APORTACIONES MATEMÁTICAS
Memorias **40** (2009) 1–31
Artículo de Exposición

LA LÓGICA PROPOSICIONAL DE SEGUNDO ORDEN

FAVIO EZEQUIEL MIRANDA PEREA

RESUMEN. En este artículo expositorio se presenta un panorama general de la lógica proposicional de segundo orden Prop2, la cual se obtiene al extender la lógica proposicional usual con un mecanismo de cuantificación universal y existencial sobre proposiciones. Tras dar una descripción matemática concisa tanto de la sintaxis como de las semánticas clásica e intuicionista, la exposición se centra en la relación entre Prop2 y la lógica de predicados de segundo orden, así como en ciertas aplicaciones relevantes para las Ciencias Computacionales.

1. INTRODUCCIÓN

Cuando se habla de Lógica Matemática, generalmente nos viene a la mente la lógica proposicional, donde no hay una representación separada ni una distinción explícita entre individuos y sus propiedades, o bien la lógica de predicados de primer orden que incluye términos para representar individuos, así como predicados y cuantificación sobre estos. Si hablamos de lógica de segundo orden usualmente nos referimos a una extensión de la lógica de predicados de primer orden que permite cuantificación sobre predicados y funciones. Por ejemplo el principio de inducción para números naturales se expresa en lógica de predicados de segundo orden mediante la fórmula $\forall R(\forall x(Rx \to Rsx) \to (R0 \to \forall yRy))$ donde x, y son variables de primer orden cuyo dominio de interpretación intencional es el conjunto de los números naturales, R es una variable de segundo orden, 0 es una constante de primer orden y s es una constante de segundo orden o bien un símbolo de función.

Si bien la lógica de predicados de segundo orden Pred2 es claramente más expresiva que la lógica proposicional, en presencia de la cuantificación de segundo orden muchas características de primer orden pierden importancia y diversas propiedades estructurales y de expresividad de la lógica pueden estudiarse en el fragmento proposicional, al cual llamamos lógica proposicional de segundo orden, denotada Prop2. Esta lógica se obtiene agregando cuantificadores directamente a la lógica proposicional, es decir, los cuantificadores operan sobre variables proposicionales consideradas predicados de primer orden que reciben cero argumentos, de aquí el calificativo "segundo orden".

El propósito principal de este artículo es presentar de manera rigurosa la sintaxis y semántica de la lógica proposicional de segundo orden para posteriormente enfatizar su

---

2000 *Mathematics Subject Classification.* Primary 03B20, 03B70, 03D70; Secondary 68Q65.
Key words and phrases. lógica proposicional, segundo orden, sistema F, modelos de Kripke, deducción natural, tipos de datos inductivos, tipos de datos abstractos.

poder expresivo así como su relación con la lógica de predicados de segundo orden y su relevancia en las Ciencias Computacionales.

La exposición se estructura como sigue: la sección 2 se dedica a preliminares, presentamos la lógica proposicional usual utilizando un sistema de deducción natural con contextos de fórmulas y haciendo énfasis en los sistemas minimal, intuicionista y clásico obtenidos de acuerdo a las propiedades postuladas de la negación; además recordamos la semántica clásica y presentamos con cierto detalle la semántica intuicionista motivada por la interpretación intuitiva de Brouwer-Heyting-Kolmogorov (BHK) que formalizamos mediante modelos de Kripke. En la sección 3 motivamos y definimos formalmente a la lógica proposicional de segundo orden $\mathsf{Prop2}$ utilizando nuevamente sistemas de deducción natural con contextos para después, en la sección 4, mostrar que la semántica clásica es trivial, puesto que los cuantificadores son definibles, y dedicarnos a la semántica impredicativa intuicionista de modelos completos de Kripke. Como ejemplo extendido mostramos a detalle la invalidez intuicionista del llamado axioma de Gabbay. La sección 5 se ocupa del fragmento $\mathsf{Prop2}^{\rightarrow,\forall}$ obtenido al considerar proposiciones que involucran únicamente a la implicación y al cuantificador universal, el poder expresivo de este fragmento se pone en evidencia al desarrollar definiciones para el resto de los conectivos, lo cual es imposible en la lógica proposicional intuicionista sin cuantificadores. Estas y otras definiciones se obtienen a partir de definiciones intuitivas formalizadas en la lógica de predicados de segundo orden y eliminando posteriormente las referencias a objetos de primer orden, haciendo explícita así una fuerte relación entre ambas lógicas la cual se discute en la sección 6. La relevancia de $\mathsf{Prop2}$ en las Ciencias Computacionales se manifiesta en la sección 7 mediante el desarrollo de definiciones de tipos de datos inductivos y abstractos. Finalmente en la sección 8 ofrecemos algunos comentarios finales para el lector interesado en profundizar en estos temas.

2. Preliminares

Los conceptos presentados en esta sección, si bien se suponen conocidos, se desarrollan hasta cierto punto para recordar, fijar la notación y hacer autocontenida la exposición. Sin embargo cualquier concepto utilizado y no definido aquí debe consultarse por ejemplo en [16, 2]

2.1. Acerca del orden. En la descripción de sistemas lógicos el concepto de orden se refiere a una clasificación de las variables para distinguir sobre cuales se permite cuantificar. En el caso de la lógica de predicados de primer orden, el nombre indica que la cuantificación se permite sobre variables que denotan individuos, las cuales se llaman variables de primer orden. Como ejemplo tenemos la fórmula $\forall x(Px \to Pfx)$ donde P es un símbolo de predicado de índice uno y f un símbolo de función también de índice uno definidos en una signatura dada, aunque también podemos considerar a P y a f como variables que representen predicados o funciones, las cuales reciben objetos de primer orden como argumentos. De esta forma dichas variables son llamadas de segundo orden y si permitimos la cuantificación sobre ellas obtenemos la lógica de predicados de segundo orden que admite por ejemplo a la fórmula $\forall P\forall f\forall x(Px \to Pfx)$. Ahora bien, si consideramos que las proposiciones no son más que predicados que reciben cero argumentos entonces las variables proposicionales son variables de segundo orden, de manera que si admitimos cuantificarlas obtenemos nuestro objeto de estudio, la lógica proposicional de segundo orden habitada por fórmulas como $\forall p(p \to p)$.

El lector tal vez se pregunte en este momento cual es el orden de la lógica proposicional usual, es decir, la lógica proposicional sin cuantificadores. Al no haber cuantificación explícita puede considerarse una lógica de orden cero, aunque también se conoce como lógica proposicional de primer orden debido a que tanto los objetos de primer orden como la cuantificación sobre estos figura implicitamente en el significado dado a una variable proposicional en el lenguaje natural. Sin embargo, es más común hablar simplemente de lógica proposicional sin otorgarle un calificativo de orden. Un estudio matemático formal de lógicas de diversos órdenes tanto de predicados como de proposiciones se ha desarrollado en [4].

2.2. Sintaxis de la lógica proposicional. Presentamos la lógica proposicional usual mediante un sistema de deducción natural con hipótesis localizadas, es decir, con contextos de fórmulas.

Definición 2.1. Sea $\mathsf{VProp} = \{p,q,r,\ldots\}$ un conjunto infinito numerable de variables proposicionales. El conjunto de *fórmulas* de la lógica proposicional es el mínimo conjunto FProp tal que:

- $\mathsf{VProp} \subseteq \mathsf{FProp}$.
- $\bot \in \mathsf{FProp}$.
- Si $A, B \in \mathsf{FProp}$ entonces $(A \to B), (A \vee B), (A \wedge B) \in \mathsf{FProp}$.

Se observan las siguientes convenciones:

- Como es usual se omiten los paréntesis más externos y se define $A \leftrightarrow B =_{def} (A \to B) \wedge (B \to A)$.
- La implicación se asocia a la derecha, es decir, $A \to B \to C$ significa $A \to (B \to C)$.
- La precedencia de operadores de mayor a menor es $\wedge, \vee, \to$ sin distinción entre $\wedge$ y $\vee$. De manera que $p \vee r \to q$ significa $((p \vee r) \to q)$.
- La relación de equivalencia lógica se denota con $\equiv$. Por ejemplo $A \vee B \equiv B \vee A$.

Obsérvese que la negación no esta presente en nuestra definición de fórmulas, su inclusión se hará más adelante.

La noción de sustitución, que en este caso no es más que una operación de sustitución textual puramente mecánica será de importancia más adelante.

Definición 2.2. La *sustitución* de una variable proposicional p por la fórmula B en la fórmula A, denotada $A[p := B]$ se define recursivamente como sigue:

- $p[p := B] = B, \quad q[p := B] = q$
- $\bot[p := B] = \bot$
- $(A \to C)[p := B] = A[p := B] \to C[p := B]$
- $(A \wedge C)[p := B] = A[p := B] \wedge C[p := B]$
- $(A \vee C)[p := B] = A[p := B] \vee C[p := B]$

Definición 2.3. Un *contexto* es un conjunto de fórmulas $\Gamma = \{A_1, \ldots, A_n\}$.

En el manejo de contextos convenimos en omitir las llaves de conjunto así como la operación de unión escribiendo Γ, Δ en vez de $\Gamma \cup \Delta$ y Γ, A en vez de $\Gamma \cup \{A\}$. Mas aún, en un contexto de la forma Γ, A suponemos que A no figura en Γ.

Definición 2.4. Un *secuente* es una expresión de la forma $\Gamma \vdash A$ donde Γ es un contexto y A es una fórmula. En particular el secuente $\vdash A$ significa $\varnothing \vdash A$.

El secuente $\Gamma \vdash A$ expresa la relación de derivabilidad de la fórmula A a partir de las hipótesis dadas en Γ. Esta relación se define formalmente mediante las siguientes reglas de inferencia entre secuentes.

Definición 2.5. La relación de *derivabilidad* $\Gamma \vdash A$ se define recursivamente como sigue:

- Regla de inicio:
$$\frac{}{\Gamma, A \vdash A}\ (Hip)$$
- Implicación:
$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ (\to I) \qquad \frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\ (\to E)$$
- Conjunción:
$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ (\wedge I) \quad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ (\wedge_1 E) \quad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ (\wedge_2 E)$$
- Disyunción
$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ (\vee_1 I) \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ (\vee_2 I)$$
$$\frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\ (\vee E)$$

Una derivación de un secuente particular se define como sigue:

Definición 2.6. Una *derivación* del secuente $\Gamma \vdash A$ es una sucesión finita de secuentes $\Gamma_1 \vdash A_1, \ldots, \Gamma_n \vdash A_n$ tal que:

- $\Gamma_i \vdash A_i$ es instancia de la regla (Hip) ó
- $\Gamma_i \vdash A_i$ es conclusión de alguna regla de inferencia tal que las premisas necesarias figuran antes en la sucesión.
- $\Gamma \vdash A$ es el último elemento de la sucesión.

Obsérvese que a cada conectivo le corresponden reglas que lo introducen, denotadas con I, así como reglas que lo eliminan, denotadas con E. Esta simetría o dualidad en las reglas es de gran importancia y proporciona un determinismo en la relación de derivabilidad. Cada fórmula compuesta puede ser derivada usando una única regla de introducción. Así mismo, la información de cada fórmula compuesta puede ser utilizada para derivar otras fórmulas mediante una única regla de eliminación.

Las siguientes propiedades estructurales son de importancia.

Proposición 2.7. *Las reglas de inferencia*
$$\frac{\Gamma \vdash A}{\Gamma, B \vdash A}\ (Mon.) \qquad \frac{\Gamma \vdash A}{\Gamma[p := B] \vdash A[p := B]}\ (Sust.)$$

son derivables. En el caso de la regla $(Sust)$*, si* $\Gamma = \{A_1, \ldots, A_n\}$ *entonces* $\Gamma[p := B] =_{def} \{A_1[p := B], \ldots, A_n[p := B]\}$.

Demostración. Inducción sobre $\vdash$. □

A continuación nos ocupamos de la negación.

2.3. La negación. En la presencia de $\perp$ no es necesario agregar a la negación como un conectivo independiente sino que esta operación puede definirse como

$$\neg A =_{def} A \to \perp.$$

De acuerdo a las reglas de inferencia que se postulen para su manejo existen tres sistemas lógicos distintos:

- Lógica Minimal: no hay reglas de inferencia para la negación ni para $\perp$. En un sistema minimal la negación $\neg$ y la constante $\perp$ pueden estar presentes pero no tienen propiedades específicas. En particular probar $\vdash\ \perp$ no implica la inconsistencia del sistema.
- Lógica Intuicionista o Constructiva: se obtiene al agregar a la lógica minimal la siguiente regla:
$$\frac{\Gamma \vdash \perp}{\Gamma \vdash A}\ (\perp E)$$
conocida también como *ex-falso-quodlibet*.
- Lógica Clásica: en este caso la negación $\neg$ puede considerarse nuevamente como abreviatura o como un conectivo nuevo regido por la regla del tercero excluido:
$$\frac{}{\Gamma \vdash A \vee \neg A}\ (TE)$$
o equivalentemente por alguna de las siguientes reglas:
  - Eliminación de la doble negación (negación clásica):
  $$\frac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A}\ (\neg C)$$
  - Regla de prueba por contradicción o reducción al absurdo:
  $$\frac{\Gamma, \neg A \vdash \perp}{\Gamma \vdash A}\ (\neg RAA)$$

Obsérvese que la regla (TE) permite derivar el secuente $\vdash A \vee \neg A$ situación imposible de motivar en el ámbito constructivo. Esta situación rompe con la simetría dada por las reglas de introducción y eliminación de los conectivos dadas en la definición (2.5). Más aún, en la lógica clásica podemos obtener disyunciones por medio de una regla distinta a la regla de introducción de la disyunción, a saber mediante el tercero excluido. De esta manera se destruye el determinismo de la relación de derivabilidad.

