



Revista de Matemática: Teoría y

Aplicaciones

ISSN: 1409-2433

mta.cimpa@ucr.ac.cr

Universidad de Costa Rica

Costa Rica

MARTÍNEZ, IVETH; PIZA, EDUARDO
PROBLEMAS DE DECISIÓN Y RECURSIVIDAD EN SISTEMAS LÓGICOS
FORMALES

Revista de Matemática: Teoría y Aplicaciones, vol. 23, núm. 1, enero, 2016, pp. 11-39
Universidad de Costa Rica
San José, Costa Rica

Disponible en: <http://www.redalyc.org/articulo.oa?id=45343487001>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

PROBLEMAS DE DECISIÓN Y RECURSIVIDAD EN SISTEMAS LÓGICOS FORMALES

DECISION PROBLEMS AND RECUSIVENESS IN FORMAL LOGIC SYSTEMS

IVETH MARTÍNEZ* EDUARDO PIZA†

*Received: 30 May 2014; Revised: 28 Aug 2015;
Accepted: 30 Sep 2015*

*IDEN, Universidad de Panamá, Ciudad de Panamá, Panamá. E-Mail:
iveth.martinez@up.ac.pa

†CIMPA, Universidad de Costa Rica, San José, Costa Rica. E-Mail: eduardo.piza@ucr.ac.cr

Resumen

En la teoría de la recursión se dice que un problema de decisión es recursivamente resoluble si existe un procedimiento mecánico para resolverlo. Dentro del contexto de las lógicas formales, el *problema de decisión* consiste simplemente en determinar si una *fórmula bien formada* cualquiera del sistema es, o no es, un teorema.

En este trabajo se analizan, entre otros asuntos, primeramente el famoso problema de decisión de la *lógica canónica de primer orden* \mathcal{F}_0 (también llamado *Entscheidungsproblem*) desde una perspectiva moderna. Luego se estudian los problemas de decisión de las *lógicas proposicionales parciales*. Se aprovecha el desarrollo alcanzado por la teoría de la recursión y de los sistemas productivos semi-Thue, luego de los trabajos de Post y Kleene en los años 40's y de Davis en la década de los 70's, entre otros, para explicar una solución a estos problemas de decisión.

Palabras clave: problemas de decisión, lógicas formales, lógicas de primer orden, lógicas proposicionales parciales, Entscheidungsproblem, sistemas productivos semi-Thue.

Abstract

The recursion theory states that a decision problem is recursively solvable if there is a mechanical process to solve it. Within the context of formal logic, the decision problem consist to determine whether any *well-formed formula* of the system is a theorem or not.

This paper first discusses, among other things, the famous problem of decision of the *canonical first-order logic* \mathcal{F}_0 (also called *Entscheidungsproblem*) from a modern perspective. Then we study the decision problem of the *partial propositional logics*. It exploits the development achieved by recursion theory and semi-Thue production systems after the work of Post and Kleene in the 40's and Davis in the early 70's, among others, to explain a solution to these decision problems.

Keywords: decision problems, formal logic, first-order logic, Entscheidungsproblem, partial propositional logics, semi-Thue production systems.

Mathematics Subject Classification: 03D80, 03D03, 03D25, 03B05, 03B10, 03B25.

1 Introducción

En pocas palabras, el *problema de decisión* para una lógica de primer orden consiste en determinar si una *fórmula bien formada* cualquiera dentro del sistema es o no es un teorema. Se dice que un problema de decisión es *recursivamente*

resoluble si existe un procedimiento efectivo (esto es, una lista de instrucciones o un algoritmo que no requiera de ninguna ingeniosidad para ser ejecutado) para resolverlo.

Dentro de las instancias de lógicas de primer orden, quizás la más estudiada es la “*lógica canónica de primer orden* \mathcal{F}_0 ”, también llamada *el cálculo de predicados restringido* por Hilbert y Ackermann, mientras que Church la denominaba “*el cálculo funcional puro de primer orden*”. La importancia de \mathcal{F}_0 proviene del resultado, bien conocido¹ desde principios del siglo XX, que establece que si el problema de decisión para \mathcal{F}_0 es recursivamente resoluble entonces también lo es el de cualquier lógica de primer orden.

Ya Hilbert y su reconocida Escuela Formalista dominaban intuitivamente este importante resultado, aún antes de la aparición y desarrollo formal de la teoría de la recursión, a mediados del siglo XX. Acerca de este problema Hilbert llegó a declarar que el problema de decisión para \mathcal{F}_0 (referido en alemán como el “*Entscheidungsproblem*”, que significa simplemente *problema de decisión*) era el problema central de la lógica matemática. Subsecuentemente se realizó mucha investigación dirigida a buscar una solución positiva al *Entscheidungsproblem*.

Algunos resultados preliminares para subclases de \mathcal{F}_0 arrojaron resultados positivos, alimentando la errónea esperanza que el *Entscheidungsproblem* fuera recursivamente resoluble y que el menor avance en cualquier dirección terminaría resolviendo positivamente el problema. Sin embargo, en 1936 tanto Church como Turing demostraron precisamente lo contrario, en forma independiente uno del otro: ¡el *Entscheidungsproblem* es recursivamente irresoluble! Church utilizó la técnica del λ -cálculo, mientras que Turing introdujo sus ahora célebres máquinas teóricas para resolver este problema.

Analizaremos aquí el *Entscheidungsproblem* y otros problemas asociados, entre ellos los problemas de decisión, completitud e independencia de las *lógicas proposicionales parciales*, desde la perspectiva moderna de la teoría de la recursión y de los *sistemas productivos semi-Thue*, introducidos por Post en la década de los 40’s.

2 Lógicas formales y sus aritmétizaciones

Trabajaremos con objetos matemáticos abstractos tales como “palabras”, “predicados de palabras” y funciones sobre conjuntos de palabras. Emplearemos la siguiente *aritmétización* de estos objetos matemáticos abstractos:

¹Aunque de manera intuitiva, ya que la teoría de la recursión aún no estaba suficientemente desarrollada para aquel entonces, lo que obstaculizaba la formalización del concepto de “problema de decisión”.

- Si $\Sigma = \{a_0, a_1, \dots\}$ es el alfabeto (finito o numerable) de símbolos, a cada símbolo a_i le asociamos el número Gödel $\varrho(a_i) := 2i + 1$.
- Si $u = a_{i_0}a_{i_1} \cdots a_{i_m}$ es una *palabra* (esto es, concatenación de símbolos sobre Σ), le asociamos el número Gödel $\varrho(u)$ definido por

$$\varrho(u) := \prod_{k=0}^m p_k^{\varrho(a_{i_k})},$$

donde p_k es el k -ésimo número primo, esto es, $p_0 = 2, p_1 = 3, p_2 = 5, \dots$, etc.

Un conjunto \mathcal{U} de palabras sobre Σ se dice que es *recursivo* si el conjunto asociado \mathcal{U}^* de los números Gödel de todas las palabras de \mathcal{U} es recursivo. Una función f entre conjuntos de palabras, $f: \mathcal{U} \mapsto \mathcal{V}$, se dice que es *recursiva* si su función asociada f^* es recursiva, donde $f^*: \mathcal{U}^* \mapsto \mathcal{V}^*$ es tal que

$$f^*(\varrho(u)) := \varrho(f(u)), \quad \forall u \in \mathcal{U}.$$

Emplearemos libremente la teoría de las funciones parcialmente recursivas. Por \mathfrak{R} denotamos la clase de las funciones parcialmente recursivas. Escribimos

$$\mathfrak{R} = \{\varphi_i^{(n)} : i \in \mathbb{N}, n \in \mathbb{N} \setminus \{0\}\},$$

donde $\varphi_i^{(n)}$ denota la i -ésima función parcialmente recursiva n -aria.

Empezamos definiendo lo que son los *sistemas lógico formales*, y dentro de ellos los conceptos primitivos siguientes: *axiomas, reglas de inferencia, consecuencia, prueba, etapa de una prueba y teorema*.

Definición 1 *Por una lógica \mathcal{L} (o sistema lógico formal) entenderemos:*

- Un conjunto recursivo de palabras \mathcal{U} , llamado los axiomas de \mathcal{L} .*
- Un conjunto finito de predicados recursivos de palabras, ninguno de los cuales es 1-ario, llamados reglas de inferencia de \mathcal{L} .*

Cuando \mathcal{R} es una regla de inferencia $(n+1)$ -aria de \mathcal{L} , y $\mathcal{R}(Y, X_1, \dots, X_n)$ es verdadera, entonces diremos que Y es una consecuencia de X_1, \dots, X_n en \mathcal{L} , a través de la regla de inferencia \mathcal{R} .

Definición 2 *Una sucesión finita de palabras X_1, X_2, \dots, X_m se llama una prueba en la lógica \mathcal{L} si, para cada $1 \leq i \leq m$, se cumple:*

- (a) *O bien X_i es un axioma de \mathcal{L} ,*
- (b) *O bien existen enteros $j_1, j_2, \dots, j_k < i$ tales que X_i es una consecuencia lógica de $X_{j_1}, X_{j_2}, \dots, X_{j_k}$ en \mathcal{L} a través de alguna de las reglas de inferencia de \mathcal{L} .*

Cada una de las palabras X_i es llamada una etapa de la prueba.

Definición 3 *Decimos que W es un teorema de \mathcal{L} , o que W es demostrable en \mathcal{L} , y escribimos*

$$\vdash_{\mathcal{L}} W$$

si existe una prueba en \mathcal{L} cuya etapa final sea W . Esta prueba es entonces llamada una prueba de W en \mathcal{L} .

