Aproximando los escalares de un λ-cálculo algebraico mediante cotas inferiores

Tesina de grado presentada
por

Pablo Buiras
B-3745/1

al
Departamento de Ciencias de la Computación
en cumplimiento parcial de los requerimientos
para la obtención del grado de

Licenciado en Ciencias de la Computación

Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario
Av. Pellegrini 250, Rosario, República Argentina

Noviembre 2011

Supervisores

Alejandro Díaz-Caro

LIPN - UMR CNRS 7030
Institut Galilée - Université Paris-Nord
99, avenue Jean-Baptiste Clément
93430 Villetaneuse, Francia

Mauro Jaskelioff

Departamento de Ciencias de la Computación
Facultad de Cs. Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario
Av. Pellegrini 250
S2000BTP Rosario, Santa Fe, Argentina

Dedicado a la memoria de mi abuelo Toto

Índice general

Agradecimientos

Este trabajo no se podría haber realizado si no fuera por todas las personas que me acompañaron en este viaje que empezó hace tanto tiempo, y que finalmente llega a su fin. Estos años estuvieron marcados por un sinnúmero de experiencias y actividades, y experiencias dentro de las experiencias; a todas las personas que, de una u otra forma, compartieron su tiempo conmigo, quiero extender mis agradecimientos, porque de cada uno me llevo algo distinto:

A mi familia, por el cariño y el apoyo de siempre. A mis padres, que son los que me enseñaron las cosas más importantes, les agradezco porque confiaron en mí y me alentaron desde el principio a perseguir mis metas. A Nacho, mi hermano, porque desde lo cotidiano logra enriquecer mi vida inmensamente, por esos momentos tan valiosos en donde el tiempo y el mundo se detienen.

A mis directores, Alejandro y Mauro, por guiarme y mostrarme el camino, por aguantarse las horas de *reunión virtual*, por tantas conversaciones interesantes y constructivas, por los consejos, las críticas y las correcciones. Hubiera sido imposible presentar este trabajo en su forma actual sin sus aportes e ideas.

A mis compañeros, de todos los años de la carrera (desde los más chicos a los más grandes), con quienes tuve el honor de compartir este viaje en sus diferentes etapas, y que hoy en día puedo decir que están entre mis amigos más cercanos. No los voy a nombrar porque son demasiados, y aparte la enumeración impone un orden que para mí no existe; de todos modos, ustedes saben quiénes son. Gracias por estar siempre, por las risas, por las charlas inspiradoras, los cumpleaños, los asados, las reuniones, y las tardes de cervezas. Gracias por tantos momentos compartidos e inolvidables; su amistad es lo más valioso que me llevo de mi paso por la carrera.

A los *ñoños*, *selenitas* e *infras* de todos los tiempos, por esos espacios de *ñoñez* saludable, los asados en el club Mitre, las cenas en la Maltería y los encuentros en Antares; por las discusiones científicas, el café y la abstracción en su forma más pura.

A todos los que integran esta gran *familia* que es LCC, por hacer de esta comunidad algo único, por la solidaridad para ayudarnos entre nosotros y la *buena onda* que, en general, impera en nuestras aulas. Éste es nuestro mayor tesoro, y espero que nunca se pierda.

A mis docentes, tanto de la facultad como del secundario. A Fidel, por contagiarme la pasión por la docencia, por tanta *iluminación* e inspiración. A Guido, por enseñarme a encarar la vida como un *científico*, dentro y fuera del aula. A José María, por enseñarme a hacer preguntas y a valorar los libros.

Al *ilustre* club de go de Rosario, por la calidez con la que me recibieron en el grupo desde el principio, por los partidos y los encuentros, por compartir conmigo su pasión por el go y permitirme encontrar en este juego una forma ideal de equilibrio.

A los *barrieros* por compartir conmigo casi dos años en los que, cada sábado por la mañana, logro transportarme a otra realidad, y ver que hay otro mundo, y está en éste. A cada uno de los *barrieritos*, por permitirme formar parte de su vida y, en muchas ocasiones, ser ellos quienes dan más de lo que reciben.

A todos, a los que mencioné y a los que olvidé, les digo *¡gracias!*

Capítulo 1

Introducción

El objetivo de este trabajo es el de desarrollar la metateoría para una variante de λ-cálculo tipado. En líneas generales, puede enmarcarse dentro del área de teoría de la computación, particularmente en lo que se refiere a los fundamentos de lenguajes de programación y sistemas de tipos.

El λ-cálculo, descubierto por Alonzo Church [Chu33], es una teoría de funciones que sirve como modelo de cómputo. Este cálculo consta de un lenguaje de expresiones o *términos*, junto con un mecanismo para *reducirlas*, es decir, transformarlas en otras expresiones más simples. En forma intuitiva, podemos decir que el proceso de cómputo se representa mediante la reducción sucesiva de una expresión hasta llegar a una expresión irreducible, que será el resultado del cómputo. El estudio del λ-cálculo es relevante por su valor teórico y práctico; mediante este formalismo, pueden explorarse cuestiones fundamentales en teoría de la computación, lógica, semántica, diseño de lenguajes y programación funcional.

Actualmente, existen infinidad de variantes del λ-cálculo, cada uno de ellos con distintas características particulares, aunque todos tienen en común la misma noción esencial de cómputo. Una de las principales variantes del λ-cálculo surge con la introducción de los *sistemas de tipos*. Los tipos son etiquetas que permiten clasificar a los programas según el valor que producen. Esto permite que, mediante un análisis de estas etiquetas, se puedan descartar programas inválidos. En estos sistemas, los programas válidos son aquéllos a los que puede asignarse un tipo, mientras que los que no tienen tipo son rechazados. Desde el punto de vista de la programación, el uso de sistemas de tipos implica ciertas ventajas importantes; los tipos permiten detectar errores comunes, sirven como especificación rudimentaria del comportamiento de un programa, y son útiles para estructurar y organizar el flujo de información.

Cuando un programa tiene tipo, usualmente se tiene la garantía de que "no puede pasar nada malo" [Mil78], entendida como la propiedad de ejecutarse sin producir un error en tiempo de ejecución (también llamada *consistencia*) y, adicionalmente, la certeza de que todo programa bien tipado termina (conocida como *normalización fuerte*). En algunos casos, ocurre que hay más de una reducción posible para una expresión dada, con lo cual aparecen ramificaciones. Generalmente, se desea que, a pesar de estas ramificaciones, cuando el cómputo finaliza, se llegue siempre a la misma expresión, independientemente del camino de reducción que se elija. Esta propiedad se conoce como *confluencia*. En su conjunto, estas tres propiedades (consistencia, normalización fuerte, y confluencia) comprenden el andamiaje teórico básico que hace que un cálculo tipado se comporte como debe, sin contradecir a la intuición que se tiene sobre su funcionamiento.

La consistencia se suele presentar como un par de propiedades: la primera asegura que los términos bien tipados no son erróneos; la segunda, que el tipado se preserva a través de la reducción. A esta última se la conoce como *subject reduction*. Por razones de delimitación del trabajo, nos centraremos solamente en esta propiedad en lo que a consistencia se refiere.

A su vez, los tipos tienen un valor teórico intrínseco que surge de su interpretación desde el punto de vista lógico: cada tipo se corresponde con una proposición, y cada expresión con ese tipo se corresponde con una demostración de dicha proposición. Esta noción se conoce como *correspondiencia de Curry-Howard* [Cur34, Cur58, How80], y pone de manifiesto la conexión profunda que existe entre la computación y la lógica. Esta correspondencia permite extraer, de cada cálculo tipado, una lógica asociada.

En este trabajo, estudiaremos una familia particular de cálculos, llamados *cálculos algebraicos* [AD08, Vau09], que ha despertado gran interés en los últimos tiempos. Estos cálculos tienen la particularidad de combinar λ-cálculo con álgebra lineal, permitiendo combinaciones lineales de términos, es decir, términos de la forma $\alpha.\mathbf{t} + \beta.\mathbf{r}$ donde α y β son escalares y $\mathbf{t}$ y $\mathbf{r}$ son términos. Estos cálculos tienen su origen en dos comunidades diferentes: computación cuántica, para la cual los cálculos algebraicos son un posible modelo; y lógica lineal, ya que este cálculo es el resultado de eliminar el operador diferencial del λ-cálculo diferencial [ER01], un cálculo basado en lógica lineal.

Teniendo en cuenta las ventajas ya expuestas de los sistemas de tipos, es natural querer encontrar un sistema de tipos para cálculos algebraicos. En particular, tomaremos como punto de partida un cálculo no tipado llamado λ_{lin} [AD08], que sirve como plataforma para varias aplicaciones, como computación cuántica, computación paralela, y computación probabilística, entre otras. Actualmente, existen tres sistemas de tipos para λ_{lin}: *Scalar* [ADC11], *Additive* [DCP11] y *Vectorial* [ADCV11].

Siendo cada uno el resultado de una etapa distinta del estudio del tipado para cálculos algebraicos, éstos tienen características particulares que los vuelven interesantes en sí mismos. Scalar introdujo la idea de medir la "cantidad" de cada tipo presente en un término, reflejando esta información en el sistema de tipos. Additive estudia el fragmento aditivo del cálculo (no permite escalares) y se caracteriza porque puede interpretarse en un cálculo ya conocido (System F con pares). Vectorial es un cálculo complejo que combina ambos enfoques, introduciendo un espacio vectorial al nivel de los tipos que permite expresar en forma precisa la estructura algebraica de los términos.

En el presente trabajo, proponemos un sistema de tipos (λCA, por *Complete Additive*) para λ_{lin} que, en esencia, es una versión de Additive extendida al cálculo completo, pero más simple que Vectorial. La idea central de λCA será la de aproximar los escalares que aparecen en los términos mediante cotas inferiores, y reflejar esta información en los tipos. Como resultado de esta aproximación, obtenemos un cálculo más expresivo que Scalar y Additive y a la vez menos complejo que Vectorial, a cambio de la pérdida de precisión. Además, λCA tiene *subject reduction*, normalización fuerte y es confluente. A su vez, existe una interpretación abstracta de λCA en System F con pares. Uno de los aportes centrales del trabajo es que podemos obtener la confluencia del sistema como consecuencia de la normalización fuerte. Esto nos permite simplificar considerablemente la definición del mecanismo de reducción, obteniendo uno que no tiene tantas restricciones a la hora de aplicar las reglas, a diferencia de λ_{lin}, que necesita aplicar las reducciones en un orden especial para garantizar la confluencia. Estos resultados sobre λCA aparecen en un artículo [BDCJ11] que fue presentado en el workshop *Logical and Semantic Frameworks, with Applications* (LSFA) en 2011, y que será publicado como parte de un volumen de la serie *Electronic Proceedings in Theoretical Computer Science* (EPTCS).

Uno de los objetivos finales de esta línea de investigación en cálculos algebraicos tipados es el de obtener un cálculo que caracterice a la computación cuántica y así llegar, mediante la correspondencia de Curry-Howard, a una lógica cuántica. El cálculo presentado en este trabajo puede considerarse un paso más en dirección a este resultado, a pesar de no poder aplicarse directamente a este dominio.

Concretamente, los aportes de esta tesina son:

- El cálculo λCA, un sistema de tipos para λ_{lin} que tiene una interpretación abstracta en System F con pares;
- Demostraciones de las propiedades de *subject reduction*, normalización fuerte y confluencia para λCA;
- Un mecanismo de reducción más simple que el de λ_{lin}.

En el capítulo 2, presentamos el marco teórico sobre λ-cálculo y sistemas de tipos que utilizaremos en el resto del trabajo. El capítulo 3 presenta tres sistemas de tipos para λ_{lin} que se desarrollaron antes que λCA. En el capítulo 4 presentamos el cálculo λCA en sí, y comentamos las demostraciones de sus propiedades. En el capítulo 5 se presentan conclusiones y se comenta sobre posibles trabajos futuros. Las demostraciones completas pueden encontrarse en el apéndice.

Capítulo 2

Preliminares

En este capítulo, trataremos las nociones teóricas básicas necesarias para el resto del trabajo. Veremos una introducción al λ-cálculo como modelo de cómputo, algunos conceptos generales sobre sistemas de tipos y finalmente presentaremos System F, cálculo que tomaremos como base en los capítulos subsiguientes.

2.1. Una breve introducción al λ-cálculo

El λ-cálculo fue introducido por el lógico norteamericano Alonzo Church en la década de 1930, como parte de una teoría general de funciones y lógica que pretendía modelar la matemática [Chu32, Chu33]. El cálculo, en su forma moderna, consiste en un lenguaje formal de *términos* o *expresiones*, junto con una teoría ecuacional que define la equivalencia entre ellos, y una relación de reducción que representa la transformación de expresiones. La teoría ecuacional es útil para razonar sobre los términos del lenguaje, mientras que la reducción nos presenta el lenguaje como modelo de cómputo.

Informalmente, cuenta con dos construcciones básicas: la abstracción y la aplicación. Dado un término M, la abstracción $\lambda x.M$ intuitivamente denota a la función f tal que $f(x) = M$, donde x puede ocurrir en M. La aplicación $f\,N$ denota el resultado de pasar el término N como argumento a f, reemplazando todas las ocurrencias de x por el término N.

Estas dos construcciones actúan en conjunto, ya que la aplicación se utiliza para evaluar los elementos en la imagen de la función denotada por una abstracción. Así, el término $(\lambda x.x + 1)\,2$ se considera equivalente al término $2 + 1$, que resulta de reemplazar todas las ocurrencias libres de x en el término $x + 1$ por 2. Más generalmente, esta relación, llamada β-equivalencia, se escribe

$$(\lambda x.M)\,N = M[N/x]$$

donde $[N/x]$ denota la sustitución de x por N. Éste es el esquema de axioma esencial del cálculo.

La versión original del λ-cálculo resultó ser inconsistente desde un punto de vista lógico. Si se considera el término $T = (\lambda x.\neg(x\,x))$, donde $\neg\,M$ es un término que representa la negación lógica de M, entonces resulta que los términos $T\,T$ y $\neg(T\,T)$ son equivalentes. En otras palabras, $T\,T$ debe considerarse verdadero si y sólo si es falso. Esto no es más que una codificación en λ-cálculo de la conocida paradoja de Russell, formulada originalmente para teoría de conjuntos, en la que se construye el conjunto $R = \{x \mid x \notin x\}$, que implica que $R \in R$ si y sólo si $R \notin R$.

A pesar de esta inconsistencia en la teoría general, Church separó la parte útil para la definición de computabilidad, y esta presentación es la que hoy se conoce como λ-cálculo [Chu36]. A su vez, Kleene [Kle36] demostró que todas las funciones recursivas son representables en λ-cálculo.

Definición formal del λ-cálculo

Tomaremos como punto de partida un conjunto infinito numerable $\mathcal{V}$ de *variables*, es decir, identificadores alfabéticos de la forma x, y o z. Asumiremos que todas las letras del alfabeto se encuentran en

$\mathcal{V}$. Además, nos permitiremos el uso de subíndices numéricos para indicar más variables, de manera que x_1, x_2 y x_3 también pertenecen a $\mathcal{V}$.

Definición 2.1.1. Definimos el conjunto Λ de λ-términos, de forma inductiva, como el menor conjunto tal que

$$\frac{x \in \mathcal{V}}{x \in \Lambda} \qquad \frac{M \in \Lambda \quad x \in \mathcal{V}}{(\lambda x.M) \in \Lambda} \qquad \frac{M, N \in \Lambda}{(M\,N) \in \Lambda}.$$

Ejemplo 1. Algunos λ-términos válidos son

$$x \quad (x\,x) \quad (\lambda z.z) \quad (z\,(\lambda x.y)) \quad ((\lambda x.(\lambda y.(x\,y)))\,z).$$

Convenciones. Usualmente, para ahorrar paréntesis y facilitar la lectura de los términos, adoptamos las siguientes convenciones con respecto a notación:

- La aplicación asocia a izquierda, es decir que $M_1\,M_2\,M_3\,M_4$ significa $(((M_1\,M_2)\,M_3)\,M_4)$;
- Las abstracciones se extienden lo más a la derecha posible, es decir que $\lambda x.M\,N$ significa $(\lambda x.M\,N)$;
- Las abstracciones anidadas pueden agruparse bajo un solo símbolo λ, de manera que $\lambda xyz.M$ significa $(\lambda x.(\lambda y.(\lambda z.M)))$.

De esta manera, es posible omitir el uso de la mayoría de los paréntesis. Se utilizarán explícitamente para expresar términos que no se ajusten a estas convenciones, o cuando su omisión resulte confusa.

De acuerdo con estas convenciones, los términos del ejemplo 1 se escribirían

$$x \quad x\,x \quad \lambda z.z \quad z\,\lambda x.y \quad (\lambda xy.x\,y)\,z.$$

Notaremos con $\equiv$ a la igualdad sintáctica entre λ-términos; es decir, que $M \equiv N$ cuando M y N son exactamente el mismo término.

Variables libres y ligadas

En la abstracción $\lambda x.M$, decimos x es la variable *ligada*, o bien que la abstracción *liga* a la variable x. Si una ocurrencia de una variable no está ligada por ninguna abstracción, decimos que dicha ocurrencia está *libre*.

Definición 2.1.2. El conjunto de variables libres de un término M, denotado por $\mathrm{FV}(M)$, se define inductivamente:

$$\begin{array}{rcl} \mathrm{FV}(x) & = & \{x\} \\ \mathrm{FV}(M\,N) & = & \mathrm{FV}(M) \cup \mathrm{FV}(N) \\ \mathrm{FV}(\lambda x.M) & = & \mathrm{FV}(M) \backslash \{x\} \end{array}$$

El conjunto de variables ligadas de un término M, denotado por $\mathrm{BV}(M)$, se define inductivamente:

$$\begin{array}{rcl} \mathrm{BV}(x) & = & \emptyset \\ \mathrm{BV}(M\,N) & = & \mathrm{BV}(M) \cup \mathrm{BV}(N) \\ \mathrm{BV}(\lambda x.M) & = & \mathrm{BV}(M) \cup \{x\} \end{array}$$

Ejemplo 2. En el término $T \equiv (\lambda x.x\,(\lambda y.z))$, se tiene que x ocurre ligada ($x \in \mathrm{BV}(T)$), mientras que z ocurre libre ($\mathrm{FV}(T) = \{z\}$).

Un λ-término M se dice *cerrado* si $\mathrm{FV}(M) = \emptyset$, es decir, si no tiene variables libres.

Cuando una variable está ligada, su nombre es irrelevante para el significado del término. Por ejemplo, los términos $(\lambda y.y)$ y $(\lambda x.x)$ tienen la misma estructura; ambos representan la función identidad y se comportan como tal, independientemente del nombre de las variables. Así, nos permitimos *renombrar* las variables ligadas de un término cuando nos sea conveniente.

La sustitución sólo reemplaza las ocurrencias libres de una variable, es decir, que $((\lambda y.y)\, y)[N/y] \equiv (\lambda y.y)\, N$. Además, si consideramos la sustitución

$$(\lambda x.x\, z)[(x\, y)/z]$$

vemos que es necesario *renombrar* la variable x en esta abstracción, para que no entre en conflicto con la x del término que se está reemplazando por z. Para ello, se elige un nombre nuevo, y la sustitución resulta

$$(\lambda x.x\, z)[(x\, y)/z] \equiv (\lambda u.u\, (x\, y)).$$

Nótese que, si no realizamos este renombramiento, el término resultante sería $(\lambda x.x\, (x\, y))$, que intuitivamente representa una función diferente. Decimos que esta operación evita la *captura* de la variable x. Salvo que se indique lo contrario, consideraremos a los términos que sólo difieren en los nombres de sus variables como equivalentes.

Definición 2.1.3. De acuerdo con estas consideraciones, la sustitución se define inductivamente de la siguiente manera:

$$\begin{array}{rcl}
x[R/x] & \equiv & R \\
y[R/x] & \equiv & y \quad (\text{donde } y \not\equiv x) \\
(M\, N)[R/x] & \equiv & M[R/x]\, N[R/x] \\
(\lambda x.M)[R/x] & \equiv & (\lambda x.M) \\
(\lambda y.M)[R/x] & \equiv & \begin{cases} (\lambda y.M[R/x]) & \text{si} \quad y \notin \mathrm{FV}(R) \\ (\lambda z.M[z/y])[R/x] & \text{si} \quad y \in \mathrm{FV}(R) \end{cases} \\
 & & (\text{donde } z \notin \mathrm{FV}(M) \cup \mathrm{FV}(R))
\end{array}$$

Reducción

El λ-cálculo incluye una relación binaria entre términos que representa su aspecto computacional u operacional. Esta relación, conocida como *conversión*, corresponde a la visión de los términos como expresiones que se transforman (reducen, convierten) en otras expresiones más "simples", hasta llegar a una forma que ya no puede reducirse, que representa el resultado.

Definición 2.1.4. La relación de conversión $\to$ del λ-cálculo se define en términos de las siguientes reglas:

$$\frac{}{(\lambda x.M)N \to M[N/x]}\ (\beta)$$

$$\frac{M \to M'}{M\, Z \to M'\, Z} \qquad \frac{M \to M'}{Z\, M \to Z\, M'} \qquad \frac{M \to M'}{\lambda x.M \to \lambda x.M'}$$

Notamos con $\to^*$ a la clausura reflexiva y transitiva de $\to$. Si tenemos $M \to N$ decimos que M *convierte* a N, mientras que si tenemos $M \to^* N$ decimos que M *reduce* a N.

Decimos que un término es una *forma normal β* cuando no puede convertirse a ningún otro término. Una *forma normal* de un término M es un término N tal que $M \to^* N$, siendo N una forma normal.

