\Delta\,\Delta)]\!]\eta = (\lambda z \in V.[\![e]\!][\eta|v:z])_{\perp\!\!\!\perp}\,\perp = \perp$$

pero

$$[\![(\lambda y.y)/x \mapsto (\Delta\,\Delta)]\!]\eta = [\![\lambda y.y]\!]\eta \neq \perp$$

Puesto que queremos modelar la evaluación eager, deberíamos esperar que $e \Rightarrow_E z$ implique $[\![e]\!]\eta = [\![z]\!]\eta$.

Lo podemos probar por inducción en la derivación $e \Rightarrow_E z$.

Propiedades de la Semántica Denotacional Eager

Los teoremas que vimos antes siguen valiendo.

Ya no vale la regla β, (para contra-ejemplo alcanza un $\perp$):

$$[\![(\lambda x\, y.y)\,(\Delta\,\Delta)]\!]\eta = (\lambda z \in V.[\![e]\!][\eta|v : z])_{\perp\!\!\!\perp}\,\perp = \perp$$

pero

$$[\![(\lambda y.y)/x \mapsto (\Delta\,\Delta)]\!]\eta = [\![\lambda y.y]\!]\eta \neq \perp$$

Puesto que queremos modelar la evaluación eager, deberíamos esperar que $e \Rightarrow_E z$ implique $[\![e]\!]\eta = [\![z]\!]\eta$.

Lo podemos probar por inducción en la derivación $e \Rightarrow_E z$.

Propiedades de la Semántica Denotacional Eager

Los teoremas que vimos antes siguen valiendo.

Ya no vale la regla β, (para contra-ejemplo alcanza un $\perp$):

$$[\![(\lambda x\, y.y)\,(\Delta\,\Delta)]\!]\eta = (\lambda z \in V.[\![e]\!][\eta|v : z])_{\perp\!\!\!\perp}\,\perp = \perp$$

pero

$$[\![(\lambda y.y)/x \mapsto (\Delta\,\Delta)]\!]\eta = [\![\lambda y.y]\!]\eta \neq \perp$$

Puesto que queremos modelar la evaluación eager, deberíamos esperar que $e \Rightarrow_E z$ implique $[\![e]\!]\eta = [\![z]\!]\eta$.

Lo podemos probar por inducción en la derivación $e \Rightarrow_E z$.