



HAL
open science

Principios fundamentales de la Teoría Constructiva de Tipos

Shahid Rahman, Nicolas Clerbout

► **To cite this version:**

Shahid Rahman, Nicolas Clerbout. Principios fundamentales de la Teoría Constructiva de Tipos. 2015. halshs-01229579v1

HAL Id: halshs-01229579

<https://shs.hal.science/halshs-01229579v1>

Preprint submitted on 20 Nov 2015 (v1), last revised 23 Nov 2015 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Principios fundamentales de la Teoría Constructiva de Tipos¹

Shahid Rahman y Nicolas Clerbout

Dentro de la Teoría Constructiva de Tipos (en adelante TCT) las constantes lógicas son interpretadas a través de la correspondencia Curry-Howard entre proposiciones y conjuntos. Una proposición es interpretada como un conjunto cuyos elementos representan las pruebas de la proposición. También es posible ver un conjunto como la descripción de un problema, en un sentido similar a la explicación de Kolmogorov sobre el cálculo proposicional intuicionista. En particular un conjunto puede ser visto como la especificación de la programación de un problema: los elementos del conjunto son entonces los programas que satisfacen la especificación (Martin-Löf 1984, p. 7). Más aún en TCT, los conjuntos son también entendidos como tipos en el cual las proposiciones pueden ser vistas como tipos de datos (*data-types*) o tipos de pruebas (*proof-types*)². Partiremos con la introducción de los principios fundamentales de la TCT. Y luego revisaremos las reglas de la lógica intuicionista de predicados en TCT.

La idea filosófica general está ligada al programa que se conoce bajo el nombre de *lenguajes totalmente interpretados*³, en los cuales se toma especial cuidado de

evitar de mantener el contenido y la forma apartes. Por el contrario, lo que haremos [en el marco de un lenguaje totalmente interpretado] es exhibir ciertas formas de juicio e inferencia que son usadas en las pruebas matemáticas y proveer al mismo tiempo una explicación de significado de tales formas. De esta manera hacemos explícito lo que usualmente es tomado implícitamente por garantizado (Martin-Löf, 1984, p.2).

La tarea de *explicitación* envuelve expresar en el interior del lenguaje objeto aquellas características que determinan el significado y que usualmente son formuladas en el metalenguaje.

De acuerdo al punto de vista de la TCT, las premisas y la conclusión en una inferencia lógica no son proposiciones, sino juicios. La justificación de una tal regla requiere una explicación de significado de los juicios que constituyen premisas y conclusión:

Una regla de inferencia está justificada cuando se explica la conclusión bajo el supuesto de que las premisas son conocidas. Por lo tanto, antes de que una regla de inferencia pueda ser justificada, debe explicarse qué es lo que hay que saber para tener el derecho de hacer un juicio sobre cualquiera de las diversas formas en que las premisas y la conclusión pueden tener (Martin-Löf 1984, p. 2).

Otros dos principios básicos de la TCT son los siguientes:

1. Ninguna entidad sin tipo
2. Ningún tipo sin identidad

En consecuencia, podemos tomar la afirmación de que un individuo es un elemento del conjunto A como la afirmación de que dicho individuo instancia o ejemplifica el tipo A . Un conjunto se define en TCT especificando sus elementos canónicos, y no-canónicos. Los no-canónicos son aquellos

1 Traducido del inglés por Mario Tapia Ramírez del **capítulo** correspondiente del artículo de Rahman/Clerbout *Immanent Reasoning and the Dialogical Genealogy of Equality*, en preparación.

2 Cf. Nordström et al. (1990) and Granström (2011).

3 Para una discusión más profunda de este asunto, ver Sundholm (1997, 2001).

elementos de los que se puede mostrar, usando algún método prescrito de transformación, que son iguales (en este conjunto) a uno canónico. La igualdad en el conjunto es precisamente lo que prescribe el segundo principio básico y que, en otras palabras; consiste en la introducción de una relación de equivalencia en un conjunto. Así, si A es un tipo y tenemos un objeto b que satisface las condiciones correspondientes entonces b es un objeto de tipo A , que se escribe formalmente $b : A$.⁴ Es esencial distinguir entre el *elemento de prueba* b (*proof-object*)⁵, el *tipo* A y el *juicio* $b : A$. La última expresión expresa, que b es un elemento de prueba para la proposición A (si A es del tipo proposición). En lógica estándar, que hay una prueba para una proposición dada se expresa en el nivel de metalenguaje. El hecho de que haya algo (un elemento de prueba) b que fundamenta la proposición de que *Primus le da 100 monedas a Secundus* (lo que supondría la afirmación correspondiente) se da en el análisis habitual a nivel metalenguaje. En TCT, el fundamento de una afirmación se formula en el nivel de lenguaje objeto por medio de la afirmación de que hay un elemento de prueba de la proposición correspondiente. En un tal marco

$b : A$	<i>verdadero</i> A
Puede ser leído como	
b es un elemento del conjunto A	A tiene un elemento
b es una prueba de la proposición A	A es verdadero
b satisface las expectativas de A	A es satisfecha
b es una solución al problema A	A tiene solución

Es importante notar que '*Conjunto*' mismo no es instancia del tipo *conjunto*, (i.e. **no** es el caso que *set* : *set*) puesto que no tenemos un método general para generar todas las posibles formas de construir un conjunto. Sin embargo, dado el tipo '*conjunto*' podemos construir los objetos que lo instancian mediante las reglas de introducción de conjuntos descritas anteriormente.

Hipotéticos: Los juicios de los que hablamos hasta ahora no dependen de ninguna suposición. Son juicios categóricos. Sin embargo, el lenguaje de la TCT permite también de expresar juicios hipotéticos por medio de la siguiente formulación

$$B \text{ type}^6 (x : A)$$

en dónde A es un tipo que no depende de ninguna suposición y B es un tipo cuando se da el caso que $x : A$ (decimos entonces que $x : A$ es la *hipótesis* de B). En el caso que B sea del tipo conjunto (del tipo *set*), b es un elemento de B , bajo la hipótesis de que x sea un elemento de A :

⁴ Martin-Löf usa el signo " \in " con el fin de indicar que algo, por ejemplo a , es de tipo B ; incluso sugiere que se puede entender como la cópula "es". Nordström, Petersson y Smith (1990) también hacen uso de esta notación, mientras que otros autores, como Ranta (1994), utilizan el doble punto ":". Granström (2011) distingue el doble punto del épsilon, donde el primero se aplica a los elementos no canónicos y el segundo a los canónicos. Nosotros vamos a utilizar el doble punto.