A su vez, en general se define un conjunto de *valores*, es decir, términos que representan resultados válidos de los cómputos. Los valores deben ser siempre formas normales. Si una reducción termina en un término que no es un valor, se dice que está *atascada*.

No todo término tiene forma normal. En particular, si consideramos el término $\Omega \equiv (\lambda x.x\,x)\,(\lambda x.x\,x)$, vemos que éste reduce a sí mismo, formando una cadena infinita de reducción:

$$\Omega \to \Omega \to \Omega \to \cdots$$

Como siempre se puede seguir realizando un paso de conversión por la regla (β), concluimos que Ω carece de forma normal. Cuando un término tiene al menos una forma normal, decimos que es *normalizante*.

En general, suele ocurrir que hay más de una secuencia de reducción posible partiendo de un término dado. En estos casos, es deseable que, independientemente del camino elegido, siempre se llegue a un mismo término. Esta propiedad, conocida como *confluencia* o *propiedad Church-Rosser*, se suele enunciar de la siguiente manera:

Definición 2.1.5. Una relación de conversión $\to$ se dice *confluente* (o *Church-Rosser*) si para todo término M tal que $M \to^* N_1$ y $M \to^* N_2$, existe un término N tal que $N_1 \to^* N \,^*\!\!\leftarrow N_2$.

Ejemplo 3. En λ-cálculo, podemos representar a los números naturales mediante lo que se conoce como *numerales de Church*. La idea es representar al cero y al sucesor mediante las siguientes definiciones:

$$c_0 = (\lambda f z.z), \qquad\qquad c_{n+1} = (\lambda f z.f\,(c_n\,f\,z)).$$

Cada numeral c_n codifica la operación de repetir la aplicación de una función n veces a un valor inicial, lo cual equivale al patrón de recursión estructural sobre los naturales. De esta manera, podemos identificar al número n con su numeral c_n correspondiente, y sin demasiado trabajo pueden definirse todas las operaciones usuales sobre naturales.

El λ-cálculo tipado

El λ-cálculo tipado surgió, de la mano de Haskell Curry y Alonzo Church [Chu40], como solución al problema de la inconsistencia del cálculo. Los *tipos* son entes sintácticos que pueden asignarse estáticamente a λ-términos para clasificarlos según sus propiedades y comportamiento. Si a un término M se le puede asignar el tipo T, decimos que "M tiene tipo T", que en símbolos se escribe $M : T$. Por ejemplo, en la mayoría de los cálculos, el término $\lambda x.x$ tiene tipo $T \to T$, ya que representa una función que toma algo de cualquier tipo (T) y devuelve algo del mismo tipo (la función identidad).

Una de las principales consecuencias de la introducción de los tipos en el cálculo es la *normalización*. Todos los términos de los principales cálculos tipados son normalizantes, a diferencia del λ-cálculo no tipado presentado anteriormente. Por razones de decidibilidad, esto necesariamente excluye del sistema a ciertos términos computables.

Ejemplo: λ-cálculo simplemente tipado

El λ-cálculo simplemente tipado es un cálculo que tiene exactamente la misma sintaxis de términos que la versión no tipada, y además define la siguiente sintaxis para tipos:

$$T ::= X \mid T \to T$$

El no-terminal X representa un conjunto infinito numerable de variables, llamadas variables de tipo. Un tipo T es o bien una variable de tipo, o bien una función entre dos tipos representada por $T \to T$, siguiendo la notación matemática usual.

También se cuenta con un *contexto de tipado*, usualmente denotado por la letra Γ, que representa una asociación entre variables de término y tipos. El entorno se da como un conjunto de asignaciones de la forma $x : T$. Su propósito es dar una asignación de tipos para las variables libres de un término.

Así, se define inductivamente una relación $\vdash$ entre contextos, términos y tipos, también conocida como *juicio de tipado*:

$$\frac{x : T \in \Gamma}{\Gamma \vdash x : T}\ \textsc{Var} \qquad \frac{\Gamma \vdash M : T \to R \quad \Gamma \vdash N : T}{\Gamma \vdash M\, N : R}\ \textsc{App} \qquad \frac{\Gamma, x : T \vdash M : R}{\Gamma \vdash (\lambda x.M) : T \to R}\ \textsc{Abs}$$

Si se puede demostrar $\Gamma \vdash M : T$, esto significa que en el contexto Γ, es posible asignarle el tipo T al término M. Por ejemplo, a partir de estas reglas se puede probar que en cualquier contexto Γ, se tiene $\Gamma \vdash (\lambda x.x) : X \to X$ para cualquier tipo X, lo cual coincide con nuestra intuición sobre el comportamiento de la función identidad.

La reducción en el λ-cálculo simplemente tipado sigue las mismas reglas que la versión no tipada (definición 2.1.4).

Existen distintas versiones tipadas del λ-cálculo, ofreciendo cada una un nivel de expresividad distinto. En la próxima sección, veremos el λ-cálculo polimórfico o λ-cálculo de segundo orden (también llamado System F o $\lambda 2$), introducido por Girard y Reynolds en forma independiente [Gir71, Rey74].

System F: el λ-cálculo polimórfico

En los sistemas de tipos, puede ocurrir que un mismo término pueda tener más de un tipo posible. En estos casos, se dice que se trata de un término *polimórfico*. Por ejemplo, suponiendo que T y R son dos tipos diferentes, el término $(\lambda x.x)$ tiene tipo $T \to T$ y $R \to R$, por lo que se dice que es un término polimórfico. En particular, los términos que trabajan exactamente de la misma manera sobre cualquier tipo poseen lo que se conoce como *polimorfismo paramétrico*.

System F [Gir71, Rey74] es la versión más conocida del λ-cálculo polimórfico. Partiendo del λ-cálculo simplemente tipado, y siguiendo con el ejemplo del término $(\lambda x.x)$, vemos que para todo tipo X podemos derivar $\Gamma \vdash (\lambda x.x) : X \to X$. La idea central de System F es poder expresar esta cuantificación universal en el propio sistema de tipos, en lugar de utilizar el metalenguage, mediante un *tipo universal* que se denota por $\forall X.X \to X$.

A continuación, presentaremos una extensión de System F, llamada System F con pares, que incluye construcciones para representar pares ordenados de elementos. Definimos la sintaxis de System F con pares como sigue:

$$\begin{array}{rcl} T & ::= & X \mid T \to T \mid \forall X.T \mid \star \mid T \times T \\ M & ::= & x \mid (M\, M) \mid (\lambda x : T.M) \mid (\Lambda X.M) \mid (M@T) \mid 1 \mid (M, M) \mid (\pi_1\, M) \mid (\pi_2\, M) \end{array}$$

El no-terminal T representa los tipos, mientras que M representa los términos. Una característica de esta presentación de System F es que posee *anotaciones de tipo* para las variables de las λ-abstracciones, es decir que se indica explícitamente qué tipo debe tener dicha variable en el mismo término. Este cálculo también presenta una forma de *abstracción sobre tipos*, $\Lambda X.M$, y una *aplicación sobre tipos*, $M@T$, que permiten expresar explícitamente el polimorfismo como términos que dependen de tipos. El término 1, también es conocido como "unit", representa la tupla vacía y su tipo es $\star$. El tipo $T \times R$ representa los pares de la forma (M, N) donde M y N tienen tipos T y R respectivamente. Los términos $\pi_i\, M$ para $i = 1, 2$ son *proyecciones*, es decir que representan la operación de extracción de la primera o segunda componente del par (según corresponda).

Se extiende la definición de sustitución sobre términos trivialmente a los tipos. Las reglas de tipado son

$$\frac{\Gamma \vdash M : T \quad X \notin \mathrm{FV}(T)}{\Gamma \vdash (\Lambda X.M) : \forall X.T}\ \forall\text{-I} \qquad \frac{\Gamma \vdash M : \forall X.T}{\Gamma \vdash M@R : T[R/X]}\ \forall\text{-E} \qquad \frac{x : T \in \Gamma}{\Gamma \vdash x : T}\ \textsc{Var}$$

$$\frac{\Gamma \vdash M : T \to R \quad \Gamma \vdash N : T}{\Gamma \vdash M\,N : R}\ \textsc{App} \qquad \frac{\Gamma, x : T \vdash M : R}{\Gamma \vdash (\lambda x : T.M) : T \to R}\ \textsc{Abs} \qquad \frac{\Gamma \vdash M : T \quad \Gamma \vdash N : R}{\Gamma \vdash (M, N) : T \times R}\ \textsc{Pair}$$

$$\frac{\Gamma \vdash M : T \times R}{\Gamma \vdash \pi_1\,M : T}\ \textsc{Prj-1} \qquad \frac{\Gamma \vdash M : T \times R}{\Gamma \vdash \pi_2\,M : R}\ \textsc{Prj-2} \qquad \frac{}{\Gamma \vdash 1 : \star}\ \textsc{Unit}$$

En System F, la función identidad se expresa

$$\Gamma \vdash \Lambda X.(\lambda x : X.x) : \forall X.X \to X.$$

En esta definición, vemos que se utiliza la abstracción de tipos para representar el hecho de que la función se puede utilizar con un tipo arbitrario. A la hora de usarla, es necesario *aplicar* este término al tipo que se desea. Más formalmente, para System F se agrega una nueva regla a la relación de conversión:

$$\frac{}{(\Lambda X.M)@T \to M[T/X]}$$

Así, un ejemplo de uso de la función identidad, si tomamos $y : A$, es $(\Lambda X.\lambda x : X.x)@A\,y \to (\lambda x : A.x)\,y \to y$, donde la estaríamos primero instanciando al tipo A, y luego pasándole el parámetro y.

También se agregan reglas para reducir las componentes de los pares y sus proyecciones:

$$\frac{M \to M'}{(M, N) \to (M', N)} \quad \frac{N \to N'}{(M, N) \to (M, N')}$$

$$\frac{}{\pi_1\,(M, N) \to M} \quad \frac{}{\pi_2\,(M, N) \to N}$$

Es interesante destacar que System F, a diferencia del λ-cálculo simplemente tipado, permite representar a los numerales de Church[1]: $c_n \equiv \Lambda X.\lambda f : X \to X.\lambda z : X.f^n z$. De esta manera tenemos

$$\Gamma \vdash c_n : \forall X.(X \to X) \to X \to X \qquad \text{para todo } n \geq 0$$

Hay dos propiedades de los cálculos tipados que son de gran importancia teórica y práctica. En primer lugar, enunciaremos la propiedad conocida como *subject reduction*, que dice que el tipado se preserva a través de la reducción:

Teorema (Subject reduction)**.** Si $\Gamma \vdash M : T$ y $M \to^* N$, entonces $\Gamma \vdash N : T$.

Esta propiedad, junto con el hecho de que los términos cerrados bien tipados nunca representan una reducción *atascada* (propiedad que usualmente se demuestra por separado), garantiza la *consistencia* del cálculo, es decir que las reducciones que parten de términos cerrados bien tipados no pueden atascarse.

En segundo lugar, tenemos la propiedad de *normalización fuerte*, que nos garantiza que las reducciones de términos bien tipados siempre terminan, y que además pueden realizarse en cualquier orden:

Teorema (Normalización fuerte)**.** Si $\Gamma \vdash M : T$, entonces M es fuertemente normalizante, es decir, que todas las secuencias de reducción que parten de M son finitas.

Observación. Nótese la diferencia entre normalizante y fuertemente normalizante. Si un término es normalizante, significa que existe al menos una reducción que lleva a una forma normal, aunque pueden

[1] Notamos con f^n al término que aplica n veces la función f a su argumento; es decir, $f^0 = (\lambda x.x)$, $f^{n+1} = (\lambda x.f\,(f^n\,x))$

existir otras secuencias de reducción que no lo hagan. Sin embargo, si un término es fuertemente normalizante, entonces siempre se llega a una forma normal, independientemente de qué orden de reducción se elija.

System F posee estas dos propiedades. Veremos un esbozo de una de las técnicas para demostrar normalización fuerte conocida como *candidatos de reducibilidad*, atribuida a Girard.

Esbozo de prueba. Un candidato de reducibilidad es un conjunto de términos que poseen ciertas propiedades, entre ellas, la de ser fuertemente normalizantes. La demostración se divide en dos partes: por un lado, se establece una correspondencia entre cada tipo de System F y determinado candidato de reducibilidad (definido en función del tipo en cuestión); por otro lado, se demuestra que cuando un término tiene tipo, entonces pertenece al candidato de reducibilidad asociado a dicho tipo. Finalmente, se tiene que si un término M está bien tipado, entonces pertenece a algún candidato de reducibilidad, y dado que estos conjuntos tienen la propiedad de normalización fuerte, entonces M debe ser fuertemente normalizante. $\square$

System F, gracias al polimorfismo, permite asignar tipo a más términos que el λ-cálculo simplemente tipado. En particular, puede demostrarse que todas las funciones recursivas estructurales (folds) totales sobre tipos de datos algebraicos regulares son expresables en System F, mediante la codificación de Church de los tipos algebraicos. Esto hace que las funciones que se utilizan en la práctica puedan generalmente ser expresadas en este cálculo. Sin embargo, como tiene normalización fuerte, necesariamente debe excluir algunos términos computables: en este caso, no pueden representarse las funciones totales cuya totalidad es indecidible en aritmética de segundo orden [GLT89].

2.2. Resumen

Hemos presentado el λ-cálculo no tipado, un modelo de cómputo que permite representar a todas las funciones computables. Vimos que la noción de *reducción* es fundamental para entender el aspecto operacional del cálculo. También presentamos variantes tipadas de λ-cálculo, que permiten clasificar los términos según los valores que calculan. Tomaremos System F con pares como punto de partida para el cálculo que presentaremos en el capítulo 4.

En el próximo capítulo, presentamos una extensión del λ-cálculo que da lugar a la familia de cálculos algebraicos.

Capítulo 3

Cálculos algebraicos

Como vimos en el capítulo anterior, el λ-cálculo es un módelo de cómputo general. Es posible definir extensiones al cálculo, con el objetivo de estudiar características particulares de lenguajes de programación.

En este capítulo presentamos la familia de cálculos algebraicos, que surgen de extender el λ-cálculo tradicional con construcciones algebraicas. Estos cálculos pueden considerarse trabajos previos al cálculo que es objeto de este trabajo, y se presentan aquí a fines de dar un marco adecuado para el capítulo 4.

En esencia, estos cálculos combinan λ-cálculo no tipado con álgebra lineal, permitiendo que se formen combinaciones lineales de λ-términos. Así, en un cálculo algebraico tenemos que si $\mathbf{t}$ y $\mathbf{r}$ son términos y α y β son escalares tomados de un anillo conmutativo, entonces $\alpha.\mathbf{t} + \beta.\mathbf{r}$ también es un término del cálculo.

Los cálculos λ_{lin} [AD08] y λ_{alg} [Vau09] son dos cálculos algebraicos no tipados. El primero, que tiene su origen en la comunidad de computación cuántica, es el que estudiaremos en detalle en la próxima sección. El segundo, proveniente de la comunidad de lógica lineal, posee una semántica diferente a la de λ_{lin} por lo que no lo trataremos. Ambos cálculos pueden simularse mutuamente [DCPTV10].

Luego de presentar λ_{lin}, veremos algunos sistemas de tipos que se han propuesto para este cálculo, así como sus ventajas y desventajas.

3.1. El cálculo λ_{lin}

El cálculo λ_{lin} [AD08] fue introducido por Pablo Arrighi y Gilles Dowek como un cálculo algebraico pensado para computación cuántica. Una combinación lineal de términos refleja el fenómeno de la *superposición*, es decir, la cualidad de un sistema cuántico de estar en más de un estado al mismo tiempo. Más generalmente, el cálculo λ_{lin} también puede utilizarse para representar cómputos no-determinísticos: el + denotaría una elección no-determinística entre dos términos. Por lo tanto, este cálculo se puede ver como un punto de partida para aplicaciones en computación paralela, computación probabilística, y computación cuántica, entre otros.

Desde el punto de vista operacional, se combinan las reglas de reducción usuales del λ-cálculo no tipado, con versiones orientadas de los axiomas de los espacios vectoriales, en el formato de sistema de reescritura de términos.

La figura 3.1 presenta el cálculo. Los términos se clasifican en términos base y términos generales; los términos base son los únicos que pueden sustituir a una variable en un paso de β-reducción. Los escalares pertenecen a un anillo conmutativo.

Esta estrategia, denominada *"call-by-*$\mathbf{b}$*"*[1], juega un papel importante en la interacción con la linealidad del álgebra lineal. Por ejemplo, el término $(\lambda x.x\ x)\ (y+z)$ puede reducirse a $(y+z)\ (y+z) \to^* y\ y+y\ z+z\ y+z\ z$ en un marco *call-by-name*. Sin embargo, si decidimos que las abstracciones deben comportarse como funciones lineales, podemos utilizar la estrategia *call-by-*$\mathbf{b}$, bajo la cual el término anterior se

[1] El conjunto de términos en $\mathbf{b}$ no es el conjunto de valores de λ_{lin}, por lo que no podemos hablar de *call-by-value*.

$$
\begin{array}{lll}
\textit{Términos} & \mathbf{t} ::= & \mathbf{b} \mid (\mathbf{t}\,\mathbf{t}) \mid \mathbf{0} \mid \alpha.\mathbf{t} \mid \mathbf{t}+\mathbf{r} \\
\textit{Términos base} & \mathbf{b} ::= & x \mid \lambda x.\mathbf{t}
\end{array}
$$

$$
\begin{array}{lll}
\textit{Reglas elementales} & \textit{Reglas de factorización} & \textit{Reglas de aplicación} \\
\mathbf{u}+\mathbf{0} \to \mathbf{u} & \alpha.\mathbf{u}+\beta.\mathbf{u} \to (\alpha+\beta).\mathbf{u}(*) & (\mathbf{u}+\mathbf{v})\,\mathbf{w} \to \mathbf{u}\,\mathbf{w}+\mathbf{v}\,\mathbf{w}(**) \\
0.\mathbf{u} \to \mathbf{0} & \alpha.\mathbf{u}+\mathbf{u} \to (\alpha+1).\mathbf{u}(*) & \mathbf{w}\,(\mathbf{u}+\mathbf{v}) \to \mathbf{w}\,\mathbf{u}+\mathbf{w}\,\mathbf{v}(**) \\
1.\mathbf{u} \to \mathbf{u} & \mathbf{u}+\mathbf{u} \to (1+1).\mathbf{u}(*) & (\alpha.\mathbf{u})\,\mathbf{v} \to \alpha.(\mathbf{u}\,\mathbf{v})(*) \\
\alpha.\mathbf{0} \to \mathbf{0} & & \mathbf{v}\,(\alpha.\mathbf{u}) \to \alpha.(\mathbf{v}\,\mathbf{u})(*) \\
\alpha.(\beta.\mathbf{u}) \to (\alpha\times\beta).\mathbf{u} & \beta\textit{-reducción:} & \mathbf{0}\,\mathbf{u} \to \mathbf{0} \\
\alpha.(\mathbf{u}+\mathbf{v}) \to \alpha.\mathbf{u}+\alpha.\mathbf{v} & (\lambda x.\mathbf{t})\,\mathbf{b} \to \mathbf{t}[\mathbf{b}/x](***) & \mathbf{u}\,\mathbf{0} \to \mathbf{0}
\end{array}
$$

Reglas contextuales: Sea $\mathbf{t} \to \mathbf{r}$, entonces

$$
\begin{array}{lll}
\mathbf{t}\,\mathbf{u} \to \mathbf{r}\,\mathbf{u} & \mathbf{t}+\mathbf{u} \to \mathbf{r}+\mathbf{u} & \alpha.\mathbf{t} \to \alpha.\mathbf{r} \\
\mathbf{u}\,\mathbf{t} \to \mathbf{u}\,\mathbf{r} & \mathbf{u}+\mathbf{t} \to \mathbf{u}+\mathbf{r} & \lambda x.\mathbf{t} \to \lambda x.\mathbf{r}
\end{array}
$$

Reescritura módulo asociatividad-conmutatividad de $+$; escalares pertenecen a un anillo conmutativo.
(*) Se aplican cuando $\mathbf{u}$ es un término cerrado normal.
(**) Se aplican cuando $\mathbf{t}+\mathbf{r}$ es un término cerrado normal.
(***) Se aplica sólo cuando $\mathbf{b}$ es un término base.

Figura 3.1: Cálculo λ_{lin}

reducirá a $(\lambda x.x\ x)\ y + (\lambda x.x\ x)\ z$ y luego a $y\ y + z\ z$. Esta elección es compatible con la computación cuántica [WZ82], la motivación original detrás de λ_{lin}.

Algunas reglas de reducción de λ_{lin} están sujetas a las restricciones (*), (**) y (***), ya que de otro modo el cálculo no sería confluente (ver definición 2.1.5). Por ejemplo, dado un término base $\mathbf{b}$, consideremos el término $Y_{\mathbf{b}} = (\lambda x.\mathbf{b} + x\,x)(\lambda x.\mathbf{b} + x\,x)$, de manera que $Y_{\mathbf{b}} \to \mathbf{b} + Y_{\mathbf{b}}$. Así, al reducir $Y_{\mathbf{b}}+(-1).Y_{\mathbf{b}}$ sin restricciones, podemos pasar a $(1-1).Y_{\mathbf{b}}$ por factorización, o bien pasar a $\mathbf{b}+Y_{\mathbf{b}}+(-1).Y_{\mathbf{b}}$ por β-reducción. Pero esto implica que podemos llegar a formas normales diferentes por cada uno de estos caminos: en el primer caso, tenemos $Y_{\mathbf{b}} + (-1).Y_{\mathbf{b}} \to^* 0$; en el segundo, tenemos $Y_{\mathbf{b}} + (-1).Y_{\mathbf{b}} \to \mathbf{b}+(Y_{\mathbf{b}}+(-1).Y_{\mathbf{b}}) \to^* \mathbf{b}+0 \to \mathbf{b}$. En λ_{lin}, esto se soluciona restringiendo la aplicación de la factorización, de módo que en este ejemplo sólo sea posible la β-reducción, dando lugar a una reducción infinita. Veremos que es posible eliminar estas restricciones, garantizando la confluencia del cálculo mediante un sistema de tipos.

