Facultad de Ciencias
Universidad de la República
Montevideo, Uruguay

Trabajo Monográfico

Realizabilidad Clásica de Krivine

Juan Pablo Martínez Delbugio
Tutor: Dr. Mauricio Guillermo

Julio, 2022

Presentado al Tribunal Evaluador como requisito de graduación de la carrera Licenciatura en Matemática

Había aprendido sin esfuerzo el inglés, el francés, el portugués, el latín. Sospecho, sin embargo, que no era muy capaz de pensar. Pensar es olvidar diferencias, es generalizar, abstraer. En el abarrotado mundo de Funes no había sino detalles, casi inmediatos.

Jorge Luis Borges, *Funes el memorioso*

Resumen

El objetivo de esta monografía es presentar la Realizabilidad Clásica de Krivine. Para ello, comenzamos introduciendo una extensión del λ-cálculo, como es el λ_c-cálculo, y aspectos de la lógica de segundo orden. En base a esto, definimos la Realizabilidad Clásica de Krivine como una semántica de la aritmética de Peano de segundo orden, la cual permite interpretar las fórmulas de dicha teoría como programas del λ_c-cálculo, diferenciándose de interpretaciones *booleanas* como los modelos de Henkin. Posteriormente, vemos cómo esta semántica induce modelos de Henkin y estudiamos dichos modelos probando, por ejemplo, la existencia de enteros no estándar en algunos de ellos, al igual que la validez del axioma de elección dependiente (DC) y con ello la posibilidad de realizar el análisis clásico. Para finalizar, especificamos el comportamiento de los programas en el valor de verdad de algunas fórmulas interesantes, como la Ley de Peirce.

Agradecimientos

A mis padres y a mi hermana, quienes desde su lugar siempre me han brindado apoyo en los caminos que he recorrido. A mis amigos, que me han acompañado en todos estos años. A mi tutor Mauricio Guillermo, por haberme presentado la temática para esta monografía, y permitirme conocer un mundo de ideas completamente desconocido para mi. A la UDELAR, por brindarme la posibilidad de estudiar aquello que me apasiona. Gracias a todos.

Prefacio

Esta monografía está basada, principalmente, en los capítulos 1 y 2 de la tesis doctoral de Lionel Rieg de la ENS de Lyon [1]. Además, con el fin de añadir algunos temas y resultados omitidos en dicha tesis, se utilizaron distintos textos referenciados en las secciones correspondientes.

Índice

1. Contexto Histórico

En esta primera sección, presentamos someramente la realizabilidad de Kleene. Para esto, primero introducimos la interpretación BHK de las demostraciones, comenzando por el intuicionismo de Brouwer y la lógica intuicionista. Dicha exposición se basa en los contenidos e ideas de [2], [3], [4] y [5]. A partir de dicha interpretación, describimos las ideas y algunos resultados de la realizabilidad de Kleene, con el fin de motivar el estudio de la realizabilidad clásica de Krivine. Esta última parte se basa principalmente en resultados expuestos en [6].

1.1. Intuicionismo vs Formalismo

El intuicionismo, introducido por el matemático y filósofo holandés Luitzen Egbertus Jan Brouwer (1881 - 1966), es una doctrina que se opuso al formalismo, defendido por el prestigioso matemático alemán David Hilbert (1862 - 1943), en una polémica que tomó relevancia a principios del S XX.

Una de las principales características del intuicionismo, es que entiende que la matemática es independiente de la experiencia del mundo externo y, en particular, del lenguaje. Por lo tanto, establece que no es posible reducir el pensamiento matemático a un conjunto de reglas formales sobre el lenguaje. Además de dicho rechazo al formalismo, el intuicionismo de Brouwer también incluía una visión constructivista, sosteniendo que la veracidad de una sentencia matemática sólo puede ser concebida a partir de una construcción mental que pruebe que es verdadera. Algunas de las ideas de Brouwer eran:

- La matemática no es formal. Los objetos matemáticos son construcciones mentales en la mente de un matemático "ideal" (el *Creative Subject*) y sólo las construcciones mentales de dicho matemático son exactas.

- La matemática es independiente del lenguaje. La comunicación a través del lenguaje puede servir para sugerir construcciones mentales similares a otros matemáticos, pero no hay garantía de que estas otras construcciones sean las mismas (este es un elemento solipsista de la filosofía de Brouwer).

- La matemática no depende de la lógica, por el contrario, la lógica es una parte de la matemática.

Diametralmente opuesta a esta es la visión del formalismo, la cual -según Brouwer- sostiene que la exactitud matemática consiste meramente en el método de desarrollo de relaciones entre objetos matemáticos, y que es independiente del significado que uno le quiera dar a las mismas, o a los objetos que se relacionan. Por lo tanto, para el formalista, estas series de relaciones existen matemáticamente sólo cuando son representadas en un lenguaje junto con las reglas lógicas de las cuales depende su desarrollo, formando así lo que es conocido como lógica simbólica.

En el formalismo, de ciertas relaciones entre objetos matemáticos, las cuales se asumen como axiomas, se pueden deducir otras relaciones a partir de reglas de deducción/inferencia, con la convicción de que se deducen verdades a partir de verdades al razonar lógicamente. Uno de los cuestionamientos que Brouwer hacía sobre el formalismo, era que dicha visión debería llevar al convencimiento de que si se modificaran las reglas de inferencia, esto no debería invalidar la exactitud matemática. Más aún, planteaba que se debería llevar a cabo la tarea de comprender por qué algunos sistemas de lógica simbólica se proyectaban mejor en la naturaleza que otros, y que se debería explicar por qué creemos en algunos sistemas y no en otros que, por ejemplo, permiten que tanto una sentencia como su negación pueden ser válidas al mismo tiempo.

1.2. Lógica intuicionista

Dentro del discurso filosófico del intuicionismo, además del rechazo al formalismo, también se puede encontrar otro componente interesante como es una nueva noción de demostración y de verdad matemática. Este segundo aspecto fue formalizado por Arend Heyting (1898 - 1990), lo cual podría verse como irónico debido a la postura de Brouwer ante el formalismo, y el hecho de que Heyting fuese su alumno.

Según el intuicionismo, la veracidad de una sentencia matemática sólo puede ser concebida por una construcción que pruebe que es verdadera. La dependencia del intuicionismo en el tiempo es esencial, las sentencias se pueden volver demostrables en el transcurso del tiempo y, por lo tanto, se pueden volver verdaderas en el sentido intuicionista, sin haberlo sido antes. Esto último, difiere de la idea Platonista -presente en la lógica clásica- en donde las sentencias matemáticas son desde un principio verdaderas o falsas, y las construcciones matemáticas nos permiten descubrir/conocer la realidad que ya existe.

Esta visión de Brouwer tiene implicaciones importantes en la práctica diaria de la disciplina. Principalmente, una de sus consecuencias es que el principio del tercer excluido ($A \vee \neg A$) no continúa siendo válido. De hecho, Brouwer afirmaba que el hecho de que el tercer excluido fuera válido en sistemas finitos, donde tanto una sentencia o su negación se puede validar luego de una cantidad finita de pasos, había llevado a que se extrapolara incorrectamente a la generalidad de las sentencias matemáticas.

Debido a que, en el intuicionismo, probar la negación de una sentencia se reduce a probar que toda prueba de la sentencia se puede transformar en una prueba del absurdo, $A \vee \neg A$ se interpreta como que existe una prueba de A, o que se puede probar de forma intuicionista que no hay una prueba de A. Según Brouwer, debido a la existencia de sentencias que aún no han sido probadas ni refutadas, no es válido afirmar $A \vee \neg A$ para dichos casos. Incluso, adelantándose a la existencia de problemas indecidibles, sospechaba que quizás hubieran sentencias las cuales nunca podrían ser probadas ni refutadas.

A pesar de que Brouwer se oponía al formalismo, Heyting formalizó la lógica intuicionista con el fin de reflejar esta nueva visión de las pruebas. Para esto, se pueden utilizar las reglas de deducción clásicas pero eliminando el principio del tercer excluido o, equivalentemente, la reducción por absurdo o la eliminación de la doble negación ($\neg\neg A \Rightarrow A$).

Es interesante ver que si bien se elimina la reducción al absurdo, representada en el sistema de deducción natural clásica generalmente por la siguiente regla:

$$\frac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}\ \text{RAA}$$

en el sistema intuicionista se mantiene la eliminación del $\bot$:

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash A}\ \bot_e$$

También es importante no confundir la reducción al absurdo con la introducción de la negación, que sigue siendo válida en la lógica intuicionista:

$$\frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\ \neg_i$$

Algunas de las implicancias/equivalencias que se pueden probar en la lógica intuicionista, son las siguientes:

$$\begin{array}{rcl} A & \Rightarrow & \neg\neg A \\ (A \Rightarrow B) & \Rightarrow & (\neg B \Rightarrow \neg A) \\ (\neg A \vee B) & \Rightarrow & (A \Rightarrow B) \\ \neg A & \Leftrightarrow & \neg\neg\neg A \end{array}$$

Sin embargo, el recíproco no vale en los primeros tres casos, como sí pasa en la lógica clásica. De todos modos, vale remarcar que no es que dichos recíprocos, o el tercer excluido, no sean nunca válidos en el intuicionismo, sino que deben ser probados constructivamente (en algunas teorías pueden valer). Por ejemplo, en la teoría de conjuntos, el axioma de elección implica el tercer excluido aún en el sistema de deducción intuicionista.

1.3. Interpretación BHK

En 1934, Heyting introdujo una forma, que luego se conoció como la interpretación BHK (Brouwer-Heyting-Kolmogorov), que captura el significado de los símbolos lógicos en el intuicionismo y, en general, en el constructivismo. Básicamente, define de manera informal en qué consiste una prueba intuicionista al indicar cómo se deben interpretar los cuantificadores y los conectores:

- No existen pruebas de $\perp$.
- Una prueba de $A \wedge B$ consiste de una prueba de A y de una prueba de B.
- Una prueba de $A \vee B$ consiste de una prueba de A o una prueba de B.
- Una prueba de $A \Rightarrow B$ es una construcción que transforma pruebas de A en pruebas de B.
- Una prueba de $\exists x.A(x)$ está conformada por un elemento d del dominio, y una prueba de $A(d)$.
- Una prueba de $\forall x.A(x)$ es una construcción que transforma cualquier d en el dominio, en una prueba de $A(d)$.

Como mencionamos anteriormente, la existencia de una prueba equivale a la validez de una fórmula, por lo que también es una manera de darle semántica al intuicionismo.

1.4. Realizabilidad de Kleene

La realizabilidad de Kleene se aprovecha de la interpretación BHK para, valga la redundancia, interpretar las fórmulas de la aritmética de Heyting (en general hablaremos de la teoría HA). Dicha teoría consiste de los mismos axiomas que la aritmética de Peano (la teoría PA), pero utiliza las reglas de inferencia de la lógica intuicionista.

En esta sección, presentaremos someramente la realizabilidad de Kleene -y algunos de sus resultados interesantes- sin entrar en los detalles técnicos que se requieren. A lo largo de la monografía, veremos cómo se formalizan estos conceptos, o conceptos similares, al presentar la realizabilidad de Krivine, pero esta sección servirá simplemente para motivar el estudio de dicha realizabilidad clásica.

Siguiendo el enfoque filosófico del intuicionismo, en donde la validez de una fórmula A depende únicamente de la existencia de una prueba de A, se interpreta cada fórmula efectivamente como el conjunto de sus evidencias.

Para la noción de *evidencia* de A, basta seguir la línea de la interpretación BHK, que en cierto modo establece que una evidencia de $A \wedge B$ es una evidencia de A y una de B, y así para los otros casos, a excepción del caso de la igualdad en donde la interpretación BHK no toma una postura. Para dicho caso, cualquier evidencia será suficiente si los términos en la igualdad son representados de la misma manera en el modelo estándar de los naturales.

Nuestras evidencias serán naturales y, en lugar de decir que un natural es una evidencia de A, diremos que dicho natural **realiza** A. Los naturales nos permitirán flexibilidad suficiente como para poder formalizar la noción de "transformador de evidencias", necesarios en la interpretación del para todo y del implica. Para esto, utilizamos la codificación ϕ_n: para todo n, ϕ_n es una función $\phi_n : \mathbb{N} \to \mathbb{N}$ y la podemos pensar como un programa cuya codificación es el número n. En la siguiente sección, formalizaremos el concepto de programa al introducir dos modelos de computación, pero por ahora no es necesario entrar en dicho detalle.

En base a esto, tenemos la siguiente definición de realizabilidad:

- La fórmula $\perp$ no tiene realizadores.
- Un natural n realiza una fórmula $t = t'$ si, y sólo si, las interpretaciones de t y t' coinciden en el modelo estándar de los naturales.
- Un par (n, m) realiza $A \wedge B$ si, y sólo si, n realiza A y m realiza B.
- Un par (n, m) realiza $A \vee B$ si, y sólo si, $n = 0$ y m realiza A, o $n = 1$ y m realiza B.
- Un natural n realiza $A \Rightarrow B$ si, y sólo si, para todo m que realice A, $\phi_n(m)$ realiza B.
- Un par (n, m) realiza $\exists x.A(x)$ si, y sólo si, m realiza $A(n)$.
- Un natural n realiza la fórmula $\forall x.A(x)$ si, y sólo si, para todo m, $\phi_n(m)$ realiza $A(m)$.

Observación 1.1 *Es importante hacer algunas observaciones:*

1. *Para bajar a tierra la idea de la igualdad, basta saber que en el lenguaje de HA existe la función sucesor (con su símbolo s), la constante 0, y la función suma (con su símbolo $+$), las cuales tienen tienen su interpretación trivial en el modelo estándar $\mathbb{N}$. Por lo tanto, la fórmula $(s(0) + 0) = s(0)$, por ejemplo, es realizable (cualquier natural sirve como evidencia).*

2. *Cualquier par* (n, m) *se puede codificar como un único natural, por lo que en general podemos pensar a los realizadores siempre como naturales.*

3. *Si definimos* $\neg A \equiv A \Rightarrow \bot$*, entonces* A *es realizable si, y sólo si,* $\neg A$ *no lo es.*

El primer resultado, el cual no es trivial, es que la aritmética de Heyting es realizable. En otras palabras, que todas las fórmulas derivables a partir de los axiomas de Peano (utilizando las reglas de la lógica intuicionista) tienen un realizador. Este resultado de corrección, es conocido como el Teorema de Adecuación.

Teorema 1.1 (Teorema de Adecuación) *Si* $HA \vdash A$*, entonces existe un* n *natural que realiza* A*.*

Este primer resultado nos da la idea de que nuestra interpretación modela, en algún sentido, la teoría HA.

Corolario 1.1 HA *es consistente* ($HA \nvdash \bot$)*.*

Demostración. Si $HA \vdash \bot$, entonces la fórmula $\bot$ es realizable, lo cual es imposible por definición. $\square$

Si bien al suponer la existencia del modelo estándar de PA, ya podíamos ver la consistencia de HA (ya que $HA \subseteq PA$ y PA es consistente por la existencia de un modelo), es interesante ver cómo se replica el mismo argumento utilizado para los modelos de Tarski en la lógica clásica: la existencia de un modelo implica la consistencia.

Algo interesante de esta interpretación intuicionista es que, a diferencia de las interpretaciones clásicas, no necesariamente valdrá el tercer excluido. Para esto, basta tomar un problema indecidible como el *Halting Problem*. En este problema, dado un natural x, se debe retornar *sí* o *no* según si la máquina de Turing codificada por x finaliza su ejecución o no lo hace (queda ejecutando para siempre). En 1936, Turing probó que no existe un programa que resuelva dicho problema para toda entrada x.

Nuevamente, sin entrar en detalles, se puede escribir una fórmula de primer orden $H(x)$ que exprese que "la máquina de Turing x finaliza su ejecución en una cantidad finita de pasos", y luego definir la fórmula $HP := \forall x.\, H(x) \vee \neg H(x)$. Recordemos que, en lógica intuicionista, no podemos entender dicha fórmula como "toda máquina de Turing finaliza su ejecución o no lo hace", sino que hay que interpretarla como "para toda máquina de Turing, se puede probar que su ejecución termina, o que no lo hace". Más específicamente, en la realizabilidad de Kleene esto se traduce a "existe un programa que, dado x, nos retorna una prueba de $(H(x) \vee \neg H(x))$".

Como una prueba de $(H(x) \vee \neg H(x))$ es un par (n, m) donde $n = 0$ implica la existencia de una prueba de $H(x)$, y $n = 1$ implica la existencia de una prueba de $\neg H(x)$, intuitivamente dicho programa resolvería el *Halting Problem*, por lo que es de esperar que no exista:

Proposición 1.1 *La fórmula HP no es realizable.*

Como habíamos anticipado, esto evidencia que la realizabilidad de Kleene no es una semántica clásica. Además, dicho resultado permite probar que HA está contenido estrictamente en PA.

Teorema 1.2 *El principio del tercer excluido no es derivable en HA.*

Demostración. Como $HA \nvdash HP \equiv \forall x.\, H(x) \vee \neg H(x)$ (sino tendríamos que HP es realizable), se concluye que $HA \nvdash H(x) \vee \neg H(x)$. □

Otro aspecto importante es distinguir la derivabilidad en HA, de la realizabilidad de una fórmula. Si bien la primera implica la segunda, el recíproco no es válido (la realizabilidad es más laxa).

Proposición 1.2 *Para cualquier natural n, n realiza $\neg HP$, pero $HA \nvdash \neg HP$.*

Demostración. Como HP no es realizable, obtenemos que $\neg HP \equiv HP \Rightarrow \perp$ es realizable por cualquier natural (para todo n, $\phi_n(m)$ realiza $\perp$ para cualquier m que realice HP, ya que no hay quien realice a este último). Sin embargo, como $PA \nvdash \neg HP$ ya que $PA \vdash HP$ (por el tercer excluido), nos queda que $HA \nvdash \neg HP$ tampoco. □

Los modelos de realizabilidad son útiles también para probar resultados de independencia. Particularmente, el principio de Markov que denotamos MP ($MP :=$ $\forall x(A(x) \vee \neg A(x)) \Rightarrow \neg\neg\exists x A(x) \Rightarrow \exists x A(x)$), el cual es válido en la lógica clásica (ya que vale la eliminación de la doble negación), es independiente de la teoría HA. Para esto, se puede puede ver que es realizable en la realizabilidad de Kleene y que no es realizable en otra realizabilidad similar, conocida como la realizabilidad de Kreisel.

Proposición 1.3 *MP es realizable en la realizabilidad de Kleene.*

Proposición 1.4 *MP no es realizable en la realizabilidad de Kreisel.*

Teorema 1.3 *MP es independiente de HA y, de hecho, $HA \subset HA + MP \subset PA$.*

Demostración. Como MP es realizable en la realizabilidad de Kleene, entonces $\neg MP$ no lo es y, por lo tanto, $HA \nvdash \neg MP$. Por otro lado, como no es realizable en la realizabilidad de Kreisel, se tiene que $HA \nvdash MP$ ya que en la realizabilidad de Kreisel también vale que $HA \vdash A$ implica A realizable. □

2. Computación y Matemática

En la realizabilidad de Kleene, se puede observar la utilización de programas con el fin de realizar fórmulas. Si bien se puede entender que dicha relación nace de la búsqueda de formalizar el concepto de "transformador de pruebas" presente en la interpretación BHK, la relación entre la computación y la matemática fue un tema sumamente estudiado en el S XX y va más allá de la interpretación BHK.

En esta sección, formalizamos dos modelos de computación que serán útiles para definir la realizabilidad clásica de Krivine, al mismo tiempo que presentamos la correspondencia Curry-Howard, la cual nos permitirá entender de mejor manera la relación pruebas-programas. Para esta sección en particular, complementamos la tesis doctoral de Lionel Rieg [1] con definiciones e ideas de la β-reducción (y sus estrategias) extraídas de [7].

2.1. Vínculo histórico

El S XX fue un siglo de suma importancia para la computación, la cual impactó en la matemática, principalmente, a través de dos vías. La primera fue el famoso programa de Hilbert, principalmente a través del *Entscheidungsproblem*, cuyo fin era crear un procedimiento mecánico que permitiera decidir la veracidad de una fórmula de primer orden. Este problema fue probado imposible tanto por Alonzo Church, como por Alan Turing en 1936 de forma independiente. Dicho descubrimiento fue el punto de partida de las Ciencias de la Computación, y la primera vez que la computación formal apareció explícitamente en la matemática.

De todos modos, en dicha primera aparición, la participación de la computación se limitaba a problemas específicos de lógica. Esto cambió a partir de la segunda gran aparición de la computación que fue a través de la correspondencia Curry-Howard (nombrada debido a Haskell Curry y William Howard). Esta correspondencia permite vincular la computación con las demostraciones matemáticas, las cuales son un pilar fundamental del área.

Aún así, dicha correspondencia tenía una limitante importante: estaba restringida a la lógica constructiva. En 1990, Timothy Griffin removió esta restricción y extendió la correspondencia a la lógica clásica a través del operador de control `callcc`. Si bien Peter Landin había introducido previamente el operador J para modelar el comportamiento de la instrucción `goto` de Algol60, Griffin fue el primero en asignar un tipo a un operador de control. El descubrimiento es particularmente interesante porque vincula el operador de control `callcc` con la Ley de Peirce, ambos conocidos hacía ya muchos años, pero no antes vinculados entre sí.

Con el fin de comprender de mejor manera la correspondencia Curry-Howard, en la siguiente sección introduciremos uno de los modelos de computación más populares: el λ-cálculo. Una vez introducido dicho modelo de computación y la correspondencia Curry-Howard, presentaremos el λ_c-cálculo que permite extender esta correspondencia a la lógica clásica.

2.2. El λ-cálculo

Cuando la noción de computación fue formalizada en el S XX, principalmente para responder al *Entscheidungsproblem*, varios métodos equivalentes fueron creados al mismo tiempo. Entre estos, los modelos de computación más importantes son la Máquina de Turing, el λ-cálculo, y las Funciones Recursivas.

La Máquina de Turing tiene una descripción principalmente algorítmica, lo que la vuelve fácil de implementar en comparación a otros modelos de computación. Por otro lado, las Funciones Recursivas son definidas a partir de funciones básicas (constantes y proyecciones) y tomando clausura bajo tres reglas de formación (composición, recursión y minimización), lo cual hace que los razonamientos inductivos se vuelvan naturales pero con la desventaja de que la implementación en un dispositivo físico sea más compleja. Por último, el λ-cálculo se sitúa en el medio, teniendo conexiones con la lógica y con una definición más a bajo nivel que las Funciones Recursivas, haciéndolo más fácil de implementar.

Cada modelo tiene sus propias ventajas: la Máquina de Turing es adecuada para realizar análisis de complejidad, las Funciones Recursivas para estudios de computabilidad y el λ-cálculo para el estudio de la computación en lógica. Debido al interés en las propiedades lógicas de los programas, nos centraremos en el λ-cálculo, que es la base teórica detrás de los lenguajes de programación funcional.

Una de las principales características del λ-cálculo es que es un lenguaje funcional, todo lo que existe son funciones. En particular, los argumentos de una función, por ejemplo, también son funciones. En la sintaxis, esto se traduce en que todos los objetos del lenguaje son de la misma especie: λ-términos. Empezaremos definiendo dichos términos (evitaremos el prefijo λ cuando sea posible) y luego la β-reducción la cual es la regla que nos permite evaluarlos.

Definición 2.1 (Términos) *Asumiendo un conjunto infinito numerable de variables, los términos se definen a partir de tres reglas:*

$$\lambda\textbf{-términos} \quad t, u \quad := \quad x \quad | \quad \lambda x.\, t \quad | \quad t\, u \qquad \textit{donde } x \textit{ es una variable}$$

Básicamente, un término (en general usamos las letras t y u para referirnos a estos) puede ser una variable (usualmente utilizamos x, y y z para referirnos a estas), una *abstracción* de un término o una *aplicación* de un término t a otro término u.

La abstracción la podemos pensar como la manera de crear funciones al explicitar un parámetro y el "cuerpo" de la función. Por ejemplo, el término $\lambda x.\, x$ es la función identidad. Lo interesante es que al manipular solamente términos, el argumento que puede recibir la función identidad puede ser la función misma (o cualquier otra función).

Por otro lado, la aplicación es similar a la intuición matemática, podemos pensar $t\, u$ como $t(u)$. Siguiendo el ejemplo, a un término u le podemos aplicar la función identidad utilizando la siguiente sintaxis: $(\lambda x.\, x)\, u$.

A partir de la definición de un término, se puede definir su conjunto de variables libres (aquellas variables que no están ligadas a una abstracción) y la sustitución de variables por términos, que será fundamental para definir la β-reducción.

Definición 2.2 (Variables libres) *Se definen las variables libres de un λ-término t (utilizaremos la notación $FV(t)$) recursivamente:*

$$
\begin{array}{lcl}
FV(x) & := & \{x\} \\
FV(t\, u) & := & FV(t) \cup FV(u) \\
FV(\lambda x.\, t) & := & FV(t) \setminus \{x\}
\end{array}
$$

Tanto en el λ-cálculo como en el λ_c-cálculo que presentaremos luego, trabajamos con términos a menos de α-equivalencias. En general, se utiliza la relación $\equiv$ para indicar la igualdad sintáctica de dos términos a menos de α-equivalencia: básicamente, $t \equiv t'$ si, y sólo si, cambiando las variables asociadas a un λ en t (y las ocurrencias ligadas a esa abstracción), se puede obtener t' (por más detalles, ver [7]).

Definición 2.3 (Sustitución) *Se define la sustitución de una variable x por un término u en un término t (utilizaremos la notación $t[u/x]$) recursivamente:*

$$
\begin{array}{lcll}
x[u/x] & := & u & \\
y[u/x] & := & y & \text{donde } y \neq x \\
(t\, v)[u/x] & := & t[u/x]\, v[u/x] & \\
(\lambda x.\, t)[u/x] & := & \lambda x.\, t & \\
(\lambda y.\, t)[u/x] & := & \lambda y.\, t[u/x] & \text{donde } y \neq x \\
\multicolumn{4}{l}{\quad \text{y se } \alpha\text{-renombra } y \text{ en } \lambda y.\, t \text{ si } y \in FV(u)}
\end{array}
$$

2.2.1. β-reducción y β-equivalencia

A partir de la sustitución, podemos definir la β-reducción la cual es la regla utilizada para evaluar λ-términos. Básicamente, la regla nos indica que cuando aplicamos una abstracción a un término, lo podemos reducir a una sustitución. En términos formales, sean t, u λ-términos y x una variable, decimos que $(\lambda x.\, t)\, u$ es una *expresión reducible* (y abreviamos *redex*) y su reducción es $t[u/x]$. Siguiendo el ejemplo de la función identidad, la regla nos indica que $(\lambda x.\, x)\, u$ reduce a u.

Es interesante ver que un mismo λ-término puede contener distintas *redexes*. Debido a esto, se define la relación binaria β_0 que debe entenderse como: $t\, \beta_0\, t'$ si, y sólo si, t' se obtiene al reducir una *redex* en t.

Definición 2.4 *Se define la relación binaria β_0 por inducción en un término t:*

- *Si $t \equiv x$, no existe t' tal que $t\, \beta_0\, t'$.*
- *Si $t \equiv u\, v$ entonces $t\, \beta_0\, t'$ si, y sólo si:*
    - *$t' \equiv u\, v'$ con $v\, \beta_0\, v'$.*
    - *$t' \equiv u'\, v$ con $u\, \beta_0\, u'$.*
    - *$u \equiv \lambda x.\, u'$ y $t' \equiv u'[v/x]$.*
- *Si $t \equiv \lambda x.\, u$, entonces $t\, \beta_0\, t'$ si, y sólo si, $t' \equiv \lambda x.\, u'$ con $u\, \beta_0\, u'$.*

A partir de β_0, se define su clausura reflexiva, transitiva y simétrica $\simeq_\beta$.

Definición 2.5 (β-equivalencia) *Se define la relación binaria $\simeq_\beta$ como: $t \simeq_\beta t'$ si, y sólo si, existe una secuencia de términos $t \equiv t_0, \ldots, t_n \equiv t'$ tal que $t_i\, \beta_0\, t_{i+1}$ o $t_{i+1}\, \beta_0\, t_i$ para todo $0 \leq i \leq n-1$.*

Cuando se cumple que $t \simeq_\beta t'$, decimos que dichos términos son β-equivalentes.

2.2.2. Estrategias de β-reducción

Debido a que, como ya mencionamos, un término puede contener múltiples *redexes*, existen distintas estrategias de evaluación a partir de la β-reducción que difieren en cuáles *redexes* tienen prioridad para ser reducidas. En nuestro caso, sólo nos centraremos en dos de ellas: *Normal* y *Weak Head Reduction*. En ambos casos, será necesario descomponer los términos como una serie de aplicaciones, lo cual se desprende de la siguiente observación:

Observación 2.1 *Para todo λ-término t, $t \equiv u\, u_1\, \ldots u_k$ donde u no es una aplicación y k puede ser 0 ($t \equiv u$ en ese caso), y esta forma es única.*

En base a esto, se definen las siguientes estrategias de reducción:

1. **Normal**. La *redex* más superficial y más a la izquierda se reduce primero. Notamos $t \to_\beta u$ cuando t se reduce a u utilizando un único paso de esta estrategia.

2. **Weak Head Reduction (*whr* para abreviar)**. Sea $t \equiv u\, u_1 \ldots u_k$ donde u no es una aplicación y k puede ser 0, en esta estrategia reducimos sólo cuando $k > 0$ y $u \equiv \lambda x.\, v$ ($u\, u_1$ es una *redex*). En dicho caso, podemos reducir t a $v[u_1/x]\, u_2 \ldots u_k$, y lo notamos $t \to_{whr} v[u_1/x]\, u_2 \ldots u_k$.

Es importante notar que las relaciones $\to_\beta$ y $\to_{whr}$ no son ni reflexivas, ni transitivas, ni simétricas. De todos modos, se definen sus clausuras transitivas reflexivas como $\to_{\beta *}$ y $\to_{whr*}$.

En la estrategia *whr*, el término *weak* proviene de que no se reducen las *redexes* dentro de una abstracción, mientras que *head* se debe a que sólo reducimos aquello que está al principio del término.

Cuando un término no contiene *redexes*, se dice que está en forma normal (no se puede reducir utilizando la estrategia normal). Similarmente, cuando un término no se puede reducir utilizando la estrategia *whr*, decimos que está en forma normal débil. Notar que si t está en forma normal débil, se pueden dar dos casos: $t \equiv \lambda x.u$ o $t \equiv x\, u_1 \, \ldots \, u_k$ para alguna variable x.

Veamos algunos ejemplos:

- Si tomamos $t \equiv \lambda y.\, (\lambda x.\, x)\, y$, podemos ver que $t \to_\beta \lambda y.\, y$, por lo que t no está en forma normal. A pesar de eso, t sí está en forma normal débil ya que no se puede reducir utilizando la estrategia *whr* (aquí se observa bien el porqué de la denominación *weak*).

- Si tomamos $t \equiv x\ ((\lambda x.\ x)\ y)$, es trivial ver que no está en forma normal ($t \to_\beta x\, y$), pero sí está en forma normal débil (aquí se observa el porqué de la denominación *head*).

Si $t \to_{\beta *} u$ y u está en forma normal, decimos que u es la forma normal de t. Similarmente, si $t \to_{whr*} u$ y u está en forma normal débil, decimos que u es la forma normal débil de t. Las formas normales (débiles) pueden no existir (veremos luego el caso de Ω, que cumple que $\Omega \to_\beta \Omega$) pero, en caso de existir, son únicas para cada término.

Otro aspecto importante, es que la estrategia normal es una estrategia de normalización completa. En otras palabras, si $t \simeq_\beta t'$ y t' está en forma normal (se dice que t es normalizable), entonces $t \rightarrow_{\beta *} t'$.

Para relacionar ambas estrategias, es interesante ver que si $t \rightarrow_{whr} u$ entonces $t \rightarrow_{\beta} u$. Por lo tanto, si $t \rightarrow_{\beta *} u$, donde u es la forma normal de t, tenemos que $t \rightarrow_{whr*} u' \rightarrow_{\beta *} u$ donde u' es la forma normal débil de t. Básicamente, la reducción con mayor prioridad en la estrategia normal es la reducción que *whr* permite. Por lo tanto, cualquier reducción normal comienza con reducciones *whr* hasta que se llega a la forma normal débil del término inicial, y luego continúa con el resto de las *redexes*.

2.3. Correspondencia Curry-Howard

El λ-cálculo es una presentación meramente sintáctica de la computación, que no asocia un significado particular a las funciones que manipula. Por ejemplo, se pueden describir términos que carecen de sentido lógico como $\delta := \lambda x.x\, x$, el cual aplica una variable a sí misma. Más aún, si aplicamos δ a sí misma, tenemos $\Omega := \delta\delta$. Esta función cumple una propiedad particular y es que se puede ver fácilmente que $\Omega \rightarrow_{\beta} \Omega$.

Para evitar estos casos patológicos, usualmente se utiliza algún sistema de tipos sobre el λ-cálculo. Si bien hay varios de estos sistemas, el más sencillo es la teoría de tipos simples de Church. En dicho caso, se cuenta con un conjunto $\mathcal{B}$ de tipos simples o primitivos, y un único constructor de tipos: la flecha. Este constructor se utiliza para tipar funciones. Por ejemplo, $A \rightarrow B$ es el tipo de una función que recibe un argumento de tipo A y retorna un término de tipo B.

Tipos simples $\quad A, B \quad := \quad \alpha \quad | \quad A \rightarrow B \qquad\qquad \alpha \in \mathcal{B}$

Para asociar tipos con términos, utilizamos reglas inductivas. Dada una función de tipo $A \rightarrow B$, esperamos que reciba argumentos de tipo A y retorne términos de tipo B. Esta intuición se formaliza con la siguiente regla, que se lee de arriba a abajo como una regla de inferencia, donde notamos $t : A$ cuando t tiene tipo A.

$$\frac{t : A \rightarrow B \qquad u : A}{t\, u : B}$$

Si bien asignarle un tipo a una aplicación parece ser inmediato, en el caso de la abstracción no es tan sencillo. Para esto se requiere saber tanto el tipo del argumento como el tipo del cuerpo de la función **condicionado** al tipo del argumento. En otras palabras, para derivar el tipo del cuerpo de una función deberíamos poder tener una hipótesis sobre el tipo del argumento. Con dicho fin se introducen los *contextos de tipado*, que asocian tipos a variables que en general luego son abstraídas. En base a dichos contextos, trabajamos con juicios de tipado $\Gamma \vdash t : A$ que establecen que, asumiendo un contexto de tipado Γ, t tiene tipo A.

Con la introducción de los juicios de tipado, la regla que nos permite asignarle un tipo a una función es la siguiente:

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B}$$

Agregando una regla trivial que nos permita deducir que una variable tiene un tipo, a partir de asumirlo, y transformando la primera regla que dimos para ser compatible con la introducción de los juicios de tipado, nos quedan las siguientes tres reglas:

$$\frac{}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

La correspondencia de Curry-Howard requiere ver los sistemas de tipos como sistemas de deducción. En particular, el sistema de tipos simples de Church se corresponde con el siguiente sistema de deducción minimalista de lógica intuicionista:

$$\frac{}{\Gamma, A \vdash A} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B} \qquad \frac{\Gamma \vdash A \Rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B}$$

La única diferencia es que el sistema de deducción minimalista no contiene términos y que la implicación se escribe $\Rightarrow$, y no $\to$. Esta similitud, que en realidad termina siendo un isomorfismo, sugiere que los λ-términos pueden ser vistos como testigos de una demostración. Esta es la base de la correspondencia y puede ser adaptada a otras extensiones del λ-cálculo.

En cierta manera, y como profundizaremos más adelante, el sistema de tipos nos permite pensar cada regla de inferencia del sistema de deducción como una transformación de evidencias. Por ejemplo, la tercera regla la podemos leer como: si t es una evidencia de $A \Rightarrow B$ (o, desde otra perspectiva, convierte evidencias de A en evidencias de B), y u es una evidencia de A, entonces $t\, u$ es una evidencia de B.

2.4. El λ_c-cálculo y la KAM

La Máquina Abstracta de Krivine (KAM por sus siglas en inglés) fue diseñada originalmente por Jean-Louis Krivine como una máquina de pilas para evaluar λ-términos. Dado un conjunto de argumentos, estos son evaluados en la cima de un contexto de ejecución representado por una pila. Una pila, en este caso, es una lista finita de términos cerrados que finaliza con una *constante de pila* y debe ser pensada como una memoria donde se almacena información de la ejecución de un proceso. La KAM fue luego extendida a la versión clásica del cálculo lambda, el λ_c-cálculo, que agrega principalmente dos funcionalidades: *instrucciones* y *constantes de continuación*.

Las instrucciones en el λ_c-cálculo son utilizadas para agregar funcionalidades de programación, y el conjunto de dichas instrucciones se denomina $\mathcal{K}$. Dicho conjunto debe contener al menos la instrucción `callcc`, cuya finalidad es permitir almacenar la pila de argumentos actual π en una constante de continuación k_π, la cual luego permitirá restaurar dicha pila.

Las constantes de pila (de naturaleza completamente distinta a las constantes de continuación) son tomadas de un conjunto Π_0 y, como se mencionó previamente, indican el final de una pila. Las mismas son utilizadas como marcadores y son muy importantes para algunas construcciones de la realizabilidad clásica como el modelo de hilos que veremos más adelante (en 6.2.3). De todos modos, en la mayoría de los casos las podemos ignorar y pensar una pila como una lista finita de λ_c-términos cerrados.

El λ_c-cálculo es uno de los ingredientes fundamentales de esta monografía. De la misma manera que hay una correspondencia entre el sistema de tipos simples de Church y un sistema de deducción de lógica intuicionista, veremos en las próximas secciones que se puede definir un sistema de tipos para el λ_c-cálculo que corresponde con un sistema de deducción de lógica clásica de segundo orden.

Empezamos definiendo el λ_c-cálculo similar a cómo se define el λ-cálculo:

Definición 2.6 (Términos, pilas y variables libres) *Sea $\mathcal{K}$ un conjunto de instrucciones tal que $\texttt{callcc} \in \mathcal{K}$, y Π_0 un conjunto de constantes de pila no vacío, se definen los λ_c-términos y las pilas por inducción mutua:*

λ_c-términos Si x es una variable, x es un λ_c-término y $FV(x) := \{x\}$.
Si x es una variable y t es un λ_c-término, $\lambda x.\, t$ es un λ_c-término y $FV(\lambda x.\, t) := FV(t) \setminus \{x\}$.
Si t y u son λ_c-términos, $t\,u$ es un λ_c-término y $FV(t\,u) := FV(u) \cup FV(v)$.
Si $\mathtt{k} \in \mathcal{K}$, $\mathtt{k}$ es un λ_c-término y $FV(\mathtt{k}) := \emptyset$.
Si π es una pila, k_π es un λ_c-término y $FV(k_\pi) := \emptyset$.

pilas Si $\alpha \in \Pi_0$, α es una pila.
Si t es un λ_c-término cerrado ($FV(t) = \emptyset$) y π es una pila, $t \cdot \pi$ es una pila.

Definimos las variables libres de un λ_c-término al mismo tiempo (utilizando la notación $FV(t)$), ya que necesitamos que las pilas contengan sólo términos cerrados. El conjunto de los λ_c-términos cerrados se denota Λ, mientras que el conjunto de todas las pilas se denomina Π.

A la operación $\cdot$ que agrega un λ_c-término a la cima de una pila se le dice *apilamiento*. Para simplificar la notación, las abstracciones consecutivas son reagrupadas dentro de un solo λ: $\lambda xy.\, x$ representa al término $\lambda x \lambda y.\, x$. Además, cuando una variable abstraída no es utilizada, a veces se utiliza un guión bajo en lugar de la variable (siguiendo el ejemplo, podríamos escribir dicho término como $\lambda x\_.\, x$). Por último, la aplicación tiene mayor precedencia que la abstracción, lo que significa que el término $z\, \lambda x.\, y\, x$ equivale a $z\, (\lambda x.\, (y\, x))$.