⁵ El término en inglés es *proof-object*. Nos decidimos por la traducción *elemento de prueba* que no es ideal pues puede sugerir que tal prueba no provee una justificación completa. Sin embargo, en el cuadro de la TCT cada *proof-object* por sí mismo provee una justificación suficiente de la proposición correspondiente. En suma, cuando usamos la expresión *elemento de prueba* debe pensarse como un objeto que es suficiente para la justificación requerida, aunque puede haber otros tales objetos (que también son suficientes).

⁶ A fin de facilitar la comparación con la literatura original en TCT y el lenguaje de programación que se ha ya difundido e impuesto optamos por no traducir los nombres de los tipos tales como, *type*, *set*, etc.

$$b : B (x : A)$$

La introducción explícita de juicios hipotéticos en el lenguaje requiere la introducción de reglas de sustitución apropiadas, pero trataremos esto más adelante en nuestro texto, cuando presentemos de forma sistemática el punto de vista de la TCT sobre la igualdad.

Granström [2011, p.112] señala que el juicio $b : B (x : A)$ puede ser generalizado en tres direcciones:

1. Extender la hipótesis a un número arbitrario;
2. El rango de un variable que ocurre en un juicio hipotético puede depender de otras variables ya introducidas;
3. El conjunto B puede depender de todas las variables ya introducidas.

Una lista de hipótesis lleva el nombre de *contexto*. Los contextos permiten abreviar la notación:

Γ es un contexto (i.e., es una lista de hipótesis)

$$b : B (\Gamma)$$

$$\Gamma : \text{contexto}$$

Aquello que es *dado en un contexto* es todo lo que puede inferirse de las hipótesis que constituye un tal contexto. Eso es lo que llamamos *conocimiento contextualmente dependiente*. De hecho, se distingue usualmente entre lo que es *actualmente dado* en el contexto (*conocimiento actual*) – las variables y los juicios en los que ocurren tales variables, y sobre todo lo que puede *potencialmente* inferirse por medio de las reglas de la teoría de tipos desde lo que es actualmente dado.

Si escribimos explícitamente la lista que constituye el contexto, la forma general de un juicio hipotético es la siguiente:

$$x : A (x_1 : A_1, x_2 : A_2, \dots, x_n : A_n)$$

presuponiendo que ya sabemos que A_1 es un tipo (que no depende de ningún otro tipo), que A_2 es un tipo en el contexto $x_1 : A_1$, y que A_n es un tipo en el contexto $x_1 : A_1, x_2 : A_2, \dots, x_{n-1} : A_{n-1}$. Esto es:

A_1 type

A_2 type ($x_1 : A_1$)

...

A_n type ($x_1 : A_1, x_2 : A_2, \dots, x_{n-1} : A_{n-1}$)

A type ($x_1 : A_1, x_2 : A_2, \dots, x_n : A_n$)

$$x : A (x_1 : A_1, x_2 : A_2, \dots, x_n : A_n)$$

En este punto de la presentación ya debiera quedarle claro al lector que los elementos de prueba de un juicio hipotético en el que B depende de la hipótesis A son funciones de A a B :

$$f(x) : B (x : A)$$

Es crucial tener claro que la noción de función es intensional más que extensional. De hecho, se desprende del significado de un juicio hipotético que introduce una función que si sustituimos x por un elemento arbitrario a de A en $f(x)$, obtenemos un elemento $f(a)$ de B – la expresión $f(x)$ es así

interpretada como una función de A a B ⁷. Más aún, la igualdad de dos funciones es definida de tal manera que que la sustitución de elementos iguales de A resulta en elementos intensionalmente iguales de B .

De hecho tales expresiones pueden leerse de diversas maneras, por ejemplo:

$f(x) : B$ para un x arbitrario tal que $x : A$
 $f(x) : B$ bajo la hipótesis $x : A$
 $f(x) : B$ provisto que $x : A$
 $f(x) : B$ dado que $x : A$
 $f(x) : B$ si $x : A$
 $f(x) : B$ en el contexto $x : A$

En adición a los dominios de individuos, un lenguaje científico interpretado requiere proposiciones. Éstas son introducidas en TCT mediante reglas que establecen qué cuenta como prueba de una proposición. De acuerdo a esto, una proposición es verdadera si y solamente si hay una prueba para ella. Escribimos

$$A : prop$$

para expresar el juicio que A es una proposición. Las funciones proposicionales son introducidas mediante juicios hipotéticos. El juicio hipotético requerido para introducir funciones proposicionales tiene la forma:

$$B(x) : prop (x : A)$$

que se lee, $B(x)$ es del tipo proposición, provisto que la función $B(x)$ se aplica a elementos de A , tal que A es del tipo set. Esto motiva las siguiente reglas:

$$\begin{array}{c}
 a : A \quad B(x) : prop (x : A) \\
 \hline
 B(a) : prop \\
 \\
 a=b : A \quad B(x) : prop (x : A) \\
 \hline
 B(a) = B(b) : prop
 \end{array}$$

La formulación de la noción de función proposicional como elemento de prueba de un juicio hipotético permite introducir subconjuntos por medio de una regla intensional de separación:

$$\begin{array}{c}
 A : set \quad B(x) : prop (x : A) \\
 \hline
 \{x : A / B(x)\} : set \\
 \\
 b : \{x : A / B(x)\} \\
 \hline
 b : A
 \end{array}
 \qquad
 \begin{array}{c}
 b : A \quad B(b) true \\
 \hline
 b : \{x : A / B(x)\} \\
 \\
 b : \{x : A / B(x)\} \\
 \hline
 B(b) true
 \end{array}$$

Esta explicación de subconjuntos también justifica las siguiente reglas

Dado que este método está basado en conjuntos pre-existentes que han sido construidos por medio de la descripción de sus elementos canónicos, las paradojas estándar de teoría de conjuntos no

⁷ Cf. Ranta (1994, p. 21), Nordström/Petersson/Smith (1990, chapter 3.3), Primiero (2008, pp. 47-55) y Granström (2011, pp. 77-102)

aparecen (también desaparecen así algunas paradojas que aparecen en algunas formulaciones tempranas del método de Lorenzen para la construcción de conjuntos).