3 Problema de decisión de una lógica formal

Definición 4 *Dada una lógica formal \mathcal{L} , al conjunto de todos los números Gödel correspondientes a los teoremas de \mathcal{L} lo denotamos por $T_{\mathcal{L}}$ (“conjunto engendrado” por \mathcal{L}), esto es:*

$$T_{\mathcal{L}} := \{\varrho(W) : \vdash_{\mathcal{L}} W\}.$$

Es bien conocido que $T_{\mathcal{L}}$ es un conjunto *recursivamente enumerable*, esto es, existe un procedimiento efectivo para ir enumerando los teoremas de \mathcal{L} uno detrás de otro. Este hecho no lo vamos a demostrar en este trabajo. El lector interesado en la demostración puede consultar Piza [10]. No obstante, el hecho que $T_{\mathcal{L}}$ sea recursivamente enumerable no nos brinda un método eficaz para decidir si una fórmula bien formada arbitraria W es un teorema en \mathcal{L} o no lo es. La dificultad surge al buscar W dentro de la lista de teoremas recursivamente enumerados de \mathcal{L} , pues en caso que W no aparezca en la lista de los primeros teoremas generados, no tendremos manera de establecer si W es o no un teorema de \mathcal{L} . Extendemos la definición de problema de decisión para \mathcal{L} de la siguiente manera:

Definición 5 *Por “el problema de decisión para la lógica \mathcal{L} ” entenderemos el problema de determinar, para una palabra cualquiera dada, si es o no un teorema de \mathcal{L} . Diremos que el problema de decisión para una lógica \mathcal{L} es recursivamente resoluble si $T_{\mathcal{L}}$ es recursivo. De otra forma, diremos que el problema de decisión de \mathcal{L} es recursivamente irresoluble.*

En la década de los años 40 Emil Post introdujo los llamados *sistemas combinatorios* o *sistemas semi-Thue* y demostró que, dado cualquier conjunto recursivamente enumerable A , es posible construir una lógica \mathcal{L} cuyo conjunto engendrado $T_{\mathcal{L}}$ coincide con A . Como consecuencia de lo anterior, podemos enunciar el siguiente resultado.

Teorema 1 *Existe una lógica \mathcal{L} cuyo problema de decisión es recursivamente irresoluble.*

Demarcación: Los detalles de la construcción de esta lógica los veremos en el teorema 11. Aquí solamente nos adelantamos comentando que la lógica \mathcal{L} obtenida corresponderá a un *sistema semi-Thue* asociado con el conjunto K de Gödel,

$$K := \{x : \varphi_x^{(1)}(x) \downarrow\},$$

conjunto que es bien sabido que es recursivamente enumerable pero no es recursivo. ■

4 Lógicas de primer orden

El alfabeto de una *lógica de primer orden* \mathcal{F} contiene a los 7 símbolos²

$$- \quad \supset \quad [\quad] \quad , \quad (\quad)$$

También en el alfabeto habrá una infinidad numerable de símbolos llamados *variables individuales*, las cuales las escribiremos como

$$x, y, z, x_1, y_1, z_1, x_2, y_2, z_2, \dots$$

Los símbolos restantes son las *constantes individuales*, los *símbolos de predicados* y los *símbolos de funciones*. Cada símbolo de predicado o de función está asociado con un entero mayor o igual a 1, llamado el *grado* del símbolo en cuestión, que corresponde al número de argumentos del mismo. Por *símbolo individual* entenderemos o bien una variable individual o bien una constante individual.

Por lo general nuestro desarrollo será elaborado en términos de un alfabeto fijo y numerable a_0, a_1, a_2, \dots Supondremos además que los conjuntos correspondientes a las variables individuales, las constantes individuales, los símbolos

²En este trabajo emplearemos el símbolo “ \supset ” para denotar al símbolo condicional (“truth-functional condicional”), denotado en otros contextos con el símbolo “ \rightarrow ” o con el símbolo “ \Rightarrow ”. Lo anterior debido a utilizaremos estas últimas notaciones de flechas en los sistemas semi-Thue.

de predicados y los símbolos de funciones son todos recursivos. Supondremos además que es posible determinar de manera recursiva cuál es el grado de un predicado o de una función.

Decimos que una palabra X es un *término* de una lógica de primer orden \mathcal{F} si existe una secuencia de palabras X_1, \dots, X_n tal que X_n es X y, para cada $i \in \{1, \dots, n\}$ se cumple una de las siguientes condiciones:

- (a) X_i es un símbolo individual, o bien
- (b) X_i es $a(X_{j_1}, \dots, X_{j_m})$, donde $j_1, \dots, j_m < i$ y a es un símbolo de función de grado m .

Decimos que una palabra W es una *fórmula bien formada* (que abreviaremos por f.b.f.) de una lógica de primer orden \mathcal{F} , si existe una secuencia de palabras W_1, \dots, W_n tal que W_n es W y, para cada $i \in \{1, \dots, n\}$ se cumple una de las siguientes condiciones:

- (c) W_i es $p(X_1, \dots, X_m)$, donde X_1, \dots, X_m son términos y p es un símbolo de predicado de grado m , o bien
- (d) W_i es $[W_j \supset W_k]$, donde $j, k < i$, o bien
- (e) W_i es $\neg W_j$, donde $j < i$, o bien
- (f) W_i es $(v)W_j$, donde $j < i$ y v es una variable individual.

En el caso de un símbolo de función o de un predicado de grado 2, usualmente escribiremos (XaY) en vez de $a(X, Y)$.

Una lógica de primer orden tiene dos *reglas de inferencia*, a saber:

- **MODUS PONENS:** Esta regla de inferencia está gobernada por el predicado de 3 variables $\mathcal{R}(Y, X_1, X_2)$, el cual es verdadero si y solo si X_2 es $[X_1 \supset Y]$.³
- **GENERALIZACIÓN:** Esta regla de inferencia está gobernada por el predicado $\Upsilon(Y, X)$, el cual es verdadero si y solo si Y es $(v)X$, donde v es una variable individual.⁴

³Intuitivamente, la regla de inferencia del *modus ponens* establece que si tenemos el teorema $\vdash_{\mathcal{F}} [X_1 \supset Y]$ y además tenemos $\vdash_{\mathcal{F}} X_1$, entonces también tendremos $\vdash_{\mathcal{F}} Y$.

⁴Intuitivamente, la regla de inferencia de *generalización* establece que si tenemos el teorema $\vdash_{\mathcal{F}} X$ y v es una variable individual, entonces también tendremos el teorema $\vdash_{\mathcal{F}} (v)X$.

No es difícil verificar que estas reglas de inferencia son predicados recursivos de palabras. Decimos que una ocurrencia de una variable individual v en una f.b.f. W es una *ocurrencia acotada* si se trata de una parte bien formada de W de la forma $(v)A$. Una ocurrencia de v en una f.b.f. W que no es acotada es llamada *ocurrencia libre*. Una f.b.f. en la cual ninguna variable individual tenga alguna ocurrencia libre se dice que es *cerrada*. Escribimos f.b.f.c. para las f.b.f. cerradas.

Si W es una f.b.f., v es una variable individual y X es un término, entonces definimos el *predicado de sustitución* $\mathcal{S}(W, v, X)$ como sigue: $\mathcal{S}(W, v, X)$ es el resultado de reemplazar v por X en todas las ocurrencias de v en W . Excepción se hace cuando v no es libre en X , en cuyo caso $\mathcal{S}(W, v, X)$ arroja como resultado W .

Los axiomas de una lógica de primer orden \mathcal{F} incluyen a todas las palabras obtenidas al reemplazar v por una variable individual, X por un término, y A , B y C por fórmulas bien formadas en los siguientes cinco esquemas:

- (E1) $[A \supset [B \supset A]].$
- (E2) $[[A \supset [B \supset C]] \supset [[A \supset B] \supset [A \supset C]]].$
- (E3) $[[\neg B \supset \neg A] \supset [A \supset B]].$
- (E4) $[(v)A \supset \mathcal{S}(A, v, X)].$
- (E5) $[(v)[A \supset B] \supset [A \supset (v)B]],$ si v no tiene ocurrencias libres en A .

A modo de ejemplo del contexto en que estamos trabajando, analizamos a continuación un primer resultado elemental, válido en cualquier lógica de primer orden.

Teorema 2 *Sea \mathcal{F} una lógica de primer orden y sea A una fórmula bien formada de \mathcal{F} . Entonces, $\vdash_{\mathcal{F}} [A \supset A]$.*

Demuestra: Al reemplazar B por $[A \supset A]$ y C por A en el esquema básico (E2), obtenemos

$$\vdash_{\mathcal{F}} [[A \supset [[A \supset A] \supset A]] \supset [[A \supset [A \supset A]] \supset [A \supset A]]].$$

Por otra parte, del esquema básico (E1) obtenemos los siguientes dos teoremas triviales:

$$\vdash_{\mathcal{F}} [A \supset [[A \supset A] \supset A]] \quad , \quad \vdash_{\mathcal{F}} [A \supset [A \supset A]].$$

El resultado entonces se obtiene de aplicar la regla del *modus ponens* dos veces.

■

5 Lógicas especializadas de primer orden

Adicionalmente a los 5 esquemas básicos E1-E5 generadores de axiomas, una lógica de primer orden \mathcal{F} contiene un número *finito* (posiblemente ninguno) de axiomas adicionales, todos los cuales son fórmulas bien formadas cerradas; estos son llamados los *axiomas especializados* de la lógica.