2.4.1. Sustituciones

Para realizar sustituciones de variables libres en un λ_c-término, además de las sustituciones usuales, extenderemos dicho concepto a sustituciones simultáneas con el fin de sustituir múltiples variables al mismo tiempo.

Definición 2.7 (Sustitución) *Se define la sustitución de una variable x por un término u en un término t (utilizaremos la notación $t[u/x]$) recursivamente:*

$$
\begin{array}{lcll}
x[u/x] & := & u & \\
y[u/x] & := & y & \text{donde } y \neq x \\
\mathtt{k}[u/x] & := & \mathtt{k} & \\
k_\pi[u/x] & := & k_\pi & \\
(t\, v)[u/x] & := & t[u/x]\, v[u/x] & \\
(\lambda x.\, t)[u/x] & := & \lambda x.\, t & \\
(\lambda y.\, t)[u/x] & := & \lambda y.\, t[u/x] & \text{donde } y \neq x \\
\multicolumn{4}{l}{\quad\quad \text{y se } \alpha\text{-renombra } y \text{ en } \lambda y.\, t \text{ si } y \in FV(u)}
\end{array}
$$

Definición 2.8 (Sustituciones simultáneas) *Una sustitución simultánea σ es un mapa de variables a λ_c-términos. Dada una sustitución σ, una variable x y un λ_c-término t, escribimos $\sigma, x \leftarrow t$ a la sustitución σ donde el valor correspondiente a la variable x es reemplazado por t. La aplicación de una sustitución simultánea σ a un λ_c-término t, que se escribe $t[\sigma]$, se define de la siguiente manera:*

$$
\begin{array}{lcll}
x[\sigma] & := & \sigma(x) & \\
\mathtt{k}[\sigma] & := & \mathtt{k} & \\
k_\pi[\sigma] & := & k_\pi & \\
(t\,u)[\sigma] & := & t[\sigma]\,u[\sigma] & \\
(\lambda x.\,t)[\sigma] & := & \lambda x.\,t[\sigma, x \leftarrow x] & \text{y se } \alpha\text{-renombra } x \text{ en } \lambda x.\,t \\
 & & & \text{si } x \in FV(\sigma(y)) \text{ con } y \neq x
\end{array}
$$

Decimos que una sustitución es cerrada si para toda variable x, $\sigma(x)$ es un λ_c-término cerrado. Esto es equivalente a decir que sustituir utilizando σ siempre produce un término cerrado. En caso de que σ sea cerrada, no son necesarias α-conversiones cuando se sustituye.

Las sustituciones simultáneas cerradas son utilizadas para cerrar términos, lo cual es necesario para ejecutar dichos términos en la KAM.

2.4.2. Máquina Abstracta de Krivine (KAM)

La KAM, como anticipamos, ejecuta λ_c-términos cerrados en un determinado contexto (que está dado por una pila). A este par le denominamos *proceso*:

Definición 2.9 (Procesos) *Un proceso es un par $t \star \pi$, donde t es un λ_c-término cerrado y π es una pila. El conjunto de todos los procesos es $\Lambda \star \Pi = \{t \star \pi \mid t \in \Lambda, \pi \in \Pi\}$.*

Para ejecutar procesos en la KAM se utilizan cuatro reglas de evaluación. Dichas reglas no deben ser vistas como una definición de la evaluación, sino como una axiomatización de las propiedades que debe cumplir una implementación de una KAM:

Definición 2.10 (Relaciones de evaluación) *Una relación de evaluación es cualquier relación transitiva e irreflexiva $\succ$ que satisface las siguientes propiedades:*

$$
\begin{array}{llcclcl}
\mathit{PUSH} & t\,u & \star & \pi & \succ & t & \star & u \cdot \pi \\
\mathit{GRAB} & \lambda x.\,t & \star & u \cdot \pi & \succ & t[u/x] & \star & \pi \\
\mathit{SAVE} & \mathtt{callcc} & \star & t \cdot \pi & \succ & t & \star & k_\pi \cdot \pi \\
\mathit{RESTORE} & k_{\pi'} & \star & t \cdot \pi & \succ & t & \star & \pi'
\end{array}
$$

A la clausura reflexiva de $\succ$ la notamos $\succeq$ (la clausura transitiva no es necesaria debido a que $\succ$ ya es transitiva).

Es importante observar que debido a que la relación $\succ$ es transitiva, estas reglas no son necesariamente pasos atómicos y simplemente significan que la evaluación de los procesos del lado izquierdo alcanza a los del lado derecho en una cantidad finita de pasos atómicos.

Observar que gracias a las propiedades $PUSH$ y $GRAB$ tenemos, en algún sentido, la estrategia de evaluación *whr*:

$$(\lambda x.\, t)\, u\, u_1\ \ldots u_k \star \pi \succ t[u/x] \star u_1 \cdots u_k \cdot \pi.$$

Además, de las propiedades $SAVE$ y $RESTORE$ se deduce lo siguiente:

1. $\texttt{callcc}\, t \star \pi \succ t \star k_\pi \cdot \pi$

2. $k_{\pi'}\, t \star \pi \succ t \star \pi'$

De aquí, observamos que `callcc` aplicada a un término nos permite evaluar dicho término apilando una constante de continuación de la pila actual, a dicha pila. Por otro lado, una constante de continuación aplicada a un término nos permite evaluar dicho término pero cambiando el contexto actual por la pila asociada a la constante de continuación.

Presentar la relación de evaluación como axiomas en lugar de reglas concretas nos permite mayor modularidad: cualquier relación que satisfaga dichas reglas definirá una KAM válida. En particular, esto nos va a permitir agregar libremente nuevas instrucciones con sus respectivas reglas de evaluación.

Denominaremos una *instancia del λ_c-cálculo* a una tripleta $(\mathcal{K}, \Pi_0, \succ)$. Como veremos más adelante, dicha tripleta es un parámetro de la realizabilidad clásica. Esto significa que todo resultado que probemos será válido para cualquier conjunto de instrucciones, conjunto de constantes de pila y relación de evaluación.

3. Lógica de Segundo Orden

En esta sección introduciremos un sistema de deducción para la lógica clásica de segundo orden. Las fórmulas de esta presentación de la lógica serán luego realizadas por λ_c-términos, y la similitud entre dicho sistema de deducción y un sistema de tipos del λ_c-cálculo nos permitirá obtener, para cada regla de inferencia, un programa que convierta evidencias de las premisas en evidencias de la conclusión.

Además, se definen las teorías de segundo orden y los modelos de dichas teorías (modelos de Henkin más precisamente). Particularmente, se presenta la aritmética de Peano de segundo orden (la teoría $PA2$). Buscaremos realizar dicha teoría y, análogo a lo que ocurre en la realizabilidad de Kleene, donde se utiliza un modelo de HA para definir la realizabilidad, necesitaremos de un modelo de $PA2$ para definir la realizabilidad de Krivine.

En esta sección, se complementa el material de la tesis doctoral de Lionel Rieg [1] con definiciones extraídas de [8].

3.1. Lenguajes de segundo orden

Toda teoría de segundo orden está definida a partir de un lenguaje de segundo orden, que distingue dos tipos de expresiones:

- Los términos de primer orden, que se utilizan para representar objetos de la teoría.
- Las fórmulas, que representan los enunciados de la teoría.

Formalmente, un lenguaje de segundo orden está definido a partir de un alfabeto:

Definición 3.1 (Alfabeto) *Un alfabeto $\mathcal{V}$ consta de:*

- *un conjunto de símbolos de función (notación: f, g, h, etc) dados con sus aridades.*
- *un conjunto de símbolos de predicado (notación: p, q, r, etc) dados con sus aridades.*

La *aridad* de un símbolo de función/predicado es un entero positivo o nulo que indica (de modo convencional) la cantidad de argumentos esperada por dicho símbolo. En particular, a un símbolo de función de aridad nula se le denomina *símbolo de constante.*

Sea $\mathcal{V}$ un alfabeto, se construyen los términos de primer orden y las fórmulas del correspondiente lenguaje mediante conjuntos de símbolos que denominamos *variables*. Tendremos:

- Un conjunto numerable de símbolos, cuyos elementos se llaman *variables de primer orden* (comúnmente utilizamos x,y,z para referirnos a estas).
- Para cada aridad $k \geq 0$, un conjunto numerable de símbolos, cuyos elementos se llaman *variables de segundo orden de aridad* k (generalmente utilizamos X,Y,Z para referirnos a estas).

Todos estos conjuntos son disjuntos entre sí, y disjuntos al alfabeto $\mathcal{V}$. A partir del alfabeto, y utilizando las variables de primer orden, se definen los términos de primer orden (también llamados *expresiones de primer orden* o *expresiones aritméticas*).

Expresiones $e, e_i \quad := \quad x \quad | \quad f(e_1, \ldots, e_k)$ donde x es una variable de primer orden y f un símbolo de función de aridad k

Para evitar confusiones, en general cuando hablemos de términos -a secas- nos referiremos a λ_c-términos, mientras que a los términos de primer orden les continuaremos llamando expresiones, expresiones de primer orden, o expresiones aritméticas.

A partir de las expresiones de primer orden, los predicados, las variables de segundo orden, la implicación, y los cuantificadores universales, se definen las fórmulas del lenguaje.

Fórmulas $A, B \quad := \quad p(e_1, \ldots, e_k) \quad | \quad X(e_1, \ldots, e_k) \quad | \quad A \Rightarrow B \quad | \quad \forall x.\, A \quad | \quad \forall X.\, A$ donde p es un predicado de aridad k, X es una variable de segundo orden de aridad k y x es una variable de primer orden

Siguiendo la notación usual de la lógica, escribir un cuantificador universal seguido de un punto "$\forall x.$" significa que tiene la menor precedencia. Por simplicidad, a veces utilizamos un único cuantificador universal para múltiples variables: por ejemplo, $\forall xy.\, X(x, y)$ equivale a $\forall x. \forall y.\, X(x, y)$. Más aún, al trabajar con una tupla $(x_1, \ldots, x_k)$ de variables, a veces abreviamos $\forall \vec{x}.\, A \equiv \forall x_1 \ldots \forall x_k.\, A$.

Al igual que con los λ/λ_c-términos, trabajaremos con las fórmulas a menos de α-equivalencias.

Definición 3.2 (Lenguaje de segundo orden) *Un lenguaje de segundo orden $\mathscr{L}$ consiste entonces del conjunto de expresiones de primer orden y del conjunto de fórmulas generados a partir de un alfabeto $\mathcal{V}$.*

Definición 3.3 (Variables libres) *Se define el conjunto de las variables libres de las fórmulas y de las expresiones de primer orden recursivamente (utilizaremos las notaciones $FV(A)$ y $FV(e)$ respectivamente). Diremos que una fórmula es cerrada si $FV(A) = \emptyset$, y que una expresión de primer orden es cerrada si $FV(e) = \emptyset$.*

$$\begin{array}{lcl}
FV(x) & := & \{x\} \\
FV(f(e_1, \ldots, e_k)) & := & FV(e_1) \cup \cdots \cup FV(e_k) \\
\\
FV(p(e_1, \ldots, e_k)) & := & FV(e_1) \cup \cdots \cup FV(e_k) \\
FV(X(e_1, \ldots, e_k)) & := & X \cup FV(e_1) \cup \cdots \cup FV(e_k) \\
FV(A \Rightarrow B) & := & FV(A) \cup FV(B) \\
FV(\forall x.\, A) & := & FV(A) \setminus \{x\} \\
FV(\forall X.\, A) & := & FV(A) \setminus \{X\}
\end{array}$$

Debido a que utilizamos un lenguaje minimal, con sólo la implicación y la cuantificación universal como conectivos primitivos, es necesario codificar los demás conectivos. Esto se hace a través de las codificaciones de segundo orden dadas a continuación.

Absurdo	$\bot$	$:=$	$\forall Z.\, Z$
Verdad	1	$:=$	$\forall Z.\, Z \Rightarrow Z$
Negación	$\neg A$	$:=$	$A \Rightarrow \bot$
Conjunción	$A \wedge B$	$:=$	$\forall Z.\, (A \Rightarrow B \Rightarrow Z) \Rightarrow Z$
Disjunción	$A \vee B$	$:=$	$\forall Z.\, (A \Rightarrow Z) \Rightarrow (B \Rightarrow Z) \Rightarrow Z$
Equivalencia	$A \Leftrightarrow B$	$:=$	$(A \Rightarrow B) \wedge (B \Rightarrow A)$
Existencial de primer orden	$\exists x.\, A$	$:=$	$\forall Z.\, (\forall x.\, A \Rightarrow Z) \Rightarrow Z$
Existencial de segundo orden	$\exists X.\, A$	$:=$	$\forall Z.\, (\forall X.\, A \Rightarrow Z) \Rightarrow Z$
Igualdad de Leibniz	$e = e'$	$:=$	$\forall Z.\, Z(e) \Rightarrow Z(e')$

Donde Z es una variable fresca de segundo orden. La intuición es clara: que $A \wedge B$ sea válida, por ejemplo, es equivalente a que podemos transformar cualquier demostración de Z con hipótesis A y B, en una demostración de Z.

Se puede probar que la igualdad de Leibniz es, de hecho, una relación de equivalencia. Observamos también que no se utiliza el símbolo $\top$ sino el símbolo 1, esto será más cómodo después debido a que 1 es el tipo de aquellos λ_c-términos que se comportan como la función identidad.

3.1.1. Sustituciones

Al igual que para la manipulación de λ-términos, será fundamental definir la sustitución de variables en expresiones de primer orden y en fórmulas. En este caso particular, contamos con dos sustituciones de naturaleza distinta: las variables de primer orden pueden ser sustituidas por expresiones de primer orden, mientras que las variables de segundo orden pueden ser sustituidas por *predicados*.

Definición 3.4 (Sustitución de primer orden) *Se define la sustitución de una variable x por una expresión de primer orden e en una expresión e' o en una fórmula A (utilizaremos las notaciones $e'[e/x]$ y $A[e/x]$) recursivamente:*

$$
\begin{array}{rcll}
x[e/x] & := & e & \\
y[e/x] & := & y & \text{donde } y \neq x \\
f(e_1, \dots, e_k)[e/x] & := & f(e_1[e/x], \dots, e_k[e/x]) & \\
 & & & \\
p(e_1, \dots, e_k)[e/x] & := & p(e_1[e/x], \dots, e_k[e/x]) & \\
X(e_1, \dots, e_k)[e/x] & := & X(e_1[e/x], \dots, e_k[e/x]) & \\
(A \Rightarrow B)[e/x] & := & A[e/x] \Rightarrow B[e/x] & \\
(\forall x.\, A)[e/x] & := & \forall x.\, A & \\
(\forall y.\, A)[e/x] & := & \forall y.\, A[e/x] & \text{donde } y \neq x \text{ y se} \\
 & & & \alpha\text{-renombra } y \text{ en } \forall y.\, A \\
 & & & \text{si } y \in FV(e) \\
(\forall X.\, A)[e/x] & := & \forall X.\, A[e/x] &
\end{array}
$$

Al igual que vimos en el λ_c-cálculo, es posible definir sustituciones simultáneas de primer orden como funciones del conjunto de variables de primer orden, en el conjunto de expresiones. Esto nos será de utilidad para definir la sustitución de segundo orden a través de los predicados.

Definición 3.5 (Predicado) *Un predicado $\lambda x_1 \dots x_k . A$ de aridad k es una fórmula A abstraída sobre un conjunto de k variables de primer orden $x_1, \dots, x_k$. Las variables libres de un predicado se definen como $FV(A) \setminus \{x_1, \dots, x_k\}$.*

Esta abstracción no es parte del lenguaje sino una solución *ad-hoc* para distinguir las variables de primer orden que tienen que ser sustituidas durante una sustitución de segundo orden. Si $P := \lambda x_1 \dots x_k.\, A$, definimos $P(e_1, \dots, e_k) := A[\sigma]$, donde σ es la sustitución simultánea que cumple que $\sigma(x_i) := e_i$ para todo $1 \leq i \leq k$, y $\sigma(y) := y$ para cualquier otra variable.

Definición 3.6 (Sustitución de segundo orden) *Se define la sustitución de una variable X por un predicado P en una fórmula A (utilizaremos la notación $A[P/X]$) recursivamente:*

$$\begin{array}{rcll}
p(e_1,\ldots,e_k)[P/X] & := & p(e_1,\ldots,e_k) & \\
X(e_1,\ldots,e_k)[P/X] & := & P(e_1,\ldots,e_k) & \\
Y(e_1,\ldots,e_k)[P/X] & := & Y(e_1,\ldots,e_k) & \text{donde } Y \neq X \\
(A \Rightarrow B)[P/X] & := & A[P/X] \Rightarrow B[P/X] & \\
(\forall x.\, A)[P/X] & := & \forall x.\, A[P/X] & \text{y se } \alpha\text{-renombra} \\
 & & & x \text{ en } \forall x.\, A \text{ si } x \in FV(P) \\
(\forall X.\, A)[P/X] & := & \forall X.\, A & \\
(\forall Y.\, A)[P/X] & := & \forall Y.A\,[P/X] & \text{donde } Y \neq X \text{ y se} \\
 & & & \alpha\text{-renombra } Y \text{ en } \forall Y.\, A \\
 & & & \text{si } Y \in FV(P)
\end{array}$$

3.2. Sistema de deducción

Para formalizar la noción de demostración, utilizaremos un sistema de deducción natural para esta presentación particular de la lógica de segundo orden. Empezamos primero desde la definición de *secuente*.

Definición 3.7 (Secuentes) *Se llama secuente a todo par* (Γ, A) *escrito* $\Gamma \vdash A$, *donde* Γ *es un conjunto finito de fórmulas (potencialmente vacío) y* A *es una fórmula.*

Intuitivamente, un secuente $\Gamma \vdash A$ representa una etapa particular de razonamiento, en la cual se trata de establecer A (el consecuente) a partir de las hipótesis en Γ (el contexto).

En lo que sigue, escribiremos los contextos como listas, separando sus fórmulas con comas: $\Gamma \equiv A_1, \ldots, A_n$. También utilizaremos la notación Γ, Δ para referirnos a la unión de dos contextos Γ y Δ. Además, un secuente con un contexto vacío lo escribiremos simplemente $\vdash A$ (en lugar de $\emptyset \vdash A$). Las notaciones $FV(\Gamma)$, $\Gamma[e/x]$ y $\Gamma[P/X]$ se extienden a los contextos de forma obvia: si $\Gamma \equiv A_1, \ldots, A_n$, entonces

$$\begin{array}{rcl}
FV(\Gamma) & := & FV(A_1) \cup \cdots \cup FV(A_n) \\
\Gamma[e/x] & := & A_1[e/x], \ldots, A_n[e/x] \\
\Gamma[P/X] & := & A_1[P/X], \ldots, A_n[P/X]
\end{array}$$

El sistema de deducción queda definido por las siguientes *reglas de deducción* (o *reglas de inferencia*). Cada regla de deducción es de la forma:

$$\frac{\Gamma_1 \vdash A_1 \qquad \cdots \qquad \Gamma_p \vdash A_p}{\Gamma \vdash A}$$

donde los secuentes $\Gamma_1 \vdash A_1, \ldots, \Gamma_p \vdash A_p$ se llaman las *premisas* de la regla, y el secuente $\Gamma \vdash A$ la *conclusión* de la regla. Intuitivamente, tal regla expresa un paso de deducción que permite establecer la validez de la conclusión a partir de la validez de las premisas.

Algunas reglas tienen una *condición de borde* (por la izquierda) que restringe el uso de la regla a elementos sintácticos con una forma particular.

Tendremos reglas de introducción y eliminación para los tres conectivos utilizados para definir fórmulas. Además, se utilizará la Ley de Peirce como axioma, la cual es equivalente al tercer excluido y a la reducción al absurdo. Las reglas de inferencia son las siguientes (donde A y B son fórmulas cualesquiera):

$$\frac{}{\Gamma, A \vdash A}\ \text{Axioma} \qquad \frac{}{\Gamma \vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A}\ \text{Peirce}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\ \Rightarrow_i \qquad \frac{\Gamma \vdash A \Rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ \Rightarrow_e$$

$$x \notin FV(\Gamma)\ \frac{\Gamma \vdash A}{\Gamma \vdash \forall x.\, A}\ \forall_i^1 \qquad \frac{\Gamma \vdash \forall x.A}{\Gamma \vdash A[e/x]}\ \forall_e^1$$

$$X \notin FV(\Gamma)\ \frac{\Gamma \vdash A}{\Gamma \vdash \forall X.\, A}\ \forall_i^2 \qquad \frac{\Gamma \vdash \forall X.A}{\Gamma \vdash A[P/X]}\ \forall_e^2$$

Las reglas de deducción se pueden agregar hasta formar *árboles de deducción*, o *derivaciones.*

Definición 3.8 (Derivaciones) *Llamamos derivación a todo árbol finito de secuentes construido por la aplicación finita de la siguiente regla de agregación:*

$$\textit{Si} \qquad \frac{\vdots_{d_1}}{\Gamma_1 \vdash A_1} \quad \cdots \quad \frac{\vdots_{d_p}}{\Gamma_p \vdash A_p}$$

son derivaciones de los secuentes $\Gamma_1 \vdash A_1, \ldots, \Gamma_p \vdash A_p$ *respectivamente, y además*

$$\frac{\Gamma_1 \vdash A_1 \qquad \cdots \qquad \Gamma_p \vdash A_p}{\Gamma \vdash A}$$

es una regla de deducción. Entonces

$$d \equiv \quad \frac{\dfrac{\vdots_{d_1}}{\Gamma_1 \vdash A_1} \quad \cdots \quad \dfrac{\vdots_{d_p}}{\Gamma_p \vdash A_p}}{\Gamma \vdash A}$$

es una derivación de $\Gamma \vdash A$.

Cuando existe una derivación para el secuente $\Gamma \vdash A$, decimos que dicho secuente es *derivable*. Y cuando el secuente $\vdash A$ es derivable, decimos que la fórmula A es derivable.

3.3. Teorías y modelos

A partir de un lenguaje, se puede definir entonces una teoría de segundo orden.

Definición 3.9 (Teorías) *Una teoría* $\mathscr{T}$ *de segundo orden está definida por:*

- *Un lenguaje* $\mathscr{L}$ *de segundo orden.*
- *Un conjunto de fórmulas cerradas del lenguaje* $\mathscr{L}$*, escrito* $Ax(\mathscr{T})$*, cuyos elementos se llaman los axiomas de la teoría* $\mathscr{T}$*.*

Las teorías pueden tener modelos, los cuales son interpretaciones del lenguaje que satisfacen los axiomas de la teoría. Para formalizar esto, comenzamos definiendo el concepto de *interpretación*.

Definición 3.10 (Interpretaciones) *Una interpretación* $\mathscr{M}$ *de un lenguaje de segundo orden* $\mathscr{L}$ *está constituida por:*

- *Un conjunto base* $\mathscr{M} \neq \emptyset$.
- *Una función* $f^{\mathscr{M}} : \mathscr{M}^k \to \mathscr{M}$ *para cada símbolo de función* $f \in \mathscr{L}$ *de aridad* $k \geq 0$.
- *Un conjunto de subconjuntos* $P_k^{\mathscr{M}} \subseteq \mathfrak{P}(\mathscr{M}^k)$ *para cada aridad* $k \geq 0$.
- *Un conjunto* $p^{\mathscr{M}} \in P_k^{\mathscr{M}}$ *para cada símbolo de predicado* $p \in \mathscr{L}$ *de aridad* $k \geq 0$.

A los objetos de primer orden de una interpretación los llamaremos **individuos**. Las interpretaciones permiten interpretar cualquier expresión cerrada por un individuo de su conjunto base, y a las fórmulas cerradas por algún valor de verdad (*verdadero* o *falso*). Para poder lidiar con las fórmulas y expresiones no cerradas, introducimos el concepto de *valuación*.

Definición 3.11 (Valuaciones) *Sea* $\mathscr{M}$ *una interpretación de un lenguaje de segundo orden, una valuación* ρ *es una función que le asocia valores en* $\mathscr{M}$ *a las variables de primer orden del lenguaje, y funciones características de los conjuntos en* $P_k^{\mathscr{M}}$ *a las variables de segundo orden (preservando la aridad). Denotamos* $\rho, x \leftarrow v$ *a la valuación* ρ *donde el valor asociado a* x *es reemplazado por* v.

En otras palabras, $\rho(x) \in \mathscr{M}$ para toda variable x de primer orden, y $\rho(X) = \mathcal{X}_C$ con $C \in P_k^{\mathscr{M}}$ para toda variable X de segundo orden y aridad k.

Podemos pensar que a cada variable de segundo orden X de aridad k le estamos asignando un conjunto $C \in P_k^{\mathscr{M}}$ (recordar que $C \subseteq \mathscr{M}^k$). Luego, si $m_1, \ldots, m_k \in \mathscr{M}$ tenemos que $\rho(X)(m_1, \ldots, m_k) = 1$ si, y sólo si, $(m_1, \ldots, m_k) \in C$. Pensaremos al valor 1 como la verdad y al 0 como la falsedad.

Las valuaciones nos permiten interpretar expresiones y fórmulas al poder lidiar con aquellas que no son necesariamente cerradas.

Definición 3.12 (Interpretación de expresiones) *Sea e una expresión de un lenguaje de segundo orden, $\mathscr{M}$ una interpretación de dicho lenguaje y ρ una valuación, definimos la interpretación de dicha expresión (utilizamos la notación $[\![e]\!]_\rho$) recursivamente:*

$$\begin{array}{rcl} [\![x]\!]_\rho & := & \rho(x) \\ [\![f(e_1, \ldots, e_k)]\!]_\rho & := & f^{\mathscr{M}}([\![e_1]\!]_\rho, \ldots, [\![e_k]\!]_\rho) \end{array}$$

Básicamente, utilizamos una valuación para poder interpretar también expresiones que no sean cerradas. De hecho, cuando una expresión aritmética es cerrada, su interpretación no depende de la valuación y quitamos el subíndice ρ.

La importancia de poder interpretar expresiones y fórmulas cualesquiera, se debe a que las subfórmulas de una fórmula cerrada no necesariamente son cerradas y, entonces, para poder definir recursivamente la semántica se precisa una definición en la clase de todas las fórmulas.

Definición 3.13 (Interpretación de fórmulas) *Sea A una fórmula de un lenguaje de segundo orden, $\mathscr{M}$ una interpretación de dicho lenguaje y ρ una valuación, definimos la interpretación de dicha fórmula (utilizamos la notación $[\![A]\!]_\rho$) recursivamente:*

$$\begin{array}{rcl} [\![p(e_1, \ldots, e_k)]\!]_\rho & := & \mathcal{X}_{(p^{\mathscr{M}})}([\![e_1]\!]_\rho, \ldots, [\![e_k]\!]_\rho) \\ [\![X(e_1, \ldots, e_k)]\!]_\rho & := & \rho(X)([\![e_1]\!]_\rho, \ldots, [\![e_k]\!]_\rho) \\ [\![A \Rightarrow B]\!]_\rho & := & \text{máx}\{[\![B]\!]_\rho, 1 - [\![A]\!]_\rho\} \\ [\![\forall x.\, A]\!]_\rho & := & \text{mín}_{a \in \mathscr{M}}\, [\![A]\!]_{\rho, x \leftarrow a} \\ [\![\forall X.\, A]\!]_\rho & := & \text{mín}_{C \in P_k^{\mathscr{M}}}\, [\![A]\!]_{\rho, X \leftarrow \mathcal{X}_C} \end{array}$$

Se puede probar que si una fórmula es cerrada, su interpretación no depende de la valuación, por lo que podemos quitar el subíndice ρ. Particularmente, cuando $[\![A]\!] = 1$ para una fórmula A cerrada, decimos que $\mathscr{M}$ satisface A y lo notamos

$\mathscr{M} \vDash A$.

Por convención, cuando $k = 0$, entendemos que $\mathscr{M}^k = \{()\}$ (existe una tupla de largo 0), por lo que $\mathfrak{P}(\mathscr{M}^k) = \{\emptyset, \{()\}\}$. Por lo tanto, si X tiene aridad cero, y tomamos una valuación ρ tal que $\rho(X) = \mathcal{X}_{\{()\}}$, se cumple que $[\![X]\!]_\rho = 1$, mientras que si $\rho(X) = \mathcal{X}_\emptyset$, tenemos que $[\![X]\!]_\rho = 0$. Básicamente, a las variables de segundo orden y aridad cero, se les puede asignar el valor *verdadero* (el conjunto $\{()\}$), o el valor *falso* (el conjunto $\emptyset$). Esto aplica de la misma manera para símbolos de predicado de aridad nula.

En base a las definiciones anteriores, definimos un modelo de una teoría como una interpretación de su lenguaje que satisface sus axiomas, y el *esquema de comprensión*.

Definición 3.14 (Modelos) *Sea $\mathscr{T}$ una teoría de segundo orden, denominamos modelo (o modelo de Henkin) a toda interpretación $\mathscr{M}$ del lenguaje de $\mathscr{T}$ tal que:*

- *$\mathscr{M} \vDash A$ para toda fórmula $A \in Ax(\mathscr{T})$.*
- *$\mathscr{M} \vDash \exists X.\, \forall \vec{x}.\, A \Leftrightarrow X(x_1, \ldots, x_k)$ para toda fórmula A tal que $FV(A) \subseteq \{x_1, \ldots, x_k\}$.*

En dicho caso, escribimos $\mathscr{M} \vDash \mathscr{T}$.

Cuando $P_k^{\mathscr{M}} = \mathfrak{P}(\mathscr{M}^k)$ para todo k, la interpretación satisface el axioma de comprensión. En este caso, si además se satisfacen los axiomas de la teoría, decimos que $\mathscr{M}$ es un modelo *pleno* de $\mathscr{T}$.

3.4. Aritmética de Peano (PA2)

Definiremos la teoría de la aritmética de Peano en segundo orden, la cual utilizaremos fuertemente en la realizabilidad clásica. Para esta teoría, el alfabeto a partir del cual se define el lenguaje de segundo orden consta del símbolo de constante 0, y un símbolo de función para cada *función recursiva primitiva* (veremos más adelante el porqué de esta decisión).

Definición 3.15 (Funciones recursivas primitivas) *Las funciones recursivas primitivas se definen inductivamente:*

1. *La función sucesor es una función recursiva primitiva.*
2. *Para todo k natural, la función $zero_k$ (de aridad k) definida como $zero_k(x_1, \ldots, x_k) := 0$, es una función recursiva primitiva.*

3. *Para todo k e i naturales tales que $1 \leq i \leq k$, la función Π_i^k (de aridad k) definida como $\Pi_i^k(x_1, \ldots, x_k) := x_i$ es una función recursiva primitiva.*

4. *Sea g una función recursiva primitiva de aridad m, y $h_1, \ldots, h_m$ m funciones recursivas primitivas de aridad k, la función f (de aridad k) definida como $f(x_1, \ldots, x_k) = g(h_1(x_1, \ldots, x_k), \ldots, h_m(x_1, \ldots, x_k))$ es una función recursiva primitiva.*

5. *Sea g una función recursiva primitiva de aridad k, y h una función recursiva primitiva de aridad $k+2$, la función f (de aridad $k+1$) definida como $f(n+1, x_1, \ldots, x_k) := h(n, f(n, x_1, \ldots, x_k), x_1, \ldots, x_k)$ y $f(0, x_1, \ldots, x_k) := g(x_1, \ldots, x_k)$ es una función recursiva primitiva.*

Observamos que las primeras tres reglas conforman los “casos base”, mientras que las últimas dos nos permiten construir nuevas funciones a través de la composición, y de la recursión primitiva.

Debido a que la función sucesor, la suma y el producto son funciones recursivas primitivas, tenemos símbolos para dichas funciones (s, $+$ y $*$ respectivamente). Por comodidad, escribiremos $s\,x$ en lugar de $s(x)$, $x + y$ en lugar de $+(x, y)$ y $x * y$ en lugar de $*(x, y)$. Además, evitamos el uso de paréntesis cuando el significado no sea ambiguo.

Utilizando este lenguaje, definimos la teoría $PA2$ al establecer sus tres axiomas principales. Además de estos, se deben agregar axiomas con las definiciones de las funciones recursivas primitivas:

Sucesor	$\forall xy.\, s\,x = s\,y \Rightarrow x = y$
	$\forall x. \neg(s\,x = 0)$
Inducción	$\forall x. \forall Z. Z(0) \Rightarrow (\forall y. Z(y) \Rightarrow Z(s\,y)) \Rightarrow Z(x)$
f **recursiva primitiva distinta a** s	$\forall \vec{x}.\, f(\ldots) = \ldots$

A diferencia de lo que sucede en la teoría de primer orden, observamos que el axioma de inducción es un único axioma y no un esquema de axiomas. Además, en el último caso, estamos englobando lo siguiente:

1. Para las funciones $zero_k$ tenemos axiomas de la forma:

$$\forall x_1 \ldots x_k.\, zero_k(x_1, \ldots, x_k) = 0$$

2. Para las funciones Π_i^k tenemos axiomas de la forma:

$$\forall x_1 \dots x_k.\, \Pi_i^k(x_1, \dots, x_k) = x_i$$

3. Para toda función recursiva primitiva f que sea una composición de una función recursiva primitiva g de aridad m y m funciones recursivas primitivas h_i de aridad k, tenemos un axioma de la forma:

$$\forall x_1 \dots x_k.\, f(x_1, \dots, x_k) = g(h_1(x_1, \dots, x_k), \dots, h_m(x_1, \dots, x_k))$$

4. Para toda función recursiva primitiva f definida a partir de una recursión con caso base g y paso inductivo h, contamos con dos axiomas:

$$\forall x_1 \dots x_k.\, f(0, x_1, \dots, x_k) = g(x_1, \dots, x_k) \text{ y}$$
$$\forall y x_1 \dots x_k.\, f(s\, y, x_1, \dots, x_k) = h(y, f(y, x_1, \dots, x_k), x_1, \dots, x_k)$$

4. Introducción a la Realizabilidad Clásica

De la misma manera en que mostramos cómo el sistema de tipos simples de Church para el λ-cálculo se correspondía con un sistema de deducción de lógica de primer orden intuicionista, podemos (y lo veremos en profundidad en la próxima sección) definir un sistema de tipos simples para el λ_c-cálculo donde las reglas de tipado se correspondan con las reglas del sistema de deducción de segundo orden visto previamente.

Debido a dicha correspondencia (que no es más que la correspondencia Curry-Howard), hacer una derivación de una fórmula A es equivalente a probar que un término t tiene tipo A. En algún sentido, t sostiene/justifica la fórmula A: si el conjunto de términos de tipo A es nulo, A no es derivable, mientras que si no es nulo, A es derivable.

La realizabilidad clásica se puede ver como una flexibilización de dicho sistema de tipos que también nos permite vincular λ_c-términos con fórmulas de segundo orden. Decimos que es más flexible ya que veremos que si t tiene tipo A, entonces t *realiza* A. Esto es equivalente a afirmar que los *realizadores* de A contienen aquellos programas de tipo A.

Para fijar ideas y ver algunos ejemplos, introduciremos primero la realizabilidad a través de lo que denominaremos *modelos de realizabilidad*, y en la siguiente sección veremos el sistema de tipos subyacente. Los modelos de realizabilidad son básicamente interpretaciones de un lenguaje de segundo orden que difieren de las de Henkin, debido a que la interpretación de una fórmula es el conjunto de términos que la realizan (ya que, de cierta manera, sostienen la fórmula) en lugar de un valor *verdadero* o *falso*. Hablamos en plural debido a que obtendremos distintos modelos al variar distintos parámetros de los cuales depende la realizabilidad.

4.1. Parámetros

Los modelos de realizabilidad no se enfocan en si una fórmula es demostrable o no, sino en el contenido computacional asociado a ella. De esta forma, el valor de verdad de una fórmula es el conjunto de programas que la realizan. La realizabilidad clásica no es la excepción, aunque el valor de verdad es derivado de la noción primitiva de valor de falsedad. Como evidencian sus nombres, tanto el valor de falsedad como el de verdad son de naturaleza muy diferente: el valor de verdad de una fórmula es un conjunto de realizadores (por lo tanto, un conjunto de λ_c-términos) mientras que el valor de falsedad de una fórmula es un conjunto de pilas.

La idea detrás es la siguiente: cada fórmula tiene asociado un conjunto de pilas (a través de su valor de falsedad) que son *tests* (les diremos tests en lugar de pruebas para evitar confusiones con el concepto de demostración) que un programa debe "pasar" para poder realizar dicha fórmula. Para definir qué es "pasar" un test, introduciremos el concepto de *polo*.

Los modelos de realizabilidad clásica están parametrizados por un polo $\perp\!\!\!\perp$. Un polo es un conjunto de procesos ($\perp\!\!\!\perp \subseteq \Lambda \star \Pi$) el cual debe ser cerrado por *anti-evaluación*: esto quiere decir que si $p \succ p'$ y $p' \in \perp\!\!\!\perp$ entonces $p \in \perp\!\!\!\perp$. Desde un punto de vista computacional, un polo representa una propiedad que queremos observar. El concepto de anti-evaluación implica que nos interesamos en los procesos que exhiben dicha propiedad computacional en algún momento de su evaluación. Por ejemplo, para caracterizar los procesos que ejecutan una instrucción `k` (podría ser `print`) denotamos $\perp\!\!\!\perp := \{p \in \Lambda \star \Pi \mid \exists \pi \in \Pi.\, p \succeq \mathtt{k} \star \pi\}$. Notamos que una vez que un proceso en $\perp\!\!\!\perp$ alcanza $\mathtt{k} \star \pi$, al continuar la evaluación puede no volver a dicho estado debido a que ya exhibió la propiedad computacional en la que estábamos interesados.

Definición 4.1 (Polos) *Un polo $\perp\!\!\!\perp$ es un subconjunto de $\Lambda \star \Pi$ tal que si $p' \in \perp\!\!\!\perp$ y $p \succ p'$, entonces $p \in \perp\!\!\!\perp$.*

La idea será entonces la siguiente: dada una fórmula A con un conjunto de tests asociados, un programa realiza dicha fórmula si pasa todos esos tests. Para pasar dichos tests, para cada test π el proceso $t \star \pi$ debe pertenecer al polo. Por anti-evaluación, en general nos bastará ver que el proceso $t \star \pi$, al ser evaluado, alcanza un proceso que pertenece al polo.

Los modelos de realizabilidad dependen entonces de tres parámetros:

- Una instancia $(\mathcal{K}, \Pi_0, \succ)$ del λ_c-cálculo.
- Un modelo $\mathscr{M}$ de $PA2$ el cual denominaremos *modelo de partida*.
- Un polo $\perp\!\!\!\perp$.

4.2. Modelos de realizabilidad

Al fijar los parámetros anteriores, podremos interpretar fórmulas del lenguaje de segundo orden que definimos para la teoría $PA2$ que será de nuestro interés.

Para interpretar una fórmula al estilo Henkin, necesitamos un conjunto base $\mathscr{M}$ para interpretar los objetos de primer orden, una función $f^{\mathscr{M}}$ para cada símbolo de función f del lenguaje, y conjuntos $P_k^{\mathscr{M}}$ que definen las funciones características que pueden ser utilizadas para interpretar los objetos de segundo orden.

En este caso, tomaremos el modelo de partida $\mathscr{M}$ para definir el universo de primer orden, pero en lugar de utilizar funciones características para interpretar los objetos de segundo orden, utilizaremos funciones de falsedad.

Definición 4.2 (Funciones de falsedad) *Una función de falsedad de aridad k, es una función con dominio $\mathscr{M}^k$ y codominio $\mathfrak{P}(\Pi)$.*

Luego, dado que las expresiones de primer orden y las fórmulas que interpretamos pueden no ser cerradas, introdujimos el concepto de *valuación* para lidiar con ello. Con el mismo objetivo, se definen las valuaciones en el contexto de realizabilidad. A partir de ahora, a las primeras las denominaremos *valuaciones booleanas*, mientras que a las segundas las denominaremos valuaciones a secas.