1.4 Las bases de la lógica intuicionista de predicados en el cuadro de la TCT

En esta sección introduciremos rápidamente las reglas básicas de la lógica intuicionista en el marco de la TCT. Nuestra presentación está inspirada en la descripción sucinta que da Ranta (1991, sección 3) del sistema de reglas introducido por Martin-Löf (1984, pp.24-25) y de la presentación de Nordström et al. (1990).

1.4.1 Cuatro tipos de reglas

Dado que en este marco teórico las proposiciones son conjuntos, los operadores lógicos están definidos como operadores conjuntistas. El significado de tales operadores se establece por medio de cuatro tipos de reglas diferentes, a saber: reglas de formación, de introducción, de eliminación y reglas de igualdad:

- *Reglas de formación.* La inclusión explícita de reglas de formación en un sistema inferencial es una de las características más distintivas de la TCT. Las reglas de formación establecen simultáneamente la sintaxis y los tipos básicos a los que corresponden las constantes lógicas y no lógicas del lenguaje considerado. Más precisamente, las reglas de formación especifican bajo qué condiciones podemos inferir que algo es un conjunto (tipo), y bajo qué condiciones podemos decir que dos conjuntos son iguales. Por ende, dado que la buena formación incluye no solo los modos de composición sintáctica sino también la identificación de los tipos básicos correspondientes, podemos decir que las reglas de formación despliegan al mismo tiempo las reglas de buena formación sintáctica y semántica específicas a un lenguaje determinado.

De hecho, toda demostración TCT⁸ comienza verificando que las expresiones del juicio a demostrar resultan de la aplicación de las reglas de formación correspondientes. Es éste el modo de implementar en el cuadro de la TCT la idea de un lenguaje totalmente interpretado: cuando leemos una demostración TCT de abajo hacia arriba ella despliega los elementos sintácticos y semánticos del juicio demostrado.

- *Reglas de introducción.* Las reglas de introducción que definen los tipos del sistema, prescriben el modo de formar elementos canónicos y el modo de determinar si dos elementos son iguales.
- *Reglas de eliminación.* Ellas establecen el modo de definir funciones (llamadas *selectores*) en el conjunto definido por las reglas de introducción.
- *Reglas de igualdad.* Como ya mencionamos en la sección anterior, las reglas de igualdad relacionan la “armonía” entre las reglas de introducción y eliminación. Más precisamente las reglas de igualdad especifican el modo en el que operan los selectores definidos por las reglas de eliminación y cómo ejecutar su computación dados los elementos canónicos generados por las reglas de introducción.

1.4.2 La lógica intuicionista en TCT

El operador Π

(el producto cartesiano de una familia de conjuntos):

$(x : A)$

$(x : A)$

⁸ Recordamos al lector que usamos la expresión *demostración TCT* como una abreviación de la expresión *demostración en el cuadro de la teoría constructiva de tipos*.

$$\begin{array}{c}
\begin{array}{c}
\dots \\
A : set \quad B(x) : set \\
\dots
\end{array} \\
\hline
(\Pi x : A)B(x) : set
\end{array}
\quad \text{PIF} \quad
\begin{array}{c}
\dots \\
b(x) : B(x) \\
\dots
\end{array}
\quad \text{PII} \\
\hline
(\lambda x)b(x) : (\Pi x : A)B(x)$$

$$\begin{array}{c}
c : (\Pi x : A)B(x) \quad a : A \\
\hline
Ap(c, a) : B(a) \\
(x : A)
\end{array}
\quad \text{PIE}$$

$$\begin{array}{c}
a : A \quad b(x) : B(x) \\
\hline
Ap((\lambda x)b(x), a) = b(a) : B(a)
\end{array}
\quad \text{PIEq}^1$$

$$\begin{array}{c}
c : (\Pi x : A)B(x) \\
\hline
c = (\lambda x)Ap(c, x) : (\Pi x : A)B(x)
\end{array}
\quad \text{PIEq}^2$$

La regla de introducción asume, como usualmente, que x no ocurra libre en ninguna otra suposición que aquellas de la forma $x : A$. La función diádica $Ap(x, y)$ (el selector *aplicación*) está definida por la forma en que es introducida (por medio de la regla de eliminación) y por la especificación de su modo de computación (por medio de las reglas de igualdad). Puede leerse como “aplicación de x a y ” y es de hecho un método de obtener un objeto canónico $B(a)$, provisto que $a : A$ (véase Martin-Löf, 1984, pp.28-29).

En otras palabras, la regla de computación para $Ap((\lambda x)b(x), a)$ establece que si $b(x)$ es un elemento de prueba para $B(x)$ bajo el supuesto de que $x : A$, dado que a es una prueba cualquiera de A , entonces su ejecución produce un objeto prueba $b(a) : B(a)$, tal que a reemplaza a x en todo $b(a)$. Dicho brevemente:

$$Ap((\lambda x)b(x), a) \quad \rightarrow \quad b(a/x)$$

El operador Π permite definir el cuantificador universal y la implicación material de la siguiente manera:

- $(\forall x : A)B(x) = (\Pi x : A)B(x) : prop$, provisto que $A : set$ y que $B(x) : prop (x : A)$.
- $A \rightarrow B = (\Pi x : A)B : prop$, provisto que $A : prop$ y que $B : prop$.

El operador Σ

(la unión disyunta de una familia de conjuntos)

$$\begin{array}{c}
(x : A) \\
\dots \\
A : set \quad B(x) : set \\
\dots
\end{array}
\quad \text{PIF} \quad
\begin{array}{c}
a : A \quad b : B(a) \\
\dots
\end{array}
\quad \text{PII} \\
\hline
(\Sigma x : A)B(x) : set$$

$$\begin{array}{c}
(x : A, y : B(x)) \\
\dots \\
c : (\Sigma x : A)B(x) d(x, y) : C((x, y)) \\
\dots
\end{array}$$

$$\begin{array}{c}
\Sigma E \\
\mathbf{E}(c, (x, y)d(x, y)) : C(c) \\
(x : A, y : B(x)) \\
\dots \\
a : A \quad d(x, y) : C((x, y)) \\
\hline
\mathbf{E}(a, b, (x, y)d(x, y)) = d(a, b) : C((a, b)) \quad \Sigma Eq
\end{array}$$

La expresión $\mathbf{E}(c, (x, y)d(x, y))$ que ocurre en la conclusión de la regla de eliminación se lee informalmente como la siguiente instrucción compuesta:

Ejecute c .

