Agregando polimorfismo a una lógica que identifica proposiciones isomorfas

Cristian F. Sottile

Instituto de Investigación en Ciencias de la Computación (ICC).
CONICET – Universidad de Buenos Aires.

Laboratorio de Investigación y Formación en Informática Avanzada.
Facultad de Informática, Universidad Nacional de La Plata.

Resumen Sistema I es un cálculo lambda simplemente tipado con pares, extendido con una teoría ecuacional obtenida a partir de considerar a los tipos isomorfos como iguales. En este trabajo presentamos una extensión de Sistema I a polimorfismo, agregando los isomorfismos correspondientes, y proveemos pruebas no estándar de las propiedades de preservación de tipos y normalización fuerte.

Keywords: Cálculo lambda, Polimorfismo, Tipos isomorfos

1. Introducción

Dos tipos A y B son considerados isomorfos ($\equiv$) si existen dos funciones f de tipo $A \Rightarrow B$ y g de tipo $B \Rightarrow A$ tales que la composición $g \circ f : A \Rightarrow A$ es la identidad en A y la composición $f \circ g : B \Rightarrow B$ es la identidad en B. Di Cosmo et al. [9] caracterizaron los tipos isomorfos en diferentes sistemas, entre ellos, tipos simples, tipos simples con pares y polimorfismo. Utilizando esta caracterización, Díaz-Caro y Dowek [12] definieron Sistema I, un cálculo lambda simplemente tipado con pares donde los tipos isomorfos son considerados iguales. De esta manera, si A y B son isomorfos, cada término de tipo A puede usarse como un término de tipo B. Por ejemplo, el isomorfismo $(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C$, en el que está basada la currificación, permite pasar argumentos de a uno a funciones que esperan pares. Normalmente, esto requeriría que una función $f : (A \wedge B) \Rightarrow C$ sea transformada explícitamente mediante otra función h en la función hf de tipo $A \Rightarrow B \Rightarrow C$. Sistema I considera que f tiene ambos tipos ($(A \wedge B) \Rightarrow C$ y $A \Rightarrow B \Rightarrow C$), y la transformación ocurre implícitamente sin necesidad de utilizar h. Para que esto funcione, Sistema I incluye una relación de equivalencia entre términos ($\rightleftarrows$); por ejemplo, $r\langle s, t\rangle$ es equivalente a rst, estando sustentada esta equivalencia por el isomorfismo de la currificación y también por la intuición: si r espera como argumento un par de valores, entonces puede recibir esos valores de forma separada. Además, modifica la semántica operacional restringiéndola con respecto al tipo del argumento: si s tiene tipo A, entonces $(\lambda x^A.r)s$ β-convierte a $[x := s]r$. Esto impide conversiones incorrectas como

$(\lambda x^{A\wedge B}.r)s^A t^B \rightarrow_\beta ([x^{A\wedge B} := s^A]r)t^B$ (notar que la expresión original es tipable debido al isomorfismo de la currificación), forzando el uso de la equivalencia $(\lambda x^{A\wedge B}.r)s^A t^B \rightleftarrows (\lambda x^{A\wedge B}.r)\langle s^A, t^B\rangle$ para luego proceder con la β-conversión correcta, obteniendo $[x^{A\wedge B} := \langle s^A, t^B\rangle]r$.

La identificación de proposiciones ha sido previamente investigada, por ejemplo, en la teoría de tipos de Martin-Löf [21], en el Cálculo de Construcciones [6], y en *Deduction modulo theory* [16,17], donde diferentes proposiciones como $A \subseteq B$, $A \in \mathcal{P}(B)$ y $\forall x\ (x \in A \Rightarrow x \in B)$, pueden ser identificadas. Sin embargo, la igualdad definicional no trata isomorfismos. Por ejemplo, $A \wedge B$ y $B \wedge A$ no se identifican en estas lógicas. Además de la igualdad definicional, identificar tipos isomorfos en teoría de tipos es también un objetivo del axioma de univalencia [29]. Desde el punto de vista de la programación, los isomorfismos capturan la correspondencia de utilidad computacional entre tipos, es decir, dados dos tipos isomorfos, podremos obtener los mismos resultados a partir de ellos. Tomando como ejemplo nuevamente a la currificación, una función f de tipo $(A \wedge B) \Rightarrow C$ puede transformarse en una f' de tipo $A \Rightarrow B \Rightarrow C$, y a partir de ambas podremos obtener el mismo C, ya sea mediante la aplicación de f con un par de valores $A \wedge B$, o de la aplicación de f' con esos mismos valores A y B por separado. Estas dos funciones f y f' difieren entonces en la forma en que pueden ser combinadas con otros programas, pero comparten la misma utilidad final: computan un C a partir de un A y un B. En este sentido, la propuesta de Sistema I es permitir que los programadores se centren en la utilidad de los programas, admitiendo combinar cualquier programa con aquellos combinables con sus correspondientes isomorfos (e.g., $fr^A s^B$ y $f'\langle r^A, s^B\rangle$), relajando la rigidez de los sistemas de tipos clásicos dentro del contexto seguro provisto por la teoría de tipos isomorfos. Desde la perspectiva de la lógica, los isomorfismos hacen que las pruebas sean más naturales. Por ejemplo, probar $(A \wedge (A \Rightarrow B)) \Rightarrow B$ en deducción natural requiere introducir la hipótesis conjuntiva $A \wedge (A \Rightarrow B)$, que debe descomponerse luego en las proposiciones A y $A \Rightarrow B$, mientras que la currificación permite transformar la proposición a probar en $A \Rightarrow (A \Rightarrow B) \Rightarrow B$, e introducir directamente las hipótesis A y $A \Rightarrow B$, eliminando la necesidad de introducir la hipótesis conjuntiva.

Uno de los pioneros en el uso de isomorfismos en lenguajes de programación fue Rittri [24], que utilizó tipos igualados por isomorfismos para buscar programas en bibliotecas.

Un intérprete de una versión preliminar de Sistema I extendido con números y recursión fue implementado en Haskell [15]. Este lenguaje tiene algunas características particulares; por ejemplo, mediante el uso del isomorfismo entre $A \Rightarrow (B \wedge C)$ y $(A \Rightarrow B) \wedge (A \Rightarrow C)$, es posible proyectar una función que computa un par de elementos (sin haber aplicado la función), y obtener mediante evaluación una función más sencilla que computa solo el elemento del par que interesa, descartando el código que computa el elemento que no interesa. En el paper se incluyen ejemplos no triviales.

En este artículo presentamos una extensión de Sistema I a polimorfismo, considerando los isomorfismos correspondientes. Este trabajo es un resumen de

$$A \wedge B \equiv B \wedge A \tag{1}$$
$$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C \tag{2}$$
$$A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C) \tag{3}$$
$$(A \wedge B) \Rightarrow C \equiv A \Rightarrow B \Rightarrow C \tag{4}$$
$$\text{Si } X \notin FTV(A),\ \forall X.(A \Rightarrow B) \equiv A \Rightarrow \forall X.B \tag{5}$$
$$\forall X.(A \wedge B) \equiv \forall X.A \wedge \forall X.B \tag{6}$$

Cuadro 1. Isomorfismos considerados en SIP

la tesina de grado con mismo título [28], con el agregado de una prueba de normalización fuerte, un resultado posterior derivado de la misma. En octubre de 2019 fueron publicados [25] en el XXV Congreso Argentino de Ciencias de la Computación (CACIC) los resultados preliminares de la tesina; en marzo de 2020 fue defendida la tesina en la Universidad Nacional de La Plata; y en septiembre de 2020 fueron publicados [26] en el 32nd Symposium on Implementation and Application of Functional Languages (IFL) los resultados finales con la prueba de normalización fuerte.

El trabajo se organiza como sigue: la Sección 2 introduce el sistema propuesto, y la Sección 3 da ejemplos para clarificar las construcciones. Las Secciones 4 y 5 prueban, respectivamente, las propiedades de preservación de tipos y normalización fuerte, que son los dos teoremas principales. Finalmente, la Sección 6 discute algunas decisiones de diseño y posibles direcciones para trabajo futuro.

2. Sistema I Polimórfico

Definimos Sistema I Polimórfico (SIP) como una extensión de Sistema I [12] a tipos polimórficos. La sintaxis de los tipos coincide con la de Sistema F [20, Chapter 11] con pares:

$$A \quad := \quad X \mid A \Rightarrow A \mid A \wedge A \mid \forall X.A$$

donde $X \in \mathcal{TV}ar$, un conjunto de variables de tipo.

La extensión con respecto a Sistema F con pares consiste en agregar una regla de tipado tal que si t tiene tipo A y $A \equiv B$, entonces t también tiene tipo B, lo que vale para cualquier par de tipos isomorfos A y B. Este agregado no trivial induce una modificación de la semántica operacional del cálculo.

Hay ocho isomorfismos que caracterizan a todos los isomorfismos válidos en Sistema F con pares (cf. [9, Table 1.4]). De esos ocho, consideramos los seis dados como una congruencia en el Cuadro 1, donde $FTV(A)$ es el conjunto de variables de tipo libres, definido como es usual.

Los dos isomorfismos no listados son los siguientes:

$$\forall X.A \equiv \forall Y.[X := Y]A \tag{7}$$

$$\forall X.\forall Y.A \equiv \forall Y.\forall X.A \tag{8}$$

El isomorfismo (7) es de hecho una α-equivalencia, y SIP ya considera a los términos y tipos módulo α-equivalencia. No hacemos explícito el isomorfismo para evitar confusión. El isomorfismo (8), por otro lado, no es incorporado en este trabajo porque SIP es presentado *à la Church* (al igual que Sistema I), por lo que intercambiar los parámetros en una abstracción de tipos implicaría intercambiar los argumentos de tipo con una notación incómoda y poca ganancia. Discutimos más al respecto en la Sección 6.1.

La nueva regla de tipado induce ciertas equivalencias entre términos. Por ejemplo, el isomorfismo (1) implica que los pares $\langle r, s\rangle$ y $\langle s, r\rangle$ son indistinguibles, dado que ambos tienen tipo $A \wedge B$ y $B \wedge A$, por lo que los consideramos equivalentes. De igual modo, como consecuencia del isomorfismo (2), $\langle r, \langle s, t\rangle\rangle$ es equivalente a $\langle\langle r, s\rangle, t\rangle$.

Esta equivalencia entre términos provoca que la proyección con respecto a la posición (i.e. $\pi_i(\langle r_1, r_2\rangle) \hookrightarrow r_i$), como es definida usualmente, no sea adecuada para este sistema: $\pi_1(\langle r, s\rangle)$ reduciría a r, y a su vez dado que $\langle r, s\rangle$ es equivalente a $\langle s, r\rangle$, también reduciría a s. Es por esto que SIP (así como Sistema I), define la proyección con respecto al tipo: si $\Gamma \vdash r : A$, entonces $\pi_A(\langle r, s\rangle) \hookrightarrow r$.