Definición 6 *Sea \mathcal{F} una lógica de primer orden y sean A_1, A_2, \dots, A_n fórmulas bien formadas cerradas de \mathcal{F} . Entonces, $\mathcal{F}(A_1, \dots, A_n)$ es la lógica de primer orden obtenida a partir de \mathcal{F} al agregarle como axiomas especializados aquellas fórmulas A_1, \dots, A_n que no sean axiomas de \mathcal{F} .*

A continuación se establece la relación fundamental entre los teoremas de una lógica especializada y sus correspondientes versiones en la lógica que le dio origen.

Teorema 3 *Sea \mathcal{F} una lógica de primer orden y sea A una fórmula bien formada cerrada de \mathcal{F} . Entonces, $\vdash_{\mathcal{F}(A)} W$ si y solo si $\vdash_{\mathcal{F}} [A \supset W]$.*

Demuestra: Supongamos que $\vdash_{\mathcal{F}} [A \supset W]$. Entonces, tendremos $\vdash_{\mathcal{F}(A)} [A \supset W]$. Luego, en virtud que $\vdash_{\mathcal{F}(A)} A$, aplicando la regla del *modus ponens* obtenemos $\vdash_{\mathcal{F}(A)} W$.

Recíprocamente, supongamos que $\vdash_{\mathcal{F}(A)} W$. Sea W_1, \dots, W_n una prueba de W en $\mathcal{F}(A)$, de manera que W_n es W . Mostraremos que, para cada $i \in \{1, \dots, n\}$, tendremos $\vdash_{\mathcal{F}(A)} [A \supset W_i]$. Para ello supongamos, como hipótesis inductiva, que este resultado se sabe cierto para todo $j < i$. Distinguiremos varios casos:

Caso I: Cuando W_i es A . Entonces, $[A \supset W_i]$ es $[A \supset A]$, que en virtud del teorema 2 se trata de un teorema de \mathcal{F} .

Caso II: Cuando W_i es un axioma de \mathcal{F} . Entonces, $\vdash_{\mathcal{F}} W_i$. Empleando el esquema (E1) obtenemos $\vdash_{\mathcal{F}} [W_i \supset [A \supset W_i]]$. Por la regla del *modus ponens*, tendremos $\vdash_{\mathcal{F}} [A \supset W_i]$.

Caso III: Cuando W_j es $[W_k \supset W_i]$, para $j, k < i$. Luego, por la hipótesis inductiva, tendremos

$$\vdash_{\mathcal{F}} [A \supset W_k] \quad , \quad \vdash_{\mathcal{F}} [A \supset [W_k \supset W_i]].$$

Del esquema (E2), además tendremos

$$\vdash_{\mathcal{F}} [[A \supset [W_k \supset W_i]] \supset [[A \supset W_k] \supset [A \supset W_i]]].$$

Entonces, aplicando la regla del *modus ponens* dos veces, obtenemos finalmente $\vdash_{\mathcal{F}} [A \supset W_i]$.

Caso IV: Cuando W_i es $(v)W_j$, para $j < i$, donde v es una variable individual. Aplicando la regla de generalización, obtenemos $\vdash_{\mathcal{F}} (v)[A \supset W_j]$. Como A es una f.b.f.c., el esquema (E5) es aplicable, obteniéndose

$$\vdash_{\mathcal{F}} [(v)[A \supset W_j] \supset [A \supset (v)W_j]].$$

Aplicamos la regla del *modus ponens* para obtener $\vdash_{\mathcal{F}} [A \supset (v)W_j]$, de donde finalmente se deduce $\vdash_{\mathcal{F}} [A \supset W_i]$. Esto completa la prueba. ■

La generalización al caso de n axiomas especializados es elemental y se presenta en el siguiente corolario.

Corolario 4 *Sea \mathcal{F} una lógica de primer orden y sean A_1, A_2, \dots, A_n fórmulas bien formadas cerradas de \mathcal{F} . Luego, $\vdash_{\mathcal{F}(A_1, A_2, \dots, A_n)} W$ si y solo si $\vdash_{\mathcal{F}} [A_1 \supset [A_2 \supset \dots \supset [A_n \supset W] \dots]]$.*

Demostración: Se aplica el teorema precedente n veces. ■

6 Traslabilidad entre lógicas formales

Uno de los mecanismos más útiles para poder comparar sistemas lógicos formales es el concepto de la *traslabilidad*, el cual se introduce a continuación.

Definición 7 *Sean \mathcal{F} y \mathcal{F}' dos lógicas. Decimos que \mathcal{F} es trasladable hacia \mathcal{F}' si existe una función de palabras f , recursiva y 1-1, tal que para toda palabra X :*

$$\vdash_{\mathcal{F}} X \iff \vdash_{\mathcal{F}'} f(X).$$

La importancia de este concepto se pone de manifiesto en los siguientes resultados.

Teorema 5 *Sea \mathcal{F} una lógica trasladable hacia \mathcal{F}' y suponga que el problema de decisión para \mathcal{F}' es recursivamente resoluble. Entonces, el problema de decisión para \mathcal{F} es también recursivamente resoluble.*

Demostración: De las hipótesis, tenemos que el conjunto $T_{\mathcal{F}'}$ es recursivo. Sea f la función recursiva y 1-1 de palabras tal que $\vdash_{\mathcal{F}} X$ si y solo si $\vdash_{\mathcal{F}'} f(X)$. Recordemos que la función $f^*(\varrho(X)) := \varrho(f(X))$ es también recursiva. Entonces,

$$T_{\mathcal{F}} = \{ x : f^*(x) \in T_{\mathcal{F}'} \}.$$

Llamando por χ y χ' las funciones características de $T_{\mathcal{F}}$ y $T_{\mathcal{F}'}$ respectivamente, obtenemos entonces que $\chi = \chi' \circ (f^*)$. Como por hipótesis tanto χ' como f^* son recursivas, lo anterior pone en evidencia que χ es recursiva. Luego el conjunto $T_{\mathcal{F}}$ es recursivo y por tanto \mathcal{F} es recursivamente resoluble. ■

Teorema 6 *Sea \mathcal{F} una lógica trasladable hacia \mathcal{F}' y supongamos que \mathcal{F}' es trasladable hacia \mathcal{F}'' . Entonces, \mathcal{F} es trasladable hacia \mathcal{F}'' .*

Demostración: Existen funciones recursivas y 1-1 de palabras f y g tales que

$$\begin{aligned}\vdash_{\mathcal{F}} X &\iff \vdash_{\mathcal{F}'} f(X), \\ \vdash_{\mathcal{F}'} X &\iff \vdash_{\mathcal{F}''} g(X).\end{aligned}$$

Sea $h = g \circ (f)$. Luego h es recursiva y 1-1. Además, para cada palabra X se cumple:

$$\begin{aligned}\vdash_{\mathcal{F}} X &\iff \vdash_{\mathcal{F}'} f(X) \\ &\iff \vdash_{\mathcal{F}''} g(f(X)) \iff \vdash_{\mathcal{F}''} h(X).\end{aligned}\blacksquare$$

Teorema 7 *Sea \mathcal{F} una lógica de primer orden y sean A_1, \dots, A_n fórmulas bien formadas cerradas de \mathcal{F} . Luego, $\mathcal{F}(A_1, \dots, A_n)$ es trasladable hacia \mathcal{F} .*

Demostración: En virtud del corolario 4, solamente es necesario demostrar que la función de palabras f dada por

$$f(W) = [A_1 \supset [A_2 \supset \dots \supset [A_n \supset W] \dots]]$$

es recursiva, asunto cuya demostración es rutinaria. ■

Definición 8 *Decimos que una lógica de primer orden \mathcal{F} es no-especializada si \mathcal{F} no tiene axiomas especializados.*

Teorema 8 *Toda lógica de primer orden es trasladable hacia una lógica de primer orden no-especializada.*

Demostración: Sea \mathcal{F} una lógica de primer orden y sea \mathcal{F}' la lógica de primer orden no-especializada obtenida de \mathcal{F} al suprimirle los axiomas especializados. Más aún, supongamos que los axiomas especializados de \mathcal{F} son A_1, \dots, A_n . Entonces, $\mathcal{F} = \mathcal{F}'(A_1, \dots, A_n)$. Luego, del teorema 7, tendremos que \mathcal{F} es trasladable hacia \mathcal{F}' . ■

7 La lógica canónica de primer orden \mathcal{F}_0

Introducimos ahora la lógica no-especializada de primer orden \mathcal{F}_0 , denominada *lógica canónica de primer orden*. \mathcal{F}_0 tiene una infinidad numerable de constantes individuales y, para cada entero $m \geq 1$, \mathcal{F}_0 tiene una infinidad numerable de símbolos de funciones y predicados de grado m . Concretamente, los siete símbolos

– \supset [] , ()

serán identificados con $a_0, a_1, a_2, a_3, a_4, a_5, a_6$, respectivamente. Además, para $n > 6$, a_n denotará una *variable individual* si $\sigma_2(n) = 0$ y $\sigma_1(n)$ es par⁵, mientras que a_n denotará una *constante individual* si $\sigma_2(n) = 0$ y $\sigma_1(n)$ es impar. Luego, las variables individuales

$x, y, z, x_1, y_1, z_1, x_2, y_2, z_2, \dots$

son identificadas por $a_{\pi(4,0)}, a_{\pi(6,0)}, a_{\pi(8,0)}, \dots$, etc. Finalmente, para $n > 6$, con $\sigma_2(n) > 0$, tendremos que cuando $\sigma_2(n)$ es par entonces a_n denotará el *símbolo de predicado* de grado $\frac{1}{2}\sigma_2(n)$, mientras que cuando $\sigma_2(n)$ es impar, entonces a_n denotará el *símbolo de función* de grado $\frac{1}{2}(\sigma_2(n) + 1)$.