A continuación, presentamos tres cálculos tipados basados en λ_{lin} con distintas características. Estos cálculos fueron los primeros intentos de dar tipos a un λ-cálculo algebraico.

3.2. Scalar

Scalar [ADC11] es un sistema de tipos para λ_{lin} que permite la multiplicación por escalares y la suma de términos con mismo tipo, pero no la suma de términos con tipos diferentes. Los tipos reflejan los escalares de los términos. El sistema está basado en System F, por lo que también incluye polimorfismo.

La figura 3.2 presenta el sistema de tipos de Scalar. Los términos y las reglas de reducción son exactamente las mismas que λ_{lin}, aunque veremos más adelante que las restricciones sobre la aplicación de las reglas no son necesarias en presencia de un sistema de tipos. Se busca que los escalares en los tipos representen a los escalares en los términos. La interpretación intuitiva es que un tipo de la forma $\alpha.T$ nos dice la "cantidad" de $\mathbf{t} : T$ en el término. Una característica de Scalar es que no permite sumar términos con tipos diferentes.

Distinguimos entre tipos *unitarios* (no-terminal U en la figura 3.2) y tipos generales. Los tipos unitarios

$$\begin{array}{lll} \textit{Tipos} & T ::= & U \mid \forall X.T \mid \alpha.T \mid \bar{0} \\ \textit{Tipos unitarios} & U ::= & X \mid U \to T \mid \forall X.U \end{array}$$

$$\frac{}{\Gamma, x : U \vdash x : U}\ \text{AX} \qquad \frac{}{\Gamma \vdash \mathbf{0} : \bar{0}}\ \text{AX}_{\bar{0}} \qquad \frac{\Gamma \vdash \mathbf{t} : \alpha.(U \to T) \qquad \Gamma \vdash \mathbf{r} : \beta.U}{\Gamma \vdash \mathbf{t}\,\mathbf{r} : \alpha.(\beta.T)}\ \to_{\text{E}}$$

$$\frac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x.\mathbf{t} : U \to T}\ \to_{\text{I}} \qquad \frac{\Gamma \vdash \mathbf{t} : \forall X.U}{\Gamma \vdash \mathbf{t} : U[V/X]}\ \forall_E \qquad \frac{\Gamma \vdash \mathbf{t} : U \qquad X \notin \text{FV}(\Gamma)}{\Gamma \vdash \mathbf{t} : \forall X.U}\ \forall_I$$

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}\ \text{sI} \qquad \frac{\Gamma \vdash \mathbf{t} : \alpha.T \qquad \Gamma \vdash \mathbf{r} : \beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : (\alpha + \beta).T}\ +\text{I} \qquad \frac{\Gamma \vdash \mathbf{t} : T \qquad T \equiv R}{\Gamma \vdash \mathbf{t} : R}\ \text{EQ}$$

Figura 3.2: Sistema de tipos de Scalar

no pueden incluir escalares excepto en el codominio de un tipo de función, y contienen todos los tipos de System F. Los tipos generales son, o bien tipos unitarios con escalares, o bien el tipo especial $\bar{0}$. Los términos base sólo pueden tener tipos unitarios. El juicio de tipado $\Gamma \vdash \mathbf{t} : T$ significa que al término $\mathbf{t}$ se le puede asignar el tipo T en el contexto Γ, con la definición usual de contexto de tipado para System F. Como consecuencia de la decisión de diseño de sólo permitir que términos básicos sustituyan variables en la β-conversión, los contextos de tipado ligan variables a tipos unitarios.

La regla de tipado $\to_E$ está motivada por la linealidad de la aplicación (ver Grupo Aplicativo en el conjunto de reglas de reescritura de λ_{lin}). Consideremos el tipado de un término de la forma $\mathbf{t}\,\mathbf{r}$. En primer lugar, es aceptable que $\mathbf{t}$ y $\mathbf{r}$ tengan escalares. Además, el término $(\alpha.\mathbf{u})\,\mathbf{r}$ reduce a $\alpha.(\mathbf{u}\,\mathbf{r})$. Simétricamente, $\mathbf{t}\,(\alpha.\mathbf{u})$ reduce a $\alpha.(\mathbf{t}\,\mathbf{u})$. Esto nos muestra que debemos permitir escalares en el tipo de una aplicación.

Ejemplo 4. Scalar se puede utilizar para representar un cálculo probabilístico. Usaremos a la suma como operador de alternativa, y a los escalares como “pesos” que indican la probabilidad de que el resultado sea igual al término que multiplican. Un ejemplo de esto sería el término

$$(\lambda xy.(1/4).x + (3/4).y)\,u\,v$$

Mediante el sistema de tipos, podemos verificar que si $u : A$ y $v : A$, la distribución de probabilidades del término es baricéntrica, es decir, que los escalares suman 1. Se puede derivar

$$u : A, v : A \vdash (\lambda xy.(1/4).x + (3/4).y)\,u\,v : A$$

y así comprobar que el término posee una distribución de probabilidades válida.

Scalar cuenta con las propiedades de *subject reduction* y normalización fuerte. La demostración de las mismas es una extensión de los métodos de demostración clásicos para System F. *Subject reduction* se demuestra por inducción en la estructura de la derivación de tipado, recurriendo en cada caso a un *generation lemma* que dice la forma que tiene un término con un tipo determinado. La normalización fuerte de Scalar se demuestra traduciéndolo a una variante de System F que “olvida” los escalares, y luego se utiliza la técnica de candidatos de reducibilidad de Girard para mostrar que dicho sistema tiene normalización fuerte. Las pruebas de estas propiedades pueden consultarse en [ADC11].

$$
\begin{array}{lrl}
\textit{Términos} & \mathbf{t} ::= & \mathbf{b} \mid (\mathbf{t}\,\mathbf{t}) \mid \mathbf{0} \mid \mathbf{t} + \mathbf{t} \\
\textit{Términos base} & \mathbf{b} ::= & x \mid \lambda x.\mathbf{t} \\
\\
\textit{Tipos} & T ::= & U \mid T + T \mid \bar{0} \\
\textit{Tipos unitarios} & U ::= & X \mid U \to T \mid \forall X.U
\end{array}
$$

$$
\frac{}{\Gamma, x : U \vdash x : U}\ \mathrm{AX} \qquad \frac{}{\Gamma \vdash \mathbf{0} : \bar{0}}\ \mathrm{AX}_{\bar{0}} \qquad \frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{\alpha}(U \to T_i) \qquad \Gamma \vdash \mathbf{r} : \sum_{j=1}^{\beta} U}{\Gamma \vdash \mathbf{t}\,\mathbf{r} : \sum_{i=1}^{\alpha}(\sum_{j=1}^{\beta} T_i)}\ \to_{\mathrm{E}}
$$

$$
\frac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x.\mathbf{t} : U \to T}\ \to_{\mathrm{I}} \qquad \frac{\Gamma \vdash \mathbf{t} : \forall X.U}{\Gamma \vdash \mathbf{t} : U[V/X]}\ \forall_E \qquad \frac{\Gamma \vdash \mathbf{t} : U \qquad X \notin \mathrm{FV}(\Gamma)}{\Gamma \vdash \mathbf{t} : \forall X.U}\ \forall_I
$$

$$
\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}\ +\mathrm{I} \qquad \frac{\Gamma \vdash \mathbf{t} : T \qquad T \equiv R}{\Gamma \vdash \mathbf{t} : R}\ \mathrm{E}_{\mathrm{Q}}
$$

Figura 3.3: Sintaxis y sistema de tipos de Additive

3.3. Additive

Additive [DCP11] es un sistema de tipos que considera el fragmento aditivo de λ_{lin}; es decir, permite la suma de términos pero no la multiplicación por escalares. Los tipos suma reflejan los distintos tipos presentes en un término suma. La reducción en Additive consta de las mismas reglas de λ_{lin}, excepto aquellas que involucran escalares, que no son necesarias. Additive también tiene *subject reduction*.

En la figura 3.3, presentamos el sistema de tipos de Additive, incluida la sintaxis para los términos y tipos y sus reglas de tipado. Se eliminan los escalares de la gramática y también se eliminan las reglas de reducción relacionadas con los escalares. Se usa la notación usual de sumatoria para representar sumas, de manera que $T_1 + T_2 + \cdots + T_\alpha = \sum_{i=1}^{\alpha} T_i$.

Para el tipado de la aplicación, se flexibiliza un poco la regla de Scalar para que en una aplicación de la forma $(\mathbf{u} + \mathbf{v})\,(\mathbf{t} + \mathbf{r})$ se permita que $\mathbf{u}$ y $\mathbf{v}$ tengan tipos distintos (los T_i de la regla), siempre que $\mathbf{t}$ y $\mathbf{r}$ tengan el mismo tipo.

Ejemplo 5. Sea $\Gamma \vdash \mathbf{b} : U$, $\Gamma \vdash (\lambda x.\mathbf{t}) : U \to T$ y $\Gamma \vdash (\lambda x.\mathbf{r}) : U \to R$, entonces

$$
\frac{\Gamma \vdash (\lambda x.\mathbf{t}) + (\lambda x.\mathbf{r}) : (U \to T) + (U \to R) \quad \Gamma \vdash \mathbf{b} : U}{\Gamma \vdash ((\lambda x.\mathbf{t}) + (\lambda x.\mathbf{r}))\,\mathbf{b} : T + R}\ \to_{\mathrm{E}}
$$

Podemos ver que $((\lambda x.\mathbf{t}) + (\lambda x.\mathbf{r}))\,\mathbf{b} \to^* (\lambda x.\mathbf{t})\,\mathbf{b} + (\lambda x.\mathbf{r})\,\mathbf{b}$.

3.4. Vectorial

El cálculo Vectorial [ADCV11] combina los dos enfoques anteriores, de manera que se pueda trabajar con el cálculo completo y al mismo tiempo se puedan utilizar sumas de términos con tipos distintos. Así, aparecen tanto escalares como sumas al nivel de los tipos, y los tipos de las combinaciones lineales de términos resultan ser combinaciones lineales de los tipos de los términos. Sin embargo, no se puede relacionar el cálculo con otra teoría conocida (como System F).

$$
\begin{array}{lll}
\textit{Tipos} & T ::= & U \mid T+T \mid \alpha.T \\
\textit{Tipos unitarios} & U ::= & X \mid U \to T \mid \forall X.U
\end{array}
$$

$$
\frac{}{\Gamma, x:U \vdash x:U}\ \text{AX} \qquad \frac{\Gamma \vdash \mathbf{t}:T}{\Gamma \vdash \mathbf{0}:0.T}\ \text{AX}_{\vec{0}} \qquad \frac{\Gamma, x:U \vdash \mathbf{t}:T}{\Gamma \vdash \lambda x.\mathbf{t}:U \to T}\ \to_{\text{I}}
$$

$$
\frac{\Gamma \vdash \mathbf{t}:\sum_{i=1}^{n}\alpha_i.\forall\vec{X}.(U \to T_i) \qquad \Gamma \vdash \mathbf{r}:\sum_{j=1}^{m}\beta_j.V_j \qquad \forall V_j, \exists\vec{W}_j, U[\vec{W}_j/\vec{X}] = V_j}{\Gamma \vdash \mathbf{t}\,\mathbf{r}:\sum_{i=1}^{n}\sum_{j=1}^{m}(\alpha_i \times \beta_j).T_i[\vec{W}_j/\vec{X}]}\ \to_{\text{E}}
$$

$$
\frac{\Gamma \vdash \mathbf{t}:\sum_{i=1}^{n}\alpha_i.\forall X.U_i}{\Gamma \vdash \mathbf{t}:\sum_{i=1}^{n}\alpha_i.U_i[V/X]}\ \forall_E \qquad \frac{\Gamma \vdash \mathbf{t}:\sum_{i=1}^{n}\alpha_i.U_i \qquad X \notin \text{FV}(\Gamma)}{\Gamma \vdash \mathbf{t}:\sum_{i=1}^{n}\alpha_i.\forall X.U_i}\ \forall_I
$$

$$
\frac{\Gamma \vdash \mathbf{t}:T}{\Gamma \vdash \alpha.\mathbf{t}:\alpha.T}\ \text{sI} \qquad \frac{\Gamma \vdash \mathbf{t}:T \qquad \Gamma \vdash \mathbf{r}:R}{\Gamma \vdash \mathbf{t}+\mathbf{r}:T+R}\ +\text{I} \qquad \frac{\Gamma \vdash \mathbf{t}:T \qquad T \equiv R}{\Gamma \vdash \mathbf{t}:R}\ \text{EQ}
$$

Figura 3.4: Tipos y reglas de tipado para Vectorial

En la figura 3.4 definimos los tipos y las reglas de tipado para Vectorial. Los términos de Vectorial son los de λ_{lin}. Las reglas de reescritura para la reducción son las mismas que para λ_{lin}, sólo que sin las restricciones en la aplicación de las reglas.

La regla de tipado de la aplicación es altamente compleja porque debe tener en cuenta las reglas de reescritura que distribuyen sumas y escalares sobre la aplicación. A diferencia de la regla para los cálculos anteriores, esta regla permite que, para una aplicación de la forma $\mathbf{t}\,\mathbf{r}$ donde $\mathbf{t}$ y $\mathbf{r}$ son sumas, los sumandos de *ambos* términos pueden tener tipos diferentes (siempre y cuando sean compatibles para la aplicación). Esto se logra utilizando el polimorfismo, por lo que la regla $\to_{\text{E}}$ hace varias cosas al mismo tiempo: no sólo elimina la flecha sino también la cuantificación universal.

Ejemplo 6. Sean $\mathbf{b_1}$ y $\mathbf{b_2}$ dos términos con tipos distintos U_1 y U_2, respectivamente. Quisiéramos que el término $(\lambda x.x)\,(\mathbf{b_1} + \mathbf{b_2})$ tuviera tipo $U_1 + U_2$, y esto se logra precisamente con la regla de tipado para la aplicación que presentamos, ya que se puede dar el tipo $\forall X.X \to X$ al término $(\lambda x.x)$. Con la regla de la aplicación de Scalar o de Additive, esto no hubiera sido posible ya que no permitiría que U_1 y U_2 fueran distintos.

Vectorial tiene *subject reduction* en un sentido más débil que el usual. En particular, si $\Gamma \vdash \mathbf{t} : T$ y $\mathbf{t} \to^* \mathbf{t}'$, no siempre se puede probar que $\Gamma \vdash \mathbf{t}' : T$, pero sí que $\Gamma \vdash \mathbf{t}' : R$, donde $R \sqsubseteq T$ para determinada relación de orden parcial $\sqsubseteq$ entre tipos, dependiente de los términos. Sin embargo, otra versión de Vectorial consigue *subject reduction* en el sentido usual [DC11, Capítulo 7]. Además, Vectorial tiene normalización fuerte.

3.5. Resumen

Hemos presentado una familia de cálculos algebraicos, haciendo hincapié en λ_{lin}, el cálculo algebraico inspirado en computación cuántica. A su vez, mostramos tres sistemas de tipos para λ_{lin}, cada uno con sus ventajas y desventajas.

En el próximo capítulo, presentamos la principal contribución del presente trabajo: un sistema de tipos alternativo para λ_{lin}, el cual puede verse como una versión de Additive extendida al cálculo completo, pero más simple que Vectorial.

Capítulo 4

El cálculo λCA

En este capítulo, presentamos el cálculo λCA (por *Complete Additive*), un sistema de tipos para el cálculo algebraico λ_{lin} que introdujimos en el capítulo anterior. En esencia, λCA es una generalización de Additive para que trabaje con el lenguaje completo con escalares restringidos a reales no negativos, en lugar de tomar sólo el fragmento aditivo. Mostraremos que λCA tiene propiedades de *subject reduction* y normalización fuerte. Finalmente, veremos que se puede demostrar la confluencia de todo el sistema a partir del teorema de normalización Fuerte.

Las demostraciones completas de los resultados presentados en este capítulo pueden encontrarse en el apéndice.

4.1. Presentación del cálculo

El cálculo λCA extiende System F con la capacidad de formar combinaciones lineales de términos. En la figura 4.1 mostramos la sintaxis abstracta de tipos y términos para el cálculo. Los términos están basados en λ_{lin}, mientras que los tipos están basados en Additive.

Se eligió System F explícito en lugar de una presentación implícita [ADC11, DCP11] porque, como se muestra en [ADCV11], las reglas de reducción de "factorización" (ver Grupo F en la figura 4.2) introducen imprecisiones en un marco de tipado completamente implícito.

Al igual que en λ_{lin}, los términos *básicos* (no-terminal **b** en la figura 4.1) son los únicos que pueden reemplazar a una variable en un paso de β-conversión.

De la misma forma que Scalar, distinguimos entre tipos *unitarios* (no-terminal U en la figura 4.1) y tipos generales. Los tipos unitarios no pueden incluir sumas de tipos excepto en el codominio de un tipo de función, y contienen todos los tipos de System F. Los tipos generales son o bien sumas de tipos unitarios o bien el tipo especial $\bar{0}$. Los términos básicos sólo pueden recibir tipos unitarios.

Una diferencia con Scalar es que restringimos los escalares a los números reales no negativos. Además, si bien se permiten escalares en los términos, no existen los escalares al nivel de los tipos, al igual que Additive. Introducimos la siguiente notación: para $n \geq 0$ entero, denotamos con $n.T$ al tipo $T+T+\cdots+T$ (n veces), donde $0.T = \bar{0}$. También nos permitimos utilizar el símbolo de sumatoria estándar, $\sum_{i=1}^{n} T_i$, donde $\sum_{i=1}^{0} T_i = \bar{0}$.

La figura 4.2 define el sistema de reescritura de términos para λCA, que consiste en versiones dirigidas de los axiomas de espacios vectoriales y la β-conversión para ambas clases de abstracciones. Todas las reducciones se llevan a cabo módulo asociatividad y conmutatividad del operador $+$. Se trata, esencialmente, del sistema de reescritura de λ_{lin}, con el agregado de la regla para reducir la aplicación de tipos. Como de costumbre, $\to^*$ denota la clausura reflexiva y transitiva de la relación de conversión $\to$.

La prueba de confluencia local para este sistema de reescritura ya fue realizada anteriormente para λ_{lin} [ADCV11] para un semianillo arbitrario. La confluencia surge, por el lema de Newman [Kah05, teorema 1.2.1], combinando la confluencia local y la normalización fuerte, que demostraremos en este capítulo.

Tipos	$T ::=$	$U \mid T+T \mid \bar{0}$	*Términos*	$\mathbf{t} ::=$	$\mathbf{b} \mid (\mathbf{t}\,\mathbf{t}) \mid \mathbf{t}@T \mid \mathbf{0} \mid \alpha.\mathbf{t} \mid \mathbf{t}+\mathbf{r}$
Unitarios	$U ::=$	$X \mid U \to T \mid \forall X.U$	*Términos básicos*	$\mathbf{b} ::=$	$x \mid \lambda x.\mathbf{t} \mid \Lambda X.\mathbf{t}$

Figura 4.1: Tipos y términos de λCA

Sistema de reescritura de términos:

Grupo E

$\mathbf{u}+\mathbf{0} \to \mathbf{u}$

$0.\mathbf{u} \to \mathbf{0}$

$1.\mathbf{u} \to \mathbf{u}$

$\alpha.\mathbf{0} \to \mathbf{0}$

$\alpha.(\beta.\mathbf{u}) \to (\alpha \times \beta).\mathbf{u}$

$\alpha.(\mathbf{u}+\mathbf{v}) \to \alpha.\mathbf{u}+\alpha.\mathbf{v}$

Grupo F

$\alpha.\mathbf{u}+\beta.\mathbf{u} \to (\alpha+\beta).\mathbf{u}$

$\alpha.\mathbf{u}+\mathbf{u} \to (\alpha+1).\mathbf{u}$

$\mathbf{u}+\mathbf{u} \to (1+1).\mathbf{u}$

Grupo A

$(\mathbf{u}+\mathbf{v})\,\mathbf{w} \to \mathbf{u}\,\mathbf{w}+\mathbf{v}\,\mathbf{w}$

$\mathbf{w}\,(\mathbf{u}+\mathbf{v}) \to \mathbf{w}\,\mathbf{u}+\mathbf{w}\,\mathbf{v}$

$(\alpha.\mathbf{u})\,\mathbf{v} \to \alpha.(\mathbf{u}\,\mathbf{v})$

$\mathbf{v}\,(\alpha.\mathbf{u}) \to \alpha.(\mathbf{v}\,\mathbf{u})$

$\mathbf{0}\,\mathbf{u} \to \mathbf{0}$

$\mathbf{u}\,\mathbf{0} \to \mathbf{0}$

Beta red.