Esta regla torna SIP en un sistema no determinista (y por lo tanto no confluente), ya que si r y s tienen tipo A, $\pi_A\langle r, s\rangle$ podrá reducir a r o a s. De todas formas, este no determinismo no constituye un problema. Si se piensa a SIP como un sistema de pruebas, y en presencia de la propiedad de preservación de tipos, el no determinismo implica que el sistema identifica las pruebas diferentes de proposiciones isomorfas (como una forma de *proof-irrelevance*). Por otra parte, si se considera a SIP como un lenguaje de programación, el determinismo se puede recuperar mediante la siguiente codificación: si r y s tienen el mismo tipo, la proyección determinista de $\langle r, s\rangle$ a r será $\pi_{B\Rightarrow A}(\langle \lambda x^B.r, \lambda x^C.s\rangle)t$, con t de tipo B y $B \not\equiv C$. Por lo tanto, el no determinismo de SIP, heredado de Sistema I, se considera una característica deseable y no un inconveniente (cf. [12] para una discusión más profunda).

SIP (así como Sistema I), es uno más de los cálculos no deterministas que se pueden encontrar en la literatura, e.g. [4,5,7,8,22], y nuestro operador de construcción de pares puede considerarse como el operador de composición paralela de un cálculo no determinista.

En los cálculos no deterministas, la elección no determinista $r \oplus s$, con r y s abstracciones, representa a la computación que ejecuta r o s de manera no determinista, es decir tal que $(r \oplus s)t$ reduce a rt o a st. Por otro lado, con el operador de composición paralela $\|$, el término $(r \parallel s)t$ reduce a $rt \parallel st$ y ejecuta ambos rt y st en paralelo. En nuestro caso, dados r y s de tipo $A \Rightarrow B$ y t de tipo A, el término $\pi_B(\langle r, s\rangle t)$ es equivalente a $\pi_B(\langle rt, st\rangle)$, que reduce a rt o a st, mientras que el término $\langle rt, st\rangle$ ejecutaría ambas computaciones en paralelo. Nuestro constructor de pares es entonces equivalente a la composición paralela, mientras que la elección no determinista se descompone en la construcción de pares seguida de su destrucción.

$$\frac{}{\Gamma, x : A \vdash x : A}(\text{ax}) \quad \frac{\Gamma \vdash r : A \qquad A \equiv B}{\Gamma \vdash r : B}(\equiv) \quad \frac{\Gamma, x : A \vdash r : B}{\Gamma \vdash \lambda x^A.r : A \Rightarrow B}(\Rightarrow_i)$$

$$\frac{\Gamma \vdash r : A \qquad \Gamma \vdash s : B}{\Gamma \vdash \langle r, s\rangle : A \wedge B}(\wedge_i) \quad \frac{\Gamma \vdash r : A \wedge B}{\Gamma \vdash \pi_A(r) : A}(\wedge_e) \quad \frac{\Gamma \vdash r : A \Rightarrow B \qquad \Gamma \vdash s : A}{\Gamma \vdash rs : B}(\Rightarrow_e)$$

$$\frac{\Gamma \vdash r : A \qquad X \notin FTV(\Gamma)}{\Gamma \vdash \Lambda X.r : \forall X.A}(\forall_i) \quad \frac{\Gamma \vdash r : \forall X.A}{\Gamma \vdash r[B] : [X := B]A}(\forall_e)$$

Cuadro 2. Reglas de tipado

En SIP y en Sistema I, el no determinismo se produce por la interacción de dos operadores: $\langle , \rangle$ y π. Esto también tiene relación con los cálculos algebraicos [1,2,3,30,11,14], algunos de los cuales fueron diseñados para expresar algoritmos cuánticos. Hay un claro vínculo entre nuestros operadores de construcción y proyección de pares, y los operadores de superposición $(+)$ y medición π de dichos cálculos. Allí el par $r+s$ no se interpreta como la elección no determinista, sino como la superposición de dos procesos que ejecutan r y s, y el operador π constituye la proyección relacionada a la medición, que es el único operador no determinista. En estos cálculos la regla de distributividad $(r + s)t \rightleftarrows rt + st$ se entiende como la definición punto a punto de la suma de dos funciones.

La sintaxis de los términos es similar a la de Sistema F con pares, con la diferencia de que la proyección se efectúa con respecto al tipo en vez a la posición.

$$r \quad := \quad x^A \mid \lambda x^A.r \mid rr \mid \langle r, r\rangle \mid \pi_A(r) \mid \Lambda X.r \mid r[A]$$

donde $x^A \in \mathcal{V}ar$, un conjunto de variables tipadas. Omitimos el tipo de las variables cuando es evidente por contexto. Por ejemplo, escribimos $\lambda x^A.x$ en vez de $\lambda x^A.x^A$.

El sistema de tipos de SIP es estándar, con solo dos cambios con respecto al de Sistema F con pares: la eliminación de la proyección $(\wedge_e)$y la regla añadida para los isomorfismos $(\equiv)$. El sistema completo se muestra en el Cuadro 2. Escribimos $\Gamma \vdash r : A$ para expresar que r tiene tipo A en el contexto Γ. Notar que, dado que el sistema se define *à la Church*, el contexto es redundante [19,23], por lo que podremos escribir "r tiene tipo A" sin ambugüedad. Excepto cuando se indique de otra manera, usaremos las últimas letras mayúsculas del alfabeto latino (W, X, Y, Z) para variables de tipo, las primeras letras mayúsculas del alfabeto latino $(A, B, C, \dots)$ para tipos cualesquiera, las últimas letras minúsculas del alfabeto latino (x, y, z) para variables de término, otras letras minúsculas del alfabeto latino $(r, s, t, \dots)$ para términos cualesquiera, y letras mayúsculas del alfabeto griego $(\Gamma, \Delta, \dots)$ para contextos.

Así como los isomorfismos (1) y (2) inducen conmutatividad y asociatividad en pares y una consecuente modificación en la proyección, el isomorfismo (3) provoca que una abstracción de tipo $A \Rightarrow (B \wedge C)$ pueda considerarse como un par de abstracciones de tipo $(A \Rightarrow B) \wedge (A \Rightarrow C)$, y como tal pueda ser proyectada. Luego una abstracción que computa un par se identifica con

un par de abstracciones, y un par aplicado distribuye su argumento. Es decir, $\lambda x^A.\langle r, s\rangle \rightleftarrows \langle \lambda x^A.r, \lambda x^A.s\rangle$, y $\langle r, s\rangle t \rightleftarrows \langle rt, st\rangle$, donde $\rightleftarrows$ es un símbolo simétrico (y $\rightleftarrows^*$ su clausura transitiva).

De igual modo, el isomorfismo (4) induce una equivalencia entre $r\langle s, t\rangle$ y rst. Esta equivalencia produce una ambigüedad con la β-reducción. Por ejemplo, si s tiene tipo A y t tiene tipo B, el término $(\lambda x^{A\wedge B}.r)\langle s, t\rangle$ puede β-reducir a $[x := \langle s, t\rangle]r$, pero también a $([x := s]r)t$, mediante su equivalencia con $(\lambda x^{A\wedge B}.r)st$. Así, la β-reducción no sería estable en la equivalencia. Para asegurar la estabilidad en la equivalencia, la β-reducción debe efectuarse solo cuando el tipo del argumento coincida con el de la variable ligada: si $\Gamma \vdash s : A$, entonces $(\lambda x^A.r)s \hookrightarrow [x := s]r$.

Finalmente, los isomorfismos de polimorfismo inducen seis equivalencias, dos relacionadas con la conmutación entre el cuantificador universal y la flecha (derivadas de (5)) y cuatro relacionadas con la distributividad entre el cuantificador universal y el producto (derivadas de (6)). La existencia de diferentes equivalencias inducidas por el mismo isomorfismo se debe a las distintas maneras de construir términos, dadas por los constructores de tipos involucrados y sus respectivas introducciones y eliminaciones.

La semántica operacional de SIP se define mediante la relación $\hookrightarrow$ módulo la relación simétrica $\rightleftarrows$:

$$\rightarrow \quad := \quad \rightleftarrows^* \circ \hookrightarrow \circ \rightleftarrows^*$$

Como es usual, escribimos $\rightarrow^*$ para la clausura reflexiva y transitiva de $\rightarrow$. También podremos escribir $\hookrightarrow^n$ para exprear n pasos en la relación $\hookrightarrow$, y $\hookrightarrow_R$ para especificar que la regla usada es R. Puede verse la definición completa de estas relaciones en el Cuadro 3.

3. Ejemplos

En esta sección presentamos algunos ejemplos para discutir uso y necesidad de las reglas presentadas.

Ejemplo 1. Mostraremos el uso de equivalencias entre términos para construir aplicaciones no válidas en Sistema F con pares. Por ejemplo, la función *apply* $:= \lambda f^{A\Rightarrow B}.\lambda x^A.fx$ puede aplicarse con un par, e.g. $\langle g, r\rangle$, con $\vdash g : A \Rightarrow B$ y $\vdash r : A$, porque, debido al isomorfismo (4), la derivación en el Cuadro 4 es correcta. Tenemos entonces $(\lambda f^{A\Rightarrow B}.\lambda x^A.fx)\langle g, r\rangle \rightleftarrows (\lambda f^{A\Rightarrow B}.\lambda x^A.fx)gr \hookrightarrow^2_{\beta_\lambda} gr$.