Definición 4.3 (Valuaciones) *Una valuación es una función ρ que le asocia valores en $\mathscr{M}$ a las variables de primer orden del lenguaje, y funciones de falsedad a las variables de segundo orden (preservando la aridad). Denotamos $\rho, x \leftarrow v$ a la valuación ρ donde el valor asociado a x es reemplazado por v.*

Utilizando una valuación ρ, podemos interpretar expresiones aritméticas al interpretar sus variables libres utilizando ρ e interpretando los símbolos de función por su correspondiente función recursiva primitiva.

Definición 4.4 (Interpretación de expresiones) *Sea ρ una valuación y e una expresión del lenguaje de $PA2$, definimos la interpretación de dicha expresión en el modelo de realizabilidad (utilizamos la notación $[\![e]\!]_\rho$) recursivamente:*

$$\begin{array}{rcl} [\![x]\!]_\rho & := & \rho(x) \\ [\![f(e_1, \ldots, e_k)]\!]_\rho & := & f^{\mathscr{M}}([\![e_1]\!]_\rho, \ldots, [\![e_k]\!]_\rho) \end{array}$$

Básicamente, estamos interpretando las expresiones de primer orden de la misma manera en que lo hace el modelo de partida. En general, haremos un abuso de notación y utilizaremos f, a secas, para referirnos también a $f^{\mathscr{M}}$. Por otro lado, cuando una expresión aritmética es cerrada, su interpretación no depende de la valuación y quitamos el subíndice ρ.

Además, utilizando dicha valuación ρ se definen, a partir de recursión mutua, los valores de falsedad y los valores de verdad de las fórmulas. Observamos que dejamos de lado las interpretaciones de Henkin donde la interpretación de una fórmula tiene sólo dos valores posibles (*verdadero* y *falso*), para obtener una interpretación más rica en contenido.

Definición 4.5 (Interpretación de fórmulas) *Sea ρ una valuación, $\perp\!\!\!\perp$ un polo y A una fórmula del lenguaje de $PA2$, definimos los valores de verdad y falsedad de dicha fórmula en el modelo de realizabilidad (utilizamos las notaciones $|A|$ y $||A||$ respectivamente) a partir de recursión mutua:*

$$
\begin{array}{rcl}
||X(e_1,\ldots,e_k)||_{\rho,\perp\!\!\!\perp} & := & \rho(X)([\![e_1]\!]_\rho,\ldots,[\![e_k]\!]_\rho) \\
||A \Rightarrow B||_{\rho,\perp\!\!\!\perp} & := & |A|_{\rho,\perp\!\!\!\perp} \cdot ||B||_{\rho,\perp\!\!\!\perp} \\
 & := & \{t \cdot \pi \mid t \in |A|_{\rho,\perp\!\!\!\perp}, \pi \in ||B||_{\rho,\perp\!\!\!\perp}\} \\
||\forall x.\, A||_{\rho,\perp\!\!\!\perp} & := & \bigcup_{n \in \mathscr{M}} ||A||_{(\rho, x \leftarrow n),\perp\!\!\!\perp} \\
||\forall X.\, A||_{\rho,\perp\!\!\!\perp} & := & \bigcup_{F : \mathscr{M}^k \to \mathfrak{P}(\Pi)} ||A||_{(\rho, X \leftarrow F),\perp\!\!\!\perp} \\
 & & \\
|A|_{\rho,\perp\!\!\!\perp} & := & \{t \in \Lambda \mid \forall \pi \in ||A||_{\rho,\perp\!\!\!\perp}, t \star \pi \in \perp\!\!\!\perp\}
\end{array}
$$

Observemos cómo se crean los valores de verdad y de falsedad:

- Como ya se mencionó, el valor de verdad de una fórmula es el conjunto de términos cerrados que "pasan todos los tests". Donde los tests son las pilas del valor de falsedad de dicha fórmula.
- Los tests de $\forall x.\, A$ y de $\forall X.\, A$ son todos los tests posibles de A, sustituyendo x por cualquier individuo del modelo de partida o X por cualquier función de falsedad.
- Los tests de $A \Rightarrow B$ se construyen apilando realizadores de A a los tests de B. Básicamente: para realizar $A \Rightarrow B$ hay que crear un programa que, recibiendo cualquier realizador de A, pueda pasar los tests de B.
- Los tests de $X(e_1,\ldots,e_k)$ se obtienen interpretando X (se transforma en una función de falsedad) y evaluando dicha función al interpretar las expresiones aritméticas e_i.

Mientras los términos en el valor de verdad (que serán los realizadores) sostienen una fórmula, los tests en el valor de falsedad intentan refutarla. Por eso, por ejemplo, un test (una refutación) de $A \Rightarrow B$ consiste de un término que sostiene A, y un test que refuta B.

Para una fórmula cerrada, el valor de verdad y el de falsedad no dependen de la valuación ρ por lo que simplemente omitimos dicho subíndice.

Observación 4.1 *$\perp\!\!\!\perp$ es una relación de ortogonalidad. Si se define $t \perp\!\!\!\perp \pi := t \star \pi \in \perp\!\!\!\perp$, entonces el valor de verdad $|A|_{\rho,\perp\!\!\!\perp}$ es el ortogonal al valor de falsedad $||A||_{\rho,\perp\!\!\!\perp}$, que se escribe $||A||_\rho^{\perp\!\!\!\perp}$.*
En particular, por la teoría de las relaciones ortogonales, tenemos:

- *Si* $||A||_{\rho,\perp\!\!\!\perp} \subseteq ||B||_{\rho,\perp\!\!\!\perp}$*, entonces* $|B|_{\rho,\perp\!\!\!\perp} \subseteq |A|_{\rho,\perp\!\!\!\perp}$.
- $||A||_{\rho,\perp\!\!\!\perp} \subseteq (||A||_{\rho}^{\perp\!\!\!\perp})^{\perp\!\!\!\perp}$
- $||A||_{\rho}^{\perp\!\!\!\perp} = (((||A||_{\rho}^{\perp\!\!\!\perp})^{\perp\!\!\!\perp})^{\perp\!\!\!\perp}$ *o, en otras palabras,* $|A|_{\rho,\perp\!\!\!\perp} = (|A|_{\rho}^{\perp\!\!\!\perp})^{\perp\!\!\!\perp}$

4.3. Lenguaje extendido

Por conveniencia, normalmente extendemos el lenguaje para introducir parámetros. Básicamente, agregamos un símbolo de constante $\dot{n}$ para cada $n \in \mathscr{M}$, y un símbolo de predicado $\dot{F}$ de aridad k para cada función de falsedad F con dicha aridad. Observamos que esto hace que las fórmulas no sean numerables (recordemos que cada F es una función $\mathscr{M}^k \to \mathfrak{P}(\Pi)$).

Definición 4.6 (Lenguaje extendido) *Definimos el lenguaje extendido a partir del modelo de realizabilidad al redefinir las expresiones y fórmulas de la siguiente manera:*

$$\textbf{Expresiones} \quad e, e_i \quad := \quad x \quad | \quad f(e_1, \ldots, e_k) \quad | \quad \dot{n}$$

donde x es una variable de primer orden, f un símbolo de una función recursiva primitiva de aridad k y $n \in \mathscr{M}$.

$$\begin{array}{llllll} \textbf{Fórmulas} & A, B & := & X(e_1, \ldots, e_k) & | \ A \Rightarrow B & | \ \forall x.\, A \\ & & | & \forall X.\, A & | \ \dot{F}(e_1, \ldots, e_k) & \end{array}$$

donde X es una variable de segundo orden de aridad k, x es una variable de primer orden y $F : \mathscr{M}^k \to \mathfrak{P}(\Pi)$ es una función de falsedad de aridad k.

De esta forma, podemos referirnos a predicados específicos en la sintaxis y asegurarnos su adecuada interpretación en el modelo. Para esto, extendemos la interpretación de las expresiones y de las fórmulas:

$$\begin{array}{rcl} [\![\dot{n}]\!]_\rho & := & n \\ ||\dot{F}(e_1, \ldots, e_k)||_{\rho,\perp\!\!\!\perp} & := & F([\![e_1]\!]_\rho, \ldots, [\![e_k]\!]_\rho) \end{array}$$

Los parámetros también nos permiten reescribir el valor de falsedad del cuantificador universal de segundo orden, de la siguiente manera:

$$||\forall X.\, A||_{\rho,\perp\!\!\!\perp} \quad := \quad \bigcup_{F:\mathscr{M}^k \to \mathfrak{P}(\Pi)} ||A[\dot{F}/X]||_{\rho,\perp\!\!\!\perp}$$

Cuando el valor de falsedad de un parámetro de aridad cero es un *singleton*, omitimos las llaves por comodidad y escribimos $\dot{\pi}$ en lugar de $\dot{\{\pi\}}$. Tenemos entonces

$||\dot{\pi}||_{\perp\!\!\!\perp} := \{\pi\}$. Por ejemplo, un parámetro importante es $\top := \dot{\emptyset}$. Dicho parámetro hace referencia a la función de falsedad que no recibe argumentos, y retorna un conjunto nulo de pilas.

Observamos que $|\top|_{\perp\!\!\!\perp} = \{t \in \Lambda \mid \forall \pi \in \emptyset, t \star \pi \in \perp\!\!\!\perp\} = \Lambda$. En otras palabras, todo término cerrado es un realizador de $\top$ debido a que no tiene tests asociados (y esto no depende del polo).

Utilizando parámetros, podemos utilizar una valuación ρ para transformar una fórmula/expresión aritmética, en otra fórmula/expresión aritmética **cerrada**, cuya interpretación (única, ya que es cerrada) coincida con la interpretación de la original (utilizando ρ).

Definición 4.7 (Clausura de fórmulas por una valuación) *Sea ρ una valuación, e una expresión y A una fórmula, definimos la clausura de e por ρ y de A por ρ (utilizaremos las notaciones $e[\rho]$ y $A[\rho]$ respectivamente) recursivamente:*

$$\begin{array}{rcll}
x[\rho] & := & \dot{n} & \text{con } n = \rho(x) \\
\dot{n}[\rho] & := & \dot{n} & \\
f(e_1, \ldots, e_k)[\rho] & := & f(e_1[\rho], \ldots, e_k[\rho]) & \\
\\
(X(e_1, \ldots, e_k))[\rho] & := & \dot{P}(e_1[\rho], \ldots, e_k[\rho]) & \text{con } P = \rho(X) \\
(\dot{F}(e_1, \ldots, e_k))[\rho] & := & \dot{F}(e_1[\rho], \ldots, e_k[\rho]) & \\
(A \Rightarrow B)[\rho] & := & A[\rho] \Rightarrow B[\rho] & \\
(\forall x.\, A)[\rho] & := & \forall x.\, A[\rho, x \leftarrow x] & \\
(\forall X.\, A)[\rho] & := & \forall X.\, A[\rho, X \leftarrow X] &
\end{array}$$

Observar que utilizamos, informalmente, $\rho, x \leftarrow x$ y $\rho, X \leftarrow X$ para notar que a estas variables no se les aplicará ρ. Esto se podría formalizar extendiendo el codominio de una valuación, para permitir variables de primer y segundo orden, o, en la clausura, trabajar con valuaciones parciales.

Se puede verificar que las fórmulas y expresiones aritméticas resultantes son siempre cerradas y que $|A[\rho]|_{\perp\!\!\!\perp} = |A|_{\rho,\perp\!\!\!\perp}$, $||A[\rho]||_{\perp\!\!\!\perp} = ||A||_{\rho,\perp\!\!\!\perp}$ y $[\![e[\rho]]\!] = [\![e]\!]_\rho$. En base a esto, se pueden ver dos igualdades que serán útiles:

1. $$\begin{array}{rcl}
||(\forall x.\, A)[\rho]||_{\perp\!\!\!\perp} & = & ||\forall x.\, A||_{\rho,\perp\!\!\!\perp} \\
& = & \bigcup_{n \in \mathscr{M}} ||A||_{(\rho, x \leftarrow n),\perp\!\!\!\perp} \\
& = & \bigcup_{n \in \mathscr{M}} ||A[\rho, x \leftarrow n]||_{\perp\!\!\!\perp}
\end{array}$$

2. $$\begin{array}{rcl}
||(\forall X.\, A)[\rho]||_{\perp\!\!\!\perp} & = & ||\forall X.\, A||_{\rho,\perp\!\!\!\perp} \\
& = & \bigcup_{F : \mathscr{M}^k \to \mathfrak{P}(\Pi)} ||A||_{(\rho, X \leftarrow F),\perp\!\!\!\perp} \\
& = & \bigcup_{F : \mathscr{M}^k \to \mathfrak{P}(\Pi)} ||A[\rho, X \leftarrow F]||_{\perp\!\!\!\perp}
\end{array}$$

En general, cuando nos referimos a fórmulas, por ejemplo en la definición anterior, estamos hablando de fórmulas en el lenguaje extendido. Cuando sea necesario, aclararemos que estamos trabajando con fórmulas sin parámetros.

4.4. Realizadores y primeros resultados

En general, vamos a trabajar con un polo fijo y, por lo tanto, no utilizaremos dicho subíndice en los valores de verdad y en los valores de falsedad. Las siguientes definiciones y demostraciones son, a menos que se indique lo contrario, para un polo fijo.

Definición 4.8 (Realizadores) *Sea t un λ_c-término y A una fórmula cerrada, decimos que t realiza A (que se escribe $t \Vdash A$) si $t \in |A|$. Cuando especificar el polo sea necesario, escribiremos $t \Vdash_{\perp\!\!\!\perp} A$. Cuando t realiza A para todo polo, diremos que t es un realizador universal de A y lo escribiremos $t \Vvdash A$.*

Proposición 4.1

1. *Los tests asociados a la fórmula de verdad $1 \equiv \forall Z.\, Z \Rightarrow Z$, están compuestos por un término y una pila que juntos están en el polo: $||1|| = \{t \cdot \pi \mid t \star \pi \in \perp\!\!\!\perp\}$.*
2. *La función identidad realiza universalmente a la fórmula de verdad: $\lambda x.\, x \Vvdash 1$. Veremos más adelante (en la sección 9) que todos los realizadores universales de 1 se comportan como la identidad, y es por dicha razón que utilizamos el símbolo 1 en lugar de $\top$ para referirnos a ella.*

Demostración.

1. $||1|| = ||\forall Z.Z \Rightarrow Z|| = \bigcup_{F:\mathfrak{P}(\Pi)} ||\dot{F} \Rightarrow \dot{F}|| = \bigcup_{F:\mathfrak{P}(\Pi)} ||\dot{F}||^{\perp\!\!\!\perp} \cdot ||\dot{F}|| = \bigcup_{G \subseteq \Pi} G^{\perp\!\!\!\perp} \cdot G = \{t \cdot \pi \mid t \star \pi \in \perp\!\!\!\perp\}$.
2. Sea $\perp\!\!\!\perp$ un polo cualquiera, hay que ver que $t \Vdash_{\perp\!\!\!\perp} 1$. Tomando un test $t \cdot \pi \in ||1||_{\perp\!\!\!\perp}$, basta ver que $\lambda x.x$ pasa dicho test, esto es: $\lambda x.x \star t \cdot \pi \in \perp\!\!\!\perp$. Vemos que $\lambda x.\, x \star t \cdot \pi \succ t \star \pi$ y, como $t \star \pi \in \perp\!\!\!\perp$ por la parte anterior, por anti-evaluación concluimos que $\lambda x.\, x \star t \cdot \pi \in \perp\!\!\!\perp$. □

Proposición 4.2

1. *$||\perp|| = \Pi$. Esto quiere decir que para realizar $\perp$ hay que pasar todos los tests posibles.*
2. *$t \Vdash \perp$ sii $t \Vdash B$ para toda fórmula B cerrada. Básicamente, un realizador de $\perp$ es un realizador de cualquier fórmula cerrada.*
3. *Dada una fórmula cerrada A, $t \Vdash \neg A$ sii $t \Vdash A \Rightarrow B$ para toda fórmula B cerrada.*

Demostración.

1. Recordando que definimos $\perp \equiv \forall Z.\, Z$, podemos obtener el valor de falsedad de $\perp$. $||\perp|| = ||\forall Z.\, Z|| = \bigcup_{F:\mathfrak{P}(\Pi)} ||\dot{F}|| = \bigcup_{G\subseteq\Pi} G = \Pi$.

2. La vuelta es directa ya que $\perp$ es una fórmula cerrada. Para la ida, sea B una fórmula cerrada cualquiera, hay que probar que $t \Vdash B$. Para esto, se toma $\pi_B \in ||B||$ y se demuestra que $t \star \pi_B \in \perp\!\!\!\perp$ (t pasa cualquier test de B). Esto último es trivial a partir del punto 1: $\pi_B \in ||\perp||$ y por hipótesis $t \Vdash \perp$, entonces $t \star \pi_B \in \perp\!\!\!\perp$.

3. Para la vuelta basta recordar que $\neg A \equiv A \Rightarrow \perp$, y tomar $B \equiv \perp$. Para la ida, sea B una fórmula cerrada cualquiera, hay que probar que $t \star \pi_{A\Rightarrow B} \in \perp\!\!\!\perp$ para toda $\pi_{A\Rightarrow B} \in ||A \Rightarrow B||$ (t pasa cualquier test de $A \Rightarrow B$). Recordando que $\pi_{A\Rightarrow B} \equiv t_A \cdot \pi_B$ con t_A un realizador de A y π_B un test de B, hay que probar que $t \star t_A \cdot \pi_B \in \perp\!\!\!\perp$. Esto es idéntico a la parte anterior: $t_A \cdot \pi_B \in ||A \Rightarrow \perp||$ y, por hipótesis, t pasa dichos test. □

Si bien parece difícil encontrar realizadores para $\perp$ ya que, como acabamos de ver, dichos realizadores realizan cualquier fórmula, vamos a ver que podemos obtenerlos fácilmente.

Proposición 4.3 (Realizadores de $\perp$) *Siempre y cuando el polo no sea vacío, hay realizadores para la fórmula $\perp$ (y, por lo tanto, para cualquier fórmula cerrada).*

Demostración. Sea $t \star \pi$ un proceso en $\perp\!\!\!\perp$, entonces $k_\pi\, t$ es un realizador de $\perp$. Basta ver que para cualquier pila π', $k_\pi\, t \star \pi' \succ t \star \pi \in \perp\!\!\!\perp$ entonces, por anti-evaluación, tenemos que $k_\pi\, t \star \pi' \in \perp\!\!\!\perp$. □

Básicamente, lo que hacemos es utilizar la constante de continuación para cambiar el test que recibimos (cambiamos la pila). De todos modos, esta estrategia nos sirve para un polo fijo pero no hay realizadores universales de $\perp$.

Proposición 4.4 *$\perp$ no es realizable universalmente.*

Demostración. Supongamos que existe $t \in \Lambda$ tal que $t \Vvdash \perp$. Si tomamos el polo $\perp\!\!\!\perp := \emptyset$, entonces $t \Vdash_{\perp\!\!\!\perp} \perp$. Como $||\perp||_{\perp\!\!\!\perp} = \Pi$, concluimos que $t \star \pi \in \perp\!\!\!\perp$ para toda pila π, lo cual es absurdo ya que el polo es vacío. □

La idea anterior de utilizar una constante de evaluación para cambiar el test, nos sirve también para encontrar realizadores de la implicancia entre dos fórmulas. En realidad, como encontramos realizadores de $\perp$, ya conocemos realizadores de $A \Rightarrow B$ (o de cualquier fórmula), siempre que el polo no sea vacío. Aún así, los siguientes realizadores, que son también dependientes del polo, nos permitirán demostrar que `callcc` es un realizador **universal** de la Ley de Peirce.

Proposición 4.5 (Realizador universal de la Ley de Peirce) *Para todo par de fórmulas cerradas A y B, se cumple que:*

1. *si $\pi \in ||A||$, entonces $k_\pi \Vdash A \Rightarrow B$.*
2. *$\texttt{callcc} \Vvdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A$.*

Demostración.

1. Sea t un λ_c-término que realiza A, $\pi_A \in ||A||$ y $\pi_B \in ||B||$, queremos probar que $k_{\pi_A} \star t \cdot \pi_B \in \perp\!\!\!\perp$. Como $k_{\pi_A} \star t \cdot \pi_B \succ t \star \pi_A$ y $t \star \pi_A \in \perp\!\!\!\perp$ porque t es un realizador de A, se concluye por anti-evaluación lo que buscábamos. Como se mencionó anteriormente, nos aprovechamos de las constantes de continuación: recibimos un realizador de A y una test de B, y cambiamos el test para ejecutar el realizador de A contra un test de A.

2. Necesitamos probar que `callcc` pasa los tests de la Ley de Peirce para cualquier polo. Dichos tests están formados por un realizador de $(A \Rightarrow B) \Rightarrow A$ en la cima y luego un test de A. Fijado un polo $\perp\!\!\!\perp$, sean $t \Vdash_{\perp\!\!\!\perp} (A \Rightarrow B) \Rightarrow A$ y $\pi_A \in ||A||_{\perp\!\!\!\perp}$ entonces $\texttt{callcc} \star t \cdot \pi_A \succ t \star k_{\pi_A} \cdot \pi_A$.
   Por la parte anterior, $k_{\pi_A} \cdot \pi_A \in ||(A \Rightarrow B) \Rightarrow A||_{\perp\!\!\!\perp}$, por lo que $t \star k_{\pi_A} \cdot \pi_A \in \perp\!\!\!\perp$ y, por anti-evaluación, $\texttt{callcc} \star t \cdot \pi_A \in \perp\!\!\!\perp$. Como tomamos un polo y un $t \cdot \pi_A \in ||((A \Rightarrow B) \Rightarrow A) \Rightarrow A||_{\perp\!\!\!\perp}$ cualquiera, concluimos que `callcc` es un realizador universal de $((A \Rightarrow B) \Rightarrow A) \Rightarrow A$. □

Del punto 1, como se puede tomar cualquier B, observamos que k_π es un realizador de $\neg A$ si $\pi \in ||A||$. Además, es interesante ver cómo dichas constantes de continuación son un conjunto representativo de los realizadores de $\neg A$. ¿A qué nos referimos con representativo? Bueno, básicamente a que si tenemos un término t que pasa tests formados por una constante de continuación apilada sobre un test de B, podemos obtener un término que pasa tests formados por **cualquier** realizador de $\neg A$ apilado sobre un test de B (un test de $\neg A \Rightarrow B$). Formalmente:

Proposición 4.6 *Sean A y B fórmulas cerradas y t un λ_c-término tal que para cualquier par de pilas $\pi_A \in ||A||$ y $\pi_B \in ||B||$, $t \star k_{\pi_A} \cdot \pi_B \in \perp\!\!\!\perp$, entonces definiendo*

$$M_\perp := \lambda xy.\, \texttt{callcc}\,(\lambda k.\, y\,(\texttt{callcc}\,(\lambda k'.\, k\,(x\,k'))))$$

tenemos que $M_\perp\, t \Vdash \neg A \Rightarrow B$.

Demostración. La idea es la siguiente: hay que construir, a partir de t, un realizador $(M_\perp\, t)$ de $\neg A \Rightarrow B$. Basta entonces con que $M_\perp\, t$ reciba un realizador de $\neg A$ y retorne un realizador de B. Lo que hace $M_\perp\, t$ es lo siguiente: al realizador de $\neg A$ (que denominaremos u), le pasa un realizador de A, y como resultado obtiene un realizador de $\perp$ (en particular, un realizador de B).

Formalmente, sea t como en el enunciado, $u \Vdash \neg A$ y $\pi_B \in ||B||$, queremos probar que $M_\perp\, t \star u \cdot \pi_B \in \perp\!\!\!\perp$.

Veamos la siguiente secuencia de reducción:

$$\begin{array}{rcl}
M_\perp\, t \star u \cdot \pi_B & \equiv & \lambda xy.\, \texttt{callcc}\, (\lambda k.\, y\, (\texttt{callcc}\, (\lambda k'.\, k\, (x\, k'))))\, t \star u \cdot \pi_B \\
& \succ & \lambda xy.\, \texttt{callcc}\, (\lambda k.\, y\, (\texttt{callcc}\, (\lambda k'.\, k\, (x\, k')))) \star t \cdot u \cdot \pi_B \\
& \succ & \lambda y.\, \texttt{callcc}\, (\lambda k.\, y\, (\texttt{callcc}\, (\lambda k'.\, k\, (t\, k')))) \star u \cdot \pi_B \\
& \succ & \texttt{callcc}\, (\lambda k.\, u\, (\texttt{callcc}\, (\lambda k'.\, k\, (t\, k')))) \star \pi_B \\
& \succ & \lambda k.\, u\, (\texttt{callcc}\, (\lambda k'.\, k\, (t\, k'))) \star k_{\pi_B} \cdot \pi_B \\
& \succ & u\, (\texttt{callcc}\, (\lambda k'.\, k_{\pi_B}\, (t\, k'))) \star \pi_B \\
& \succ & u \star \texttt{callcc}\, (\lambda k'.\, k_{\pi_B}\, (t\, k')) \cdot \pi_B
\end{array}$$

Observar que, como $u \Vdash \neg A$ y $\pi_b \in ||\perp||$, basta ver que $\texttt{callcc}(\lambda k'.k_{\pi_B}(tk')) \Vdash A$ para concluir $M_\perp\, t \Vdash \neg A \Rightarrow B$ por anti-evaluación.

Esta última parte es la interesante de la prueba: $M_\perp\, t$ recibe u y un test para B y, como ya mencionamos, crea un realizador de A y se lo pasa a u para obtener un realizador de $\perp$ y poder pasar cualquier test. Para crear dicho realizador de A puede utilizar t y π_B. Lo que hace el realizador creado $(\texttt{callcc}\,(\lambda k'.k_{\pi_B}\,(t\,k')))$ es, al recibir un test de A, utilizarlo para obtener una constante k_{π_A} y, al apilarla al π_B que ya tenía, armar un test que pueda ser pasado por t para llegar al polo. Formalmente, sea $\pi_A \in ||A||$

$$\begin{array}{rcl}
\texttt{callcc}\, (\lambda k'.\, k_{\pi_B}\, (t\, k')) \star \pi_A & \succ & \lambda k'.\, k_{\pi_B}(t\, k') \star k_{\pi_A} \cdot \pi_A \\
& \succ & k_{\pi_B}\, (t\, k_{\pi_A}) \star \pi_A \\
& \succ & k_{\pi_B} \star (t\, k_{\pi_A}) \cdot \pi_A \\
& \succ & t\, k_{\pi_A} \star \pi_B \\
& \succ & t \star k_{\pi_A} \cdot \pi_B
\end{array}$$

Esto último pertenece al polo debido a la hipótesis de t y, por anti-evaluación, concluimos que $\texttt{callcc}\, (\lambda k'.\, k_{\pi_B}\, (t\, k')) \Vdash A$. $\square$

4.5. Subtipos

Con el fin de facilitar algunas demostraciones, es conveniente introducir una relación de subtipado $\leq_{\perp\!\!\!\perp}$ entre fórmulas.

Definición 4.9 (Subtipos) *Sea $\perp\!\!\!\perp$ un polo, A y B fórmulas cualesquiera, decimos que $A \leq_{\perp\!\!\!\perp} B$ si para toda valuación ρ, $||B||_{\rho,\perp\!\!\!\perp} \subseteq ||A||_{\rho,\perp\!\!\!\perp}$.*

En general, escribiremos sólo $\leq$ salvo que sea necesario indicar el polo. Observar que $A \leq_{\perp\!\!\!\perp} B$ si A es "más falso" que B (contiene más tests y, por lo tanto, es más difícil de realizar). En particular, vemos que en esas condiciones $|A|_{\rho,\perp\!\!\!\perp} \subseteq |B|_{\rho,\perp\!\!\!\perp}$ para toda valuación ρ. Más aún, si tomamos fórmulas cerradas tenemos que los realizadores de A son realizadores de B.

Observación 4.2 *Sean A y B fórmulas cerradas:*

1. *Si $A \leq_{\perp\!\!\!\perp} B$ y $t \Vdash_{\perp\!\!\!\perp} A$, entonces $t \Vdash_{\perp\!\!\!\perp} B$.*
2. *Si $A \leq_{\perp\!\!\!\perp} B$ para todo polo y $t \Vvdash A$, entonces $t \Vvdash B$.*

Proposición 4.7 *Para cualquier polo, la relación $\leq_{\perp\!\!\!\perp}$ satisface las reglas de subtipos dadas a continuación. Además, debido a las dos primeras reglas, es una relación de preorden en las fórmulas y su equivalencia se escribe $\approx_{\perp\!\!\!\perp}$.*

$$\frac{}{A \leq A} \qquad \frac{A \leq B \qquad B \leq C}{A \leq C}$$

$$\frac{A \leq B \qquad C \leq D}{B \Rightarrow C \leq A \Rightarrow D} \qquad \frac{}{A \Rightarrow \top \approx \top}$$

$$\frac{A \leq B}{\forall x.\, A \leq \forall x.\, B} \qquad x \notin FV(A)\ \frac{A \leq B}{A \leq \forall x.\, B}$$

$$\frac{A \leq B}{\forall X.\, A \leq \forall X.\, B} \qquad X \notin FV(A)\ \frac{A \leq B}{A \leq \forall X.\, B}$$

$$\frac{}{\forall x.\, A \leq A[e/x]} \qquad \frac{}{\forall X.\, A \leq A[P/X]}$$

$$\frac{}{A \leq \top} \qquad \frac{}{\bot \leq A}$$

$$x \notin FV(A)\ \frac{}{\forall x.\, A \Rightarrow B \approx A \Rightarrow \forall x.\, B} \qquad X \notin FV(A)\ \frac{}{\forall X.\, A \Rightarrow B \approx A \Rightarrow \forall X.\, B}$$

Podemos ver, por ejemplo, que `callcc` $\Vvdash \forall X.\, ((X \Rightarrow \bot) \Rightarrow \bot) \Rightarrow X$. Para esto asumiremos que `callcc` $\Vvdash \forall X.\, ((X \Rightarrow \bot) \Rightarrow X) \Rightarrow X$ (la prueba es análoga a la ya dada para probar que `callcc` realiza la Ley de Peirce).

Utilizando la observación 4.2, bastará ver que $\forall X.\, ((X \Rightarrow \bot) \Rightarrow X) \Rightarrow X \leq_{\perp\!\!\!\perp} \forall X.\, ((X \Rightarrow \bot) \Rightarrow \bot) \Rightarrow X$ para todo polo. Esto último se puede obtener fácilmente utilizando las reglas presentadas anteriormente:

$$\cfrac{\cfrac{\cfrac{\cfrac{}{X \Rightarrow \bot \leq X \Rightarrow \bot} \qquad \cfrac{}{\bot \leq X}}{(X \Rightarrow \bot) \Rightarrow \bot \leq (X \Rightarrow \bot) \Rightarrow X} \qquad \cfrac{}{X \leq X}}{((X \Rightarrow \bot) \Rightarrow X) \Rightarrow X \leq ((X \Rightarrow \bot) \Rightarrow \bot) \Rightarrow X}}{\forall X.\, ((X \Rightarrow \bot) \Rightarrow X) \Rightarrow X \leq \forall X.\, ((X \Rightarrow \bot) \Rightarrow \bot) \Rightarrow X}$$

5. Realizabilidad como un Sistema de Tipos

Como se sugirió al principio de la sección anterior, la realizabilidad se puede ver como una flexibilización de un sistema de tipos cuyas reglas se corresponden con las reglas de deducción del sistema de segundo orden introducido anteriormente. En esta sección, presentaremos dicho sistema de tipos al presentar sus reglas: para esto bastará extender el sistema de deducción con el fin de agregar *realizadores explícitos.*

5.1. Extendiendo el sistema de deducción

Análogo a como introdujimos el sistema de deducción en 3.2, empezaremos actualizando la definición de secuente: en esta extensión del sistema de deducción, un secuente es escrito $\Gamma \vdash t : A$ donde t es un λ_c-término, A es una fórmula y Γ es un *contexto.*

A partir de ahora, los contextos serán conjuntos de pares conformados por una variable del λ_c-cálculo y una fórmula. Para diferenciar entre los dos tipos de variables que manejamos, a las variables del λ_c-cálculo les diremos *variables de prueba*, mientras que a las variables del lenguaje de segundo orden las llamaremos *variables de tipo.*

Definición 5.1 (Contextos) *Un contexto es un conjunto finito (potencialmente vacío) de pares (x, A) escritos $x : A$, donde x es una variable de prueba, A es una fórmula y las variables de prueba no están asociadas a más de una fórmula.*

A dichos contextos los escribiremos como listas, separando los pares por comas: $\Gamma \equiv x_1 : A_1, \ldots, x_n : A_n$. Al igual que antes, utilizaremos la notación Γ, Δ para referirnos a la unión de dos contextos Γ y Δ (siempre y cuando dicha unión sea también un contexto). En base a los contextos, podemos definir los secuentes.

Definición 5.2 (Secuentes) *Se denomina secuente a toda tripleta (Γ, t, A) escrita $\Gamma \vdash t : A$, donde Γ es un contexto, t un λ_c-término y A una fórmula.*

Siguiendo las convenciones que habíamos visto, a un secuente con un contexto vacío lo escribimos simplemente $\vdash t : A$ (en lugar de $\emptyset \vdash t : A$). Además, las notaciones $FV(\Gamma)$, $\Gamma[e/x]$ y $\Gamma[P/X]$ se extienden a los contextos de forma obvia: si $\Gamma \equiv x_1 : A_1, \ldots, x_n : A_n$, entonces

$$\begin{array}{rcl} FV(\Gamma) & := & FV(A_1) \cup \cdots \cup FV(A_n) \\ \Gamma[e/x] & := & x_1 : A_1[e/x], \ldots, x_n : A_n[e/x] \\ \Gamma[P/X] & := & x_1 : A_1[P/X], \ldots, x_n : A_n[P/X] \end{array}$$

Observamos que esto es idéntico a la definición que ya habíamos dado. Las variables libres **no** incluyen las variables de prueba y las sustituciones son definidas sobre las variables de tipo.

Un contexto se dice cerrado cuando todas las fórmulas que contiene son cerradas (no tiene variables de tipo libres). Además, el conjunto de variables de prueba declaradas en un contexto Γ es llamado el dominio de Γ y se escribe $dom\,\Gamma$.

En base a la nueva definición de secuente, las reglas de deducción quedan:

$$\frac{}{\Gamma, x : A \vdash x : A}\ \text{Axioma} \qquad \frac{}{\Gamma \vdash \mathtt{callcc} : ((A \Rightarrow B) \Rightarrow A) \Rightarrow A}\ \text{Peirce}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \Rightarrow B}\ \Rightarrow_i \qquad \frac{\Gamma \vdash t : A \Rightarrow B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}\ \Rightarrow_e$$

$$x \notin FV(\Gamma)\ \frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall x.A}\ \forall_i^1 \qquad \frac{\Gamma \vdash t : \forall x.A}{\Gamma \vdash t : A[e/x]}\ \forall_e^1$$

$$X \notin FV(\Gamma)\ \frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall X.A}\ \forall_i^2 \qquad \frac{\Gamma \vdash t : \forall X.A}{\Gamma \vdash t : A[P/X]}\ \forall_e^2$$

Intuitivamente, si $x_1 : A_1, \ldots, x_n : A_n \vdash t : A$ es derivable, además de interpretar que A es "verdad" asumiendo $A_1, \ldots, A_n$, vamos a tener que t realiza A si x_i realiza A_i para todo $1 \leq i \leq n$.

Lo primero que observamos es que los realizadores que vamos a extraer de este sistema no utilizan constantes de continuación. En base a esto, definimos dicha clase de términos y luego probamos la afirmación.

Definición 5.3 (Términos de prueba) *Se define el conjunto de términos de prueba PL, como aquellos λ_c-términos que no contienen constantes de continuación.*

Proposición 5.1 *Si $\Gamma \vdash t : A$ es derivable, entonces $t \in PL$.*

Demostración. La demostración es directa ya que ninguna regla introduce constantes de continuación. $\square$

Es importante notar que PL proviene de *Proof Like* (a veces se denota QP por *Quasi Proof*), y significa "casi prueba" o de "tipo prueba". Cualquier término sin constantes de continuación pertenece a PL, aún cuando no exista una derivación de $\Gamma \vdash t : A$ para dicho término t.

Otra propiedad simple pero importante es que las variables libres de los términos están en el dominio del contexto.

Proposición 5.2 *Si* $\Gamma \vdash t : A$ *es derivable, entonces* $FV(t) \subseteq \mathit{dom}\,\Gamma$.

Demostración. Se prueba por inducción en las reglas de deducción:

- **Axioma.** Cualquier derivación que finalice con dicha regla cumple que $t \equiv x$ y que $\Gamma \equiv \Delta, x : A$, de donde $FV(t) = \{x\} \subseteq \mathit{dom}\,(\Delta, x : A) = \mathit{dom}\,\Gamma$.
- **Peirce.** Este caso es trivial ya que $FV(\texttt{callcc}) = \emptyset$.
- **Introducción del $\Rightarrow$.** Al aplicar esta regla en última instancia, se deriva el secuente $\Gamma \vdash \lambda x.\, u : B \Rightarrow C$ a partir del secuente $\Gamma, x : B \vdash u : C$.
  Por hipótesis inductiva, sabemos que $FV(u) \subseteq \mathit{dom}\,(\Gamma, x : B) = \mathit{dom}\,\Gamma \cup \{x\}$.
  Concluimos que $FV(\lambda x.\, u) = FV(u) \backslash \{x\} \subseteq (\mathit{dom}\,\Gamma \cup \{x\}) \backslash \{x\} \subseteq \mathit{dom}\,\Gamma$.
- **Eliminación del $\Rightarrow$.** En este caso, obtenemos que $t \equiv u\ v$. Por hipótesis inductiva sabemos que $FV(u) \subseteq \mathit{dom}\,\Gamma$ y que $FV(v) \subseteq \mathit{dom}\,\Gamma$. Por lo tanto, $FV(t) = FV(u) \cup FV(v) \subseteq \mathit{dom}\,\Gamma$.
- **Introducción y eliminación del $\forall^1$ y del $\forall^2$.** En los cuatro casos tenemos que $FV(t) \subseteq \mathit{dom}\,\Gamma$ por hipótesis inductiva. □

Notemos que podemos modificar fácilmente una derivación del sistema de deducción sin realizadores explícitos para obtener una derivación en el nuevo sistema (y viceversa). Para esto, basta asignarle una variable de prueba distinta a cada fórmula que aparezca como hipótesis y luego utilizar las reglas de deducción para ir obteniendo realizadores explícitos.

Teorema 5.1 *Si* $\Gamma \vdash A$ *es derivable en el sistema de deducción presentado en la sección 3.2, entonces existe una derivación en el sistema presentado previamente para un secuente* $\Gamma' \vdash t : A$ *donde* Γ' *contiene las mismas fórmulas que* Γ.

Demostración. Primero, le asignamos a cada fórmula A que aparezca en la derivación una única variable x_A de tipo. Luego, probamos el enunciado por inducción en las reglas de deducción (del sistema sin realizadores explícitos):

- **Axioma.** Si el secuente $\Gamma \vdash A$ es derivable utilizando esta regla, entonces $\Gamma \equiv B_1, \ldots, B_n, A$. Luego, tomando $\Gamma' \equiv x_{B_1} : B_1, \ldots, x_{B_n} : B_n, x_A : A$, podemos derivar $\Gamma' \vdash x_A : A$.
- **Peirce.** En este caso basta tomar Γ' como para el caso anterior, y observar que el secuente $\Gamma' \vdash \texttt{callcc} : ((A \Rightarrow B) \Rightarrow A) \Rightarrow A$ es derivable.