\mathcal{F}_0 es esencialmente la lógica denominada “*Engere Prädikatenkalkül*” por Hilbert y Ackermann, mientras que Church la denominaba el “*cálculo funcional puro de primer orden*”. La importancia de \mathcal{F}_0 proviene del siguiente hecho.

Teorema 9 *Toda lógica de primer orden no-especializada es trasladable hacia \mathcal{F}_0 .*

Demostración: Sea \mathcal{F} una lógica de primer orden no-especializada. Definimos la función recursiva φ como sigue. Para cada n , buscamos el papel que juega el símbolo a_n dentro de \mathcal{F} :

- Si a_n es uno de los siete símbolos “–”, “ \supset ”, “[”, “ ”, “ ”, “(”, “ ”)” de \mathcal{F} , entonces $\varphi(n)$ tendrá el valor 0, 1, 2, 3, 4, 5 o 6, respectivamente.
- Si $\varphi(x)$ está definida para $x < n$ y a_n es una *variable individual* en \mathcal{F} , o una *constante individual* en \mathcal{F} , o un *símbolo de predicado* de grado m en \mathcal{F} , o un *símbolo de función* de grado m en \mathcal{F} , entonces definimos $\varphi(n)$ como el menor entero p tal que $\varphi(x) \neq p$, para $x < n$ y a_p es una *variable individual* en \mathcal{F}_0 , o una *constante individual* en \mathcal{F}_0 , o un *símbolo de*

⁵Aquí $\pi: \mathbb{N}^2 \rightarrow \mathbb{N}$ es la función codificadora de Cantor dada por $\pi(x, y) = x + (x + y)(x + y + 1)/2$, mientras que σ_1 y σ_2 son las funciones decodificadoras de Cantor, caracterizadas por la identidad $n = \pi(\sigma_1(n), \sigma_2(n))$, para todo $n \in \mathbb{N}$. Tanto π como σ_1 y σ_2 son funciones primitivo recursivas.

	0	1	2	3	4	5	6	7	...
0	a_0	a_1	a_3	a_6	$P_0^{(2)}$	$f_0^{(3)}$	$P_0^{(3)}$	\dots	
1	a_2	a_4	$P_0^{(1)}$	$f_0^{(2)}$	$P_1^{(2)}$	$f_1^{(3)}$	$P_1^{(3)}$	\dots	
2	a_5	$f_0^{(1)}$	$P_1^{(1)}$	$f_1^{(2)}$	$P_2^{(2)}$	$f_2^{(3)}$	$P_2^{(3)}$	\dots	
3	c_0	$f_1^{(2)}$	$P_2^{(1)}$	$f_2^{(2)}$	$P_3^{(2)}$	$f_3^{(3)}$	$P_3^{(3)}$	\dots	
4	x_0	$f_2^{(1)}$	$P_3^{(1)}$	$f_3^{(2)}$	$P_4^{(2)}$	\vdots	\vdots		
5	c_1	$f_3^{(1)}$	$P_4^{(1)}$	$f_4^{(2)}$	$P_5^{(2)}$				
6	y_0	$f_4^{(1)}$	$P_5^{(1)}$	$f_5^{(2)}$	$P_6^{(2)}$				
7	c_2	\vdots	\vdots	\vdots	\vdots				
8	z_0								
9	\vdots								

Figura 1: Radiografía de \mathcal{F}_0 . Aquí $P_i^{(n)}$ es el *el predicado i-ésimo de grado n*, $f_i^{(n)}$ es la *función i-ésima de grado n*, c_i es la *i-ésima constante*, x_i , y_i , z_i son las *variables i-ésimas* ($i = 0, 1, 2, \dots$), $a_0 = “-”$, $a_1 = “\supset”$, $a_2 = “[”$, $a_3 = “]”$, $a_4 = “,”$, $a_5 = “(”$, $a_6 = “)”$.

*predicado de grado m en \mathcal{F}_0 , o un *símbolo de función* de grado m en \mathcal{F}_0 , respectivamente.*

Ahora, definimos la función de palabras $f(X)$ mediante

$$f(a_{r_1} a_{r_2} \cdots a_{r_k}) = a_{\varphi(r_1)} a_{\varphi(r_2)} \cdots a_{\varphi(r_k)}.$$

Es claro que f es recursiva y 1-1 y además satisface la condición de traslabilidad:

$$\vdash_{\mathcal{F}} X \iff \vdash_{\mathcal{F}_0} f(X). \blacksquare$$

Teorema 10 *Si el problema de decisión para \mathcal{F}_0 es recursivamente resoluble, entonces también lo es el de cualquier lógica de primer orden.*

Demostración: Inmediato, de los teoremas 5, 6, 8 y 9. \blacksquare

El teorema precedente era ya bien conocido (necesariamente a nivel informal) por Hilbert y su Escuela Formalista aún antes del desarrollo formal de la teoría de las funciones recursivas. Sobre la base de este teorema, Hilbert declaró

que el problema de decisión para \mathcal{F}_0 (usualmente referido simplemente como el *Entscheidungsproblem*) era el problema central de la lógica matemática. Subsecuentemente se realizó mucha investigación dirigida a buscar una solución positiva al *Entscheidungsproblem*. En parte, estos esfuerzos tomaron la forma de buscar reducciones del problema de decisión para \mathcal{F}_0 al correspondiente problema de decisión para clases especiales de fórmulas bien formadas, como por ejemplo, la clase de las fórmulas bien formadas del tipo⁶

$$(\exists x_1)(\exists x_2)(\exists x_3)(y_1)(y_2) \cdots (y_n)A,$$

donde A es un término libre de cuantificadores. Otros trabajos consistieron directamente en demostrar que, para clases de fórmulas bien formadas cada vez mayores, el problema de decisión podía ser resuelto. En particular, se llegó a demostrar que el problema de decisión puede ser resuelto para la clase de todas las fórmulas bien formadas del tipo

$$(\exists x_1)(\exists x_2)(y_1)(y_2) \cdots (y_n)A,$$

donde A es un término libre de cuantificadores. Estos resultados preliminares alimentaron por un tiempo la conjetura errónea que el *Entscheidungsproblem* era recursivamente resoluble. Parecía además que el menor avance en cualquier dirección podría resolver el problema. No obstante, en 1936 Church⁷ y Turing⁸ demostraron al mismo tiempo (aunque de manera independiente y con métodos muy diferentes el uno del otro) que el problema de decisión para \mathcal{F}_0 era recursivamente irresoluble.

8 Los sistemas combinatorios semi-Thue

El término *semi-Thue* fue introducido por Emil Post a principios de la década de los años 40, cuando introdujo ciertos sistemas combinatorios de generación de producciones que los bautizó con este nombre, en honor al matemático noruego

⁶Aquí, por $(\exists x_i)W$ entenderemos $-(x_i) - W$.

⁷El artículo de Church se titulaba “*A Note on the Entscheidungsproblem*”, publicado en la revista *The Journal of Symbolic Logic*, vol. 1, pp. 40–41, 1936. Church empleó su instrumental de λ -cálculo.

⁸El artículo de Turing se titulaba “*On Computable Numbers, with an Application to the Entscheidungsproblem*”, publicado en los *Proceedings of the London Mathematical Society*, serie 2, vol. 42, pp. 230–265, 1936. Fue en este famoso artículo que Turing introdujo por primera vez el concepto de las máquinas que ahora llevan su nombre. Por aquel entonces, Turing era apenas un muchacho que acababa de terminar sus estudios de matemática a nivel de pregrado. Church quedó tan bien impresionado con la metodología empleada por Turing, que lo invitó a realizar estudios doctorales en Princeton bajo su tutela, cosa que Turing aceptó gustoso.

Thue, quien había introducido algunas de las nociones preliminares al principio del siglo XX.

Definición 9 *Un sistema formal semi-Thue es un par ordenado (T, Λ) , donde:*

- (a) *T es un conjunto finito de pares ordenados (α, β) , con α y β hileras de símbolos de un alfabeto finito Σ , llamado el alfabeto del sistema. Los pares ordenados $(\alpha, \beta) \in T$ son llamados las producciones del sistema y alternativamente se acostumbra escribirlos en la forma “ $\alpha \rightarrow \beta$ ”.*
- (b) *Λ es una hilera no nula y fija sobre Σ . A Λ se le llama el axioma inicial del sistema.*

La interpretación de la producción “ $\alpha \rightarrow \beta$ ” es como sigue: sean δ y γ hileras de Σ (posiblemente hileras nulas). Cuando $(\alpha, \beta) \in T$, entonces escribimos

$$\delta \alpha \gamma \xrightarrow[T]{*} \delta \beta \gamma$$

y decimos que la hilera $\delta \beta \gamma$ es *directamente derivable* de la hilera $\delta \alpha \gamma$. Sean w_1 y w_2 dos hileras de Σ . Decimos que w_2 es *derivable* de w_1 y escribimos

$$w_1 \xrightarrow[T]{*} w_2$$

si $w_1 \xrightarrow[T]{*} w_2$, o bien existe una secuencia de hileras z_1, z_2, \dots, z_n sobre Σ tales que

$$w_1 \xrightarrow[T]{*} z_1 \xrightarrow[T]{*} z_2 \xrightarrow[T]{*} \cdots \xrightarrow[T]{*} z_n \xrightarrow[T]{*} w_2.$$

Por ejemplo, consideremos el sistema semi-Thue que resulta de tomar el alfabeto $\Sigma = \{a, b\}$ y T compuesto por las siguientes dos producciones: $ab \rightarrow bbb$, $bb \rightarrow \lambda$, donde λ denota la hilera nula (esto es, $T = \{(ab, bbb), (bb, \lambda)\}$), con cualquier axioma inicial Λ . Entonces, fácilmente se comprueban los siguientes hechos:

1. $abbab \xrightarrow[T]{*} bbbbbbb$,
2. $babb \xrightarrow[T]{*} b$,
3. Es falso que $abbabbb \xrightarrow[T]{*} bbbb$.