Ejemplo 2. Continuando con el ejemplo anterior, se pueden construir aplicaciones equivalentes de otras maneras. Por ejemplo, el término $(\lambda f^{A\Rightarrow B}.\lambda x^A.fx)rg$ tiene tipo debido a los isomorfismos (1) y (4), y reduce a gr:

$$(\lambda f^{A\Rightarrow B}.\lambda x^A.fx)rg \rightleftarrows (\lambda f^{A\Rightarrow B}.\lambda x^A.fx)\langle r, g\rangle \rightleftarrows (\lambda f^{A\Rightarrow B}.\lambda x^A.fx)\langle g, r\rangle \rightarrow^* gr$$

Ejemplo 3. Concluyendo con el ejemplo anterior, la versión sin currificar de *apply*, $\lambda z^{(A\Rightarrow B)\wedge A}.\pi_{A\Rightarrow B}(z)\pi_A(z)$, puede aplicarse a $\vdash g : A \Rightarrow B$ y $\vdash r : A$ como si estuviera currificada:

$$(\lambda z^{(A\Rightarrow B)\wedge A}.\pi_{A\Rightarrow B}(z)\pi_A(z))gr \rightleftarrows (\lambda z^{(A\Rightarrow B)\wedge A}.\pi_{A\Rightarrow B}(z)\pi_A(z))\langle g, r\rangle$$

$$\langle r, s\rangle \rightleftarrows \langle s, r\rangle \qquad (\text{COMM})$$
$$\langle r, \langle s, t\rangle\rangle \rightleftarrows \langle \langle r, s\rangle, t\rangle \qquad (\text{ASSO})$$
$$\lambda x^A.\langle r, s\rangle \rightleftarrows \langle \lambda x^A.r, \lambda x^A.s\rangle \qquad (\text{DIST}_\lambda)$$
$$\langle r, s\rangle t \rightleftarrows \langle rt, st\rangle \qquad (\text{DIST}_{\text{app}})$$
$$r\langle s, t\rangle \rightleftarrows rst \qquad (\text{CURRY})$$
$$\text{Si } X \notin FTV(A),\ \ \Lambda X.\lambda x^A.r \rightleftarrows \lambda x^A.\Lambda X.r \qquad (\text{P-COMM}_{\forall_i \Rightarrow_i})$$
$$\text{Si } X \notin FTV(A),\ \ (\lambda x^A.r)[B] \rightleftarrows \lambda x^A.r[B] \qquad (\text{P-COMM}_{\forall_e \Rightarrow_i})$$
$$\Lambda X.\langle r, s\rangle \rightleftarrows \langle \Lambda X.r, \Lambda X.s\rangle \qquad (\text{P-DIST}_{\forall_i \wedge_i})$$
$$\langle r, s\rangle[A] \rightleftarrows \langle r[A], s[A]\rangle \qquad (\text{P-DIST}_{\forall_e \wedge_i})$$
$$\pi_{\forall X.A}(\Lambda X.r) \rightleftarrows \Lambda X.\pi_A(r) \qquad (\text{P-DIST}_{\forall_i \wedge_e})$$
$$\text{Si } r : \forall X.(B \wedge C),\ \ (\pi_{\forall X.B}(r))[A] \rightleftarrows \pi_{[X:=A]B}(r[A]) \qquad (\text{P-DIST}_{\forall_e \wedge_e})$$

$$\text{Si } \Gamma \vdash s : A,\ \ (\lambda x^A.r)s \hookrightarrow [x := s]r \qquad (\beta_\lambda)$$
$$(\Lambda X.r)[A] \hookrightarrow [X := A]r \qquad (\beta_\Lambda)$$
$$\text{Si } \Gamma \vdash r : A,\ \ \pi_A(\langle r, s\rangle) \hookrightarrow r \qquad (\pi)$$

$$\frac{r \rightleftarrows s}{\lambda x^A.r \rightleftarrows \lambda x^A.s} \quad \frac{r \rightleftarrows s}{rt \rightleftarrows st} \quad \frac{r \rightleftarrows s}{tr \rightleftarrows ts} \quad \frac{r \rightleftarrows s}{\langle t, r\rangle \rightleftarrows \langle t, s\rangle} \quad \frac{r \rightleftarrows s}{\pi_A(r) \rightleftarrows \pi_A(s)} \quad \frac{r \rightleftarrows s}{\langle r, t\rangle \rightleftarrows \langle s, t\rangle}$$
$$\frac{r \rightleftarrows s}{\Lambda X.r \rightleftarrows \Lambda X.s} \quad \frac{r \rightleftarrows s}{r[A] \rightleftarrows s[A]} \quad \frac{r \hookrightarrow s}{\lambda x^A.r \hookrightarrow \lambda x^A.s} \quad \frac{r \hookrightarrow s}{rt \hookrightarrow st} \quad \frac{r \hookrightarrow s}{tr \hookrightarrow ts}$$
$$\frac{r \hookrightarrow s}{\langle t, r\rangle \hookrightarrow \langle t, s\rangle} \quad \frac{r \hookrightarrow s}{\pi_A(r) \hookrightarrow \pi_A(s)} \quad \frac{r \hookrightarrow s}{\langle r, t\rangle \hookrightarrow \langle s, t\rangle} \quad \frac{r \hookrightarrow s}{\Lambda X.r \hookrightarrow \Lambda X.s} \quad \frac{r \hookrightarrow s}{r[A] \hookrightarrow s[A]}$$

Cuadro 3. Semántica operacional de SIP

$$\dfrac{\dfrac{\vdash \lambda f^{A\Rightarrow B}.\lambda x^A.fx : (A \Rightarrow B) \Rightarrow A \Rightarrow B}{\vdash \lambda f^{A\Rightarrow B}.\lambda x^A.fx : ((A \Rightarrow B) \wedge A) \Rightarrow B}(\equiv) \qquad \dfrac{\vdash g : A \Rightarrow B \qquad \vdash r : A}{\vdash \langle g, r\rangle : (A \Rightarrow B) \wedge A}(\wedge_i)}{\vdash (\lambda f^{A\Rightarrow B}.\lambda x^A.fx)\langle g, r\rangle : B}(\Rightarrow_e)$$

Cuadro 4. Derivación de tipos del ejemplo 1.

$$\dfrac{\dfrac{\vdash \Lambda X.\lambda x^A.\lambda f^{A\Rightarrow X}.fx : \forall X.(A \Rightarrow (A \Rightarrow X) \Rightarrow X)}{\vdash \Lambda X.\lambda x^A.\lambda f^{A\Rightarrow X}.fx : A \Rightarrow \forall X.((A \Rightarrow X) \Rightarrow X)}(\equiv) \qquad \vdash r : A}{\vdash (\Lambda X.\lambda x^A.\lambda f^{A\Rightarrow X}.fx)r : \forall X.((A \Rightarrow X) \Rightarrow X)}(\Rightarrow_e)$$

Cuadro 5. Derivación del tipo del ejemplo 5.

$$\hookrightarrow_{\beta_\lambda} \pi_{A\Rightarrow B}(\langle g, r\rangle)\pi_A(\langle g, r\rangle) \hookrightarrow^2_\pi gr$$

En los tres ejemplos previos la β-reducción no puede ocurrir antes que las equivalencias por la condición de tipado de la regla (β_λ).

Ejemplo 4. Otro ejemplo interesante es el mencionado en la Sección 1: una función que computa un par puede proyectarse sin haber sido aplicada, dando como

resultado una función que computa solo uno de los elementos del par. Consideramos el término $\pi_{A\Rightarrow B}(\lambda x^A.\langle r, s\rangle)$, con $x : A \vdash r : B$ y $x : A \vdash s : C$. Tiene tipo debido al isomorfismo (3), dado que $A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C)$, y reduce $\pi_{A\Rightarrow B}(\lambda x^A.\langle r, s\rangle) \rightleftarrows \pi_{A\Rightarrow B}(\langle \lambda x^A.r, \lambda x^A.s\rangle) \hookrightarrow_\pi \lambda x^A.r$.

Ejemplo 5. La regla ($\text{P-COMM}_{\forall_i \Rightarrow_i}$) es consecuencia del isomorfismo (5). Como ejemplo tenemos el término $(\Lambda X.\lambda x^A.\lambda f^{A\Rightarrow X}.fx)r$, bien tipado asumiendo $\vdash r : A$ y $X \notin FTV(A)$, como se muestra en el Cuadro 5, y cuya reducción es la siguiente:

$$(\Lambda X.\lambda x^A.\lambda f^{A\Rightarrow X}.fx)r \rightleftarrows (\lambda x^A.(\Lambda X.\lambda f^{A\Rightarrow X}.fx))r \hookrightarrow_{\beta_\lambda} (\Lambda X.\lambda f^{A\Rightarrow X}.fr)$$

Ejemplo 6. La regla ($\text{P-COMM}_{\forall_e \Rightarrow_i}$) es también consecuencia del isomorfismo (5). Consideramos el término $(\lambda x^{\forall X.(X\Rightarrow X)}.x)[A](\Lambda X.\lambda x^X.x)$. Sea $B = \forall X.(X \Rightarrow X)$, y dado que $B \Rightarrow B \equiv \forall Y.(B \Rightarrow (Y \Rightarrow Y))$ (renombrando la variable por legibilidad), tenemos $\vdash (\lambda x^B.x)[A](\Lambda X.\lambda x^X.x) : A \Rightarrow A$ La reducción es como sigue:

$$\begin{aligned}(\lambda x^{\forall X.(X\Rightarrow X)}.x)[A](\Lambda X.\lambda x^X.x) &\rightleftarrows (\lambda x^{\forall X.(X\Rightarrow X)}.x[A])(\Lambda X.\lambda x^X.x)\\ &\hookrightarrow_{\beta_\lambda} (\Lambda X.\lambda x^X.x)[A] \hookrightarrow_{\beta_\Lambda} \lambda x^A.x\end{aligned}$$

Ejemplo 7. Las reglas ($\text{P-DIST}_{\forall_i \wedge_i}$) y ($\text{P-DIST}_{\forall_i \wedge_e}$) son consecuencia del isomorfismo (6). Consideramos el término $\pi_{\forall X.(X\Rightarrow X)}(\Lambda X.\langle \lambda x^X.x, r\rangle)$, con $\vdash r : A$. Dado que $\forall X.((X \Rightarrow X) \wedge A) \equiv (\forall X.(X \Rightarrow X)) \wedge \forall X.A$, podemos derivar $\vdash \pi_{\forall X.(X\Rightarrow X)}(\Lambda X.\langle \lambda x^X.x, r\rangle) : \forall X.(X \Rightarrow X)$. Una posible reducción es:

$$\pi_{\forall X.(X\Rightarrow X)}(\Lambda X.\langle \lambda x^X.x, r\rangle) \rightleftarrows \pi_{\forall X.(X\Rightarrow X)}(\langle \Lambda X.\lambda x^X.x, \Lambda X.r\rangle) \hookrightarrow_\pi \Lambda X.\lambda x^X.x$$

Ejemplo 8. La regla ($\text{P-DIST}_{\forall_e \wedge_i}$) es también consecuencia del isomorfismo (6). Consideramos el término $\langle \Lambda X.\lambda x^X.\lambda y^A.r,\ \Lambda X.\lambda x^X.\lambda z^B.s\rangle[C]$, con $\vdash r : D$ y $\vdash s : E$. Tiene tipo $(C \Rightarrow A \Rightarrow D) \wedge (C \Rightarrow B \Rightarrow E)$, y reduce como sigue:

$$\begin{aligned}\langle \Lambda X.\lambda x^X.\lambda y^A.r,\ \Lambda X.\lambda x^X.\lambda z^B.s\rangle[C] &\rightleftarrows \langle (\lambda x^X.\lambda y^A.r)[C], (\lambda x^X.\lambda z^B.s)[C]\rangle\\ &\hookrightarrow_{\beta_\Lambda} \langle \lambda x^C.\lambda y^A.r,\ \lambda x^C.\lambda z^B.s\rangle\end{aligned}$$

Ejemplo 9. La regla ($\text{P-DIST}_{\forall_e \wedge_e}$) también es consecuencia del isomorfismo (6). Consideramos el término $(\pi_{\forall X.(X\Rightarrow X)}(\Lambda X.\langle \lambda x^X.x, r\rangle))[A]$ cuyo tipo es $A \Rightarrow A$, y reduce así:

$$\begin{aligned}(\pi_{\forall X.(X\Rightarrow X)}(\Lambda X.\langle \lambda x^X.x, r\rangle))[A] &\rightleftarrows \pi_{A\Rightarrow A}((\Lambda X.\langle \lambda x^X.x, r\rangle)[A])\\ &\hookrightarrow_{\beta_\Lambda} \pi_{A\Rightarrow A}(\langle \lambda x^A.x, [X := A]r\rangle) \hookrightarrow_\pi \lambda x^A.x\end{aligned}$$

4. Preservación de tipos

En esta sección probamos la preservación de tipos en la reducción. Primero necesitamos caracterizar las equivalencias entre tipos, por ejemplo "si $\forall X.A \equiv$

$B \wedge C$, entonces $B \equiv \forall X.B'$ y $C \equiv \forall X.C'$, con $A \equiv B' \wedge C'$" (Lema 6). Debido a la cantidad de isomorfismos, este tipo de lemas no es trivial. Para probar estas relaciones definimos los factores primos de un tipo (Definición 1): el multiconjunto de tipos que no son equivalentes a una conjunción, tal que la conjunción de todos sus elementos es equivalente al tipo original. Esta técnica fue previamente utilizada en Sistema I [12] con tipos simples con un solo tipo básico τ. En SIP, en cambio, tenemos infinitas variables como tipos básicos, por lo que la prueba es más compleja.