Dado que centraremos nuestro interés en la lógica intuicionista, reservamos el símbolo $\vdash$ para este sistema, utilizando $\vdash_c$ para la lógica clásica y $\vdash_m$ para la lógica minimal. De las definiciones es claro que el sistema intuicionista es una extensión conservativa del minimal y el clásico del intuicionista. Es decir, $\Gamma \vdash_m A$ implica $\Gamma \vdash A$, que a su vez implica $\Gamma \vdash_c A$. Sin embargo ninguna de las afirmaciones recíprocas es válida en general.

La siguiente proposición proporciona diversos ejemplos de fórmulas derivables en los tres sistemas.

Proposición 2.8. *Los siguientes secuentes son derivables:*

- $\vdash_m A \to \neg\neg A$
- $\vdash_m \neg\neg(A \vee \neg A)$
- $\vdash_m (A \to B) \to (A \to \neg B) \to \neg A$
- $\vdash_m \neg(A \wedge \neg A)$
- $\vdash_m (\neg A \to A) \to \neg\neg A$
- $\vdash_m A \wedge \neg B \to \neg(A \to B)$
- $\vdash_m (A \to \neg A) \to \neg A$
- $\vdash_m (A \to B) \to (\neg B \to \neg A)$
- $\vdash_m (A \to \neg B) \leftrightarrow (B \to \neg A)$
- $\vdash_m \neg(A \vee B) \leftrightarrow \neg A \wedge \neg B$
- $\vdash_m \neg A \vee \neg B \to \neg(A \wedge B)$
- $\vdash A \to \neg A \to B$
- $\vdash A \vee B \to \neg A \to B$
- $\vdash \neg A \vee B \to A \to B$
- $\vdash A \vee \neg A \to \neg\neg A \to A$
- $\vdash \neg\neg A \to \neg A \to A$
- $\vdash A \to (B \leftrightarrow \neg A \vee B)$
- $\vdash_c \neg\neg A \leftrightarrow A$
- $\vdash_c (A \to B) \leftrightarrow \neg B \to \neg A$
- $\vdash_c \neg(A \wedge B) \leftrightarrow \neg A \vee \neg B$
- $\vdash_c (A \to B) \leftrightarrow \neg A \vee B$
- $\vdash_c (A \to B) \vee (B \to A)$
- $\vdash_c ((A \to B) \to A) \to A$

Demostración. Probamos sólo algunos incisos, dejando los restantes como ejercicio.

- $\vdash_m A \to \neg\neg A$. Basta mostrar que $A, A \to \bot \vdash \bot$.

$$\begin{array}{lll} 1. & A, A \to \bot \vdash A & (Hip) \\ 2. & A, A \to \bot \vdash A \to \bot & (Hip) \\ 3. & A, A \to \bot \vdash \bot & (\to E)\ 1,2 \end{array}$$

- $\vdash_m \neg\neg(A \vee \neg A)$. Basta derivar $A \vee \neg A \to \bot \vdash_m \bot$.

$$\begin{array}{lll} 1. & A \vee \neg A \to \bot, A \vdash A & (Hip) \\ 2. & A \vee \neg A \to \bot, A \vdash A \vee \neg A & (\vee I)\ 1 \\ 3. & A \vee \neg A \to \bot, A \vdash A \vee \neg A \to \bot & (Hip) \\ 4. & A \vee \neg A \to \bot, A \vdash \bot & (\to E)\ 2,3 \\ 5. & A \vee \neg A \to \bot \vdash A \to \bot & (\to I)\ 4 \\ 6. & A \vee \neg A \to \bot \vdash A \vee \neg A & (\vee I)\ 5 \\ 7. & A \vee \neg A \to \bot \vdash \bot & (\to E)\ 3,6 \end{array}$$

- $\vdash_m \neg(A \vee B) \leftrightarrow \neg A \wedge \neg B$. Hay que mostrar ambas implicaciones:

- $\vdash_m \neg(A \vee B) \to \neg A \wedge \neg B$. Basta mostrar $A \vee B \to \bot, A \vdash_m \bot$ y $A \vee B \to \bot, B \vdash_m \bot$.

1.	$A \vee B \to \bot, A \vdash A$	(Hip)
2.	$A \vee B \to \bot, A \vdash A \vee B$	$(\vee I)$ 1
3.	$A \vee B \to \bot, A \vdash A \vee B \to \bot$	(Hip)
4.	$A \vee B \to \bot, A \vdash \bot$	$(\to E)$ 2, 3

  la derivación faltante es análoga.
- $\vdash_m \neg A \wedge \neg B \to \neg(A \vee B)$. Basta mostrar $\neg A \wedge \neg B, A \vee B \vdash \bot$.

1.	$\neg A \wedge \neg B, A \vee B \vdash A \vee B$	(Hip)
2.	$\neg A \wedge \neg B, A \vee B, A \vdash \neg A \wedge \neg B$	(Hip)
3.	$\neg A \wedge \neg B, A \vee B, A \vdash A$	(Hip)
4.	$\neg A \wedge \neg B, A \vee B, A \vdash \neg A$	$(\wedge E)$ 2
5.	$\neg A \wedge \neg B, A \vee B, A \vdash \bot$	$(\to E)$ 3, 4
6.	$\neg A \wedge \neg B, A \vee B, B \vdash \neg A \wedge \neg B$	(Hip)
7.	$\neg A \wedge \neg B, A \vee B, B \vdash B$	(Hip)
8.	$\neg A \wedge \neg B, A \vee B, B \vdash \neg B$	$(\wedge E)$ 6
9.	$\neg A \wedge \neg B, A \vee B, B \vdash \bot$	$(\to E)$ 7, 8
10.	$\neg A \wedge \neg B, A \vee B \vdash \bot$	$(\vee E)$ 1, 5, 9

- $\vdash \neg A \vee B \to A \to B$. Basta mostrar $\neg A \vee B, A \vdash B$

1.	$\neg A \vee B, A \vdash \neg A \vee B$	(Hip)
2.	$\neg A \vee B, A, \neg A \vdash A$	(Hip)
3.	$\neg A \vee B, A, \neg A \vdash \neg A$	(Hip)
4.	$\neg A \vee B, A, \neg A \vdash \bot$	$(\to E)$ 2, 3
5.	$\neg A \vee B, A, \neg A \vdash B$	$(\bot E)$ 4
6.	$\neg A \vee B, A, B \vdash B$	(Hip)
7.	$\neg A \vee B, A \vdash B$	$(\vee E)$ 1, 5, 6

- $\vdash A \vee \neg A \to \neg\neg A \to A$. Basta mostrar $A \vee \neg A, \neg\neg A \vdash A$.

1.	$A \vee \neg A, \neg\neg A \vdash A \vee \neg A$	(Hip)
2.	$A \vee \neg A, \neg\neg A, A \vdash A$	(Hip)
3.	$A \vee \neg A, \neg\neg A, \neg A \vdash \neg A$	(Hip)
4.	$A \vee \neg A, \neg\neg A, \neg A \vdash \neg\neg A$	(Hip)
5.	$A \vee \neg A, \neg\neg A, \neg A \vdash \bot$	$(\to E)$ 3, 4
6.	$A \vee \neg A, \neg\neg A, \neg A \vdash A$	$(\bot E)$ 5
7.	$A \vee \neg A, \neg\neg A \vdash A$	$(\vee E)$ 1, 2, 6

- $\vdash_c \neg\neg A \leftrightarrow A$. La parte $\vdash A \to \neg\neg A$ ya fue probada. Basta probar entonces $\vdash \neg\neg A \to A$, es decir, $\neg\neg A \vdash A$. Pero esto es inmediato por la regla $(\neg C)$
- $\vdash_c \neg(A \wedge B) \leftrightarrow \neg A \vee \neg B$. La parte "$\leftarrow$" es válida minimalmente. Basta probar $\neg(A \wedge B) \vdash_c \neg A \vee \neg B$.

1.	$\neg(A \wedge B) \vdash A \vee \neg A$	(TE)
2.	$\neg(A \wedge B), A \vdash B \vee \neg B$	(TE)
3.	$\neg(A \wedge B), A, B \vdash A$	(Hip)
4.	$\neg(A \wedge B), A, B \vdash B$	(Hip)
5.	$\neg(A \wedge B), A, B \vdash A \wedge B$	$(\wedge I)\ 3,4$
6.	$\neg(A \wedge B), A, B \vdash \neg(A \wedge B)$	(Hip)
7.	$\neg(A \wedge B), A, B \vdash \bot$	$(\to E)\ 5,6$
8.	$\neg(A \wedge B), A, B \vdash \neg A \vee \neg B$	$(\bot E)\ 7$
9.	$\neg(A \wedge B), A, \neg B \vdash \neg B$	(Hip)
10.	$\neg(A \wedge B), A, \neg B \vdash \neg A \vee \neg B$	$(\vee I)\ 9$
11.	$\neg(A \wedge B), A \vdash \neg A \vee \neg B$	$(\vee E)\ 2,8,10$
12.	$\neg(A \wedge B), \neg A \vdash \neg A$	(Hip)
13.	$\neg(A \wedge B), \neg A \vdash \neg A \vee \neg B$	$(\vee I)\ 12$
14.	$\neg(A \wedge B) \vdash \neg A \vee \neg B$	$(\vee E)\ 1,11,13$

- $\vdash_c ((A \to B) \to A) \to A$. Por la ley de contrapositiva basta mostrar $\neg A \vdash_c \neg((A \to B) \to A)$. Por otra parte, por un inciso anterior se tiene $A \wedge \neg B \vdash_m \neg(A \to B)$, por lo que basta mostrar $\neg A \vdash_c (A \to B) \wedge \neg A$, lo cual se sigue de $\neg A, A \vdash_c B$. Pero esto es inmediato de la regla $(\bot E)$.

□

En la proposición anterior, en los casos de $\vdash$ y $\vdash_c$ debe entenderse que el sistema correspondiente es estrictamente necesario para las derivaciones en $\vdash$ $(\vdash_c)$, es decir, no existe una derivación en $\vdash_m$ $(\vdash)$, aunque para mostrar formalmente estas afirmaciones se necesitan técnicas semánticas como las presentadas en la sección 2.7.

Nos ocupamos ahora de la semántica en sus versiones clásica e intuicionista.

2.4. Semántica Clásica. La semántica clásica es bien conocida y la recordamos brevemente.

Definición 2.9. Un *estado* de las variables proposicionales es una función $v : \mathsf{VProp} \to \mathsf{Bool}$ donde $\mathsf{Bool} = \{0, 1\}$ es el tipo de los valores booleanos verdadero 1 y falso 0.

Definición 2.10. Sea $v : \mathsf{VProp} \to \mathsf{Bool}$ un estado de las variables proposicionales. Dada una fórmula A definimos el *valor de verdad* de A en el estado v, denotado $v^\star(A)$, recursivamente como sigue:

$$
\begin{aligned}
v^\star(p) &= v(p) \\
v^\star(\bot) &= 0 \\
v^\star(A \vee B) &= v^\star(A) \ \mathsf{or} \ v^\star(B) \\
v^\star(A \wedge B) &= v^\star(A) \ \mathsf{and} \ v^\star(B) \\
v^\star(A \to B) &= (\mathsf{not}\, v^\star(A)) \ \mathsf{or} \ v^\star(B)
\end{aligned}
$$

donde $\mathsf{not} : \mathsf{Bool} \to \mathsf{Bool}$, or , and $: \mathsf{Bool} \times \mathsf{Bool} \to \mathsf{Bool}$ son los operadores booleanos usuales. Dado que el valor de verdad de cualquier fórmula queda determinado por el estado v es común usar simplemente v y no $v^\star$. Como es usual, si $\Gamma \subseteq \mathsf{FProp}$ entonces $v(\Gamma) = b$ significa $v(A) = b$ para toda $A \in \Gamma$.

Proposición 2.11. *La semántica recien definida recupera la lógica clásica al cumplirse* $\neg\neg A \equiv A$.

Demostración. Recordemos que $\neg A =_{def} A \to \bot$ por lo que se cumple

$$v(\neg A) = v(A \to \bot) = (\mathsf{not}\, v(A))\ \mathsf{or}\ v(\bot) = (\mathsf{not}\, v(A))\ \mathsf{or}\ 0 = \mathsf{not}\, v(A)$$

de donde $v(\neg\neg A) = \mathsf{not}\,\mathsf{not}\, v(A) = v(A)$. □

Definición 2.12. Decimos que A es *consecuencia lógica* de Γ, denotado $\Gamma \models A$ si y sólo si $v(\Gamma) = 1$ implica $v(A)$=1

Para terminar la discusión de la semántica clásica enunciamos el conocido teorema de completud-correctud para la lógica proposicional clásica.