$(\lambda x : U.\mathbf{t})\,\mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$

$(\Lambda X.\mathbf{t})@U \to \mathbf{t}[U/X]$

Reglas contextuales

$$\frac{\mathbf{t} \to \mathbf{t}'}{\mathbf{t}+\mathbf{r} \to \mathbf{t}'+\mathbf{r}} \qquad \frac{\mathbf{r} \to \mathbf{r}'}{\mathbf{t}+\mathbf{r} \to \mathbf{t}+\mathbf{r}'} \qquad \frac{\mathbf{t} \to \mathbf{t}'}{\alpha.\mathbf{t} \to \alpha.\mathbf{t}'}$$

$$\frac{\mathbf{t} \to \mathbf{t}'}{\mathbf{t}\,\mathbf{r} \to \mathbf{t}'\,\mathbf{r}} \qquad \frac{\mathbf{r} \to \mathbf{r}'}{\mathbf{t}\,\mathbf{r} \to \mathbf{t}\,\mathbf{r}'} \qquad \frac{\mathbf{t} \to \mathbf{t}'}{\mathbf{t}@T \to \mathbf{t}'@T}$$

$$\frac{\mathbf{t} \to \mathbf{t}'}{\lambda x : U.\mathbf{t} \to \lambda x : U.\mathbf{t}'} \qquad \frac{\mathbf{t} \to \mathbf{t}'}{\Lambda X.\mathbf{t} \to \Lambda X.\mathbf{t}'}$$

Figura 4.2: Relación de conversión $\to$

La figura 4.3 define la noción de equivalencia de tipos y muestra las reglas de tipado para el sistema. El juicio de tipado $\Gamma \vdash \mathbf{t} : T$ significa que al término $\mathbf{t}$ se le puede asignar el tipo T en el contexto Γ, con la definición usual de contexto de tipado para System F. Al igual que en los sistemas anteriores, los contextos de tipado ligan variables a tipos unitarios.

La regla de tipado $\to_E$ está motivada por la linealidad de la aplicación (ver Grupo A en el conjunto de reglas de reescritura). Consideremos el tipado de un término de la forma $\mathbf{t}\,\mathbf{r}$. En primer lugar, es aceptable que $\mathbf{t}$ y $\mathbf{r}$ sean sumas. Además, el término $(\mathbf{u}+\mathbf{v})\,\mathbf{r}$ reduce a $\mathbf{u}\,\mathbf{r}+\mathbf{v}\,\mathbf{r}$. Simétricamente, $\mathbf{t}\,(\mathbf{u}+\mathbf{v})$ reduce a $\mathbf{t}\,\mathbf{u}+\mathbf{t}\,\mathbf{v}$. Estos ejemplos nos muestran que debemos permitir sumas de tipos en el tipo de una aplicación.

Equivalencia de tipos La equivalencia es la menor congruencia $\equiv$ tal que

$$T + \bar{0} \equiv T, \quad T + R \equiv R + T, \quad T + (R + S) \equiv (T + R) + S$$

Reglas de tipado

$$\frac{}{\Gamma, x : U \vdash x : U}\ \text{AX} \qquad \frac{}{\Gamma \vdash \mathbf{0} : \bar{0}}\ \text{AX}_{\bar{0}}$$

$$\frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{\alpha}(U \to T_i) \qquad \Gamma \vdash \mathbf{r} : \beta.U}{\Gamma \vdash \mathbf{t}\,\mathbf{r} : \sum_{i=1}^{\alpha}(\beta.T_i)}\ \to_{\mathrm{E}} \qquad \frac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x : U.\mathbf{t} : U \to T}\ \to_{\mathrm{I}}$$

$$\frac{\Gamma \vdash \mathbf{t} : \forall X.U}{\Gamma \vdash \mathbf{t}@V : U[V/X]}\ \forall_E \qquad \frac{\Gamma \vdash \mathbf{t} : U \qquad X \notin \mathrm{FV}(\Gamma)}{\Gamma \vdash \Lambda X.\mathbf{t} : \forall X.U}\ \forall_I$$

$$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}\ +\text{I} \qquad \frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \lfloor \alpha \rfloor.T}\ \text{sI}$$

$$\frac{\Gamma \vdash \mathbf{t} : T \qquad T \equiv R}{\Gamma \vdash \mathbf{t} : R}\ \text{Eq}$$

Figura 4.3: Equivalencia de tipos y reglas de tipado para λCA

La principal innovación del cálculo es su tratamiento de los escalares (regla sI). Para evitar tener escalares al nivel de los tipos, a la hora de dar tipo a $\alpha.\mathbf{t}$, tomamos la parte entera inferior del escalar α, y asignamos al término el tipo $\lfloor \alpha \rfloor.T$, que es una suma de Ts. La interpretación intuitiva es que un tipo de la forma $n.T$ nos da una cota inferior para la "cantidad" de $\mathbf{t} : T$ en el término.

El cálculo λCA cuenta con una interpretación abstracta en Additive, que se basa en la idea de aproximar los escalares en los términos de la misma manera que en los tipos, es decir, mediante enteros. Así, los términos resultantes sólo contienen sumas, por lo que son términos válidos de Additive. Teniendo en cuenta que Additive es, a su vez, interpretable en System F con pares, podemos componer ambas interpretaciones y así relacionar λCA con una teoría conocida. Este resultado puede consultarse en [BDCJ11].

4.2. Propiedades

Subject reduction con cantidades imprecisas

Una propiedad de consistencia básica en un cálculo tipado es la garantía de que los tipos serán preservados por la reducción, también conocida como *subject reduction*. Sin embargo, los tipos de λCA son imprecisos acerca de la "cantidad" de cada tipo en el término. Por ejemplo, sea $\Gamma \vdash \mathbf{t} : T$ y consideremos el término $\mathbf{s} = 0,9.\mathbf{t} + 1,1.\mathbf{t}$. Es fácil ver que $\Gamma \vdash \mathbf{s} : T$ y $\mathbf{s} \to^* 2.\mathbf{t}$, pero $\Gamma \vdash 2.\mathbf{t} : T + T$. En este ejemplo, tenemos que un término con tipo T reduce a un término con tipo $T + T$, mostrando que la propiedad de *subject reduction* estricta no es válida para λCA. No obstante, podemos probar una propiedad similar: a medida que la reducción avanza, los tipos o bien se preservan o bien se *fortalecen*, es decir, se vuelven más precisos en la cantidad de acuerdo con una relación de orden. Esto implica que el tipo de un término

$$\frac{\alpha \leq \beta}{\alpha.T \preccurlyeq \beta.T}\ \textsc{Sub-Wk} \qquad \frac{T \equiv R}{T \preccurlyeq R}\ \textsc{Sub-Eq}$$

$$\frac{T \preccurlyeq S \qquad S \preccurlyeq R}{T \preccurlyeq R}\ \textsc{Sub-Tr} \qquad \frac{T_1 \preccurlyeq T_2 \qquad S_1 \preccurlyeq S_2}{T_1 + S_1 \preccurlyeq T_2 + S_2}\ \textsc{Sub-Ctxt}_1$$

$$\frac{U_2 \preccurlyeq U_1 \qquad T_1 \preccurlyeq T_2}{U_1 \to T_1 \preccurlyeq U_2 \to T_2}\ \textsc{Sub-Ctxt}_2 \qquad \frac{T \preccurlyeq R}{\forall X.T \preccurlyeq \forall X.R}\ \textsc{Sub-Ctxt}_3$$

Figura 4.4: Definición inductiva de la relación $\preccurlyeq$

es una cota inferior (con respecto a la relación de orden) para el tipo del término reducido.

Teorema 4.2.1 (Subject Reduction módulo $\preccurlyeq$)**.** Para todo término $\mathbf{t}$ y $\mathbf{t}'$, contexto Γ y tipo T, si $\mathbf{t} \to \mathbf{t}'$ y $\Gamma \vdash \mathbf{t} : T$, entonces existe algún tipo R tal que $\Gamma \vdash \mathbf{t}' : R$ y $T \preccurlyeq R$, donde la relación $\preccurlyeq$ está definida inductivamente en la figura 4.4.

Intuitivamente, $T \preccurlyeq R$ (R es al menos tan preciso como T) significa que hay más sumandos del mismo tipo en R que en T, por ejemplo $A \preccurlyeq A + A$ para un tipo fijo A.

La prueba de este teorema requiere varios lemas preliminares, llamados *lemas de generación*. Hay un lema de generación por cada construcción sintáctica del cálculo. En líneas generales, si un término $\mathbf{t}$ tiene tipo T, el lema de generación correspondiente establece ciertas condiciones que nos permiten determinar qué forma debe tener T, en función de la forma que tiene $\mathbf{t}$.

Lema 4.2.1 (Lemas de generación)**.** *Sea T un tipo y Γ un contexto de tipado.*

1. *Para términos arbitrarios $\mathbf{u}$ y $\mathbf{v}$, si $\Gamma \vdash \mathbf{u}\,\mathbf{v} : T$, entonces existen enteros no negativos α, β, y tipos $U \in \mathcal{U}, T_1, \dots, T_\alpha \in \mathcal{T}$, tales que $\Gamma \vdash \mathbf{u} : \sum_{i=1}^{\alpha}(U \to T_i)$ y $\Gamma \vdash \mathbf{v} : \beta.U$ con $\sum_{i=1}^{\alpha}(\beta.T_i) \equiv T$.*

2. *Para todo término $\mathbf{t}$ y tipo unitario U, si $\Gamma \vdash \lambda x : U.\mathbf{t} : T$, entonces existe un tipo R tal que $\Gamma, x : U \vdash \mathbf{t} : R$ y $U \to R \equiv T$.*

3. *Para términos arbitrarios $\mathbf{u}$ y $\mathbf{v}$, si $\Gamma \vdash \mathbf{u} + \mathbf{v} : T$, entonces existen tipos R y S tales que $\Gamma \vdash \mathbf{u} : R$ y $\Gamma \vdash \mathbf{v} : S$, con $R + S \equiv T$.*

4. *Para todo término $\mathbf{u}$ y número real no negativo α, si $\Gamma \vdash \alpha.\mathbf{u} : T$, entonces existe un tipo R tal que $\Gamma \vdash \mathbf{u} : R$ y $\lfloor \alpha \rfloor.R \equiv T$.*

5. *Para todo término $\mathbf{t}$, si $\Gamma \vdash \Lambda X.\mathbf{t} : T$, entonces existe un tipo R tal que $\Gamma \vdash \mathbf{t} : R$ y $\forall X.R \equiv T$ con $X \notin FV(\Gamma)$.*

6. *Para todo término $\mathbf{t}$ y tipo unitario U, si $\Gamma \vdash \mathbf{t}@U : T$, entonces existe un tipo V tal que $\Gamma \vdash \mathbf{t} : \forall X.V$ y $V[U/X] \equiv T$.*

El siguiente lema es estándar en las pruebas de *subject reduction* para sistemas similares a System F (ver [Kri90, props. 8.2 y 8.5] o [Bar92, prop. 4.1.19]). Garantiza que la tipabilidad se preserva bajo sustitución sobre variables de tipo y de término.

Lema 4.2.2 (Lema de sustitución)**.** *Para todo término* $\mathbf{t}$*, término básico* $\mathbf{b}$*, contexto* Γ*, tipo unitario* U *y tipo general* T*,*

1. *Si* $\Gamma \vdash \mathbf{t} : T$*, entonces* $\Gamma[U/X] \vdash \mathbf{t}[U/X] : T[U/X]$*.*

2. *Si* $\Gamma, x : U \vdash \mathbf{t} : T$ *y* $\Gamma \vdash \mathbf{b} : U$*, entonces* $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$*.*

Ahora podemos esbozar la prueba del teorema 4.2.1.

Demostración del teorema 4.2.1 (Subject Reduction módulo $\preccurlyeq$*).* Por inducción en la relación de conversión $\to$. Verificamos que cada regla de reducción preserva el tipo módulo la relación $\preccurlyeq$. En cada caso, primero aplicamos uno o más lemas de generación al lado izquierdo de la regla. Luego, construimos un tipo para el lado derecho que es o bien más preciso (de acuerdo con $\preccurlyeq$) o equivalente al del lado izquierdo.

A modo de ejemplo, mostramos la prueba para el caso correspondiente a la regla de reescritura $\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha + \beta).\mathbf{t}$. El resto de los casos pueden consultarse en el apéndice.

Debemos probar que para cualquier término $\mathbf{t}$, números reales no negativos α y β, contexto Γ y tipo T, si $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t} : T$ entonces $\Gamma \vdash (\alpha + \beta).\mathbf{t} : R$ con $T \preccurlyeq R$.

Por el lema 4.2.1.3, existen T_1, T_2 tales que $\Gamma \vdash \alpha.\mathbf{t} : T_1$ y $\Gamma \vdash \beta.\mathbf{t} : T_2$, con $T_1 + T_2 \equiv T$. Por el lema 4.2.1.4, existen R_1, R_2 tales que $\Gamma \vdash \mathbf{t} : R_1$ con $\lfloor\alpha\rfloor.R_1 \equiv T_1$, y $\Gamma \vdash \mathbf{t} : R_2$ con $\lfloor\beta\rfloor.R_2 \equiv T_2$. Entonces de $\Gamma \vdash \mathbf{t} : R_1$ podemos concluir que $\Gamma \vdash (\alpha + \beta).\mathbf{t} : \lfloor\alpha + \beta\rfloor.R_1$ usando la regla sI.

Ahora probaremos que $T \preccurlyeq \lfloor\alpha + \beta\rfloor.R_1$. Dado que R_1 y R_2 son ambos tipos válidos para $\mathbf{t}$, tenemos que $R_1 \equiv R_2$ entonces $\lfloor\alpha+\beta\rfloor.R_1 \succcurlyeq (\lfloor\alpha\rfloor + \lfloor\beta\rfloor).R_1 \equiv \lfloor\alpha\rfloor.R_1 + \lfloor\beta\rfloor.R_1 \equiv \lfloor\alpha\rfloor.R_1 + \lfloor\beta\rfloor.R_2 \equiv T_1 + T_2 \equiv T$. Por lo tanto, concluimos $T \preccurlyeq \lfloor\alpha + \beta\rfloor.R_1$. □

Normalización fuerte

En esta sección, probamos la propiedad de normalización fuerte para λCA, es decir, mostramos que todas las reducciones posibles para términos bien tipados son finitas. Usamos la noción estándar de *candidatos de reducibilidad* [GLT89, Capítulo 14], extendida para tener en cuenta las combinaciones lineales de términos. La prueba es similar a la esbozada en el capítulo 2 para System F. Nótese que no podemos reutilizar las pruebas de versiones tipadas de λ_{lin} anteriores (como Scalar y Additive) dado que en Scalar sólo los términos del mismo tipo pueden sumarse, y en Additive el cálculo bajo consideración es un fragmento de λCA. Esto significa que ninguno de ellos tiene el mismo conjunto de términos que λCA.

Un término de λCA es un *valor* si es una abstracción, una suma de valores, o un escalar multiplicado por un valor. En otras palabras, los valores se ajustan a la siguiente gramática:

$$\mathbf{v} ::= \lambda x\colon U.\mathbf{t} \mid \Lambda X.\mathbf{t} \mid \mathbf{v} + \mathbf{v} \mid \alpha.\mathbf{v}$$

Si un término no es un valor, decimos que es *neutral*.

Sea $\mathbf{t}$ un término normalizante, entonces denotamos a su forma normal con $\mathbf{t}\!\downarrow$. Definimos $\mathrm{Red}(\mathbf{t})$ como el conjunto de términos alcanzables en un paso de reducción desde $\mathbf{t}$, y $\mathrm{Red}_*(\mathbf{t})$ como el conjunto de términos alcanzables en cualquier número de pasos (incluyendo cero) desde $\mathbf{t}$.

Un término $\mathbf{t}$ es *fuertemente normalizante* si es normalizante y todas las secuencias de reducción maximales que empiezan en $\mathbf{t}$ son finitas y terminan en $\mathbf{t}\!\downarrow$. Notamos con SN_0 al conjunto de términos fuertemente normalizantes de λCA.

Definición 4.2.1 (Candidatos de reducibilidad)**.** Un conjunto de términos R es un *candidato de reducibilidad* si satisface las siguientes condiciones:

(CR$_1$) *Normalización fuerte*: $\mathsf{R} \subseteq \mathsf{SN}_0$

(CR$_2$) *Estabilidad bajo reducción*: Si $\mathbf{t} \in \mathsf{R}$ y $\mathbf{t} \to^* \mathbf{t}'$, entonces $\mathbf{t}' \in \mathsf{R}$.

(CR$_3$) *Estabilidad bajo expansión de neutrales*: Si $\mathbf{t}$ es neutral y $\mathrm{Red}(\mathbf{t}) \subseteq \mathsf{R}$, entonces $\mathbf{t} \in \mathsf{R}$.

Representamos con A, B a los candidatos de reducibilidad, y con RC al conjunto de todos los candidatos de reducibilidad.

La idea de la prueba de normalización fuerte es interpretar los tipos mediante candidatos de reducibilidad, por lo que si un término tiene tipo, entonces pertenece a un candidato de reducibilidad.

Observación 1. Nótese que $\mathsf{SN}_0 \in \mathsf{RC}$. Además, el término $\mathbf{0}$ es neutral y normal, por lo que está en todo candidato de reducibilidad, y en consecuencia $\emptyset \notin \mathsf{RC}$.

El siguiente lema garantiza que la propiedad de normalización fuerte se preserva por combinación lineal.

Lema 4.2.3. *Si $\mathbf{t}$ y $\mathbf{r}$ son fuertemente normalizantes, entonces $\alpha.\mathbf{t} + \beta.\mathbf{r}$ es fuertemente normalizante.*

Demostración. Inducción en una medida algebraica positiva definida sobre los términos de λ_{lin} [AD08, Proposición 10], mostrando que toda reducción algebraica hace decrecer estrictamente este número. Los detalles se encuentran en el apéndice A. □

Los siguientes operadores, que son cerrados en RC, garantizan que todos los tipos de λCA sean interpretados por un candidato de reducibilidad.

Definición 4.2.2 (Operadores en RC)**.** Sean A, B candidatos de reducibilidad.
Definimos los operadores $\rightarrow$, $\oplus$, Λ sobre RC y $\overline{\emptyset}$ tales que

- $\mathsf{A} \rightarrow \mathsf{B}$ es la clausura de $\{\mathbf{t} \mid \forall \mathbf{b} \in \mathsf{A},\ \mathbf{b}$ término básico $\Rightarrow (\mathbf{t})\,\mathbf{b} \in \mathsf{B}\}$ sobre (CR_3),
- $\mathsf{A} \oplus \mathsf{B}$ es la clausura de $\{\alpha.\mathbf{t} + \beta.\mathbf{r} \mid \mathbf{t} \in \mathsf{A}, \mathbf{r} \in \mathsf{B}\}$ sobre (CR_2) y (CR_3),
- $\Lambda\mathsf{A}$ es el conjunto $\{\mathbf{t} \mid \forall V, \mathbf{t}@V \in \mathsf{A}\}$
- $\overline{\emptyset}$ es la clausura de $\emptyset$ sobre (CR_3).

Lema 4.2.4. *Sean A y B candidatos de reducibilidad. Entonces, $\mathsf{A} \rightarrow \mathsf{B}$, $\mathsf{A} \oplus \mathsf{B}$, y $\Lambda\mathsf{A}$, $A \cap B$ y $\overline{\emptyset}$ son todos candidatos de reducibilidad.*

Demostración. Se verifican las condiciones de reduciblidad en cada caso. Ver sección A.2 en el apéndice. □

Ahora podemos introducir la función de interpretación para los tipos de λCA. La definición utiliza los operadores definidos más arriba.

Una *valuación* ρ es una función parcial de variables de tipos en candidatos de reducibilidad, escrita como una secuencia de asociaciones separadas por comas, de la forma $X \mapsto \mathsf{A}$, siendo $\emptyset$ la valuación vacía.

Definición 4.2.3 (Modelo de reducibilidad)**.** Sea T un tipo y ρ una valuación. Definimos la *interpretación* $[\![T]\!]_\rho$ como sigue:

$$\begin{array}{rcl}
[\![X]\!]_\rho & = & \rho(X) \\
[\![\bar{0}]\!]_\rho & = & \overline{\emptyset} \\
[\![U \rightarrow T]\!]_\rho & = & [\![U]\!]_\rho \rightarrow [\![T]\!]_\rho \\
[\![T + R]\!]_\rho & = & [\![T]\!]_\rho \oplus [\![R]\!]_\rho \\
[\![\forall X.U]\!]_\rho & = & \bigcap_{\mathsf{S} \in \mathsf{RC}} \Lambda[\![U]\!]_{\rho, X \mapsto S}
\end{array}$$

Nótese que el lema 4.2.4 garantiza que todos los tipos sean interpretados por un candidato de reducibilidad.

Una *sustitución* σ es una función parcial de variables de término en términos, escrita como una secuencia de asociaciones separadas por signos de punto y coma, de la forma $x \mapsto \mathbf{t}$, siendo $\emptyset$ la sustitución vacía. La acción de las sustituciones sobre los términos está dada por

$$\mathbf{t}_\emptyset = \mathbf{t}, \qquad \mathbf{t}_{x \mapsto \mathbf{r}; \sigma} = \mathbf{t}[\mathbf{r}/x]_\sigma.$$

Una *sustitución de tipos* δ es una función parcial de variables de tipo en tipos unitarios, escrita como una secuencia de asociaciones separadas por signos de punto y coma, de la forma $X \mapsto U$, siendo $\emptyset$ la sustitución vacía. La acción de las sustituciones de tipo sobre los tipos está dada por

$$T_{\emptyset} = T, \qquad\qquad T_{X \mapsto U;\delta} = T[U/X]_{\delta}$$

Se extienden para actuar sobre términos en la forma natural.