- **Introducción del $\Rightarrow$.** Al aplicar esta regla se deriva el secuente $\Gamma \vdash B \Rightarrow C$ a partir de un secuente derivable $\Gamma, B \vdash C$. Por hipótesis inductiva sabemos que existe un secuente $\Gamma', x_B : B \vdash u : C$ que es derivable y tal que Γ' contiene las mismas fórmulas que Γ (pero asociadas a las variables de tipo que le corresponden). Aplicando la regla de introducción del $\Rightarrow$ (en el sistema con realizadores explícitos), obtenemos que $\Gamma' \vdash \lambda x_B.\, u : B \Rightarrow C$ es derivable.

- **Eliminación del $\Rightarrow$.** Utilizando esta regla, podemos derivar $\Gamma \vdash A$ a partir de derivaciones de $\Gamma \vdash B \Rightarrow A$ y de $\Gamma \vdash B$.
  Por hipótesis inductiva, existen Γ', Γ'', u y v tales que $\Gamma' \vdash u : B \Rightarrow A$ y $\Gamma'' \vdash v : B$ son derivables. Además, por construcción, $\Gamma' \equiv \Gamma''$ (ambos tienen las mismas fórmulas que Γ, y las variables de tipo asignadas a cada fórmula siguen el mapeo que hicimos al principio de la prueba).
  A partir de lo anterior, podemos derivar $\Gamma' \vdash u\, v : A$.

- **Introducción y eliminación del $\forall^1$ y del $\forall^2$.** En los cuatro casos es trivial al tomar Γ' como lo venimos haciendo, y utilizando el mismo realizador que tenemos por hipótesis (es interesante observar que las restricciones para ambas reglas de introducción no afectan ya que $FV(\Gamma) = FV(\Gamma')$). $\square$

Recíprocamente, podemos ver que si $\Gamma' \vdash t : A$ es derivable en el sistema de deducción con realizadores explícitos, entonces $\Gamma \vdash A$ es derivable en el otro sistema, al tomar Γ con las mismas fórmulas que Γ' (basta remover los realizadores y las variables de tipo en los contextos).

Teorema 5.2 *Si $\Gamma' \vdash t : A$ es derivable en el sistema de deducción con realizadores explícitos, entonces existe una derivación en el sistema presentado en 3.2 para un secuente $\Gamma \vdash A$ donde Γ contiene las mismas fórmulas que Γ'.*

Como anticipamos, a esta extensión del sistema de deducción la podemos ver como un sistema de tipos para el λ_c-cálculo en donde t tiene tipo A si $\vdash t : A$ es derivable. Particularmente, en base a los teoremas anteriores, vemos que derivar una fórmula es equivalente a tiparla. Dicha equivalencia es la esencia de la correspondencia Curry-Howard: que una fórmula sea derivable es equivalente a que exista un programa con dicha fórmula como tipo. Por lo tanto, los términos de tipo A de alguna manera soportan/evidencian la fórmula.

Cuando introdujimos la realizabilidad, comentamos que es una flexibilización de este sistema de tipos. Encontrar realizadores de una fórmula es “más sencillo” que derivarla. Veremos en el Teorema de Adecuación que si t tiene tipo A, entonces t realiza A.

5.2. Teorema de Adecuación

Una vez extendido el sistema de deducción, es necesario probar que efectivamente los términos explícitos de las demostraciones realizan las fórmulas asociadas a ellos. Este resultado, de suma importancia, se conoce como el Teorema de Adecuación, aunque para probarlo utilizaremos un resultado más fuerte conocido como el Lema de Adecuación. Lo que veremos es básicamente que la realizabilidad es más flexible que el sistema de tipos anterior.

Empezamos distinguiendo aquellas sustituciones simultáneas que convierten a las variables del dominio de un contexto en realizadores de sus fórmulas asociadas.

Definición 5.4 *Dado un polo $\perp\!\!\!\perp$, decimos que una sustitución cerrada σ realiza un contexto cerrado Γ, escrito $\sigma \Vdash_{\perp\!\!\!\perp} \Gamma$, cuando $\sigma(x) \Vdash_{\perp\!\!\!\perp} A$ para todo $(x : A) \in \Gamma$. Si $\sigma(x) \Vvdash A$ para todo $(x : A) \in \Gamma$, notamos $\sigma \Vvdash \Gamma$.*

Como siempre, en general notamos $\sigma \Vdash \Gamma$ salvo cuando se necesite explicitar el polo. En base a esto, podemos formalizar la idea del teorema de adecuación: si podemos realizar un contexto, vamos a poder realizar la fórmula consecuente.

Teorema 5.3 (Teorema de Adecuación) *Si el secuente $\Gamma \vdash t : A$ es derivable en el sistema introducido, donde Γ y A son cerrados, entonces para cualquier polo $\perp\!\!\!\perp$ y cualquier sustitución cerrada σ tal que $\sigma \Vdash_{\perp\!\!\!\perp} \Gamma$, tenemos que $t[\sigma] \Vdash_{\perp\!\!\!\perp} A$. Más aún, si tomamos una sustitución cerrada σ tal que $\sigma \Vvdash \Gamma$, entonces $t[\sigma] \Vvdash A$.*

Básicamente, el teorema nos dice que si un secuente $\Gamma \vdash t : A$ es derivable y cambiamos las variables libres de t por realizadores de la fórmula a la cual están asociadas en el contexto, entonces obtendremos un realizador de A.

Recordemos que para que un término realice una fórmula, pedíamos que ambos fueran cerrados. Es por esto que solicitamos que todas las fórmulas sean cerradas y que la sustitución también lo sea (esto último implica que $t[\sigma]$ sea cerrado).

Corolario 5.1 *Si el secuente $\vdash t : A$ es derivable y A es cerrado, entonces $t \Vvdash A$.*

Demostración. Basta tomar una sustitución cerrada cualquiera, llamémosle σ, luego $\sigma \Vvdash \emptyset$ y, por el teorema de adecuación, $t[\sigma] \Vvdash A$. Como t es cerrado (ya que $FV(t) \subseteq dom\,\emptyset = \emptyset$), $t[\sigma] \equiv t$ y tenemos lo que buscábamos. □

Para la prueba del teorema, como en un árbol de derivación no siempre estamos trabajando con fórmulas cerradas, demostramos una afirmación más fuerte conocida como el Lema de Adecuación.

Lema 5.1 (Lema de Adecuación) *Si el secuente $\Gamma \vdash t : A$ es derivable en el sistema introducido, donde Γ y A no tienen por qué ser cerrados, entonces para cualquier polo $\perp\!\!\!\perp$, cualquier valuación ρ y cualquier sustitución cerrada σ tal que $\sigma \Vdash_{\perp\!\!\!\perp} \Gamma[\rho]$, tenemos que $t[\sigma] \Vdash_{\perp\!\!\!\perp} A[\rho]$.*

La notación $\Gamma[\rho]$ la utilizamos para referirnos a un contexto análogo a Γ, donde convertimos cada $(x : B) \in \Gamma$ en $x : B[\rho]$. Es importante recordar que $B[\rho]$ es cerrada para toda fórmula B y toda valuación ρ. Usando este lema, la demostración del teorema es trivial debido a que $A[\rho] \equiv A$ y $\Gamma[\rho] \equiv \Gamma$ cuando A y Γ son cerrados.

5.2.1. Demostración del Lema de Adecuación

Para demostrar el lema de adecuación, vamos a realizar una prueba modularizada.

Definición 5.5 *Un secuente $\Gamma \vdash t : A$ se dice adecuado con respecto a un polo $\perp\!\!\!\perp$ si para toda valuación ρ y toda sustitución cerrada σ, tal que $\sigma \Vdash_{\perp\!\!\!\perp} \Gamma[\rho]$, se cumple que $t[\sigma] \Vdash_{\perp\!\!\!\perp} A[\rho]$.*

Observemos que para probar el lema, tendríamos que probar que un secuente derivable es siempre adecuado con respecto a cualquier polo.

Definición 5.6 *Una regla de inferencia se dice adecuada con respecto a un polo $\perp\!\!\!\perp$, si siempre que las premisas sean secuentes adecuados con respecto a $\perp\!\!\!\perp$, la conclusión es un secuente adecuado con respecto a $\perp\!\!\!\perp$.*

En base a estas definiciones, cualquier secuente derivado a partir de reglas de inferencia adecuadas con respecto a un polo $\perp\!\!\!\perp$, es adecuado con respecto a dicho polo. Por lo tanto, basta demostrar que toda regla de inferencia es adecuada (con respecto a cualquier polo) para probar el lema de adecuación.

Teorema 5.4 *Todas las reglas del sistema de deducción presentado son adecuadas con respecto a cualquier polo.*

Demostración. En todos los casos trabajamos con un polo $\perp\!\!\!\perp$ cualquiera, aunque no lo explicitamos en las notaciones $\Vdash$ y $||A||$.

- **Axioma.** Esta regla no tiene premisas, por lo que basta demostrar que la conclusión es adecuada. La conclusión, en este caso, es un secuente de la forma $\Gamma, x : A \vdash x : A$. Sean ρ y σ tales que $\sigma \Vdash (\Gamma, x : A)[\rho]$, tenemos en particular que $x[\sigma] \equiv \sigma(x) \Vdash A[\rho]$.

- **Ley de Peirce.** De la misma manera que para la regla anterior, hay que demostrar que la conclusión es adecuada. Como `callcc` es cerrado, la sustitución no hará ningún efecto, por lo que basta probar que `callcc` $\Vdash (((A \Rightarrow B) \Rightarrow A) \Rightarrow A)[\rho]$ con A, B y ρ cualesquiera.

  Esto es trivial ya que $(((A \Rightarrow B) \Rightarrow A) \Rightarrow A)[\rho] \equiv ((A[\rho] \Rightarrow B[\rho]) \Rightarrow A[\rho]) \Rightarrow A[\rho]$, $A[\rho]$ y $B[\rho]$ son cerradas, y habíamos probado que `callcc` realizaba este tipo de fórmulas en la proposición 4.5.

- **Introducción del $\Rightarrow$.** Supongamos que el secuente $\Gamma, x : A \vdash t : B$ es adecuado. Sean ρ, σ tales que $\sigma \Vdash \Gamma[\rho]$, queremos probar que $(\lambda x.\, t)[\sigma] \Vdash (A \Rightarrow B)[\rho] \equiv A[\rho] \Rightarrow B[\rho]$.

  Sean $u \Vdash A[\rho]$ y $\pi_B \in ||B[\rho]||$ $(u \cdot \pi_B \in ||(A \Rightarrow B)[\rho]||)$, hay que probar que $(\lambda x. t)[\sigma] \star u \cdot \pi_B \in \perp\!\!\!\perp$. Ahora, como $\sigma \Vdash \Gamma[\rho]$ y x sólo puede aparecer como $(x : A)$ en Γ (ya que $\Gamma, x : A$ es un contexto), tenemos que $\sigma, x \leftarrow u \Vdash \Gamma[\rho], x : A[\rho]$ (i.e. $\sigma, x \leftarrow u \Vdash (\Gamma, x : A)[\rho]$). Como $\Gamma, x : A \vdash t : B$ es adecuado por hipótesis, podemos afirmar que $t[\sigma, x \leftarrow u] \Vdash B[\rho]$ y, por lo tanto, $t[\sigma, x \leftarrow u] \star \pi_B \in \perp\!\!\!\perp$.

  Finalmente, como σ es cerrado, $(\lambda x.\, t)[\sigma] \star u \cdot \pi_B \equiv \lambda x.\, t[\sigma, x \leftarrow x] \star u \cdot \pi_B \succ (t[\sigma, x \leftarrow x])[u/x] \star \pi_B \equiv t[\sigma, x \leftarrow u] \star \pi_B \in \perp\!\!\!\perp$, y se concluye que $(\lambda x.\, t)[\sigma] \star u \cdot \pi_B \in \perp\!\!\!\perp$ por anti-evaluación.

- **Eliminación del $\Rightarrow$.** Asumimos que los secuentes $\Gamma \vdash t : A \Rightarrow B$ y $\Gamma \vdash u : A$ son adecuados. Sean ρ y σ tales que $\sigma \Vdash \Gamma[\rho]$. Queremos probar que $(t\, u)[\sigma] \Vdash B[\rho]$ o, dicho de otra forma, que $t[\sigma]\, u[\sigma] \Vdash B[\rho]$.

  Sea $\pi_B \in ||B[\rho]||$, por hipótesis tenemos que $t[\sigma] \Vdash (A \Rightarrow B)[\rho]$ y que $u[\sigma] \Vdash A[\rho]$, entonces $u[\sigma] \cdot \pi_B \in ||(A \Rightarrow B)[\rho]||$. Luego, $t[\sigma] u[\sigma] \star \pi_B \succ t[\sigma] \star u[\sigma] \cdot \pi_B \in \perp\!\!\!\perp$ y concluimos por anti-evaluación.

- **Introducción del $\forall^1$.** Asumimos que el secuente $\Gamma \vdash t : A$ es adecuado. Sea x una variable de primer orden que no pertenece a $FV(\Gamma)$, ρ y σ tales que $\sigma \Vdash \Gamma[\rho]$. Queremos probar que $t[\sigma] \Vdash (\forall x.\, A)[\rho]$, en otras palabras, que para todo $n \in \mathscr{M}$, $t[\sigma] \Vdash A[\rho, x \leftarrow n]$.

  Basta ver que, como $x \notin FV(\Gamma)$, $\sigma \Vdash \Gamma[\rho, x \leftarrow n]$ y por hipótesis podemos concluir que $t[\sigma] \Vdash A[\rho, x \leftarrow n]$.

- **Eliminación del $\forall^1$.** En este caso, queremos probar que el secuente $\Gamma \vdash t : A[e/x]$ es adecuado, asumiendo que $\Gamma \vdash t : \forall x.\, A$ lo es. Sean ρ y σ tales que $\sigma \Vdash \Gamma[\rho]$, necesitamos ver que $t[\sigma] \Vdash A[e/x][\rho]$.

  Sea $\pi_A \in ||A[e/x][\rho]|| = ||A[e/x]||_\rho = ||A||_{\rho, x \leftarrow [\![e]\!]_\rho} = ||A[\rho, x \leftarrow [\![e]\!]_\rho]||$, tenemos que $\pi_A \in \bigcup_{n \in \mathscr{M}} ||A[\rho, x \leftarrow n]|| = ||(\forall x. A)[\rho]||$. Luego, por hipótesis, $t[\sigma] \star \pi_A \in \perp\!\!\!\perp$ y concluimos lo que buscábamos.

- **Introducción del** $\forall^2$**.** Análogo a la introducción del $\forall^1$.
- **Eliminación del** $\forall^2$**.** Análogo a la eliminación del $\forall^1$. $\square$

Observación 5.1 *El hecho de que las reglas sean adecuadas individualmente es útil también para construir realizadores:*

- *De la introducción del* $\Rightarrow$ *podemos ver que si* $t[u/x] \Vdash B$ *para todo realizador* u *de* A*, entonces* $\lambda x.\, t \Vdash A \Rightarrow B$.
- *Observamos también, de la eliminación del* $\Rightarrow$*, que si* $t \Vdash A \Rightarrow B$ *y* $u \Vdash A$*, luego* $t\, u \Vdash B$.
- *A partir de la eliminación del* $\forall$*, vemos que* $t \Vdash \forall x.\, A$ *sii* $t \Vdash A[e/x]$ *para cualquier* e *cerrada (y similar para* $\forall X.\, A$*).*

El siguiente corolario se desprende de ello:

Corolario 5.2 *Sea* A *una fórmula cerrada, si* A *es realizable universalmente entonces* $\neg A$ *no lo es.*

Demostración. Supongamos que existen $t, u \in \Lambda$ tales que $t \Vvdash A$ y $u \Vvdash \neg A \equiv A \Rightarrow \bot$. Luego, $u\, t \Vvdash \bot$ lo cual es absurdo ya que $\bot$ no tiene realizadores universales. $\square$

6. Modelos de Realizabilidad

En la sección anterior, primero vimos que tenemos un sistema de tipado para el λ_c-cálculo que es equivalente a un sistema de deducción de lógica clásica de segundo orden. Luego, vimos que la realizabilidad es una definición más débil que el tipado. La pregunta que surge naturalmente es qué nos aporta la realizabilidad que no nos aporta el tipado. En otras palabras, qué diferencia a las fórmulas realizables de las tipables/derivables.

En esta sección estudiaremos la relación entre la realizabilidad de una fórmula y su validez. Veremos, por ejemplo, la siguiente propiedad que se cumple para cualquier fórmula A cerrada:

Derivable	$\Rightarrow$	**Realizable universalmente**	$\Rightarrow$	**Verdadera en el modelo de partida**
$\vdash t : A$ con $t \in PL \cap \Lambda$		$t \Vdash A$ con $t \in PL \cap \Lambda$		$\mathscr{M} \models A$

La primera implicancia ya la vimos a través del corolario 5.1, mientras que la segunda será vista en esta sección. De hecho, veremos luego ejemplos de que los recíprocos no son válidos.

En el resto de la sección, estudiaremos algunos ejemplos típicos de modelos de realizabilidad.

6.1. Realizabilidad como transformadora de modelos

En las secciones anteriores, hemos discutido la idea de que si un término realiza una fórmula, entonces la justifica en cierto sentido. Sabemos, por ejemplo, que las tautologías tienen realizadores para cualquier polo. En ese sentido, podemos utilizar la realizabilidad para crear modelos de Henkin donde las fórmulas válidas sean aquellas que son realizables y estudiar dichos modelos para comprender mejor la realizabilidad.

Lo que haremos, en pocas palabras, será utilizar la realizabilidad para, fijado un polo, transformar el modelo de partida en un modelo donde las fórmulas realizables sean válidas. Esto lo haremos para una clase particular de polos que se denominan *polos coherentes*.

Definición 6.1 (Polos coherentes) *Un polo $\perp\!\!\!\perp$ se dice coherente si para todo término de prueba cerrado $t \in PL \cap \Lambda$, existe una pila π tal que $t \star \pi \notin \perp\!\!\!\perp$. Observar que esto es equivalente a decir que no existe $t \in PL \cap \Lambda$ tal que $t \Vdash_{\perp\!\!\!\perp} \bot$.*

Proposición 6.1 (Modelos de realizabilidad henkinianos) *Sea $\perp\!\!\!\perp$ un polo coherente. Entonces existe un modelo de Henkin $\mathscr{M}_{\perp\!\!\!\perp}$ que cumple que para toda fórmula A cerrada y sin parámetros, si existe $t \in PL \cap \Lambda$ con $t \Vdash_{\perp\!\!\!\perp} A$, entonces $\mathscr{M}_{\perp\!\!\!\perp} \vDash A$.*

Demostración. Sea $\mathcal{T} = \{A$ fórmula cerrada sin parámetros : $\exists t \in PL \cap \Lambda, t \Vdash_{\perp\!\!\!\perp} A\}$, por el Teorema de Adecuación dicho conjunto es cerrado por deducción (en el universo de fórmulas cerradas). Además, $\perp \notin \mathcal{T}$ debido a que el polo es coherente, por lo que $\mathcal{T}$ es consistente. Por lo tanto, por el teorema de completitud de Gödel, existe un modelo $\mathscr{M}_{\perp\!\!\!\perp}$ que cumple con los requisitos del enunciado. □

La condición de que t sea un término de prueba es importante ya que sino, siempre que el polo fuese no vacío tendríamos un realizador para $\perp$, por lo que el único polo coherente sería el vacío.

Si bien popularmente se habla de modelos de Henkin, es importante observar que los modelos de realizabilidad henkinianos son modelos de la teoría de fórmulas realizables por términos de prueba, y no necesariamente modelos de $PA2$. De hecho, veremos más adelante que no tienen por qué satisfacer el axioma de inducción.

Utilizando esta definición, podemos agregar un paso intermedio en la propiedad que vimos anteriormente:

	Derivable $\vdash t : A$ con $t \in PL \cap \Lambda$	$\Rightarrow$	**Realizable universalmente** $t \Vvdash A$ con $t \in PL \cap \Lambda$
$\Rightarrow$	**Verdadera en todo modelo de realizabilidad henkiniano** $\mathscr{M}_{\perp\!\!\!\perp} \vDash A \quad \forall \perp\!\!\!\perp$ coherente	$\Rightarrow$	**Verdadera en el modelo de partida** $\mathscr{M} \vDash A$

La segunda implicancia se desprende de la propia definición, mientras que la última se debe a que, cuando trabajamos con el polo coherente más simple -el polo vacío-, el modelo generado es elementalmente equivalente al modelo inicial.

Teorema 6.1 *Si el modelo de partida $\mathscr{M}$ es pleno, los modelos $\mathscr{M}$ y $\mathscr{M}_{\emptyset}$ son elementalmente equivalentes. Esto es: para toda fórmula cerrada A sin parámetros, $\mathscr{M} \vDash A$ sii $\mathscr{M}_{\emptyset} \vDash A$.*

Demostración. Empezaremos probando que, cuando el polo es vacío, existen sólo dos posibles valores de verdad: Λ y $\emptyset$. De hecho, tomando una fórmula cerrada A, observamos que si $||A|| = \emptyset \Rightarrow |A| = \Lambda$, mientras que si $||A|| \neq \emptyset \Rightarrow |A| = \emptyset$. Esto es trivial ya que si no hay tests, entonces todos los términos cerrados realizan la fórmula, mientras que si hay algún test $\pi \in ||A||$, entonces para cualquier término t

cerrado, $t \star \pi \notin \emptyset = \perp\!\!\!\perp$, por lo que $t \notin |A|$.

Al tener sólo dos valores de verdad posibles (Λ y $\emptyset$), la demostración se basa en probar que estos son equivalentes a las dos interpretaciones posibles en el modelo de Henkin:

- Por un lado, vemos que si $|A| = \Lambda$, entonces existe $t \in PL \cap \Lambda$ tal que $t \Vdash A$, por lo que $\mathscr{M}_\emptyset \vDash A$.
- Por otro lado, si $|A| = \emptyset$ entonces $||\neg A|| = \emptyset \cdot \Pi = \emptyset$, lo que implica que $|\neg A| = \Lambda$ y luego $\mathscr{M}_\emptyset \vDash \neg A$, por lo que $\mathscr{M}_\emptyset \nvDash A$.

Tenemos entonces que $|A| = \Lambda \iff \mathscr{M}_\emptyset \vDash A$. Para finalizar la demostración, basta ver que $\mathscr{M} \vDash A \iff |A| = \Lambda$.

Esta prueba la realizamos por inducción. De forma similar al teorema de adecuación, utilizaremos valuaciones para poder lidiar con fórmulas no cerradas. La idea entonces será probar que, para toda fórmula A, $|A|_\rho = \Lambda \iff [\![A]\!]_{\tilde{\rho}} = 1$, donde $\tilde{\rho}$ es una valuación booleana que coincide con ρ en las variables de primer orden, y para toda variable X de segundo orden, $\tilde{\rho}(X) = \mathcal{X}_C$ con $C = \{(x_1, \ldots, x_k) \in \mathscr{M}^k : \rho(X)(x_1, \ldots, x_k) = \emptyset\}$ (el cual pertenece a $P_k^{\mathscr{M}}$ ya que $P_k^{\mathscr{M}} = \mathfrak{P}(\mathscr{M}^k)$ por ser $\mathscr{M}$ pleno).

Vemos que, dada una valuación ρ, las tuplas que satisfacen $\tilde{\rho}(X)$ son aquellas que no tienen refutaciones (tests) en $\rho(X)$.

Es importante notar que si logramos probar dicha equivalencia, tendremos lo que buscamos trivialmente ($\mathscr{M} \vDash A \iff |A| = \Lambda$ cuando A es cerrada).

La prueba se hace por inducción en A:

- Si $A \equiv X(e_1, \ldots, e_k)$, podemos ver que
$$\begin{array}{rcl} [\![A]\!]_{\tilde{\rho}} = 1 & \iff & \tilde{\rho}(X)([\![e_1]\!]_{\tilde{\rho}}, \ldots, [\![e_k]\!]_{\tilde{\rho}}) = 1 \\ & \iff & \tilde{\rho}(X)([\![e_1]\!]_{\rho}, \ldots, [\![e_k]\!]_{\rho}) = 1 \\ & \iff & \rho(X)([\![e_1]\!]_{\rho}, \ldots, [\![e_k]\!]_{\rho}) = \emptyset \\ & \iff & ||A||_\rho = \emptyset \\ & \iff & |A|_\rho = \Lambda \end{array}$$
Para dicho razonamiento basta utilizar las definiciones, observar que $[\![e]\!]_{\tilde{\rho}} = [\![e]\!]_\rho$ para toda expresión e (así se definió $\tilde{\rho}$) y recordar que la última equivalencia se cumple cuando el polo es vacío.

- Cuando $A \equiv B \Rightarrow C$, tenemos que

$$\begin{array}{rcl} [\![A]\!]_{\tilde{\rho}} = 1 & \Longleftrightarrow & [\![B]\!]_{\tilde{\rho}} = 0 \text{ o } [\![C]\!]_{\tilde{\rho}} = 1 \\ & \Longleftrightarrow & |B|_\rho = \emptyset \text{ o } |C|_\rho = \Lambda \\ & \Longleftrightarrow & |B|_\rho = \emptyset \text{ o } ||C||_\rho = \emptyset \\ & \Longleftrightarrow & |B|_\rho \cdot ||C||_\rho = \emptyset \\ & \Longleftrightarrow & ||B \Rightarrow C||_\rho = \emptyset \\ & \Longleftrightarrow & |B \Rightarrow C|_\rho = \Lambda \end{array}$$

  En este caso, además de las definiciones y la propiedad utilizada previamente ($||D||_\rho = \emptyset \Longleftrightarrow |D|_\rho = \Lambda$), se utilizan las hipótesis inductivas para B y C en el paso 2. Recordar que, por hipótesis inductiva, tenemos que $[\![B]\!]_{\tilde{\rho}} = 1 \Longleftrightarrow |B|_\rho = \Lambda$, por lo que $[\![B]\!]_{\tilde{\rho}} = 0 \Longleftrightarrow |B|_\rho = \emptyset$ (al negar de cada lado).

- Para el caso donde $A \equiv \forall x.\, B$, tenemos que

$$\begin{array}{rcl} [\![A]\!]_{\tilde{\rho}} = 1 & \Longleftrightarrow & [\![B]\!]_{\tilde{\rho}, x \leftarrow n} = 1 \text{ para todo } n \in \mathscr{M} \\ & \Longleftrightarrow & |B|_{\rho, x \leftarrow n} = \Lambda \text{ para todo } n \in \mathscr{M} \\ & \Longleftrightarrow & \bigcap_{n \in \mathscr{M}} |B|_{\rho, x \leftarrow n} = \Lambda \\ & \Longleftrightarrow & |A|_\rho = \Lambda \end{array}$$

  Es importante observar que la valuación booleana asociada a $\rho, x \leftarrow n$ es $\tilde{\rho}, x \leftarrow n$. Además, como $||\forall x.B||_\rho = \bigcup_{n \in \mathscr{M}} ||B||_{\rho, x \leftarrow n}$, es fácil ver que $|\forall x.B|_\rho = \bigcap_{n \in \mathscr{M}} |B|_{\rho, x \leftarrow n}$ (esta propiedad se cumple sin importar el polo ni la valuación).

- Por último, si $A \equiv \forall X.\, B$, tenemos que

$$\begin{array}{rcl} [\![A]\!]_{\tilde{\rho}} = 1 & \Longleftrightarrow & [\![B]\!]_{\tilde{\rho}, X \leftarrow \mathcal{X}_C} = 1 \text{ para todo } C \subseteq \mathscr{M}^k \\ & \Longleftrightarrow & |B|_{\rho, X \leftarrow F} = \Lambda \text{ para toda función de falsedad } F \\ & \Longleftrightarrow & \bigcap_{F : \mathscr{M}^k \to \mathfrak{P}(\Pi)} |B|_{\rho, X \leftarrow F} = \Lambda \\ & \Longleftrightarrow & |A|_\rho = \Lambda \end{array}$$

  Este argumento es similar al anterior, con la diferencia de que la demostración de la segunda equivalencia no es trivial.

  Si asumimos que se cumple $[\![B]\!]_{\tilde{\rho}, X \leftarrow \mathcal{X}_C} = 1$ para todo $C \subseteq \mathscr{M}^k$ y tomamos una función de falsedad F, es claro que la valuación booleana asociada a $\rho, X \leftarrow F$ es $\tilde{\rho}, X \leftarrow \mathcal{X}_C$ donde $C = \{(x_1, \ldots, x_k) \in \mathscr{M}^k : F(x_1, \ldots, x_k) = \emptyset\}$. Luego, como $[\![B]\!]_{\tilde{\rho}, X \leftarrow \mathcal{X}_C} = 1$, por hipótesis inductiva tenemos que $|B|_{\rho, X \leftarrow F} = \Lambda$. Como este argumento se cumple para cualquier función de falsedad F, concluimos el directo de dicha equivalencia.

  Para demostrar el recíproco, asumimos que $|B|_{\rho, X \leftarrow F} = \Lambda$ para toda función de falsedad F y tomamos un $C \in P_k^{\mathscr{M}}$. Sea F una función de falsedad tal que $F(x_1, \ldots, x_k) = \emptyset \Longleftrightarrow (x_1, \ldots, x_k) \in C$, podemos ver que la valuación booleana asociada a $\rho, X \leftarrow F$ es $\tilde{\rho}, X \leftarrow \mathcal{X}_C$. Luego, como $|B|_{\rho, X \leftarrow F} = \Lambda$, por hipótesis inductiva tenemos que $[\![B]\!]_{\tilde{\rho}, X \leftarrow \mathcal{X}_C} = 1$. Este argumento se cumple para cualquier $C \in P_k^{\mathscr{M}}$ por lo que podemos concluir el recíproco.

$\square$

6.2. Ejemplos de modelos

Si bien hemos hablado de los modelos de realizabilidad y cómo inducen modelos de Henkin, aún no hemos visto demasiados ejemplos de estos. En lo que queda de la sección presentaremos distintos modelos de realizabilidad y, en particular, el modelo de hilos.

6.2.1. Modelos triviales

Modelo degenerado. El modelo degenerado es definido al tomar $\perp\!\!\!\perp := \Lambda \star \Pi$, el cual es trivialmente cerrado por anti-evaluación. Luego, como cualquier proceso pertenece al polo, nos queda que $|A| = \Lambda$ para toda fórmula A. En este caso, el polo no es coherente por lo que no se induce un modelo $\mathscr{M}_{\perp\!\!\!\perp}$.

Modelo de Henkin. De forma opuesta al modelo anterior, este modelo queda definido al tomar $\perp\!\!\!\perp := \emptyset$, el cual es también trivialmente cerrado por anti-evaluación. Como ya vimos, en este modelo tenemos sólo dos posibles valores de verdad (Λ y $\emptyset$) y el modelo de Henkin inducido termina siendo elementalmente equivalente al modelo de partida.

6.2.2. Modelos generados por un único proceso

Modelos dirigidos hacia un proceso. Esta familia de modelos se enfocan en un único proceso p_0: para que un término t pase un test π, el proceso $t \star \pi$ debe eventualmente llegar a p_0. Esto se logra tomando el polo $\perp\!\!\!\perp := \{p \mid p \succeq p_0\}$, el cual es el más chico entre los que contienen a p_0.

Estos modelos son sumamente útiles para problemas de especificación (que veremos luego): si queremos probar que $p_1 \succeq p_2$, tomamos $\perp\!\!\!\perp := \{p \mid p \succeq p_2\}$ y luego probamos que $p_1 \in \perp\!\!\!\perp$. Este tipo de polos son llamados *orientados a un objetivo*, porque son definidos por el proceso que quieren alcanzar. Además, se pueden generalizar a un conjunto de procesos P al tomar $\perp\!\!\!\perp := \{p \mid \exists p' \in P.\, p \succeq p'\}$.

Modelos generados por un hilo. Esta familia de modelos toma el enfoque opuesto a la anterior: en lugar de considerar los procesos que reducen a un proceso p_0, se consideran aquellos que no.

Definición 6.2 (Hilo de un proceso) *El hilo de un proceso p se define como el conjunto $Thd(p) := \{p' \in \Lambda \star \Pi \mid p \succeq p'\}$.*

Vemos que el hilo de un proceso es cerrado por evaluación (no confundir con anti-evaluación) y, si $p \notin \perp\!\!\!\perp$, entonces su hilo tampoco. En esta familia, el polo es el complemento del hilo de un proceso p_0: $\perp\!\!\!\perp := \Lambda \star \Pi \setminus Thd(p_0)$.

A estos modelos también los podemos utilizar en problemas de especificación: para probar que $p_1 \succeq p_2$, tomamos el polo $\perp\!\!\!\perp := \Lambda \star \Pi \setminus Thd(p_1)$ y probamos que $p_2 \notin \perp\!\!\!\perp$ ya que, por definición, $p_2 \notin \perp\!\!\!\perp \iff p_2 \in Thd(p_1)$.

Podemos generalizar esto para un conjunto de procesos P al tomar $\perp\!\!\!\perp := \Lambda \star \Pi \setminus (\bigcup_{p \in P} Thd(p)) = \bigcap_{p \in P}(\Lambda \star \Pi \setminus Thd(p))$. Un ejemplo interesante de esta generalización es el modelo de hilos.

6.2.3. El modelo de hilos

El modelo de hilos intenta tener seguimiento sobre aquellos procesos que comienzan con un término de prueba. Con el fin de recordar durante la ejecución cuál era el término de prueba original del proceso, dicha información es almacenada en el único lugar que no es necesariamente utilizado para computar: las constantes de pila. Recordemos que la constante de pila de un proceso es única y se encuentra al final de la pila.

Empezamos asumiendo una biyección entre los términos de prueba y las constantes de pila. Como los términos de prueba son numerables, esto es lo mismo a requerir una cantidad numerable de constantes de pila, y una enumeración de ambos. Escribiremos α_t a la constante asociada al término de prueba t. Los procesos de nuestro interés serán de la forma $t \star \alpha_t$.

Definición 6.3 (Modelo de hilos) *El modelo de hilos queda definido por el polo*

$$\perp\!\!\!\perp := \Lambda \star \Pi \setminus (\textstyle\bigcup_{t \in PL \cap \Lambda} Thd(t \star \alpha_t))$$

Observemos que el hecho de utilizar una pila vacía (sólo con la constante) no nos quita generalidad: sean $t, u_1, \ldots, u_k \in PL \cap \Lambda$, $t \star u_1 \cdots u_k \cdot \alpha_{t\, u_1\, u_2 \ldots u_k}$ pertenece al hilo de $t\, u_1\, u_2\, \ldots\, u_k \star \alpha_{t\, u_1\, u_2 \ldots u_k}$.

Además, vemos que el polo es coherente ya que, para todo $t \in PL \cap \Lambda$, $t \star \alpha_t \notin \perp\!\!\!\perp$, por lo que ningún término de prueba realiza $\perp$. En cierto sentido, estamos tomando un polo coherente lo más grande posible: forzamos a que no haya un término de prueba que realice $\perp$ al hacer que cada uno de ellos no logre pasar al menos un test α_t.

¿Puede pasar que $p_2 \in Thd(p_1)$ ($p_1 \succeq p_2$) pero que la constante de pila de p_2 sea distinta a la de p_1? Sabemos que las constantes de continuación permiten cambiar la pila de un proceso, y por lo tanto la constante de pila. Aún así, como el término de p_1

es un término de prueba (no tiene constantes de continuación), todas las constantes de continuación que aparezcan en el hilo de p_1 sólo pueden haber sido generadas previamente por la instrucción `callcc`, por lo que estarán almacenando una copia de una pila anterior en la ejecución, manteniendo así la constante de pila.

De todos modos, el razonamiento anterior no es del todo completo. Recordemos que una instancia del λ_c-cálculo puede contener otras instrucciones, y estas instrucciones podrían cambiar la constante de pila de un proceso. Además, la relación de evaluación $\succ$ fue definida como una relación transitiva que cumplía ciertas reglas ($PUSH$, $GRAB$, $SAVE$ y $RESTORE$) pero dichas reglas no eran pasos atómicos. Podría pasar que, sin necesidad de agregar instrucciones, la relación de evaluación modifique las constantes de pila en algún paso y que después deshaga el cambio.

7. Realizando la Aritmética de Peano

Gracias al Teorema de Adecuación, toda fórmula derivable de la lógica de segundo orden es realizable. Si bien esto nos da realizadores universales para una gran cantidad de fórmulas, aún no sabemos si por ejemplo la teoría $PA2$ es realizable. En esta sección buscaremos realizar dicha teoría, lo que en particular implicaría que los modelos de realizabilidad henkinianos la modelan.

Para simplificar dicho estudio, a partir de esta sección trabajaremos con el modelo estándar de naturales $\mathbb{N}$ como el modelo de partida ($\mathscr{M} = \mathbb{N}$). El modelo $\mathbb{N}$ consiste del modelo estándar de primer orden, tomando todas las relaciones de todas las aridades (al que llamamos modelo *pleno*).

7.1. Realizando los Axiomas de Peano

Gracias al Teorema de Adecuación, basta ver que los axiomas de $PA2$ son realizables para probar que toda la teoría lo es. Recordando dichos axiomas -presentados en la sección 3.4- podemos observar cómo la igualdad tiene un rol predominante, por lo que resulta natural primero estudiar las igualdades.

7.1.1. Realizadores de igualdades

Como hasta ahora, cuando trabajamos con un polo fijo obviamos su notación en $||A||_{\perp\!\!\!\perp}$, $|A|_{\perp\!\!\!\perp}$, $\Vdash_{\perp\!\!\!\perp}$ y $\leq_{\perp\!\!\!\perp}$.

Lema 7.1 (Interpretación de igualdades cerradas) *Sea $\perp\!\!\!\perp$ un polo, e_1 y e_2 expresiones aritméticas cerradas:*

$$||e_1 = e_2|| = \begin{cases} ||1|| = \{t \cdot \pi \mid t \star \pi \in \perp\!\!\!\perp\} & si\ [\![e_1]\!] = [\![e_2]\!] \\ ||\top \Rightarrow \bot|| = \Lambda \cdot \Pi & si\ [\![e_1]\!] \neq [\![e_2]\!] \end{cases}$$

Demostración. Recordar que $e_1 = e_2 \equiv \forall Z.\ Z(e_1) \Rightarrow Z(e_2)$.

- Si $[\![e_1]\!] = [\![e_2]\!]$:

$$\begin{array}{rcl} ||e_1 = e_2|| & = & \bigcup_{F:\mathbb{N}\to\mathfrak{P}(\Pi)} ||\dot{F}(e_1) \Rightarrow \dot{F}(e_2)|| \\ & = & \bigcup_{F:\mathbb{N}\to\mathfrak{P}(\Pi)} (F([\![e_1]\!]))^{\perp\!\!\!\perp} \cdot F([\![e_2]\!]) \\ & = & \bigcup_{G\subseteq\Pi} G^{\perp\!\!\!\perp} \cdot G \\ & = & \bigcup_{H:\mathfrak{P}(\Pi)} ||\dot{H} \Rightarrow \dot{H}|| \\ & = & ||1|| \end{array}$$

- Si $[\![e_1]\!] \neq [\![e_2]\!]$, podemos tomar una función de falsedad H tal que $H([\![e_1]\!]) = \emptyset$ y $H([\![e_2]\!]) = \Pi$. Luego:

$$\begin{array}{rcl} ||e_1 = e_2|| & = & \bigcup_{F:\mathbb{N}\to\mathfrak{P}(\Pi)} ||\dot{F}(e_1) \Rightarrow \dot{F}(e_2)|| \\ & \supseteq & ||\dot{H}(e_1) \Rightarrow \dot{H}(e_2)|| \\ & = & \Lambda \cdot \Pi \\ & = & ||\top \Rightarrow \bot|| \end{array}$$

Como $\top \Rightarrow \bot$ es un subtipo de $A \Rightarrow B$ para cualquier fórmula A y B, $||A \Rightarrow B|| \subseteq ||\top \Rightarrow \bot||$. Por lo tanto, $||\top \Rightarrow \bot||$ es el valor más grande para $||e_1 = e_2||$, de donde se concluye la igualdad. $\square$

Este resultado puede ser extendido para cualquier expresión (no necesariamente cerrada). Para esto, notaremos una tupla de variables $(x_1, \ldots, x_k)$ como $\vec{x}$ y, particularmente, a las expresiones que dependan sólo de las variables $(x_1, \ldots, x_k)$ las escribiremos $e(\vec{x})$ para marcar que $FV(e) \subseteq \{x_1, \ldots, x_k\}$. Más aún, si tenemos una tupla $(n_1, \ldots, n_k) \in \mathbb{N}^k$, notaremos $e(\vec{n})$ a la expresión e donde sustituimos las variables x_i por los parámetros de los individuos n_i: $e(\vec{n}) := e[\dot{n}_1/x_1] \cdots [\dot{n}_k/x_k]$ (observar que $e(\vec{n})$ queda cerrada).