Definición 10 *Definimos el lenguaje generado por el sistema semi-Thue (T, Λ) , el cual denotaremos por $\mathcal{L}(T, \Lambda)$, mediante*

$$\mathcal{L}(T, \Lambda) = \left\{ w \in \Sigma^* : \Lambda \xrightarrow[T]{*} w \right\}.$$

Puede observarse que cada sistema semi-Thue (T, Λ) sobre un alfabeto Σ define a la vez un sistema lógico formal $\mathcal{L} = \mathcal{L}(T, \Lambda)$, en el cual Λ es el único axioma de \mathcal{L} y las reglas de inferencia son precisamente las producciones de T . Bajo esta asociación, para cualquier hilera no nula w de Σ , son equivalentes las frases

$$w \in \mathcal{L}(T, \Lambda) \iff \vdash_{\mathcal{L}} w.$$

Consideremos el siguiente problema de decisión: dado un sistema semi-Thue (T, Λ) sobre Σ y dada w una hilera no nula de Σ , >será cierto que $w \in \mathcal{L}(T, \Lambda)$? Este problema de decisión es recursivamente irresoluble, como se constata a continuación.

Teorema 11 (Post) *Existe un sistema semi-Thue (T, Λ) tal que el problema de decisión $>w \in \mathcal{L}(T, \Lambda)$? es recursivamente irresoluble.*

Demuestra: (Basada en Davis [3]) Sea g una función recursiva cualquiera y sea M_g una máquina de Turing que evalúa a g en la forma estándar. Supongamos que los estados de M_g son q_0, q_1, \dots, q_r y que M_g se detiene (cuando lo hace) solamente en el estado q_r . Sean $s_0 = 0, s_1 = 1, s_2, \dots, s_k$ los símbolos de M_g . Sea $m \in \mathbb{N}$. Construimos el sistema semi-Thue (T_{M_g}, Λ_m) de la siguiente manera:⁹

- a. **Alfabeto:** $\Sigma = \{q_0, \dots, q_r, s_0, \dots, s_k, \#\}$.
- b. **Axioma Λ_m :** $\#q_0\bar{m}\#$, donde $\bar{m} = 11 \dots 1$ ($m + 1$ veces).¹⁰
- c. **Producciones de T_{M_g} :**
 - Por cada quíntuplo¹¹ de M_g de la forma (q, s, s', R, q') , tendremos las siguientes dos producciones:

$$(qs, s'q') \in T_{M_g} \quad , \quad (q\#, s'q'\#) \in T_{M_g}.$$

⁹El sistema semi-Thue (T_{M_g}, Λ_m) que estamos construyendo se encargará de imitar “en papel y lápiz” el comportamiento de la máquina de Turing M_g . El símbolo adicional “#” denotará el principio o final de la cinta de M_g , que como se recordará es potencialmente infinita en ambas direcciones.

¹⁰La interpretación de este axioma es simplemente poner a la máquina de Turing M_g a calcular, empezando en el estado q_0 y frente al entero m .

¹¹El esquema empleado aquí para denotar los quíntuplos de las máquinas de Turing es el siguiente: los quíntuplos son de la forma (q, s, s', D, q') , donde q es el estado actual de M_g , s es el símbolo leído de la cinta, s' es el símbolo que escribe M_g en la cinta en reemplazo de s , D es la dirección de la cinta hacia la cual se mueve un cuadro la cabeza de M_g (“R” de “right” o “L” de “left”), y finalmente q' es el nuevo estado en que entra M_g luego de realizar esta acción.

- Por cada quíntuplo de M_g de la forma (q, s, s', L, q') y cada símbolo u , tendremos la producción:

$$(uqs, q'us') \in T_{M_g}.$$

- Por cada quíntuplo de la forma $(q, 0, x' \neq 0, L, q')$ y cada símbolo u , tendremos la producción:

$$(uq\#, q'us'\#) \in T_{M_g}.$$

Con esta construcción fácilmente se comprueba que

$$\#q_r0\bar{u}\# \in \mathcal{L}(T_{M_g}, \Lambda_m) \iff g(m) \downarrow \text{ y } g(m) = u.$$

Tomemos ahora la función g dada por

$$g(m) = \begin{cases} 0 & , \text{ si } m \in K \\ \uparrow & , \text{ si } m \notin K, \end{cases}$$

donde K es el conjunto de Gödel (el cual es recursivamente enumerable pero no recursivo) y consideremos la máquina M_g correspondiente, que evalúa a g con los requisitos señalados. Obtenemos un sistema semi-Thue (T_{M_g}, Λ_m) tal que

$$\#q_r01\# \in \mathcal{L}(T_{M_g}, \Gamma_m) \iff m \in K.$$

Tomemos ahora el inverso del sistema (T_{M_g}, Λ_m) , que consiste en el sistema semi-Thue (T, Λ) dado por:

1. **Alfabeto:** el mismo Σ .
2. **Axioma:** $\Lambda = \#q_r01\#$.
3. **Producciones:** $(\alpha, \beta) \in T_{M_g} \iff (\beta, \alpha) \in T$.

Obtenemos entonces: $\#q_00\bar{m}\# \in \mathcal{L}(T, \Lambda) \iff m \in K$. Por consiguiente, el problema de decisión para el sistema semi-Thue (T, Λ) es indecidible, puesto que el problema de decisión $\{m \in K\}$ es indecidible, en virtud que K no es recursivo. ■

Lo anterior justifica la existencia de sistemas lógicos \mathcal{L} con problemas de decisión recursivamente indecidibles, asunto que habíamos dejado pendiente desde el teorema 1.

9 Solución negativa al ‘‘Entscheidungsproblem’’

Teorema 12 *Todo sistema semi-Thue σ es trasladable hacia una lógica de primer orden \mathcal{F}_σ .*

Demostración: Postergamos la demostración de este resultado unas líneas, con el fin de no perder el hilo de los siguientes acontecimientos. ■

Corolario 13 *Existe una lógica de primer orden cuyo problema de decisión es recursivamente irresoluble.*

Demostración: Se trata de una consecuencia del teorema precedente, en conjunto con los teoremas 5 y 11. ■

Teorema 14 (Church, Turing) *El problema de decisión para \mathcal{F}_0 es recursivamente irresoluble.*

Demostración: Se trata de una consecuencia del teorema 10 y del corolario precedente. ■

En lo que resta de esta sección nos dedicaremos a demostrar formalmente el teorema 12.

Demostración del teorema 12: Haremos una adaptación de la construcción realizada por Davis [3]. Sea $\sigma = (T, \Lambda)$ un sistema semi-Thue sobre el alfabeto finito Σ , con reglas de producción $T = \{\alpha_i \rightarrow \beta_i : 1 \leq i \leq m\}$ y axioma Λ . Vamos a construir la lógica de primer orden \mathcal{F}_σ asociada a σ , de la siguiente manera:

CONSTANTES INDIVIDUALES DE \mathcal{F}_σ : las letras del alfabeto Σ .

SÍMBOLO DE FUNCIÓN DE \mathcal{F}_σ : \star , de grado 2 (el cual imitará en \mathcal{F}_σ la operación de la concatenación en σ)

SÍMBOLOS DE PREDICADOS DE \mathcal{F}_σ : P , de grado 1 y “ \equiv ”, de grado 2.

Para cada palabra W sobre el alfabeto Σ , asociamos el *término* W' de \mathcal{F}_σ en forma recursiva de la siguiente manera:

$$\begin{aligned} a' &= a, \quad \forall a \in \Sigma, \\ (Wa)' &= (W' \star a'), \quad \forall a \in \Sigma. \end{aligned}$$

Definimos la función de palabras f mediante $f(W) = P(W')$. Claramente la función f es recursiva. Los axiomas especializados de \mathcal{F}_σ serán escogidos de tal manera que $\vdash_\sigma W$ si y solo si $\vdash_{\mathcal{F}_\sigma} f(W)$, como sigue.

AXIOMAS ESPECIALIZADOS DE \mathcal{F}_σ :

- (A1) $(x_1)(x_1 \equiv x_1)$.
- (A2) $(x_1)(x_2)[(x_1 \equiv x_2) \supset (x_2 \equiv x_1)]$.
- (A3) $(x_1)(x_2)(x_3)[(x_1 \equiv x_2) \supset [(x_2 \equiv x_3) \supset (x_1 \equiv x_3)]]$.
- (A4) $(x_1)(x_2)[(x_1 \equiv x_2) \supset [(P(x_1) \supset P(x_2))]]$.
- (A5) $(x_1)(x_2)(x_3)[(x_1 \equiv x_2) \supset ((x_1 \star x_3) \equiv (x_2 \star x_3))]$.
- (A6) $(x_1)(x_2)(x_3)[(x_1 \equiv x_2) \supset ((x_3 \star x_1) \equiv (x_3 \star x_2))]$.
- (A7) $(x_1)(x_2)(x_3)((x_1 \star (x_2 \star x_3)) \equiv ((x_1 \star x_2) \star x_3))$.
- (A8) $P(\Lambda')$.
- (A9) $(x_1)(x_2)[P(((x_1 \star \alpha'_i) \star x_2)) \supset P(((x_1 \star \beta'_i) \star x_2))]$, para cada $i \in \{1, \dots, m\}$.