Sea Γ un contexto de tipado, entonces decimos que un par de sustituciones $\langle\sigma, \delta\rangle$ *satisface* Γ para una valuación ρ (notación $\langle\sigma, \delta\rangle \in [\![\Gamma]\!]_{\rho}$) si $(x : T) \in \Gamma$ implica $x_{\sigma} \in [\![U_{\delta}]\!]_{\rho}$.

Un juicio de tipado $\Gamma \vdash \mathbf{t} : T$ se dice *válido* (notación $\Gamma \vDash \mathbf{t} : T$) si para toda valuación ρ, toda sustitución de tipos δ y toda sustitución σ donde $\langle\sigma, \delta\rangle \in [\![\Gamma]\!]_{\rho}$, tenemos $(\mathbf{t}_{\delta})_{\sigma} \in [\![T]\!]_{\rho}$.

El siguiente lema demuestra que cualquier juicio de tipado derivable es válido.

Lema 4.2.5 (Lema de adecuación)**.** *Sea* $\Gamma \vdash \mathbf{t} : T$*, entonces* $\Gamma \vDash \mathbf{t} : T$*.*

Demostración. Por inducción en la estructura de $\Gamma \vdash \mathbf{t} : T$. Los detalles se encuentran en la sección A.2 en el apéndice. □

Dado que esto demuestra que todo término bien tipado pertenece a un candidato de reducibilidad, podemos probar que dichos términos son fuertemente normalizantes.

Teorema 4.2.2 (Normalización fuerte para λCA)**.** Todos los términos bien tipados de λCA son fuertemente normalizantes.

Demostración. Sea $\mathbf{t}$ un término de λCA con tipo T. Entonces, por el lema de adecuación (lema 4.2.5), sabemos que $\mathbf{t}_{\emptyset} \in [\![T]\!]_{\emptyset}$. Además, por el lema 4.2.4, sabemos que $[\![T]\!]_{\emptyset}$ es un candidato de reducibilidad, y entonces $[\![T]\!]_{\emptyset} \subseteq \mathsf{SN}_0$. En consecuencia, $\mathbf{t}$ es fuertemente normalizante. □

Finalmente, teniendo en cuenta que ya se ha demostrado la confluencia local para este sistema de reescritura [ADCV11], podemos combinar este resultado con el de normalización fuerte para garantizar la confluencia de λCA.

Teorema 4.2.3. El cálculo λCA es confluente.

Demostración. Por el lema de Newman [Kah05, teorema 1.2.1], que dice que un sistema de reescritura fuertemente normalizante y localmente confluente debe ser necesariamente confluente. □

4.3. Resumen

En este capítulo, presentamos el cálculo λCA, el aporte principal del trabajo. Este cálculo tiene un sistema de tipos consistente y fuertemente normalizante para todos los términos de λ_{lin}, que es menos complejo que Vectorial y a su vez más fácil de analizar, ya que cuenta con una interpretación abstracta en un sistema conocido como lo es System F con pares. Adicionalmente, este cálculo aporta confluencia al sistema de reescritura de λ_{lin} sin necesidad de restricciones; este resultado se basa en la normalización fuerte.

Capítulo 5

Conclusión

En esta tesina, hemos presentado y desarrollado el cálculo algebraico λCA, una extensión de System F explícitamente tipado que incluye combinaciones lineales de términos con escalares reales positivos. Este cálculo puede verse como un sistema de tipos para el cálculo algebraico no tipado λ_{lin} que extiende las nociones del cálculo Additive al cálculo completo. Se ha demostrado la consistencia, la normalización fuerte y la confluencia de λCA. Para la consistencia, la propiedad de *subject reduction* que se puede demostrar resulta exacta en cuanto a los tipos, pero imprecisa en cuanto a la "cantidad" de cada tipo en un término: si un término tiene un tipo que es la suma de varios tipos, entonces a través de la reducción esta cantidad puede incrementarse pero nunca decrementarse.

Este cálculo encaja en una sucesión de cálculos algebraicos tipados preexistentes (Scalar, Additive, y Vectorial). Sin embargo, Scalar sólo permite sumas de términos con el mismo tipo, Additive sólo trabaja con el fragmento aditivo, y Vectorial tiene una teoría muy compleja. Así, el cálculo λCA nos da un sistema de tipos para todo λ_{lin} que a la vez tiene una teoría más simple que Vectorial, a cambio de perder precisión sobre los escalares en los tipos, y puede ser interpretado abstractamente en Additive. El sistema también nos da cierta información sobre los términos del cálculo; en particular, nos provee de una cota inferior para los escalares presentes en los términos de una combinación lineal.

A su vez, vimos que el cálculo λ_{lin} original tenía, en su mecanismo de reducción, varias restricciones para garantizar la confluencia del sistema de reescritura; en λCA, estas restricciones a la aplicación de las reglas pueden eliminarse, dando lugar a un sistema de reescritura más simple, ya que la confluencia se prueba a partir de la normalización fuerte.

Al tratarse de un sistema derivado de System F, se tiene que necesariamente existen algunas funciones totales computables que no pueden representarse en λCA (y sí en λ_{lin}). Esto no es un problema porque System F ha demostrado ser suficientemente expresivo en la práctica, y los beneficios de contar con una teoría simple superan a las limitaciones de tener un lenguaje menos expresivo.

El estudio de λCA constituye un aporte teórico para los cálculos algebraicos tipados, sirviendo como punto de partida para futuras extensiones con aplicaciones particulares. Las demostraciones realizadas podrían adaptarse para probar resultados análogos en otros cálculos algebraicos.

Pueden identificarse algunas extensiones a λCA como trabajo futuro, particularmente en lo que refiere al tratamiento de los escalares. En este trabajo tomamos las cotas inferiores como aproximación de los escalares, pero esta decisión es arbitraria; se podrían haber tomado las cotas superiores, dando lugar a otro sistema. Una posible extensión consiste en tomar tanto cotas superiores como inferiores, y acarrear ambas para representar un intervalo, dando lugar a una mayor precisión en la aproximación. Se deja como problema abierto el tratamiento de escalares pertenecientes a un anillo arbitrario.

Apéndice A

Demostraciones

A.1. Subject reduction

Lemas preliminares

Lema 4.2.1 (Lemas de generación)**.** *Sea T un tipo y Γ un contexto de tipado.*

1. *Para términos arbitrarios $\mathbf{u}$ y $\mathbf{v}$, si $\Gamma \vdash \mathbf{u}\,\mathbf{v} : T$, entonces existen enteros no negativos α, β, y tipos $U \in \mathcal{U}, T_1, \ldots, T_\alpha \in \mathcal{T}$, tales que $\Gamma \vdash \mathbf{u} : \sum_{i=1}^{\alpha}(U \to T_i)$ y $\Gamma \vdash \mathbf{v} : \beta.U$ con $\sum_{i=1}^{\alpha}(\beta.T_i) \equiv T$.*

2. *Para todo término $\mathbf{t}$ y tipo unitario U, si $\Gamma \vdash \lambda x : U.\mathbf{t} : T$, entonces existe un tipo R tal que $\Gamma, x : U \vdash \mathbf{t} : R$ y $U \to R \equiv T$.*

3. *Para términos arbitrarios $\mathbf{u}$ y $\mathbf{v}$, si $\Gamma \vdash \mathbf{u} + \mathbf{v} : T$, entonces existen tipos R y S tales que $\Gamma \vdash \mathbf{u} : R$ y $\Gamma \vdash \mathbf{v} : S$, con $R + S \equiv T$.*

4. *Para todo término $\mathbf{u}$ y número real no negativo α, si $\Gamma \vdash \alpha.\mathbf{u} : T$, entonces existe un tipo R tal que $\Gamma \vdash \mathbf{u} : R$ y $\lfloor \alpha \rfloor.R \equiv T$.*

5. *Para todo término $\mathbf{t}$, si $\Gamma \vdash \Lambda X.\mathbf{t} : T$, entonces existe un tipo R tal que $\Gamma \vdash \mathbf{t} : R$ y $\forall X.R \equiv T$ con $X \notin FV(\Gamma)$.*

6. *Para todo término $\mathbf{t}$ y tipo unitario U, si $\Gamma \vdash \mathbf{t}@U : T$, entonces existe un tipo V tal que $\Gamma \vdash \mathbf{t} : \forall X.V$ y $V[U/X] \equiv T$.*

Demostración. Todas las demostraciones son por inducción en la longitud de la derivación de tipado. Sólo existen dos casos para cada propiedad: el caso trivial y el de la regla de equivalencia (Eq); mostramos las demostraciones correspondientes a este último caso.

1. $$\dfrac{\Gamma \vdash (\mathbf{u})\,\mathbf{v} : T \qquad T \equiv R}{\Gamma \vdash (\mathbf{u})\,\mathbf{v} : R}\ \textsc{Eq}$$

   Por hipótesis indutiva existen α, β y $W, T_1, \ldots, T_\alpha$ tales que $\Gamma \vdash \mathbf{u} : \sum_{i=1}^{\alpha}(W \to T_i)$, $\Gamma \vdash \mathbf{v} : \beta.W$ y $\sum_{i=1}^{\alpha} \beta.T_i \equiv T$. Dado que $\sum_{i=1}^{\alpha} \beta.T_i \equiv T$ y $T \equiv R$, tenemos $\sum_{i=1}^{\alpha} \beta.T_i \equiv R$.

2. $$\dfrac{\Gamma \vdash \lambda x : U.\mathbf{t} : T \qquad T \equiv R}{\Gamma \vdash \lambda x : U.\mathbf{t} : R}\ \textsc{Eq}$$

   Por hipótesis inductiva existe S tal que $\Gamma, x : U \vdash \mathbf{t} : S$ y $U \to S \equiv T$. Dado que $T \equiv R$, tenemos $U \to S \equiv R$.

3. $$\frac{\Gamma \vdash \mathbf{u}+\mathbf{v} : T \qquad T \equiv R}{\Gamma \vdash \mathbf{u}+\mathbf{v} : R}\ \text{Eq}$$

Por la hipótesis inductiva existen tipos S_1, S_2 tales que $\Gamma \vdash \mathbf{u} : S_1$, $\Gamma \vdash \mathbf{v} : S_2$ y $S_1 + S_2 \equiv T$ Entonces, dado que $T \equiv R$, tenemos $S_1 + S_2 \equiv R$.

4. $$\frac{\Gamma \vdash \alpha.\mathbf{u} : T \qquad T \equiv R}{\Gamma \vdash \alpha.\mathbf{u} : R}\ \text{Eq}$$

Por la hipótesis inductiva, existe un tipo S tal que $\Gamma \vdash \mathbf{u} : S$ y $\lfloor\alpha\rfloor.S \equiv T$. Dado que $T \equiv R$, tenemos $\lfloor\alpha\rfloor.S \equiv R$.

5. $$\frac{\Gamma \vdash \Lambda X.\mathbf{t} : T \qquad T \equiv R}{\Gamma \vdash \Lambda X.\mathbf{t} : R}\ \text{Eq}$$

Por la hipótesis inductiva existe S tal que $\Gamma \vdash \mathbf{t} : S$ y $\forall X.S \equiv T$. Dado que $T \equiv R$, tenemos $\forall X.S \equiv R$.

6. $$\frac{\Gamma \vdash \mathbf{t}@U : T \qquad T \equiv R}{\Gamma \vdash \mathbf{t}@U : R}\ \text{Eq}$$

Por la hipótesis inductiva existe V tal que $\Gamma \vdash \mathbf{t} : \forall X.V$ y $V[U/X] \equiv T$. Dado que $V[U/X] \equiv T$ y $T \equiv R$, tenemos $V[U/X] \equiv R$.

□

Corolario A.1.1. *Para todo contexto Γ, tipos U, T y término $\mathbf{t}$, se tiene que si $\Gamma \vdash \lambda x : U.\mathbf{t} : U \to T$, entonces $\Gamma, x : U \vdash \mathbf{t} : T$.*

Demostración. Por el lema 2, existe un R tal que $\Gamma, x : U \vdash \mathbf{t} : R$ y $U \to R \equiv U \to T$. Esto implica que $T \equiv R$, por lo que concluimos que $\Gamma, x : U \vdash \mathbf{t} : T$. □

Lema 4.2.2 (Lema de sustitución)**.** *Para todo término $\mathbf{t}$, término básico $\mathbf{b}$, contexto Γ, tipo unitario U y tipo general T,*

1. *Si $\Gamma \vdash \mathbf{t} : T$, entonces $\Gamma[U/X] \vdash \mathbf{t}[U/X] : T[U/X]$.*
2. *Si $\Gamma, x : U \vdash \mathbf{t} : T$ y $\Gamma \vdash \mathbf{b} : U$, entonces $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$.*

Demostración. 1. Sea $S_n = \Gamma \vdash \mathbf{u} : T$. Procedemos por inducción en n.

Casos base. $n = 0$

a) $$\frac{}{\Gamma, x : V \vdash x : V}\ \text{Ax}$$

Nótese que $(\Gamma, x : V)[U/X] \equiv \Gamma[U/X], x : V[U/X]$, entonces por la regla Ax, tenemos $(\Gamma, x : V)[U/X] \vdash x[U/X] : V[U/X]$.

b) $$\frac{}{\Gamma \vdash \mathbf{0} : \bar{0}}\ \text{Ax}_{\bar{0}}$$

Nótese que $\bar{0} \equiv \bar{0}[U/X]$ y $\mathbf{0}[U/X] = \mathbf{0}$, entonces por la regla $\text{Ax}_{\bar{0}}$ tenemos $\Gamma[U/X] \vdash \mathbf{0}[U/X] : \bar{0}[U/X]$.

Casos inductivos.

a) $$\dfrac{\Gamma \vdash \mathbf{u} : \sum_{i=1}^{\alpha}(V \to T_i) \qquad \Gamma \vdash \mathbf{v} : \beta.V}{\Gamma \vdash (\mathbf{u})\,\mathbf{v} : \sum_{i=1}^{\alpha}(\beta.T_i)} \to_{\mathrm{E}}$$

Por la hipótesis inductiva, $\Gamma[U/X] \vdash \mathbf{u}[U/X] : (\sum_{i=1}^{\alpha}(V \to T_i))[U/X]$. Sin embargo, nótese que $(\sum_{i=1}^{\alpha}(V \to T_i))[U/X] \equiv \sum_{i=1}^{\alpha} V[U/X] \to T_i[U/X]$.

Además, $\Gamma[U/X] \vdash \mathbf{v}[U/X] : (\beta.V)[U/X]$, pero $(\beta.V)[U/X] \equiv \beta.V[U/X]$, so

$$\dfrac{\Gamma[U/X] \vdash \mathbf{u}[U/X] : \sum_{i=1}^{\alpha} V[U/X] \to T_i[U/X] \qquad \Gamma[U/X] \vdash \mathbf{v}[U/X] : \beta.V[U/X]}{\Gamma[U/X] \vdash (\mathbf{u}[U/X])\,\mathbf{v}[U/X] : \sum_{i=1}^{\alpha} \beta.T_i[U/X]} \to_{\mathrm{E}}$$

Nótese que $(\mathbf{u}[U/X])\,\mathbf{v}[U/X] = ((\mathbf{u})\,\mathbf{v})[U/X]$ y $\sum_{i=1}^{\alpha} \beta.T_i[U/X] \equiv (\sum_{i=1}^{\alpha}(\beta.T_i))[U/X]$.

b) $$\dfrac{\Gamma, x : V \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x : V.\mathbf{t} : V \to T} \to_{\mathrm{I}}$$

Por la hipótesis inductiva $(\Gamma, x : V)[U/X] \vdash \mathbf{t}[U/X] : T[U/X]$. Nótese que $(\Gamma, x : V)[U/X] \equiv \Gamma[U/X], x : V[U/X]$, entonces

$$\dfrac{\Gamma[U/X], x : V[U/X] \vdash \mathbf{t}[U/X] : T[U/X]}{\Gamma[U/X] \vdash \lambda x : V[U/X].\mathbf{t}[U/X] : V[U/X] \to T[U/X]} \to_I$$

Nótese que $\lambda x : V[U/X].\mathbf{t}[U/X] = (\lambda x : V.\mathbf{t})[U/X]$ y $V[U/X] \to T[U/X] \equiv (V \to T)[U/X]$.

c) $$\dfrac{\Gamma \vdash \mathbf{t} : \forall Y.V_1}{\Gamma \vdash \mathbf{t}@V_2 : V_1[V_2/Y]} \forall E$$

Por la hipótesis inductiva $\Gamma[U/X] \vdash \mathbf{t}[U/X] : (\forall Y.V_1)[U/X] \equiv \forall Y.V_1[U/X]$, donde $X \neq Y$ y $Y \notin \mathrm{FV}(U)$. Entonces,

$$\dfrac{\Gamma[U/X] \vdash \mathbf{t}[U/X] : \forall Y.V_1[U/X]}{\Gamma[U/X] \vdash \mathbf{t}[U/X]@(V_2[U/X]) : V_1[U/X][V_2[U/X]/Y]} \forall_E$$

Nótese que $\mathbf{t}[U/X]@(V_2[U/X]) = (\mathbf{t}@V_2)[U/X]$ y $V_1[U/X][V_2[U/X]/Y] \equiv V_1[V_2/Y][U/X]$ módulo renombre de variables.

d) $$\dfrac{\Gamma \vdash \mathbf{t} : V \qquad Y \notin \mathrm{FV}(\Gamma)}{\Gamma \vdash \Lambda Y.\mathbf{t} : \forall Y.V} \forall I$$

Por la hipótesis inductiva, $\Gamma[U/X] \vdash \mathbf{t}[U/X] : V[U/X]$. Suponiendo que $Y \notin \mathrm{FV}(U)$ (por la convención de Barendregt), tenemos

$$\dfrac{\Gamma[U/X] \vdash \mathbf{t}[U/X] : V[U/X] \qquad Y \notin \mathrm{FV}(\Gamma[U/X])}{\Gamma[U/X] \vdash \Lambda Y.(\mathbf{t}[U/X]) : \forall Y.(V[U/X])} \forall I$$

Nótese que $\Lambda Y.(\mathbf{t}[U/X]) = (\Lambda Y.\mathbf{t})[U/X]$ y $\forall Y.(V[U/X]) \equiv (\forall Y.V)[U/X]$ módulo renombre de variables.

e) $\dfrac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}$

Por la hipótesis inductiva, $\Gamma[U/X] \vdash \mathbf{t}[U/X] : T[U/X]$ y $\Gamma[U/X] \vdash \mathbf{r}[U/X] : R[U/X]$. Entonces,

$$\frac{\Gamma[U/X] \vdash \mathbf{t}[U/X] : T[U/X] \qquad \Gamma[U/X] \vdash \mathbf{r}[U/X] : R[U/X]}{\Gamma[U/X] \vdash \mathbf{t}[U/X] + \mathbf{r}[U/X] : T[U/X] + R[U/X]} +_I$$

Nótese que $\mathbf{t}[U/X] + \mathbf{r}[U/X] = (\mathbf{t} + \mathbf{r})[U/X]$ y $T[U/X] + R[U/X] \equiv (T + R)[U/X]$.

f) $\dfrac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \lfloor \alpha \rfloor.T}$ sI

Por la hipótesis inductiva, $\Gamma[U/X] \vdash \mathbf{t}[U/X] : T[U/X]$. Entonces,

$$\frac{\Gamma[U/X] \vdash \mathbf{t}[U/X] : T[U/X]}{\Gamma[U/X] \vdash \alpha.\mathbf{t}[U/X] : \lfloor \alpha \rfloor.T[U/X]} \text{ sI}$$

Nótese que $\alpha.\mathbf{t}[U/X] = (\alpha.\mathbf{t})[U/X]$ y $\lfloor \alpha \rfloor.T[U/X] \equiv (\lfloor \alpha \rfloor.T)[U/X]$.

g) $\dfrac{\Gamma \vdash \mathbf{t} : T \qquad T \equiv R}{\Gamma \vdash \mathbf{t} : R}$ Eq

Por la hipótesis inductiva, $\Gamma[U/X] \vdash \mathbf{t}[U/X] : T[U/X]$. Entonces

$$\frac{\Gamma[U/X] \vdash \mathbf{t}[U/X] : T[U/X] \qquad T[U/X] \equiv R[U/X]}{\Gamma[U/X] \vdash \mathbf{t}[U/X] : R[U/X]} \text{ Eq}$$

2. Sea $S_n = \Gamma, x : U \vdash \mathbf{t} : T$ y $\Gamma \vdash \mathbf{b} : U$. Procedemos por inducción sobre n.

Casos base. $n = 1$.

a) $\dfrac{}{\Gamma, x : U \vdash x : U}$ Ax Nótese que $x[\mathbf{b}/x] = \mathbf{b}$, entonces $\Gamma \vdash \mathbf{b} : U$.

b) $\dfrac{}{\Gamma, x : U, y : V \vdash y : V}$ Ax Nótese que $y[\mathbf{b}/x] = y$, entonces $\Gamma, y : V \vdash y[\mathbf{b}/x] : V$.

c) $\dfrac{}{\Gamma, x : U \vdash \mathbf{0} : \bar{0}}$ $\text{Ax}_{\bar{0}}$ Nótese que $\mathbf{0}[\mathbf{b}/x] = \mathbf{0}$, entonces $\Gamma \vdash \mathbf{0}[\mathbf{b}/x] : \bar{0}$.