Escribimos $\vec{X}$ para $X_1, \ldots, X_n$ y $\forall \vec{X}.A$ para $\forall X_1 \ldots \forall X_n.A$, para algún n (en este último caso, si $n = 0$, entonces $\forall \vec{X}.A = A$). Además, escribimos $[A_1, \ldots, A_n]$ o $[A_i]_{i=1}^n$ para el multiconjunto con los elementos A_1 a A_n.

Definición 1 (Factores primos).

$$PF(X) = [X] \qquad PF(A \wedge B) = PF(A) \uplus PF(B)$$
$$PF(A \Rightarrow B) = [\forall \vec{X}_i.((A \wedge B_i) \Rightarrow Y_i)]_{i=1}^n \quad \textit{con } PF(B) = [\forall \vec{X}_i.(B_i \Rightarrow Y_i)]_{i=1}^n$$
$$PF(\forall X.A) = [\forall X.\forall \vec{Y}_i.(A_i \Rightarrow Z_i)]_{i=1}^n \quad \textit{con } PF(A) = [\forall \vec{Y}_i.(A_i \Rightarrow Z_i)]_{i=1}^n$$

El Lema 1 y el Corolario 1 establecen la correctitud de la Definición 1. Escribimos $\bigwedge([A_i]_{i=1}^n)$ para $\bigwedge_{i=1}^n A_i$.

Lema 1. *Para todo A, existen $\vec{X}, n, B_1, \ldots, B_n, Y_1, \ldots, Y_n$ tal que $PF(A) = [\forall \vec{X}_i.(B_i \Rightarrow Y_i)]_{i=1}^n$.*

Prueba. Por inducción directa en la estructura de A. □

Corolario 1. *Para todo A, $A \equiv \bigwedge(PF(A))$.*

Prueba. Por inducción en la estructura de A. □

El Lema 2 establece la estabilidad de los factores primos en la equivalencia y el Lema 3 establece una suerte de resultado recíproco.

Definición 2. *$[A_1, \ldots, A_n] \sim [B_1, \ldots, B_m]$ si $n = m$ y $A_i \equiv B_{p(i)}$, para $i = 1, \ldots, n$ y p permutación de $\{1, \ldots .n\}$.*

Lema 2. *Para todo A, B tal que $A \equiv B$, vale $PF(A) \sim PF(B)$.*

Prueba. Primero chequeamos $PF(A \wedge B) \sim PF(B \wedge A)$ y similar para los otros cinco isomorfismos. Luego probamos por inducción estructural que si A y B son equivalentes en un paso, entonces $PF(\mathrm{A}) \sim PF(\mathrm{B})$. Concluimos con una inducción en la longitud de la derivación de la equivalencia $A \equiv B$. □

Lema 3. *Para todo R, S multiconjuntos tal que $R \sim S$, vale $\bigwedge(R) \equiv \bigwedge(S)$.* □

Lema 4. *Para todo $\vec{X}, \vec{Z}, A, B, Y, W$ tal que $\forall \vec{X}.(A \Rightarrow Y) \equiv \forall \vec{Z}.(B \Rightarrow W)$, vale $\vec{X} = \vec{Z}$, $A \equiv B$, y $Y = W$.*

Prueba. Por inspección de los isomorfismos. □

Lema 5. *Para todo A, B, C_1, C_2 tal que $A \Rightarrow B \equiv C_1 \wedge C_2$, existen B_1, B_2 tal que $C_1 \equiv A \Rightarrow B_1$, $C_2 \equiv A \Rightarrow B_2$ y $B \equiv B_1 \wedge B_2$.* □

La prueba del Lema 5 puede encontrarse en el Apéndice A. Las pruebas de los Lemas 6 y 7 son similares.

Lema 6. *Para todo X, A, B, C tales que $\forall X.A \equiv B \wedge C$, existen B', C' tales que $B \equiv \forall X.B'$, $C \equiv \forall X.C'$ y $A \equiv B' \wedge C'$.* □

Lema 7. *Para todo X, A, B, C tales que $\forall X.A \equiv B \Rightarrow C$, existe C' tal que $C \equiv \forall X.C'$ y $A \equiv B \Rightarrow C'$.* □

Dado que el cálculo se presenta *à la Church*, SIP es dirigido por sintaxis, con excepción de la regla ($\equiv$), por lo que el lema de generación (Lema 9) es directo, y tenemos el siguiente lema de unicidad:

Lema 8 (Unicidad módulo). *Para todo Γ, r, A, B tales que $\Gamma \vdash r : A$ y $\Gamma \vdash r : B$, vale $A \equiv B$.* □

Prueba. Si la última regla de la derivación $\Gamma \vdash r : A$ es ($\equiv$), existe una derivación más corta $\Gamma \vdash r : C$ con $C \equiv A$, y, por hipótesis inductiva, $C \equiv B$, por lo tanto $A \equiv B$. Si la última regla de la derivación $\Gamma \vdash r : B$ es ($\equiv$) procedemos de igual manera. Los demás casos son dirigidos por sintaxis. □

Lema 9 (Generación). *Para todo Γ, x, r, s, X, A, B:*

1. *Si $\Gamma \vdash x : A$ y $\Gamma \vdash x : B$, entonces $A \equiv B$.*
2. *Si $\Gamma \vdash \lambda x^A.r : B$, entonces existe C tal que $\Gamma, x : A \vdash r : C$ y $B \equiv A \Rightarrow C$.*
3. *Si $\Gamma \vdash rs : A$, entonces existe C tal que $\Gamma \vdash r : C \Rightarrow A$ y $\Gamma \vdash s : C$.*
4. *Si $\Gamma \vdash \langle r, s\rangle : A$, entonces existen C, D tales que $A \equiv C \wedge D$, $\Gamma \vdash r : C$ y $\Gamma \vdash s : D$.*
5. *Si $\Gamma \vdash \pi_A(r) : B$, entonces $A \equiv B$ y existe C tal que $\Gamma \vdash r : B \wedge C$.*
6. *Si $\Gamma \vdash \Lambda X.r : A$, entonces existe C tal que $A \equiv \forall X.C$, $\Gamma \vdash r : C$ y $X \notin FTV(\Gamma)$.*
7. *Si $\Gamma \vdash r[A] : B$, entonces existe C tal que $[X := A]C \equiv B$ y $\Gamma \vdash r : \forall X.C$.* □

Lema 10 (Substitución).

1. *Para todo Γ, x, r, s, A, B tales que $\Gamma, x : B \vdash r : A$ y $\Gamma \vdash s : B$, tenemos $\Gamma \vdash [x := s]r : A$.*
2. *Para todo Γ, r, X, A, B tales que $\Gamma \vdash r : A$, tenemos $[X := B]\Gamma \vdash [X := B]r : [X := B]A$.*

Prueba. Por inducción en r. Prueba detallada en Apéndice A. □

Teorema 1 (Preservación de tipos). *Para todo Γ, r, s, A tales que $\Gamma \vdash r : A$ y $r \hookrightarrow s$ o $r \rightleftarrows s$, tenemos $\Gamma \vdash s : A$.*

Prueba. Por inducción en la relación $\rightarrow$. A partir de la información provista por el Lema 9 para cada término de un lado de la relación, construimos una derivación que da el mismo tipo al del otro lado de la relación en los casos $\rightleftarrows$, y aplicamos el Lema 10 para dar el tipo original al reducto en los casos $\hookrightarrow$. La prueba detallada puede encontrarse en la tesina [28]. □

5. Normalización fuerte

En esta sección probamos normalización fuerte para la relación $\to$, es decir, toda secuencia de reducción que parte desde un término tipado termina. Escribimos SN para el conjunto de términos tipados fuertemente normalizantes con respecto a la reducción $\to$, y $|t|$ para el tamaño de la reducción más larga que parte desde t.

Extendemos a polimorfismo la prueba de Sistema I [12]. Para probar que todo término tipado está en SN, asociamos a cada tipo A un conjunto $[\![A]\!]$ de términos fuertemente normalizantes, como es usual. Un término $\vdash r : A$ se dice "reducible" si $r \in [\![A]\!]$. Luego probamos un teorema de adecuación estableciendo que todo término bien tipado es reducible.

El conjunto $[\![A_1 \Rightarrow A_2 \Rightarrow \cdots \Rightarrow A_n \Rightarrow X]\!]$ se puede definir como el conjunto de los términos r tales que para todo $s \in [\![A_1]\!]$, $rs \in [\![A_2 \Rightarrow \cdots \Rightarrow A_n \Rightarrow X]\!]$ o, lo que es equivalente, como el conjunto de términos r tales que para todo $s_i \in [\![A_i]\!]$, $rs_1 \dots s_n \in [\![X]\!] = \mathsf{SN}$. Para probar que un término de la forma $\lambda x^A.t$ es reducible, es necesario usar la propiedad CR3 [20] en el primer caso, y la propiedad de que un término está en SN si todos sus reductos en un paso están en SN en el segundo caso. Como en SIP una introducción puede ser equivalente a una eliminación, e.g. $\langle rt, st\rangle \rightleftarrows \langle r, s\rangle t$, no podemos utilizar la noción de términos neutrales que es necesaria para tener una propiedad equivalente a CR3. Por este motivo usamos la segunda definición para $[\![\cdot]\!]$, y, dado que la reducción depende del tipado, el conjunto $[\![A]\!]$ se define como un conjunto de términos tipados.