El resultado de tal ejecución es un elemento canónico que forma el par (a, b) tal que $a : A$ y $b : B$.

Reemplace ahora x e y en la premisa derecha, con a y b respectivamente.

Obtenga de este modo $d(a, b)C((a, b))$.

La ejecución de $d(a, b)$ dará como resultado un elemento canónicos e de $C((a, b))$ – no es difícil deducir de esto que, por lo tanto, e es un elemento canónico de $C(c)$ (véase Martin-Löf, 1984, p. 40).

El operador Σ permite definir el cuantificador existencial y la conjunción de la siguiente manera:

- $(\exists x : A)B(x) = (\Sigma x : A)B(x) : prop$, provisto que $A : set$ y que $B(x) : prop (x : A)$.
- $A \wedge B = (\Sigma x : A)B : prop$, provisto que $A : prop$ y que $B : prop$.

En el caso de la conjunción obtenemos las reglas de eliminación estándar de la regla de eliminación de Σ ,

1. Si decidimos que C sea o bien A o bien B , y
2. Si definimos las reglas de proyección $p(c)$ y $q(c)$, mencionadas en la sección anterior de la siguiente manera: $p(c) \equiv \mathbf{E}(c, (x, y)x)$ y $q(c) \equiv \mathbf{E}(c, (x, y)y)$.

Es decir, si llevamos a cabo los pasos 1 y 2, de ΣE obtenemos

$$\begin{array}{c}
c : A \wedge B \\
\hline
p(c) : A \quad \wedge E^1
\end{array}
\qquad
\begin{array}{c}
c : A \wedge B \\
\hline
q(c) : A \quad \wedge E^2
\end{array}$$

Recurriendo a las reglas de igualdad obtenemos las siguientes reglas de cómputo para la ejecución de $p(c)$ y $q(c)$, donde el elemento de prueba c está constituido por el par (a, b) tal que $a : A, b : B$.

$$p(a, b) \rightarrow a$$

$$q(a, b) \rightarrow b$$

El operador +

(la unión disyunta o el co-productor de dos conjuntos)

$$\begin{array}{c}
A : set \qquad B : set \qquad a : A \\
\hline
A+B : set \qquad i(a) : A + B \quad +I^1
\end{array}$$

$a : A$

$$\begin{array}{c}
\text{-----} + I^2 \\
j(b) : A + B \\
\\
\begin{array}{ccc}
(x : A) & & (y : B) \\
\cdots & & \cdots \\
c : A + B & d(x) : C(i(x)) & e(y) : C(j(y))
\end{array} \\
\text{-----} + E \\
\mathbf{D}(c(x)d(x), (y)e(y)) : C(c) \\
\\
\begin{array}{ccc}
(x : A) & & (y : B) \\
\cdots & & \cdots \\
a : A & d(x) : C(i(x)) & e(y) : C(j(y))
\end{array} \\
\text{-----} + Eq^2 \\
\mathbf{D}(i(b), (x)d(x), (y)e(y)) = d(a) : C(i(a)) \\
\\
\begin{array}{ccc}
(x : A) & & (y : B) \\
\cdots & & \cdots \\
a : A & d(x) : C(i(x)) & e(y) : C(j(y))
\end{array} \\
\text{-----} + Eq^2 \\
\mathbf{D}(j(b), (x)d(x), (y)e(y)) = e(b) : C(j(b))
\end{array}$$

Las expresiones i y j son dos nuevas constantes primitivas (inyecciones) que proveen la información de que un elemento de $A+B$ pertenece a A o a B , y cuál es el caso.

La expresión $\mathbf{D}(c, (x)d(x), (y)e(y))$ que ocurre en la conclusión de la regla de eliminación, se lee informalmente como la siguiente instrucción compuesta:

Ejecute c ;
Si de la ejecución resulta el elemento canónico $i(a)$ entonces sustituya a por x en $d(x)$.
Si de la ejecución resulta el elemento canónico $j(b)$ entonces sustituya b por y en $e(y)$.

Si ponemos en juego las reglas de igualdad, podemos también expresar esto de la siguiente manera: la ejecución de $\mathbf{D}(c, (x)d(x), (y)e(y))$ produce el elemento de prueba para $C(c)$ construido del elemento de prueba c para $A + B$ y los elementos de prueba $d(x)$ y $e(y)$ para $C(i(x))$ y $C(j(y))$, respectivamente, tal que $x : A$ y $y : B$. Si formulamos ésto con dos reglas de cómputo de obtenemos:

$$\begin{array}{ll}
\mathbf{D}(i(a), (x)d(x), (y)e(y)) & \rightarrow d(a) \text{ (donde } d(a) : C(i(a)) \text{)} \\
\mathbf{D}(j(b), (x)d(x), (y)e(y)) & \rightarrow e(b) \text{ (donde } e(b) : C(j(b)) \text{)}
\end{array}$$

El operador $+$ permite definir la disyunción de la siguiente manera

$$A \vee B = A + B : prop, \text{ provisto que } A : prop \text{ y } B : prop.$$

Lo absurdo \perp

$$\begin{array}{ccc}
\text{-----} \perp F & & x : \perp \\
\perp : set & & \text{-----} \perp E \\
& & \perp : set
\end{array}$$

El símbolo \perp no es otra cosa que el conjunto vacío. Puesto que su regla de formación establece que es siempre un conjunto, la regla establece también que \perp es siempre una proposición. El absurdo no requiere regla de introducción. En efecto, la regla de eliminación para $A \rightarrow \perp$ es la que introduce \perp .

Es evidente que esto se sigue de la perspectiva estándar en la lógica intuicionista por medio de la cual $\sim A$ es visto como una abreviación de $A \rightarrow \perp$. Dado que no hay regla de introducción específica para \perp tampoco hay una regla de igualdad que relacione la eliminación con la introducción. Finalmente, la regla de eliminación expresa la regla de, así llamada, *ex falso sequitur quodlibet*. En efecto, el juicio de que algún x es del tipo \perp es contradictorio dado que \perp es el conjunto vacío. Por ende, de tal juicio podemos concluir que $a : A$ (véase Martin-Löf, 1984, pp. 65 – 67).