Lema 7.2 (Realizando igualdades y desigualdades) *Sean $e_1(\vec{x})$ y $e_2(\vec{x})$ dos expresiones aritméticas que dependen sólo de las variables $\vec{x}$:*

1. *Si para toda tupla $\vec{n} \in \mathbb{N}^k$, $[\![e_1(\vec{n})]\!] = [\![e_2(\vec{n})]\!]$, entonces:*

$$\lambda y.\, y \Vdash \forall \vec{x}.\, e_1(\vec{x}) = e_2(\vec{x})$$

2. *Si para toda tupla $\vec{n} \in \mathbb{N}^k$, $[\![e_1(\vec{n})]\!] \neq [\![e_2(\vec{n})]\!]$, entonces para todo λ_c-término u tal que $FV(u) \subseteq \{y\}$:*

$$\lambda y.\, y\, u \Vdash \forall \vec{x}.\, \neg(e_1(\vec{x}) = e_2(\vec{x}))$$

Demostración.

1. Esto es una consecuencia directa del lema anterior, basta observar los siguientes puntos:

   a) $||\forall \vec{x}.\, e_1(\vec{x}) = e_2(\vec{x})|| = \bigcup_{\vec{n}\in\mathbb{N}^k} ||e_1(\vec{n}) = e_2(\vec{n})||$ para todo polo.

   b) Por el lema anterior, $||1|| = ||e_1(\vec{n}) = e_2(\vec{n})||$ para toda tupla $\vec{n}$. Sumado al punto a), vemos que $||1|| = ||\forall \vec{x}.\, e_1(\vec{x}) = e_2(\vec{x})||$ para todo polo.

   c) A partir de lo anterior, se concluye que $1 \approx_{\perp\!\!\!\perp} \forall \vec{x}.\, e_1(\vec{x}) = e_2(\vec{x})$ para todo polo. Luego, como $\lambda y.\, y \Vdash 1$, concluimos que $\lambda y.\, y \Vdash \forall \vec{x}.\, e_1(\vec{x}) = e_2(\vec{x})$.

2. Por el lema anterior, sabemos que para toda tupla $\vec{n} \in \mathbb{N}^k$:

$$||\neg(e_1(\vec{n}) = e_2(\vec{n}))|| = ||(\top \Rightarrow \bot) \Rightarrow \bot|| \text{ para todo polo}$$

Por lo tanto:

$$||\forall\vec{x}.\, \neg(e_1(\vec{x}) = e_2(\vec{x}))|| = ||(\top \Rightarrow \bot) \Rightarrow \bot|| \text{ para todo polo}$$

Luego, basta ver que $\lambda y.\, y\, u \Vdash (\top \Rightarrow \bot) \Rightarrow \bot$. Esto se puede probar gracias a la demostración modular que hicimos del Teorema de Adecuación. Veamos la siguiente aplicación de reglas de deducción (que **no** es un árbol de deducción):

$$\cfrac{\cfrac{\cfrac{}{y : \top \Rightarrow \bot \vdash y : \top \Rightarrow \bot}\ \text{Axioma} \qquad y : \top \Rightarrow \bot \vdash u : \top}{y : \top \Rightarrow \bot \vdash y\, u : \bot}\ \Rightarrow_e}{\vdash \lambda y.\, y\, u : (\top \Rightarrow \bot) \Rightarrow \bot}\ \Rightarrow_i$$

Por el Lema de Adecuación, todas las reglas son adecuadas, por lo que si el secuente $y : \top \Rightarrow \bot \vdash u : \top$ es adecuado para todo polo, también lo será la conclusión. Para eso basta ver que si se toma cualquier sustitución cerrada σ tal que $\sigma(y) \Vdash \top \Rightarrow \bot$, entonces $u[\sigma] \Vdash \top$ (ya que cualquier término cerrado lo realiza). Como la conclusión es adecuada para todo polo, dado un polo $\perp\!\!\!\perp$, una sustitución cerrada σ, y una valuación ρ cualquiera, tenemos:

$$(\lambda y.\, y\, u)[\sigma] \Vdash ((\top \Rightarrow \bot) \Rightarrow \bot)[\rho]$$

Como $\lambda y.\, y\, u$ es cerrado (ya que $FV(u) \subseteq \{y\}$), y $(\top \Rightarrow \bot) \Rightarrow \bot$ también, concluimos que $\lambda y.\, y\, u \Vdash (\top \Rightarrow \bot) \Rightarrow \bot$ para todo polo. $\square$

El lema anterior nos dice, por ejemplo, que el axioma $\forall x. \neg(s\, x = 0)$ es universalmente realizable por un término de prueba. Dejando de lado el axioma de inducción, el resto de los axiomas tienen en general un formato similar: son implicancias entre igualdades. El siguiente teorema nos permitirá realizar universalmente los restantes axiomas a excepción del de inducción.

Teorema 7.1 *Sean $e_0(\vec{x}), e'_0(\vec{x}), \ldots, e_k(\vec{x}), e'_k(\vec{x})$ expresiones aritméticas que dependen sólo de las variables $\vec{x}$. Si para toda tupla $\vec{n} \in \mathbb{N}^l$ se cumple que*

$$[\![e_k(\vec{n})]\!] = [\![e'_k(\vec{n})]\!] \Rightarrow \cdots \Rightarrow [\![e_0(\vec{n})]\!] = [\![e'_0(\vec{n})]\!]$$

entonces:

$$\lambda y.\, y \Vdash \forall\vec{x}.\, e_k(\vec{x}) = e'_k(\vec{x}) \Rightarrow \cdots \Rightarrow e_0(\vec{x}) = e'_0(\vec{x})$$

Demostración. Similar a las demostraciones que venimos haciendo, basta demostrar que, para todo polo, $1 \leq e_k(\vec{n}) = e'_k(\vec{n}) \Rightarrow \cdots \Rightarrow e_0(\vec{n}) = e'_0(\vec{n})$ para toda tupla $\vec{n} \in \mathbb{N}^l$. Esta afirmación la probaremos por inducción, fijando un polo y una tupla $\vec{n}$.

Si $k = 0$, por hipótesis $[\![e_0(\vec{n})]\!] = [\![e'_0(\vec{n})]\!]$. Utilizando el lema 7.1, vemos que $||e_0(\vec{n}) = e'_0(\vec{n})|| = ||1||$ de donde $1 \leq e_0(\vec{n}) = e'_0(\vec{n})$.

Si $k \geq 1$ lo dividimos en dos casos:

- Si $[\![e_k(\vec{n})]\!] = [\![e'_k(\vec{n})]\!]$, entonces por el lema 7.1 sabemos que $e_k(\vec{n}) = e'_k(\vec{n}) \approx 1$. Además, por hipótesis, también se cumple:

  $$[\![e_{k-1}(\vec{n})]\!] = [\![e'_{k-1}(\vec{n})]\!] \Rightarrow \cdots \Rightarrow [\![e_0(\vec{n})]\!] = [\![e'_0(\vec{n})]\!]$$

  Por lo que, usando la hipótesis inductiva, podemos afirmar que:

  $$1 \leq e_{k-1}(\vec{n}) = e'_{k-1}(\vec{n}) \Rightarrow \cdots \Rightarrow e_0(\vec{n}) = e'_0(\vec{n})$$

  Combinando ambos puntos, se concluye:

  $$1 \Rightarrow 1 \leq e_k(\vec{n}) = e'_k(\vec{n}) \Rightarrow \cdots \Rightarrow e_0(\vec{n}) = e'_0(\vec{n})$$

  Faltaría ver que $1 \leq 1 \Rightarrow 1$. En realidad, se cumple algo más general y es que $1 \leq A \Rightarrow A$ para toda fórmula A cerrada. Para esto, basta observar que $||A \Rightarrow A|| = |A| \cdot ||A|| \subseteq \{t \cdot \pi \mid t \star \pi \in \perp\!\!\!\perp\} = ||1||$.

- Si $[\![e_k(\vec{n})]\!] \neq [\![e'_k(\vec{n})]\!]$, sabemos por el lema 7.1 que $e_k(\vec{n}) = e'_k(\vec{n}) \approx \top \Rightarrow \bot$. Además, como $\top \Rightarrow \bot$ es un subtipo de cualquier implicancia, tenemos que:

  $$\top \Rightarrow \bot \leq e_{k-1}(\vec{n}) = e'_{k-1}(\vec{n}) \Rightarrow \cdots \Rightarrow e_0(\vec{n}) = e'_0(\vec{n})$$

  Juntando lo anterior, podemos ver que $(\top \Rightarrow \bot) \Rightarrow \top \Rightarrow \bot$ es un subtipo de la implicancia completa. Otra vez, se concluye lo buscado ya que 1 es un subtipo de $(\top \Rightarrow \bot) \Rightarrow (\top \Rightarrow \bot)$. □

Usando el teorema y los lemas anteriores, podemos realizar los axiomas de Peano a excepción del de inducción.

Proposición 7.1 (Realizadores para los axiomas de PA^-) *Sea $PA2^-$ el subconjunto de los axiomas de Peano donde quitamos el de inducción, podemos realizar universalmente $PA2^-$. Los realizadores son los siguientes:*

Sucesor	$\lambda x.\, x$	$\Vdash$	$\forall xy.\, s\,x = s\,y \Rightarrow x = y$
	$\lambda x.\, x\,x$	$\Vdash$	$\forall x. \neg(s\,x = 0)$
f recursiva primitiva	$\lambda x.\, x$	$\Vdash$	$\forall \vec{x}.\, f(\dots) = \dots$

Demostración. Basta utilizar el teorema 7.1 para todos los axiomas, salvo para el axioma $\forall x.\neg(s\,x = 0)$. Para este, utilizamos el punto 2 del lema 7.2 y tomamos $u \equiv x$. □

Es importante observar que los realizadores que obtuvimos, además de ser universales, son términos de prueba. Por lo tanto, estos axiomas son válidos en todos los modelos de realizabilidad henkinianos.

7.1.2. Realizando el axioma de inducción

Como habíamos adelantado, a diferencia de los casos anteriores, no se puede realizar universalmente el axioma de inducción. Específicamente, esto no es posible cuando la relación de evaluación es *determinista*.

Definición 7.1 (Relación de evaluación determinista) *Sea* $(\mathcal{K}, \Pi_0, \succ)$ *una instancia del* λ_c*-cálculo, decimos que la relación* $\succ$ *es determinista si siempre que* $p \succ p_1$ *y* $p \succ p_2$ *tenemos que* $p_1 \succeq p_2$ *o* $p_2 \succeq p_1$*.*

Teorema 7.2 (El axioma de inducción no es realizable universalmente) *Si la relación de evaluación es determinista, entonces el axioma de inducción no tiene realizadores universales.*

Demostración. La idea será tomar dos procesos q y r distintos, tales que $q \not\succeq r$ y $r \not\succeq q$, y probar, asumiendo que el axioma de inducción es realizable universalmente, que existe otro proceso p tal que $p \succ q$ y $p \succ r$, lo que es absurdo por el determinismo de la relación de evaluación.

Para probar que, de hecho, hay procesos q y r tales que $q \not\succeq r$ y $r \not\succeq q$, basta buscar en la ejecución de un término que entre en un bucle infinito. Utilizaremos $\delta := \lambda x.x\,x$ y $\Omega := \delta\,\delta$ para esto. Primero observamos que, para toda pila π, tenemos la siguiente evaluación:

$$\Omega \star \pi \equiv \delta\,\delta \star \pi \succ \delta \star \delta \cdot \pi \succ (x\,x)[\delta/x] \star \pi \equiv \delta\,\delta \star \pi \equiv \Omega \star \pi$$

Particularmente, tenemos que $Thd(\Omega \star \pi)$ es finito para cualquier pila π. Más aún, si tomamos un término cerrado t_0 cualquiera y π una pila cualquiera, como $Thd(\Omega \star t_0 \cdot \pi)$ es finito existe un término cerrado t_1 tal que $\Omega \star t_1 \cdot \pi$ no está en dicho hilo y, por lo tanto, tendremos que $\Omega \star t_0 \cdot \pi \not\succeq \Omega \star t_1 \cdot \pi$ y también que $\Omega \star t_1 \cdot \pi \not\succeq \Omega \star t_0 \cdot \pi$. Esto último debido a que si $\Omega \star t_1 \cdot \pi \succeq \Omega \star t_0 \cdot \pi$, como asumimos una relación de evaluación determinista y sabemos que $\Omega \star t_1 \cdot \pi$ ejecuta en bucle, tendríamos que $\Omega \star t_1 \cdot \pi \succeq \Omega \star t_0 \cdot \pi \succeq \Omega \star t_1 \cdot \pi$, lo cual es absurdo.

Fijando π, t_0 y t_1 como recién, tomaremos $q \equiv \Omega \star t_0 \cdot \pi$ y $r \equiv \Omega \star t_1 \cdot \pi$ (con la idea que mencionamos en el primer párrafo), por lo que faltaría encontrar el proceso p tal que $p \succ q$ y $p \succ r$ para llegar al absurdo.

Asumiendo que existe t_{rec} un realizador universal del axioma de inducción, utilizaremos $p \equiv t_{rec} \star \Omega\, t_0 \cdot \lambda x.\, \Omega\, t_1 \cdot \pi$ para concluir la contradicción.

Veamos $p \succ q$. Si tomamos el polo $\perp\!\!\!\perp_0 := \{p \,|\, p \succ q\}$, basta ver que $p \in \perp\!\!\!\perp_0$. Más aún, alcanza con probar que:

$$\Omega\, t_0 \cdot \lambda x.\, \Omega\, t_1 \cdot \pi \in ||\forall x. \forall Z.\, Z(0) \Rightarrow (\forall y.\, Z(y) \Rightarrow Z(s\, y)) \Rightarrow Z(x)||_{\perp\!\!\!\perp_0}$$

Si consideramos P la función de falsedad tal que $P(0) := \{\pi\}$ y para el resto de los valores n, $P(n) := \emptyset$, bastará con probar:

$$\Omega\, t_0 \cdot \lambda x.\, \Omega\, t_1 \cdot \pi \in ||\dot{P}(0) \Rightarrow (\forall y.\, \dot{P}(y) \Rightarrow \dot{P}(s\, y)) \Rightarrow \dot{P}(\dot{0})||_{\perp\!\!\!\perp_0}$$

Observar que el primer 0 es la constante del lenguaje, mientras que el segundo es el parámetro del 0 en el modelo de partida (son distintos símbolos, aunque sus interpretaciones son iguales).
La prueba de esto último es la siguiente:

1. $||\dot{P}(0)||_{\perp\!\!\!\perp_0} = \{\pi\}$, y sabemos que $\Omega\, t_0 \star \pi \in \perp\!\!\!\perp_0$, por lo que $\Omega\, t_0 \in |\dot{P}(0)|_{\perp\!\!\!\perp_0}$.

2. Tenemos que:

$$\begin{array}{rcl} ||\forall y.\, \dot{P}(y) \Rightarrow \dot{P}(s\, y)||_{\perp\!\!\!\perp_0} & = & \bigcup_{n \in \mathbb{N}} ||\dot{P}(\dot{n}) \Rightarrow \dot{P}(s\, \dot{n})||_{\perp\!\!\!\perp_0} \\ & = & \bigcup_{n \in \mathbb{N}} |\dot{P}(\dot{n})|_{\perp\!\!\!\perp_0} \cdot ||\dot{P}(s\, \dot{n})||_{\perp\!\!\!\perp_0} \\ & = & \bigcup_{n \in \mathbb{N}} |\dot{P}(\dot{n})|_{\perp\!\!\!\perp_0} \cdot \emptyset \\ & = & \emptyset \end{array}$$

   Por lo que cualquier término cerrado es un realizador de dicha fórmula. En particular, $\lambda x.\, \Omega\, t_1 \in |\forall y.\, \dot{P}(y) \Rightarrow \dot{P}(s\, y)|_{\perp\!\!\!\perp_0}$.

3. $\pi \in ||\dot{P}(\dot{0})||_{\perp\!\!\!\perp_0}$ por definición de P.

4. En base a los puntos anteriores, concluimos que:

$$\begin{array}{rcl} \Omega\, t_0 \cdot \lambda x.\, \Omega\, t_1 \cdot \pi & \in & ||\dot{P}(0) \Rightarrow (\forall y.\, \dot{P}(y) \Rightarrow \dot{P}(s\, y)) \Rightarrow \dot{P}(\dot{0})||_{\perp\!\!\!\perp_0} \\ & \subseteq & ||\forall x. \forall Z.\, Z(0) \Rightarrow (\forall y.\, Z(y) \Rightarrow Z(s\, y)) \Rightarrow Z(x)||_{\perp\!\!\!\perp_0} \end{array}$$

   .

Veamos $p \succ r$. Similar a lo anterior, si tomamos el polo $⫫_1 := \{p \mid p \succ r\}$, basta ver que $p \in ⫫_1$. En este caso, consideramos Q la función de falsedad tal que $Q(0) := \emptyset$ y para el resto de los valores n, $Q(n) = \{\pi\}$. Veremos específicamente que:

$$\Omega\, t_0 \cdot \lambda x.\, \Omega\, t_1 \cdot \pi \in ||\dot{Q}(0) \Rightarrow (\forall y.\, \dot{Q}(y) \Rightarrow \dot{Q}(s\, y)) \Rightarrow \dot{Q}(\dot{1})||_{⫫_1}$$

1. $||\dot{Q}(0)||_{⫫_1} = \emptyset$, por lo que $\Omega\, t_0 \in |\dot{Q}(0)|_{⫫_1}$.

2. Observemos que:

$$\begin{array}{rcl} ||\forall y.\, \dot{Q}(y) \Rightarrow \dot{Q}(s\, y)||_{⫫_1} & = & \bigcup_{n\in\mathbb{N}} |\dot{Q}(\dot{n})|_{⫫_1} \cdot ||\dot{Q}(s\, \dot{n})||_{⫫_1} \\ & = & \bigcup_{n\in\mathbb{N}} |\dot{Q}(\dot{n})|_{⫫_1} \cdot \{\pi\} \\ & = & \Lambda \cdot \{\pi\} \end{array}$$

   Como $\lambda x.\, \Omega\, t_1 \star u \cdot \pi \succ \Omega\, t_1 \star \pi \in ⫫_1$ para todo $u \in \Lambda$, tenemos que $\lambda x.\, \Omega\, t_1 \in |\forall y.\, \dot{Q}(y) \Rightarrow \dot{Q}(s\, y)|_{⫫_1}$.

3. $\pi \in ||\dot{Q}(\dot{1})||_{⫫_1}$ por definición de Q.

4. En base a los puntos anteriores, concluimos que:

$$\begin{array}{rcl} \Omega\, t_0 \cdot \lambda x.\, \Omega\, t_1 \cdot \pi & \in & ||\dot{Q}(0) \Rightarrow (\forall y.\, \dot{Q}(y) \Rightarrow \dot{Q}(s\, y)) \Rightarrow \dot{Q}(\dot{1})||_{⫫_1} \\ & \subseteq & ||\forall x. \forall Z.\, Z(0) \Rightarrow (\forall y.\, Z(y) \Rightarrow Z(s\, y)) \Rightarrow Z(x)||_{⫫_1} \end{array}$$

$\square$

Observación 7.1 *Es importante observar que lo único que pedimos sobre la instancia del λ_c-cálculo es que la relación de evaluación sea determinista. Por lo tanto, incluso aunque se agreguen nuevas instrucciones, el axioma de inducción continuará sin ser universalmente realizable.*

7.1.3. Resolviendo el problema del axioma de inducción

Si bien no podemos realizar universalmente el axioma de inducción, intentaremos lidiar con dicho problema al trabajar sólo con aquellos individuos que sean naturales. Si pensamos en el axioma de inducción como un axioma que nos establece que todos los individuos son naturales, podemos condicionar las fórmulas para trabajar únicamente con ellos.

Para esto, necesitamos tener una fórmula que exprese que un individuo es un natural. Como trabajamos con lógica de segundo orden, podemos utilizar la definición de Dedekind de los números naturales: son aquellos individuos que son parte de cualquier conjunto que contiene al 0 y que es cerrado para la función sucesor:

$$\mathbb{N} \quad := \quad \{x \mid \forall X.\, 0 \in X \Rightarrow (\forall y.\, y \in X \Rightarrow (y+1) \in X) \Rightarrow x \in X\}$$

En nuestro marco de trabajo, esto se traduce directamente a la fórmula $e \in \mathbb{N}$ que establece que la expresión aritmética e es un natural:

$$\textbf{Naturales} \quad e \in \mathbb{N} \quad := \quad \forall Z.\, Z(0) \Rightarrow (\forall y.\, Z(y) \Rightarrow Z(s\, y)) \Rightarrow Z(e)$$

Notar que, con esta definición, el axioma de inducción queda $\forall x.\, x \in \mathbb{N}$, que ilustra la idea de que está restringiendo los individuos a naturales.

Siguiendo esta idea, utilizaremos una técnica conocida como *relativización* con el fin de limitar los cuantificadores de primer orden al universo de los naturales:

$$\begin{array}{rlll} \textbf{Cuantificadores} & \forall x \in \mathbb{N}.\, A & := & \forall x.\, x \in \mathbb{N} \Rightarrow A \\ \textbf{relativizados} & \exists x \in \mathbb{N}.\, A & := & \forall Z.\, (\forall x \in \mathbb{N}.\, A \Rightarrow Z) \Rightarrow Z \\ & & \equiv & \forall Z.\, (\forall x.\, x \in \mathbb{N} \Rightarrow A \Rightarrow Z) \Rightarrow Z \end{array}$$

Las fórmulas donde todas las cuantificaciones de primer orden están relativizadas, son llamadas *fórmulas aritméticas.* Notaremos $A^{\mathbb{N}}$ a la fórmula A relativizada, que consiste de la fórmula A donde se sustituyen los cuantificadores universales de primer orden por cuantificadores relativizados. Por definición, $A^{\mathbb{N}}$ es aritmética.

La relativización restringe los individuos donde se cuantifica a los naturales, permitiendo utilizar el axioma de inducción en ellos. Por lo tanto, esperaríamos que esto fuera suficiente como para recuperar todos los teoremas de $PA2$. Más precisamente, buscamos demostrar el siguiente resultado:

Teorema 7.3 (Relativización en $PA2^-$) *Para toda fórmula A cerrada, si el secuente $PA2 \vdash A$ es derivable, entonces $PA2^-, \Delta \vdash A^{\mathbb{N}}$ también lo es, donde definimos $\Delta := \{(\dot{n} \in \mathbb{N}) \ : \ n \text{ es natural}\}$.*

Dicho resultado implica que siempre que A es derivable en $PA2$, $A^{\mathbb{N}}$ es realizable universalmente.

Corolario 7.1 (Realizando la Aritmética de Peano) *Para toda fórmula A cerrada tal que $PA2 \vdash A$ es derivable, $A^{\mathbb{N}}$ es realizable universalmente por un término de prueba.*

Demostración. Si $PA2 \vdash A$ es derivable, por el teorema anterior tenemos que $PA2^-, \Delta \vdash A^{\mathbb{N}}$ también lo es. Por lo tanto, existe un contexto $\Gamma \subset PA2^- \cup \Delta$ tal que $\Gamma \vdash A^{\mathbb{N}}$ es derivable. Utilizando el teorema 5.1, sabemos que existe un contexto Γ' que contiene las mismas fórmulas que Γ pero asociadas a variables de tipo, y un término de prueba t, tal que $\Gamma' \vdash t : A^{\mathbb{N}}$ es derivable.

Sabiendo que todas las fórmulas en Γ' son cerradas y que $A^{\mathbb{N}}$ lo es (porque A es cerrada), podemos tomar σ una sustitución cerrada tal que $\sigma \Vdash \Gamma'$. Para esto, basta asignar a cada variable en el dominio de Γ' un término de prueba que realice universalmente su fórmula asociada, ya sea un axioma de $PA2^-$ o $\dot{n} \in \mathbb{N}$ para algún n natural. El primer caso ya vimos que es posible, pero el segundo lo veremos luego (en el corolario 7.2). Por el teorema de adecuación, obtenemos que $t[\sigma] \Vdash A^{\mathbb{N}}$. Como $t \in PL$, y $\sigma(x) \in PL$ para todo $x \in dom\,\Gamma' \supseteq FV(t)$, nos queda $t[\sigma] \in PL$.

$\square$

Para probar el teorema trabajaremos por inducción, pero debemos probar un lema más fuerte ya que en la derivación de un secuente $PA2 \vdash A$ los contextos pueden contener fórmulas que no sean axiomas de $PA2$.

Lema 7.3 (Relativización en $PA2^-$) *Para toda fórmula A y todo contexto Γ, si $PA2, \Gamma \vdash A$ es derivable, entonces $PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash A^{\mathbb{N}}$ también lo es. Notamos $\Gamma^{\mathbb{N}} := \{B^{\mathbb{N}} : B \in \Gamma\} \cup \{(x \in \mathbb{N}) : x \in FV(\Gamma)$ y x es de primer orden $\}$ y $FV_1(A)^{\mathbb{N}} := \{(x \in \mathbb{N}) : x \in FV(A)$ y x es de primer orden $\}$.*

Demostración. La prueba es por inducción en las reglas de deducción del sistema sin realizadores explícitos:

- **Axioma.**

  H) $A \in PA2 \cup \Gamma$
  T) $PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash A^{\mathbb{N}}$ es derivable

  Se divide la hipótesis en tres casos:

  - $A \in \Gamma$: En dicho caso $A^{\mathbb{N}} \in \Gamma^{\mathbb{N}}$, por lo que $PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash A^{\mathbb{N}}$ es derivable.
  - $A \in PA2^-$: Recordemos que $PA2^-$ es el subconjunto de los axiomas de Peano donde quitamos el de inducción. Esta parte de la prueba la omitiremos, aunque se puede ver que $PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash A^{\mathbb{N}}$ es derivable.
  - $A \equiv \forall x.\, x \in \mathbb{N}$: Si A es el axioma de inducción, es fácil ver que $\vdash A^{\mathbb{N}}$ es derivable ya que $A^{\mathbb{N}} \equiv \forall x.\, x \in \mathbb{N} \Rightarrow x \in \mathbb{N}$, por lo que también vale la tesis.

- **Ley de Peirce.**

  H) $A \equiv ((B \Rightarrow C) \Rightarrow B) \Rightarrow B$
  T) $PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash A^{\mathbb{N}}$ es derivable

Como $A^{\mathbb{N}} \equiv ((B^{\mathbb{N}} \Rightarrow C^{\mathbb{N}}) \Rightarrow B^{\mathbb{N}}) \Rightarrow B^{\mathbb{N}}$, utilizando la Ley de Peirce se deriva trivialmente el secuente $\vdash A^{\mathbb{N}}$, y por lo tanto se concluye la tesis.

- **Introducción del $\Rightarrow$.**

H) $PA2^{-}, \Delta, (\Gamma \cup \{B\})^{\mathbb{N}}, FV_1(C)^{\mathbb{N}} \vdash C^{\mathbb{N}}$ es derivable
T) $PA2^{-}, \Delta, \Gamma^{\mathbb{N}}, FV_1(B \Rightarrow C)^{\mathbb{N}} \vdash (B \Rightarrow C)^{\mathbb{N}}$ es derivable

Basta ver que $(\Gamma \cup \{B\})^{\mathbb{N}} = \Gamma^{\mathbb{N}} \cup B^{\mathbb{N}} \cup FV_1(B)^{\mathbb{N}}$. Luego, utilizando la hipótesis inductiva, tenemos:

$$PA2^{-}, \Delta, \Gamma^{\mathbb{N}}, B^{\mathbb{N}}, FV_1(B)^{\mathbb{N}}, FV_1(C)^{\mathbb{N}} \vdash C^{\mathbb{N}} \text{ es derivable}$$

Aplicando la introducción del $\Rightarrow$ se concluye:

$$PA2^{-}, \Delta, \Gamma^{\mathbb{N}}, FV_1(B)^{\mathbb{N}}, FV_1(C)^{\mathbb{N}} \vdash B^{\mathbb{N}} \Rightarrow C^{\mathbb{N}} \text{ es derivable}$$

Que es equivalente a la tesis.

- **Eliminación del $\Rightarrow$.**

H$_1$) $PA2^{-}, \Delta, \Gamma^{\mathbb{N}}, FV_1(B \Rightarrow A)^{\mathbb{N}} \vdash (B \Rightarrow A)^{\mathbb{N}}$ es derivable
H$_2$) $PA2^{-}, \Delta, \Gamma^{\mathbb{N}}, FV_1(B)^{\mathbb{N}} \vdash B^{\mathbb{N}}$ es derivable
T) $PA2^{-}, \Delta, \Gamma^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash A^{\mathbb{N}}$ es derivable

Por la primera hipótesis inductiva, se observa que:

$$PA2^{-}, \Delta, \Gamma^{\mathbb{N}}, FV_1(B)^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash B^{\mathbb{N}} \Rightarrow A^{\mathbb{N}} \text{ es derivable}$$

Por otro lado, de la segunda se puede probar por debilitamiento que:

$$PA2^{-}, \Delta, \Gamma^{\mathbb{N}}, FV_1(B)^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash B^{\mathbb{N}} \text{ es derivable}$$

Utilizando la eliminación del $\Rightarrow$ con los dos secuentes anteriores, obtenemos:

$$PA2^{-}, \Delta, \Gamma^{\mathbb{N}}, FV_1(B)^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash A^{\mathbb{N}} \text{ es derivable}$$

Luego, para toda $x \in FV(B) \backslash (FV(A) \cup FV(\Gamma))$ de primer orden, como x sólo aparece libre en la fórmula $x \in \mathbb{N}$ del contexto, se puede ver que es admisible removerla. Como conclusión:

$$PA2^{-}, \Delta, \Gamma^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash A^{\mathbb{N}} \text{ es derivable}$$

- **Introducción del $\forall^2$.**

$$\begin{array}{ll}\mathbf{H}_1) & PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash A^{\mathbb{N}} \text{ es derivable}\\ \mathbf{H}_2) & X \notin FV(PA2, \Gamma)\\ \mathbf{T)} & PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(\forall X.\, A)^{\mathbb{N}} \vdash (\forall X.\, A)^{\mathbb{N}} \text{ es derivable}\end{array}$$

Notar que $X \notin FV(PA2^-) = \emptyset$, $X \notin FV(\Delta) = \emptyset$, $X \notin FV(\Gamma^{\mathbb{N}}) = FV(\Gamma)$ por hipótesis, y $X \notin FV(FV_1(A)^{\mathbb{N}})$ por ser de segundo orden.
Por lo tanto, a partir de la primera hipótesis, utilizando la introducción del $\forall^2$ se prueba:

$$PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash \forall X.\, A^{\mathbb{N}} \text{ es derivable}$$

Como $FV_1(A)^{\mathbb{N}} = FV_1(\forall X.\, A)^{\mathbb{N}}$ y $\forall X.\, A^{\mathbb{N}} \equiv (\forall X.\, A)^{\mathbb{N}}$ esto es equivalente a la tesis.

- **Eliminación del $\forall^2$.**

$$\begin{array}{ll}\mathbf{H)} & PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(\forall X.\, A)^{\mathbb{N}} \vdash (\forall X.\, A)^{\mathbb{N}} \text{ es derivable}\\ \mathbf{T)} & PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A[P/X])^{\mathbb{N}} \vdash (A[P/X])^{\mathbb{N}} \text{ es derivable}\end{array}$$

Notar que, por lo observado en la regla anterior, de la hipótesis se obtiene que:

$$PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash \forall X.\, A^{\mathbb{N}} \text{ es derivable}$$

  - **Si** $P \equiv \lambda x_1 \ldots x_k.B$**:** Sea $P^{\mathbb{N}}$ el predicado $P^{\mathbb{N}} := \lambda x_1 \ldots x_k.B^{\mathbb{N}}$, utilizando la eliminación del $\forall^2$ se puede ver que:

$$PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash A^{\mathbb{N}}[P^{\mathbb{N}}/X] \equiv (A[P/X])^{\mathbb{N}} \text{ es derivable}$$

    Debilitando, se pueden agregar al contexto las fórmulas $x \in \mathbb{N}$ para toda variable de primer orden en $FV(A[P/X]) \backslash FV(A)$.
  - **Si** $P \equiv \dot{F}$: Basta aplicar la eliminación del $\forall^2$ para probar que:

$$PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash A^{\mathbb{N}}[\dot{F}/X] \text{ es derivable}$$

    Notando que $FV_1(A)^{\mathbb{N}} = FV_1(A[\dot{F}/X])^{\mathbb{N}}$ y que $A^{\mathbb{N}}[\dot{F}/X] \equiv (A[\dot{F}/X])^{\mathbb{N}}$, se concluye la tesis.

- **Introducción del $\forall^1$.**

$$\begin{array}{ll}\mathbf{H}_1) & PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A)^{\mathbb{N}} \vdash A^{\mathbb{N}} \text{ es derivable}\\ \mathbf{H}_2) & x \notin FV(PA2, \Gamma)\\ \mathbf{T)} & PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(\forall x.\, A)^{\mathbb{N}} \vdash (\forall x.\, A)^{\mathbb{N}} \text{ es derivable}\end{array}$$

Vemos que si $x \notin FV(A)$, por debilitamiento podemos agregar $x \in \mathbb{N}$ al contexto del secuente en la hipótesis inductiva. Por lo tanto, tenemos que:

$$PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(\forall x.\ A)^{\mathbb{N}}, x \in \mathbb{N} \vdash A^{\mathbb{N}} \text{ es derivable}$$

Luego, utilizando la introducción del $\Rightarrow$, obtenemos:

$$PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(\forall x.\ A)^{\mathbb{N}} \vdash x \in \mathbb{N} \Rightarrow A^{\mathbb{N}} \text{ es derivable}$$

Observar que $x \notin FV(PA2^-) = \emptyset$, $x \notin FV(\Delta) = \emptyset$, $x \notin FV(\Gamma^{\mathbb{N}}) = FV(\Gamma)$ por hipótesis y $x \notin FV(FV_1(\forall x.\ A)^{\mathbb{N}})$. Por lo tanto, por la introducción del $\forall^1$, llegamos a que:

$$PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(\forall x.\ A)^{\mathbb{N}} \vdash \forall x.\ x \in \mathbb{N} \Rightarrow A^{\mathbb{N}} \text{ es derivable}$$

Basta observar que $\forall x.\ x \in \mathbb{N} \Rightarrow A^{\mathbb{N}} \equiv (\forall x.\ A)^{\mathbb{N}}$ para concluir la tesis.

- **Eliminación del $\forall^1$.**

H) $PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(\forall x.\ A)^{\mathbb{N}} \vdash (\forall x.\ A)^{\mathbb{N}}$ es derivable
T) $PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A[e/x])^{\mathbb{N}} \vdash (A[e/x])^{\mathbb{N}}$ es derivable

A partir de la hipótesis se obtiene que:

$$PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(\forall x.\ A)^{\mathbb{N}} \vdash \forall x.\ x \in \mathbb{N} \Rightarrow A^{\mathbb{N}} \text{ es derivable}$$

Utilizando la eliminación del $\forall^1$ y que $A^{\mathbb{N}}[e/x] \equiv (A[e/x])^{\mathbb{N}}$ vemos que:

$$PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(\forall x.\ A)^{\mathbb{N}} \vdash e \in \mathbb{N} \Rightarrow (A[e/x])^{\mathbb{N}} \text{ es derivable}$$

Aplicando debilitamiento, como $FV(\forall x.\ A) \subseteq FV(A[e/x])$, se concluye que:

$$PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A[e/x])^{\mathbb{N}} \vdash e \in \mathbb{N} \Rightarrow (A[e/x])^{\mathbb{N}} \text{ es derivable}$$

Bastaría luego derivar el secuente $PA2^-, \Delta, \Gamma^{\mathbb{N}}, FV_1(A[e/x])^{\mathbb{N}} \vdash e \in \mathbb{N}$, o incluso un secuente más débil como $PA2^-, \Delta, FV_1(e)^{\mathbb{N}} \vdash e \in \mathbb{N}$, para probar la tesis (a través de la eliminación del $\Rightarrow$). Este último problema se resuelve con el siguiente lema.

$\square$

Lema 7.4 *Para toda expresión aritmética* $e(x_1, \ldots, x_k)$*, tal que* $FV(e) \subseteq \{x_1, \ldots, x_k\}$*, el secuente* $PA2^-, \Delta, x_1 \in \mathbb{N}, \ldots, x_k \in \mathbb{N} \vdash e(x_1, \ldots, x_k) \in \mathbb{N}$ *es derivable.*

Demostración. Se puede ver por inducción:

1. Si $e \equiv x_i$, entonces $PA2^-, \Delta, x_1 \in \mathbb{N}, \ldots, x_k \in \mathbb{N} \vdash x_i \in \mathbb{N}$ es trivialmente derivable.
2. Si $e \equiv \dot{n}$ para algún n natural, luego $PA2^-, \Delta, x_1 \in \mathbb{N}, \ldots, x_k \in \mathbb{N} \vdash \dot{n} \in \mathbb{N}$ es trivialmente derivable (recordar que $(\dot{n} \in \mathbb{N}) \in \Delta$).
3. Si $e \equiv f(e_1(x_1, \ldots, x_k), \ldots, e_l(x_1, \ldots, x_k))$, se reduce a utilizar las hipótesis inductivas y el siguiente lema.

□

Lema 7.5 (Totalidad de los símbolos de función) *Dado un símbolo de función f de aridad k, el secuente $PA2^- \vdash \forall x_1 \in \mathbb{N} \ldots \forall x_k \in \mathbb{N}.\ f(x_1, \ldots, x_k) \in \mathbb{N}$ es derivable.*

Este lema puede ser probado debido a que los símbolos de función corresponden a funciones recursivas primitivas, ya que así lo establecimos en los axiomas de $PA2^-$. De hecho, esta es la razón por la cual restringimos dichas funciones al presentar la teoría.

7.2. Trabajando con los naturales

Ahora que definimos el concepto de natural en los modelos de realizabilidad, resulta interesante estudiar cuál es su relación con los individuos del modelo de partida $\mathbb{N}$.

Nos centraremos primero en ver si es posible realizar universalmente las fórmulas "$n \in \mathbb{N}$" para todo n natural, lo que implicaría que los naturales de la metateoría lo son en los modelos de realizabilidad. La pregunta que surge es ¿a qué nos referimos con las fórmulas "$n \in \mathbb{N}$"? Observar que hay dos maneras directas de representar los naturales del modelo de partida en las fórmulas:

- Utilizando los parámetros $\dot{n}$.
- Utilizando las expresiones $s^n\, 0$: si n es un natural, $s^n\, 0$ es la expresión aritmética definida como: $s^n\, 0 := \begin{cases} 0 & si\ n = 0 \\ s\ (s^m\ 0) & si\ n = m + 1 \end{cases}$

Notar que $s^n\, 0$ es una expresión aritmética sin parámetros, mientras que $\dot{n}$ existe sólo en el lenguaje extendido. Aún así, lo que tienen en común ambas expresiones es que $[\![s^n\ 0]\!] = n = [\![\dot{n}]\!]$ para todo n natural.

Al estudiar la realizabilidad de dichas fórmulas, veremos que existen realizadores universales que son un conjunto de λ-términos conocidos como los *naturales de Krivine*.