Teorema 2.13. $\Gamma \vdash_c A$ *si y sólo si* $\Gamma \models A$.

2.5. Semántica Intuicionista. La semántica intuicionista de la lógica proposicional se basa en la semántica intuitiva conocida como interpretación de Brouwer-Heyting-Kolmogorov (BHK) en la que la noción de verdad no depende de algún valor predeterminado de las fórmulas sino en la existencia de una prueba o construcción de ellas. Es decir, el significado de una fórmula A consiste en explicar qué es una prueba de A.

Definición 2.14. La *interpretación BHK* se define recursivamente como sigue:

- Una prueba de una variable proposicional p se supone conocida y dada por un contexto previamente definido.
- No existe una prueba de $\bot$.
- Una prueba de $A \wedge B$ es una par $\langle p, q\rangle$ donde p es una prueba de A y q es una prueba de B.
- Una prueba de $A \vee B$ es un par $\langle 1, p\rangle$ donde p es una prueba de A o bien un par $\langle 2, q\rangle$ donde q es una prueba de B.
- Una prueba de $A \to B$ es un método funcional q que transforma cualquier prueba p de A en una prueba $q(p)$ de B.

De la definición anterior concluimos la siguiente interpretación para la negación:

- Una prueba de $\neg A$ consiste en un método que convierte cualquier supuesta prueba de A en un objeto inexistente.

Obsérvese que esta interpretación de la negación es más fuerte que pedir sólamente que no exista una construcción para A. Aunque la interpretación BHK definida aquí es mas bien una definición metamatemática puede formalizarse por ejemplo mediante interpretaciones de realización, para una exposición a fondo véase [18].

El carácter constructivo de la lógica intuicionista dado por la interpretación BHK la restringe de una manera importante, en particular el sistema no permite derivar el principio del tercero excluido, es decir el secuente $\vdash A \vee \neg A$, que es una tautología clásica. Para convencernos de tal situación basta recordar qué significa el hecho de que una disyunción sea demostrable. En el caso del tercero excluido tendríamos que construir ya sea una prueba de A o bien una prueba de $\neg A$ lo cual no es posible en general. Este hecho implica igualmente que la fórmula $\neg\neg A \to A$ no es válida. Por otro lado es fácil darle una interpretación constructiva a la fórmula $A \to \neg\neg A$ como sigue: sea p una prueba de A, tenemos que construir un método funcional que transforme a p en una prueba $q(p)$ de $(A \to \bot) \to \bot$, es decir $q(p)$ debe ser un método que transforme cada prueba r de $A \to \bot$ en una prueba de $\bot$. En tal caso dada p, $r(p)$ sería una prueba de $\bot$, por lo que podemos definir $q(p) = r$.

La semántica intuitiva anterior se puede formalizar de diversas maneras, siendo las más usadas la semántica de álgebras de Heyting y la semántica de mundos posibles de Kripke, ambas equivalentes. En este artículo optamos por la segunda.

Definición 2.15 (Modelo de Kripke)**.** Un *modelo de Kripke* es una terna

$$\mathcal{W} = \langle W, \leq, \Vdash \rangle$$

donde W es un conjunto no vacío cuyos elementos se llaman mundos posibles, $\leq$ es un orden parcial en W y $\Vdash \subseteq W \times \mathsf{VProp}$ es una relación binaria, llamada relación de forzamiento, que satisface la siguiente condición de monotonía:

$$\text{Si } w \leq w' \text{ y } w \Vdash p \text{ entonces } w' \Vdash p$$

La idea es que $w \Vdash p$ significa que p es cierta en el mundo w en cuyo caso la propiedad monotonía garantiza que p seguirá siendo cierta en cualquier mundo con igual o mayor información que w, situación dictada por la relación $\leq$.

Definición 2.16. La relación de *forzamiento* $\Vdash$ en un modelo $\mathcal{W}$ dado se extiende a todas las fórmulas recursivamente como sigue:

- $w \Vdash \bot$ nunca sucede
- $w \Vdash A \vee B$ si y sólo si $w \Vdash A$ ó $w \Vdash B$
- $w \Vdash A \wedge B$ si y sólo si $w \Vdash A$ y $w \Vdash B$
- $w \Vdash A \to B$ si y sólo si $w' \Vdash B$ para todo w' tal que $w \leq w'$ y $w' \Vdash A$

La notación $\mathcal{W} \Vdash A$ significa que $w \Vdash A$ para toda $w \in W$. Mientras que si Γ es un conjunto de fórmulas, $\mathcal{W} \Vdash \Gamma$ significa que para toda $A \in \Gamma$, $\mathcal{W} \Vdash A$.

Las siguientes propiedades de la negación son consecuencia de la definición anterior:

Proposición 2.17. *Sean $w \in W$ y $A \in \mathsf{FProp}$. Entonces*

- *$w \Vdash \neg A$ si y sólo si para toda $w' \geq w$, $w' \nVdash A$.*
- *$w \Vdash \neg\neg A$ si y sólo si para toda $w' \geq w$ no es cierto que para toda $w'' \geq w'$, $w'' \nVdash A$.*
- *Si para toda $w' \geq w$, existe $w'' \geq w'$ tal que $w'' \Vdash A$ entonces $w \Vdash \neg\neg A$.*

Demostración.

- Tenemos $w \Vdash \neg A$ si y sólo si $w \Vdash A \to \bot$ si y sólo si para toda $w' \geq w$, $w' \Vdash A$ implica $w' \Vdash \bot$, pero como $w' \Vdash \bot$ es imposible entonces necesariamente para toda $w' \geq w$, $w' \nVdash A$.
- Por la parte anterior tenemos $w \Vdash \neg\neg A$ si y sólo si para toda $w' \geq w$, $w' \nVdash \neg A$ si y sólo si para toda $w' \geq w$ no es cierto que para toda $w'' \geq w'$, $w'' \nVdash A$.

- El antecedente de esta implicación implica a la parte derecha de la equivalencia enunciada en la parte anterior por lo cual implica también que $w \Vdash \neg\neg A$.

$\square$

Observemos que la equivalencia entre $w \Vdash \neg\neg A$ y la condición para toda $w' \geq w$, existe $w'' \geq w'$ tal que $w'' \Vdash A$ requiere un razonamiento clásico en la metateoría puesto que la fórmula $\neg\forall x\neg A \to \exists x A$ no es válida en la lógica de primer orden intuicionista.

Proposición 2.18 (Monotonía de $\Vdash$)**.** *Sean $\mathcal{W}$ un modelo y $A \in \mathsf{FProp}$. Para cualesquiera $w, w' \in W$, si $w \leq w'$ y $w \Vdash A$ entonces $w' \Vdash A$.*

Demostración. Inducción sobre A. $\square$

Definición 2.19. Sean Γ un conjunto de fórmulas y A una fórmula, decimos que Γ *fuerza* a A, denotado $\Gamma \Vdash A$ si y sólo si para cualquier modelo de Kripke $\mathcal{W}$ la condición $\mathcal{W} \Vdash \Gamma$ implica $\mathcal{W} \Vdash A$.

La semántica de modelos de Kripke se relaciona con la relación de derivabilidad intuicionista mediante el siguiente

Teorema 2.20 (Completud-Correctud)**.** *$\Gamma \vdash A$ si y sólo si $\Gamma \Vdash A$.*

Demostración. La parte $\Rightarrow$) se conoce como teorema de correctud y se prueba mediante inducción sobre $\Gamma \vdash A$. El recíproco $\Leftarrow$) es el llamado teorema de completud de Kripke y se prueba mediante una construcción similar a la de Henkin para la completud clásica, véase [17]. $\square$

2.6. Algunas propiedades relevantes de la lógica intuicionista. La lógica intuicionista cumple diversas propiedades inválidas en la lógica clásica. Las más relevantes para nuestra exposición son:

- Propiedad disyuntiva: si $\vdash A \vee B$ entonces $\vdash A$ ó $\vdash B$.
  Esta propiedad no es válida en la lógica clásica y es una de las razones por las que la lógica intuicionista también se conoce como lógica constructiva. Una prueba mediante modelos de Kripke puede consultarse en [1].
- Infinitud: la lógica intuicionista no es una lógica finitamente valuada.
  En consecuencia no es posible darle una semántica de valuaciones, similar a la clásica, utilizando un número finito de valores de verdad. Esto causa que la lógica proposicional de segundo orden, en su versión intuicionista tenga interés, a diferencia de la versión clásica cuya semántica es trivial como veremos más adelante. La prueba de la infinitud requiere álgebras de Heyting y puede consultarse en [17].

Enseguida desarrollamos más ejemplos de fórmulas inválidas en la lógica intuicionista para lo cual nos serviremos de la semántica de mundos posibles.

2.7. Invalidez de algunas tautologías clásicas. Una de las principales aplicaciones de la semántica de Kripke es mostrar resultados negativos acerca de la relación $\vdash$ usando el teorema de completud-correctud. Los siguientes ejemplos pretenden ser una invitación a la búsqueda de tautologías clásicas que no lo son desde el punto de vista constructivo.

Proposición 2.21. *Si $p \in \mathsf{VProp}$ entonces $\nvdash \neg\neg p \vee \neg p$.*

Demostración. Debido al teorema de completud (2.20) basta mostrar que $\nVdash \neg\neg p \vee \neg p$. Sea $\mathcal{W} = \langle\{0,1,2\}, \leq, \Vdash\rangle$ con $0 \leq 1, 0 \leq 2$ y $1 \Vdash p$. $\mathcal{W}$ es un modelo de Kripke pues la condición de monotonía se cumple al ser 1 un elemento maximal. Afirmamos que $0 \nVdash \neg\neg p \vee \neg p$. Primero obsérvese que $2 \Vdash \neg p$ puesto que $2 \nVdash p$ y 2 es maximal. Ahora bien, como $2 \geq 0$ y $2 \Vdash \neg p$ entonces $0 \nVdash \neg\neg p$. Por otra parte como $1 \geq 0$ y $1 \Vdash p$ entonces $0 \nVdash \neg p$. Finalmente como $0 \nVdash \neg\neg p$ y $0 \nVdash \neg p$ entonces $0 \nVdash \neg\neg p \vee \neg p$ lo cual implica que $\mathcal{W} \nVdash \neg\neg p \vee \neg p$ y por lo tanto $\nVdash \neg\neg p \vee \neg p$. $\square$

Proposición 2.22. *Si* $p, q \in \mathsf{VProp}$ *entonces* $\nvdash (p \to q) \vee (q \to p)$.

Demostración. Sea $\mathcal{W} = \langle\{0,1,2\}, \leq, \Vdash\rangle$ con $0 \leq 1,\ 0 \leq 2$ y $1 \Vdash p,\ 2 \Vdash q$. $\mathcal{W}$ es un modelo de Kripke pues 1 y 2 son maximales. Afirmamos que $0 \nVdash (p \to q) \vee (q \to p)$ para lo cual basta ver que:

- $0 \nVdash p \to q$.
  Sabemos que $0 \Vdash p \to q$ si y sólo si para toda $w \geq 0,\ w \Vdash p$ implica $w \Vdash q$, pero tenemos $1 \geq 0, 1 \Vdash p$ y $1 \nVdash q$.
- $0 \nVdash q \to p$.
  Análogamente se tiene $2 \geq 0, 2 \Vdash q$ y $2 \nVdash p$.

Por lo tanto $\mathcal{W} \nVdash (p \to q) \vee (q \to p)$ de donde $\nVdash (p \to q) \vee (q \to p)$ y por completud $\nvdash (p \to q) \vee (q \to p)$ $\square$

Proposición 2.23. *Si* $p \in \mathsf{VProp}$ *entonces* $\nvdash (\neg\neg p \to p) \vee \neg p \vee \neg\neg p$.

Demostración. Sea $\mathcal{W} = \langle\{0,1,2,3,4\}, \leq, \Vdash\rangle$ con $0 \leq 1 \leq 2,\ 1 \leq 3,\ 0 \leq 4 \leq 3,\ 3 \Vdash p$. Veamos que $\mathcal{W}$ no fuerza a ninguno de los tres disyuntos:

- $0 \nVdash \neg\neg p$.
  Tenemos que $0 \Vdash \neg\neg p$ si y sólo si para todo $w \geq 0,\ w \nVdash \neg p$ pero $2 \geq 0$ y $2 \Vdash \neg p$ puesto que $2 \nVdash p$ y 2 es maximal. Por lo tanto $0 \nVdash \neg\neg p$.
- $0 \nVdash \neg p$. Por definición $0 \Vdash \neg p$ si y sólo si para toda $w \geq 0,\ w \nVdash p$ pero $3 \geq 0$ y $3 \Vdash p$. por lo tanto $0 \nVdash \neg p$.
- $0 \nVdash \neg\neg p \to p$. Por definición $0 \Vdash \neg\neg p \to p$ si y sólo si para toda $w \geq 0, w \Vdash \neg\neg p$ implica $w \Vdash p$. Veamos que $4 \Vdash \neg\neg p$ con lo cual habremos terminado pues $4 \geq 0$ y $4 \nVdash p$.
  Necesitamos ver que para toda $w \geq 4,\ w \nVdash \neg p$. Como $3 \Vdash p$ y 3 es maximal entonces $3 \nVdash \neg p$. Por otra parte como $3 \geq 4$ y $3 \Vdash p$ entonces $4 \nVdash \neg p$. Hemos probado que para toda $w \geq 4,\ w \nVdash \neg p$. Por lo tanto $4 \Vdash \neg\neg p$.