Puesto que una de las tesis principales de nuestro trabajo se basa en la idea de que la regla formal de la dialógica, que permite la así llamada jugada de espejo, es en realidad la forma dinámica de la noción de igualdad definicional proveniente de la teoría de tipos, es conveniente presentar en forma sistemática, aunque sea brevemente, tal concepción y su relación con la igualdad como un predicado.

Debido a limitaciones de espacio nos restringiremos aquí a discutir los puntos relevantes para los objetivos del presente trabajo. Para más información sobre nuestra perspectiva, ver Martin-Löf (1984), Ranta (1988, 1994), Nordström et al. (1990), Primiero (2008) y Granström (2011).

II. 2 Afirmaciones e igualdad

Como mencionado anteriormente, siempre que en TCT se introduce una nueva expresión, se hace por medio de una *explicación de significado*. En el caso de la introducción de un nuevo tipo, la explicación de significado consiste en (1) describir sus objetos canónicos, (2) proporcionar un algoritmo a fin de reconocer si un objeto no-canónico es de ese tipo y (3) especificar las condiciones que establecen la identidad (o no) de dos objetos respecto de un tipo. El punto 3 se entiende como la tarea definir una relación de equivalencia apropiada. De este modo las afirmaciones de la forma $a = b : A$, afirman que los dos objetos a y b satisfacen la relación de equivalencia definida para el tipo A . La afirmación $a = b : A$ es también llamada *afirmación de la igualdad definicional*, dado que por medio de ella se introducen definiciones explícitas, por ejemplo de funciones. Una tal igualdad se transmite por reflexividad, simetría y transitividad, y por sustitución de iguales definicionales (Ranta, 1994, p. 52). Más aún, definiciones reales como apuntado por Frege en su crítica a Hilbert, pueden proveer la verdad de una proposición basada en estas definiciones, pero *ellas no son portadoras de verdad*. Para decirlo brevemente, la igualdad definicional *no es* un predicado. La expresión $a = b$ es ambigua a menos que usemos la expresión de la afirmación completa: si se encuentra a la izquierda de los dos puntos ella expresa igualdad definicional y si está a la derecha de los dos puntos una tal igualdad debe entenderse como un predicado.

Cuando un tipo es una proposición constituida por constantes lógicas tales como la conjunción, la disyunción y otras, el rol inferencial de las igualdades a la izquierda de los dos puntos es armonizar el proceso de síntesis y análisis que despliega la explicación de significado de las constantes lógicas en juego. En la siguiente sección discutiremos muy brevemente el rol de la igualdad definicional en el proceso de armonización (para una discusión en profundidad Rahman/Redmond, 2015b)

II.2.1 Elementos de prueba, igualdad y su rol inferencial

En su histórico artículo de 1935 sobre deducción natural, Gerhard Gentzen señaló:

Las introducciones representan, por así decirlo, las “definiciones” de los símbolos referidos, y las eliminaciones no son más, en el análisis final, que la consecuencia de estas definiciones.⁹ (Gentzen 1969, p.80)

La idea detrás de la observación de Gentzen es que una regla de introducción de la deducción exhibe el fundamento para la afirmación de una proposición que contiene la constante lógica en cuestión, y que el resultado de la utilización de la regla de eliminación correspondiente muestra exactamente esos componentes de la proposición que la regla de introducción especificó como suficientes para su afirmación. Dummett llama *armonía* a tal coordinación. En breve y haciendo uso de la formulación de Stephen Read (2008, p. 291, nota 5), una regla de eliminación es armónica, si esta regla no hace más que explicar las consecuencias del significado conferido por la regla de introducción. Si comenzamos con las reglas de introducción, una manera de ver la contribución del concepto de armonía es que mientras las reglas de introducción muestran cómo *sintetizar* a partir de ciertas premisas, una expresión que contiene una constante lógica dada, la regla de eliminación muestra cómo *analizar* la expresión en juego mostrando exactamente aquellos componentes que son requeridos por las reglas de introducción para afirmar esta expresión.

El lenguaje explícito de la TCT permite expresar la condición de armonía por medio de igualdades definicionales. Tomemos como ejemplo el caso de la conjunción. La proposición $A \wedge B$ (o con el conjunto $A \times B$) se explica, estableciendo que un elemento canónico de $A \wedge B$ es un par de elementos de prueba (a, b) donde $a : A$ y $b : B$ – es decir, donde a es un elemento de prueba A y b de B :

$$\begin{array}{c} a : A \quad b : B \\ \text{-----} \quad \wedge \text{-introducción} \\ (a, b) : A \wedge B \end{array}$$

Con el fin de definir \wedge -eliminaciones vamos a hacer uso de cierto tipo de operadores llamados *selectores*, a partir del cual se pueden definir nuevas funciones que extraen aquellos componentes que constituyen un elemento de prueba complejo c (como por ejemplo $c = (a, b)$). En el caso de la conjunción los selectores son las funciones de proyección p y q que tienen como valor el lado izquierdo y derecho del par de elementos de prueba respectivamente. Por lo tanto, si c es un elemento de prueba para la conjunción,, entonces $p(c)$ nos da el componente izquierdo de c y $q(c)$ su componente derecho.

$$\begin{array}{c} c : A \wedge B \\ \text{-----} \\ p(c) : A \end{array} \quad \wedge \text{-}p\text{-eliminación} \quad \begin{array}{c} c : A \wedge B \\ \text{-----} \\ q(c) : B \end{array} \quad \wedge \text{-}q\text{-eliminación}$$

Si sabemos que $c = (a, b)$, entonces $p(c)$ restaura el componente izquierdo de c (obtenido por la regla de introducción) esto es: $p(c) = p((a, b)) = a$, tal que $a : A$, análogamente $q(c)$ restaura el componente derecho

$$\begin{array}{c} a : A \quad b : B \\ \text{-----} \quad \wedge \text{-}izq\text{-}\beta\text{-} \\ p((a, b)) = a : A \end{array} \quad \text{igualdad} \quad \begin{array}{c} a : A \quad b : B \\ \text{-----} \quad \wedge \text{-}der\text{-}\beta\text{-} \\ q((a, b)) = b : B \end{array} \quad \text{igualdad}$$

⁹ En el original: *The introductions represent, as it were; the ‘definitions’ of the symbols concerned, and the eliminations are no more, in the final analysis; than the consequence of these definitions.*

Tales reglas proveen ejemplos claros de cómo usar la noción de igualdad definicional mencionada anteriormente: las funciones de proyección p y q se definen explícitamente por medio de una regla de inferencia de modo que, dados los elementos de prueba a y b , la proyección q de (a, b) es definicionalmente idéntica a b , respecto a la relación de equivalencia que define el tipo B , y análogamente se introduce en la proyección p . Estas igualdades pueden ser utilizadas para describir las reglas de cómputo de la proyecciones p y q en relación al par (x, y) , tal que $x : A, y : B$.