7.2.1. Naturales de Krivine

En el λ-cálculo (y por lo tanto en el λ_c-cálculo), existen distintas maneras de representar los naturales como λ-términos. Una manera usual es codificar cada natural n como una función que recibe dos argumentos (usualmente x y f) y retorna la aplicación de n veces f a x. Estos son conocidos como los naturales de Church y su definición formal es la siguiente:

Definición 7.2 (Naturales de Church) *Sea n un natural, definimos el natural de Church $\hat{n}$ como el λ_c-término $\lambda xf.\, f^n\, x$, donde $f^n\, x$ es un λ_c-término que se define formalmente como* $f^n\, x := \begin{cases} x & si\, n = 0 \\ f\,(f^m\, x) & si\, n = m+1 \end{cases}$

Con dicha idea también se pueden codificar las funciones sucesor, suma y producto de la siguiente manera:

$$\begin{array}{lcll} \text{Sucesor} & \overline{s} & := & \lambda nxf.\, f\,(n\, x\, f) \\ \text{Suma} & \overline{+} & := & \lambda nmxf.\, m\,(n\, x\, f)\, f \\ \text{Producto} & \overline{*} & := & \lambda nmxf.\, m\, x\,(\lambda x.\, n\, x\, f) \end{array}$$

Por ejemplo, si a la función sucesor le pasamos como argumento una función $\hat{n}$, nos queda una función de dos parámetros, x y f, que se comporta como $(\widehat{n+1})$ **pero** que difiere sintácticamente. Profundizaremos luego en dicha relación, pero por ahora nos quedamos con que sintácticamente $\overline{s}\ \hat{0}$ y $\hat{1}$ no son el mismo término.

Si bien luego volveremos a los naturales de Church, utilizaremos ahora otra codificación de los naturales: los naturales de Krivine.

Definición 7.3 (Naturales de Krivine) *Sea n un natural, definimos el natural de Krivine $\overline{n}$ como el λ_c-término $\overline{s}^n\, \hat{0}$.*
Formalmente, $\overline{n} := \begin{cases} \hat{0} & si\, n = 0 \\ \overline{s}\, \overline{m} & si\, n = m+1 \end{cases}$

Observación 7.2 *Se pueden hacer dos observaciones interesantes:*

1. *Es inmediato ver que $\overline{0} \equiv \hat{0}$.*

2. *Tanto $\hat{n}$ como $\overline{n}$, más que ser λ_c-términos, son λ-términos.*

Los naturales de Krivine son interesantes debido a que realizan las fórmulas que afirman que los naturales del modelo inicial, lo son en los modelos de realizabilidad. El siguiente lema es un gran paso en dicha dirección.

Lema 7.6 (Realizando los naturales) *Los términos de prueba* $\overline{0}$ *y* $\overline{s}$ *realizan universalmente las siguientes fórmulas:*

1. $\overline{0} \Vvdash 0 \in \mathbb{N}$

2. $\overline{s} \Vvdash \forall x.\, x \in \mathbb{N} \Rightarrow (s\, x) \in \mathbb{N}$

Demostración.

1. Fijado un polo $\perp\!\!\!\perp$, buscamos ver que:

$$\overline{0} \Vdash 0 \in \mathbb{N} \equiv \forall Z.\, Z(0) \Rightarrow (\forall y.\, Z(y) \Rightarrow Z(s\, y)) \Rightarrow Z(0)$$

Tomemos F una función de falsedad de aridad uno, $t \Vdash \dot{F}(0)$, $\pi \in F(0)$ y $u \Vdash \forall y.\, \dot{F}(y) \Rightarrow \dot{F}(s\, y)$ (observar que $t \cdot u \cdot \pi \in ||0 \in \mathbb{N}||$). Luego:

$$\begin{array}{rcl} \overline{0} \star t \cdot u \cdot \pi & \equiv & \lambda x f.\, x \star t \cdot u \cdot \pi \\ & \succ & t \star \pi \end{array}$$

Como $t \star \pi \in \perp\!\!\!\perp$, concluimos que $\overline{0} \Vdash 0 \in \mathbb{N}$ para todo polo.

2. Fijado un polo $\perp\!\!\!\perp$, tomamos m un natural, $t \Vdash \dot{m} \in \mathbb{N}$, F una función de falsedad de aridad uno, $u \Vdash \dot{F}(0)$, $v \Vdash \forall y.\, \dot{F}(y) \Rightarrow \dot{F}(s\, y)$ y $\pi \in F(m+1)$, por lo que $t \cdot u \cdot v \cdot \pi \in ||\dot{m} \in \mathbb{N} \Rightarrow \dot{F}(0) \Rightarrow (\forall y.\, \dot{F}(y) \Rightarrow \dot{F}(s\, y)) \Rightarrow \dot{F}(s\, \dot{m})|| \subseteq ||\forall x.\, x \in \mathbb{N} \Rightarrow (s\, x) \in \mathbb{N}||$.

Por anti-evaluación, como $\overline{s} \star t \cdot u \cdot v \cdot \pi \succ v \star (t\, u\, v) \cdot \pi$, basta probar que:

$$v \star (t\, u\, v) \cdot \pi \in \perp\!\!\!\perp$$

Debido a que $v \Vdash \forall y.\, \dot{F}(y) \Rightarrow \dot{F}(s\, y)$ y $\pi \in F(m+1)$, nos alcanza con ver que:

$$t\, u\, v \Vdash \dot{F}(\dot{m})$$

Sea $\pi' \in F(m)$, $t\, u\, v \star \pi' \succ t \star u \cdot v \cdot \pi'$ y, como $t \Vdash \dot{m} \in \mathbb{N}$ y $u \cdot v \cdot \pi' \in ||\dot{m} \in \mathbb{N}||$, se concluye que $t \star u \cdot v \cdot \pi' \in \perp\!\!\!\perp$. □

Parece trivial, a partir de este lema, concluir que podemos realizar universalmente aquellas fórmulas que afirman que los naturales del modelo estándar lo son en los modelos de realizabilidad, pero esto no es así. Imaginemos que queremos utilizar lo anterior para demostrar por inducción que $\dot{n} \in \mathbb{N}$ es universalmente realizable para todo n. Tendremos dos "problemas":

1. En el paso base necesitamos ver que $\dot{0} \in \mathbb{N}$ es universalmente realizable, pero el lema nos dice que $0 \in \mathbb{N}$ lo es (no es lo mismo).

2. Suponiendo solucionado el paso anterior, para el paso inductivo quisiéramos obtener un realizador universal de $(n \dot{+} 1) \in \mathbb{N}$ a partir de un realizador universal de $\dot{n} \in \mathbb{N}$. Pero el punto 2 del lema, a partir de un realizador universal de $\dot{n} \in \mathbb{N}$, nos daría un realizador universal de $(s\,\dot{n}) \in \mathbb{N}$ que no es lo mismo que $(n \dot{+} 1) \in \mathbb{N}$.

Por otro lado, si intentamos demostrar que las fórmulas $(s^n\,0) \in \mathbb{N}$ son universalmente realizables, volveríamos a tener complicaciones en el paso inductivo. El problema del lema anterior es que, por el paso 2, $\overline{s} \Vdash \dot{n} \in \mathbb{N} \Rightarrow (s\,\dot{n}) \in \mathbb{N}$ para todo n natural, y la expresión $(s\,\dot{n})$ termina siendo una mezcla entre los naturales $s^n\,0$ y los $\dot{n}$. Aún así, esto se puede corregir aplicando la siguiente observación (cuya prueba es omitida).

Observación 7.3 *Sea A una fórmula, e_1, e_2 dos expresiones aritméticas cerradas tales que $[\![e_1]\!] = [\![e_2]\!]$, entonces $A[e_1/x] \approx_{\perp\!\!\!\perp} A[e_2/x]$ para todo polo $\perp\!\!\!\perp$.*

Esta observación, por ejemplo, nos permite ver que para todo n natural:

$$\overline{s} \Vdash (s^n\,0) \in \mathbb{N} \Rightarrow (s^{n+1}\,0) \in \mathbb{N}$$

Para esto, basta tomar $A \equiv x \in \mathbb{N} \Rightarrow (s\,x) \in \mathbb{N}$ y ver que:

$$\begin{aligned} (s^n\,0) \in \mathbb{N} \Rightarrow (s^{n+1}\,0) \in \mathbb{N} \quad &\equiv \quad A[s^n\,0/x] \\ &\approx \quad A[\dot{n}/x] \\ &\equiv \quad \dot{n} \in \mathbb{N} \Rightarrow (s\,\dot{n}) \in \mathbb{N} \end{aligned}$$

Como $(s^n\,0) \in \mathbb{N} \Rightarrow (s^{n+1}\,0) \in \mathbb{N} \approx \dot{n} \in \mathbb{N} \Rightarrow (s\,\dot{n}) \in \mathbb{N}$ para todo polo, y esta última es realizable universalmente por $\overline{s}$, se concluye que $\overline{s}$ también es un realizador universal de la primera.

Teorema 7.4 *Sea n un natural, entonces $\overline{n} \Vdash (s^n\,0) \in \mathbb{N}$.*

Demostración. Esta demostración se hace por inducción en n:

- Para $n = 0$ tenemos que $\overline{0} \Vdash 0 \in \mathbb{N} \equiv (s^0\,0) \in \mathbb{N}$ por la parte 1 del lema anterior.

- Para el paso inductivo, asumiendo $\overline{n} \Vdash (s^n\,0) \in \mathbb{N}$, veremos que $\overline{n+1} \equiv \overline{s}\,\overline{n}$ es un realizador universal de $(s^{n+1}\,0) \in \mathbb{N}$.

  Usando el lema y la observación previa, vimos que $\overline{s}$ realiza universalmente $(s^n\,0) \in \mathbb{N} \Rightarrow (s^{n+1}\,0) \in \mathbb{N}$. Usando el teorema de adecuación y la hipótesis inductiva, obtenemos $\overline{s}\,\overline{n} \Vdash (s^{n+1}\,0) \in \mathbb{N}$. □

Corolario 7.2 *Sea n un natural, entonces $\overline{n} \Vdash \dot{n} \in \mathbb{N}$.*

Demostración. Se desprende trivialmente del teorema anterior, y de la observación 7.3. □

Como conclusión, debido a que los λ-términos $\overline{n}$ son términos de prueba, las expresiones aritméticas de la forma $s^n\, 0$ son naturales en todos los modelos de realizabilidad.

7.2.2. Naturales de Church

Volviendo a los naturales de Church, veremos que estos también nos sirven como realizadores universales para las fórmulas $(s^n\, 0) \in \mathbb{N}$. Para esto, estudiaremos a $\hat{n}$ y $\overline{n}$ como λ-términos y utilizaremos que son β-equivalentes.

Primero, recordemos que en el λ-cálculo se define la β-reducción como una regla de evaluación que establece que $(\lambda x.\, t)\, u$ se puede reducir a $t[u/x]$. De hecho, a un término de la forma $(\lambda x.\, t)\, u$ le denominamos *redex*.

Además, habíamos visto dos estrategias de evaluación: normal y *whr*, con notaciones $\to_\beta$ y $\to_{whr}$ respectivamente, y que la primera extendía la segunda. En particular, decimos que t está en forma normal cuando no se puede reducir utilizando la estrategia normal (no contiene ninguna *redex*), y que está en forma normal débil cuando no se puede reducir utilizando la estrategia *whr*.

Sumado a esto, vimos que si t y t' son β-equivalentes y t' está en forma normal, entonces $t \to_{\beta*} t'$, y que si la forma normal (débil) existe, entonces es única. No lo demostraremos, pero se puede ver que $\hat{n}$ y $\overline{n}$ son β-equivalentes para todo n. Por lo tanto, utilizando el siguiente teorema vemos que $\hat{n} \Vdash (s^n\, 0) \in \mathbb{N}$ para todo n.

Teorema 7.5 (Naturales a menos de β-equivalencia) *Si t es un λ-término cerrado que es β-equivalente a $\overline{n}$ para algún natural n, entonces $t \Vdash (s^n\, 0) \in \mathbb{N}$.*

Para esta prueba utilizaremos primero un lema que relaciona la evaluación en la KAM, con la estrategia *whr* del λ-cálculo.

Definición 7.4 (Relación $\succeq_{P,G}$) *Definimos $\succeq_{P,G}$ como la relación reflexiva y transitiva más chica que cumple con las reglas $PUSH$ y $GRAB$. En particular, $\succeq_{P,G}$ está contenida en $\succeq$, por lo que si $p_1 \succeq_{P,G} p_2$ se cumple también que $p_1 \succeq p_2$.*

Observemos también que la definición de $\succeq_{P,G}$ no es axiomática como la de $\succeq$: la reducción $\succeq_{P,G}$ necesariamente se descompone como una secuencia de pasos elementales $PUSH$ y $GRAB$.

Lema 7.7 (Weak Head Reduction en la KAM) *Sea t un λ-término cualquiera y u su forma normal débil.*

- *Si $u \equiv \lambda x.w$ para alguna variable x, entonces para cualquier pila π y cualquier sustitución cerrada σ, se cumple que $t[\sigma] \star \pi \succeq_{P,G} u[\sigma] \star \pi$.*
- *Si $u \equiv x\, w_1 \dots w_l$ para alguna variable x, donde l puede ser 0, entonces para cualquier pila π y cualquier sustitución cerrada σ, se cumple que $t[\sigma] \star \pi \succeq_{P,G} \sigma(x) \star w_1[\sigma] \cdots w_l[\sigma] \cdot \pi$.*

Demostración. La prueba se hace por inducción en el largo de la reducción *whr* que lleva t en u.

- Si $t \equiv u$, el resultado es inmediato ya que:
  - Si $u \equiv \lambda x.\, w$: $t[\sigma] \star \pi \succeq_{P,G} u[\sigma] \star \pi$ por reflexividad.
  - Si $u \equiv x\, w_1 \dots w_l$:
    $$t[\sigma] \star \pi \equiv \sigma(x)\, w_1[\sigma]\ \dots w_l[\sigma] \star \pi \succeq_{P,G} \sigma(x) \star w_1[\sigma]\ \cdots w_l[\sigma] \cdot \pi$$
    al utilizar la regla $PUSH$ l veces.
- Si $t \to_{whr} u' \to_{whr*} u$, significa que $t \equiv (\lambda x.t')\, v\, v_1 \dots v_k$ y $u' \equiv t'[v/x]\, v_1 \dots v_k$. Por lo tanto:
  $$\begin{array}{rcl} t[\sigma] \star \pi & \equiv & (\lambda x.t')[\sigma]\, v[\sigma]\, v_1[\sigma] \dots v_k[\sigma] \star \pi \\ & \succ_{P,G} & (\lambda x.t')[\sigma] \star v[\sigma] \cdot v_1[\sigma] \cdots v_k[\sigma] \cdot \pi \\ & \equiv & \lambda x.t'[\sigma, x \leftarrow x] \star v[\sigma] \cdot v_1[\sigma] \cdots v_k[\sigma] \cdot \pi \\ & \succ_{P,G} & t'[\sigma, x \leftarrow v[\sigma]] \star v_1[\sigma] \cdots v_k[\sigma] \cdot \pi \end{array}$$
  - Si $u \equiv \lambda x.\, w$, por hipótesis inductiva sabemos que:
    $$\begin{array}{rcl} u'[\sigma] \star \pi & \equiv & t'[v/x][\sigma]\, v_1[\sigma] \dots v_k[\sigma] \star \pi \\ & \succeq_{P,G} & (\lambda x.\, w)[\sigma] \star \pi \end{array}$$
    Como dicha reducción debe comenzar con k reglas $PUSH$, obtenemos:
    $$\begin{array}{rcl} u'[\sigma] \star \pi & \equiv & t'[v/x][\sigma]\, v_1[\sigma] \dots v_k[\sigma] \star \pi \\ & \succeq_{P,G} & t'[v/x][\sigma] \star v_1[\sigma] \cdots v_k[\sigma] \cdot \pi \\ & \succeq_{P,G} & (\lambda x.\, w)[\sigma] \star \pi \end{array} .$$
  - Si $u \equiv x\, w_1 \dots w_l$, tenemos por hipótesis inductiva que:
    $$\begin{array}{rcl} u'[\sigma] \star \pi & \equiv & t'[v/x][\sigma]\, v_1[\sigma] \dots v_k[\sigma] \star \pi \\ & \succeq_{P,G} & \sigma(x) \star w_1[\sigma] \cdots w_l[\sigma] \cdot \pi \end{array}$$
    Dicha reducción también debe comenzar con k reglas $PUSH$:
    $$\begin{array}{rcl} u'[\sigma] \star \pi & \equiv & t'[v/x][\sigma]\, v_1[\sigma] \dots v_k[\sigma] \star \pi \\ & \succeq_{P,G} & t'[v/x][\sigma] \star v_1[\sigma] \cdots v_k[\sigma] \cdot \pi \\ & \succeq_{P,G} & \sigma(x) \star w_1[\sigma] \cdots w_l[\sigma] \cdot \pi \end{array}$$

Observando que $t'[v/x][\sigma] \equiv t'[\sigma, x \leftarrow v[\sigma]]$ (todo lo de rojo es lo mismo) y utilizando la propiedad transitiva, llegamos a que $t[\sigma] \star \pi \succeq_{P,G} (\lambda x.\, w)[\sigma] \star \pi$ en el primer caso, y que $t[\sigma] \star \pi \succeq_{P,G} \sigma(x) \star w_1[\sigma] \cdots w_l[\sigma] \cdot \pi$ en el segundo. □

Además del lema anterior, usaremos el siguiente lema para finalmente probar el teorema a partir de ambos.

Lema 7.8 *Sea n un natural, F una función de falsedad de aridad uno, t_0 y t_s λ_c-términos cerrados. Si $t_0 \Vdash \dot{F}(0)$ y $t_s \Vdash \forall y.\, \dot{F}(y) \Rightarrow \dot{F}(s\, y)$, entonces para todo λ-término t β-equivalente a $f^n\, x$ y toda sustitución cerrada σ tal que $\sigma(f) \equiv t_s$ y $\sigma(x) \equiv t_0$, se cumple que $t[\sigma] \Vdash \dot{F}(s^n\, 0)$.*

Demostración. La prueba se realiza por inducción en n:

- Si $n = 0$, $t \simeq_\beta x$. Afirmamos que $t \to_{whr*} x$ (que sería entonces su forma normal débil). Luego, si $\pi \in F(0) = ||\dot{F}(s^0\, 0)||$, por el lema anterior $t[\sigma] \star \pi \succeq \sigma(x) \star \pi \equiv t_0 \star \pi \in \perp\!\!\!\perp$, lo que prueba que $t[\sigma] \Vdash \dot{F}(s^0\, 0)$.

  Para ver la afirmación, si $t \simeq_\beta x$ tenemos que $t \to_{\beta*} x$ ya que x está en forma normal. Luego, si u es la forma normal débil de t, sabemos que $t \to_{whr*} u \to_{\beta*} x$. Analizando los dos posibles casos de u:

  - Si $u \equiv y\, w_1 \dots w_l$ para alguna variable y, entonces la única manera que $u \to_{\beta*} x$ es que $l = 0$ y que $y = x$ ($u \equiv x$).
  - Que $u \equiv \lambda y.\, w$ para alguna variable y es absurdo, ya que no se podría dar $u \to_{\beta*} x$.

  Como $u \equiv x$, tenemos lo afirmado ($t \to_{whr*} x$).

- Si $n = m + 1$, $t \simeq_\beta f^{m+1}\, x$. Afirmamos que $t \to_{whr*} f\, t'$ (que sería su forma normal débil) con $t' \simeq_\beta f^m\, x$. Luego, sea $\pi \in F(m+1)$, tenemos por el lema anterior que $t[\sigma] \star \pi \succeq \sigma(f) \star t'[\sigma] \cdot \pi \equiv t_s \star t'[\sigma] \cdot \pi$.

  Por hipótesis inductiva, $t'[\sigma] \Vdash \dot{F}(s^m\, 0)$, por lo que:

  $$\begin{array}{rcccl} t'[\sigma] \cdot \pi & \in & ||\dot{F}(s^m\, 0) \Rightarrow \dot{F}(s^{m+1}\, 0)|| & = & ||\dot{F}(\dot{m}) \Rightarrow \dot{F}(s\, \dot{m})|| \\ & & & \subseteq & ||\forall y.\, \dot{F}(y) \Rightarrow \dot{F}(s\, y)|| \end{array}$$

  Como $t_s \Vdash \forall y.\, \dot{F}(y) \Rightarrow \dot{F}(s\, y)$, entonces $t_s \star t'[\sigma] \cdot \pi \in \perp\!\!\!\perp$ y concluimos por anti-evaluación que $t[\sigma] \star \pi \in \perp\!\!\!\perp$.

  La afirmación se prueba por casos, de la misma manera que la afirmación que hicimos en el paso base: sabemos que $t \to_{whr*} u \to_{\beta*} f\, (f^m\, x)$ con u la forma normal débil de t y, analizando los dos posibles casos, vemos que u tiene que ser $f\, t'$ con $t' \to_{\beta *} f^m\, x$. □

Demostración del Teorema 7.5. Utilizando los lemas anteriores, ahora podemos ver que el conjunto de los realizadores universales de $s^n\, 0$ contiene a los λ-términos cerrados β-equivalentes a $\overline{n}$.

Fijado un polo $\perp\!\!\!\perp$, sea F una función de falsedad de aridad uno, $u \Vdash \dot{F}(0)$, $v \Vdash \forall y.\, \dot{F}(y) \Rightarrow \dot{F}(s\, y)$ y $\pi \in F(n)$ $(u \cdot v \cdot \pi \in ||(s^n\, 0) \in \mathbb{N}||)$, queremos ver que $t \star u \cdot v \cdot \pi \in \perp\!\!\!\perp$.

Como $t \simeq_\beta \overline{n} \simeq_\beta \lambda x f.\, f^n\, x$ y $\lambda x f.\, f^n\, x$ está en forma normal, se cumple que $t \to_{\beta *} \lambda x f.\, f^n\, x$. Más aún, usando los mismos argumentos que en el lema 7.8:

1. $t \to_{whr*} \lambda x.\, t'$ donde $\lambda x.\, t'$ es la forma normal débil de t.
2. $t' \to_{whr*} \lambda f.\, t''$ donde $\lambda f.\, t''$ es la forma normal débil de t'.
3. $t'' \simeq_\beta f^n\, x$.

Aplicando el lema 7.7, junto con los puntos 1 y 2 anteriores, tenemos la siguiente secuencia de reducción para una sustitución cerrada σ cualquiera.

$$
\begin{array}{rcl}
t \star u \cdot v \cdot \pi & \equiv & t[\sigma] \star u \cdot v \cdot \pi \\
 & \succeq & (\lambda x.\, t')[\sigma] \star u \cdot v \cdot \pi \\
 & \succeq & t'[\sigma, x \leftarrow u] \star v \cdot \pi \\
 & \succeq & (\lambda f.\, t'')[\sigma, x \leftarrow u] \star v \cdot \pi \\
 & \succeq & t''[\sigma, x \leftarrow u, f \leftarrow v] \star \pi
\end{array}
$$

Por el lema 7.8, y el punto 3 anterior, vemos que $t''[\sigma, x \leftarrow u, f \leftarrow v] \Vdash \dot{F}(s^n\, 0)$ y concluimos por anti-evaluación. □

7.2.3. Individuos no estándar

Anteriormente, vimos que los naturales del modelo $\mathbb{N}$ lo son en los modelos de realizabilidad henkinianos. La duda que surge naturalmente es si hay otros naturales en los modelos de realizabilidad que no sean de este tipo (naturales no estándar). Lo primero que vemos es que hay individuos (no necesariamente naturales) en algunos de estos modelos que no son naturales del modelo de partida.

Teorema 7.6 (Individuos no estándar) *Sea $(\pi_n)_{n \in \mathbb{N}}$ una enumeración de pilas y G una función de falsedad donde $G(n) := \{\pi_n\}$ para todo n:*

1. *$\texttt{callcc}\,(\lambda kg.\ g\ k) \Vdash\!\!\Vdash \exists x.\ \neg\dot{G}(x) \equiv \forall Z.\ (\forall x.\ \neg\dot{G}(x) \Rightarrow Z) \Rightarrow Z$*

2. *Si para toda constante de pila α, existe $t \in PL \cap \Lambda$ tal que $t \star \alpha \in \perp\!\!\!\perp$, para todo polo y todo natural n, la fórmula $\dot{G}(s^n\,0)$ es realizada por un término de prueba.*

Demostración.

1. Fijado un polo $\perp\!\!\!\perp$, tomamos una función de falsedad F de aridad nula, u un realizador de $\forall x.\ \neg\dot{G}(x) \Rightarrow \dot{F}$ y $\pi \in ||\dot{F}||$ (por lo que $u \cdot \pi \in ||\exists x.\ \neg\dot{G}(x)||$).
   Por anti-evaluación, como $\texttt{callcc}\,(\lambda kg.\ g\ k) \star u \cdot \pi \succ u \star k_{u\cdot\pi} \cdot \pi$ basta probar que:

   $$u \star k_{u\cdot\pi} \cdot \pi \in \perp\!\!\!\perp$$

   Debido a que $u \cdot \pi = \pi_n$ para algún n, utilizando la proposición 4.5 y que $\pi_n \in ||\dot{G}(\dot{n})||$, tenemos que:

   $$k_{u\cdot\pi} \Vdash \dot{G}(\dot{n}) \Rightarrow \bot \equiv \neg\dot{G}(\dot{n})$$

   Por lo tanto, $k_{u\cdot\pi} \cdot \pi \in ||\neg\dot{G}(\dot{n}) \Rightarrow \dot{F}|| \subseteq ||\forall x.\ \neg\dot{G}(x) \Rightarrow \dot{F}||$. Observando que u es un realizador de dicha fórmula, queda probado que:

   $$\texttt{callcc}\,(\lambda kg.\ g\ k) \star u \cdot \pi \in \perp\!\!\!\perp$$

   Como $\texttt{callcc}\,(\lambda kg.\ g\ k) \Vdash \exists x.\neg\dot{G}(x)$ para todo polo, concluimos que:

   $$\texttt{callcc}\,(\lambda kg.\ g\ k) \Vdash\!\!\Vdash \exists x.\neg\dot{G}(x)$$

2. Dado n, queremos construir un término de prueba $u \Vdash \dot{G}(s^n\,0)$ sabiendo que $||\dot{G}(s^n\,0)|| = \{\pi_n\}$. Primero, escribimos π_n como $t_1 \cdots t_k \cdot \alpha$. Luego, por hipótesis, existe t un término de prueba cerrado tal que $t \star \alpha \in \perp\!\!\!\perp$, por lo que basta tomar $u \equiv \lambda x_1, \ldots, x_k.\ t$ para concluir. □

Dado un polo $\perp\!\!\!\perp$, el teorema anterior demuestra que $\exists x.\neg\dot{G}(x)$ es válida en $\mathbb{N}_{\perp\!\!\!\perp}$, lo que significa que existe un individuo satisfaciendo $\neg\dot{G}(x)$. Por otro lado, vemos que si se cumplen ciertas hipótesis sobre el polo, dicho individuo no puede ser un natural del modelo inicial debido a que $\mathbb{N}_{\perp\!\!\!\perp}$ satisface $\dot{G}(s^n\,0)$ para todo n. Por lo tanto, hay algún individuo en $\mathbb{N}_{\perp\!\!\!\perp}$ que no es un natural del modelo estándar. A dichos individuos los denominamos *individuos no estándar*.

Más aún, utilizando la instrucción `quote'` -que veremos en la siguiente sección- podemos ser más precisos y probar que, de hecho, uno de estos individuos no estándar es un natural en $\mathbb{N}_{\perp\!\!\perp}$ (un natural no estándar). Aunque entraremos más en detalle acerca de la importancia de esta instrucción luego, lo único que se debe saber de la instrucción `quote'` para este resultado es su regla de evaluación, la cual indica que $\texttt{quote'} \star t \cdot \pi \succ t \star \overline{n_\pi} \cdot \pi$, donde asumimos una biyección entre $\mathbb{N}$ y Π que lleva a cada pila $\pi \mapsto n_\pi$.

Teorema 7.7 (Naturales no estándar) *Sea $(\pi_n)_{n\in\mathbb{N}}$ la enumeración de pilas inversa a la biyección $(n_\pi)_{\pi\in\Pi}$ utilizada para definir `quote'`, y G una función de falsedad donde $G(n) := \{\pi_n\}$ para todo n, entonces:*

$$\begin{array}{rcl} \lambda f.\, \texttt{quote'}\,(\lambda n.\, \texttt{callcc}\,(\lambda k.\, f\, n\, k)) & \Vdash\!\!\!\vdash & \exists x \in \mathbb{N}.\, \neg \dot{G}(x) \\ & \equiv & \forall Z.\, (\forall x \in \mathbb{N}.\, \neg \dot{G}(x) \Rightarrow Z) \Rightarrow Z \end{array}$$

Demostración. Fijado un polo, tomamos una función de falsedad F de aridad nula, $t \Vdash \forall x \in \mathbb{N}.\, \neg \dot{G}(x) \Rightarrow \dot{F}$ y $\pi \in ||\dot{F}||$ (por lo que $t \cdot \pi \in ||\exists x \in \mathbb{N}.\, \neg \dot{G}(x)||$).

Observamos la siguiente evaluación:

$$\begin{array}{rcl} \lambda f.\texttt{quote'}\,(\lambda n.\, \texttt{callcc}\,(\lambda k.\, f\, n\, k)) \star t \cdot \pi & \succ & \texttt{quote'}\,(\lambda n.\, \texttt{callcc}\,(\lambda k.\, t\, n\, k)) \star \pi \\ & \succ & \lambda n.\, \texttt{callcc}\,(\lambda k.\, t\, n\, k) \star \overline{n_\pi} \cdot \pi \\ & \succ & \texttt{callcc}\,(\lambda k.\, t\, \overline{n_\pi}\, k) \star \pi \\ & \succ & t\, \overline{n_\pi}\, k_\pi \star \pi \\ & \succ & t \star \overline{n_\pi} \cdot k_\pi \cdot \pi \end{array}$$

Por anti-evaluación, basta ver que $t \star \overline{n_\pi} \cdot k_\pi \cdot \pi \in \perp\!\!\perp$, lo cual se desprende de los siguientes puntos:

1. $\overline{n_\pi} \Vdash \dot{n_\pi} \in \mathbb{N}$ debido al corolario 7.2.
2. $k_\pi \Vdash \neg \dot{G}(\dot{n_\pi})$, por la proposición 4.5, utilizando que $\{\pi\} = ||\dot{G}(\dot{n_\pi})||$ (acá se vuelve necesario haber tomado n_π como el inverso de π_n).
3. $\pi \in ||\dot{F}||$.
4. Por los puntos anteriores, $\overline{n_\pi} \cdot k_\pi \cdot \pi \in ||\forall x.\, x \in \mathbb{N} \Rightarrow \neg \dot{G}(x) \Rightarrow \dot{F}||$.
5. $t \Vdash \forall x.\, x \in \mathbb{N} \Rightarrow \neg \dot{G}(x) \Rightarrow \dot{F}$.

$\square$

8. Realizando el Análisis Clásico

Esta sección, que explora la posibilidad de realizar el análisis clásico, si bien se apoya en parte del capítulo tres de la tesis doctoral de Lionel Rieg [1], se basa principalmente en las pruebas e ideas de [9].

La aritmética de Peano permite representar los números racionales como tripletas de naturales (signo, numerador y denominador). Más aún, la aritmética de Peano **de segundo orden** nos permite representar los números reales como conjuntos de números naturales/racionales, por ejemplo a través de los *cortes de Dedekind.*

Definición 8.1 (Cortes de Dedekind) *Un conjunto $A \subset \mathbb{Q}$ es un corte de Dedekind si cumple que:*

- $A \neq \emptyset$
- $A \neq \mathbb{Q}$
- *Si $a \in A$ y $b < a$, entonces $b \in A$*
- *Para todo $a \in A$, existe $a' \in A$ con $a < a'$*

Debido a que no sólo podemos representar los números reales, sino cuantificar sobre ellos, es posible formalizar los reales en la teoría $PA2$. Partiendo de que podemos realizar la teoría $PA2$ (a menos de relativizar los cuantificadores), en esta sección veremos que es posible realizar el análisis clásico, o al menos parte de él.

Ya que gran parte del análisis clásico se puede formalizar utilizando la aritmética de Peano de segundo orden, agregando el axioma de elección dependiente o, incluso, un resultado más débil como es el axioma de elección numerable, estudiaremos cómo realizar dichos axiomas. Para esto, serán necesarias dos instrucciones conocidas como `quote` y `quote'`.

8.1. Las instrucciones `quote` y `quote'`

Para definir estas instrucciones, asumimos una biyección entre los λ_c-términos cerrados y los naturales $t \mapsto n_t$ y otra biyección entre pilas y naturales $\pi \mapsto n_\pi$. En base a esto, quedan definidas las instrucciones a partir de sus reglas de evaluación.

Definición 8.2 (Instrucciones `quote` y `quote'`) *Se definen las instrucciones `quote` y `quote'` con las siguientes reglas de evaluación:*

Quote	$\texttt{quote} \star t \cdot \pi$	$\succ$	$t \star \overline{n_t} \cdot \pi$
Quote'	$\texttt{quote'} \star t \cdot \pi$	$\succ$	$t \star \overline{n_\pi} \cdot \pi$

La clave en esta evaluación es que tanto `quote` como `quote'` deben ser inyectivas: λ_c términos diferentes deben generar naturales n_t diferentes (y lo mismo para las pilas). Esto sugiere ver dichas instrucciones como una función de *hashing* que no tiene colisiones. Más aún, el natural asociado a un mismo término no tiene por qué ser único, lo único importante es que dos términos distintos generen naturales distintos, sin importar si un mismo término genera distintos naturales en distintos momentos. Esta segunda observación sugiere ver a `quote` como, por ejemplo, un reloj o una función aleatoria que no repite naturales.

8.2. Axioma de Elección Numerable

Mientras que la instrucción `quote` permite realizar el axioma de elección dependiente, la instrucción `quote'` permite realizar el de elección numerable. En lo que viene, veremos cómo se realizan ambos pero con mayor detenimiento en el segundo caso.

Definición 8.3 (Axioma de elección numerable) *En general, el axioma se formaliza a través del siguiente esquema de axiomas, donde x es una variable de primer orden, X es una variable de segundo orden de aridad uno, Y una variable de segundo orden de aridad dos y A una fórmula con $FV(A) \subseteq \{x, X\}$.*

$$(\forall x.\, \exists X.\, A) \Rightarrow \exists Y.\, \forall x.\, A[\lambda y.\, Y(x,y)/X]$$

Recordemos que la sustitución $A[\lambda y.\, Y(x,y)/X]$ no es más que sustituir todas las ocurrencias de $X(e)$ en A, por $Y(x,e)$.

Gracias a los cortes de Dedekind, pensaremos las variables de segundo orden y aridad uno como reales. A las variables de segundo orden de aridad dos como funciones de individuos en reales, a las de aridad tres como funciones de pares de individuos en reales, y así. Siguiendo esto, el axioma anterior nos dice que si para todo individuo x, existe un real X tal que se cumple A, entonces podemos tomar una función Y tal que para todo x, vale A cambiando X por el real $Y(x)$.

Entendemos Y como una función que da un testigo para $\exists X.\, A$, por lo que al esquema presentado anteriormente lo podemos reformular para obtener un esquema más flexible, donde la fórmula A no necesariamente tenga que ser satisfacible para cualquier x.

$$\exists Y.\, \forall x.\, (\exists X.\, A) \Rightarrow A[\lambda y.\, Y(x,y)/X]$$

En este caso, la función Y nos da un testigo sólo para aquellos x que satisfacen $\exists X.\, A$. En nuestro caso, será más fácil trabajar con la forma contrarrecíproca a la anterior:

$$\exists Y.\, \forall x.\, A[\lambda y.\, Y(x,y)/X] \Rightarrow \forall X.\, A$$

Para esto basta tomar el contrarrecíproco de la forma anterior, y pensar la fórmula A de esta última forma como la fórmula $\neg A$ de la forma anterior.

La intuición de este contrarrecíproco será la que nos acompañará en el resto de la sección: en este caso, la función Y nos da un testigo tal que si A vale para dicho real, entonces vale para cualquiera.

Es interesante ver la relación entre esta idea y *la paradoja de los bebedores*. Esta paradoja es un teorema de lógica clásica que afirma que "en todo bar, hay una persona tal que si esa persona está bebiendo, todo el bar lo está haciendo". Esto se formaliza a través de la fórmula $\exists x.\, \phi(x) \Rightarrow \forall y.\, \phi(y)$, la cual es una tautología en lógica clásica para cualquier ϕ, aunque no es válida en lógica intuicionista. De hecho, la prueba hace uso del tercer excluido (y, en particular, no nos dice quién es dicho bebedor): si todo el bar está bebiendo, se puede elegir a cualquiera, sino es así, basta elegir a alguien que no está tomando.

En nuestro caso, la función Y nos da lo que sería el bebedor que hace que todos beban en la paradoja de los bebedores: si se cumple A para dicho $Y(x)$, se cumple para cualquier X.

Empezamos analizando el comportamiento de `quote'`, para luego realizar el axioma de elección numerable (todas las fórmulas del esquema).

Proposición 8.1 *Dado un polo, para toda fórmula A tal que $FV(A) \subseteq \{x, X\}$, donde x es una variable de primer orden y X es una variable de segundo orden de aridad uno, existe una función de falsedad P de aridad tres tal que:*

$$\texttt{quote'} \Vdash \forall x.\, (\forall y.\, y \in \mathbb{N} \Rightarrow A[\lambda z.\, \dot{P}(x,y,z)/X]) \Rightarrow \forall X.\, A$$

Demostración. La clave de la prueba es encontrar la función P. Para esto se utiliza que el axioma de elección numerable vale en el modelo de partida.

Fijado un polo, sea n un natural y $\pi \in \Pi$, sabemos que si $\pi \in ||\forall X.\, A[\dot{n}/x]||$, existe una función de falsedad unaria F tal que $\pi \in ||A[\dot{n}/x][\dot{F}/X]||$. Por lo tanto, aplicando el axioma de elección numerable en el modelo, para toda π existe una función $U_\pi : \mathbb{N} \to (\mathbb{N} \to \mathfrak{P}(\Pi))$ tal que si $\pi \in ||\forall X.\, A[\dot{n}/x]||$, entonces

$\pi \in ||A[\dot{n}/x][\dot{U_\pi(n)}/X]||$. Básicamente, $U_\pi(n)$ es la función de falsedad F que nombramos antes.

Definimos entonces $P : \mathbb{N} \times \Pi \times \mathbb{N} \to \mathfrak{P}(\Pi)$ como $P(n, \pi, k) := U_\pi(n)(k)$. Luego, utilizamos la biyección $(\pi \mapsto n_\pi)$ entre $\mathbb{N}$ y Π para pensar dicha función como $P : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathfrak{P}(\Pi)$.