$\square$

Obsérvese que de la prueba anterior también podemos concluir la invalidez de la ley de doble negación $\nvdash \neg\neg p \to p$.

Usando la relación de forzamiento $\Vdash$ también es posible mostrar que ninguno de los conectivos $\wedge, \vee, \to, \bot$ es definible a partir de una combinación de los otros por lo que, a diferencia de la lógica clásica, la lógica intuicionista debe tener presentes a todos los conectivos si se desea contar con todo su poder expresivo. Véase [17].

Ahora ya contamos con todos los antecedentes necesarios para introducir nuestro principal objeto de estudio, la lógica proposicional de segundo orden.

3. LA LÓGICA Prop2

La lógica proposicional de segundo orden, denotada Prop2 se obtiene al agregar a la sintaxis de la lógica usual de proposiciones los cuantificadores $\forall, \exists$. Esto tiene diversas ventajas, por ejemplo, considérese la siguiente propiedad de la lógica proposicional

Proposición 3.1. *Si* $\vdash A$ *entonces para cualquier variable* p *y cualquier fórmula* B *se cumple* $\vdash A[p := B]$.

Demostración. Es claro por la regla de sustitución $(Sust)$ dada en la proposición (2.7). □

Este resultado permite concluir en particular que si A es cualquier fórmula entonces $\vdash A \to A$. Es decir, no importando el valor ni la forma de A la fórmula $A \to A$ es un teorema/tautología de la lógica. Esta propiedad se sirve de una cuantificación informal inexpresable en la lógica proposicional, la lógica es incapaz de agrupar formalmente a todas las instancias de $p \to p$ para después poder concluir cada caso particular. Esto se logra fácilmente en Prop2 mediante la fórmula $\forall p(p \to p)$. Similarmente dada una fórmula A sabemos que existe al menos una proposición p tal que $A \leftrightarrow p$, a saber A misma, esto puede expresarse con la fórmula $\exists p(A \leftrightarrow p)$.

Pasamos ahora a las definiciones formales.

3.1. Sintaxis de Prop2.

A continuación presentamos los pormenores de la sintaxis de Prop2

Definición 3.2. Sea $\mathsf{VProp} = \{p, q, r, \ldots\}$ un conjunto infinito numerable de variables proposicionales. El conjunto de *fórmulas* de la lógica proposicional de segundo orden Prop2, es el mínimo conjunto FProp obtenido al extender la definición (2.1) como sigue:

- Si $p \in \mathsf{VProp}$ y $A \in \mathsf{FProp}$ entonces $(\forall pA) \in \mathsf{FProp}$.
- Si $p \in \mathsf{VProp}$ y $A \in \mathsf{FProp}$ entonces $(\exists pA) \in \mathsf{FProp}$.

En adelante adoptamos algunas veces la notación de punto en un cuantificador para declarar que su alcance se considera lo más lejos posible sintácticamente, por ejemplo la proposición $\forall p.A \to B$ significa $\forall p(A \to B)$ y no $(\forall pA) \to B$.
Al igual que en la lógica de predicados de primer orden, los cuantificadores introducen un ligado de la variable cuantificada en el alcance de la cuantificación, lo cual causa una distinción necesaria entre las presencias libres y ligadas de una variable en una proposición, noción que precisamos enseguida.

Definición 3.3. El conjunto de *variables libres* de la fórmula A, denotado $FV(A)$ se define recursivamente como sigue:

- $FV(p) = \{p\}$
- $FV(\bot) = \varnothing$
- $FV(A \star B) = FV(A) \cup FV(B)$, donde $\star \in \{\wedge, \vee, \to\}$
- $FV(\nabla pA) = FV(A) - \{p\}$, donde $\nabla \in \{\forall, \exists\}$

Si una presencia de una variable q figura en A pero $q \notin FV(A)$, entonces decimos que tal presencia de q está ligada en A. Si $\Gamma = \{A_1, \ldots, A_n\}$ entonces definimos $FV(\Gamma) = FV(A_1) \cup \ldots \cup FV(A_n)$.

La noción de sustitución cobra más importancia y se vuelve más complicada pues en este caso no se trata de una sustitución textual.

Definición 3.4. La noción de *sustitución* $A[p := B]$ se define al extender la definición (2.2) para fórmulas cuantificadas como sigue:

- $(\forall qA)[p := B] = \forall q.A[p := B]$ siempre y cuando $q \notin \{p\} \cup FV(B)$
- $(\exists qA)[p := B] = \exists q.A[p := B]$ siempre y cuando $q \notin \{p\} \cup FV(B)$

La condición en esta definición siempre puede cumplirse al renombrar la variable ligada q en la fórmula original. Esto no causa problema alguno pues consideramos iguales a dos fórmulas que difieren únicamente en los nombres de sus variables ligadas. Esta convención se conoce como α-equivalencia, por ejemplo $\forall pp$ y $\forall qq$ son fórmulas α-equivalentes. Obsérvese que si queremos que la operación de sustitución defina a una función total, en lugar de fórmulas, es necesario manipular clases de α-equivalencia de fórmulas, situación que no hacemos explícita.

Una vez definida formalmente la noción de sustitución podemos enunciar las reglas de deducción natural para las fórmulas cuantificadas, extendiendo así la definición (2.5):

- Cuantificación universal:

$$\frac{\Gamma \vdash A \qquad p \notin FV(\Gamma)}{\Gamma \vdash \forall pA}\ (\forall I) \qquad\qquad \frac{\Gamma \vdash \forall pA}{\Gamma \vdash A[p := B]}\ (\forall E)$$

- Cuantificación existencial:

$$\frac{\Gamma \vdash A[p := B]}{\Gamma \vdash \exists pA}\ (\exists I) \qquad\qquad \frac{\Gamma \vdash \exists pA \qquad \Gamma, A \vdash B \qquad p \notin FV(\Gamma, B)}{\Gamma \vdash B}\ (\exists E)$$

A continuación nos ocupamos de la semántica.

4. SEMÁNTICA DE Prop2

El significado intencional de la fórmula $\forall pA$ es que $A(p)$ es válida para cualquier valor posible de p. Similarmente, si $A(p)$ es válida para algún valor de p podemos decir que $\exists pA$ es válida. En esta sección veremos primero que la semántica clásica resulta trivial por lo que la semántica relevante es la intuicionista, desarrollada posteriormente.

4.1. Semántica Clásica. Dado que en la semántica clásica existen sólo dos valores de verdad, el significado de los cuantificadores resulta equivalente a una conjunción o disyunción de dichos valores. A continuación mostramos esto formalmente concluyendo un interés nulo por esta semántica.

Definición 4.1. Dados un estado de las variables v y un valor booleano $b : \mathsf{Bool}$, definimos el *estado actualizado* en p con b, denotado $v[p/b]$ como

$$v[p/b](q) = \begin{cases} v(q) & \text{si } p \neq q \\ \\ b & \text{si } p = q \end{cases}$$

Definición 4.2. Sea $v : \mathsf{VProp} \to \mathsf{Bool}$ un estado de las variables proposicionales. Dada una fórmula A definimos el *valor de verdad* de A en el estado v, denotado $v^\star(A)$, extendiendo la definición (2.10) como sigue:

$$\begin{array}{rcl} v^\star(\forall pA) & = & v[p/0]^\star(A) \text{ and } v[p/1]^\star(A) \\ \\ v^\star(\exists pA) & = & v[p/0]^\star(A) \text{ or } v[p/1]^\star(A) \end{array}$$

Lema 4.3 (Sustitución)**.** $v(A[p := B]) = v[p/v(B)](A)$.

Demostración. Inducción sobre A. □

La siguiente proposición muestra la trivialidad de la semántica clásica para $\mathsf{Prop2}$.

Proposición 4.4. *Sea* $\top =_{def} \bot \to \bot$. *Entonces* $\forall pA \equiv A[p := \bot] \wedge A[p := \top]$ *y* $\exists pA \equiv A[p := \bot] \vee A[p := \top]$. *Por lo tanto ambos cuantificadores son definibles.*

Demostración.

$$\begin{array}{rcl} v(\forall pA) & = & v[p/0](A) \text{ and } v[p/1](A) \\ & = & v[p/v(\bot)](A) \text{ and } v[p/v(\top)](A) \\ & = & v(A[p := \bot]) \text{ and } v(A[p := \top]) \quad \text{(por lema (4.3))} \\ & = & v(A[p := \bot] \wedge A[p := \top]) \end{array}$$

$$\begin{array}{rcl} v(\exists pA) & = & v[p/0](A) \text{ or } v[p/1](A) \\ & = & v[p/v(\bot)](A) \text{ or } v[p/v(\top)](A) \\ & = & v(A[p := \bot]) \text{ or } v(A[p := \top]) \quad \text{(por lema (4.3))} \\ & = & v(A[p := \bot] \vee A[p := \top]) \end{array}$$

□

De la proposición anterior se sigue que la lógica proposicional de segundo orden clásica no tiene mayor poder expresivo que la lógica proposicional usual, por lo que resulta de poco interés. Esto es de esperarse pues la lógica proposicional clásica es funcionalmente completa, es decir, cualquier función booleana puede definirse con los conectivos ordinarios, por lo que ninguna extensión a la sintaxis causa un mayor poder expresivo.

A continuación definimos a detalle la semántica intuicionista, usando nuevamente modelos de Kripke.

4.2. Semántica Intuicionista. La propiedad de infinitud de la lógica intuicionista implica que en la semántica intuicionista no hay un conjunto finito de valores de verdad por lo que los cuantificadores varían sobre un espacio infinito de proposiciones. De hecho no hay un resultado de completud funcional. En este caso el significado intencional se explica nuevamente mediante la interpretación BHK.

Definición 4.5. La *interpretación BHK* para las fórmulas proposicionales cuantificadas se define como sigue:

- Una prueba de $\forall pA$ es un método funcional que transforma cualquier prueba de una fórmula arbitraria B en una prueba de $A[p := B]$.
- Una prueba de $\exists pA$ es un par $\langle B, q\rangle$ donde B es una proposición y q es una prueba de $A[p := B]$. La proposición B se conoce como testigo.

Formalizamos la interpretación BHK suponiendo que los cuantificadores varían sobre la clase de todas las proposiciones. Esta hipótesis se conoce como el principio de comprensión total y debemos mencionar que no es aceptada en todos los sistemas lógicos intuicionistas. Dicho principio se expresa formalmente mediante la validez para cualesquiera fórmulas A y B, de la proposición $\forall pA \to A[p := B]$ o equivalentemente de la fórmula $\exists p(B \leftrightarrow p)$ con $p \notin FV(B)$. La adopción del principio de comprensión total causa que la semántica intuicionista sea impredicativa: en la definición de $\forall pA$ la variable p representa a todas las proposiciones posibles, en particular a la misma $\forall pA$. Es decir, el significado de $\forall pA$ depende de todas las instancias $A[p := B]$ incluyendo los casos donde B es de igual o mayor complejidad que $\forall pA$. Por ejemplo para entender $\forall p.p \to p$ es necesario entender $(p \to p)[p := \forall p.p \to p] = (\forall p.p \to p) \to (\forall p.p \to p)$ que claramente es más compleja. En particular no existe una jerarquía bien fundada en la semántica y muchos métodos por inducción fallarán inevitablemente.

Para definir una semántica formal que capture a la interpretación BHK, nos servimos nuevamente de los modelos de Kripke.

Definición 4.6. Un *modelo de Kripke* de segundo orden es una terna

$$\mathcal{W} = \langle W, \leq, \mathcal{D}\rangle$$

donde

- $W \neq \varnothing$.
- $\leq \subseteq W \times W$ es un orden parcial sobre W
- $\mathcal{D} = \{D_w \mid w \in W\}$ y para cada $w \in W,\ D_w \subseteq \mathcal{P}(W)$ es una familia de subconjuntos cónicos de W.
- Un conjunto $U \subseteq W$ es *cónico* o cerrado hacia arriba, si para toda $u \in U$ y para toda $v \geq u$ se tiene $v \in U$.
- Si $w \leq w'$ entonces $D_w \subseteq D_{w'}$

A continuación definimos el concepto de valuación.

Definición 4.7. Una *valuación* v en un modelo de Kripke de segundo orden $\mathcal{W}$ es una función que asigna a cada variable proposicional p un subconjunto cónico $v(p)$ de W. Decimos que v es admisible para un mundo w y una fórmula A si y sólo si $v(p) \in D_w$ para toda $p \in FV(A)$. La valuación actualizada en p, denotada $v[p/x]$, se define análogamente a la definición (4.1), siendo x en este caso un conjunto cónico.

Es claro que si v es admisible para w y A entonces también es admisible para w' y A para cualquier $w' \geq w$.

Ahora ya podemos definir la relación $\Vdash$.