También se puede introducir una regla dual, llamada η , de la siguiente manera:

$$\frac{c : A \wedge B}{(p(c), q(c)) = c : A \wedge B} \text{---}\wedge\text{-}\eta\text{-igualdad}$$

En suma, en nuestro contexto ambas reglas de igualdad pueden ser vistas como aquello que asegura que las reglas son armoniosas. Es decir, si la conjunción ha sido compuesta por (a, b) entonces ambas reglas de eliminación proveen un análisis del elemento de prueba c del que resultan los componentes a y b .

II. 2.2 Igualdad definicional y proposicional

II. 2.2.1 La igualdad definicional

La igualdad definicional cumple con las condiciones básicas de la igualdad:

Reflexividad

$$\frac{a : A}{a = a : A}$$

$$\frac{A : set}{A = A : set}$$

Simetría

$$\frac{a=b : A}{b = a : A}$$

$$\frac{A = B : set}{B = A : set}$$

Transitividad

$$\frac{a=b : A \quad b=c : A}{a = c : A}$$

$$\frac{A = B : set \quad B = C : set}{A = C : set}$$

Extensionalidad

$$\frac{A = B : set \quad a : A}{a : B}$$

$$\frac{A = B : set \quad a=b : A}{a = b : B}$$

En el contexto de juicios hipotéticos, la igualdad definicional entre dos elementos de prueba, por ejemplo a y c , digamos, en el conjunto A , permite obtener la igualdad entre las funciones $b(a)$ y

$b(c)$, que se infiere de la sustitución de la variable x en la función $b(x)$ de A a $B(x)$ por ambas, a y c :¹⁰

Sustitución

$$\frac{a = c : A \quad b(x) : B(x)(x : A)}{\text{-----}} \quad \frac{a = c : A \quad B(x) : \text{set}(x : A)}{\text{-----}}$$

$$b(a) = b(c) : B(a) \quad B(a) = B(c) : \text{set}$$

Los juicios hipotéticos también permiten expresar igualdades hipotéticas que también se rigen por una regla de sustitución adecuada:

Igualdades definicionales entre objetos dependientes

$$\frac{a : A \quad c(x) = b(x) : B(x)(x : A)}{\text{-----}}$$

$$b(a) = c(a) : B(a)$$

Las reglas de sustitución para igualdades en el contexto de juicios hipotéticos requieren asimismo una noción general de sustitución de variables expresadas por la siguiente regla:

Regla de sustitución general para juicios hipotéticos

$$\frac{a : A \quad b(x) : B(x)(x : A)}{\text{-----}}$$

$$b(a) : B(a)^{11}$$

Veamos ahora como en el cuadro de la TCT se relacionan las nociones de igualdad definicional e igualdad proposicional.

II.2.2.2 Igualdad proposicional: Los predicados de igualdad intensional y extensional.

Discutimos ahora la formulación de reglas que muestren al mismo tiempo cómo definir un predicado de igualdad y cómo pasar de la igualdad definicional a un tal predicado. En realidad se pueden definir dos predicados de igualdad, uno intensional y otro extensional. El elemento de prueba correspondiente al predicado de identidad extensional no depende de los elementos de prueba de las premisas.

- **El predicado de igualdad intensional**

El predicado de igualdad proposicional intensional definido sobre un conjunto, digamos A , se expresa usualmente con la notación $\mathbf{Id}(A, a, b)$. A veces se utiliza la siguiente notación $a =_A b$, sin embargo nosotros usaremos la primera pues manifiesta explícitamente que \mathbf{Id} es una relación.

Formación

$$A : \text{set} \quad a : A \quad b : A$$

¹⁰ Presentamos aquí las reglas de **sustitución** para igualdades entre los **elementos de un conjunto**, sin embargo se pueden definir de forma análoga la igualdad entre elementos del tipo *set* – es decir las igualdades entre conjuntos (véase Nordström/Peterson/Smith, 1990, p. 39).

¹¹ En realidad esta regla no es suficiente, pues no **incluye** el caso de sustituciones simultáneas que son necesarias en casos como $C(x, y) : \text{set}(x : A, y : B(x))$. Hay varias soluciones a este respecto, la más práctica desde el **punto de vista** procedural (tan claro en la perspectiva dialógica), es permitir sustituciones parciales en un contexto. Volviendo a nuestro ejemplo, si es el caso que $a : A$, *llevamos a cabo una primera aplicación de la sustitución para obtener* $C(a, y) : \text{set}(y : B(a))$, y luego una segunda vez para obtener $b : B(a)$ y $C(a, b) : \text{set}$ (véase Nordström/Peterson/Smith, 1990, p. 39).

$$\frac{}{\mathbf{Id}(A, a, b) : set}$$

Como hace notar Thompson (1999, p. 110) esta regla de formación es diferente de las otras reglas de formación de tipos. Las últimas toman la forma

$$\frac{\dots : type \quad \dots : type}{\dots : type}$$

Este género de reglas permite identificar los tipos que corresponden a una expresión dada independientemente de cuáles sean los elementos de prueba que le pertenecen a estos tipos. La regla de formación **Id** no posee esta propiedad dado que ella establece que $\mathbf{Id}(A, a, b)$ es un tipo si a y b son de tipo A . Esto significa que la regla que hace explícita los tipos presupuestos en $\mathbf{Id}(A, a, b)$ está inextricablemente atada a reglas de inferencia con elementos de prueba específicos.