Es importante observar que, informalmente, $A[\lambda z.\dot{P}(x, y, z)/X]$ significa sustituir las ocurrencias de $X(e_1)$ por $\dot{P}(x, y, e_1)$, por lo que se puede ver que para $n \in \mathbb{N}$ y $\pi \in \Pi$ cualesquiera:

$$||A[\lambda z.\ \dot{P}(\dot{n}, \dot{n_\pi}, z)/X]||_\rho = ||A[\dot{U_\pi(n)}/X]||_\rho$$

Básicamente, en el lado izquierdo sustituimos toda ocurrencia $X(e_1)$ por $\dot{P}(\dot{n}, \dot{n_\pi}, e_1)$ cuya interpretación es $P(n, n_\pi, [\![e_1]\!]_\rho) = U_\pi(n)([\![e_1]\!]_\rho)$, mientras que en el derecho sustituimos toda ocurrencia $X(e_1)$ por $\dot{U_\pi(n)}(e_1)$ cuya interpretación también es $U_\pi(n)([\![e_1]\!]_\rho)$. En particular, se cumple que:

$$||A[\lambda z.\ \dot{P}(x, y, z)/X][\dot{n}/x][\dot{n_\pi}/y]|| = ||A[\dot{n}/x][\dot{U_\pi(n)}/X]||$$

De aquí, si $\pi \in ||\forall X. A[\dot{n}/x]||$ para algún individuo n, entonces habíamos definido U_π para que $\pi \in ||A[\dot{n}/x][\dot{U_\pi(n)}/X]||$ y se concluye que:

$$\text{si } \pi \in ||\forall X.\ A[\dot{n}/x]||, \text{ entonces } \pi \in ||A[\lambda z.\ \dot{P}(x, y, z)/X][\dot{n}/x][\dot{n_\pi}/y]||$$

Para demostrar que `quote'` $\Vdash \forall x.\ (\forall y.\ y \in \mathbb{N} \Rightarrow A[\lambda z.\ \dot{P}(x, y, z)/X]) \Rightarrow \forall X.\ A$, tomamos $n \in \mathbb{N}$, y probamos:

$$\texttt{quote'} \Vdash (\forall y.\ y \in \mathbb{N} \Rightarrow A[\lambda z.\ \dot{P}(x, y, z)/X][\dot{n}/x]) \Rightarrow \forall X.\ A[\dot{n}/x]$$

Para esto, dados $t \Vdash \forall y.\ y \in \mathbb{N} \Rightarrow A[\lambda z.\ \dot{P}(x, y, z)/X][\dot{n}/x]$ y $\pi \in ||\forall X.\ A[\dot{n}/x]||$, como $\texttt{quote'} \star t \cdot \pi \succ t \star \overline{n_\pi} \cdot \pi$, alcanza con probar que $t \star \overline{n_\pi} \cdot \pi \in \perp\!\!\!\perp$.

Esto último es trivial debido a que:

1. $t \Vdash \dot{n_\pi} \in \mathbb{N} \Rightarrow A[\lambda z.\ \dot{P}(x, y, z)/X][\dot{n}/x][\dot{n_\pi}/y]$.

2. $\overline{n_\pi} \Vdash \dot{n_\pi} \in \mathbb{N}$ debido al corolario 7.2.

3. $\pi \in ||A[\lambda z.\ \dot{P}(x, y, z)/X][\dot{n}/x][\dot{n_\pi}/y]||$ ya que $\pi \in ||\forall X.\ A[\dot{n}/x]||$.

$\square$

Este primer teorema se acerca a lo que buscamos. Pensando $P(x, y)$ como una función que dado dos individuos retorna un real, sabemos que para todo x, si A vale para la sucesión de reales $(P(x, n))_{n \in \mathbb{N}}$, entonces vale para todo real. Aún así, buscamos obtener un único real en lugar de una sucesión. Esto se puede lograr utilizando un símil de la paradoja de los bebedores, que lo realizaremos de la siguiente manera:

Lema 8.1 *Sean X e Y variables de segundo orden de aridad dos, existe un λ_c-término t de prueba, tal que:*

$$t \Vdash \forall X.\, \exists Y.\, (Y \; is \; Func) \wedge (\forall x.\, (\forall y.\, Y(x,y) \Rightarrow X(x,y)) \Rightarrow \forall y \in \mathbb{N}.\, X(x,y))$$

donde definimos la fórmula

$$Y \; is \; Func := \forall x.\, \forall y.\, \forall z.\, Y(x,y) \wedge Y(x,z) \Rightarrow y = z$$

Demostración. El lema es trivial debido a que la fórmula es derivable en lógica clásica de segundo orden, basta tomar $Y(x,y)$ como la fórmula que indica que y es el primer natural para el cual $\neg X(x,y)$. Más específicamente:

$$Y(x,y) := y \in \mathbb{N} \wedge \neg X(x,y) \wedge \forall p \in \mathbb{N}.\forall q \in \mathbb{N}.\, p + (s\, q) = y \Rightarrow X(x,p)$$

□

Si bien ya tenemos los ingredientes necesarios, en lugar de realizar el esquema de axiomas de elección numerable, realizaremos el esquema de axiomas CCA (el cual es equivalente).

Definición 8.4 (Axioma CCA) *Se define el esquema de axiomas CCA como el conjunto de fórmulas de la forma:*

$$\exists Z.\, \exists Y.\, (Y \; is \; Func) \wedge (\forall x.\, (\forall y.\, Y(x,y) \Rightarrow A[\lambda z.\, Z(x,y,z)/X]) \Rightarrow \forall X.\, A)$$

donde x, y son variables de primer orden, X es una variable de segundo orden de aridad uno, Y una variable de segundo orden de aridad dos, Z una variable de segundo orden de aridad tres, y A es una fórmula con $FV(A) \subseteq \{x, X\}$.

Vale la pena comparar dicho esquema con la forma que dimos del axioma de elección numerable. Primero, si eliminamos la hipótesis $Y \, is Func$ del esquema CCA, obtenemos:

$$\exists Z.\, \exists Y.\, \forall x.\, (\forall y.\, Y(x,y) \Rightarrow A[\lambda z.\, Z(x,y,z)/X]) \Rightarrow \forall X.\, A$$

Podemos pensar, de manera informal, al $\forall y.\, Y(x,y) \Rightarrow A[\lambda z.\, Z(x,y,z)/X]$ como $A[\lambda z.\, Z(x, Y(x), z)/X]$. Viendo a Y como una función de individuos en individuos, el enunciado quedaría:

$$\exists Z.\, \exists Y.\, \forall x.\, A[\lambda z.\, Z(x, Y(x), z)/X] \Rightarrow \forall X.\, A$$

Que, componiendo las funciones Z e Y, lo podríamos ver como el esquema dado para el axioma de elección numerable:

$$\exists Y.\, \forall x.\, A[\lambda y.\, Y(x,y)/X] \Rightarrow \forall X.\, A$$

Como mencionamos, el esquema de axiomas CCA es equivalente a CAC (axioma de elección numerable) y es realizado por un término que no depende de A (lo vemos en el siguiente teorema). Aún así, para derivar CAC a partir de CCA se necesita el esquema de extensionalidad:

$$\forall X.\, \forall Y.\, (\forall x.\, X(x) \Leftrightarrow Y(x)) \Rightarrow (A[\lambda x.\, X(x)/Z] \Leftrightarrow A[\lambda x.\, Y(x)/Z])$$

Este teorema es probado por inducción en A, y su realizador universal depende de A. Luego, el realizador de $CCA \Rightarrow CAC$ depende de A, por lo que el realizador de CAC obtenido también lo hace. Finalizamos este análisis realizando el esquema de axiomas CCA.

Teorema 8.1 (Realizador de CCA) *Existe un término de prueba que, para todo x, y, X, Y, Z y A como en la definición anterior, realiza universalmnete a:*

$$\exists Z.\, \exists Y.\, (Y\ is\ Func) \wedge (\forall x.\, (\forall y.\, Y(x,y) \Rightarrow A[\lambda z.\, Z(x,y,z)/X]) \Rightarrow \forall X.\, A)$$

Demostración. Analicemos primero el enunciado. Para simplificar, siempre pensaremos en un individuo fijo x para el cual depende todo lo demás. Informalmente, la fórmula indica que existen funciones Z (de individuos en reales) e Y (de individuos en individuos), tales que si vale $A[Z(Y(x))/X]$ (esta notación es informal, y refiere a cambiar X por el real $Z(Y(x))$) entonces vale $\forall X.\, A$. O sea, si vale A cambiando X por un real particular $Z(Y(x))$, entonces vale para todo X.

Recordemos que, para cualquier A como en las hipótesis, vimos que existe una función de falsedad P (de aridad tres) tal que:

$$\texttt{quote'} \Vdash \forall x.\, (\forall y \in \mathbb{N}.\, A[\lambda z.\, \dot{P}(x,y,z)/X]) \Rightarrow \forall X.\, A$$

Pensando en que el Z que buscamos es P, la fórmula realizada nos dice casi lo que necesitamos: si $A[Z(y)/X]$ vale para todo y natural, entonces vale $\forall X.\, A$. Simplificando, tenemos una sucesión de reales $Z(y)$ tales que, si vale A cambiando X por esos $Z(y)$, entonces vale para todo real X.

Para encontrar un solo real que sea suficiente como muestra de que A vale para cualquiera, podemos utilizar la paradoja de los bebedores: hay uno de todos esos reales $Z(y)$, tal que si para él vale $A[Z(y)/X]$, entonces para todos los demás reales de la sucesión también. Dicho y será $Y(x)$. Luego, si vale $A[Z(Y(x))/X]$, vale $A[Z(y)/X]$ para todo y natural y, por lo que vimos antes, vale $\forall X.\, A$.

Yendo a la prueba, dada una fórmula A como en las hipótesis y fijado un polo cualquiera, tomamos P la función de falsedad dada por la proposición 8.1 y vemos que el realizador buscado es Φ definido de la siguiente manera:

$$\Phi := \lambda w.\, w\,(\lambda x.\, t\,(\lambda y.\, x\,(\lambda z.\, y\,(\lambda ab.\, z\, a\,(\lambda c.\, \texttt{quote'}\,(b\, c))))))$$

donde, gracias al lema 8.1, podemos tomar t tal que:

$$t \Vdash\!\Vdash \forall X.\, \exists Y.\, (Y\ is\ Func) \wedge (\forall x.\, (\forall y.\, Y(x,y) \Rightarrow X(x,y)) \Rightarrow \forall y \in \mathbb{N}.\, X(x,y))$$

lo que implica en particular que si definimos:

$$F_P(n,m) := ||A[\lambda z.\, \dot{P}(x,y,z)/X][\dot{n}/x][\dot{m}/y]||$$

obtenemos que:

$$t \Vdash \exists Y.\, (Y\ is\ Func) \wedge (\forall x.\, (\forall y.\, Y(x,y) \Rightarrow \dot{F_P}(x,y)) \Rightarrow \forall y \in \mathbb{N}.\, \dot{F_P}(x,y))$$

Esto último es básicamente lo que mencionamos de la paradoja de los bebedores: Y será la función que nos indica el real $P(x, Y(x))$ que si bebe (satisface A) todos los demás beben (A es satisfacible por todo $P(x,y)$, con $y \in \mathbb{N}$).

Para ordenar la prueba, abreviamos:

$$\begin{array}{lcl} B_1 & := & Y\ is\ Func \\ B_2 & := & \forall x.\, (\forall y.\, Y(x,y) \Rightarrow A[\lambda z.\, Z(x,y,z)/X]) \Rightarrow \forall X.\, A \\ B_3 & := & \forall x.\, (\forall y.\, Y(x,y) \Rightarrow \dot{F_P}(x,y)) \Rightarrow \forall y \in \mathbb{N}.\, \dot{F_P}(x,y) \end{array}$$

Primero observamos que $FV(B_1) = \{Y\}$, $FV(B_2) \subseteq \{Y, Z\}$ y $FV(B_3) = \{Y\}$. Con estas definiciones, buscamos ver que $\Phi \Vdash \exists Z.\, \exists Y.\, B_1 \wedge B_2$, asumiendo que t realiza $\exists Y.\, B_1 \wedge B_3$.

Parte 1. $\Phi \Vdash \exists Z.\, \exists Y.\, B_1 \wedge B_2$

Lo primero a observar es que para probar que $\Phi \Vdash \exists Z.\, \exists Y.\, B_1 \wedge B_2$, nos basta con ver que $\lambda x.\, t\,(\lambda y.\, x\,(\lambda z.\, y\,(\lambda ab.\, z\, a\,(\lambda c.\, \texttt{quote'}\,(b\, c))))) \Vdash \exists Y.\, B_1 \wedge B_2[\dot{P}/Z]$ (esto se prueba en la **parte 2**).

Si este fuera el caso, como $\exists Z. \exists Y. B_1 \wedge B_2 \equiv \forall W_1. (\forall Z. (\exists Y. B_1 \wedge B_2) \Rightarrow W_1) \Rightarrow W_1$, tomamos una función de falsedad de aridad nula H_1, $v_1 \Vdash \forall Z.\, (\exists Y.\, B_1 \wedge B_2) \Rightarrow \dot{H_1}$ y $\pi_1 \in H_1$, y vemos que $\Phi \star v_1 \cdot \pi_1 \in \perp\!\!\!\perp$.

$$\begin{array}{rcl} \Phi \star v_1 \cdot \pi_1 & \equiv & \lambda w.\, w\,(\lambda x.\, t\,(\lambda y.\, x\,(\lambda z.\, y\,(\lambda ab.\, z\, a\,(\lambda c.\, \texttt{quote'}\,(b\, c)))))) \star v_1 \cdot \pi_1 \\ & \succ & v_1\,(\lambda x.\, t\,(\lambda y.\, x\,(\lambda z.\, y\,(\lambda ab.\, z\, a\,(\lambda c.\, \texttt{quote'}\,(b\, c)))))) \star \pi_1 \\ & \succ & v_1 \star (\lambda x.\, t\,(\lambda y.\, x\,(\lambda z.\, y\,(\lambda ab.\, z\, a\,(\lambda c.\, \texttt{quote'}\,(b\, c)))))) \cdot \pi_1 \end{array}$$

Como $v_1 \Vdash \forall Z. (\exists Y. B_1 \wedge B_2) \Rightarrow \dot{H_1}$, en particular $v_1 \Vdash (\exists Y. B_1 \wedge B_2[\dot{P}/Z]) \Rightarrow \dot{H_1}$. Sumado a que $\lambda x.\, t\,(\lambda y.\, x\,(\lambda z.\, y\,(\lambda ab.\, z\, a\,(\lambda c.\, \texttt{quote'}\,(b\, c))))) \Vdash \exists Y.\, B_1 \wedge B_2[\dot{P}/Z]$ y $\pi_1 \in H_1$, se concluye que $\Phi \star v_1 \cdot \pi_1 \in \perp\!\!\!\perp$.

Parte 2. $\lambda x.\, t\, (\lambda y.\, x\, (\lambda z.\, y\, (\lambda ab.\, z\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c))))) \Vdash \exists Y.\, B_1 \wedge B_2[\dot{P}/Z]$

Recordemos que $\exists Y.\, B_1 \wedge B_2[\dot{P}/Z] \equiv \forall W_2.\, (\forall Y.\, B_1 \wedge B_2[\dot{P}/Z] \Rightarrow W_2) \Rightarrow W_2$. Por lo que tenemos que probar que si tomamos una función de falsedad H_2 de aridad nula, $v_2 \Vdash \forall Y.\, B_1 \wedge B_2[\dot{P}/Z] \Rightarrow \dot{H}_2$ y $\pi_2 \in H_2$, se cumple que:

$$\lambda x.\, t\, (\lambda y.\, x\, (\lambda z.\, y\, (\lambda ab.\, z\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c))))) \star v_2 \cdot \pi_2 \in \perp\!\!\!\perp$$

Como $\lambda x.\, t\, (\lambda y.\, x\, (\lambda z.\, y\, (\lambda ab.\, z\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c))))) \star v_2 \cdot \pi_2 \succ$
$t \star (\lambda y.\, v_2\, (\lambda z.\, y\, (\lambda ab.\, z\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c))))) \cdot \pi_2$
y $t \Vdash \exists Y.\, B_1 \wedge B_3$, basta con ver que $(\lambda y.\, v_2\, (\lambda z.\, y\, (\lambda ab.\, z\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c))))) \cdot \pi_2$ se encuentra en el valor de falsedad de dicha fórmula.

Parte 3. $(\lambda y.\, v_2\, (\lambda z.\, y\, (\lambda ab.\, z\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c))))) \cdot \pi_2 \in ||\exists Y.\, B_1 \wedge B_3||$

Sabiendo que $\exists Y.\, B_1 \wedge B_3 \equiv \forall W_3.\, (\forall Y.\, B_1 \wedge B_3 \Rightarrow W_3) \Rightarrow W_3$, y sustituyendo W_3 por la función de falsedad H_2, nos bastaría ver que:

$$(\lambda y.\, v_2\, (\lambda z.\, y\, (\lambda ab.\, z\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c))))) \cdot \pi_2 \in ||(\forall Y.\, B_1 \wedge B_3 \Rightarrow \dot{H}_2) \Rightarrow \dot{H}_2||$$

Como $\pi_2 \in H_2$, es suficiente con que $\lambda y.\, v_2\, (\lambda z.\, y\, (\lambda ab.\, z\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c))))$ realice $\forall Y.\, B_1 \wedge B_3 \Rightarrow \dot{H}_2$. Para esto, dada una función de falsedad F_Y de aridad dos, un realizador $v_3 \Vdash B_1[\dot{F_Y}/Y] \wedge B_3[\dot{F_Y}/Y]$ y una pila $\pi_3 \in H_2$, vemos que:

$$(\lambda y.\, v_2\, (\lambda z.\, y\, (\lambda ab.\, z\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c))))) \star v_3 \cdot \pi_3 \succ$$
$$v_2 \star (\lambda z.\, v_3\, (\lambda ab.\, z\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c)))) \cdot \pi_3$$

Recordar que $v_2 \Vdash \forall Y.\, B_1 \wedge B_2[\dot{P}/Z] \Rightarrow \dot{H}_2$, por lo tanto, en particular se cumple que $v_2 \Vdash B_1[\dot{F_Y}/Y] \wedge B_2[\dot{P}/Z][\dot{F_Y}/Y] \Rightarrow \dot{H}_2$. Como $\pi_3 \in H_2$, para que este último proceso esté en el polo, faltaría probar que $\lambda z.\, v_3\, (\lambda ab.\, z\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c)))$ es un realizador de $B_1[\dot{F_Y}/Y] \wedge B_2[\dot{P}/Z][\dot{F_Y}/Y]$.

Parte 4. $\lambda z.\, v_3\, (\lambda ab.\, z\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c))) \Vdash B_1[\dot{F_Y}/Y] \wedge B_2[\dot{P}/Z][\dot{F_Y}/Y]$

$B_1[\dot{F_Y}/Y] \wedge B_2[\dot{P}/Z][\dot{F_Y}/Y] \equiv \forall W_4.\, (B_1[\dot{F_Y}/Y] \Rightarrow B_2[\dot{P}/Z][\dot{F_Y}/Y] \Rightarrow W_4) \Rightarrow W_4$. Por lo tanto, tomamos una función de falsedad H_4 de aridad nula, v_4 un realizador de $B_1[\dot{F_Y}/Y] \Rightarrow B_2[\dot{P}/Z][\dot{F_Y}/Y] \Rightarrow \dot{H}_4$ y $\pi_4 \in H_4$.

Sabiendo que $\lambda z.\, v_3\, (\lambda ab.\, z\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c))) \star v_4 \cdot \pi_4 \succ$
$v_3 \star (\lambda ab.\, v_4\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c))) \cdot \pi_4$
es suficiente con probar que $v_3 \star (\lambda ab.\, v_4\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c))) \cdot \pi_4 \in \perp\!\!\!\perp$.

Como $v_3 \Vdash B_1[\dot{F_Y}/Y] \wedge B_3[\dot{F_Y}/Y] \equiv \forall W_5.\, (B_1[\dot{F_Y}/Y] \Rightarrow B_3[\dot{F_Y}/Y] \Rightarrow W_5) \Rightarrow W_5$, sustituyendo W_5 por la función H_4, es suficiente con ver que $\lambda ab. v_4 a (\lambda c. \texttt{quote'}\,(bc))$ realiza a $B_1[\dot{F_Y}/Y] \Rightarrow B_3[\dot{F_Y}/Y] \Rightarrow \dot{H}_4$ ya que $\pi_4 \in H_4$.

Parte 5. $\lambda ab.\, v_4\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c)) \Vdash B_1[\dot{F_Y}/Y] \Rightarrow B_3[\dot{F_Y}/Y] \Rightarrow \dot{H}_4$

Tomando $v_5 \Vdash B_1[\dot{F_Y}/Y]$, $v_6 \Vdash B_3[\dot{F_Y}/Y]$, y $\pi_5 \in H_4$, observamos que:

$$\begin{aligned} \lambda ab.\, v_4\, a\, (\lambda c.\, \texttt{quote'}\, (b\, c)) \star v_5 \cdot v_6 \cdot \pi_5 &\succ v_4\, v_5\, (\lambda c.\, \texttt{quote'}\, (v_6\, c)) \star \pi_5 \\ &\succ v_4 \star v_5 \cdot (\lambda c.\, \texttt{quote'}\, (v_6\, c)) \cdot \pi_5 \end{aligned}$$

Por anti-evaluación, es suficiente con ver que el último proceso está en el polo. Recordando que $v_4 \Vdash B_1[\dot{F_Y}/Y] \Rightarrow B_2[\dot{P}/Z][\dot{F_Y}/Y] \Rightarrow \dot{H}_4$, $v_5 \Vdash B_1[\dot{F_Y}/Y]$ y que $\pi_5 \in H_4$, basta con probar que $\lambda c.\, \texttt{quote'}\, (v_6\, c) \Vdash B_2[\dot{P}/Z][\dot{F_Y}/Y]$.

Parte 6. $\lambda c.\, \texttt{quote'}\, (v_6\, c) \Vdash B_2[\dot{P}/Z][\dot{F_Y}/Y]$
$\equiv \forall x.\, (\forall y.\, \dot{F_Y}(x,y) \Rightarrow A[\lambda z.\, \dot{P}(x,y,z)/X]) \Rightarrow \forall X.\, A$

Sea n un natural cualquiera, buscamos ver que:

$$\lambda c.\, \texttt{quote'}\, (v_6\, c) \Vdash (\forall y.\, \dot{F_Y}(\dot{n},y) \Rightarrow A[\lambda z.\, \dot{P}(x,y,z)/X][\dot{n}/x]) \Rightarrow \forall X.\, A[\dot{n}/x]$$

Tomando $v_7 \Vdash \forall y.\, \dot{F_Y}(\dot{n},y) \Rightarrow A[\lambda z.\, \dot{P}(x,y,z)/X][\dot{n}/x]$ y $\pi_6 \in ||\forall X.\, A[\dot{n}/x]||$, vemos que $\lambda c.\, \texttt{quote'}\, (v_6\, c) \star v_7 \cdot \pi_6 \succ \texttt{quote'} \star (v_6\, v_7) \cdot \pi_6$. Además, debido a la proposición 8.1, sabemos que:

$$\texttt{quote'} \Vdash (\forall y \in \mathbb{N}.\, A[\lambda z.\, \dot{P}(x,y,z)/X][\dot{n}/x]) \Rightarrow \forall X.\, A[\dot{n}/x]$$

Por lo tanto, quedaría probar que $v_6\, v_7 \Vdash \forall y \in \mathbb{N}.\, A[\lambda z.\, \dot{P}(x,y,z)/X][\dot{n}/x]$.

Parte 7. $v_6\, v_7 \Vdash \forall y \in \mathbb{N}.\, A[\lambda z.\, \dot{P}(x,y,z)/X][\dot{n}/x]$

Como $v_6 \Vdash B_3[\dot{F_Y}/Y] \equiv \forall x.\, (\forall y.\, \dot{F_Y}(x,y) \Rightarrow \dot{F_P}(x,y)) \Rightarrow \forall y \in \mathbb{N}.\, \dot{F_P}(x,y)$, tenemos en particular:

$$v_6 \Vdash (\forall y.\, \dot{F_Y}(\dot{n},y) \Rightarrow \dot{F_P}(\dot{n},y)) \Rightarrow \forall y \in \mathbb{N}.\, \dot{F_P}(\dot{n},y)$$

Luego, usando que $F_P(n,m) := ||A[\lambda z.\, \dot{P}(x,y,z)/X][\dot{n}/x][\dot{m}/y]||$, se puede probar que:

$$v_6 \Vdash (\forall y.\, \dot{F_Y}(\dot{n},y) \Rightarrow A[\lambda z.\, \dot{P}(x,y,z)/X][\dot{n}/x]) \Rightarrow \forall y \in \mathbb{N}.\, A[\lambda z.\, \dot{P}(x,y,z)/X][\dot{n}/x]$$

Recordando que $v_7 \Vdash \forall y.\, \dot{F_Y}(\dot{n},y) \Rightarrow A[\lambda z.\, \dot{P}(x,y,z)/X][\dot{n}/x]$, se concluye trivialmente.

$\square$

8.3. Axioma de Elección Dependiente

De forma similar al axioma de elección numerable, el axioma de elección dependiente se define a partir de un esquema de axiomas:

Definición 8.5 (Axioma de elección dependiente) *Se define el axioma de elección dependiente como el conjunto de fórmulas de la forma:*

$$(\forall X.\ \exists Y.\ A) \Rightarrow \forall Z.\ \exists S.\ (\forall y.\ Z(y) \Leftrightarrow S(0,y)) \\ \wedge\ (\forall k \in \mathbb{N}.\ A[\lambda y.\ S(k,y)/X][\lambda y.\ S(s\,k,y)/Y])$$

donde y y k son variables de primer orden, X, Y y Z son variables de segundo orden de aridad uno, S es una variable de segundo orden de aridad dos y A es una fórmula con $FV(A) \subseteq \{X,Y\}$.

Si pensamos otra vez el segundo orden como los reales, tenemos que si para todo X existe un Y tal que vale A, entonces para cualquier valor inicial Z, existe una sucesión S tal que $S(0) = Z$ y para todo k natural vale A cambiando X por $S(k)$ e Y por $S(k+1)$ (lo notaremos $A(S(k), S(k+1))$).

Dicho esquema es realizable utilizando una estrategia similar a la utilizada en el axioma de elección numerable. Básicamente, se realiza un esquema equivalente:

$$\forall Z.\ \exists D.\ (\forall x.\ \forall y.\ Z(y) \Leftrightarrow D(0,x,y)) \wedge (\forall x.\ \forall k. \\ (\forall n \in \mathbb{N}.\ A[\lambda y.\ D(k,x,y)/X][\lambda y.\ D(s\,k, <n,x>, y)/Y]) \\ \Rightarrow \forall Y.\ A[\lambda y.\ D(k,x,y)/X])$$

Donde $< n, x >$ es un símbolo de función de aridad dos cuya representación es una inyección de $\mathbb{N}^2$ en $\mathbb{N}$. Podemos pensar dicho enunciado como que para todo valor inicial Z, existe una función de aridad dos -de individuos en reales- tal que $D(0,x) = Z$ para todo x, y para todo par $(k,\ x)$ siempre que valga la sentencia $A(D(k,x), D(k+1, < n, x >))$ para todo n natural, entonces vale $A(D(k,x), Y)$ para cualquier Y.

Realizando este esquema, se realiza el esquema del axioma de elección dependiente ya que se puede derivar a partir de este (pensando en A como el $\neg A$ del otro). La derivación, a grandes rasgos, es la siguiente:

1. Se asume que vale $\forall X.\ \exists Y.\ A(X,Y)$, y se toma un valor inicial Z.

2. Utilizando dicho Z en la fórmula realizada, e instanciandola con la negación de A, sabemos que existe D tal que $D(0,x) = Z$ para todo x y que cumple que, para todo par (k,x), si no se cumple $A(D(k,x), D(k+1, < n, x >))$ para todo n natural, no existe Y tal que $A(D(k,x), Y)$ valga.

3. Dados x y k, como asumimos que existe un Y tal que $A(D(k,x),Y)$ vale, tiene que existir un $m_{k,x}$ tal que $A(D(k,x),D(k+1,< m_{k,x},x >))$ se cumpla.

4. Luego, se define inductivamente la sucesión $(n_k)_{k\in\mathbb{N}}$ como $n_0 := 0$ y $n_{k+1} :=\ < m_{k,n_k}, n_k >$. De esta manera, $A(D(k,n_k),D(k+1,n_{k+1}))$ vale para todo k. Por lo tanto, tomando $S(k)$ como $D(k,n_k)$ obtenemos que:

$$\forall k \in \mathbb{N}.\ A[\lambda y.\ S(k,y)/X][\lambda y.\ S(s\,k,y)/Y]$$

y también que:

$$\forall y.\ Z(y) \Leftrightarrow S(0,y)$$

Para realizar el nuevo esquema, se utiliza el siguiente teorema, el cual evidencia el tipo de `quote`.

Teorema 8.2 *Sean x,k variables de primer orden y X e Y variables de segundo orden de aridad uno. Entonces, para todo polo y toda función de falsedad F de aridad uno, existe P una función de falsedad de aridad tres, tal que $P(0,x,y) = F(y)$ y:*

$$\texttt{quote} \Vdash \forall x.\ \forall k.\ (\forall n \in \mathbb{N}.\ A[\lambda y.\ \dot{P}(k,x,y)/X][\lambda y.\ \dot{P}(s\,k,< n,x >,y)/Y]) \\ \Rightarrow \forall Y.\ A[\lambda y.\ \dot{P}(k,x,y)/X]$$

9. El Problema de Especificación

En esta sección nos centraremos en un problema de cierta manera inverso a lo que hemos visto previamente. En las secciones anteriores, vimos cómo los realizadores de cierta manera justifican las fórmulas, determinando una teoría: la teoría de las fórmulas realizadas. Ahora analizaremos cómo las fórmulas condicionan a sus realizadores. Particularmente, estudiaremos el problema de especificación: intentaremos caracterizar los realizadores universales de una fórmula a través de su comportamiento computacional. Por ejemplo, se puede ver que los realizadores de la fórmula $1 \equiv \forall Z.\, Z \Rightarrow Z$ son exactamente aquellos términos que, en cierto sentido, se comportan como la función identidad.

Proposición 9.1 (Especificación de 1**)** *Sea* t *un* λ_c*-término cerrado, entonces* $t \Vdash\!\!\vdash 1$ *si, y sólo si,* $t \star u \cdot \pi \succ u \star \pi$ *para todo* $u \in \Lambda$, $\pi \in \Pi$.

Demostración.
($\Longrightarrow$) Sea t un realizador universal de 1, u un término cerrado y π una pila, hay que ver que $t \star u \cdot \pi \succ u \star \pi$. Al tomar el polo $\perp\!\!\!\perp := \{p \mid p \succeq u \star \pi\}$, basta con ver que $t \star u \cdot \pi \in \perp\!\!\!\perp$ (debido a la definición del polo y que $t \star u \cdot \pi \not\equiv u \star \pi$). Como $u \cdot \pi \in ||1||_{\perp\!\!\!\perp}$ (recordando la proposición 4.1) y $t \Vdash_{\perp\!\!\!\perp} 1$, obtenemos que $t \star u \cdot \pi \in \perp\!\!\!\perp$.

($\Longleftarrow$) Fijado un polo, buscamos probar $t \Vdash 1$. Dado $\pi \in ||1||$, por la proposición 4.1 sabemos que $\pi \equiv u \cdot \pi'$ con $u \in \Lambda$ y $\pi' \in \Pi$ tales que $u \star \pi' \in \perp\!\!\!\perp$. Por lo tanto, $t \star u \cdot \pi' \succ u \star \pi' \in \perp\!\!\!\perp$ y se concluye por anti-evaluación. $\square$

Esta proposición ilustra claramente el tipo de resultados que buscamos obtener. No parece extraño, debido a que la realizabilidad es -como ya dijimos- un tipado más laxo, con foco en el comportamiento computacional de los términos, que el "tipo" de los términos implique que estos tengan un comportamiento particular.

Similar a la especificación de la fórmula 1, podemos ver la especificación de $\bot \Rightarrow \bot$. Como 1 es un subtipo de $\bot \Rightarrow \bot$ para todo polo, los realizadores universales de 1 están contenidos en los realizadores universales de $\bot \Rightarrow \bot$, por lo que el comportamiento que caracterice a estos últimos debe ser más general que comportarse como la identidad.

Proposición 9.2 (Especificación de $\bot \Rightarrow \bot$**)** *Sea* t *un* λ_c*-término cerrado, entonces* $t \Vdash\!\!\vdash \bot \Rightarrow \bot$ *si, y sólo si, para todo* $u \in \Lambda$, $\pi \in \Pi$, *existe* $\pi' \in \Pi$ *tal que* $t \star u \cdot \pi \succeq u \star \pi'$.

Demostración.
($\Longrightarrow$) Similar a la especificación de 1, dados $u \in \Lambda$ y $\pi \in \Pi$, queremos ver que

$t \star u \cdot \pi \in \perp\!\!\!\perp$, donde $\perp\!\!\!\perp := \{p \mid \exists \pi'.\, p \succeq u \star \pi'\}$. Para esto basta con probar que $u \cdot \pi \in ||\bot \Rightarrow \bot||_{\perp\!\!\!\perp} = |\bot|_{\perp\!\!\!\perp} \cdot ||\bot||_{\perp\!\!\!\perp} = |\bot|_{\perp\!\!\!\perp} \cdot \Pi$, lo cual se reduce trivialmente a probar $u \in |\bot|_{\perp\!\!\!\perp}$. Tomando $\pi'' \in \Pi = ||\bot||_{\perp\!\!\!\perp}$ cualquiera, observamos que $u \star \pi'' \in \perp\!\!\!\perp$, concluyendo $u \in |\bot|_{\perp\!\!\!\perp}$.

($\Longleftarrow$) Fijado un polo, buscamos ver que $t \star \pi \in \perp\!\!\!\perp$ para toda $\pi \in ||\bot \Rightarrow \bot||$. Observar que podemos escribir $\pi \equiv u \cdot \pi'$ donde $u \Vdash \bot$ y, por lo tanto, $u \star \pi'' \in \perp\!\!\!\perp$ para toda $\pi'' \in \Pi$. Como $t \star \pi \succeq u \star \pi''$ para alguna pila π'', se concluye por anti-evaluación.

□

Otro caso interesante de especificación son los *booleanos*, los cuales se definen de manera similar a como definimos los naturales. Veremos que podemos realizar universalmente las fórmulas $0 \in \mathbb{B}$ y $(s\,0) \in \mathbb{B}$ (donde $\mathbb{B}$ será el conjunto de *booleanos*), y que dichos realizadores también realizan una fórmula más general conocida como *Bool*.

Proposición 9.3 (Especificación de booleanos) *Sea b una expresión aritmética, definimos la fórmula $b \in \mathbb{B} := \forall Z.\, Z(0) \Rightarrow Z(s\,0) \Rightarrow Z(b)$. Luego, para todo λ_c-término cerrado t:*

1. *$t \Vvdash 0 \in \mathbb{B}$ si, y sólo si, $t \star u \cdot v \cdot \pi \succ u \star \pi$ para todo $u, v \in \Lambda$ y $\pi \in \Pi$.*

2. *$t \Vvdash (s\,0) \in \mathbb{B}$ si, y sólo si, $t \star u \cdot v \cdot \pi \succ v \star \pi$ para todo $u, v \in \Lambda$ y $\pi \in \Pi$.*

3. *Para todo $n \geq 2$, no existe un realizador universal de $(s^n\,0) \in \mathbb{B}$.*

Demostración.

1. La estructura de la prueba es similar a la de las pruebas de la especificación de 1 y de $\bot \Rightarrow \bot$.

   ($\Longrightarrow$) Sean t, u, v y π como en el enunciado, necesitamos ver que $t \star u \cdot v \cdot \pi \in \perp\!\!\!\perp$ donde $\perp\!\!\!\perp := \{p \,|\, p \succeq u \star \pi\}$, y para esto basta probar que $u \cdot v \cdot \pi \in ||0 \in \mathbb{B}||_{\perp\!\!\!\perp}$. Tomando F la función de falsedad de aridad uno que cumple que $F(0) = \{\pi\}$ y $F(n) = \emptyset$ para cualquier otro n, vemos que:

   a) $u \Vdash_{\perp\!\!\!\perp} \dot{F}(0)$, porque $u \star \pi \in \perp\!\!\!\perp$ y $\{\pi\} = ||\dot{F}(0)||_{\perp\!\!\!\perp}$.

   b) $v \Vdash_{\perp\!\!\!\perp} \dot{F}(s\,0)$ ya que $||\dot{F}(s\,0)||_{\perp\!\!\!\perp} = \emptyset$.

   c) $\pi \in ||\dot{F}(0)||_{\perp\!\!\!\perp}$.

   De lo anterior, se concluye que $u \cdot v \cdot \pi \in ||\dot{F}(0) \Rightarrow \dot{F}(s\,0) \Rightarrow \dot{F}(0)||_{\perp\!\!\!\perp} \subseteq ||\forall Z.\, Z(0) \Rightarrow Z(s\,0) \Rightarrow Z(0)||_{\perp\!\!\!\perp} = ||0 \in \mathbb{B}||_{\perp\!\!\!\perp}$.

Al igual que en la prueba de la especificación de 1, al probar que $t \star u \cdot v \cdot \pi \in \perp\!\!\!\perp$, estamos probando que $t \star u \cdot v \cdot \pi \succeq u \star \pi$, lo que implica $t \star u \cdot v \cdot \pi \succ u \star \pi$ ya que $u \cdot v \cdot \pi \not\equiv \pi$.

($\Longleftarrow$) Fijado un polo, sea $u \cdot v \cdot \pi \in ||0 \in \mathbb{B}||$ (por lo que $u \star \pi \in \perp\!\!\!\perp$), por hipótesis $t \star u \cdot v \cdot \pi \succ u \star \pi$, y concluimos por anti-evaluación que $t \star u \cdot v \cdot \pi \in \perp\!\!\!\perp$.

2. La prueba es análoga a la anterior pero, en el directo, se toma la función G tal que $G(1) = \{\pi\}$ y $G(n) = \emptyset$ para cualquier otro n.

3. Asumimos que existe $t \Vdash (s^n\, 0) \in \mathbb{B}$ con $n \geq 2$, lo que implica en particular que $t \Vdash_{\perp\!\!\!\perp} (s^n\, 0) \in \mathbb{B}$ para $\perp\!\!\!\perp := \emptyset$. Observando que $||(s^n\, 0) \in \mathbb{B}||_{\perp\!\!\!\perp} \neq \emptyset$, se llega a un absurdo ($t \star \pi \in \emptyset$ para alguna pila π).
Para ver que $||(s^n\, 0) \in \mathbb{B}||_{\perp\!\!\!\perp} \neq \emptyset$, alcanza con tomar una función de falsedad F tal que $F(0) = F(1) = \emptyset$ y $F(n) = \Pi$, luego $||\dot{F}(0) \Rightarrow \dot{F}(s\, 0) \Rightarrow \dot{F}(s^n\, 0)||_{\perp\!\!\!\perp} = \Pi \cdot \Pi \cdot \Pi \subseteq ||(s^n\, 0) \in \mathbb{B}||_{\perp\!\!\!\perp}$. □

Corolario 9.1 *Las fórmulas* $0 \in \mathbb{B}$ *y* $(s\, 0) \in \mathbb{B}$ *son universalmente realizables por términos de prueba.*

Demostración. Basta tomar $t_0 \equiv \lambda xy.\, x$ y $t_1 \equiv \lambda xy.\, y$, y observar que $t_0 \Vdash 0 \in \mathbb{B}$ ($t_0 \star u \cdot v \cdot \pi \succ u \star \pi$ para todo u, v y π), mientras que $t_1 \Vdash (s\, 0) \in \mathbb{B}$ por lo mismo. □

Como anunciamos, los realizadores universales de los *booleanos* son también realizadores de una fórmula más general conocida como *Bool*.

Proposición 9.4 *Sea* $Bool := \forall Z.Z \Rightarrow Z \Rightarrow Z$ *y* t *un* λ_c*-término cerrado, entonces* $t \Vdash Bool$ *si, y sólo si,* $t \star u \cdot v \cdot \pi \succ u \star \pi$ *o* $t \star u \cdot v \cdot \pi \succ v \star \pi$ *para todo* $u, v \in \Lambda$ *y* $\pi \in \Pi$.