Definición 4.8. La relación de *forzamiento* $w, v \Vdash A$ entre un mundo w, una valuación v y una fórmula A se define cuando v es admisible para w y A recursivamente como sigue:

- $w, v \Vdash p$ si y sólo si $w \in v(p)$
- $w, v \Vdash \bot$ nunca sucede
- $w, v \Vdash A \vee B$ si y sólo si $w, v \Vdash A$ ó $w, v \Vdash B$
- $w, v \Vdash A \wedge B$ si y sólo si $w, v \Vdash A$ y $w, v \Vdash B$
- $w, v \Vdash A \to B$ si y sólo si $w', v \Vdash B$ para todo w' tal que $w \leq w'$ y $w', v \Vdash A$
- $w, v \Vdash \forall pA$ si y sólo si $w', v[p/x] \Vdash A$ para cualesquiera $w' \geq w$ y $x \in D_{w'}$
- $w, v \Vdash \exists pA$ si y sólo si $w, v[p/x] \Vdash A$ para alguna $x \in D_w$

Escribimos $w, v \Vdash \Gamma$ si $w, v \Vdash A$ para toda $A \in \Gamma$ y $\mathcal{W} \Vdash A$ si $w, v \Vdash A$ para toda $w \in \mathcal{W}$ y v admisible para w y A.

A continuación damos algunas propiedades y definiciones que involucran a la relación de forzamiento y que serán de utilidad más tarde.

Proposición 4.9 (Monotonía de $\Vdash$)**.** *Sean $w, w' \in W$, $A \in \mathsf{FProp}$ y v admisible para w y A. Si $w, v \Vdash A$ y $w' \geq w$ entonces $w', v \Vdash A$.*

Demostración. Inducción sobre A. □

Corolario 4.10. *Sean $A \in \mathsf{FProp}$ y v una valuación. El conjunto $\{w \in W \mid w, v \Vdash A\}$ es cónico.*

Lema 4.11 (Coincidencia)**.** *Sean v_1, v_2 admisibles para w y A. Si $v_1(p) = v_2(p)$ para toda $p \in FV(A)$ entonces $w, v_1 \Vdash A$ si y sólo si $w, v_2 \Vdash A$.*

Demostración. Inducción sobre A. □

Definición 4.12. Sean $A \in \mathsf{FProp}$, $w \in W$, y v admisible para w y A. Definimos

$$[\![A]\!]_{w,v} = \{w' \in W \mid w' \geq w \text{ y } w', v \Vdash A\}$$

Proposición 4.13. *$[\![A]\!]_{w,v}$ es cónico.*

Demostración. Inmediato □

Lema 4.14 (Sustitución)**.** *Sean $A, B \in \mathsf{FProp}$, $w \in W$ y v una valuación admisible para w y $A[p := B]$ tales que $[\![B]\!]_{w,v} \in D_w$. Entonces*

$$w, v \Vdash A[p := B] \text{ si y sólo si } w, v[p/[\![B]\!]_{w,v}] \Vdash A.$$

Demostración. Inducción sobre A. □

4.3. Modelos Completos. Al considerar un sistema de inferencia arbitrario $\vdash$ y una relación semántica $\models$ sobre el mismo conjunto de fórmulas de una lógica, lo ideal es obtener un teorema de correctud-completud que permita concluir la equivalencia de ambos conceptos. Si consideramos la semántica de Kripke para la lógica proposicional intuicionista $\Vdash$ y el sistema de deducción natural $\vdash$ este es el caso. Sin embargo para $\mathsf{Prop2}$ la semántica de Kripke es demasiado general por lo que debemos restringir la relación $\Vdash$ a una clase especial de modelos llamados completos, los cuales son indispensables para preservar la correctud de la lógica.

Definición 4.15. Un modelo de Kripke $\mathcal{W}$ es *completo* si y sólo si para cada fórmula A, $w \in W$ y v tales que v es admisible para w y A se cumple que $[\![A]\!]_{w,v} \in D_w$.

A continuación veremos que el uso de modelos completos es imprescindible para lograr un sistema lógico correcto.

Proposición 4.16. *Un modelo $\mathcal{W}$ es completo si y sólo si $\mathcal{W} \Vdash \exists p(p \leftrightarrow A)$ para cada A tal que $p \notin FV(A)$.*

Demostración. $\mathcal{W} \Vdash \exists p(p \leftrightarrow A)$ si y sólo si para todo $w \in W$ y v admisible para w, A se tiene $w, v \Vdash \exists p(p \leftrightarrow A)$ lo cual sucede si y sólo si existe $x \in D_w$ tal que $w, v[p/x] \Vdash p \leftrightarrow A$. Esta última condición es equivalente a que para toda $w' \geq w,\ w', v[p/x] \Vdash p$ si y sólo si $w', v[p/x] \Vdash A$, que, por definición de la relación de forzamiento y el lema de coincidencia (4.11) dado que $p \notin FV(A)$, resulta equivalente a $w' \in x$ si y sólo si $w', v \Vdash A$ para toda $w' \geq w$. Es decir, $x = [\![A]\!]_{w,v}$. Finalmente como $x \in D_w$ entonces $[\![A]\!]_{w,v} \in D_w$ lo cual es equivalente a que $\mathcal{W}$ es completo.

$\square$

No es complicado cerciorarse de que la completud de un modelo también equivale a la validez de la fórmula $\forall pA \to A[p := B]$ para cualquier $B \in \mathsf{FProp}$. Además debido a la proposición anterior la relación de forzamiento $\Gamma \Vdash A$ debe restringirse a la clase de modelos completos. Esto quedará más claro en la demostración del teorema de completud.

Definición 4.17. Sean Γ un conjunto de fórmulas y A una fórmula, decimos que Γ *fuerza* a A, denotado $\Gamma \Vdash A$ si para cualquier modelo completo $\mathcal{W}$ la condición $\mathcal{W}, v \Vdash \Gamma$ implica que $\mathcal{W}, v \Vdash A$ para cualquier valuación admisible para W y $\Gamma \cup \{A\}$

Teorema 4.18 (Completud-Correctud)**.** *$\Gamma \Vdash A$ si y sólo si $\Gamma \vdash A$.*

Demostración. La prueba de la completud $\Rightarrow$) puede consultarse en [17].
Para la correctud hacemos una inducción sobre $\vdash$. Veamos con detalle los casos que atestiguan la necesidad de la restricción a modelos completos.

- Caso $(\forall E)$. Sea $\Gamma \vdash A[p := B]$ obtenido de $\Gamma \vdash \forall pA$. Queremos demostrar que $\Gamma \Vdash A[p := B]$. Dado $\mathcal{W}$ un modelo completo tal que $\mathcal{W}, v \Vdash \Gamma$, debemos mostrar que $w, v \Vdash A[p := B]$ para toda $w \in W$ y v admisible. Por hipótesis de inducción, $\Gamma \Vdash \forall pA$ por lo que se tiene $w, v \Vdash \forall pA$, es decir, dados cualesquiera $w' \geq w$ y $x \in D_{w'}$ se tiene $w', v[p/x] \Vdash A$. En particular, como $\mathcal{W}$ es completo entonces $[\![B]\!]_{w,v} \in D_w$ y podemos concluir $w, v[p/[\![B]\!]_{w,v}] \Vdash A$. Finalmente el lema de sustitución (4.14) nos lleva a concluir que $w, v \Vdash A[p := B]$.
- Caso $(\exists I)$. Sea $\Gamma \vdash \exists pA$ derivado de $\Gamma \vdash A[p := B]$. Queremos mostrar que $\Gamma \Vdash \exists pA$. Sea $\mathcal{W}$ un modelo completo tal que $\mathcal{W}, v \Vdash \Gamma$, debemos mostrar que $w, v \Vdash \exists pA$ para toda $w \in W$ y v admisible. Por hipótesis de inducción, $\Gamma \Vdash A[p := B]$ por lo que $w, v \Vdash A[p := B]$, que, por el lema de sustitución (4.14), equivale a $w, v[p/[\![B]\!]_{w,v}] \Vdash A$ puesto que $[\![B]\!]_{w,v} \in D_w$ por la completud de $\mathcal{W}$, pero esto implica, por definición, que $w, v \Vdash \exists pA$.

$\square$

Para terminar nuestra exposición de la semántica intuicionista presentamos a detalle un ejemplo de invalidez.

4.4. Invalidez del Axioma de Gabbay. El siguiente esquema se conoce como axioma de Gabbay

$$\mathsf{Gab} =_{def} \forall p(A \vee B) \to A \vee \forall pB \;\; p \notin FV(A)$$

Es claro que Gab es válido clásicamente. Además es bien sabido que Gab es válido en la clase de modelos de Kripke donde todos los D_w son iguales, véase [3]. Aquí mostraremos que no lo es en el caso de modelos completos, para lo cual nos serviremos del siguiente modelo

$$\mathcal{W} = \langle \{1,2,3,4\}, \leq, \{D_1, D_2, D_3, D_4\} \rangle$$

donde

- $1 \leq 2 \leq 3,\; 2 \leq 4$
- $D_1 = \{\varnothing,\; \{3,4\},\; \{2,3,4\}, \{1,2,3,4\}\}$
- $D_2 = D_3 = D_4 = D_1 \cup \{\{3\},\{4\}\}$

Es fácil ver que $\mathcal{W}$ es un modelo de Kripke de segundo orden, en particular $D_2 = D_3 = D_4$ constan de todos lo subconjuntos cónicos de $\mathcal{W}$ lo cual se verifica por simple inspección. Considérese la función $\overline{(\cdot)} : W \to W$ tal que

$$\overline{1} = 1 \qquad \overline{2} = 2 \qquad \overline{3} = 4 \qquad \overline{4} = 3$$

Si $V \subseteq W$ entonces definimos $\overline{V} = \{\overline{w} \mid w \in V\}$ y si v es una valuación en W definimos la valuación $\overline{v}$ como $\overline{v}(w) = \overline{v(w)}$. Obsérvese además que $\overline{(\cdot)}$ es una involución, es decir, $\overline{\overline{w}} = w$ para toda $w \in W$.

Los siguientes lemas nos encaminan a nuestro objetivo final.

Lema 4.19. *$w, \overline{v} \Vdash A$ si y sólo si $\overline{w}, v \Vdash A$.*

Demostración. Inducción sobre A. □

Lema 4.20. *Si $V \subseteq W$ es cónico entonces $V \in D_1$ si y sólo si $\overline{V} = V$.*

Demostración. Inmediato por inspección directa. □

Corolario 4.21. *Si v es una valuación sobre D_1 entonces $\overline{v} = v$.*

Demostración. Por el lema anterior, $v(p) \in D_1$ para toda $p \in \mathsf{VProp}$ implica $\overline{v}(p) = \overline{v(p)} = v(p)$ □

Lema 4.22. $\overline{[\![A]\!]_{1,v}} = [\![A]\!]_{1,v}$.

Demostración.

$$\begin{array}{rcll} w \in [\![A]\!]_{1,v} & \Leftrightarrow & w \leq 1,\;\; w, v \Vdash A & \\ & \Leftrightarrow & w \leq 1,\;\; \overline{\overline{w}}, v \Vdash A & (w = \overline{\overline{w}}) \\ & \Leftrightarrow & w \leq 1,\;\; \overline{w}, \overline{v} \Vdash A & (\text{lema } (4.19)) \\ & \Leftrightarrow & w \leq 1,\;\; \overline{w}, v \Vdash A & (\text{corolario } (4.21)) \\ & \Leftrightarrow & \overline{w} \leq 1,\;\; \overline{w}, v \Vdash A & (w \leq 1 \Leftrightarrow \overline{w} \leq 1) \\ & \Leftrightarrow & \overline{w} \in [\![A]\!]_{1,v} & \\ & \Leftrightarrow & w \in \overline{[\![A]\!]_{1,v}} & \end{array}$$

v es sobre D_1 y el corolario (4.21) es aplicable pues v es admisible para 1 y A. □

Lema 4.23. *$\mathcal{W}$ es un modelo completo.*

Demostración. Sean $w \in W$, $A \in \mathsf{FProp}$, v admisible para w y A. Debemos mostrar que $[\![A]\!]_{w,v} \in D_w$. En los casos para $w = 2,3,4$ no hay nada que probar pues D_w consta de todos los cónicos de W y $[\![A]\!]_{w,v}$ es cónico. El caso importante es cuando $w = 1$ pero por los lemas (4.20) y (4.22) es claro que $[\![A]\!]_{1,v} \in D_1$. □

Lema 4.24. *Sea v una valuación en $\mathcal{W}$ tal que $v(q) = \{2,3,4\}$. Entonces*

$$1, v \Vdash \forall p(q \vee \neg p \vee \neg\neg p).$$

Demostración. Debemos probar que para todo $w' \geq 1$ y para cualquier $x \in D_{w'}$ se cumple $w', v[p/x] \Vdash q \vee \neg p \vee \neg\neg p$. Analicemos todos los casos:

- $w' = 1$ con $x \in D_1 = \{\varnothing, \{3,4\}, \{2,3,4\}, \{1,2,3,4\}\}$
  - $x = \varnothing$. En este caso tenemos $1, v[p/x] \Vdash \neg p$ pues para toda $u \geq 1$, $u \notin \varnothing = x$, es decir, $u, v[p/x] \nVdash p$.
  - $x = \{3,4\}$. En este caso se tiene que para toda $r \geq 1$ existe $u \geq r$ con $u \in \{3,4\}$ lo cual es suficiente para concluir que $1, v[p/x] \Vdash \neg\neg p$, usando la versión para $\mathsf{Prop2}$ de la proposición (2.17).
  - $x = \{2,3,4\}$ y $x = \{1,2,3,4\}$. Se prueba nuevamente $1, v[p/x] \Vdash \neg\neg p$ análogamente al caso anterior.
- $w' = 2$ con $x \in D_2 = \{\varnothing, \{3\}, \{4\}, \{3,4\}, \{2,3,4\}, \{1,2,3,4\}\}$
  - $x \in D_1$. Inmediato por monotonía.
  - $x = \{3\}$. En este caso se cumple $2, v[p/x] \Vdash q$ pues $2 \in \{2,3,4\} = v[p/x](q)$.
  - $x = \{4\}$. Análogo al caso anterior.
- $w' = 3$ y $w' = 4$. Como $D_3 = D_4 = D_2$ todos los casos son inmediatos por monotonía

Como en todos los casos se fuerza alguno de los disyuntos entonces se fuerza la disyunción que es la conclusión deseada. □

Lema 4.25. *Sea v una valuación en $\mathcal{W}$ tal que $v(q) = \{2,3,4\}$. Entonces*

$$1, v \nVdash q \vee \forall p(\neg p \vee \neg\neg p).$$

Demostración. Hay que mostrar que $1, v$ no fuerzan ni q ni $\forall p(\neg p \vee \neg\neg p)$. El primer caso es claro pues $1, v \Vdash q$ si y sólo si $1 \in v(q) = \{2,3,4\}$ lo cual claramente es falso. Para mostrar que $1, v \nVdash \forall p(\neg p \vee \neg\neg p)$ basta mostar un $w \geq 1$ y un $x \in D_w$ tales que $w, v[p/x] \nVdash \neg p \vee \neg\neg p$. Para tal efecto sean $w = 2$ y $x = \{3\} \in D_2$.