La regla de introducción básica introduce reflexividad y proviene del hecho que dada la afirmación $a : A$, tenemos inmediatamente $a = a : A$. Desde el punto de vista inferencial, esto equivale a introducir el caso de la reflexividad $\mathbf{Id}(A, a, a)$ provisto que la premisa esté constituida por la afirmación $a : A$.

$$\frac{a : A}{r(a) : \mathbf{Id}(A, a, a)}$$

El elemento de prueba para $\mathbf{Id}(A, a, a)$ es $r(a)$. La estructura interna de este elemento de prueba es nada más y nada menos que su dependencia de a . Por ende, lo que hace que la proposición $\mathbf{Id}(A, a, a)$ sea verdadera es el operador r , que cuando se aplica a a , produce a .

Mediante la aplicación de la regla de sustitución en conjuntos a $a = b : A$ y $\mathbf{Id}(A, a, x) : prop (x : A)$, obtenemos $\mathbf{Id}(A, a, a) = \mathbf{Id}(A, a, b) : prop$, de tal igualdad más el resultado de la aplicación de la regla de introducción-**Id** a a , obtenemos, por medio de la igualdad entre conjuntos $r(a) : \mathbf{Id}(A, a, b)$.

$$\frac{\begin{array}{l} A : set \quad a : A \quad x : A \\ \text{-----} \mathbf{Id-F} \\ a = b : A \quad \mathbf{Id}(A, a, x) : prop \\ \text{-----} \text{sust}^1 \end{array} \quad \begin{array}{l} a : A \\ \text{-----} \mathbf{Id-introducción} \end{array}}{\mathbf{Id}(A, a, a) = \mathbf{Id}(A, a, b) : prop \quad r(a) : \mathbf{Id}(A, a, a)} \text{-----set-igualdad}$$

$$r(a) : \mathbf{Id}(A, a, b)^{12}$$

En otras palabras, de $a : A$ y $a = b : A$ obtenemos también la introducción del caso no-reflexivo **Id**¹³.

$$\frac{a : A \quad a = b : A}{r(a) : \mathbf{Id}(A, a, b)} \text{-----} \mathbf{Id-Introducción}$$

12 Cf. Ranta, 1994, p. 53.

13 Cf. Nordström/Peterson/Smith, 1990, p. 57.

Eliminación y sustitución

La regla de eliminación es una generalización de la regla de sustitución de Leibniz, sin hacer uso de la cuantificación de segundo orden – presupuesta en las formulaciones estándar de tal regla. La idea rectora detrás de la regla de sustitución de Leibniz es que aquellas expresiones que el predicado de identidad establece como iguales respecto a un conjunto dado, pueden ser sustituidas mutuamente en el seno de un tal conjunto. Supongamos que tenemos una prueba p de una proposición C en la que ocurre a , y supongamos también que sabemos que $c : \mathbf{Id}(A, a, b)$. Bajo estas condiciones, estamos capacitados entonces para inferir la proposición C' resultante de reemplazar (al menos) alguna de las ocurrencias de a en C por b . Para capturar la idea de sustitución por alguna de las ocurrencias, pensamos en C teniendo la forma $C[a/x, a/y]$ en la cual a reemplaza dos variables libres x e y . De hecho, más generalmente, podemos también pensar una proposición C en la que ocurra $r(a)$. La idea subyacente es que toda proposición en la cual se lleva a cabo una sustitución respecto a \mathbf{Id} , puede incluir aquellas proposiciones P en la cuales $r(x)$ ocurre explícitamente, y por ende tales P pueden ser llevados a la forma $C[a/x, a/y, r(a/x)]$. Estas consideraciones dan paso a la siguiente regla

$$\begin{array}{l}
 a : A \\
 b : A \\
 c : \mathbf{Id}(A, a, b) \\
 C[x, y, z] : \text{set}(x : A, y : A, z : \mathbf{Id}(A, a, b)) \\
 d(x) : C[x, x, r(x)](x : A) \\
 \hline
 id(c, d) : C[a, b, c]
 \end{array}
 \quad \mathbf{Id}\text{-eliminación}$$

El operador id en $id(c, d)$ indica que se está aplicando la identidad $c : \mathbf{Id}(A, a, b)$ a d , en nuestro caso $d(a)$ y esta regresa el elemento de prueba $d(a)$ mismo. Esto sugiere las siguientes reglas que determinan cuándo dos $id(c, d)$ son iguales:

$$\begin{array}{l}
 a : A \\
 C[x, y, z] : \text{set}(x : A, y : A, z : \mathbf{Id}(A, a, b)) \\
 d(x) : C[x, x, r(x)](x : A) \\
 \hline
 id(r(a), d) = d(a) : C[a, b, r(a)]
 \end{array}
 \quad \mathbf{Id}\text{-eliminación}$$

Veamos ahora algunos casos que ejemplifican el uso de la regla de eliminación. Partiremos mostrando que la ley de sustitución de Leibniz puede ser inferida de la regla de eliminación. Como ya hemos mencionado, la idea de la regla de Leibniz es que las premisas $B(a)$ y $\mathbf{Id}(A, a, b)$, nos permiten concluir $B(b)$ reemplazando a con b en $B(a)$.

Ley de Sustitución de Leibniz

Se asume que $(a, b) : A$ y $c : \mathbf{Id}(A, a, b)$ y también que $p : B(a)$.

La primera tarea a llevar a cabo es inferir a partir de $B(x)$ alguna expresión que tenga la forma $C[x, x, r(x)]$ – puesto que es ésta la forma que necesitamos para aplicar la regla de eliminación de \mathbf{Id} . El modo más directo de obtener una expresión que posea tal forma es inferir $B(x) \rightarrow B(x)$ a partir de $B(x)$ – en este caso, la proposición tiene la forma $C[x, x]$ que no menciona $r(a)$. Entonces dado que el elemento de prueba de $B(x) \rightarrow B(x)$ es la función de identidad, dado $\lambda x.x$ obtenemos $\lambda x.x : B(x)$

$\rightarrow B(x)$. De esta última afirmación y las premisas $a : A$ y $b : A$, podemos aplicar la regla de eliminación de **Id** para obtener $id(c, (x)\lambda x.x) : B(a) \rightarrow B(b)$.