Vamos a demostrar que $\vdash_\sigma W$ si y solo si $\vdash_{\mathcal{F}_\sigma} P(W')$. Un término de \mathcal{F}_σ se dice ser una *constante* si no contiene variables individuales. Si X es una constante, escribimos $\{X\}$ para denotar la palabra obtenida de X al remover todos los paréntesis y las estrellas \star . Por ejemplo, $\{W'\} = W$. Decimos que dos constantes X, Y son *asociadas* si $\{X\} = \{Y\}$. Luego, tendremos los siguientes lemas.

Lema 15 *Si X y Y son constantes, entonces $\vdash_{\mathcal{F}_\sigma} (X \equiv Y)$ si y solo si X y Y son asociadas.*

Demarcación: Si X y Y son asociadas, entonces $\{X\} = \{Y\}$, de manera que $\{X\}$ y $\{Y\}$ tienen el mismo largo. Si este largo es 1 o 2, el lema es obvio. Si el largo es 3, luego $\{X\} = \{Y\} = acb$, con $a, b, c \in \Sigma$. Entonces, las únicas posibilidades para X, Y son $(a \star (b \star c))$, $((a \star b) \star c)$. Ahora, aplicando el axioma especializado (A7), tendremos

$$\vdash_{\mathcal{F}_\sigma} (x_1)(x_2)(x_3)((x_1 \star (x_2 \star x_3)) \equiv ((x_1 \star x_2) \star x_3)).$$

Por otra parte, en virtud del esquema de axiomas (E4) para lógicas de primer orden, tendremos además los siguientes tres teoremas de \mathcal{F}_σ :

$$\begin{aligned} \vdash_{\mathcal{F}_\sigma} & [(x_1)(x_2)(x_3)((x_1 \star (x_2 \star x_3)) \equiv ((x_1 \star x_2) \star x_3)) \\ & \quad \supset (x_2)(x_3)((a \star (x_2 \star x_3)) \equiv ((a \star x_2) \star x_3))], \\ \vdash_{\mathcal{F}_\sigma} & [(x_2)(x_3)((a \star (x_2 \star x_3)) \equiv ((a \star x_2) \star x_3)) \\ & \quad \supset (x_3)((a \star (b \star x_3)) \equiv ((a \star b) \star x_3))], \\ \vdash_{\mathcal{F}_\sigma} & [(x_3)((a \star (b \star x_3)) \equiv ((a \star b) \star x_3)) \supset ((a \star (b \star c)) \equiv ((a \star b) \star c))]. \end{aligned}$$

Aplicando la regla del *modus ponens* tres veces, obtenemos

$$\vdash_{\mathcal{F}_\sigma} ((a \star (b \star c)) \equiv ((a \star b) \star c)),$$

esto es, $\vdash_{\mathcal{F}_\sigma} (X \equiv Y)$. El resultado para largos mayores que 3 se demuestra usando inducción matemática, de manera similar. Finalmente, el recíproco se obtiene de la observación que los axiomas especializados (A1) y (A7) son los únicos que producen abiertamente equivalencias “ \equiv ”, las cuales ciertamente no pueden ser engendradas por constantes no-asociadas. Por otra parte, los axiomas del (A2) al (A6) producen equivalencias entre asociadas a partir de otras equivalencias entre asociadas. ■

Lema 16 *Si $\vdash_\sigma W$, entonces $\vdash_{\mathcal{F}_\sigma} P(W')$.*

Demarcación: Sea W_1, W_2, \dots, W_n una prueba de W en σ , donde W_n es W . Mostraremos que, para cada $i \in \{1, \dots, n\}$, tendremos $\vdash_{\mathcal{F}_\sigma} P(W'_i)$. Esto es claramente cierto para $i = 1$, pues W_1 es Λ . Supongamos que es cierto para $i = k$, con $k < n$. Luego, para algún índice j , tendremos $W_k = Q\alpha_j R$, $W_{k+1} = Q\beta_j R$. Del esquema de axiomas (E4) de las lógicas de primer orden, tendremos

$$\begin{aligned} \vdash_{\mathcal{F}_\sigma} & [(x_1)(x_2)[P(((x_1 \star \alpha'_j) \star x_2)) \supset P(((x_1 \star \beta'_j) \star x_2))] \\ & \quad \supset [P(((Q' \star \alpha'_j) \star R')) \supset P(((Q' \star \beta'_j) \star R'))]]. \end{aligned}$$

Esto, junto con el axioma (A9) y la regla del *modus ponens*, nos brinda

$$\vdash_{\mathcal{F}_\sigma} [P(((Q' \star \alpha'_j) \star R')) \supset P(((Q' \star \beta'_j) \star R'))].$$

Por otra parte, del lema 15, tenemos

$$\vdash_{\mathcal{F}_\sigma} ((Q\alpha_j R)' \equiv ((Q' \star \alpha'_j) \star R')).$$

Utilizando el axioma especializado (A4), obtenemos entonces

$$\vdash_{\mathcal{F}_\sigma} [P((Q\alpha_j R)') \supset P((Q' \star \alpha'_j) \star R')].$$

Ahora, aplicando nuestra hipótesis de inducción y la regla del *modus ponens* dos veces, obtenemos

$$\vdash_{\mathcal{F}_\sigma} P((Q' \star \beta'_j) \star R').$$

Finalmente, aplicando el lema 15, el axioma especializado (A4), el esquema de axiomas (E4) de las lógicas de primer orden y la regla de *modus ponens*, obtenemos

$$\vdash_{\mathcal{F}_\sigma} P((Q\beta_j R)'),$$

esto es, $\vdash_{\mathcal{F}_\sigma} P(W'_{k+1})$. ■

Lema 17 *Si X es una constante y $\vdash_\sigma \{X\}$, entonces $\vdash_{\mathcal{F}_\sigma} P(X)$.*

Demuestra: Si $X = \{X\}'$, el resultado es consecuencia inmediata del lema 16. En caso contrario, X es un asociado de $Y = \{X\}'$. Luego, de los lemas 15, 16 y la hipótesis, tendremos

$$\vdash_{\mathcal{F}_\sigma} (X \equiv Y) \quad , \quad \vdash_{\mathcal{F}_\sigma} P(Y).$$

Aplicando el axioma especializado (A2), obtenemos

$$\vdash_{\mathcal{F}_\sigma} (Y \equiv X).$$

Aplicando el axioma especializado (A4), obtenemos

$$\vdash_{\mathcal{F}_\sigma} [P(Y) \supset P(X)].$$

Finalmente, al aplicar la regla del *modus ponens* a esta última expresión, obtenemos el resultado $\vdash_{\mathcal{F}_\sigma} P(X)$. ■

Lema 18 *Si X es una constante y $\vdash_{\mathcal{F}_\sigma} P(X)$, entonces $\vdash_\sigma \{X\}$.*

Demuestra: El único axioma de \mathcal{F}_σ de la forma $P(X)$, donde X es una constante, tiene la propiedad deseada. Por otra parte, para derivar $P(X)$ en \mathcal{F}_σ , con X constante, deberán ser empleados los axiomas especializados (A4) o (A9). Pero, en virtud del lema 15, el axioma (A4) puede producir a $P(X)$ como teorema solamente si ya conocemos que $\vdash_{\mathcal{F}_\sigma} P(Y)$, donde Y y X son asociados. Por otra parte, el axioma (A9) puede producir a $P(X)$ como teorema solamente si $\{X\}$ es una consecuencia de $\{Y\}$ al aplicar una de las producciones de σ y si $\vdash_{\mathcal{F}_\sigma} P(Y)$. ■

Lema 19 $\vdash_\sigma W$ si y solo si $\vdash_{\mathcal{F}_\sigma} P(W')$.

Demuestra: Es clara, de los lemas 16 y 18. Esto completa la prueba del teorema 12. ■

10 Cálculos proposicionales parciales

El alfabeto de un *cálculo proposicional parcial* \mathcal{B} consiste en los cuatro símbolos

– \supset []

y en una infinitud numerable de símbolos $p_1, q_1, r_1, p_2, q_2, r_2, p_3, q_3, r_3, \dots$ llamados *variables proposicionales*.

Una palabra W se dice ser una *fórmula bien formada* (abreviado a veces por f.b.f.) de un cálculo proposicional parcial \mathcal{B} si existe una secuencia de palabras W_1, W_2, \dots, W_n tal que W_n es W y, para cada $i \in \{1, \dots, n\}$, se cumple alguna de las siguientes condiciones:

- (1) W_i es una variable proposicional, o bien
- (2) W_i es $[W_j \supset W_k]$, con $j, k < i$, o bien
- (3) W_i es $\neg W_j$, con $j < i$.

Un cálculo proposicional parcial \mathcal{B} tiene dos reglas de inferencia, a saber:

- *Modus ponens*: Esta regla de inferencia está gobernada por el predicado de tres variables $\mathcal{R}(Y, X_1, X_2)$, el cual es verdadero cuando y solo cuando X_2 es $[X_1 \supset Y]$.¹²
- *Sustitución*: Regla de inferencia está gobernada por el predicado cuaternario $\mathcal{S}(Y, X, p, W)$, el cual es verdadero cuando y solo cuando Y resulta de X al reemplazar la variable proposicional p en todas sus ocurrencias en X por la fórmula bien formada W .