- $2, v[p/x] \nVdash \neg p$ pues $3 \geq 2$ y $3, v[p/x] \Vdash p$ dado que $3 \in \{3\} = v[p/x](p)$.
- $2, v[p/x] \nVdash \neg\neg p$ pues $4 \geq 2$ y para toda $u \geq 4$, $u, v[p/x] \nVdash p$ dado que necesariamente $u = 4$ y $4 \notin \{3\} = v[p/x](p)$.

□

Nuestro objetivo final es ahora alcanzable.

Proposición 4.26. *La tautología clásica* Gab *no es válida en la semántica intuicionista.*

Demostración. Basta dar un contraejemplo para A, B particulares. Elegimos $A = q$, $B = \neg p \vee \neg\neg p$ entonces $\not\Vdash \forall p(q \vee \neg p \vee \neg\neg p) \to q \vee \forall p(\neg p \vee \neg\neg p)$ puesto que $\mathcal{W}$ es un modelo completo por el lema (4.23) y $\mathcal{W} \not\Vdash \forall p(q \vee \neg p \vee \neg\neg p) \to q \vee \forall p(\neg p \vee \neg\neg p)$ es consecuencia inmediata de los lemas (4.24) y (4.25). □

Pasamos ahora a discutir el poder expresivo de la lógica proposicional de segundo orden.

5. Poder Expresivo del fragmento $\mathsf{Prop2}^{\to,\forall}$

En esta sección presentamos algunos ejemplos que muestran el poder expresivo del fragmento $\mathsf{Prop2}^{\to,\forall}$, obtenido al restringir las fórmulas de Prop2 al uso exclusivo de $\to$ y $\forall$. El propósito principal aquí no es sólo enunciar dichos ejemplos, como es común en la literatura, sino explicarlos con detalle usando para esto una lógica tal vez más familiar, a saber la lógica de predicados de segundo orden Pred2. La idea general es que en presencia de objetos de segundo orden las características de primer orden se vuelven poco importantes en muchos casos por lo que muchas propiedades estructurales pueden estudiarse de forma simplificada en Prop2.
Es importante remarcar que tanto en la lógica intuicionista de predicados de primer orden, como en la lógica proposicional intuicionista los conectivos y cuantificadores no son definibles entre sí (veáse [17]), característica que sí posee la lógica Prop2, como veremos a continuación.

5.1. Disyunción. Sean P, Q dos predicados de un argumento, el predicado que representa la unión de P y Q, denotado $P \cup Q$, se define inductivamente[1] como sigue:

- Si Px entonces $(P \cup Q)x$.
- Si Qx entonces $(P \cup Q)x$.
- $(P \cup Q)$ es el predicado más pequeño que cumple las dos reglas anteriores, es decir, si R es un predicado tal que:
  - Si Px entonces Rx.
  - Si Qx entonces Rx.

  entonces $\forall x((P \cup Q)x \to Rx)$.

Es decir $P \cup Q$ es el predicado más pequeño que contiene a P y a Q, o equivalentemente, es la intersección de todos los predicados que contienen a P y a Q. Esta última propiedad puede usarse para definirlo de manera directa:

$(P \cup Q)x \Leftrightarrow_{def}$ para todo predicado R tal que $P \subseteq R$ y $Q \subseteq R$ se cumple Rx.

Lo cual puede formalizarse en lógica de predicados de segundo orden como sigue:

$$(P \cup Q)x \Leftrightarrow_{def} \forall R.\forall y(Py \to Ry) \to \forall z(Qz \to Rz) \to Rx$$

Ahora bien, en el estudio de ciertas propiedades del predicado unión, no son de importancia sus elementos, por lo que podemos deshacernos de los objetos de primer orden obteniendo la siguiente fórmula en Prop2.

$$P \cup Q \Leftrightarrow_{def} \forall R.(P \to R) \to (Q \to R) \to R, \qquad R \neq P, Q$$

[1] Usamos *inductivamente* y no *recursivamente* como es costumbre en Ciencias Computacionales.

Finalmente observando que un predicado no sólo se define mediante un símbolo sino que puede definirse mediante cualquier fórmula, y que en terminología lógica la unión corresponde a la disyunción, obtenemos la definición usual en Prop2:

$$A \vee B \Leftrightarrow_{def} \forall r.(A \to r) \to (B \to r) \to r \qquad r \notin FV(A,B)$$

5.2. Conjunción. La conjunción corresponde a la definición inductiva del producto cartesiano de dos predicados de un argumento P y Q, que resulta ser un predicado binario denotado $(P \times Q)$:

$(P \times Q)xy \Leftrightarrow_{def} Rxy$ para todo predicado binario R tal que Px y Qy implican Rxy

Esta representación surge de la siguiente definición inductiva:

- Si Px y Qy entonces $(P \times Q)xy$.
- $(P \times Q)$ es el predicado más pequeño que cumple la regla anterior, es decir, si R es un predicado tal que:
  - Si Px y Qy entonces Rxy

  entonces $\forall x \forall y((P \times Q)xy \to Rxy)$.

Esta propiedad puede expresarse en lógica de predicados de segundo orden como:

$$(P \times Q)xy \Leftrightarrow_{def} \forall R.\forall x \forall y(Px \to Qy \to Rxy) \to Rxy$$

Finalmente eliminando los objetos de primer orden, generalizando a cualesquiera fórmulas A, B y usando la notación de conjunción obtenemos la definición usual en Prop2:

$$A \wedge B \Leftrightarrow_{def} \forall r.(A \to B \to r) \to r, \quad r \notin FV(A,B)$$

5.3. Falso. Considérese la fórmula $\forall pp$, que afirma la validez de cualquier proposición. Utilizando la regla de eliminación del cuantificador universal a partir de $\forall pp$ obtenemos $p[p := A] = A$ para cualquier fórmula A. Esto es lo que expresa la regla de eliminación del falso $(\perp E)$, por lo que resulta adecuado definir

$$\perp \Leftrightarrow_{def} \forall pp$$

Otra manera de entender esta definición es considerando que $\perp$ corresponde a la definición del predicado vacío $\varnothing$ visto como intersección de todos los predicados

$$\varnothing x \Leftrightarrow_{def} \forall P.Px$$

5.4. Cuantificador Existencial. El cuantificador existencial debe definirse de forma que se cumpla la fórmula $A[p := B] \to \exists pA$. Dado que la fórmula B es arbitraria debemos considerar válido el existencial cuando se pruebe $A[p := B]$ para alguna B, que es lo que afirma la regla de introducción $(\exists I)$. Sea E_A un predicado de índice uno tal que $E_A x$ se cumple si y sólo si $A[p := B]x$ para alguna fórmula B. Nuevamente podemos dar una definición inductiva.

- Se cumple $\forall x(A[p := B]x \to E_A x)$
- E_A es el predicado más pequeño con tal propiedad, es decir, si Q es un predicado tal que
  - $\forall x(A[p := B]x \to Qx)$

  entonces $\forall x(E_A x \to Qx)$

La definición correspondiente en lógica de segundo orden es:

$$E_A x \;\Leftrightarrow_{def}\; \forall Q.\forall y(A[p := B]y \to Qy) \to Qx$$

y eliminando los objetos de primer orden obtenemos:

$$E_A \Leftrightarrow_{def} \forall Q.\big((A[p := B] \to Q) \to Q\big)$$

Hasta ahora la fórmula definida depende de B pero B es una fórmula arbitraria de manera que la condición $A[p := B] \to Q$ debe generalizarse para eliminar esta dependencia, para lo cual usamos una cuantificación universal obteniendo la nueva condición $\forall p(A \to Q)$. De esta forma concluimos la definición usual de existencial en $\mathsf{Prop2}$:

$$\exists pA \Leftrightarrow_{def} \forall q\big(\forall p(A \to q) \to q\big),\;\; q \notin FV(A),\;\; p \neq q$$

Otra manera de entender esta definición nos viene de la lógica infinitaria, donde el existencial $\exists pA$ no es más que una disyunción infinita de todas las fórmulas $A[p := B]$. Esto queda claro al comparar la definición anterior con la definición de disyunción de la sección 5.1. En particular se sigue que $\vdash \exists pA \to \neg\forall p\neg A$.

Los ejemplos anteriores proponen definiciones de los operadores $\bot, \land, \lor, \exists$. Sin embargo para garantizar que estas definiciones son correctas debemos verificar que cumplen con las reglas de inferencia del sistema de deducción natural. Es decir, debemos mostrar que las reglas correspondientes de introducción y eliminación de cada operador son derivables usando estas definiciones. Esto lo asegura la siguiente

Proposición 5.1. *Las definiciones anteriores para $\bot, \land, \lor, \exists$ son correctas, es decir, las reglas usuales del sistema de deducción natural para dichos operadores son derivables.*

Demostración. Derivamos cada una de las reglas:

- $(\bot E)$ Suponemos $\Gamma \vdash \bot$.

1.	$\Gamma \vdash \bot$	Supuesto
2.	$\Gamma \vdash \forall pp$	Definición de $\bot$ en 1
3.	$\Gamma \vdash A$	$(\forall E)$ $2, p := A$

- $(\land I)$. Suponemos $\Gamma \vdash A,\; \Gamma \vdash B$.

1.	$\Gamma, A \to B \to r \vdash A$	Supuesto, (Mon)
2.	$\Gamma, A \to B \to r \vdash B$	Supuesto, (Mon)
3.	$\Gamma, A \to B \to r \vdash A \to B \to r$	(Hip)
4.	$\Gamma, A \to B \to r \vdash B \to r$	$(\to E)$ $1,3$
5.	$\Gamma, A \to B \to r \vdash r$	$(\to E)$ $2,4$
6.	$\Gamma \vdash (A \to B \to r) \to r$	$(\to I)$ 5
7.	$\Gamma \vdash \forall r.(A \to B \to r) \to r$	$(\forall I)$ 6, s.p.g. $r \notin FV(\Gamma)$
8.	$\Gamma \vdash A \land B$	Definición de $\land$ en 7, s.p.g. $r \notin FV(A,B)$

- $(\wedge_2 E)$ Suponemos $\Gamma \vdash A \wedge B$

1.	$\Gamma \vdash A \wedge B$	Supuesto
2.	$\Gamma \vdash \forall r.(A \to B \to r) \to r$	Definición de $\wedge$ en 1
3.	$\Gamma \vdash (A \to B \to B) \to B$	$(\forall E)$ 2, $r := B$, $r \notin FV(A,B)$
4.	$\Gamma, A, B \vdash B$	(Hip)
5.	$\Gamma, A \vdash B \to B$	$(\to I)$ 4
6.	$\Gamma \vdash A \to B \to B$	$(\to I)$ 5
7.	$\Gamma \vdash B$	$(\to E)$ 3, 6

La regla $(\wedge_1 E)$ se verifica de forma similar.

- $(\vee_1 I)$ Suponemos $\Gamma \vdash A$

1.	$\Gamma, A \to r, B \to r \vdash A$	Supuesto, (Mon)
2.	$\Gamma, A \to r, B \to r \vdash A \to r$	(Hip)
3.	$\Gamma, A \to r, B \to r \vdash r$	$(\to E)$ 1, 2
4.	$\Gamma, A \to r \vdash (B \to r) \to r$	$(\to I)$ 3
5.	$\Gamma \vdash (A \to r) \to (B \to r) \to r$	$(\to I)$ 4
6.	$\Gamma \vdash \forall r.(A \to r) \to (B \to r) \to r$	$(\forall I)$ 5, s.p.g. $r \notin FV(\Gamma)$
7.	$\Gamma \vdash A \vee B$	Definición de $\vee$ en 6, s.p.g. $r \notin FV(A,B)$

La regla $(\vee_2 I)$ se verifica de forma similar.