5.1. Reducibilidad

Definición 3 (Contexto de eliminación). *Consideramos una extensión del lenguaje donde introducimos un símbolo extra $(\!|)^A$, que llamaremos agujero de tipo A. Definimos el conjunto de contextos de eliminación con un agujero $(\!|)^A$ como el menor conjunto tal que:*

- *$(\!|)^A$ es un contexto de eliminación de tipo A,*
- *si $K_A^{B\Rightarrow C}$ es un contexto de eliminación de tipo $B \Rightarrow C$ con un agujero de tipo A y $r \in \mathsf{SN}$ es un término de tipo B, entonces $K_A^{B\Rightarrow C} r$ es un contexto de eliminación de tipo C con un agujero de tipo A,*
- *si $K_A^{B\wedge C}$ es un contexto de eliminación de tipo $B \wedge C$ con un agujero de tipo A, entonces $\pi_B(K_A^{B\wedge C})$ es un contexto de eliminación de tipo B con un agujero de tipo A,*
- *y si $K_A^{\forall X.B}$ es un contexto de eliminación de tipo $\forall X.B$ con un agujero de tipo A, entonces $K_A^{\forall X.B}[C]$ es un contexto de eliminación de tipo $[X := C]B$ con un agujero de tipo A.*

Escribimos $K_A^B(\!|r|\!)$ para $[(\!|)^A := r](K_A^B)$, donde $(\!|)^A$ es el agujero de K_A^B. En particular, r puede ser un contexto de eliminación.

Ejemplo 10. Sean

$$K_X^X = (\!|)^X,$$

$$
\begin{aligned}
K'^{X}_{X\Rightarrow(X\wedge X)} &= K^{X}_{X}(\!|\pi_X((\!|)^{X\Rightarrow(X\wedge X)}x)|\!),\\
K''^{X}_{\forall X.X\Rightarrow(X\wedge X)} &= K'^{X}_{X\Rightarrow(X\wedge X)}(\!|(\!|)^{\forall X.X\Rightarrow(X\wedge X)}[X]|\!)\\
&= K^{X}_{X}(\!|\pi_X((\!|)^{\forall X.X\Rightarrow(X\wedge X)}[X]x)|\!).
\end{aligned}
$$

Tenemos $K''^{X}_{\forall X.X\Rightarrow(X\wedge X)}(\!|\Lambda X.\lambda y^X.\langle y,y\rangle|\!) = \pi_X((\Lambda X.\lambda y^X.\langle y,y\rangle)[X]x)$.

Definición 4 (Términos internos de un contexto de eliminación). *Sea K^A_B un contexto de eliminación. El multiconjunto de términos internos de K^A_B se define como*

$$
\begin{aligned}
\mathcal{T}((\!|)^A) &= \emptyset & \mathcal{T}(K^{B\Rightarrow C}_A r) &= \mathcal{T}(K^{B\Rightarrow C}_A) \uplus \{r\}\\
\mathcal{T}(\pi_B(K^{B\wedge C}_A)) &= \mathcal{T}(K^{B\wedge C}_A) & \mathcal{T}(K^{\forall X.B}_A[C]) &= \mathcal{T}(K^{\forall X.B}_A)
\end{aligned}
$$

Escribimos $|K^B_A|$ para $\sum_{i=1}^n |r_i|$ donde $[r_1,\ldots,r_n] = \mathcal{T}(K^B_A)$.

Ejemplo 11. $\mathcal{T}((\!|)^A rs) = [r,s]$ y $\mathcal{T}((\!|)^A\langle r,s\rangle) = [\langle r,s\rangle]$. Notar que $K^B_A(\!|t|\!) \rightleftarrows^* K'^B_A(\!|t|\!)$ no implica $\mathcal{T}(K^B_A) \sim \mathcal{T}(K'^B_A)$.

Definición 5 (Reducibilidad). *El conjunto de términos reducibles de tipo A (notación $[\![A]\!]$) se define como el conjunto de términos r de tipo A tal que para cualquier contexto de eliminación K^X_A cuyos términos internos están en SN, vale $K^X_A(\!|r|\!) \in \mathsf{SN}$.*

El siguiente lema es una consecuencia trivial de la definición de reducibilidad.

Lema 11. *Para todo A, B tales que $A \equiv B$, vale $[\![A]\!] = [\![B]\!]$.* □

Lema 12. *Para todo A, $[\![A]\!] \subseteq \mathsf{SN}$.*

Prueba. Para todo A, existe un contexto de eliminación K^X_A, dado que las variables están en SN y pueden tener cualquier tipo. Luego, sea $r \in [\![A]\!]$, vale $K^X_A(\!|r|\!) \in \mathsf{SN}$, y por lo tanto $r \in \mathsf{SN}$. □

5.2. Adecuación

Probamos el teorema de adecuación (Teorema 2) mostrando que todo término tipado es reducible, y el teorema de normalización fuerte (Teorema 3) como consecuencia del mismo.

Definición 6 (Adecuación de la substitución). *Una substitución θ es adecuada para un contexto Γ (notación $\theta \vDash \Gamma$) si para todo $x : A \in \Gamma$, vale $\theta(x) \in [\![A]\!]$.*

Teorema 2 (Adecuación). *Para todo Γ, r, A, y substitución θ tales que $\Gamma \vdash r : A$ y $\theta \vDash \Gamma$, vale $\theta r \in [\![A]\!]$.* □

Prueba. Por inducción en r. Prueba detallada en Apéndice B. □

Teorema 3 (Normalización fuerte). *Para todo Γ, r, A tales que $\Gamma \vdash r : A$, vale $r \in \mathsf{SN}$.*

Prueba. Por el Lema 12, la substitución identidad es adecuada. Entonces, por el Teorema 2 y el Lema 12, $r \in [\![A]\!] \subseteq \mathsf{SN}$. □

6. Conclusión, discusión y trabajo futuro

Sistema I es un cálculo lambda simplemente tipado con pares, extendido con una teoría ecuacional obtenida a partir de considerar a los tipos isomorfos como iguales. En este sentido, el sistema permite a los programadores enfocarse en la utilidad de los programas, ignorando la sintaxis rígida de los términos dentro del contexto seguro provisto por la teoría de tipos isomorfos. En este trabajo extendimos Sistema I con polimorfismo, incluyendo los isomorfismos correspondientes, enriqueciendo el lenguaje con una característica comúnmente esperada.

Desde el punto de vista de la lógica, Sistema I es un sistema de pruebas para la lógica proposicional en el que las proposiciones isomorfas tienen la misma prueba, y SIP extiende Sistema I con el cuantificador universal.

Los teoremas principales en este trabajo prueban la preservación de tipos (Teorema 1) y la normalización fuerte (Teorema 3). La prueba del segundo es una adaptación no trivial de la prueba de Girard [20] para Sistema F.

6.1. Swap

Como mencionamos en la Sección 2, hay dos isomorfismos de los caracterizados por Di Cosmo [9] para Sistema F con pares que no fueron considerados, al menos de manera explícita: (7) y (8). Sin embargo, el isomorfismo (7) es la α-equivalencia, incorporada de manera implícita y por lo tanto considerada. El isomorfismo que no fue considerado es (8), el cual permite intercambiar los cuantificadores universales en las abstracciones de tipo: $\forall X.\forall Y.A \equiv \forall Y.\forall X.A$. Este isomorfismo es el análogo a $A \Rightarrow B \Rightarrow C \equiv B \Rightarrow A \Rightarrow C$ en primer orden, consecuencia de los isomorfismos (4) y (1). En primer orden ese isomorfismo induce la equivalencia $(\lambda x^A.\lambda y^B.r)st \rightleftarrows (\lambda x^A.\lambda y^B.r)\langle s,t\rangle \rightleftarrows (\lambda x^A.\lambda y^B.r)\langle t,s\rangle \rightleftarrows (\lambda x^A.\lambda y^B.r)ts$. Un enfoque alternativo podría haber sido introducir una equivalencia entre $\lambda x^A.\lambda y^B.r$ y $\lambda y^B.\lambda x^A.r$. En cualquier caso, para mantener la preservación de tipos, la reducción β_λ debe verificar que los tipos del argumento y de la variable de la abstracción sean el mismo. Esta solución no es sencillamente extensible a la reducción β_Λ, ya que, en el nivel de los términos, los tipos de las variables se usan como etiqueta para determinar qué argumento le corresponde a qué variable (dejando la posibilidad de no determinismo si hay "etiquetas" duplicadas), pero en el nivel de los tipos no tenemos una forma natural de etiquetado.

Otra posible solución, en el mismo sentido, es la implementada por el cálculo lambda selectivo [18], donde únicamente la flecha fue considerada, dejando fuera al producto, por lo que el isomorfismo que se incorpora es $A \Rightarrow B \Rightarrow C \equiv B \Rightarrow A \Rightarrow C$. Allí la solución es incluir etiquetado externo (no basado en tipos) para determinar qué argumento corresponde a qué variable. Podríamos haber agregado un etiquetado para las aplicaciones de tipos $t[A_X]$ junto con la regla $r[A_X][B_Y] \rightleftarrows r[B_Y][A_X]$, modificado la reducción β_Λ a $(\Lambda X.r)[A_X] \hookrightarrow [X := A]r$. A pesar de que esta solución parece funcionar, no encontramos que contribuya al lenguaje en ningún aspecto, mientras que sí lo hace menos legible. Es por esto que decidimos excluir el isomorfismo (8) de SIP.

6.2. Trabajo futuro

η-expansión Una extensión de una versión previa de Sistema I [10] fue implementada [15] en Haskell. En ella fueron agregadas algunas reglas ad-hoc para poder probar la propiedad de progreso (i.e., todas las formas normales de términos cerrados son introducciones). Por ejemplo, una de ellas es "si s tiene tipo B, entonces $(\lambda x^A.\lambda y^B.r)s \hookrightarrow \lambda x^A.((\lambda y^B.r)s)$". Esta regla, junto con otras introducidas en esta implementación, es un caso particular de una más general regla de η-expansión. Efectivamente, con la regla $r \hookrightarrow \lambda x^A.rx$ podemos derivar $(\lambda x^A.\lambda y^B.r)s \hookrightarrow \lambda z^A.(\lambda x^A.\lambda y^B.r)sz \rightleftarrows^* \lambda z^A.(\lambda x^A.\lambda y^B.r)zs \hookrightarrow \lambda z^A.((\lambda y^B.[x := z]r)s)$. En [13] se probó que todas las reglas ad-hoc en [10] pueden eliminarse agregando reglas de extensionalidad.

Adicionalmente, la prueba de consistencia de SIP como lenguaje de *proof-terms* para la lógica de segundo orden fue dejada de lado adrede en este trabajo. Como se muestra en [12], la misma hubiera requerido restringir las variables a "tipos primos", es decir, tipos no conjuntivos. Cuando el lenguaje se extiende con reglas η, esta restricción deja de ser necesaria [13], por lo que dejamos la prueba de consistencia para una versión futura de SIP con reglas η.