Sea U un conjunto que contiene solamente dos objetos distintos. Estos objetos serán denotados por V y F y los llamaremos *valores de verdad*.¹³ Una función cuyo dominio consiste en un conjunto de n -é tuplos de valores de verdad, y cuyos valores son siempre V o F , se llama *función de verdad*.

Con cada fórmula bien formada W de un cálculo proposicional parcial asociaremos la función de verdad \bar{W} , como sigue:

¹²Como en otros sistemas lógicos, la regla del *modus ponens* establece que si tenemos el teorema $\vdash_{\mathcal{B}} [X_1 \supset Y]$ y además tenemos el teorema $\vdash_{\mathcal{B}} X_1$ (el *antecedente*), entonces se infiere el teorema $\vdash_{\mathcal{B}} Y$ (la *consecuencia*).

¹³En la interpretación del cálculo proposicional parcial, las letras V y F significan simplemente “Verdadero” y “Falso”.

- (1) Si W es una variable proposicional, entonces \widetilde{W} es la función identidad de una variable.
- (2) Si W es $[B \supset C]$, entonces \widetilde{W} es F solamente para aquellos argumentos que hacen $\widetilde{B} = V$ y $\widetilde{C} = F$. En los otros casos, $\widetilde{W} = V$.
- (3) Si W es $\neg B$, entonces \widetilde{W} es V cuando y solamente cuando \widetilde{B} es F .

Una fórmula bien formada W se dice ser una *tautología* si \widetilde{W} solamente toma el valor V . Inmediatamente tendremos:

Teorema 20 *En un cálculo proposicional parcial, la clase de las tautologías es recursiva.*

Demostración: Es fácil comprobar en forma algorítmica si una fórmula bien formada W es una tautología, utilizando por ejemplo una tabla de verdad. ■

Nuestra estipulación final concerniendo a los cálculos proposicionales parciales es que solamente pueden tener un número finito de axiomas, *cada uno de los cuales es una tautología*. Inmediatamente tendremos el siguiente resultado.

Teorema 21 *En un cálculo proposicional parcial, cada teorema es una tautología.*

Demostración: Los axiomas son tautologías. Además, es evidente que las dos reglas de inferencia (*modus ponens* y sustitución) preservan la propiedad de ser tautología. ■

Un cálculo proposicional parcial muy particular es \mathcal{B}_0 , denominado el *cálculo proposicional parcial canónico*, o a veces llamado simplemente el *cálculo proposicional*, cuyos axiomas son las siguientes fórmulas bien formadas:

- (C1) $[p_1 \supset [q_1 \supset p_1]].$
- (C2) $[[p_1 \supset [q_1 \supset r_1]] \supset [[p_1 \supset q_1] \supset [p_1 \supset r_1]]].$
- (C3) $[[\neg q_1 \supset \neg p_1] \supset [p_1 \supset q_1]].$

Definición 11 *Un cálculo proposicional parcial \mathcal{B} se dice ser completo si toda tautología de \mathcal{B} es un teorema de \mathcal{B} .*

Corolario 22 *Sea \mathcal{B} un cálculo proposicional parcial y completo. Entonces, el problema de decisión para \mathcal{B} es recursivamente resoluble.*

Demostración: Inmediato, de los teoremas 20 y 21. ■

Uno de los resultados fundamentales de la lógica moderna es el siguiente.

Teorema 23 (Kalmar) \mathcal{B}_0 es completo.

Demostración: No la veremos aquí. Una demostración corta y elegante de este resultado puede consultarse en la obra de Church “*Introduction to Mathematical Logic*” (ver [2]). ■

Estos resultados nos podrían conducir a la creencia errónea que, dentro del dominio de los cálculos proposicionales parciales, no existirá ningún problema de decisión recursivamente irresoluble. No obstante, Linial y Post fueron los primeros en demostrar que, por el contrario, sí existe un cálculo proposicional parcial cuyo problema de decisión es recursivamente irresoluble. Ese resultado se obtiene como una consecuencia del siguiente teorema.

Teorema 24 (Linial, Post) Todo sistema semi-Thue $\sigma = (T, \Lambda)$ con un alfabeto de dos letras es trasladable hacia un cálculo proposicional parcial.

Demostración: Seguiremos el enfoque de Davis (ver [3]), que a la vez es una adaptación de la prueba original de Linial y Post (ver [9]). Sea $\Sigma = \{1, b\}$ el alfabeto de σ , sea Λ el axioma de σ y sean $Q\alpha_i R \rightarrow Q\beta_i R$, $i = 1, 2, \dots, m$, las producciones de σ . Construiremos un cálculo proposicional parcial que “imita” el comportamiento del sistema semi-Thue σ . Empezamos definiendo:

$$\begin{aligned} 1' &= \text{--}[-p_1 \supset -p_1] \\ b' &= \text{----}[-p_1 \supset -p_1] \\ (W1)' &= [W' \& 1'] \\ (Wb)' &= [W' \& b'], \end{aligned}$$

donde por $[A \& B]$ entenderemos $-\neg[A \supset \neg B]$. Luego, si W es cualquier palabra sobre $\{1, b\}$, la fórmula bien formada W' es una tautología. Definimos el cálculo proposicional parcial \mathcal{B}_σ que tenga los siguientes seis axiomas:

- (B1) $[[p_1 \& q_1 \& r_1]] \supset [[p_1 \& q_1] \& r_1]]$.
- (B2) $[[[p_1 \& q_1] \& r_1] \supset [p_1 \& [q_1 \& r_1]]]$.
- (B3) $[[p_1 \supset q_1] \supset [[q_1 \supset p_1] \supset [[r_1 \& p_1] \supset [r_1 \& q_1]]]]$.
- (B4) $[[p_1 \supset q_1] \supset [[q_1 \supset p_1] \supset [[p_1 \& r_1] \supset [q_1 \& r_1]]]]$.

(B5) $\Lambda'; [p_1 \supset p_1]$.

(B6) $[[[p_1 \& \alpha_i] \& q_i] \supset [[p_1 \& \beta_i] \& q_1]]$, para $i = 1, 2, \dots, m$.

Sin mucha dificultad se verifica que cada uno de estos axiomas es una tautología. Diremos que una palabra X es *regular* si existe una secuencia finita de palabras X_1, X_2, \dots, X_n , donde $X_n = X$, tal que, para cada $i \in \{1, \dots, n\}$, o bien X_i es $1'$ o b' , o bien X_i es $[X_j \& X_k]$, para $j, k < i$. Luego, W' es siempre regular, como también lo es por ejemplo la palabra $[[1' \& b'] \& [b' \& 1']]$.

Si X es una palabra regular, definimos $\langle X \rangle$ como la palabra definida sobre $\{1, b\}$ obtenida al reemplazar primero todas las ocurrencias de b' en X por b , luego reemplazando todas las restantes ocurrencias de $1'$ por 1 y finalmente removiendo todas las ocurrencias de “[”, ”]” y “&”. Diremos que dos palabras regulares X y Y son *asociadas* si $\langle X \rangle = \langle Y \rangle$.

Como parte de la demostración del teorema tendremos los siguientes cinco lemas:

Lema 25 *Si X y Y son asociadas, entonces $\vdash_{\mathcal{B}_\sigma} [X \supset Y]$ y también $\vdash_{\mathcal{B}_\sigma} [Y \supset X]$.*

Demostración: Por inducción matemática sobre el número de símbolos, empleando los axiomas (B1) a (B4). ■

Lema 26 *Si W_2 es una consecuencia de W_1 por medio de una de las producciones de σ y $\vdash_{\mathcal{B}_\sigma} W'_1$, entonces $\vdash_{\mathcal{B}_\sigma} W'_2$.*

Demostración: Sean $W_1 = Q\alpha_i R$, $W_2 = Q\beta_i R$. Entonces, tendremos

$$\vdash_{\mathcal{B}_\sigma} (Q\alpha_i R)'.$$

En virtud del lema 25,

$$\vdash_{\mathcal{B}_\sigma} [(Q\alpha_i R)' \supset [[Q' \& \alpha'_i] \& R']]$$

Aplicando el axioma (B6), tendremos entonces

$$\vdash_{\mathcal{B}_\sigma} [[[Q' \& \alpha'_i] \& R'] \supset [[Q' \& \beta'_i] \& R']]$$

Aplicando el lema 25 otra vez, obtenemos

$$\vdash_{\mathcal{B}_\sigma} [[[Q' \& \beta'_i] \& R'] \supset (Q\beta_i R)']$$

Luego, aplicando la regla del *modus ponens* tres veces, obtenemos

$$\vdash_{\mathcal{B}_\sigma} (Q\beta_i R)',$$

esto es, $\vdash_{\mathcal{B}_\sigma} W'_2$. ■

Lema 27 Si $\vdash_{\sigma} W$, entonces $\vdash_{\mathcal{B}_{\sigma}} W'$.

Demuestra: Λ' es un axioma y luego también un teorema. El resultado entonces se sigue del lema 26. ■

Lema 28 Si X es regular y $\vdash_{\mathcal{B}_{\sigma}} X$, entonces $\vdash_{\sigma} \langle X \rangle$.