- $(\vee E)$ Suponemos $\Gamma \vdash A \vee B$, $\Gamma, A \vdash C$ y $\Gamma, B \vdash C$.

1.	$\Gamma \vdash A \vee B$	Supuesto
2.	$\Gamma \vdash \forall r.(A \to r) \to (B \to r) \to r$	Definición de $\vee$ en 1
3.	$\Gamma \vdash (A \to C) \to (B \to C) \to C$	$(\forall E)$ 2, $r := C$, $r \notin FV(A,B)$
4.	$\Gamma, A \vdash C$	Supuesto
5.	$\Gamma \vdash A \to C$	$(\to I)$ 4
6.	$\Gamma \vdash (B \to C) \to C$	$(\to E)$ 3, 5
7.	$\Gamma, B \vdash C$	Supuesto
8.	$\Gamma \vdash B \to C$	$(\to I)$ 7
9.	$\Gamma \vdash C$	$(\to E)$ 6, 8

- $(\exists I)$ Suponemos $\Gamma \vdash A[p := B]$.

1.	$\Gamma, \forall p(A \to q) \vdash A[p := B]$	Supuesto, (Mon)
2.	$\Gamma, \forall p(A \to q) \vdash \forall p(A \to q)$	(Hip)
3.	$\Gamma, \forall p(A \to q) \vdash A[p := B] \to q$	$(\forall E)$ 2, $p := B$
4.	$\Gamma, \forall p(A \to q) \vdash q$	$(\to E)$ 1, 3
5.	$\Gamma \vdash \forall p(A \to q) \to q$	$(\to I)$ 4
6.	$\Gamma \vdash \forall q(\forall p(A \to q) \to q)$	$(\forall I)$ 5, s.p.g. $q \notin FV(\Gamma)$
7.	$\Gamma \vdash \exists pA$	Definición de $\exists$ en 6, s.p.g. $q \notin FV(A)$, $p \neq q$

- $(\exists E)$ Suponemos $\Gamma \vdash \exists pA,\ \Gamma, A \vdash B$ y $p \notin FV(\Gamma, B)$

1.	$\Gamma \vdash \exists pA$	Supuesto
2.	$\Gamma \vdash \forall q(\forall p(A \to q) \to q)$	Definición de $\exists$ en 1
3.	$\Gamma \vdash \forall p(A \to B) \to B$	$(\forall E)$ $2, q := B,\ p \notin FV(B),\ q \notin FV(A)$
4.	$\Gamma, A \vdash B$	Supuesto
5.	$\Gamma \vdash A \to B$	$(\to I)$ 4
6.	$\Gamma \vdash \forall p(A \to B)$	$(\forall I)$ 5, $p \notin FV(\Gamma)$
7.	$\Gamma \vdash B$	$(\to E)$ $6, 3$

□

La proposición anterior muestra que los operadores $\bot, \wedge, \vee, \exists$ son definibles a partir de $\forall, \to$ únicamente, de manera que al mostrar alguna propiedad de Prop2 bastará con mostrarla para el fragmento $\mathsf{Prop2}^{\to,\forall}$.

6. La relación entre Pred2 y Prop2

Después de haber visto diversos ejemplos del poder expresivo de Prop2, los cuales se basan en definiciones en la lógica de predicados de segundo orden Pred2, resulta natural preguntarnos si existirá una relación general entre ambas lógicas. La respuesta es positiva y será demostrada en esta sección, para lo cual empezamos definiendo las reglas de deducción natural para los cuantificadores universales de primer y segundo orden en Pred2:

$$\frac{\Gamma \vdash A \quad x \notin FV(\Gamma)}{\Gamma \vdash \forall xA}\ (\forall I) \qquad \frac{\Gamma \vdash \forall xA}{\Gamma \vdash A[x := t]}\ (\forall E)$$

donde $A[x := t]$ denota a la sustitución de la variable de primer orden x por el término t en la fórmula A, la cual se define análogamente a la definición (2.2) evitando la captura de variables libres.

$$\frac{\Gamma \vdash A \quad X \notin FV(\Gamma)}{\Gamma \vdash \forall XA}\ (\forall^2 I) \qquad \frac{\Gamma \vdash \forall XA}{\Gamma \vdash A[X := F]}\ (\forall^2 E)$$

la operación $A[X := F]$ es esencialmente la dada en la definición (3.4) con la restricción de que la variable de segundo orden X se sustituya por una fórmula F que reciba el mismo número de argumentos.

De manera análoga al caso de Prop2 los operadores $\bot, \wedge, \vee, \exists, \exists^2$ resultan definibles en el fragmento $\mathsf{Pred2}^{\to,\forall,\forall^2}$, por lo que basta estudiar la relación entre este fragmento y $\mathsf{Prop2}^{\to,\forall}$. La siguiente proposición muestra que la estructura de las derivaciones es exactamente la misma en ambas lógicas por lo que es válido eliminar en diversos casos las referencias a objetos de primer orden.

Definición 6.1. Dada una fórmula A en Pred2 la fórmula A' de la lógica Prop2 obtenida al eliminar los objetos de primer orden se define recursivamente como sigue:

$$\begin{aligned} X(t_1, \ldots, t_n)' &= X \\ (A \to B)' &= A' \to B' \\ (\forall xA)' &= A' \\ (\forall XA)' &= \forall XA' \end{aligned}$$

Obsérvese que en el caso de una fórmula atómica y una cuantificación de segundo orden la variable X que representa a un predicado de índice arbitrario n se convierte en una variable proposicional con el mismo nombre X. Además si $\Gamma = \{A_1, \ldots, A_n\}$ es un contexto en $\mathsf{Pred2}$ entonces el contexto Γ' de $\mathsf{Prop2}$ se define como $\Gamma' = \{A_1', \ldots, A_n'\}$.

La operación de traslación $(\cdot)'$ es compatible con la operación de sustitución de acuerdo al siguiente

Lema 6.2. *Se cumplen las siguientes propiedades para cualquier fórmula A de* $\mathsf{Pred2}$.

- $A[x := t]' = A'$.
- $A[X := F]' = A'[X := F']$.

Demostración. Inducción sobre A en ambos casos. □

La siguiente proposición muestra la relación exacta entre $\mathsf{Pred2}$ y $\mathsf{Prop2}$.

Proposición 6.3. *Sea A una fórmula de* $\mathsf{Pred2}$. *Entonces* $\Gamma \vdash_{\mathsf{Pred2}} A$ *si y sólo si* $\Gamma' \vdash_{\mathsf{Prop2}} A'$.

Demostración. La parte $\Leftarrow$) es inmediata pues $\mathsf{Prop2}$ está contenida en $\mathsf{Pred2}$.
Para $\Rightarrow$) procedemos por inducción sobre $\vdash_{\mathsf{Pred2}}$. Analicemos con detalle el caso de los cuantificadores universales, para los de primer orden no hay nada que mostrar pues las reglas $(\forall I)$, $(\forall E)$ se colapsan en $\mathsf{Prop2}$ dado que $(\forall x A)' = A'$ y por el lema (6.2), $A[x := t]' = A'$. Veamos el caso de los cuantificadores de segundo orden:

- $(\forall^2 I)$. Sea $\Gamma \vdash_{\mathsf{Pred2}} \forall X A$ concluido de $\Gamma \vdash_{\mathsf{Pred2}} A$ con $X \notin FV(\Gamma)$. Por H.I. se tiene $\Gamma' \vdash_{\mathsf{Prop2}} A'$ y es fácil ver que si $X \notin FV(\Gamma)$ entonces $X \notin FV(\Gamma')$ de donde $\Gamma' \vdash_{\mathsf{Prop2}} \forall X A'$ por $(\forall^2 I)$ y terminamos.
- $(\forall^2 E)$. Sea $\Gamma \vdash_{\mathsf{Pred2}} A[X := F]$ obtenido de $\Gamma \vdash_{\mathsf{Pred2}} \forall X A$. Por H.I. se cumple $\Gamma' \vdash_{\mathsf{Prop2}} \forall X A'$ y aplicando $(\forall^2 E)$ tenemos $\Gamma' \vdash_{\mathsf{Prop2}} A'[X := F']$. Finalmente por el lema (6.2) podemos concluir $\Gamma' \vdash_{\mathsf{Prop2}} A[X := F]'$.

□

A la luz de la proposición anterior debe quedar clara nuestra afirmación previa acerca de que diversas propiedades estructurales de $\mathsf{Pred2}$ pueden estudiarse de manera más simple y sin perder generalidad en $\mathsf{Prop2}$.

Terminamos nuestra exposición mostrando algunas aplicaciones de la lógica $\mathsf{Prop2}$ a las Ciencias Computacionales.

7. Relevancia en Ciencias de la Computación

La lógica $\mathsf{Prop2}$ es de gran importancia en Ciencias de la Computación dado que corresponde, bajo la correspondencia de Curry-Howard (véase [17]) al cálculo lambda polimórfico de segundo orden $\lambda 2$, también conocido como sistema F (véase [7]). Dicha correspondencia afirma esencialmente que las derivaciones en $\mathsf{Prop2}$ pueden codificarse mediante programas del sistema F. No hablaremos más aquí del cálculo lambda pues lo que nos concierne en este artículo es la lógica. En su lugar explicamos desde el punto de vista lógico dos de sus mecanismos relevantes, la definición de tipos de datos inductivos y abstractos.

7.1. Definición de Tipos de Datos Inductivos. El método de definición para los operadores lógicos visto en la sección 5 puede aplicarse para definir cualquier tipo de datos definido inductivamente. Veamos con detalle el caso de los números naturales, estructura de datos presente en todo lenguaje de programación. Los naturales son el predicado más pequeño $\mathbb{N}$ cerrado bajo el cero y la función sucesor, es decir:

- $\mathbb{N}0$
- $\forall x(\mathbb{N}x \to \mathbb{N}sx)$
- Si P es un predicado tal que
  - $P0$
  - $\forall x(Px \to Psx)$

  entonces $\forall x(\mathbb{N}x \to Px)$

De donde se obtiene la definición en lógica de predicados de segundo orden:

$$\mathbb{N}y \Leftrightarrow_{def} \forall P.\forall x(Px \to Psx) \to P0 \to Py$$

Es de importancia observar que existen pruebas canónicas, llamadas normales en teoría de la demostración, de la fórmula $\mathbb{N}t$, para cada término t de la forma 0 ó $s^n x$. Por ejemplo:

- $\mathbb{N}0$ se deriva así:

$$\begin{array}{lll}
1. & \forall x(Px \to Psx),\ P0 \vdash P0 & (Hip) \\
2. & \forall x(Px \to Psx) \vdash P0 \to P0 & (\to I)\ 1 \\
3. & \vdash \forall x(Px \to Psx) \to P0 \to P0 & (\to I)\ 2 \\
4. & \vdash \forall P.\forall x(Px \to Psx) \to P0 \to P0 & (\forall^2 I)\ 3
\end{array}$$

- $\mathbb{N}s0$ se deriva así

$$\begin{array}{lll}
1. & \forall x(Px \to Psx),\ P0 \vdash \forall x(Px \to Psx) & (Hip) \\
2. & \forall x(Px \to Psx),\ P0 \vdash P0 \to Ps0 & (\forall E)\ 1,\ x := 0 \\
3. & \forall x(Px \to Psx),\ P0 \vdash P0 & (Hip) \\
4. & \forall x(Px \to Psx),\ P0 \vdash Ps0 & (\to E)\ 2,3 \\
5. & \forall x(Px \to Psx) \vdash P0 \to Ps0 & (\to I)\ 4 \\
6. & \vdash \forall x(Px \to Psx) \to P0 \to Ps0 & (\to I)\ 5 \\
7. & \vdash \forall P.\forall x(Px \to Psx) \to P0 \to Ps0 & (\forall^2 I)\ 6
\end{array}$$

- $\mathbb{N}ss0$ se deriva así:

$$\begin{array}{lll}
1. & \forall x(Px \to Psx),\ P0 \vdash \forall x(Px \to Psx) & (Hip) \\
2. & \forall x(Px \to Psx),\ P0 \vdash P0 \to Ps0 & (\forall E)\ 1,\ x := 0 \\
3. & \forall x(Px \to Psx),\ P0 \vdash Ps0 \to Pss0 & (\forall E)\ 1,\ x := s0 \\
4. & \forall x(Px \to Psx),\ P0 \vdash P0 & (Hip) \\
5. & \forall x(Px \to Psx),\ P0 \vdash Ps0 & (\to E)\ 2,4 \\
6. & \forall x(Px \to Psx),\ P0 \vdash Pss0 & (\to E)\ 3,5 \\
7. & \forall x(Px \to Psx) \vdash P0 \to Pss0 & (\to I)\ 6 \\
8. & \vdash \forall x(Px \to Psx) \to P0 \to Pss0 & (\to I)\ 7 \\
9. & \vdash \forall P.\forall x(Px \to Psx) \to P0 \to Pss0 & (\forall^2 I)\ 8
\end{array}$$