Implementación La implementación mencionada de una versión preliminar de Sistema I incluye un operador de punto fijo y números, mostrando algunos ejemplos de programación interesantes. Planeamos extenderla con polimorfismo, siguiendo el diseño de SIP.

Referencias

1. Arrighi, P., Díaz-Caro, A.: A System F accounting for scalars. LMCS 8(1:11), 1–32 (2012)
2. Arrighi, P., Díaz-Caro, A., Valiron, B.: The vectorial lambda-calculus. Inf. and Comp. 254(1), 105–139 (2017)
3. Arrighi, P., Dowek, G.: Lineal: A linear-algebraic lambda-calculus. LMCS 13(1:8), 1–33 (2017)
4. Boudol, G.: Lambda-calculi for (strict) parallel functions. Inf. and Comp. 108(1), 51–127 (1994)
5. Bucciarelli, A., Ehrhard, T., Manzonetto, G.: A relational semantics for parallelism and non-determinism in a functional setting. APAL 163(7), 918–934 (2012)
6. Coquand, T., Huet, G.: The calculus of constructions. Inf. and Comp. 76(2–3), 95–120 (1988)
7. de'Liguoro, U., Piperno, A.: Non deterministic extensions of untyped λ-calculus. Inf. and Comp. 122(2), 149–177 (1995)
8. Dezani-Ciancaglini, M., de'Liguoro, U., Piperno, A.: A filter model for concurrent λ-calculus. SIAM JComp. 27(5), 1376–1419 (1998)
9. Di Cosmo, R.: Isomorphisms of types: from λ-calculus to information retrieval and language design. Progress in Theoretical Computer Science, Birkhauser, Switzerland (1995)
10. Díaz-Caro, A., Dowek, G.: Non determinism through type isomorphism. EPTCS (LSFA'12) 113, 137–144 (2013)

11. Díaz-Caro, A., Dowek, G.: Typing quantum superpositions and measurement. LNCS (TPNC'17) 10687, 281–293 (2017)
12. Díaz-Caro, A., Dowek, G.: Proof normalisation in a logic identifying isomorphic propositions. LIPIcs (FSCD'19) 131, 14:1–14:23 (2019)
13. Díaz-Caro, A., Dowek, G.: Extensional proofs in a propositional logic modulo isomorphisms. `arXiv:2002.03762` (2020)
14. Díaz-Caro, A., Guillermo, M., Miquel, A., Valiron, B.: Realizability in the unitary sphere. In: Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2019). pp. 1–13. IEEE, Vancouver, BC, Canada (2019)
15. Díaz-Caro, A., Martínez López, P.E.: Isomorphisms considered as equalities: Projecting functions and enhancing partial application through an implementation of λ^+. ACM IFL 2015(9), 1–11 (2015)
16. Dowek, G., Hardin, T., Kirchner, C.: Theorem proving modulo. JAR 31(1), 33–72 (2003)
17. Dowek, G., Werner, B.: Proof normalization modulo. JSL 68(4), 1289–1316 (2003)
18. Garrigue, J., Aït-Kaci, H.: The typed polymorphic label-selective λ-calculus. In: Proceedings of the 21st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. p. 35–47. POPL '94, Association for Computing Machinery, New York, NY, USA (1994)
19. Geuvers, H., Krebbers, R., McKinna, J., Wiedijk, F.: Pure type systems without explicit contexts. In: Crary, K., Miculan, M. (eds.) Proceedings of LFMTP 2010. EPTCS, vol. 34, pp. 53–67 (2010)
20. Girard, J.Y., Taylor, P., Lafont, Y.: Proofs and types. Cambridge U.P., UK (1989)
21. Martin-Löf, P.: Intuitionistic type theory. Bibliopolis, Napoli, Italy (1984)
22. Pagani, M., Ronchi Della Rocca, S.: Linearity, non-determinism and solvability. Fund. Inf. 103(1–4), 173–202 (2010)
23. Park, J., Seo, J., Park, S., Lee, G.: Mechanizing metatheory without typing contexts. Journal of Automated Reasoning 52(2), 215–239 (2014)
24. Rittri, M.: Retrieving library identifiers via equational matching of types. In: Proceedings of CADE 1990. LNCS, vol. 449, pp. 603–617 (1990)
25. Sottile, C., Díaz Caro, A., Martínez López, P.E.: Hacia un Sistema I Polimórfico. In: XXV Congreso Argentino de Ciencias de la Computación (CACIC) (2019)
26. Sottile, C.F., Díaz-Caro, A., Martínez López, P.E.: Polymorphic System I. In: IFL 2020: Proceedings of the 32nd Symposium on Implementation and Application of Functional Languages. p. 127–137. IFL 2020, Association for Computing Machinery, New York, NY, USA (2020), https://doi.org/10.1145/3462172.3462198
27. Sottile, C.F., Díaz-Caro, A., Martínez López, P.E.: Polymorphic system i. arXiv:2101.03215 (2021)
28. Sottile, C.F.: Agregando polimorfismo a una lógica que identifica proposiciones isomorfas. Tesina de Grado. Universidad Nacional de La Plata (2020)
29. The Univalent Foundations Program: HoTT: Univalent Foundations of Mathematics. Institute for Advanced Study, Princeton, NJ, USA (2013)
30. Vaux, L.: The algebraic lambda calculus. MSCS 19(5), 1029–1059 (2009)

A. Pruebas y lemas auxiliares de la Sección 4

Lema 5. *Para todo A, B, C_1, C_2 tal que $A \Rightarrow B \equiv C_1 \wedge C_2$, existen B_1, B_2 tal que $C_1 \equiv A \Rightarrow B_1$, $C_2 \equiv A \Rightarrow B_2$ y $B \equiv B_1 \wedge B_2$.*

Prueba. Por el Lema 2, $PF(A \Rightarrow B) \sim PF(C_1 \wedge C_2) = PF(C_1) \uplus PF(C_2)$.

Por el Lema 1, sean $PF(B) = [\forall \vec{X}_i.(D_i \Rightarrow Z_i)]_{i=1}^{n}$, $PF(C_1) = [\forall \vec{Y}_j.(E_j \Rightarrow Z'_j)]_{j=1}^{k}$, y $PF(C_2) = [\forall \vec{Y}_j.(E_j \Rightarrow Z'_j)]_{j=k+1}^{m}$. Luego, $[\forall \vec{X}_i.((A \wedge D_i) \Rightarrow Z_i)]_{i=1}^{n} \sim [\forall \vec{Y}_j.(E_j \Rightarrow Z'_j)]_{j=1}^{m}$. Por definición de $\sim$, $n = m$ y para $i = 1, \ldots, n$ y una permutación p, tenemos $\forall \vec{X}_i.((A \wedge D_i) \Rightarrow Z_i) \equiv \forall \vec{Y}_{p(i)}.(E_{p(i)} \Rightarrow Z'_{p(i)})$. Luego, por el Lema 4, tenemos $\vec{X}_i = \vec{Y}_{p(i)}$, $A \wedge D_i \equiv E_{p(i)}$, y $Z_i = Z'_{p(i)}$.

Por lo tanto, existe I tal que $I \cup \bar{I} = \{1, \ldots, n\}$ y

$$PF(C_1) = [\forall \vec{Y}_{p(i)}.(E_{p(i)} \Rightarrow Z'_{p(i)})]_{i \in I} \qquad PF(C_2) = [\forall \vec{Y}_{p(i)}.(E_{p(i)} \Rightarrow Z'_{p(i)})]_{i \in \bar{I}}$$

Entonces, por el Corolario 1,

$$C_1 \equiv \bigwedge_{i \in I} \forall \vec{Y}_{p(i)}.(E_{p(i)} \Rightarrow Z'_{p_i}) \equiv \bigwedge_{i \in I} \forall \vec{X}_i.((A \wedge D_i) \Rightarrow Z_i)$$

$$C \equiv \bigwedge_{i \in \bar{I}} \forall \vec{X}_i.((A \wedge D_i) \Rightarrow Z_i)$$

Sean $B_1 = \bigwedge_{i \in I} \forall \vec{X}_i.(D_i \Rightarrow Z_i)$ y $B_2 = \bigwedge_{i \in \bar{I}} \forall \vec{X}_i.(D_i \Rightarrow Z_i)$. Entonces, $C_1 \equiv A \Rightarrow B_1$ y $C_2 \equiv A \Rightarrow B_2$. Además, también por el Corolario 1, tenemos $B \equiv \bigwedge_{i=1}^{n} \forall \vec{X}_i.(D_i \Rightarrow Z_i) \equiv B_1 \wedge B_2$. □

Lema 10 (Substitución)**.**

1. *Para todo Γ, x, r, s, A, B tales que $\Gamma, x : B \vdash r : A$ y $\Gamma \vdash s : B$, tenemos $\Gamma \vdash [x := s]r : A$.*
2. *Para todo Γ, r, X, A, B tales que $\Gamma \vdash r : A$, tenemos $[X := B]\Gamma \vdash [X := B]r : [X := B]A$.*

Prueba.

1. Por inducción en r.
   - Sea $r = x$. Por el Lema 9, $A \equiv B$, por lo tanto, $\Gamma \vdash s : A$. Dado que $[x := s]x = s$, tenemos $\Gamma \vdash [x := s]x : A$.
   - Sea $r = y$, con $y \neq x$. Dado que $[x := s]y = y$, tenemos $\Gamma \vdash [x := s]y : A$.
   - Sea $r = \lambda x^C.t$. Tenemos $[x := s](\lambda x^C.t) = \lambda x^C.t$, por lo tanto, $\Gamma \vdash [x := s](\lambda x^C.t) : A$.
   - Sea $r = \lambda y^C.t$, con $y \neq x$. Por el Lema 9, $A \equiv C \Rightarrow D$ y $\Gamma, y : C \vdash t : D$. Por hipótesis inductiva, $\Gamma, y : C \vdash [x := s]t : D$, y luego, por la regla, $(\Rightarrow_i)$, $\Gamma \vdash \lambda y^C.[x := s]t : C \Rightarrow D$. Dado que $\lambda y^C.[x := s]t = [x := s](\lambda y^C.t)$, usando la regla $(\equiv)$, $\Gamma \vdash [x := s](\lambda x^C.t) : A$.