Demostración.
($\Longrightarrow$) Sea t un realizador universal de *Bool*, $u, v \in \Lambda$ y $\pi \in \Pi$, tomamos $\perp\!\!\!\perp := \{p \mid p \succeq u \star \pi \text{ o } p \succeq v \star \pi\}$ y vemos que $u \cdot v \cdot \pi \in ||Bool||_{\perp\!\!\!\perp}$, concluyendo que $t \star u \cdot v \cdot \pi \in \perp\!\!\!\perp$. Lo anterior ($u \cdot v \cdot \pi \in ||Bool||_{\perp\!\!\!\perp}$) se debe a que $u \Vdash_{\perp\!\!\!\perp} \dot{\pi}$ y $v \Vdash_{\perp\!\!\!\perp} \dot{\pi}$, por lo que $u \cdot v \cdot \pi \in ||\dot{\pi} \Rightarrow \dot{\pi} \Rightarrow \dot{\pi}||_{\perp\!\!\!\perp} \subseteq ||Bool||_{\perp\!\!\!\perp}$.

($\Longleftarrow$) Fijado un polo, tomamos $u \cdot v \cdot \pi \in ||Bool||$ y probamos que $t \star u \cdot v \cdot \pi \in \perp\!\!\!\perp$. Esto es trivial ya que, por hipótesis, $t \star u \cdot v \cdot \pi \succ u \star \pi$, o $t \star u \cdot v \cdot \pi \succ v \star \pi$ y tanto $u \star \pi$ como $v \star \pi$ pertenecen al polo (si $u \cdot v \cdot \pi \in ||Bool|| = ||\forall Z.Z \Rightarrow Z \Rightarrow Z||$, entonces $u \star \pi \in \perp\!\!\!\perp$ y $v \star \pi \in \perp\!\!\!\perp$).
□

9.1. Especificación de la Ley de Peirce

Más interesante que los ejemplos anteriores, es determinar el comportamiento computacional de los realizadores de una fórmula clásica como lo es la Ley de Peirce. Este resultado, que se debe a Guillermo y Miquel, es sumamente interesante y establece que los realizadores universales de dicha fórmula coinciden con las estrategias ganadoras de un juego. En lo que queda de la sección presentamos parte de los resultados y demostraciones de [10].

Hasta ahora, el único realizador universal de la Ley de Peirce que conocemos es `callcc`, por lo que es importante ver primero otros realizadores con el fin de generar una idea más general de cómo se comportan todos aquellos términos que realizan dicha fórmula. Más específicamente, veremos realizadores que son similares a `callcc` pero que generan constantes de continuación "perezosas" que no necesariamente funcionan en su primer uso.

9.1.1. Realizadores perezosos de la Ley de Peirce

Dado un par de naturales (n, p) tal que $1 \leq p \leq n$ y $x_0, k, x_1, \ldots, x_n$ $n+2$ variables distintas, definimos para todo $1 \leq i \leq n$ el término de prueba $K^i_{n,p}$ (no cerrado) como:

$$K^i_{n,p} := \begin{cases} \lambda x_n.\, k\, x_p & si\ i = n \\ \lambda x_i.\, k\, (x_0\, K^{i+1}_{n,p}) & si\ i < n \end{cases}$$

Pronto veremos el porqué de estas definiciones, pero por ahora es suficiente con pensar en dichos términos como las constantes de continuación "perezosas" que serán generadas por los nuevos realizadores universales de la Ley de Peirce ($K^i_{n,p}$ sería la constante que funciona en su n-ésimo uso, y la i indica que ya se utilizó $i-1$ veces).

Es fácil ver que $FV(K^i_{n,p}) \subseteq \{x_0, k\} \cup \{x_p \text{ si } p < i\} \subseteq \{x_0, k, x_1, \ldots, x_{i-1}\}$ para todo $1 \leq i \leq n$. Particularmente, sea $1 \leq i \leq n$ y $u_0, l, u_1, \ldots, u_{i-1} \in \Lambda$, notamos $K^i_{n,p}[u_0, l, u_1, \ldots, u_{i-1}]$ a la sustitución $K^i_{n,p}[u_0/x_0][l/k][u_1/x_1] \cdots [u_{i-1}/x_{i-1}]$. Observar que, por lo anterior, $K^i_{n,p}[u_0, l, u_1, \ldots, u_{i-1}]$ es un término cerrado.

A partir de dichas constantes, para todo par (n, p) de naturales (otra vez con $1 \leq p \leq n$), se define el término de prueba cerrado $\texttt{callcc}_{n,p}$ como:

$$\texttt{callcc}_{n,p} := \lambda x_0.\, \texttt{callcc}\, (\lambda k.\, x_0\, K^1_{n,p})$$

Observamos que es trivialmente cerrado ya que $FV(K^1_{n,p}) \subseteq \{x_0, k\}$. El comportamiento de dichos realizadores, y de las constantes de continuación perezosas, queda evidenciado gracias a la siguiente proposición.

Proposición 9.5 *Sea (n,p) un par de naturales con $1 \leq p \leq n$, $u_0,\ldots,u_n \in \Lambda$ y $\pi_0,\ldots,\pi_n \in \Pi$, entonces:*

$$\begin{array}{rcccccl} \mathtt{callcc}_{n,p} & \star & u_0 \cdot \pi_0 & \succ & u_0 & \star & K^1_{n,p}[u_0, k_{\pi_0}] \cdot \pi_0 \\ K^i_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_{i-1}] & \star & u_i \cdot \pi_i & \succ & u_0 & \star & K^{i+1}_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_i] \cdot \pi_0 \\ K^n_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_{n-1}] & \star & u_n \cdot \pi_n & \succ & u_p & \star & \pi_0 \end{array}$$

Para entender el comportamiento de los nuevos realizadores es mejor pensar en el caso donde $p = n$. En la siguiente explicación, simplificamos notación y escribimos sólo $K^i_{n,n}$ en lugar de explicitar las sustituciones:

1. Sean $u_0 \in \Lambda$ y $\pi_0 \in \Pi$, al utilizar `callcc` obteníamos que $\mathtt{callcc} \star u_0 \cdot \pi_0 \succ u_0 \star k_{\pi_0} \cdot \pi_0$, donde k_{π_0} nos permitía recuperar la pila π_0. En este nuevo caso, tenemos a priori algo similar: $\mathtt{callcc}_{n,n} \star u_0 \cdot \pi_0 \succ u_0 \star K^1_{n,n} \cdot \pi_0$.

2. La diferencia entre k_{π_0} y $K^1_{n,n}$ es fácilmente apreciable. Sean $u_1 \in \Lambda$ y $\pi_1 \in \Pi$ cualesquiera, mientras k_{π_0} nos permite recuperar la pila, $K^1_{n,n}$ falla en este cometido. Específicamente, $K^1_{n,n} \star u_1 \cdot \pi_1 \succ u_0 \star K^2_{n,n} \star \pi_0$.

   Si observamos el proceso resultante, es el mismo que obtuvimos al ejecutar $\mathtt{callcc}_{n,n}$ pero, en lugar de $K^1_{n,n}$, tenemos $K^2_{n,n}$. En cierto sentido, $\mathtt{callcc}_{n,n}$ nos generó una constante de continuación fallida y, al utilizarla, volvemos hacia atrás y nos da otra.

3. Al intentar utilizar $K^2_{n,n}$ para retomar la pila π_0, fallará también y nos llevará de nuevo a la misma situación, pero ahora con $K^3_{n,n}$ en lugar de $K^2_{n,n}$. Esto seguirá así hasta llegar a $K^n_{n,n}$ que funcionará como esperamos: $K^n_{n,n} \star u_n \cdot \pi_n \succ u_n \star \pi_0$.

4. De cierta manera, como anunciábamos, $K^i_{n,p}$ es una constante de continuación perezosa. Vemos que si tomamos otro p, lo único que cambia es que al utilizar la última constante, $K^n_{n,p} \star u_n \cdot \pi_n \succ u_p \star \pi_0$. O sea, la constante no sólo retoma la pila π_0, sino que modifica el término por uno con el que hayamos invocado una constante fallida. En otras palabras, después de $n-1$ usos fallidos de la constante de continuación perezosa, la n-ésima ejecución se comporta como si la ejecución número p hubiese funcionado correctamente.

Demostración de la proposición 9.5. La demostración es directa, el primer punto se debe a la siguiente evaluación:

$$\begin{array}{rcl} \mathtt{callcc}_{n,p} \star u_0 \cdot \pi_0 & \equiv & \lambda x_0.\ \mathtt{callcc}\ (\lambda k.\ x_0\ K^1_{n,p}) \star u_0 \cdot \pi_0 \\ & \succ & \mathtt{callcc}\ (\lambda k.\ u_0\ K^1_{n,p}[u_0]) \star \pi_0 \\ & \succ & \lambda k.\ u_0\ K^1_{n,p}[u_0] \star k_{\pi_0} \cdot \pi_0 \\ & \succ & u_0\ K^1_{n,p}[u_0, k_{\pi_0}] \star \pi_0 \\ & \succ & u_0 \star K^1_{n,p}[u_0, k_{\pi_0}] \cdot \pi_0 \end{array}$$

Para el segundo punto, cuando $1 \leq i < n$, tenemos la siguiente evaluación:

$$\begin{array}{rcl} K^i_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_{i-1}] \star u_i \cdot \pi_i & \equiv & \lambda x_i.\, k_{\pi_0}(u_0\, K^{i+1}_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_{i-1}]) \star u_i \cdot \pi_i \\ & \succ & k_{\pi_0}(u_0\, K^{i+1}_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_i]) \star \pi_i \\ & \succ & u_0\, K^{i+1}_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_i] \star \pi_0 \\ & \succ & u_0 \star K^{i+1}_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_i] \cdot \pi_0 \end{array}$$

El último punto, lo dividimos en dos casos:

Si $p < n$. $$\begin{array}{rcl} K^n_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_{n-1}] \star u_n \cdot \pi_n & \equiv & \lambda x_n.\, k_{\pi_0}\, u_p \star u_n \cdot \pi_n \\ & \succ & k_{\pi_0}\, u_p \star \pi_n \\ & \succ & u_p \star \pi_0 \end{array}$$

Si $p = n$. $$\begin{array}{rcl} K^n_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_{n-1}] \star u_n \cdot \pi_n & \equiv & \lambda x_n.\, k_{\pi_0}\, x_n \star u_n \cdot \pi_n \\ & \succ & k_{\pi_0}\, u_n \star \pi_n \\ & \succ & u_n \star \pi_0 \end{array}$$

□

Utilizando este comportamiento, podemos ver que estos términos son efectivamente realizadores universales de la Ley de Peirce.

Proposición 9.6 (Realizadores de la Ley de Peirce) *Sea (n, p) un par de naturales, con $1 \leq p \leq n$, A y B fórmulas cerradas, $\texttt{callcc}_{n,p} \Vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A$*

Demostración. Fijado un polo, sean $u_0 \Vdash (A \Rightarrow B) \Rightarrow A$ y $\pi_0 \in ||A||$, queremos ver que $\texttt{callcc}_{n,p} \star u_0 \cdot \pi_0 \in \perp\!\!\!\perp$.

Por la proposición anterior $\texttt{callcc}_{n,p} \star u_0 \cdot \pi_0 \succ u_0 \star K^1_{n,p}[u_0, k_{\pi_0}] \cdot \pi_0$, por lo tanto, por anti-evaluación y utilizando que $u_0 \Vdash (A \Rightarrow B) \Rightarrow A$ y que $\pi_0 \in ||A||$, es suficiente con probar que $K^1_{n,p}[u_0, k_{\pi_0}] \Vdash A \Rightarrow B$.

Para esto, veremos algo más general: dados $u_0 \Vdash (A \Rightarrow B) \Rightarrow A$ y $\pi_0 \in ||A||$, para todo $1 \leq i \leq n$ se cumple la siguiente afirmación:

Sean $u_1, \ldots, u_{i-1} \in \Lambda$, tales que $u_j \Vdash A$ para todo $1 \leq j < i$, entonces $K^i_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_{i-1}] \Vdash A \Rightarrow B$.

Observamos que tomando $i = 1$ obtenemos lo buscado ($K^1_{n,p}[u_0, k_{\pi_0}] \Vdash A \Rightarrow B$). La afirmación se prueba para todo $1 \leq i \leq n$ en orden inverso:

- Si $i = n$, tomamos $u_n \cdot \pi_n \in ||A \Rightarrow B||$ y buscamos ver que:

$$K^n_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_{n-1}] \star u_n \cdot \pi_n \in \perp\!\!\!\perp$$

Por la proposición anterior, sabemos que:

$$K^n_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_{n-1}] \star u_n \cdot \pi_n \succ u_p \star \pi_0$$

Como $\pi_0 \in ||A||$ y $u_p \Vdash A$ (si $p = n$ porque tomamos $u_n \cdot \pi_n \in ||A \Rightarrow B||$, y si $1 \leq p < n$ por hipótesis), $u_p \star \pi_0 \in \perp\!\!\!\perp$ y se concluye por anti-evaluación.

- Si $i < n$, asumimos que la afirmación se cumple para $i + 1$. Luego, si tomamos $u_i \cdot \pi_i \in ||A \Rightarrow B||$, por la proposición anterior:

$$K^i_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_{i-1}] \star u_i \cdot \pi_i \succ u_0 \star K^{i+1}_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_i] \cdot \pi_0$$

Ahora, como $K^{i+1}_{n,p}[u_0, k_{\pi_0}, u_1, \ldots, u_i] \Vdash A \Rightarrow B$ por hipótesis para $i + 1$, $\pi_0 \in ||A||$ y $u_0 \Vdash (A \Rightarrow B) \Rightarrow A$, se concluye la pertenencia al polo por anti-evaluación.

□

Estos términos nos dan una idea más general del comportamiento de los realizadores de la Ley de Peirce. Para especificar esta ley, al menos bajo ciertas hipótesis, intentaremos representar el comportamiento de los realizadores $\texttt{callcc}_{n,p}$ a través de un juego.

9.1.2. El juego $\mathbb{G}_0$

El juego $\mathbb{G}_0$, que nos permitirá conceptualizar de alguna manera el comportamiento que vimos recién, es un juego entre dos jugadores $\exists$ (jugador) y $\forall$ (oponente) que comienza con una *fase de iniciación*. En dicha fase, $\exists$ presenta un término $t_0 \in \Lambda$ con la intención de realizar la Ley de Peirce, mientras $\forall$ presenta un *manejador* (del inglés *handler*) $(u_0, \pi_0) \in \Lambda \times \Pi$ cuyo fin es ser utilizado por $\exists$ para comunicarle sus movimientos a $\forall$. El resto del juego es parametrizado por el manejador (u_0, π_0), y empieza en la $\exists$-posición $t_0 \star u_0 \cdot \pi_0$.

En este juego, las $\exists$-posiciones son procesos, mientras que las $\forall$-posiciones son términos cerrados que corresponden al último $\exists$-movimiento. El juego alterna $\exists$-movimientos y $\forall$-movimientos de la siguiente manera:

- $\exists$-movimiento: Empezando por una $\exists$-posición p, el jugador $\exists$ evalúa el proceso p con el fin de alcanzar uno de los siguientes estados:
  - Si $p \succ u \star \pi_0$, donde u es la primera componente de un par (u, π) jugado previamente por $\forall$ (el manejador (u_0, π_0) no cuenta como un movimiento), entonces $\exists$ gana.

- Si $p \succ u_0 \star t \cdot \pi_0$, entonces $\exists$ le puede comunicar su movimiento t a $\forall$, en cuyo caso t se vuelve la posición de $\forall$. Notar que dicho término t no necesariamente es único, por lo tanto $\exists$ puede elegir cualquiera de dichos términos.

- $\forall$-movimiento: Empezando en una $\forall$-posición t, el oponente $\forall$ puede jugar cualquier par $(u, \pi) \in \Lambda \times \Pi$. Luego, el proceso $p \equiv t \star u \cdot \pi$ se vuelve la nueva $\exists$-posición.

Cuando un juego es infinito, o $\exists$ se queda sin movimientos para hacer, consideramos que el ganador es $\forall$.

En este juego teórico, es fácil ver que partiendo de $\texttt{callcc}_{n,p}$ existe una estrategia ganadora para $\exists$ en $2n + 1$ movimientos:

- En la fase inicial, $\exists$ presenta $\texttt{callcc}_{n,p}$ y $\forall$ un manejador (u_0, π_0) cualquiera.
- En el primer movimiento de $\exists$, su $\exists$-posición es $\texttt{callcc}_{n,p} \star u_0 \cdot \pi_0$. Como $\texttt{callcc}_{n,p} \star u_0 \cdot \pi_0 \succ u_0 \star K^1_{n,p}[u_0, k_{\pi_0}] \cdot \pi_0$, $\exists$ puede jugar $K^1_{n,p}[u_0, k_{\pi_0}]$.
- Luego, la posición de $\forall$ es $K^1_{n,p}[u_0, k_{\pi_0}]$, y su movimiento puede ser un par (u_1, π_1) cualquiera.
- Previo al segundo movimiento de $\exists$, la $\exists$-posición es $K^1_{n,p}[u_0, k_{\pi_0}] \star u_1 \cdot \pi_1$, y como $K^1_{n,p}[u_0, k_{\pi_0}] \star u_1 \cdot \pi_1 \succ u_0 \star K^2_{n,p}[u_0, k_{\pi_0}, u_1] \cdot \pi_0$, $\exists$ puede jugar $K^2_{n,p}[u_0, k_{\pi_0}, u_1]$.
- Así seguimos hasta la jugada $(n + 1)$-ésima de $\exists$, en donde su posición es $K^n_{n,p}[u_0, k_{\pi_0}, u_1, \dots, u_{n-1}] \star u_n \cdot \pi_n$, (el par (u_n, π_n) es el último movimiento de $\forall$). Como $K^n_{n,p}[u_0, k_{\pi_0}, u_1, \dots, u_{n-1}] \star u_n \cdot \pi_n \succ u_p \star \pi_0$, $\exists$ gana ya que u_p es la primera componente de la jugada p-ésima de $\forall$.

Definiremos formalmente el conjunto de los λ_c-términos para los cuales hay una estrategia ganadora, los cuales son realizadores universales de la Ley de Peirce. Para esto, primero debemos definir los estados ganadores.

Definición 9.1 (Estados ganadores) *Definimos un estado como un par $\langle p, l \rangle$, donde p es una $\exists$-posición y $l \subset \Lambda$ un conjunto finito. Intuitivamente, el estado codifica la situación de una partida en el momento en que juega $\exists$: lo único que interesa es la posición actual de $\exists$ y las primeras componentes de los $\forall$-movimientos anteriores (representadas por l).*

Dado un manejador $(u_0, \pi_0) \in \Lambda \times \Pi$, definimos el conjunto de los estados ganadores $W_{(u_0,\pi_0)}$ a partir de las siguientes dos reglas:

$$\frac{}{\langle p, l\rangle \in W_{(u_0,\pi_0)}} \quad (\text{si } p \succ u \star \pi_0 \text{ para algún } u \in l)$$

$$\frac{\langle t \star u \cdot \pi, l \cup \{u\}\rangle \in W_{(u_0,\pi_0)} \quad \text{para todo } (u,\pi) \in \Lambda \times \Pi}{\langle p, l\rangle \in W_{(u_0,\pi_0)}} \quad (\text{si } p \succ u_0 \star t \cdot \pi_0)$$

Observemos que si $\exists$ juega un término t en su turno, y l es la lista de las primeras componentes de los movimientos anteriores de $\forall$, el estado en el próximo turno de $\exists$ va a ser de la forma $\langle t \star u \cdot \pi, l \cup \{u\}\rangle$ donde $(u,\pi) \in \Lambda \times \Pi$ sería el movimiento de $\forall$. Por lo tanto, la segunda regla nos indica que si en nuestro estado actual podemos jugar t, y los posibles estados del siguiente turno son todos ganadores, entonces el estado actual es ganador.

Definición 9.2 (Estrategia ganadora) *Decimos que un λ_c-término cerrado t_0 tiene una estrategia ganadora para el juego $\mathbb{G}_0$ si $\langle t_0 \star u_0 \cdot \pi_0, \emptyset\rangle \in W_{(u_0,\pi_0)}$ para todo manejador $(u_0,\pi_0) \in \Lambda \times \Pi$.*

Como dijimos anteriormente, los términos con estrategias ganadoras realizan la Ley de Peirce.

Proposición 9.7 (Adecuación de las estrategias ganadoras) *Sea t_0 un λ_c-término con una estrategia ganadora para $\mathbb{G}_0$, A y B dos fórmulas cerradas, entonces $t_0 \Vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A$.*

Demostración. Fijado un polo, tomamos u_0 un realizador de $(A \Rightarrow B) \Rightarrow A$ y $\pi_0 \in ||A||$, y buscamos ver que $t_0 \star u_0 \cdot \pi_0 \in \perp\!\!\!\perp$. Para esto, probaremos que para todos los estados $\langle p, l\rangle \in W_{(u_0,\pi_0)}$, si $l \subseteq |A|$ entonces $p \in \perp\!\!\!\perp$. Luego, como $\langle t_0 \star u_0 \cdot \pi_0, \emptyset\rangle \in W_{(u_0,\pi_0)}$, tendremos que $t_0 \star u_0 \cdot \pi_0 \in \perp\!\!\!\perp$, concluyendo la prueba.

Dicha afirmación la demostraremos por inducción, dividiendo en casos según las dos reglas utilizadas para construir el conjunto $W_{(u_0,\pi_0)}$:

- En el primer caso, $\langle p, l\rangle \in W_{(u_0,\pi_0)}$ debido a que $p \succ u \star \pi_0$ con $u \in l$. Por lo tanto, como $u \star \pi_0 \in \perp\!\!\!\perp$ (ya que $u \in l \subseteq |A|$ y $\pi_0 \in ||A||$) concluimos por anti-evaluación que $p \in \perp\!\!\!\perp$.

- En el segundo caso, $\langle p, l\rangle \in W_{(u_0,\pi_0)}$ a partir de que $p \succ u_0 \star t \cdot \pi_0$, y que $\langle t \star u \cdot \pi, l \cup \{u\}\rangle \in W_{(u_0,\pi_0)}$ para todo $(u,\pi) \in \Lambda \times \Pi$. Como $p \succ u_0 \star t \cdot \pi_0$, $u_0 \Vdash (A \Rightarrow B) \Rightarrow A$ y $\pi_0 \in ||A||$, falta ver que $t \Vdash A \Rightarrow B$ para concluir por anti-evaluación que $p \in \perp\!\!\!\perp$.

  Sea $u \cdot \pi \in ||A \Rightarrow B|| = |A| \cdot ||B||$, como $\langle t \star u \cdot \pi, l \cup \{u\}\rangle \in W_{(u_0,\pi_0)}$ y $l \cup \{u\} \subseteq |A|$ (recordar que por hipótesis teníamos $l \subseteq |A|$), por hipótesis inductiva $t \star u \cdot \pi \in \perp\!\!\!\perp$ y, por lo tanto, $t \Vdash A \Rightarrow B$.

$\square$

9.1.3. Especificación bajo ciertas restricciones

El recíproco lo probaremos bajo ciertas restricciones sobre la instancia del λ_c-cálculo, para las cuales debemos primero introducir algunos conceptos. Aún así, es importante aclarar que existe otro juego $\mathbb{G}_1$ -que no será abordado- que especifica la Ley de Peirce en cualquier instancia sin estas restricciones.

Empezamos primero extendiendo la sustitución con el fin de permitir sustituir instrucciones por λ_c-términos, y constantes de pila por pilas:

Definición 9.3 (Sustitución de instrucciones) *Se define la sustitución de una instrucción $\mathtt{i}$ por un término cerrado u en un término t y en una pila π (utilizaremos las notaciones $t[u/\mathtt{i}]$ y $\pi[u/\mathtt{i}]$ respectivamente) recursivamente:*

$$
\begin{array}{rcll}
x[u/\mathtt{i}] & := & x & \\
\mathtt{i}[u/\mathtt{i}] & := & u & \\
\mathtt{k}[u/\mathtt{i}] & := & \mathtt{k} & \text{si } \mathtt{k} \not\equiv \mathtt{i} \\
k_\pi[u/\mathtt{i}] & := & k_{\pi[u/\mathtt{i}]} & \\
(t\,v)[u/\mathtt{i}] & := & t[u/\mathtt{i}]\,v[u/\mathtt{i}] & \\
(\lambda x.\,t)[u/\mathtt{i}] & := & \lambda x.\,t[u/\mathtt{i}] & \\
& & & \\
\alpha[u/\mathtt{i}] & := & \alpha & \\
(t\cdot\pi)[u/\mathtt{i}] & := & t[u/\mathtt{i}]\cdot\pi[u/\mathtt{i}] &
\end{array}
$$

Definición 9.4 (Sustitución de constantes de pila) *Se define la sustitución de una constante de pila α por una pila π' en un término t y en una pila π (utilizaremos las notaciones $t[\pi'/\alpha]$ y $\pi[\pi'/\alpha]$ respectivamente) recursivamente:*

$$
\begin{array}{rcll}
x[\pi'/\alpha] & := & x & \\
\mathtt{k}[\pi'/\alpha] & := & \mathtt{k} & \\
k_\pi[\pi'/\alpha] & := & k_{\pi[\pi'/\alpha]} & \\
(t\,u)[\pi'/\alpha] & := & t[\pi'/\alpha]\,u[\pi'/\alpha] & \\
(\lambda x.\,t)[\pi'/\alpha] & := & \lambda x.\,t[\pi'/\alpha] & \\
& & & \\
\alpha[\pi'/\alpha] & := & \pi' & \\
\beta[\pi'/\alpha] & := & \beta & \text{si } \beta \neq \alpha \\
(t\cdot\pi)[\pi'/\alpha] & := & t[\pi'/\alpha]\cdot\pi[\pi'/\alpha] &
\end{array}
$$

Notar que estamos sustituyendo todas las ocurrencias de las instrucciones o constantes de pila en un término, incluso buscando en la pila de una constante de continuación. De hecho, en lo que queda, cuando hablemos de que una instrucción/constante de pila no ocurre en un término, haremos referencia a que tampoco ocurre en las pilas asociadas a las constantes de continuación.

En base a dichas extensiones de la sustitución, podemos introducir las siguientes definiciones que nos permitirán establecer las hipótesis bajo las cuales especificaremos la Ley de Peirce:

Definición 9.5 *Sea ($\mathcal{K}$, Π_0, $\succ$) una instancia del λ_c-cálculo y $\boldsymbol{k} \in \mathcal{K}$ una instrucción, decimos que $\boldsymbol{k}$ es:*

- ***inerte**: si, para toda $\pi \in \Pi$, no existe un proceso p tal que $\boldsymbol{k} \star \pi \succ p$.*
- ***sustitutiva**: si, para todo término cerrado t y todo par de procesos p, p' tales que $p \succ p'$, se cumple que $p[t/\boldsymbol{k}] \succ p'[t/\boldsymbol{k}]$.*
- ***no generativa**: si, para todo par de procesos p, p' tales que $p \succ p'$, $\boldsymbol{k}$ no puede ocurrir en p' si no lo hace en p.*

*Si la instrucción $\boldsymbol{k}$ es inerte, sustitutiva y no generativa, decimos que es una **constante de interacción**.*

De la misma manera, podemos definir conceptos similares para una constante de pila.

Definición 9.6 *Sea ($\mathcal{K}$, Π_0, $\succ$) una instancia del λ_c-cálculo y $\alpha \in \Pi_0$ una constante de pila, decimos que α es:*

1. ***sustitutiva**: si, para toda pila π y todo par de procesos p, p' tales que $p \succ p'$, se cumple que $p[\pi/\alpha] \succ p'[\pi/\alpha]$.*
2. ***no generativa**: si, para todo par de procesos p, p' tales que $p \succ p'$, α no puede ocurrir en p' si no lo hace en p.*

La utilización de las constantes de interacción y de las constantes de pila es fundamental para probar que los realizadores universales de la Ley de Peirce tienen estrategias ganadoras para $\mathbb{G}_0$. Básicamente, vamos a probar que si los movimientos de $\forall$ fueran de la forma (c_i, α_i) (donde c_i es una constante de interacción y α_i es una constante de pila), podemos encontrar una estrategia ganadora. Y luego, a través de la sustitución, extenderemos esto permitiendo movimientos $(u_i, \pi_i) \in \Lambda \times \Pi$ cualesquiera.

Lema 9.1 *Sea t_0 un realizador universal de la Ley de Peirce, $(c_i)_{i \in \mathbb{N}}$ una sucesión de constantes de interacción, distintas dos a dos y que no ocurren en t_0, y $(\alpha_i)_{i \in \mathbb{N}}$ una sucesión de constantes de pila, entonces existe un par de naturales (n, p) con $1 \leq p \leq n$ y una secuencia finita $t_1, \ldots, t_n$ de n términos cerrados tales que:*

$$\begin{array}{ccccccc} t_0 & \star & c_0 \cdot \alpha_0 & \succ & c_0 & \star & t_1 \cdot \alpha_0 \\ t_i & \star & c_i \cdot \alpha_i & \succ & c_0 & \star & t_{i+1} \cdot \alpha_0 \\ t_n & \star & c_n \cdot \alpha_n & \succ & c_p & \star & \alpha_0 \end{array}$$

Demostración. Primero, tomamos la sucesión de conjuntos de procesos $(\mathcal{Q}_i)_{i\in\mathbb{N}}$ definida como: $\mathcal{Q}_0 := Thd(t_0 \star c_0 \cdot \alpha_0)$ y $\mathcal{Q}_i := \bigcup_{t\in\Lambda\,:\,c_0\star t\cdot\alpha_0\in\mathcal{Q}_{i-1}} Thd(t \star c_i \cdot \alpha_i)$.

La idea será encontrar n y p, con $1 \leq p \leq n$, tales que $c_p \star \alpha_0 \in \mathcal{Q}_n$. Si tenemos eso, por definición debe existir $t_n \in \Lambda$, con $c_0 \star t_n \cdot \alpha_0 \in \mathcal{Q}_{n-1}$, tal que $t_n \star c_n \cdot \alpha_n \succ c_p \star \alpha_0$. A su vez, si $c_0 \star t_n \cdot \alpha_0 \in \mathcal{Q}_{n-1}$, también debe existir $t_{n-1} \in \Lambda$, con $c_0 \star t_{n-1} \cdot \alpha_0 \in \mathcal{Q}_{n-2}$, tal que $t_{n-1} \star c_{n-1} \cdot \alpha_{n-1} \succ c_0 \star t_n \cdot \alpha_0$, y así hasta obtener los términos $t_1, \ldots, t_n$.

Llamemosle $\mathcal{Q}_\infty = \cup_{i\in\mathbb{N}}\mathcal{Q}_i$, y consideremos el polo definido como $\perp\!\!\!\perp := (\mathcal{Q}_\infty)^c$. Tomemos ahora las fórmulas $A \equiv \dot{\alpha}_0$ y $B \equiv \perp$, por lo que $t_0 \Vdash_{\perp\!\!\!\perp} (\neg\dot{\alpha}_0 \Rightarrow \dot{\alpha}_0) \Rightarrow \dot{\alpha}_0$. Por la definición del polo, $t_0 \star c_0 \cdot \alpha_0 \notin \perp\!\!\!\perp$, pero como $\alpha_0 \in ||\dot{\alpha}_0||_{\perp\!\!\!\perp}$, entonces $c_0 \nVdash_{\perp\!\!\!\perp} \neg\dot{\alpha}_0 \Rightarrow \dot{\alpha}_0$.

Como $c_0 \nVdash_{\perp\!\!\!\perp} \neg\dot{\alpha}_0 \Rightarrow \dot{\alpha}_0$, entonces existe $t \Vdash_{\perp\!\!\!\perp} \neg\dot{\alpha}_0$ tal que $c_0 \star t \cdot \alpha_0 \notin \perp\!\!\!\perp$ o, lo que es equivalente, $c_0 \star t \cdot \alpha_0 \in \mathcal{Q}_{p-1}$ para algún $p \geq 1$. Por la definición de $\mathcal{Q}_p$, esto significa que $t \star c_p \cdot \alpha_p \in \mathcal{Q}_p$. Pero, como $t \Vdash_{\perp\!\!\!\perp} \neg\dot{\alpha}_0$, tenemos que $c_p \nVdash_{\perp\!\!\!\perp} \dot{\alpha}_0$, por lo que $c_p \star \alpha_0 \notin \perp\!\!\!\perp$ y luego $c_p \star \alpha_0 \in \mathcal{Q}_n$ para algún $n \geq 0$.

Se puede observar que, como las instrucciones c_i son distintas dos a dos, no generativas, y no ocurren en t_0, si el proceso $c_p \star \alpha_0 \in \mathcal{Q}_n$ entonces necesariamente $n \geq p$. □

Teorema 9.1 (Especificación débil de la Ley de Peirce) *Si la instancia del λ_c-cálculo contiene infinitas constantes de interacción, a la vez que infinitas constantes de pila no generativas y sustitutivas, entonces los realizadores de la Ley de Peirce son exactamente los términos con estrategias ganadoras para el juego $\mathbb{G}_0$.*

Demostración. Ya probamos (en la proposición 9.6) que los términos t_0 que tienen estrategias ganadoras, realizan universalmente la Ley de Peirce. Recíprocamente, hay que ver que si t_0 es un realizador universal de dicha Ley, tiene una estrategia ganadora para $\mathbb{G}_0$ (formalmente, que $\langle t_0 \star u_0 \cdot \pi_0, \emptyset\rangle \in W_{(u_0,\pi_0)}$ para todo manejador $(u_0, \pi_0) \in \Lambda \times \Pi$).

Tomamos una sucesión $(c_i)_{i\in\mathbb{N}}$ de constantes de interacción, distintas dos a dos y que no ocurren en t_0, al igual que una sucesión $(\alpha_i)_{i\in\mathbb{N}}$ de constantes de pila, distintas dos a dos, sustitutivas y no generativas, y que tampoco ocurren en t_0 (esto es siempre posible, porque t_0 sólo contiene una cantidad finita de ellas). Por el lema anterior, existe un par de naturales (n,p), con $1 \leq p \leq n$, y una secuencia de términos cerrados $t_1, \ldots, t_n$ tales que:

$$\begin{array}{lllllll} t_0 & \star & c_0 \cdot \alpha_0 & \succ & c_0 & \star & t_1 \cdot \alpha_0 \\ t_i & \star & c_i \cdot \alpha_i & \succ & c_0 & \star & t_{i+1} \cdot \alpha_0 \\ t_n & \star & c_n \cdot \alpha_n & \succ & c_p & \star & \alpha_0 \end{array}$$

Para probar el resultado bastará ver que, para todo $0 \leq i \leq n$, se cumple la siguiente afirmación:

$$\begin{array}{c} \text{Para todo } u_0, \ldots, u_i \in \Lambda,\ \pi_0, \ldots, \pi_i \in \Pi\text{:} \\ \langle t_i[u_j/c_j]_{j=0}^{i-1}[\pi_j/\alpha_j]_{j=0}^{i-1} \star u_i \cdot \pi_i, \{u_1, \ldots, u_i\}\rangle \in W_{(u_0,\pi_0)} \end{array}$$

Donde $t_i[u_j/c_j]_{j=0}^{i-1}[\pi_j/\alpha_j]_{j=0}^{i-1} := t_i[u_0/c_0]\cdots[u_{i-1}/c_{i-1}][\pi_0/\alpha_0]\cdots[\pi_{i-1}/\alpha_{i-1}]$. Para probar la afirmación para todo $0 \leq i \leq n$, lo hacemos en orden inverso:

- Si $i = n$, teníamos que:

  $$t_n \star c_n \cdot \alpha_n \succ c_p \star \alpha_0$$

  Por sustitutividad, para todo $u_0, \ldots, u_n \in \Lambda$, y $\pi_0, \ldots, \pi_n \in \Pi$, se cumple que:

  $$t_n[u_j/c_j]_{j=0}^{n}[\pi_j/\alpha_j]_{j=0}^{n} \star u_n \cdot \pi_n \succ u_p \star \pi_0.$$

  Por las hipótesis (constantes no generativas, distintas dos a dos, y que no ocurren en t_0), el término t_n puede contener constantes c_j/α_j para cualquier $j < n$, pero no contiene ni a c_n ni a α_n. Como consecuencia, $t_n[u_j/c_j]_{j=0}^{n}[\pi_j/\alpha_j]_{j=0}^{n} \equiv t_n[u_j/c_j]_{j=0}^{n-1}[\pi_j/\alpha_j]_{j=0}^{n-1}$.

  Combinando ambos resultados, se obtiene por la primera regla de la construcción de $W_{(u_0,\pi_0)}$:

  $$\langle t_n[u_j/c_j]_{j=0}^{n-1}[\pi_j/\alpha_j]_{j=0}^{n-1} \star u_n \cdot \pi_n, \{u_1, \ldots, u_n\}\rangle \in W_{(u_0,\pi_0)}$$

- Si $0 \leq i < n$, sean $u_0, \ldots, u_i \in \Lambda$, $\pi_0, \ldots, \pi_i \in \Pi$ cualesquiera, sabemos por sustitutividad que:

  $$t_i[u_j/c_j]_{j=0}^{i}[\pi_j/\alpha_j]_{j=0}^{i} \star u_i \cdot \pi_i \succ u_0 \star t_{i+1}[u_j/c_j]_{j=0}^{i}[\pi_j/\alpha_j]_{j=0}^{i} \cdot \pi_0$$

  Como $t_i[u_j/c_j]_{j=0}^{i}[\pi_j/\alpha_j]_{j=0}^{i} \equiv t_i[u_j/c_j]_{j=0}^{i-1}[\pi_j/\alpha_j]_{j=0}^{i-1}$ (con el mismo argumento que vimos para $i = n$), basta ver que para todo $(u_{i+1}, \pi_{i+1}) \in \Lambda \times \Pi$:

$$\langle t_{i+1}[u_j/c_j]_{j=0}^{i}[\pi_j/\alpha_j]_{j=0}^{i} \star u_{i+1} \cdot \pi_{i+1}, \{u_1, \ldots, u_i\} \cup \{u_{i+1}\}\rangle \in W_{(u_0,\pi_0)}$$

(lo cual es trivial asumiendo que la afirmación se cumple para $i+1$), para concluir que, para todo $u_0, \ldots, u_i \in \Lambda$, $\pi_0, \ldots, \pi_i \in \Pi$:

$$\langle t_i[u_j/c_j]_{j=0}^{i-1}[\pi_j/\alpha_j]_{j=0}^{i-1} \star u_i \cdot \pi_i, \{u_1, \ldots, u_i\}\rangle \in W_{(u_0,\pi_0)}$$

Tomando $i = 0$, nos queda: para todo $u_0 \in \Lambda$, $\pi_0 \in \Pi$, $\langle t_0 \star u_0 \cdot \pi_0, \emptyset\rangle \in W_{(u_0,\pi_0)}$.
$\square$

Referencias

[1] Lionel Rieg. *On Forcing and Classical Realizability.* 2014.

[2] L.E.J. Brouwer. “Intuitionism and formalism”. En: *Bulletin of the American Mathematical Society* 20.2 (1913), págs. 81-96.

[3] Mark van Atten y Göran Sundholm. *L.E.J. Brouwer's ‘Unreliability of the logical principles’. A new translation, with an introduction.* 2015.

[4] Rosalie Iemhoff. “Intuitionism in the Philosophy of Mathematics”. En: *The Stanford Encyclopedia of Philosophy.* Ed. por Edward N. Zalta. Fall 2020. Metaphysics Research Lab, Stanford University, 2020.

[5] A.S. Troelstra. *History of Constructivism in the Twentieth Century.* University of Amsterdam, 1991.

[6] Jaap van Oosten. “Realizability: A Historical Essay”. En: *Mathematical Structures in Computer Science* 12 (2001).

[7] Jean-Louis Krivine. *Lambda-Calculus, Types and Models.* USA: Ellis Horwood, 1993. ISBN: 0130624071.

[8] Dirk van Dalen. *Logic and Structure.* Springer, 1999.

[9] Jean-Louis Krivine. “Dependent choice, ‘quote’ and the clock”. En: *Theoretical Computer Science* 308 (2003), págs. 259-276.

[10] Mauricio Guillermo y Alexandre Miquel. “Specifying Peirce's law in classical realizability”. En: *Mathematical Structures in Computer Science* (2014), págs. 1-35.