Casos inductivos.

a) $$\frac{\Gamma, x : U \vdash \mathbf{u} : \sum_{i=1}^{\alpha} V \to T_i \qquad \Gamma, x : U \vdash \mathbf{v} : \beta.V}{\Gamma, x : U \vdash (\mathbf{u})\, \mathbf{v} : \sum_{i=1}^{\alpha} \beta.T_i} \to_E$$

Por la hipótesis inductiva $\Gamma \vdash \mathbf{u}[\mathbf{b}/x] : \sum_{i=1}^{\alpha} V \to T_i$ y $\Gamma \vdash \mathbf{v}[\mathbf{b}/x] : \beta.V$, entonces usando la regla $\to_E$ tenemos $\Gamma \vdash (\mathbf{u}[\mathbf{b}/x])\, \mathbf{v}[\mathbf{b}/x] : \sum_{i=1}^{\alpha} \beta.T_i$. Nótese que $(\mathbf{u}[\mathbf{b}/x])\, \mathbf{v}[\mathbf{b}/x] = ((\mathbf{u})\, \mathbf{v})[\mathbf{b}/x]$.

b) $\dfrac{\Gamma, y : V, x : U \vdash \mathbf{t} : T}{\Gamma, x : U \vdash \lambda y : V.\mathbf{t} : V \to T} \to_I$

Por hipótesis inductiva, $\Gamma, y : V \vdash \mathbf{t}[\mathbf{b}/x] : T$. Entonces,

$$\frac{\Gamma, y : V \vdash \mathbf{t}[\mathbf{b}/x] : T}{\Gamma \vdash \lambda y : V.\mathbf{t}[\mathbf{b}/x] : V \to T} \to_I$$

Nótese que $\lambda y : V.\mathbf{t}[\mathbf{b}/x] = (\lambda y : V.\mathbf{t})[\mathbf{b}/x]$.

c) $$\dfrac{\Gamma, x : U \vdash \mathbf{u} : \forall X.U}{\Gamma, x : U \vdash \mathbf{u}@V : U[V/X]}\ \forall_E$$

Por la hipótesis inductiva, $\Gamma \vdash \mathbf{u}[\mathbf{b}/x] : \forall X.U$, entonces

$$\dfrac{\Gamma \vdash \mathbf{u}[\mathbf{b}/x] : \forall X.U}{\Gamma \vdash \mathbf{u}[\mathbf{b}/x]@V : U[V/X]}\ \forall_E$$

Nótese que $\mathbf{u}[\mathbf{b}/x]@V = (\mathbf{u}@V)[\mathbf{b}/x]$.

d) $$\dfrac{\Gamma, x : U \vdash \mathbf{u} : U \qquad X \notin \mathrm{FV}(\Gamma)}{\Gamma, x : U \vdash \Lambda X.\mathbf{u} : \forall X.U}\ \forall_I$$

Por la hipótesis inductiva $\Gamma \vdash \mathbf{u}[\mathbf{b}/x] : U$, entonces por la regla $\forall_I$, tenemos $\Gamma \vdash \Lambda X.\mathbf{u}[\mathbf{b}/x] : \forall X.U$. Nótese que $\Lambda X.\mathbf{u}[\mathbf{b}/x] = (\Lambda X.\mathbf{u})[\mathbf{b}/x]$.

e) $$\dfrac{\Gamma, x : U \vdash \mathbf{u} : T \qquad \Gamma, x : U \vdash \mathbf{v} : R}{\Gamma, x : U \vdash \mathbf{u} + \mathbf{v} : T + R}\ +_I$$

Por la hipótesis inductiva $\Gamma \vdash \mathbf{u}[\mathbf{b}/x] : T$ y $\Gamma \vdash \mathbf{v}[\mathbf{b}/x] : R$, entonces por la regla $+_I$ tenemos $\Gamma \vdash \mathbf{u}[\mathbf{b}/x] + \mathbf{v}[\mathbf{b}/x] : T + R$. Nótese que $\mathbf{u}[\mathbf{b}/x] + \mathbf{v}[\mathbf{b}/x] = (\mathbf{u} + \mathbf{v})[\mathbf{b}/x]$.

f) $$\dfrac{\Gamma, x : U \vdash \mathbf{u} : T}{\Gamma, x : U \vdash \alpha.\mathbf{u} : \lfloor\alpha\rfloor.T}\ \mathrm{sI}$$

Por la hipótesis inductiva $\Gamma \vdash \mathbf{u}[\mathbf{b}/x] : T$, entonces por la regla sI tenemos $\Gamma \vdash \alpha.\mathbf{u}[\mathbf{b}/x] : \lfloor\alpha\rfloor.T$. Nótese que $\alpha.\mathbf{u}[\mathbf{b}/x] = (\alpha.\mathbf{u})[\mathbf{b}/x]$.

g) $$\dfrac{\Gamma, x : U \vdash \mathbf{u} : T \qquad T \equiv R}{\Gamma, x : U \vdash \mathbf{u} : R}\ \mathrm{Eq}$$

Por la hipótesis inductiva $\Gamma \vdash \mathbf{u}[\mathbf{b}/x] : T$, entonces por la regla Eq tenemos $\Gamma \vdash \mathbf{u}[\mathbf{b}/x] : R$.

□

Lema A.1.1 (Tipo para 0)**.** $\Gamma \vdash \mathbf{0} : T \Rightarrow T \equiv \bar{0}$.

Demostración. Trivial. □

Lema A.1.2 (Linealidad de 0)**.**

- $\Gamma \vdash (\mathbf{0})\,\mathbf{u} : T \Rightarrow T \equiv \bar{0}$.
- $\Gamma \vdash (\mathbf{u})\,\mathbf{0} : T \Rightarrow T \equiv \bar{0}$.

Demostración.

- Por el lema 1, existen enteros no negativos α, β y tipos $U, T_1, \dots, T_\alpha$ tales que

$$\begin{cases} \Gamma \vdash \mathbf{0} : \sum_{i=1}^{\alpha} U \to T_i \\ \Gamma \vdash \mathbf{u} : \beta.U \\ \sum_{i=1}^{\alpha} \beta.T_i \equiv T \end{cases}$$

Por el lema A.1.1, tenemos que $\alpha = 0$ y $\bar{0} \equiv T \Rightarrow T \equiv \bar{0}$.

- Por el lema 1, existen enteros no negativos α, β y tipos $U, T_1, \dots, T_\alpha$ tales que

$$\begin{cases} \Gamma \vdash \mathbf{u} : \sum_{i=1}^{\alpha} U \to T_i \\ \Gamma \vdash \mathbf{0} : \beta.U \\ \sum_{i=1}^{\alpha} \beta.T_i \equiv T \end{cases}$$

Por el lema A.1.1, tenemos que $\beta = 0$ y $\bar{0} \equiv T \Rightarrow T \equiv \bar{0}$.

$\square$

Lema A.1.3 (Producto)**.** *Si* $\Gamma \vdash \alpha.(\beta.\mathbf{u}) : T$ *entonces* $\Gamma \vdash (\alpha \times \beta).\mathbf{u} : R$ *con* $T \preccurlyeq R$.

Demostración. Por lema 4, existe R_1 tal que $\Gamma \vdash \beta.\mathbf{u} : R_1$ donde $\lfloor\alpha\rfloor.R_1 \equiv T$. Por lema 4 nuevamente, existe R_2 tal que $\Gamma \vdash \mathbf{u} : R_2$ donde $\lfloor\beta\rfloor.R_2 \equiv R_1$. Entonces tenemos

$$\dfrac{\Gamma \vdash \mathbf{u} : R_2}{\Gamma \vdash (\alpha \times \beta).\mathbf{u} : \lfloor\alpha \times \beta\rfloor.R_2}\ \mathrm{sI}$$

Dado que $\alpha, \beta \geq 0$, se cumple que $\lfloor\alpha \times \beta\rfloor.R_2 \succcurlyeq \lfloor\alpha\rfloor\lfloor\beta\rfloor.R_2 \equiv \lfloor\alpha\rfloor.R_1 \equiv T$. Por lo tanto, concluimos que $T \preccurlyeq \lfloor\alpha \times \beta\rfloor.R_2$. $\square$

Lema A.1.4 (Distributividad)**.** *Si* $\Gamma \vdash \alpha.(\mathbf{u} + \mathbf{v}) : T$ *entonces* $\Gamma \vdash \alpha.\mathbf{u} + \alpha.\mathbf{v} : T$.

Demostración. Por lema 4, existe R tal que $\Gamma \vdash \mathbf{u} + \mathbf{v} : R$ donde $\lfloor\alpha\rfloor.R \equiv T$. Por lema 3, existen S_1, S_2 tales que $\Gamma \vdash \mathbf{u} : S_1$ y $\Gamma \vdash \mathbf{v} : S_2$, donde $S_1 + S_2 \equiv R$.

Por reglas sI y $+_I$, tenemos que

$$\dfrac{\dfrac{\Gamma \vdash \mathbf{u} : S_1}{\Gamma \vdash \alpha.\mathbf{u} : \lfloor\alpha\rfloor.S_1}\ \mathrm{sI} \quad \dfrac{\Gamma \vdash \mathbf{v} : S_2}{\Gamma \vdash \alpha.\mathbf{v} : \lfloor\alpha\rfloor.S_2}\ \mathrm{sI}}{\Gamma \vdash \alpha.\mathbf{u} + \alpha.\mathbf{v} : \lfloor\alpha\rfloor.S_1 + \lfloor\alpha\rfloor.S_2}\ +_I$$

Nótese que $\lfloor\alpha\rfloor.S_1 + \lfloor\alpha\rfloor.S_2 \equiv \lfloor\alpha\rfloor.(S_1 + S_2) \equiv \lfloor\alpha\rfloor.R \equiv T$

$\square$

Lema A.1.5 (Factorización)**.** *Si* $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t} : T$ *entonces* $\Gamma \vdash (\alpha + \beta).\mathbf{t} : R$ *y* $T \preccurlyeq R$.

Demostración. Por lema 3, existen T_1, T_2 tales que $\Gamma \vdash \alpha.\mathbf{t} : T_1$ y $\Gamma \vdash \beta.\mathbf{t} : T_2$ donde $T_1 + T_2 \equiv T$. Por lema 4, existen R_1, R_2 tales que

$$\begin{cases} \Gamma \vdash \mathbf{t} : R_1 \\ \lfloor\alpha\rfloor.R_1 \equiv T_1 \end{cases} \quad \text{and} \quad \begin{cases} \Gamma \vdash \mathbf{t} : R_2 \\ \lfloor\beta\rfloor.R_2 \equiv T_2 \end{cases}$$

Tenemos

$$\dfrac{\Gamma \vdash \mathbf{t} : R_1}{\Gamma \vdash (\alpha + \beta).\mathbf{t} : \lfloor\alpha + \beta\rfloor.R_1}\ \mathrm{sI}$$

Ahora probaremos que $\lfloor\alpha + \beta\rfloor.R_1 \succcurlyeq T$. Como R_1 y R_2 son dos tipos para $\mathbf{t}$, tenemos $R_1 \equiv R_2$ entonces

$$\begin{array}{rcl} \lfloor\alpha + \beta\rfloor.R_1 & \succcurlyeq & (\lfloor\alpha\rfloor + \lfloor\beta\rfloor).R_1 \\ & \equiv & \lfloor\alpha\rfloor.R_1 + \lfloor\beta\rfloor.R_1 \\ & \equiv & T_1 + T_2 \\ & \equiv & T \end{array}$$

Por lo tanto, concluimos $T \preccurlyeq \lfloor\alpha + \beta\rfloor.R_1$. $\square$

Lema A.1.6 (Términos básicos con tipos unitarios)**.** *Sea* $\mathbf{b}$ *una variable o una abstracción (es decir, un término básico). Entonces* $\Gamma \vdash \mathbf{b} : T$ *implica que existe un tipo unitario* U *tal que* $\Gamma \vdash \mathbf{b} : U$.

Demostración. Si $\mathbf{b}$ es una variable, debe tener un tipo dado por su contexto, que debe ser un tipo unitario. Si $\mathbf{b}$ es una abstracción, debe tener un tipo dado por la regla $\to_{\mathrm{I}}$ o la regla $\forall_{\mathrm{I}}$. En ambos casos, se trata de tipos unitarios. $\square$

Lema A.1.7 (Sumas de unitarios)**.** *Sea T un tipo, entonces existe un entero no negativo α y tipos unitarios $U_1, U_2, \dots, U_\alpha$ tales que $T \equiv \sum_{i=1}^{\alpha} U_i$.*

Demostración. Inducción en la estructura de T.

Casos base. Si T es un tipo unitario o $\bar{0}$, el resultado es trivial.

Casos inductivos. Sea $T \equiv T_1 + T_2$. Por hipótesis inductiva tenemos que $T_1 \equiv \sum_{i=1}^{\alpha_1} V_i$ y $T_2 \equiv \sum_{i=1}^{\alpha_2} W_i$ para ciertos enteros α_1, α_2 y tipos unitarios $V_1, V_2, \dots, V_{\alpha_1}$ y $W_1, W_2, \dots, W_{\alpha_2}$. Por lo tanto, podemos tomar $\alpha = \alpha_1 + \alpha_2$ y $U_i \equiv V_i$ para $i = 1, \dots, \alpha_1$, y $U_{\alpha_1+i} \equiv W_i$ para $i = 1, \dots, \alpha_2$. En consecuencia, tenemos $T \equiv \sum_{i=1}^{\alpha} U_i$. □

Subject Reduction

Teorema 4.2.1 (Subject Reduction módulo $\preccurlyeq$)**.** Para todo término $\mathbf{t}$ y $\mathbf{t}'$, contexto Γ y tipo T, si $\mathbf{t} \to \mathbf{t}'$ y $\Gamma \vdash \mathbf{t} : T$, entonces existe algún tipo R tal que $\Gamma \vdash \mathbf{t}' : R$ y $T \preccurlyeq R$, donde la relación $\preccurlyeq$ está definida inductivamente en la figura 4.4.

Demostración. Procedemos por inducción en la estructura de la relación $\to$.

Casos base

Grupo E

regla u+0 $\to$ u. Sea $\Gamma \vdash \mathbf{u} + \mathbf{0} : T$. Entonces por lema 3, existen tipos R, S tales que

$$\begin{cases} \Gamma \vdash \mathbf{u} : R & (1) \\ \Gamma \vdash \mathbf{0} : S & (2) \\ R + S \equiv T \end{cases}$$

Por lema A.1.1 y (2), tenemos $S \equiv \bar{0}$, entonces $R + \bar{0} \equiv R \equiv T$, y, por la regla $\text{E}\textsc{q}$ y (1), concluimos $\Gamma \vdash \mathbf{u} : T \succcurlyeq T$.

regla $0.\mathbf{u} \to \mathbf{0}$. Sea $\Gamma \vdash 0.\mathbf{u} : T$, entonces por lema 4 tenemos $\bar{0} \equiv T$, en consecuencia $T \equiv \bar{0}$. Por regla $\text{Ax}_{\bar{0}}$ tenemos $\Gamma \vdash \mathbf{0} : \bar{0} \succcurlyeq T$.

regla $1.\mathbf{u} \to \mathbf{u}$. Sea $\Gamma \vdash 1.\mathbf{u} : T$, entonces por lema 4, existe un tipo R tal que $\Gamma \vdash \mathbf{u} : R$ y $R \equiv T$. Entonces, por regla $\text{E}\textsc{q}$ tenemos $\Gamma \vdash \mathbf{u} : T \succcurlyeq T$.

regla $\alpha.\mathbf{0} \to \mathbf{0}$. Sea $\Gamma \vdash \alpha.\mathbf{0} : T$, entonces por el lema 4 tenemos $\Gamma \vdash \mathbf{0} : R$ y $\lfloor \alpha \rfloor.R \equiv T$. Por el lema A.1.1, tenemos $R \equiv \bar{0}$ y $\bar{0} \equiv T$. Por la regla $\text{E}\textsc{q}$ concluimos $\Gamma \vdash \mathbf{0} : T \succcurlyeq T$.

regla $\alpha.(\beta.\mathbf{u}) \to (\alpha \times \beta).\mathbf{u}$. Verdadero por lema A.1.3.

regla $\alpha.(\mathbf{u} + \mathbf{v}) \to \alpha.\mathbf{u} + \alpha.\mathbf{v}$. Verdadero por lema A.1.4.

Grupo F

regla $\alpha.\mathbf{u} + \beta.\mathbf{u} \to (\alpha + \beta).\mathbf{u}$. Verdadero por lema A.1.5.

regla $\alpha.\mathbf{u} + \mathbf{u} \to (\alpha + 1).\mathbf{u}$. Sea $\Gamma \vdash \alpha.\mathbf{u} + \mathbf{u} : T$, entonces

$$\frac{\Gamma \vdash \alpha.\mathbf{u} + \mathbf{u} : T}{\Gamma \vdash 1.(\alpha.\mathbf{u} + \mathbf{u}) : 1.T} \text{sI}$$

Entonces por lema A.1.4, $\Gamma \vdash 1.\alpha.\mathbf{u} + 1.\mathbf{u} : R_1$ con $R_1 \preccurlyeq 1.T$. Por lema 3, existen tipos R, S tales que

$$\begin{cases} \Gamma \vdash 1.\alpha.\mathbf{u} : R \\ \Gamma \vdash 1.\mathbf{u} : S \\ R + S \equiv R_1 \end{cases}$$

Por lema A.1.3, tenemos $\Gamma \vdash \alpha.\mathbf{u} : R$, entonces

$$\dfrac{\Gamma \vdash \alpha.\mathbf{u} : R \qquad \Gamma \vdash 1.\mathbf{u} : S}{\Gamma \vdash \alpha.\mathbf{u} + 1.\mathbf{u} : R + S} +\mathrm{I}$$

y $\Gamma \vdash \alpha.\mathbf{u} + 1.\mathbf{u} : R_1$ por la regla EQ. Finalmente, por el lema A.1.5, concluimos que $\Gamma \vdash (\alpha + 1).\mathbf{u} : R_2 \succcurlyeq R_1 \succcurlyeq 1.T \equiv T$. Entonces, $T \preccurlyeq R_2$.

regla $\mathbf{u}+\mathbf{u} \to (1+1).\mathbf{u}$. Sea $\Gamma \vdash \mathbf{u} + \mathbf{u} : T$. Entonces, por la regla SI, $\Gamma \vdash 1.(\mathbf{u} + \mathbf{u}) : 1.T$. Entonces por el lema A.1.4, tenemos $\Gamma \vdash 1.\mathbf{u} + 1.\mathbf{u} : R_1$ with $T \preccurlyeq R_1$. Entonces por el lema A.1.5, $\Gamma \vdash (1 + 1).\mathbf{u} : R_2$ con $R_1 \preccurlyeq R_2$, y por lo tanto $T \preccurlyeq R_2$.

regla $(\mathbf{u} + \mathbf{v})\,\mathbf{w} \to (\mathbf{u})\,\mathbf{w} + (\mathbf{v})\,\mathbf{w}$. Sea $\Gamma \vdash (\mathbf{u} + \mathbf{v})\,\mathbf{w} : T$. Entonces, por lema 1 existen enteros no negativos α, β y tipos $U, T_1, \dots, T_\alpha$ tales que

$$\begin{cases} \Gamma \vdash \mathbf{u} + \mathbf{v} : \sum_{i=1}^{\alpha}(U \to T_i) \\ \Gamma \vdash \mathbf{w} : \beta.U \\ \sum_{i=1}^{\alpha}(\beta.T_i) \equiv T \end{cases}$$

Entonces por lema 3, existen tipos R, S tales que

$$\begin{cases} \Gamma \vdash \mathbf{u} : R \\ \Gamma \vdash \mathbf{v} : S \\ R + S \equiv \sum_{i=1}^{\alpha}(U \to T_i) \end{cases}$$

Además, por lema A.1.7, existen enteros no negativos δ, γ y tipos $V_1, \dots, V_\delta, W_1, \dots, W_\gamma$ tales que $R \equiv \sum_{j=1}^{\delta} V_j$, $S \equiv \sum_{k=1}^{\gamma} W_k$, donde estas sumas son *mínimas*, es decir, $W_i \neq \bar{0}$ y $V_i \neq \bar{0}$ para todo i.