- Sea $r = tu$. Por el Lema 9, $\Gamma \vdash t : C \Rightarrow A$ y $\Gamma \vdash u : C$. Por hipótesis inductiva, $\Gamma \vdash [x := s]t : C \Rightarrow A$ y $\Gamma \vdash [x := s]u : C$, y luego, por la regla, $(\Rightarrow_e)$, $\Gamma \vdash ([x := s]t)([x := s]u) : A$. Dado que $([x := s]t)([x := s]u) = [x := s](tu)$, tenemos $\Gamma \vdash [x := s](tu) : A$.
- Sea $r = \langle t, u\rangle$. Por el Lema 9, $\Gamma \vdash t : C$ y $\Gamma \vdash u : D$, con $A \equiv C \wedge D$. Por hipótesis inductiva, $\Gamma \vdash [x := s]t : C$ y $\Gamma \vdash [x := s]u : D$, y luego, por la regla, $(\wedge_i)$, $\Gamma \vdash \langle [x := s]t, [x := s]u\rangle : C \wedge D$. Dado que $\langle [x := s]t, [x := s]u\rangle = [x := s]\langle t, u\rangle$, usando la regla $(\equiv)$, tenemos $\Gamma \vdash [x := s]\langle t, u\rangle : A$.
- Sea $r = \pi_A(t)$. Por el Lema 9, $\Gamma \vdash t : A \wedge C$. Por hipótesis inductiva, $\Gamma \vdash [x := s]t : A \wedge C$, y luego, por la regla, $(\wedge_e)$, $\Gamma \vdash \pi_A([x := s]t) : A$. Dado que $\pi_A([x := s]t) = [x := s](\pi_A(t))$, tenemos $\Gamma \vdash [x := s](\pi_A(t)) : A$.
- Sea $r = \Lambda X.t$. Por el Lema 9, $A \equiv \forall X.C$ y $\Gamma \vdash t : C$. Por hipótesis inductiva, $\Gamma \vdash [x := s]t : C$, y luego, por la regla, $(\forall_i)$, $\Gamma \vdash \Lambda X.[x := s]t : \forall X.C$. Dado que $\Lambda X.[x := s]t = [x := s](\Lambda X.t)$, usando la regla $(\equiv)$, tenemos $\Gamma \vdash [x := s](\Lambda X.t) : A$.
- Sea $r = t[C]$. Por el Lema 9, $A \equiv [X := C]D$ y $\Gamma \vdash t : \forall X.D$. Por hipótesis inductiva, $\Gamma \vdash [x := s]t : \forall X.D$, y luego, por la regla, $(\forall_e)$, $\Gamma \vdash ([x := s]t)[C] : [X := C]D$. Dado que $([x := s]t)[C] = [x := s](t[C])$, usando la regla $(\equiv)$, tenemos $\Gamma \vdash [x := s](t[C]) : A$.

2. Por inducción en la relación de tipado.
   - (ax): Sea $\Gamma, x : A \vdash x : A$. Entonces, usando la regla (ax), tenemos $[X := B]\Gamma, x : [X := B]A \vdash [X := B]x : [X := B]A$.
   - $(\equiv)$: Sea $\Gamma \vdash r : A$, con $A \equiv C$. Por hipótesis inductiva, $[X := B]\Gamma \vdash [X := B]r : [X := B]C$. Dado que $A \equiv C$, $[X := B]A \equiv [X := B]C$. por la regla $(\equiv)$, tenemos $[X := B]\Gamma \vdash [X := B]r : [X := B]A$.
   - $(\Rightarrow_i)$: Sea $\Gamma \vdash \lambda x^C.t : C \Rightarrow D$. Por hipótesis inductiva, $[X := B]\Gamma, x : [X := B]C \vdash [X := B]t : [X := B]D$. por la regla $(\Rightarrow_i)$, $[X := B]\Gamma \vdash \lambda x^{[X:=B]C}.[X := B]t : [X := B]C \Rightarrow [X := B]D$. Dado que $\lambda x^{[X:=B]C}.[X := B]t = [X := B](\lambda x^C.t)$, tenemos $[X := B]\Gamma \vdash [X := B](\lambda x^C.t) : [X := B](C \Rightarrow D)$.
   - $(\Rightarrow_e)$: Sea $\Gamma \vdash ts : D$. Por hipótesis inductiva, $[X := B]\Gamma \vdash [X := B]t : [X := B](C \Rightarrow D)$ y $[X := B]\Gamma \vdash [X := B]s : [X := B]C$. Since $[X := B](C \Rightarrow D) = [X := B]C \Rightarrow [X := B]D$, usando la regla $(\Rightarrow_e)$, tenemos $[X := B]\Gamma \vdash ([X := B]t)([X := B]s) : [X := B]D$. Dado que $([X := B]t)([X := B]s) = [X := B](ts)$, tenemos $[X := B]\Gamma \vdash [X := B](ts) : [X := B]D$.
   - $(\wedge_i)$: Sea $\Gamma \vdash \langle t, s\rangle : C \wedge D$. Por hipótesis inductiva, $[X := B]\Gamma \vdash [X := B]t : [X := B]C$ y $[X := B]\Gamma \vdash [X := B]s : [X := B]D$. por la regla $(\wedge_i)$, $[X := B]\Gamma \vdash \langle [X := B]t, [X := B]s\rangle : [X := B]C \wedge [X := B]D$. Dado que $\langle [X := B]t, [X := B]s\rangle = [X := B]\langle t, s\rangle$, y $[X := B]C \wedge [X := B]D = [X := B](C \wedge D)$, tenemos $[X := B]\Gamma \vdash [X := B]\langle t, s\rangle : [X := B](C \wedge D)$.
   - $(\wedge_e)$: Sea $\Gamma \vdash t : C \wedge D$. Por hipótesis inductiva, $[X := B]\Gamma \vdash [X := B]t : [X := B](C \wedge D)$. Dado que $[X := B](C \wedge D) = [X := B](C) \wedge [X := B](D)$, usando la regla $(\wedge_e)$tenemos $[X := B]\Gamma \vdash \pi_{[X:=B]C}([X := B]t) :$

$[X := B](C)$. Dado que $\pi_{[X:=B]C}([X := B]t) = [X := B]\pi_C(t)$, tenemos $[X := B]\Gamma \vdash [X := B]\pi_C(t) : [X := B](C)$.

- $(\forall_i)$: Sea $\Gamma \vdash \Lambda Y.t : \forall Y.C$, con $X \notin FTV(\Gamma)$. Por hipótesis inductiva, $[X := B]\Gamma \vdash [X := B]t : [X := B]C$. Dado que $X \notin FTV(\Gamma)$, $X \notin FTV([X := B]\Gamma)$. por la regla $(\forall_i)$, tenemos $[X := B]\Gamma \vdash \Lambda Y.[X := B]t : \Lambda Y.[X := B]C$. Dado que $\Lambda Y.[X := B]t = [X := B]\Lambda Y.t$, y $\forall Y.[X := B]C = [X := B]\forall Y.C$, tenemos $[X := B]\Gamma \vdash [X := B]\Lambda Y.t : [X := B]\forall Y.C$.
- $(\forall_e)$: Sea $\Gamma \vdash t[D] : [Y := D]C$. Por hipótesis inductiva, $[X := B]\Gamma \vdash [X := B]t : [X := B]\forall Y.C$. Dado que $[X := B]\forall Y.C = \forall Y.[X := B]C$, por la regla $(\forall_e)$, tenemos $[X := B]\Gamma \vdash ([X := B]t)[[X := B]D] : [Y := [X := B]D][X := B]C$.
  Dado que $([X := B]t)[[X := B]D] = [X := B](t[D])$, y $[Y := [X := B]D][X := B]C = [X := B][Y := D]C$, tenemos $[X := B]\Gamma \vdash [X := B](t[D]) : [X := B][Y := D]C$. □

B. Pruebas y lemas auxiliares de la Sección 5

El tamaño de un término no es invariante en la equivalencia $\rightleftarrows$. Por ejemplo, contando la cantidad de *lambda*s en un término, podemos ver que $\lambda x^A.\langle r, s\rangle$ difiere de $\langle \lambda x^A.r, \lambda x^A.s\rangle$. Por ello introducimos una medida $M(\cdot)$ para términos.

Definición 7 (Medida de términos).

$$\begin{aligned} P(x) &= 0 & M(x) &= 1 \\ P(\lambda x^A.r) &= P(r) & M(\lambda x^A.r) &= 1 + M(r) + P(r) \\ P(rs) &= P(r) & M(rs) &= M(r) + M(s) + P(r)M(s) \\ P(\langle r, s\rangle) &= 1 + P(r) + P(s) & M(\langle r, s\rangle) &= M(r) + M(s) \\ P(\pi_A(r)) &= P(r) & M(\pi_A(r)) &= 1 + M(r) + P(r) \\ P(\Lambda X.r) &= P(r) & M(\Lambda X.r) &= 1 + M(r) + P(r) \\ P(r[A]) &= P(r) & M(r[A]) &= 1 + M(r) + P(r) \end{aligned}$$

Lema 13. *Para todo r, s tales que $r \rightleftarrows s$, vale $P(r) = P(s)$.* □

Prueba. Analizamos el caso de cada regla de la Tabla 3, y concluimos con inducción estructural. □

Lema 14. *Para todo r, s tales que $r \rightleftarrows s$, vale $M(r) = M(s)$.* □

Prueba. Analizamos el caso de cada regla de la Tabla 3, y concluimos con inducción estructural. □

Lema 15. *Para todo r, s, X, A,*

$$M(\lambda x^A.r) > M(r) \quad M(\langle r, s\rangle) > M(s) \quad M(\langle r, s\rangle) > M(r) \quad M(\Lambda X.r) > M(r)$$
$$M(rs) > M(r) \quad M(rs) > M(s) \quad M(\pi_A(r)) > M(r) \quad M(r[A]) > M(r) \quad \square$$

Prueba. Para todo t, $M(t) \geq 1$. Concluimos con una inspección de casos. □

Al extender al cálculo lambda con pares, probar que si $r_1 \in \mathsf{SN}$ y $r_2 \in \mathsf{SN}$ entonces $\langle r_1, r_2 \rangle \in \mathsf{SN}$ es sencillo. Pero no es el caso de Sistema I y SIP, donde esta propiedad (Lemma 18) resulta más compleja de probar, ya que requiere una caracterización de los términos equivalentes al producto $\langle r_1, r_2 \rangle$ (Lemma 16) y de todos los reductos del mismo (Lemma 17).