$$\begin{array}{c}
 \begin{array}{c}
 x : B(x) \\
 \hline
 \rightarrow I
 \end{array} \\
 \begin{array}{c}
 a : A \quad b : A \quad c : \mathbf{Id}(A, a, b) \quad \lambda x.x : B(x) \rightarrow B(x) (x : A) \\
 \hline
 \text{Id-eliminación} \\
 id(c, (x)\lambda x.x) : B(a) \rightarrow B(b)
 \end{array}
 \end{array}$$

Haciendo uso de la premisa $p : B(a)$ y aplicando la regla de eliminación de la implicación, obtenemos $subst(c, p) : B(b)$, donde $subst(c, p)$ es una forma de abreviar la expresión $ap(id(c, (x)\lambda x.x), p)$.

$$\begin{array}{c}
 \begin{array}{c}
 x : B(x) \\
 \hline
 \text{III}
 \end{array} \\
 \begin{array}{c}
 a : A \quad b : A \quad c : \mathbf{Id}(A, a, b) \quad \lambda x.x : B(x) \rightarrow B(x) (x : A) \\
 \hline
 \text{Id-elimination} \\
 id(c, (x)\lambda x.x) : B(a) \rightarrow B(b) \quad p : B(a) \\
 \hline
 \text{PIE} \\
 ap(id(c, (x)\lambda x.x), p) : B(b)
 \end{array}
 \end{array}$$

Esto nos lleva a la siguiente formulación de la regla de Leibniz:

$$\begin{array}{c}
 B(x) : set(x : A) \quad a:A \quad b:A \quad c : \mathbf{Id}(A, a, b) \quad p : B(a) \\
 \hline
 subst(c, p) : B(b)
 \end{array}$$

También **Id** es transmitida por reflexividad, transitividad y simetría. Esto se sigue de lo anterior de las siguientes consideraciones:

Puesto que la regla de introducción nos la reflexividad directamente, nos queda aún por mostrar que la regla de eliminación nos permite obtener simetría y transitividad.

Simetría: Asumamos que $a, b : A$, y que $d : \mathbf{Id}(A, a, b)$; y consideremos que C es $e : \mathbf{Id}(A, y, x)$. Dado que $a : A$ es una de las premisas asumidas, obtenemos de ella (por medio de la regla de introducción de **Id**) $r(a) : \mathbf{Id}(A, a, a)$. Aplicando ahora la regla de eliminación de **Id**, obtenemos $id(d, r(a)) : \mathbf{Id}(A, b, a)$. Esto permite anotar la siguiente regla para simetría:

$$\begin{array}{c}
 d : \mathbf{Id}(A, a, b) \\
 \hline
 symm(d) : \mathbf{Id}(A, b, a)
 \end{array}$$

Donde $symm(d)$ es una abreviación de $id(d, r(a))$.

Transitividad: el paso principal para inferir la transitividad de **Id** está en considerar que C es $\lambda y.y : \mathbf{Id}(A, y, c) \rightarrow \mathbf{Id}(A, x, c)$. De las premisas $d : \mathbf{Id}(A, a, b)$ y $e : \mathbf{Id}(A, b, c)$, obtenemos $id(d, (x)\lambda y.y) : \mathbf{Id}(A, b, c) \rightarrow \mathbf{Id}(A, a, c)$ como resultado de la aplicación de la regla de eliminación de **Id**. Por Π -

eliminación -haciendo uso de $e : \mathbf{Id}(A, b, c)$ – Obtenemos $\mathit{trans}(d, e) : \mathbf{Id}(A, a, c)$, donde $\mathit{trans}(d, e)$ abrevia la expresión $\mathit{ap}(\mathit{id}(d, (x)\lambda y.y), e)$. Esto nos conduce a la formulación de la siguiente regla:

$$\frac{d : \mathbf{Id}(A, a, b) \quad e : \mathbf{Id}(A, b, c)}{\mathit{trans}(d, e) : \mathbf{Id}(A, a, c)}$$

• El predicado de igualdad extensional

La regla de formación del predicado de igualdad extensional entre a y b definida en el conjunto A ($\mathbf{Eq}(A, a, b)$) es análoga a la del predicado de identidad intensional.

$$\frac{A : \mathit{set} \quad a : A \quad b : A}{\mathbf{Eq}(A, a, b) : \mathit{set}} \mathbf{EqF}$$

Introducción de \mathbf{Eq}

El elemento de prueba \mathbf{Eq} que verifica $\mathbf{Eq}(A, a, b)$, no depende de a ni de b . La verificación de $\mathbf{Eq}(A, a, b)$ solo necesita de la identidad de algún elemento de prueba arbitrario. Sea eq entonces por ejemplo $s = s$.

$$\frac{a = b : A}{eq : \mathbf{Eq}(A, a, b)} \mathbf{EqI}$$

La regla representa la forma más débil de relacionar inferencialmente la igualdad ontológica con la proposicional. En efecto el elemento de prueba eq no contiene, ni depende de ninguno de los elementos de prueba sobre los que se basa la afirmación de la premisa. Esta falta de lazos entre los elementos de prueba de la premisa y la conclusión se hace aún más evidente en las siguientes reglas de eliminación no inductivas:

Eliminación de \mathbf{Eq}

$$\frac{c : \mathbf{Eq}(A, a, b)}{a = b : A} \mathbf{EqE}^1$$

En esta regla de eliminación el elemento de prueba c de la premisa no contiene nada que pueda conducir a la igualdad $a = b$ de la conclusión. Es más, dado que la igualdad expresada por $\mathbf{Eq}(A, a, b)$ es identidad numérica, todo elemento c de $\mathbf{Eq}(A, a, b)$ es igual a eq .¹⁴

$$\frac{c : \mathbf{Eq}(A, a, b)}{c = eq : \mathbf{Eq}(A, a, b)} \mathbf{EqE}^2$$

Se puede mostrar que con ambas reglas de eliminación podemos obtener una regla de sustitución análoga a la del predicado de identidad intensional y, por tanto, obtener también reglas de transmisión análogas¹⁵.

14 Véase Nordström /Peterson/Smith (1990), p. 61.

15 Véase Nordström /Peterson/Smith (1990), p. 61.

$a : A$
 $b : A$
 $c : \mathbf{Eq}(A, a, b)$
 $C[x, y, z] : \text{set } (x : A, y : A, z : \mathbf{Eq}(A, a, b))$
 $d(x) : C[x, x, \text{eq}] (x : A)$
----- \mathbf{EqE}^3
 $d(a) C[a, b, c]$