Por la forma de $\sum_{i=1}^{\alpha}(U \to T_i)$, tenemos que para cada $j = 1, \dots, \delta$, existe un i_j tal que $V_j \equiv U \to T_{i_j}$. Sea $A \subset \{1, \dots, \alpha\}$ tal que $i_j \in A$ para todo j. Análogamente, para cada $k = 1, \dots, \gamma$, existe un i_k tal que $W_k \equiv U \to T_{i_k}$. Entonces,

$$\begin{aligned} R &\equiv \sum_{j=1}^{\delta} V_j \equiv \sum_{j=1}^{\delta} U \to T_{i_j} \equiv \sum_{i \in A} U \to T_i \\ S &\equiv \sum_{k=1}^{\gamma} W_k \equiv \sum_{k=1}^{\gamma} U \to T_{i_k} \equiv \sum_{k \in \bar{A}} U \to T_k \end{aligned}$$

y por la regla EQ, tenemos que $\Gamma \vdash \mathbf{u} : \sum_{i \in A} U \to T_i$ y $\Gamma \vdash \mathbf{v} : \sum_{k \in \bar{A}} U \to T_k$. En consecuencia,

$$\dfrac{\dfrac{\Gamma \vdash \mathbf{u} : \sum\limits_{i \in A} U \to T_i \qquad \Gamma \vdash \mathbf{w} : \beta.U}{\Gamma \vdash (\mathbf{u})\,\mathbf{w} : \sum\limits_{i \in A} \beta.T_i} \to_{\mathrm{E}} \qquad \dfrac{\Gamma \vdash \mathbf{v} : \sum\limits_{k \in \bar{A}} U \to T_k \qquad \Gamma \vdash \mathbf{w} : \beta.U}{\Gamma \vdash (\mathbf{v})\,\mathbf{w} : \sum\limits_{k \in \bar{A}} \beta.T_k} \to_{\mathrm{E}}}{\Gamma \vdash (\mathbf{u})\,\mathbf{w} + (\mathbf{v})\,\mathbf{w} : \sum\limits_{i \in A} \beta.T_i + \sum\limits_{k \in \bar{A}} \beta.T_k} +_{\mathrm{I}}$$

Nótese que $\sum_{i \in A} \beta.T_i + \sum_{k \in \bar{A}} \beta.T_k \equiv \sum_{i=1}^{\alpha} \beta.T_i \equiv T$. Entonces por regla EQ tenemos que $\Gamma \vdash (\mathbf{u})\,\mathbf{w} + (\mathbf{v})\,\mathbf{w} : T \succcurlyeq T$.

regla $(\mathbf{w})\,(\mathbf{u} + \mathbf{v}) \to (\mathbf{w})\,\mathbf{u} + (\mathbf{w})\,\mathbf{v}$. Sea $\Gamma \vdash (\mathbf{w})\,(\mathbf{u} + \mathbf{v}) : T$. Entonces por lema 1, existen enteros no negativos α, β y tipos $U, T_1, \dots, T_\alpha$ tales que

$$\begin{cases} \Gamma \vdash \mathbf{w} : \sum_{i=1}^{\alpha}(U \to T_i) \\ \Gamma \vdash \mathbf{u} + \mathbf{v} : \beta.U \\ \sum_{i=1}^{\alpha}(\beta.T_i) \equiv T \end{cases}$$

Entonces por lema 3, existen tipos R, S tales que

$$\begin{cases} \Gamma \vdash \mathbf{u} : R \\ \Gamma \vdash \mathbf{v} : S \\ R + S \equiv \beta.U \equiv \sum_{i=1}^{\beta} U \end{cases}$$

Sabemos que $\beta = \delta + \gamma$, donde

$$\begin{array}{rcl} R & \equiv & \delta.U \\ S & \equiv & \gamma.U \end{array}$$

Entonces, por la regla Eq, tenemos $\Gamma \vdash \mathbf{u} : \delta.U$ y $\Gamma \vdash \mathbf{v} : \gamma.U$. Por lo tanto,

$$\dfrac{\dfrac{\Gamma \vdash \mathbf{w} : \sum_{i=1}^{\alpha} U \to T_i \qquad \Gamma \vdash \mathbf{u} : \delta.U}{\Gamma \vdash (\mathbf{w})\,\mathbf{u} : \sum_{i=1}^{\alpha} \delta.T_i} \to_{\mathrm{E}} \qquad \dfrac{\Gamma \vdash \mathbf{w} : \sum_{i=1}^{\alpha} U \to T_i \qquad \Gamma \vdash \mathbf{v} : \gamma.U}{\Gamma \vdash (\mathbf{w})\,\mathbf{v} : \sum_{i=1}^{\alpha} \gamma.T_i} \to_{\mathrm{E}}}{\Gamma \vdash (\mathbf{w})\,\mathbf{u} + (\mathbf{w})\,\mathbf{v} : \sum_{i=1}^{\alpha} \delta.T_i + \sum_{i=1}^{\alpha} \gamma.T_i} +_{\mathrm{I}}$$

Nótese que $\sum_{i=1}^{\alpha} \delta.T_i + \sum_{i=1}^{\alpha} \gamma.T_i \equiv \sum_{i=1}^{\alpha}(\delta.T_i + \gamma.T_i) \equiv \sum_{i=1}^{\alpha} \beta.T_i \equiv T$. Entonces por la regla Eq tenemos $\Gamma \vdash (\mathbf{w})\,\mathbf{u} + (\mathbf{w})\,\mathbf{v} : T \succcurlyeq T$.

regla $(\mathbf{0})\,\mathbf{u} \to \mathbf{0}$. Verdadero por el lema A.1.2 y la regla $\mathrm{Ax}_{\vec{0}}$.

regla $(\mathbf{u})\,\mathbf{0} \to \mathbf{0}$. Verdadero por el lema A.1.2 y la regla $\mathrm{Ax}_{\vec{0}}$.

Grupo B

regla $(\lambda x : U.\mathbf{t})\,\mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$. Sea $\Gamma \vdash (\lambda x : U.\mathbf{t})\,\mathbf{b} : T$. Entonces, por el lema 1 existen enteros no negativos α, β y tipos $T_1, \ldots, T_\alpha$ tales que

$$\begin{cases} \Gamma \vdash \lambda x : U.\mathbf{t} : \sum_{i=1}^{\alpha} U \to T_i \\ \Gamma \vdash \mathbf{b} : \beta.U \\ \sum_{i=1}^{\alpha} \beta.T_i \equiv T \end{cases}$$

Sin embargo, por el lema A.1.6, tenemos $\Gamma \vdash \lambda x : U.\mathbf{t} : U \to T_1$ y $\Gamma \vdash \mathbf{b} : U$. Entonces por el corolario A.1.1, tenemos $\Gamma, x : U \vdash \mathbf{t} : T_1$. Por lo tanto, por el lema 4.2.2, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T_1$. Finalmente, dado que $T_1 \equiv T$, por la regla Eq tenemos $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$.

regla $(\Lambda X.\mathbf{t})@U \to \mathbf{t}[U/X]$. Sea $\Gamma \vdash (\Lambda X.\mathbf{t})@U : T$. Entonces, por el lema 6 existe un tipo R tal que $\Gamma \vdash \Lambda X.\mathbf{t} : \forall X.R$ y $R[U/X] \equiv T$. Además, por el lema 5 existe un tipo S tal que $\Gamma \vdash \mathbf{t} : S$, donde $X \notin \mathrm{FV}(\Gamma)$ y $\forall X.S \equiv \forall X.R$. Por el lema 4.2.2, $\Gamma[U/X] \vdash \mathbf{t}[U/X] : S[U/X]$ y dado que $X \notin \mathrm{FV}(\Gamma)$ tenemos $\Gamma \vdash \mathbf{t}[U/X] : S[U/X]$. Finalmente, dado que $S \equiv R$ y $R[U/X] \equiv T$, podemos usar la regla Eq para concluir $\Gamma \vdash \mathbf{t}[U/X] : T$.

Equivalencias asociativo-conmutativas

Conmutatividad de $+$. Debemos probar que $\mathbf{u}+\mathbf{v}$ y $\mathbf{v}+\mathbf{u}$ tienen tipos equivalentes. Sea $\Gamma \vdash \mathbf{u}+\mathbf{v} : T$. Entonces, por el lema 3 existen R, S tales que $\Gamma \vdash \mathbf{u} : R$ y$\Gamma \vdash \mathbf{v} : S$, donde $R+S \equiv T$. Entonces

$$\frac{\Gamma \vdash \mathbf{v} : S \qquad \Gamma \vdash \mathbf{u} : R}{\Gamma \vdash \mathbf{v}+\mathbf{u} : S+R}\ +_{\mathrm{I}}$$

Nótese que $S+R \equiv R+S \equiv T$, entonces por la regla $\textsc{Eq}$ tenemos $\Gamma \vdash \mathbf{v}+\mathbf{u} : T$.

Asociatividad de $+$. Debemos probar que $(\mathbf{u}+\mathbf{v})+\mathbf{w}$ y $\mathbf{u}+(\mathbf{v}+\mathbf{w})$ tienen tipos equivalentes. Sea $\Gamma \vdash (\mathbf{u}+\mathbf{v})+\mathbf{w} : T$. Entonces, por el lema 3, existen R_1, R_2 tales que $\Gamma \vdash \mathbf{u}+\mathbf{v} : R_1$ y $\Gamma \vdash \mathbf{w} : R_2$ donde $R_1+R_2 \equiv T$. Entonces por el lema 3, existen S_1, S_2 tales que $\Gamma \vdash \mathbf{u} : S_1$ y $\Gamma \vdash \mathbf{v} : S_2$ donde $S_1+S_2 \equiv R_1$. Por lo tanto,

$$\frac{\Gamma \vdash \mathbf{u} : S_1 \qquad \dfrac{\Gamma \vdash \mathbf{v} : S_2 \qquad \Gamma \vdash \mathbf{w} : R_2}{\Gamma \vdash \mathbf{v}+\mathbf{w} : S_2+R_2}\ +_{\mathrm{I}}}{\Gamma \vdash \mathbf{u}+(\mathbf{v}+\mathbf{w}) : S_1+(S_2+R_2)}\ +_{\mathrm{I}}$$

Nótese que $S_1+(S_2+R_2) \equiv (S_1+S_2)+R_2 \equiv R_1+R_2 \equiv T$, entonces por la regla $\textsc{Eq}$, tenemos $\Gamma \vdash \mathbf{u}+(\mathbf{v}+\mathbf{w}) : T$.

Casos inductivos
Los casos inductivos son aplicaciones de las reglas contextuales, que son triviales. □

A.2. Normalización fuerte

Lema 4.2.3. *Si* $\mathbf{t}$ *y* $\mathbf{r}$ *son fuertemente normalizantes, entonces* $\alpha.\mathbf{t}+\beta.\mathbf{r}$ *es fuertemente normalizante.*

Demostración. Consideramos la siguiente medida algebraica positiva definida sobre los términos (tomada de [AD08]):

$$\begin{array}{rcl} |\mathbf{u}\,\mathbf{v}| & = & (3|\mathbf{u}|+2)(3|\mathbf{v}|+2) \\ |\mathbf{u}+\mathbf{v}| & = & 2+|\mathbf{u}|+|\mathbf{v}| \\ |\alpha.\mathbf{u}| & = & 1+2|\mathbf{u}| \\ |\mathbf{0}| & = & 0 \end{array}$$

Demostraremos que todas las reducciones algebraicas (es decir, las que no involucran aplicaciones o λ-abstracciones) hacen decrecer estrictamente esta medida.

- **regla $\mathbf{u}+\mathbf{0} \to \mathbf{u}$.** $|\mathbf{u}+\mathbf{0}| = 2+|\mathbf{u}|+|\mathbf{0}| > |\mathbf{u}|$.
- **regla $\alpha.\mathbf{u}+\beta.\mathbf{u} \to (\alpha+\beta).\mathbf{u}$.** $|\alpha.\mathbf{u}+\beta.\mathbf{u}| = 2+|\alpha.\mathbf{u}|+|\beta.\mathbf{u}| = 2+1+2|\mathbf{u}|+1+2|\mathbf{u}|$ $= 4+4|\mathbf{u}| > 1+2|\mathbf{u}| = |(\alpha+\beta).\mathbf{u}|$
- **regla $0.\mathbf{u} \to \mathbf{0}$.** $|0.\mathbf{u}| = 1+2|\mathbf{u}| > 0 = |\mathbf{0}|$
- **regla $\alpha.\mathbf{u}+\mathbf{u} \to (\alpha+1).\mathbf{u}$.** $|\alpha.\mathbf{u}+\mathbf{u}| = 2+|\alpha.\mathbf{u}|+|\mathbf{u}| = 2+1+2|\mathbf{u}|+|\mathbf{u}| = 3+3|\mathbf{u}| > 1+2|\mathbf{u}| = |(\alpha+1).\mathbf{u}|$
- **regla $1.\mathbf{u} \to \mathbf{u}$.** $|1.\mathbf{u}| = 1+2|\mathbf{u}| > |\mathbf{u}|$
- **regla $\mathbf{u}+\mathbf{u} \to (1+1).\mathbf{u}$.** $|\mathbf{u}+\mathbf{u}| = 2+|\mathbf{u}|+|\mathbf{u}| = 2+2|\mathbf{u}| > 1+2|\mathbf{u}| = |(1+1).\mathbf{u}|$
- **regla $\alpha.\mathbf{0} \to \mathbf{0}$.** $|\alpha.\mathbf{0}| = 1+2|\mathbf{0}| > 0 = |\mathbf{0}|$
- **regla $\alpha.(\beta.\mathbf{u}) \to (\alpha\times\beta).\mathbf{u}$.** $|\alpha.(\beta.\mathbf{u})| = 1+2(1+2|\mathbf{u}|) = 3+4|\mathbf{u}| > 1+2|\mathbf{u}| = |(\alpha\times\beta).\mathbf{u}|$

- **regla** $\alpha.(\mathbf{u}+\mathbf{v}) \to \alpha.\mathbf{u}+\alpha.\mathbf{v}$. $|\alpha.(\mathbf{u}+\mathbf{v})| = 1+2(2+|\mathbf{u}|+|\mathbf{v}|) = 5+2|\mathbf{u}|+2|\mathbf{v}|$
$> 2+(1+2|\mathbf{u}|)+(1+2|\mathbf{v}|) = |\alpha.\mathbf{u}+\alpha.\mathbf{v}|$

Finalmente, podemos completar la prueba. El término $\alpha.\mathbf{t}+\beta.\mathbf{r}$ sólo se reduce por reglas contextuales o algebraicas. Si se usan las contextuales, se estarán reduciendo los términos $\mathbf{t}$ y $\mathbf{r}$, que son fuertemente normalizantes, mientras que si se usan las reglas algebraicas, la reducción eventualmente terminará cuando la medida alcance su valor mínimo.

$\square$

Lema 4.2.4. *Sean* A *y* B *candidatos de reducibilidad. Entonces,* $\mathsf{A} \to \mathsf{B}$*,* $\mathsf{A} \oplus \mathsf{B}$*, y* $\Lambda\mathsf{A}$*,* $A \cap B$ *y* $\overline{\emptyset}$ *son todos candidatos de reducibilidad.*

Demostración. En la prueba que sigue, denotamos con $\nu(\mathbf{t})$ a la longitud de la secuencia de reducción más larga que parte de $\mathbf{t}$. Verificaremos que se cumplan las tres condiciones de reducibilidad para cada operador.

- Sea $\mathbf{t} \in \mathsf{A} \to \mathsf{B}$.

  ($\mathbf{CR}_1$) Inducción sobre la construcción de $\mathsf{A} \to \mathsf{B}$. Sea $\mathbf{f} \in \{\mathbf{t} \mid \forall \mathbf{a} \in \mathsf{A}.(\mathbf{t})\,\mathbf{a} \in \mathsf{B}\}$ y $\mathbf{a} \in \mathsf{A}$. Sabemos que $(\mathbf{f})\,\mathbf{a}$ es fuertemente normalizante, y $\nu(\mathbf{f}) \leq \nu((\mathbf{f})\,\mathbf{a})$. Dado que $\nu((\mathbf{f})\,\mathbf{a})$ es finito por (CR_1) en B, tenemos que $\nu(\mathbf{f})$ es finito, entonces $\mathbf{f}$ es fuertemente normalizante.

  Sea $\mathbf{t} \in \mathsf{A} \to \mathsf{B}$ un término neutral donde $\mathrm{Red}(\mathbf{t}) \subseteq \mathsf{A} \to \mathsf{B}$. Por hipótesis inductiva, $\mathrm{Red}(\mathbf{t}) \subseteq \mathsf{SN}_0$, entonces $\mathbf{t}$ debe ser fuertemente normalizante.

  ($\mathbf{CR}_2$) Inducción en la construcción de $\mathsf{A} \to \mathsf{B}$. Sea $\mathbf{f} \in \{\mathbf{t} \mid \forall \mathbf{a} \in \mathsf{A}.(\mathbf{t})\,\mathbf{a} \in \mathsf{B}\}$ y $\mathbf{a} \in \mathsf{A}$. Sea $\mathbf{f} \to^* \mathbf{f}'$ y consideremos $(\mathbf{f})\,\mathbf{a} \in \mathsf{B}$, entonces $(\mathbf{f})\,\mathbf{a} \to^* (\mathbf{f}')\,\mathbf{a}$. Por ($\mathrm{CR}_2$) en B, tenemos $(\mathbf{f}')\,\mathbf{a} \in \mathsf{B}$. Por lo tanto, $\mathbf{f}' \in \mathsf{A} \to \mathsf{B}$.

  Sea $\mathbf{t} \in \mathsf{A} \to \mathsf{B}$ un término neutral donde $\mathrm{Red}(\mathbf{t}) \subseteq \mathsf{A} \to \mathsf{B}$. El resultado es trivial.

  ($\mathbf{CR}_3$) Trivial por construcción.

- Sea $\mathbf{t} \in \mathsf{A} \oplus \mathsf{B}$.

  ($\mathbf{CR}_1$) Inducción en la construcción de $\mathsf{A} \oplus \mathsf{B}$. Si $\mathbf{t} \in \{\alpha.\mathbf{t}+\beta.\mathbf{r} \mid \mathbf{t} \in \mathsf{A}, \mathbf{r} \in \mathsf{B}\}$, el resultado es trivial por la condición (CR_1) en A y B y el lema 4.2.3. Si $\mathbf{t} \to^* \mathbf{t}'$ con $\mathbf{t} \in \mathsf{A} \oplus \mathsf{B}$, entonces $\mathbf{t}$ es fuertemente normalizante por hipótesis inductiva; por lo tanto, también lo es $\mathbf{t}'$. Si $\mathbf{t}$ es neutral y $\mathrm{Red}(\mathbf{t}) \subseteq \mathsf{A} \oplus \mathsf{B}$, entonces $\mathbf{t}$ es fuertemente normalizante dado que, por hipótesis inductiva, todos los elementos de $\mathrm{Red}(\mathbf{t})$ son fuertemente normalizantes.

  ($\mathbf{CR}_2$) and ($\mathbf{CR}_3$) Triviales por construcción de $\mathsf{A} \oplus \mathsf{B}$.

- Sea $\mathbf{t} \in \Lambda\mathsf{A}$.

  ($\mathbf{CR}_1$) Sea V un tipo arbitrario, entonces $\mathbf{t}@V \in \mathsf{A}$, y por ($\mathrm{CR}_1$) en A tenemos que $\mathbf{t}@V$ es fuertemente normalizante. Pero $\nu(\mathbf{t}) < \nu(\mathbf{t}@V)$, entonces $\mathbf{t}$ es fuertemente normalizante.

  ($\mathbf{CR}_2$) Sea $\mathbf{t} \to^* \mathbf{t}'$, entonces para todo tipo V tenemos $\mathbf{t}@V \in \mathsf{A}$ y $\mathbf{t}@V \to^* \mathbf{t}'@V$. Por ($\mathrm{CR}_2$) sabemos que $\mathbf{t}'@V \in \mathsf{A}$, entonces $\mathbf{t}' \in \Lambda\mathsf{A}$.

  ($\mathbf{CR}_3$) Sea $\mathbf{t}$ un término neutral y supongamos que $\mathrm{Red}(\mathbf{t}) \subseteq \Lambda\mathsf{A}$. Sea V un tipo arbitrario, entonces realizando un paso de reducción en $\mathbf{t}@V$ debe dar $\mathbf{t}'@V$ para algún $\mathbf{t}'$ ya que $\mathbf{t}$ es neutral. Además, $\mathbf{t}'@V \in \mathsf{A}$ porque $\mathbf{t}' \in \Lambda\mathsf{A}$. Por ($\mathrm{CR}_3$) tenemos que $\mathbf{t}@V \in \mathsf{A}$, y entonces $\mathbf{t} \in \Lambda\mathsf{A}$.

- En el caso de $\mathsf{A} \cap \mathsf{B}$, todas las condiciones se cumplen trivialmente ya que se cumplen tanto para A como para B.

- Para el caso de $\overline{\emptyset}$, notamos que si $\mathbf{t} \in \overline{\emptyset}$, entonces $\mathbf{t}$ debe ser neutral y $\mathrm{Red}(\mathbf{t}) \subseteq \emptyset$. Esto implica que $\mathbf{t}$ es neutral y normal, por lo que las condiciones de reducibilidad se cumplen trivialmente.

$\square$

Lema A.2.1. *Para todos los tipos T y T', si $T \equiv T'$ entonces para cada valuación ρ, tenemos $[\![T]\!]_\rho = [\![T']\!]_\rho$.*

Demostración. Es suficiente probar que si $\mathsf{A}, \mathsf{B}, \mathsf{C}$ son candidatos de reducibilidad, entonces se cumple que

1. $\mathsf{A} \oplus \mathsf{B} = \mathsf{B} \oplus \mathsf{A}$
2. $(\mathsf{A} \oplus \mathsf{B}) \oplus \mathsf{C} = \mathsf{A} \oplus (\mathsf{B} \oplus \mathsf{C})$
3. $\mathsf{A} \oplus \overline{\emptyset} = \mathsf{A}$

Las propiedades (1), (2) y (3) se obtienen de las reglas de reescritura para la conmutatividad, la asociatividad y la neutralidad de $\mathbf{0}$, junto con la condición (CR_2). $\square$

Lema A.2.2. *Para todos los tipos unitarios U, V y valuaciones ρ, $[\![U[V/X]]\!]_\rho = [\![U]\!]_{\rho, X \mapsto [\![V]\!]_\rho}$*

Demostración. Inducción sobre U. Sea ρ una valuación.