Lema 16. *Para todo r, s, t tales que $\langle r, s \rangle \rightleftarrows^* t$, tenemos*

1. $t = \langle u, v \rangle$ *donde*
   a) $u \rightleftarrows^* \langle t_{11}, t_{21} \rangle$ *y* $v \rightleftarrows^* \langle t_{12}, t_{22} \rangle$ *con* $r \rightleftarrows^* \langle t_{11}, t_{12} \rangle$ *y* $s \rightleftarrows^* \langle t_{21}, t_{22} \rangle$, *o*
   b) $v \rightleftarrows^* \langle w, s \rangle$ *con* $r \rightleftarrows^* \langle u, w \rangle$, *o cualquier de los tres casos simétricos, o*
   c) $r \rightleftarrows^* u$ *y* $s \rightleftarrows^* v$, *o el caso simétrico.*
2. $t = \lambda x^A.a$ *y* $a \rightleftarrows^* \langle a_1, a_2 \rangle$ *con* $r \rightleftarrows^* \lambda x^A.a_1$ *y* $s \rightleftarrows^* \lambda x^A.a_2$.
3. $t = av$ *y* $a \rightleftarrows^* \langle a_1, a_2 \rangle$, *con* $r \rightleftarrows^* a_1 v$ *y* $s \rightleftarrows^* a_2 v$.
4. $t = \Lambda X.a$ *y* $a \rightleftarrows^* \langle a_1, a_2 \rangle$ *con* $r \rightleftarrows^* \Lambda X.a_1$ *y* $s \rightleftarrows^* \Lambda X.a_2$.
5. $t = a[A]$ *y* $a \rightleftarrows^* \langle a_1, a_2 \rangle$, *con* $r \rightleftarrows^* a_1[A]$ *y* $s \rightleftarrows^* a_2[A]$.

Prueba. Por inducción doble, primero sobre $M(t)$ y luego sobre la longitud de la relación $\rightleftarrows^*$. Sea la prueba $\langle r, s \rangle \rightleftarrows^* t' \rightleftarrows t$ con una derivación más corta $\langle r, s \rangle \rightleftarrows^* t'$. Por la segunda hipótesis inductiva, el término t' tiene la forma descripta por el lema. Consideramos los cinco casos y en cada uno las posibles reglas que transforman t' en t. □

Lema 17. *Para todo r_1, r_2, s, t tales que $\langle r_1, r_2 \rangle \rightleftarrows^* s \hookrightarrow t$, existen u_1, u_2 tales que $t \rightleftarrows^* \langle u_1, u_2 \rangle$ y: (1) $r_1 \to u_1$ y $r_2 \to u_2$, (2) $r_1 \to u_1$ y $r_2 \rightleftarrows^* u_2$, o (3) $r_1 \rightleftarrows^* u_1$ y $r_2 \to u_2$.*

Prueba. Por inducción en $M(\langle r_1, r_2 \rangle)$, considerando que por el Lema 16 conocemos las formas que puede tener s, y analizando los posibles reductos de s. □

Lema 18. *Para todo r_1, r_2 tales que $r_1 \in \mathsf{SN}$ y $r_2 \in \mathsf{SN}$, vale $\langle r_1, r_2 \rangle \in \mathsf{SN}$.*

Prueba. Por el Lema 17, de una secuencia de reducción que parte desde $\langle r_1, r_2 \rangle$ podemos extraer una partiendo desde r_1, desde r_2 o desde ambos. Luego esta secuencia de reducción es finita. □

Lema 19 (Adecuación de variables). *Para todo A y x^A, vale $x^A \in [\![A]\!]$.*

Prueba. Necesitamos probar $K_A^X(\!|x|\!) \in \mathsf{SN}$. En el término $K_A^X(\!|x|\!)$, la variable x se encuentra en una posición que no crea ningún redex, por lo tanto los únicos redexes son aquellos en $\mathcal{T}(K_A^X)$, que ya están en SN. Entonces $K_A^X(\!|x|\!) \in \mathsf{SN}$. □

Lema 20 (Adecuación de la proyección). *Para todo r, A, B tales que $r \in [\![A \wedge B]\!]$, vale $\pi_A(r) \in [\![A]\!]$.*

Prueba. Necesitamos probar $K_A^X(\!|\pi_A(r)|\!) \in \mathsf{SN}$. Sea $K'^X_{A\wedge B} = K_A^X(\!|\pi_A((\!|)^{A\wedge B})|\!)$, y dado que $r \in [\![A \wedge B]\!]$, tenemos $K'^X_{A\wedge B}(\!|r|\!) = K_A^X(\!|\pi_A(r)|\!) \in \mathsf{SN}$. □

Lema 21 (Adecuación de la aplicación). *Para todo r, s, A, B tales que $r \in [\![A \Rightarrow B]\!]$ y $s \in [\![A]\!]$, vale $rs \in [\![B]\!]$.*

Prueba. Necesitamos probar $K_B^X(\!|rs|\!) \in \mathsf{SN}$. Sea $K'^X_{A\Rightarrow B} = K_B^X(\!|\!|)^{A\Rightarrow B}s|\!)$, y dado que $r \in [\![A \Rightarrow B]\!]$, tenemos $K'^X_{A\Rightarrow B}(\!|r|\!) = K_B^X(\!|rs|\!) \in \mathsf{SN}$. □

Lema 22 (Adecuación de la aplicación de tipos). *Para todo r, X, A, B tales que $r \in [\![\forall X.A]\!]$, vale $r[B] \in [\![[X := B]A]\!]$.*

Prueba. Necesitamos probar $K^Y_{[X:=B]A}(\!|r[B]|\!) \in \mathsf{SN}$. Considerando el contexto $K'^Y_{\forall X.A} = K^Y_{[X:=B]A}(\!|\!|)^{\forall X.A}[B]|\!) \in \mathsf{SN}$, y dado que $r \in [\![\forall X.A]\!]$, tenemos $K'^Y_{\forall X.A}(\!|r|\!) = K^Y_{[X:=B]A}(\!|r[B]|\!) \in \mathsf{SN}$. □

Lema 23 (Adecuación del producto). *Para todo r, s, A, B tales que $r \in [\![A]\!]$ y $s \in [\![B]\!]$, vale $\langle r, s\rangle \in [\![A \wedge B]\!]$.*

Prueba. Necesitamos probar $K^X_{A\wedge B}(\!|\langle r, s\rangle|\!) \in \mathsf{SN}$. Procedemos por inducción en la cantidad de proyecciones en $K^X_{A\wedge B}$. Dado que el agujero de $K^X_{A\wedge B}$ tiene tipo $A \wedge B$, y $K^X_{A\wedge B}(\!|t|\!)$ tiene tipo X para todo t de tipo $A\wedge B$, podemos asumir, sin pérdida de generalidad, que el contexto $K^X_{A\wedge B}$ tiene forma $K'^X_C(\!|\pi_C(|\!|)^{A\wedge B}\alpha_1 \dots \alpha_n)|\!)$, donde cada α_i es o bien un término o bien un argumento de tipo. Probamos que $K'^X_C(\!|\pi_C(\langle r\alpha_1 \dots \alpha_n, s\alpha_1 \dots \alpha_n\rangle)|\!) \in \mathsf{SN}$ mostrando, con mayor generalidad, que si r' y s' son dos reductos de $r\alpha_1 \dots \alpha_n$ y $s\alpha_1 \dots \alpha_n$, entonces $K'^X_C(\!|\pi_C(\langle r', s'\rangle)|\!) \in \mathsf{SN}$. Para ello mostramos que todos sus reductos en un paso están en SN, por inducción en $|K'^X_C| + |r'| + |s'|$. La prueba detallada puede encontrarse en el apéndice publicado en [27]. □

Lema 24 (Adecuación de la abstracción). *Para todo t, r, x, A, B tales que $t \in [\![A]\!]$ y $[x := t]r \in [\![B]\!]$, vale $\lambda x^A.r \in [\![A \Rightarrow B]\!]$.*

Prueba. Por inducción en $M(r)$.

- Si $r \rightleftarrows^* \langle r_1, r_2\rangle$, entonces por el Lema 9, tenemos $B \equiv B_1 \wedge B_2$ con r_1 de tipo B_1 y r_2 de tipo B_2, y entonces por el Lema 10, $[x := t]r_1$ tiene tipo B_1 y $[x := t]r_2$ tiene tipo B_2. Dado que $[x := t]r \in [\![B]\!]$, vale $\langle [x := t]r_1, [x := t]r_2\rangle \in [\![B]\!]$. Por el Lema 20, $[x := t]r_1 \in [\![B_1]\!]$ y $[x := t]r_2 \in [\![B_2]\!]$. Por hipótesis inductiva, $\lambda x^A.r_1 \in [\![A \Rightarrow B_1]\!]$ y $\lambda x^A.r_2 \in [\![A \Rightarrow B_2]\!]$, entonces, por el Lema 23, $\lambda x^A.r \rightleftarrows^* \langle \lambda x^A.r_1, \lambda x^A.r_2\rangle \in [\![(A \Rightarrow B_1) \wedge (A \Rightarrow B_2)]\!]$. Finalmente, por el Lema 11, tenemos $[\![(A \Rightarrow B_1) \wedge (A \Rightarrow B_2)]\!] = [\![A \Rightarrow B]\!]$.
- Si $r \not\rightleftarrows^* \langle r_1, r_2\rangle$, necesitamos probar que para todo contexto de eliminación $K^X_{A\Rightarrow B}$, vale $K^X_{A\Rightarrow B}(\!|\Lambda X.r|\!) \in \mathsf{SN}$. Como r y todos los términos en $\mathcal{T}(K^X_{A\Rightarrow B})$ están en SN, procedemos por inducción en el orden lexicográfico de $(|K^X_{A\Rightarrow B}| + |r|, M(r))$ para mostrar que todos los reductos en un paso de $K^X_{A\Rightarrow B}(\!|\lambda x^A.r|\!)$ están en SN. Dado que r no es un producto, sus únicos reductos en un paso son los siguientes.
  - Un término en el que la reducción tiene lugar en uno de los términos en $\mathcal{T}(K^X_{A\Rightarrow B})$ o en r, por lo que aplicamos la hipótesis inductiva.

- $K'^{X}_{B}(\!|[x := s]r|\!)$, con $K^{X}_{A\Rightarrow B}(\!|\lambda x^{A}.r|\!) = K'^{X}_{B}[(\lambda x^{A}.r)s]$. Como $[x := s]r \in [\![B]\!]$, tenemos $K'^{X}_{B}(\!|[x := s]r|\!) \in \mathsf{SN}$.
- $K'^{X}_{A\Rightarrow B'}(\!|\lambda x^{A}.[X := C]r'|\!)$, con $r \rightleftarrows^{*}$-equivalente a $\Lambda X.r'$, $B \equiv \forall X.B'$, y $K^{X}_{A\Rightarrow B}(\!|\lambda x^{A}.\Lambda X.r'|\!)$ es igual a $K'^{X}_{A\Rightarrow B'}(\!|(\lambda x^{A}.\Lambda X.r')[C]|\!)$. Como $M([X := C]r') < M(\Lambda X.r')$, aplicamos la hipótesis inductiva. □

Lema 25 (Adecuación de la abstracción de tipos). *Para todo r, X, A, B tales que $[X := B]r \in [\![[X := B]A]\!]$, vale $\Lambda X.r \in [\![\forall X.A]\!]$.*

Prueba. Por inducción en $M(r)$, con una prueba similar a la del Lema 24. La prueba detallada puede encontrarse en el apéndice publicado en [27]. □