De estos ejemplos podemos observar que cada prueba normal de una fórmula $\mathbb{N}s^n x$ puede obtenerse efectivamente, de manera que es posible generarla o verificarla mecanicamente, por lo que dichas pruebas proporcionan una manera de implementar al tipo de los números naturales en algún lenguaje de programación identificando al número n con la prueba correspondiente Π_n de la fórmula $\mathbb{N}s^n0$. En la práctica, en un lenguaje de programación utilizamos variables para programar, las cuales en la evaluación final se sustituyen por valores concretos, por lo que si pensamos en una implementación resulta inadecuado usar variables en la representación de un tipo como los números naturales nat y en la sintaxis misma del lenguaje. Tendríamos que distinguir al menos dos clases de variables. Afortunadamente podemos servirnos de la lógica $\mathsf{Prop2}$ para eliminar las variables de primer orden, obteniendo así una representación más simple y ventajosa de los números naturales. Si definimos

$$\mathsf{nat} \Leftrightarrow_{def} \forall p.(p \to p) \to p \to p$$

entonces un número natural n puede implementarse como una prueba canónica π_n de la fórmula nat. Por ejemplo:

- El cero corresponde a la siguiente derivación π_0:

1.	$p \to p, p \vdash p$	(Hip)
2.	$p \to p \vdash p \to p$	$(\to I)$ 1
3.	$\vdash (p \to p) \to p \to p$	$(\to I)$ 2
4.	$\vdash \forall p.(p \to p) \to p \to p$	$(\forall I)$ 3

- El uno corresponde a la siguiente derivación π_1:

1.	$p \to p, p \vdash p \to p$	(Hip)
2.	$p \to p, p \vdash p$	(Hip)
3.	$p \to p, p \vdash p$	$(\to E)$ 1, 2
4.	$p \to p \vdash p \to p$	$(\to I)$ 3
5.	$\vdash (p \to p) \to p \to p$	$(\to I)$ 4
6.	$\vdash \forall p.(p \to p) \to p \to p$	$(\forall I)$ 5

- El dos corresponde a la siguiente derivación π_2:

1.	$p \to p, p \vdash p \to p$	(Hip)
2.	$p \to p, p \vdash p$	(Hip)
3.	$p \to p, p \vdash p$	$(\to E)$ 1, 2
4.	$p \to p, p \vdash p$	$(\to E)$ 1, 3
5.	$p \to p \vdash p \to p$	$(\to I)$ 4
6.	$\vdash (p \to p) \to p \to p$	$(\to I)$ 5
7.	$\vdash \forall p.(p \to p) \to p \to p$	$(\forall I)$ 6

Obsérvese que estas pruebas son más simples que las anteriores desarrolladas en $\mathsf{Pred2}$ y si bien parecen redundantes ahora sabemos que provienen de pruebas canónicas en $\mathsf{Pred2}$. En particular la prueba π_n de nat en $\mathsf{Prop2}$ corresponde a la prueba Π_n de $\mathbb{N}s^n0$ en $\mathsf{Pred2}$, pues esencialmente se tiene $\Pi'_n = \pi_n$. De esta manera la lógica $\mathsf{Prop2}$ puede verse como un prototipo de lenguaje de programación donde las pruebas son programas, esto no es más que una instancia de la conocida correspondencia o isomorfismo de Curry-Howard. En particular nuestra fórmula nat corresponde al tipo de los llamados numerales de Church en el cálculo lambda, véase [16].

De manera similar podemos obtener fórmulas para representar otras estructuras de datos como listas y árboles:

- Listas sobre A: la definición inductiva de listas finitas de objetos de tipo A nos lleva a la siguiente definición en Pred2

$$\mathbb{L}(A)x \Leftrightarrow_{def} \forall P.\forall z\forall y(Az \to Py \to P(cons\, z\, y)) \to P(nil) \to Px$$

donde nil es una constante que denota a la lista vacía y $cons\, x\, y$ es un término que denota a la función de construcción de listas que agrega un elemento x de A al inicio de una lista dada y.
La fórmula correspondiente en Prop2 es

$$\mathsf{list}(A) \Leftrightarrow_{def} \forall p.(A \to p \to p) \to p \to p$$

Obsérvese que podemos usar el poder de la cuantificación universal proposicional para definir un constructor genérico de listas:

$$\mathsf{list} \Leftrightarrow_{def} \forall q\forall p.(q \to p \to p) \to p \to p$$

- Árboles binarios sin etiquetas: Si $void$ denota al árbol vacío y $mktree$ a una función que construye un nuevo árbol binario a partir de dos dados, entonces se define el predicado de árboles binarios no etiquetados como:

$$\mathbb{T}x \Leftrightarrow_{def} \forall P.\forall z\forall y(Pz \to Py \to P(mktree\, z\, y)) \to P(void) \to Px$$

De donde el constructor para árboles binarios no etiquetados en Prop2 es

$$\mathsf{btree} \Leftrightarrow_{def} \forall p.(p \to p \to p) \to p \to p$$

Los ejemplos anteriores dejan ver un patrón general, de hecho todo conjunto definido inductivamente puede modelarse en Prop2, véase [8, 7]. Si se desea programar utilizando estos tipos de datos, es común usar la contraparte funcional, es decir, el cálculo lambda F, véase por ejemplo [14, 12], aunque también es posible programar directamente con pruebas en Pred2, como se explica en [13, 9].
También es importante observar que el uso de las definiciones anteriores no es necesariamente amigable por lo que se han desarrollado extensiones que simplifican el mecanismo de definiciones inductivas tanto en la lógica de proposiciones como en la de predicados. A este respecto es útil referirse a [9, 10].

7.2. Fórmulas Existenciales y Tipos de Datos Abstractos. Al igual que las fórmulas universales sirven para definir tipos de datos inductivos, las existenciales nos sirven para definir tipos de datos abstractos, los cuales son esencialmente especificaciones de un conjunto de datos junto con diversas operaciones sobre los mismos sin importar su implementación específica. La fórmula $\exists pA$ modela a un tipo definido por A que involucra a cierta proposición p, la cual no está disponible para el usuario, dicha p representa a una proposición B tal que $A[p := B]$ es válida, en este sentido B es una implementación particular de p.

Veamos un ejemplo partiendo nuevamente desde la lógica de predicados de segundo orden: supóngase que se quiere especificar una pila con valores en cierto conjunto A y las siguientes operaciones: nil que denota a la pila vacía, $push$ que dados un elemento x de A y una pila y construye una nueva pila agregando x al tope de y, pop que dada una pila x, elimina el elemento en el tope y top que devuelve el elemento en el tope de una pila.

Una especificación en Pred2 consta de las siguientes fórmulas, donde $\mathbb{P}_A x$ significa que x es una pila de elementos de A:

- $\mathbb{P}_A\, nil$
- $\forall x \forall y. A x \to \mathbb{P}_A y \to \mathbb{P}_A\, push\, x\, y$
- $\forall x. \mathbb{P}_A x \to \mathbb{P}_A\, pop\, x$
- $\forall x. \mathbb{P}_A x \to A\, top\, x$

En esta especificación $\mathbb{P}_A$ representa a una implementación particular de las pilas, la cual es irrelevante para un usuario en muchos casos por lo que nos bastaría con decir que tal implementación existe, obteniendose la fórmula

$$\begin{array}{l} \exists Q.\, Q\, nil \;\wedge \\ \quad \forall x \forall y. A\, x \to Q\, y \to Q\, push\, x\, y \;\wedge \\ \quad \forall x. Q\, x \to Q\, pop\, x \;\wedge \\ \quad \forall x. Q\, x \to A\, top\, x \end{array}$$

de donde se genera la siguiente representación en Prop2:

$$\mathsf{pila}(A) \Leftrightarrow_{def} \exists q.\, q \;\wedge\; (A \to q \to q) \;\wedge\; (q \to q) \;\wedge\; (q \to A)$$

Finalmente usando todo el poder de Prop2 podemos eliminar la referencia a A obteniendo la siguiente definición genérica de pila:

$$\mathsf{pila} \Leftrightarrow_{def} \forall p.\, \exists q.\, q \;\wedge\; (p \to q \to q) \;\wedge\; (q \to q) \;\wedge\; (q \to p)$$

Es pertinente observar en este momento que este uso de las fórmulas existenciales es adecuado únicamente en la lógica intuicionista donde se cumple la siguiente propiedad constructiva:

- Si $\vdash \exists p A$ entonces $\vdash A[p := B]$ para algúna $B \in \mathsf{FProp}$

Es decir, si un existencial es válido es que hay un testigo de él, lo cual en particular implica que si usamos la especificación de pilas dada arriba es que hay una implementación particular disponible. La afirmación anterior, llamada propiedad existencial, es consecuencia directa del teorema de normalización para Prop2 (véase [17]) o bien del teorema de eliminación de corte en un sistema con secuentes à la Gentzen, el autor desconoce y es de su interés, si existe una prueba semántica con modelos de Kripke o álgebras de Heyting, como en el caso de la propiedad disyuntiva.

Estas y otras aplicaciones de las fórmulas universales y existenciales vistas como tipos pueden estudiarse a profundidad en [14, 11].

8. Comentarios Finales

La Lógica Proposicional de Segundo Orden Prop2 y su versión funcional, conocida como sistema F o $\lambda 2$, fueron desarrollados por Jean-Yves Girard en [5, 6] en el ámbito de la teoría de la demostración, con el propósito de mostrar la consistencia de la aritmética de Heyting de segundo orden HA2. Girard necesitaba un sistema notacional para las derivaciones el cual es esencialmente el sistema F cuya introducción permitio mostrar que $\not\vdash_{\mathsf{HA2}} \bot$. Independientemente, un poco más tarde, John Reynolds redescubre el sistema con propósitos totalmente distintos, a saber, la formalización del concepto de polimorfismo en lenguajes de programación, véase [15]. El sistema F ha servido de base constantemente para la investigación de fundamentos en lenguajes de programación y es componente esencial del diseño de diversos lenguajes. Precisamente su importancia en Ciencias de la Computación y la inexistencia, hasta donde sé, de un artículo expositorio de la contraparte puramente lógica, me han llevado a desarrollar este artículo donde no sólo he expuesto a detalle la sintaxis y semántica intuicionista de Prop2, también he explicado desde el punto de vista lógico ciertas aplicaciones computacionales relevantes. Cualquier otro comentario o pregunta se recibirá con gusto en mi dirección de correo electrónico.

REFERENCIAS

[1] Dirk van Dalen, "Logic and Structure", 4th. ed. *Springer*., 2004.

[2] Herbert B. Enderton, "Una introducción matemática a la lógica", *IIF-UNAM*., 2004.

[3] Dov M. Gabbay, *On 2nd order intuitionistic propositional calculus with full comprehension*, Archiv für mathematische Logik und Grundlagenforschung, **16** (1974), 177–186.

[4] Herman Geuvers, Logics and Type Systems. Ph.D. Thesis. University of Nijmegen., 1993.

[5] Jean-Yves Girard, *Une extension de l'interpretation de Gödel à l'analyse, et son application à l'elimination des coupures dans l'analyse et la théorie des types*, En J.E. Fenstad, editor. Second Scandinavian Logic Symposium, 63–92, *North-Holland*., 1971.

[6] Jean-Yves Girard, Interpretation fonctionelle et elimination des coupures de l'arithmetique d'ordre superieur, These D'Etat,*Universite Paris VIII*., 1972.

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

[8] Daniel Leivant, *Contracting proofs to programs*, En P.Odifreddi, editor, Logic and Computer Science, 279–327. *Academic Press*., 1990.

[9] Favio E. Miranda Perea, On Extensions of AF2 with Monotone and Clausular (Co)inductive Definitions. Dissertation, *Ludwig-Maximilians-Universität München*., 2004.

[10] Favio E. Miranda Perea, *Two Extensions of System* F *with (Co)iteration and Primitive (Co)recursion Principles*, Por aparecer en Theoretical Informatics and Applications. 2009.

[11] John Mitchell, "Foundations for Programming Languages", *MIT Press*., 1996

[12] Benjamin C. Pierce, Scott Diezen, Spiro Michaylov, Programming in Higher-Order Typed Lambda Calculi. Technical Report CMU-CS-89-111 *Carnegie Mellon University*., 1989.

[13] Michel Parigot, *Recursive Programming with Proofs*. Theoretical Computer Science **94** (1992), 335–356.

[14] Benjamin C. Pierce, "Types and Programming Languages", *MIT Press*., 2002.

[15] John Reynolds, *Towards a theory of type structure*, En B. Robinet, editor. Proc. Colloque sur la Programmation. *Springer*. Lecture Notes in Computer Science **19**, 1974.

[16] Helmut Schwichtenberg, Anne S. Troelstra. "Basic Proof Theory", *Cambridge University Press*, Cambridge Tracts in Theoretical Computer Science **43**, 1996.

[17] Morten H. Sørensen, Pawel Urzyczyn, "Lectures on the Curry-Howard Isomorphism", *Elsevier*, Studies in Logic and the Foundations of Mathematics, **149**, 2006.

[18] Anne S. Troelstra, Dirk van Dalen, "Constructivism in Mathematics, An Introduction, vol. 1", *North-Holland*, Studies in Logic and the Foundations of Mathematics, **121**, 1988.

DEPARTAMENTO DE MATEMÁTICAS, FACULTAD DE CIENCIAS, UNAM, CIRCUITO EXTERIOR S/N, CD. UNIVERSITARIA 04510, MÉXICO D.F., MÉXICO

E-mail address: `favio@matematicas.unam.mx`