- Si U es la variable X, entonces $[\![U[V/X]]\!]_\rho = [\![V]\!]_\rho = (\rho, X \mapsto [\![V]\!]_\rho)(X) = [\![U]\!]_{\rho, X \mapsto [\![V]\!]_\rho}$. Si U es alguna variable $Y \neq X$, el resultado es trivial.
- Si $U = W \to R$, entonces $[\![(W \to R)[V/X]]\!]_\rho = [\![W[V/X] \to R[V/X]]\!]_\rho = [\![W[V/X]]\!]_\rho \to [\![R[V/X]]\!]_\rho$. Por hipótesis inductiva, tenemos $[\![W[V/X]]\!]_\rho \to [\![R[V/X]]\!]_\rho =$ $[\![W]\!]_{\rho, X \mapsto [\![V]\!]_\rho} \to [\![R]\!]_{\rho, X \mapsto [\![V]\!]_\rho} = [\![W \to R]\!]_{\rho, X \mapsto [\![V]\!]_\rho}$.
- Si $U = \forall Y.W$, entonces $[\![(\forall Y.W)[V/X]]\!]_\rho = [\![\forall Y.W[V/X]]\!]_\rho = \bigcap_{\mathsf{S} \in \mathsf{RC}} \Lambda [\![W[V/X]]\!]_{\rho, Y \mapsto S}$. Por hipótesis inductiva, tenemos que $\bigcap_{\mathsf{S} \in \mathsf{RC}} \Lambda [\![W[V/X]]\!]_{\rho, Y \mapsto S} = \bigcap_{\mathsf{S} \in \mathsf{RC}} \Lambda [\![W]\!]_{\rho, Y \mapsto S, X \mapsto [\![V]\!]_\rho} =$ $[\![\forall Y.W]\!]_{\rho, X \mapsto [\![V]\!]_\rho}$.

$\square$

El siguiente lema demuestra que cualquier juicio de tipado derivable es válido.

Lema 4.2.5 (Lema de adecuación). *Sea $\Gamma \vdash \mathbf{t} : T$, entonces $\Gamma \vDash \mathbf{t} : T$.*

Demostración. Procedemos por inducción en la derivación de $\Gamma \vdash \mathbf{t} : T$.

Casos base.

1. Caso Ax: $\Gamma', x : U \vdash x : U$. Trivial.
2. Caso $\mathrm{Ax}_{\bar{0}}$: $\Gamma \vdash \mathbf{0} : \bar{0}$. Trivial porque $\mathbf{0}$ es irreducible y neutral.

Casos inductivos. Sea ρ una valuación y $\langle \sigma, \delta \rangle$ un par de sustituciones que satisfacen Γ en ρ.

1. Caso $+_I$: $\dfrac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}$

   Debemos probar que $\mathbf{t}_{\delta\sigma} + \mathbf{r}_{\delta\sigma} \in [\![T_\delta]\!]_\rho \oplus [\![R_\delta]\!]_\rho$. Sabemos que $\mathbf{t}_{\delta\sigma} \in [\![T_\delta]\!]_\rho$ y $\mathbf{r}_{\delta\sigma} \in [\![R_\delta]\!]_\rho$ por hipótesis inductiva. Por lo tanto, $\mathbf{t}_{\delta\sigma} + \mathbf{r}_{\delta\sigma} \in [\![T_\delta]\!]_\rho \oplus [\![R_\delta]\!]_\rho$ por definición de $\oplus$.

2. Caso $\forall_I$: $\dfrac{\Gamma \vdash \mathbf{t} : U \qquad X \notin \mathrm{FV}(U)}{\Gamma \vdash \Lambda X.\mathbf{t} : \forall X.U}$

   Por hipótesis inductiva, tenemos que $\Gamma \vDash \mathbf{t} : U$. Debemos probar que $((\Lambda X.\mathbf{t})_\delta)_\sigma \in [\![(\forall X.U)_\delta]\!]_\rho$, entonces mostraremos que $((\Lambda X.\mathbf{t})_\delta)_\sigma @V \in [\![U_\delta]\!]_{\rho,X\mapsto\mathsf{S}}$ para cada tipo V y candidato de reducibilidad $\mathsf{S} \in \mathsf{RC}$. Dado que $\mathbf{t}_{\delta\sigma} \in [\![U_\delta]\!]_\rho$, sabemos que es fuertemente normalizante por (CR_1). Por lo tanto, procedemos por inducción en la longitud de su secuencia de reducción. Las reducciones de $((\Lambda X.\mathbf{t})_\delta)_\sigma @V$ pueden ser

   - $((\Lambda X.\mathbf{t}')_\delta)@V$ con $\mathbf{t}_{\delta\sigma} \to \mathbf{t}'$, que está en $[\![U_\delta]\!]_{\rho,X\mapsto\mathsf{S}}$ por hipótesis inductiva (interna).
   - $\mathbf{t}_{\delta\sigma}[V/X] \in [\![U_\delta]\!]_{\rho,X\mapsto\mathsf{S}}$ por hipótesis inductiva (externa).

3. Caso $\forall_E$: $\dfrac{\Gamma \vdash \mathbf{t} : \forall X.U}{\Gamma \vdash \mathbf{t}@V : U[V/X]}$

   Por hipótesis inductiva, $\mathbf{t}_{\delta\sigma}@V \in [\![U_\delta]\!]_{\rho,X\mapsto\mathsf{S}}$ para cada candidato S. Tomando $\mathsf{S} = [\![V_\delta]\!]_\rho$, el resultado se concluye a partir del lema A.2.2.

4. Caso $\to_I$: $\dfrac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x : U.\mathbf{t} : U \to T}$

   Por hipótesis inductiva, tenemos que $\Gamma, x : U \vDash \mathbf{t} : T$. Probaremos que $(\lambda x : U_\delta.\mathbf{t}_\delta)_\sigma \in [\![U_\delta \to T_\delta]\!]_\rho$. Podemos asumir sin pérdida de generalidad que $x \notin \mathrm{FV}(\sigma)$, ya que siempre podemos realizar una α-conversión para lograrlo. Sea $\mathbf{u} \in [\![U_\delta]\!]_\rho$, y sea $\sigma' = (x \mapsto \mathbf{u}; \sigma)$. Entonces $\langle \sigma', \delta \rangle \in [\![\Gamma]\!]_\rho$, por lo tanto $\mathbf{t}_{\delta\sigma'} \in [\![T_\delta]\!]_\rho$. Esto significa que tanto $\mathbf{u}$ como $\mathbf{t}_{\delta\sigma'}$ son fuertemente normalizantes, por lo que primero probaremos que todas las reducciones de $(\lambda x : U_\delta.\mathbf{t}_\delta)_\sigma\, \mathbf{u}$ están en $[\![T_\delta]\!]_\rho$.

   - $(\lambda x : U_\delta.\mathbf{t}_\delta)_\sigma\, \mathbf{u} \to (\lambda x : U_\delta.\mathbf{t}')\, \mathbf{u}$ o $(\lambda x : U_\delta.\mathbf{t}_\delta)_\sigma\, \mathbf{u} \to (\lambda x : U_\delta.\mathbf{t}_\delta)_\sigma\, \mathbf{u}'$, con $\mathbf{t}_{\delta\sigma} \to \mathbf{t}'$ o $\mathbf{u} \to \mathbf{u}'$. El resultado se prueba por inducción en $\nu(\mathbf{u})$ y $\nu(\mathbf{t}_{\delta\sigma'})$, respectivamente: por hipótesis inductiva tenemos $\mathbf{t}', \mathbf{u}' \in [\![T_\delta]\!]_\rho$, entonces $(\lambda x : U_\delta.\mathbf{t}')\, \mathbf{u}$, y $((\lambda x : U_\delta.\mathbf{t}_\delta)_\sigma)\, \mathbf{u}' \in [\![T_\delta]\!]_\rho$.
   - $(\lambda x : U_\delta.\mathbf{t}_\delta)_\sigma\, \mathbf{u} \to (\mathbf{t}_\delta)_\sigma[x := \mathbf{u}] = (\mathbf{t}_\delta)_{\sigma'} \in [\![T_\delta]\!]_\rho$.

   Por lo tanto, $(\lambda x : U_\delta.\mathbf{t}_\delta)_\sigma\, \mathbf{u}$ es un término neutral con todas sus reducciones en $[\![T_\delta]\!]_\rho$, entonces $(\lambda x : U_\delta.\mathbf{t}_\delta)_\sigma\, \mathbf{u} \in [\![T_\delta]\!]_\rho$. En consecuencia, por la definición de $\to$, concluimos que $(\lambda x : U_\delta.\mathbf{t}_\delta)_\sigma \in [\![U_\delta \to T_\delta]\!]_\rho$.

5. Caso $\to_E$: $\dfrac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{\alpha} U \to T_i \qquad \Gamma \vdash \mathbf{r} : \beta.U}{\Gamma \vdash (\mathbf{t})\, \mathbf{r} : \sum_{i=1}^{\alpha} \beta.T_i}$

   Por hipótesis inductiva, tenemos que $\Gamma \vDash \mathbf{t} : \sum_{i=1}^{\alpha} U \to T_i$ y $\Gamma \vDash \mathbf{r} : \beta.U$. Entonces por (CR_1) tanto $\mathbf{t}_{\delta\sigma}$ como $\mathbf{r}_{\delta\sigma}$ son fuertemente normalizantes. Dado que $(\mathbf{t}_{\delta\sigma})\, \mathbf{r}_{\delta\sigma}$ es neutral, por (CR_3) basta probar que todas sus reducciones $\mathbf{s}$ pertenecen a $[\![(\sum_{i=1}^{\alpha} \beta.T_i)_\delta]\!]_\rho$ para asegurar que $(\mathbf{t}_{\delta\sigma})\, \mathbf{r}_{\delta\sigma} \in [\![(\sum_{i=1}^{\alpha} \beta.T_i)_\delta]\!]_\rho$.

   Si $\mathbf{s} = (\mathbf{t}')\, \mathbf{r}_{\delta\sigma}$, o $\mathbf{s} = (\mathbf{t}_{\delta\sigma})\, \mathbf{r}'$ (donde $\mathbf{t}_{\delta\sigma} \to \mathbf{t}'$ y $\mathbf{r}_{\delta\sigma} \to \mathbf{r}'$), el resultado se prueba por inducción en $\nu(\mathbf{t}_{\delta\sigma})$ y $\nu(\mathbf{r}_{\delta\sigma})$ respectivamente.

   En todos los otros casos, procedemos por inducción en $\alpha + \beta$.

   - Si $\mathbf{s} = \mathbf{0}$, entonces $\mathbf{s} \in [\![(\sum_{i=1}^{\alpha} \beta.T_i)_\delta]\!]_\rho$ porque $\mathbf{0}$ está en todo candidato de reducibilidad.

- Si $\mathbf{s} = (\mathbf{t}_{\delta\sigma})\,\mathbf{r_1} + (\mathbf{t}_{\delta\sigma})\,\mathbf{r_2}$ (donde $\mathbf{r}_{\delta\sigma} = \mathbf{r_1} + \mathbf{r_2}$), entonces por lema 3 existen tipos R_1, R_2 tales que $\Gamma \vdash \mathbf{r_1} : R_1$ y $\Gamma \vdash \mathbf{r_2} : R_2$ y $R_1 + R_2 \equiv \beta.U$. Esto significa que $R_1 = \delta_1.U$, $R_2 = \delta_2.U$ para ciertos enteros δ_1, δ_2 tales que $\beta = \delta_1 + \delta_2$. Por lo tanto, por hipótesis inductiva, $(\mathbf{t}_{\delta\sigma})\,\mathbf{r_1}_{\delta\sigma} \in [\![(\sum_{i=1}^{\alpha} \delta_1.T_i)_\delta]\!]_\rho$ y $(\mathbf{t}_{\delta\sigma})\,\mathbf{r_2}_{\delta\sigma} \in [\![(\sum_{i=1}^{\alpha} \delta_2.T_i)_\delta]\!]_\rho$. Entonces, por definición de $\oplus$, $(\mathbf{t}_{\delta\sigma})\,\mathbf{r_1}_{\delta\sigma} + (\mathbf{t}_{\delta\sigma})\,\mathbf{r_2}_{\delta\sigma} \in [\![(\sum_{i=1}^{\alpha} \beta.T_i)_\delta]\!]_\rho$.
- Si $\mathbf{s} = (\mathbf{t_1})\,\mathbf{r}_{\delta\sigma} + (\mathbf{t_2})\,\mathbf{r}_{\delta\sigma}$ (donde $\mathbf{t} = \mathbf{t_1} + \mathbf{t_2}$), entonces por lema 3 existen tipos R, S tales que $\Gamma \vdash \mathbf{t_1} : R$ y $\Gamma \vdash \mathbf{t_2} : S$, donde $R + S \equiv \sum_{i=1}^{\alpha} U \to T_i$. Esto significa que $R \equiv \sum_{i=1}^{\alpha'} U \to T_{\pi(i)}$ y $S \equiv \sum_{i=\alpha'+1}^{\alpha} U \to T_{\pi(i)}$ para algún $1 < \alpha' < \alpha$ y alguna permutación π de $[1 \dots \alpha]$. Por hipótesis inductiva, $(\mathbf{t_1}_{\delta\sigma})\,\mathbf{r}_{\delta\sigma} \in [\![(\sum_{i=1}^{\alpha'} \beta.T_{\pi(i)})_\delta]\!]_\rho$ y $(\mathbf{t_2}_{\delta\sigma})\,\mathbf{r}_{\delta\sigma} \in [\![(\sum_{i=\alpha'+1}^{\alpha} \beta.T_{\pi(i)})_\delta]\!]_\rho$, entonces $\mathbf{s} \in [\![(\sum_{i=1}^{\alpha'} \beta.T_{\pi(i)})_\delta]\!]_\rho \oplus [\![(\sum_{i=\alpha'+1}^{\alpha} \beta.T_{\pi(i)})_\delta]\!]_\rho = [\![(\sum_{i=1}^{\alpha} \beta.T_i)_\delta]\!]_\rho$.
- Si $\mathbf{s} = \gamma.(\mathbf{b}_{\delta\sigma}\,\mathbf{r}_{\delta\sigma})$ (donde $\mathbf{t} = \gamma.\mathbf{b}$), entonces por el lema 4 existe un tipo R tal que $\Gamma \vdash \mathbf{b} : R$ y $\lfloor\gamma\rfloor.R \equiv \sum_{i=1}^{\alpha} U \to T_i$. Esto significa que $R \equiv \sum_{i=\phi}^{\psi} U \to T_{\pi(i)}$ para algún $1 \leq \phi < \psi \leq \alpha$ y alguna permutación π de $[1 \dots \alpha]$. Entonces por hipótesis inductiva $(\mathbf{b}_{\delta\sigma})\,\mathbf{r}_{\delta\sigma} \in [\![(\sum_{i=\phi}^{\psi} \beta.T_{\pi(i)})_\delta]\!]_\rho$, y así tenemos que $\mathbf{s} \in [\![(\sum_{i=1}^{\alpha} \beta.T_i)_\delta]\!]$.
- Si $\mathbf{s} = \gamma.((\mathbf{t}_{\delta\sigma})\,\mathbf{b}_{\delta\sigma})$ (donde $\mathbf{r} = \gamma.\mathbf{b}$), entonces por el lema 4 existe un tipo R tal que $\Gamma \vdash \mathbf{b} : R$ y $\lfloor\gamma\rfloor.R \equiv \beta.U$. Esto significa que hay un entero δ tal que $R \equiv \delta.U$ y $\beta = \lfloor\gamma\rfloor \times \delta$. Pero entonces $[\![R_\delta]\!]_\rho = [\![\beta.U_\delta]\!]_\rho$, por lo que $\mathbf{b}_{\delta\sigma} \in [\![\beta.U_\delta]\!]_\rho$. En consencuencia, $(\mathbf{t}_{\delta\sigma})\,\mathbf{b}_{\delta\sigma} \in [\![(\sum_{i=1}^{\alpha} \beta.T_i)_\delta]\!]_\rho$ y así tenemos que $\mathbf{s} \in [\![(\sum_{i=1}^{\alpha} \beta.T_i)_\delta]\!]_\rho$.
- Si $\mathbf{s} = \mathbf{t}'[x := \mathbf{r}]$ donde $\mathbf{t} = \lambda x : U.\mathbf{t}'$ y $\mathbf{r}$ es un término básico, con $\alpha = \beta = 1$, entonces $\mathbf{t}_{\delta\sigma} \in [\![U_\delta \to (T_1)_\delta]\!]_\rho$ y $\mathbf{r}_{\delta\sigma} \in [\![U_\delta]\!]_\rho$ implica que $\mathbf{s} \in [\![(T_1)_\delta]\!]_\rho$ por la definición de $\to$.

Por lo tanto, por (CR_3) concluimos que $(\mathbf{t}_{\delta\sigma})\,\mathbf{r}_{\delta\sigma} \in [\![(\sum_{i=1}^{\alpha} \beta.T_i)_\delta]\!]$.

6. Caso Eq: $\dfrac{\Gamma \vdash \mathbf{t} : T \qquad T \equiv R}{\Gamma \vdash \mathbf{t} : R}$

   Verdadero por lema A.2.1.

7. Caso sI: $\dfrac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \lfloor\alpha\rfloor.T}$

   Por hipótesis inductiva, tenemos $\Gamma \vDash \mathbf{t} : T$. Entonces $\mathbf{t}_{\delta\sigma} \in [\![T_\delta]\!]_\rho$, y por lo tanto $(\alpha.\mathbf{t})_{\delta\sigma} \in \bigoplus_{i=1}^{\lfloor\alpha\rfloor} [\![T_\delta]\!]_\rho = [\![\lfloor\alpha\rfloor.T_\delta]\!]_\rho$ por construcción.

$\square$

Bibliografía

[AD08] Pablo Arrighi and Gilles Dowek. Linear-algebraic lambda-calculus: higher-order, encodings, and confluence. In Andrei Voronkov, editor, *Proceedings of RTA-2008*, volume 5117 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 2008.

[ADC11] Pablo Arrighi and Alejandro Díaz-Caro. Scalar system F for linear-algebraic λ-calculus: Towards a quantum physical logic. In Bob Coecke, Prakash Panangaden, and Peter Selinger, editors, *Proceedings of QPL-2009*, volume 270/2 of *Electronic Notes in Theoretical Computer Science*, pages 206–215. Springer, 2011.

[ADCV11] Pablo Arrighi, Alejandro Díaz-Caro, and Benoît Valiron. A type system for the vectorial aspects of the linear-algebraic lambda-calculus. In *Proceedings of the 7th International Workshop on Developments of Computational Methods (DCM 2011)*, Zurich, Switzerland, 2011. A publicar en EPTCS. Borrador en `http://www.diaz-caro.info/vectorial.pdf`.

[Bar92] Henk P. Barendregt. *Lambda calculi with types*, volume 2 of *Handbook of Logic in Computer Science*. Clarendon Press, Oxford, UK, 1992.

[BDCJ11] Pablo Buiras, Alejandro Díaz-Caro, and Mauro Jaskelioff. Confluence via strong normalisation in an algebraic λ-calculus with rewriting. In *Proceedings of the 6th Workshop on Logical and Semantic Frameworks, with Applications (LSFA 2011)*, Belo Horizonte, Brazil, 2011. A publicar en EPTCS. Borrador en `http://www.diaz-caro.info/CA.pdf`.

[Chu32] Alonzo Church. A Set of Postulates for the Foundation of Logic. *Annals of Mathematics*, 2(33):346–366, 1932.

[Chu33] Alonzo Church. A Set of Postulates for the Foundation of Logic, Second Paper. *The Annals of Mathematics*, 34(4):839–864, October 1933.

[Chu36] Alonzo Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2):345–363, April 1936.

[Chu40] A. Church. A formulation of the simple theory of types. *The journal of symbolic logic*, 5(2):56–68, 1940.

[Cur34] H. B. Curry. Functionality in Combinatory Logic. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 20, pages 584–590, November 1934.

[Cur58] Haskell B. Curry. *Combinatory logic / [by] Haskell B. Curry [and] Robert Feys. With two sections by William Craig.* North-Holland Pub. Co., Amsterdam, :, 1958.

[DC11] Alejandro Díaz-Caro. *Du typage vectoriel.* PhD thesis, Université de Grenoble, France, September 23, 2011.

[DCP11] Alejandro Díaz-Caro and Barbara Petit. Sums in linear algebraic lambda-calculus. Draft at `http://www.diaz-caro.info/additive.pdf`, November 2011.

[DCPTV10] Alejandro Díaz-Caro, Simon Perdrix, Christine Tasson, and Benoît Valiron. Equivalence of algebraic λ-calculi. In *Informal proceedings of HOR-2010*, pages 6–11, Edinburgh, UK, July 14, 2010.

[ER01] Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309:2003, 2001.

[Gir71] J. Y. Girard. Une extension de l'interprétation de gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types. In J. E. Fenstad, editor, *Proceedings of the Scandinavian Logic Symposium*, pages 63–92. North-Holland, 1971.

[GLT89] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 1989.

[How80] William A. Howard. The formulae-as-types notion of construction. pages 479–490, London, 1980. Academic Press.

[Kah05] Stefan Kahrs. Term rewriting systems by terese, cambridge university press, 2003. *J. Funct. Program.*, 15, July 2005.

[Kle36] S.C. Kleene. λ-definability and recursiveness. *Duke Mathematical Journal*, 2(2):340–353, 1936.

[Kri90] Jean-Louis Krivine. *Lambda-calcul: types et modèles*. Études et recherches en informatique. Masson, Paris, France, 1990.

[Mil78] Robin Milner. A theory of type polymorphism in programming. *Journal of Computer and System Sciences*, 17:348–375, 1978.

[Rey74] John C. Reynolds. Towards a theory of type structure. In B. Robinet, editor, *Proceedings of the Colloque sur la Programmation*, volume 19 of *Lecture Notes in Computer Science*, pages 408–425. Springer, 1974.

[Vau09] Lionel Vaux. The algebraic lambda calculus. *Mathematical Structures in Computer Science*, 19(5):1029–1059, 2009.

[WZ82] William K. Wootters and Wojciech .H. Zurek. A single quantum cannot be cloned. *Nature*, 299:802–803, 1982.