Demuestra: El único axioma de \mathcal{B}_{σ} que es regular es Λ' . Las aplicaciones de la regla de sustitución en los axiomas (B1) a (B4) y (B6), así como las subsecuentes aplicaciones de la regla del *modus ponens*, pueden brindar solamente palabras regulares a partir de palabras regulares. Los axiomas (B1) y (B2) pueden ir solamente de una palabra a una de sus asociadas. El axioma (B6) puede producir la palabra regular Y como teorema de \mathcal{B}_{σ} solamente si $\langle Y \rangle$ es una consecuencia de $\langle X \rangle$ por medio de una de las producciones de σ y además ya hemos tenido $\vdash_{\mathcal{B}_{\sigma}} X$. Finalmente, los axiomas (B3) y (B4) no alteran nada importante, pues en un sistema semi-Thue cuando R es una consecuencia de Q entonces SR es una consecuencia de SQ y RS es una consecuencia de QS . ■

Lema 29 $\vdash_{\sigma} W$ si y solo si $\vdash_{\mathcal{B}_{\sigma}} W'$.

Demuestra: Es una clara consecuencia de los lemas 27 y 28. ■

El lema 29, junto al hecho que la función $f(W) = W'$ es claramente recursiva, completa la demostración del teorema 24. ■

Teorema 30 (Linial, Post) Existe un cálculo proposicional parcial cuyo problema de decisión es recursivamente irresoluble.

Demuestra: Consecuencia directa de los teoremas 24, 5 y 11. ■

Sea σ_0 un sistema semi-Thue sobre un alfabeto finito Σ con problema de decisión recursivamente indecidible. Entonces, el cálculo proposicional parcial \mathcal{B}_{σ_0} construido a partir de σ_0 también tiene problema de decisión recursivamente indecidible. Sea W una palabra en Σ y considérese el cálculo proposicional parcial $\mathcal{R}(W)$ que se obtiene tomando los axiomas de \mathcal{B}_{σ_0} y adicionalmente los siguientes tres axiomas especializados:

$$(B7) [W' \supset [p_1 \supset [q_1 \supset p_1]]].$$

$$(B8) [W' \supset [[p_1 \supset [q_1 \supset r_1]] \supset [[p_1 \supset q_1] \supset [p_1 \supset r_1]]]].$$

$$(B9) [W' \supset [[-q_1 \supset -p_1] \supset [p_1 \supset q_1]]].$$

Tendremos entonces el siguiente resultado, conocido como la *complejión relativa de un cálculo proposicional parcial incompleto*, atribuido a Linial y Post.

Teorema 31 $\vdash_{\mathcal{B}_{\sigma_0}} W'$ *s y solo si* $\mathcal{R}(W)$ *es completo.*

Demarcación: Si $\vdash_{\mathcal{B}_{\sigma_0}} W'$, entonces $\vdash_{\mathcal{R}(W)} W'$. Luego, aplicando la regla del *modus ponens* a los axiomas (B7), (B8), (B9), tendremos que los axiomas de \mathcal{B}_0 son teoremas de $\mathcal{R}(W)$. Consecuentemente, todas las tautologías de \mathcal{B}_0 al ser teoremas de \mathcal{B}_0 , serán también teoremas de $\mathcal{R}(W)$.

Recíprocamente, supongamos que $\mathcal{R}(W)$ es completo. Entonces, en particular tendremos $\vdash_{\mathcal{R}(W)} W'$. Ahora, los axiomas (B7), (B8), (B9) son útiles como el antecedente de una aplicación de la regla del *modus ponens* en la prueba de W' . Si fuéramos a aplicar la regla del *modus ponens* a estos axiomas para derivar una fórmula más corta, entonces W' o bien el resultado de sustituir en W' deberán estar disponibles. Pero los axiomas (B1) al (B6) no arrojan un resultado proveniente de sustituir en W' excepto a través de la misma prueba de W' . Luego, W' puede ser demostrado sobre la base de los axiomas (B1) a (B6) solamente, esto es, $\vdash_{\mathcal{B}_{\sigma_0}} W'$. ■

La consecuencia de lo anterior es asombrosa: en general no es posible decidir mediante un procedimiento mecánico si un cálculo proposicional parcial es completo o no es completo. Efectivamente, tendremos:

Corolario 32 (Post) *El problema de determinar si un cálculo proposicional parcial \mathcal{B} es o no completo, es recursivamente irresoluble.*

Demarcación: Es una consecuencia inmediata del teorema 31. ■

Finalizamos este artículo analizando otro problema de decisión clásico, en relación a los cálculos proposicionales parciales: la cuestión de la independencia o no independencia de sus axiomas y el teorema de Linial-Post de 1949.

Definición 12 *Un cálculo proposicional parcial \mathcal{B} se dice ser independiente si ningún axioma A de \mathcal{B} es un teorema en el cálculo proposicional parcial \mathcal{B}' obtenido a partir de \mathcal{B} suprimiendo A de los axiomas de \mathcal{B} .*

Sea σ_0 un sistema semi-Thue como el anteriormente considerado (con problema de decisión recursivamente indecidible) sobre el alfabeto $\Sigma = \{1, b\}$ y sea W una palabra cualquiera sobre Σ . Consideremos el cálculo proposicional parcial \mathcal{B}_W consistente en los mismos axiomas de \mathcal{B}_{σ_0} y en adición los siguientes dos axiomas especializados:

(B7') $\dashv [p_1 \supset p_1]$.

(B8') $[W' \supset \dashv [p_1 \supset p_1]]$.

Tendremos, entonces el siguiente resultado, obtenido por Linial y Post [9] en 1949:

Teorema 33 $\vdash_{\mathcal{B}_{\sigma_0}} W'$ si y solo si \mathcal{B}_W no es independiente.

Demuestra: Supongamos primero que $\vdash_{\mathcal{B}_{\sigma_0}} W'$. Sea \mathcal{D}_W el cálculo proposicional parcial obtenido a partir de \mathcal{B}_W al suprimir el axioma (B7') de \mathcal{B}_W . Entonces, $\vdash_{\mathcal{D}_W} W'$. Luego, aplicando el axioma (B8') y la regla del *modus ponens*, tendremos $\vdash_{\mathcal{D}_W} \dashv [p_1 \supset p_1]$, de donde \mathcal{B}_W no es independiente.

Recíprocamente, supongamos que \mathcal{B}_W no es independiente. Ahora, es relativamente fácil establecer que \mathcal{B}_{σ_0} es independiente. Más aún, al adicionarle el axioma (B7') a \mathcal{B}_{σ_0} no cambia nada esencial: solamente las palabras obtenidas por sustitución directa en el axioma (B7') serán teoremas nuevos. Luego, $\vdash_{\mathcal{D}_W} \dashv [p_1 \supset p_1]$, pues ninguno de los axiomas (B1) al (B6) es derivable cuando el axioma (B8') se adiciona. Pero $\dashv [p_1 \supset p_1]$ no es regular, de donde este teorema de \mathcal{D}_W debió ser obtenido específicamente del axioma (B8') y la regla del *modus ponens*. En consecuencia, deberemos tener el antecedente $\vdash_{\mathcal{D}_W} W'$. Luego, $\vdash_{\mathcal{B}_{\sigma_0}} W'$. ■

La consecuencia de este resultado es también asombrosa: en general no es posible constatar algorítmicamente la independencia de los axiomas, en un cálculo proposicional parcial.

Corolario 34 (Linial, Post) *El problema de determinar si un cálculo proposicional parcial \mathcal{B} es o no independiente, es recursivamente irresoluble.*

Demuestra: Si suponemos lo contrario, entonces existiría un procedimiento recursivo para establecer la veracidad o falsedad de la afirmación “ \mathcal{B}_W no es independiente”, del enunciado del teorema 33. Pero en consecuencia, también dispondríamos de un procedimiento recursivo para establecer la veracidad o falsedad de la afirmación “ $\vdash_{\mathcal{B}_{\sigma_0}} W'$ ” del mismo teorema 33. Sin embargo, esto implicaría que el cálculo proposicional parcial \mathcal{B}_{σ_0} es completo, cosa que es falsa de antemano. ■

Referencias

- [1] Church, A. (1936) “A note on the Entscheidungsproblem”, *The Journal of Symbolic Logic* 1(1): 40–41.

- [2] Church, A. (1956) *Introduction to Mathematical Logic*, Vol. 1. Princeton University Press, Princeton NJ.
- [3] Davis, M. (1958) *Computability and Unsolvability*. Dover Publications, New York.
- [4] Davis, M.; Weyuker, E. (1983) *Computability, Complexity, and Languages*. Academic Press, Boston.
- [5] Dekker, J.C.E. (1976) *Equivalencia Recursiva*. Editorial CAEM, San José.
- [6] Hennie, F. (1977) *Introduction to Computability*. Addison-Wesley, Boston MA.
- [7] Hopcroft, J.; Ullman, J. (1979) *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading MA.
- [8] Kfoury, A.J.; Moll, R.N.; Arbib, M.A. (1982) *A Programming Approach to Computability*. Springer-Verlag, New York.
- [9] Linial, S.; Post, E. (1949) “Recursive unsolvability of the deductibility, Tarsky’s completeness and independence of axioms problems of the propositional calculus”, *Bulletin of the American Mathematical Society* **55**: 50.
- [10] Piza, E. (2001) *Aritmética Recursiva y Algunas de sus Aplicaciones*, Editorial CIMPA, San José.
- [11] Rogers, H. (1967) *Theory of Recursive Function and Effective Computability*. McGraw-Hill, New York.
- [12] Soare, R. (1980) *Recursive Enumerate Sets and Degrees: a Study of Computable Functions and Computable Generated Sets*. Springer-Verlag, Berlin.
- [13] Turing, A. (1936) “On computable numbers, with an application to the “Entscheidungsproblem”, *Proceedings of the London Mathematical Society* **42**(2) 230–265.

